A Survey on Improving QoS in Service Level Agreement for Cloud Computing Environment

https://doi.org/10.3991/ijim.v15i21.26379

Afaf Edinat, Rizik Al-Sayyed(✉), Amjad Hudaib
The University of Jordan, Amman, Jordan
r.alsayyed@ju.edu.jo

Abstract—Cloud computing is considered one of the most important techniques in the field of distributed computing which contributes to maintain increased scalability and flexibility in computer processing. This is achieved because it, using the Internet, provides different resources and shared services with minimum costs. Cloud service providers (CSPs) offer many different services to their customers, where the customers' needs are met seeking the highest levels of quality at the lowest considerate prices. The relationship between CSPs and customers must be determined in a formal agreement, and to ensure how the QoS between them will be fulfilled, a clear Service Level Agreement (SLA) must be called for. Several previously-proposed models used in the literature to improve the QoS in the SLA for cloud computing and to face many of the challenges in the SLA are reviewed in this paper. We also addressed the challenges that are related to the violations of SLAs, and how overcoming them will enhance customers' satisfaction. Furthermore, we proposed a model based on Deep Reinforcement Learning (DRL) and an enhanced DRL agent (EDRLA). In this model, and by optimizing the learning process in EDRLA, proposed agents would be able to have optimal CSPs by improving the learning process in EDRLA. This improvement will be reflected in the agent's performance and considerably affect it, especially in identifying cloud computing requirements based on the QoS metrics.

Keywords—cloud computing, quality of service, service level agreement, reinforcement learning, deep reinforcement learning

1 Introduction

Cloud computing, being one major approach among other distributed computing approaches, works to enhance computer scalability and flexibility, and this is achieved by providing different resources and standard services via the Internet [1], as it easily can reduce the expenses. Cloud Computing also provides many deployment models based on customers' needs and demands. Accordingly, the effect of cloud computing is considered enormous, as small companies, using cloud computing, may effortlessly and smoothly grow and expand over the globe with highest qualities and lowest expenses.

In addition, using cloud computing, both, companies and individuals, may quickly and cheaply develop software and hardware of their own, in order to promote their goods in the market [2]. Cloud services are delivered by cloud service providers (CSPs) that provide a wide range of services for their customers, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Program as a Service (SaaS), Storage as a Service (STaaaS), and Security as a service (SECaaS), the test environment as a service (TEaaS), and many more. While CSP tries to increase its income to the max, customers aim to achieve the best quality of services (QoS) at the lowest prices [1]. Therefore, a legal agreement should determine and clarify the relationship between CSP and the customers. To ensure that the objectives are met, an explicit Service Level Agreement (SLA) construction must be established.

The SLA manages the interactions and the relationship between CSP's and the customer by defining the terms and requirements [3]. The SLAs provide all QoS needed information about the QoS. With many CSPs started to provide a wide variety of cloud services, cloud customers cannot determine what services to use and what kind of basis to choose. Therefore, establishing an SLA is essential for cloud negotiations. Negotiating an SLA between cloud parties helps in defining QoS requirements for critical operations that are based on services. Currently, there is no Standard Framework that helps customers rating and classifying the services provided by the cloud based on the customers' needs [4].

Improving the QoS is an important research problem in CC. This research is highly important because we need to know what other researchers in CC that have done in this field; improving the QoS in SLA for CC environment, in order to help set up and define problems that exist in the research. Actually, the QoS in CC depends on the optimal selection of SLAs provided by the optimal CSP selection. Therefore, the SLA must contain the optimal QoS, which must meet the requirements required by customer needs. Many researches were introduced to evaluate QoS metrics in CC SLAs, select the optimal QoS in SLAs, and select the optimal CSP. The traditional QoS usually quantify the QoS metrics and simply displays resource selection options to the users. Other research of QoS solutions proposed using QoS engine, information service, SLAs, monitoring service, and reservation modules and frameworks. While other studies focused on providing a QoS-aware resource brokering framework that can deliver optimal performances for applications. Therefore, we will consider in our paper to proposed a new model Enhanced Deep Reinforcement Agent(EDRLA) to improve the selection of CC resources according to QoS metrics in the SLA, and to prevent SLA violations and predict SLA violation in future.

In this research, the proposed model forms a combination of Deep Neural Network (DNN), Generative Adversarial Network (GAN) and Killer Whale Optimization Algorithm (KWOA). According to our knowledge there is no such a combination proposed in the related work; as such, it comes the novelty of this work; this will be shown in section six. The proposed model EDRLA improves the selection of SLAs according to QoS parameters and selects the optimal CSP that offers the required services for the customer without any violations and predicts any violation for QoS in future.

There are not enough studies on how to select the optimal QoS in cloud computing, and based on this, the survey reviews previous studies conducted over the past decade.

The remainder of this paper is structured as follows: Section 2 provides a brief description of cloud engineering. Section 3 briefly describes the cloud computing architecture. Section 4 provides a brief discussion of Service Level Agreements (SLAs). Section 5 provides a brief description of QoS in the cloud. Section 6 provides some related work. Section 7 presents the models of analysis. Section 8 provides a brief description of our proposed model, and the last section presents the conclusion of this paper.

2 Survey methodology

In this section, we applied the rules of "Gap Analysis" steps to highlights the shortcomings and opportunities for improving the QoS in SLA for CC environment. These steps include: analyzing the current state, identify the ideal future state, finding the gap and evaluating solutions, and create and implementing a plan to bridge the gap. For this, we present our survey methodology in details following the stages mentioned below:

1. Identifying the statement of the problem: in this stage, we defined our statement of the problem of enhancing QoS selection for cloud computing parties.
2. Collecting related work and literature to identify the gaps: in this stage, we have gathered previous work done on improving QoS selection for cloud computing parties between the years of 2012 and 2021.
3. Analyzing related work: in this stage, we analyzed the related work done on cloud computing architecture, cloud computing deployments, cloud computing service models, SLAs template, and QoS parameters used in cloud computing.
4. Analyzing and discussing finding models, frameworks, and algorithms in the related work: in this stage, we discussed the previous studies in the last decade that contain various QoS advancements in cloud computing using reinforcement learning, deep reinforcement learning, or different machine learning algorithms.
5. Analyzing the statistical data: in this stage, we explored most of the QoS parameters and their percentages that were used in the related work done in the last decade.
6. Proposing a model for dynamic SLAs using an enhanced DRL to select the optimal QoS for cloud computing parties and customers: in this stage, we described the proposed model for dynamic SLAs with an enhanced DRL agent (EDRLA). We used an improved deep Q-learning agent to select the optimal QoS for cloud computing parties. Choosing the optimal action represents the selection of the QoS that meets the customer's requirements with maximum rewards.
7. Conclusion and findings: in this stage, the agent in the proposed model will help the enhanced deep Q-learning to reach the optimal action and improve the learning process in a deep Q-learning algorithm. This improvement will affect the deep learning of the agent's training that is related to the agent's performance in selecting optimal cloud computing requirements with maximum rewards, according to QoS.

Figure 1 illustrates the stages of our survey methodology.

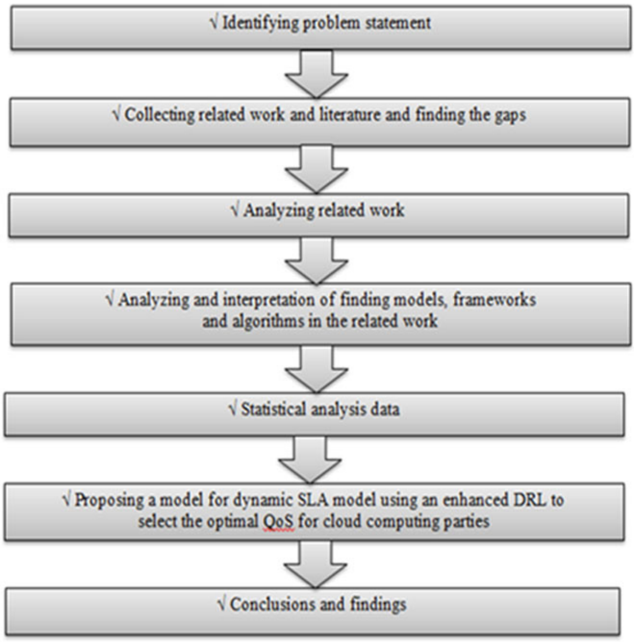
√ Identifying problem statement
√ Collecting related work and literature and finding the gaps
√ Analyzing related work
√ Analyzing and interpretation of finding models, frameworks and algorithms in the related work
√ Statistical analysis data
√ Proposing a model for dynamic SLA model using an enhanced DRL to select the optimal QoS for cloud computing parties
√ Conclusions and findings

Fig. 1. Survey methodology

3 Cloud computing architecture

Mainly, cloud computing aims to efficiently use the distributed resources, integrate them to achieve higher productivity, and increase their ability to solve large-scale complex account problems. Cloud computing focuses on virtualization, scalability, interoperability, and QoS [5]. The concept of cloud computing was first introduced in the 1960s by John McCarthy. He thought that one day, that account will be recognized good for the wider public. The origin of the term 'cloud' comes from the world of telecommunications, as telecom companies have begun to provide VPN services with similar QoS at a lower cost. A growing number of servers and networking infrastructure are now included in cloud computing, as it is expanding day by day [5]. Cloud computing architecture refers to the various components that create the whole cloud system, regarding the databases, program capabilities, or the applications that were designed to provide cloud power resources to solve business problems. The completed cloud computing infrastructure design provides customers with network security, uninterrupted services, and high bandwidth using the Internet. Figure 2 illustrates the generic cloud architecture which includes the main components of cloud computing, mainly: infrastructure, storage, service, application, management, and security [2].

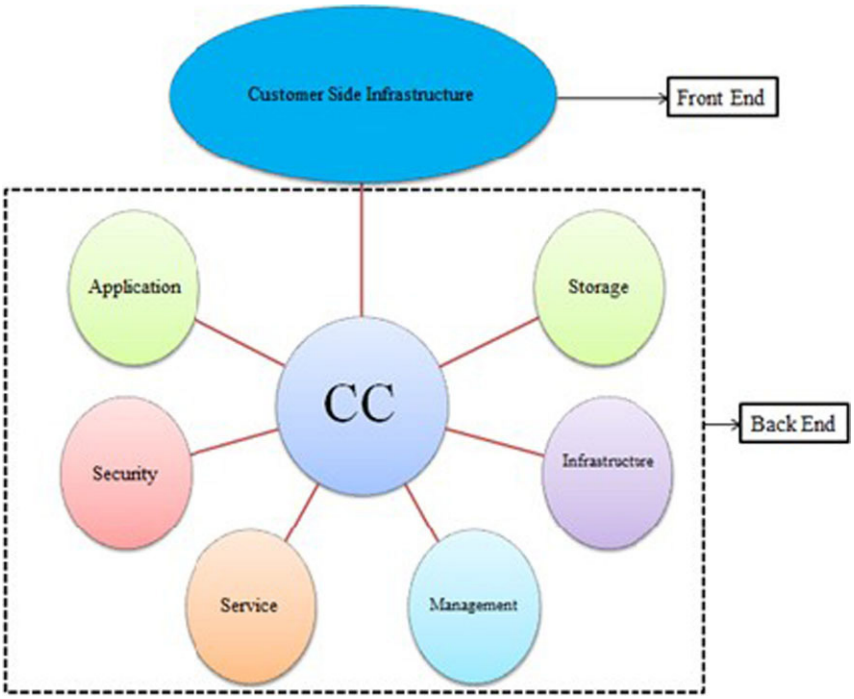

Fig. 2. A generic cloud computing architecture

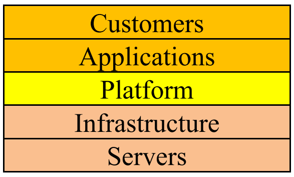
Customers
Applications
Platform
Infrastructure
Servers

Fig. 3. Different layers of cloud computing architecture [4]

Cloud computing systems can be divided into two parts: the front end and the back end, and both are connected and can be accessed via the Internet. The front end is what the customers can view, and the back end is the system's cloud [5].

3.1 Models of cloud computing services

Figure 3 shows the different layers of cloud computing architecture. Another type of cloud deployment model that can be combined with cloud computing service models is based on cloud delivery. Cloud service models are software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).

Software as a Service (SaaS). SaaS is a software delivery model by which a fully functional subscription-based model is delivered online to the customers, where the end customer can usually access SaaS offers online. Figure 4 shows the responsibility and management of the cloud customer and the CSP that manage various components of the offering SaaS [2].

Platform as a Service (PaaS). PaaS provides a primary computing platform that is based on the cloud infrastructure. It contains all software programs that the customer usually requests to be posted on. In addition, the customer can install the software he/she needs over these programs. This service allows developers to access all systems and environments required for the lifecycle of the software and to provide the development, testing, and deployment for it [2] [5]. Figure 5 illustrates the responsibility and management of the cloud customer and CSP in the PaaS service model.

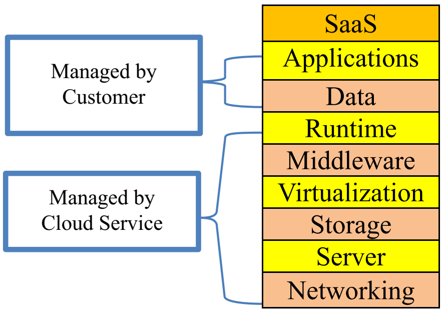

Fig. 4. SaaS service model

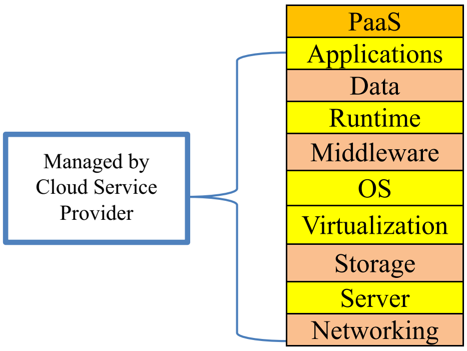

Fig. 5. PaaS service model

Infrastructure as a Service (IaaS). IaaS provides the required infrastructure as a service, where cloud customers are provided with the lowest level of the services with the highest level of flexibility [2]. In this service model, there is no need to purchase the required servers, data centers, or network resources, because customers buy what they need only. Moreover, cloud customers have access to the hardware infrastructure, and they can choose which operating system and software programs to use for meeting their needs and requirements [2] [5]. Figure 6 illustrates the responsibility and management of the cloud customer and CSP in the IaaS service model.

3.2 Deployment of cloud computing models

Cloud computing has four deployment models, mainly: Public Cloud, Private Cloud, Community Cloud, and Hybrid Cloud.

Public cloud. Public cloud allows customers to access the cloud via interfaces using web browsers. In this cloud, the physical structure is out of location, and customers only need to pay for the duration they use the service [2] [5].

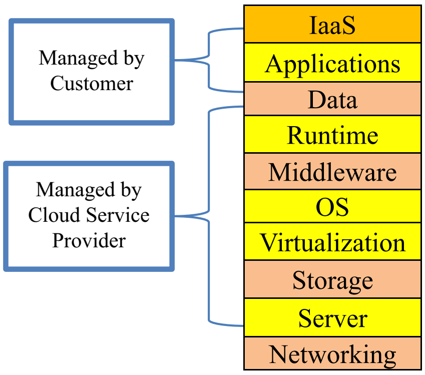

Fig. 6. IaaS service model

Private cloud. Private cloud operations are managed within one organization. This makes it is easier to manage security, maintenance, and upgrades, and this is considered the main advantage of it. It also provides more control over deployment as all resources and applications are governed by one entity [2] [5].

Community cloud. Community cloud means that many companies share the same goals and objectives, and that they build and share cloud infrastructure, requirements, and policies. An external service provider or community can host cloud infrastructure because, in the community cloud, all resources are shared between different companies and customers [2] [5].

Hybrid cloud. A hybrid cloud is a combination of two or more types of the cloud. In this model, the cloud is associated between one or more external cloud services. It is a more secure way to control data and applications, and it allows parties to access information via the Internet. The hybrid cloud can handle any overrides without giving access to third-party data centers [2] [5].

4 Service level agreements (SLAs)

The SLAs provide information regarding the quality of cloud services and the way they are used. These documents often have long texts and contain specific terminologies on the domains of the clouds [6], and the specifications of the agreement can be presented using an extensible language, such as XML. The complex nature of cloud computing is reflected in the complexity of creating SLAs. In addition, the nature of cloud computing is very dynamic; therefore, it has to monitor the QoS goals mentioned in the SLA regularly as they may change. Managing SLAs is very complicated because of the complex nature of the cloud, and the cloud is complex because of the various service providers who offer multi-tenancy with huge, distributed capabilities of resource sharing [7]. An SLA has been introduced since 1980 to manage QoS in the telecommunications sector, and it was designed to gain a shared understanding of QoS, priorities, and responsibilities. The SLA sheds light on many aspects of the relationship between the customer and the CSP, such as service performance, customer care, service supplying, and billing. The critical element of the SLA is the service level objective (SLO) that describes the level of service on which the parties agreed. It usually defines a set of indicators such as the availability, performance, and the reliability of services. The violations also set all the penalties that may be imposed when the service does not meet the objectives identified in the SLA [3].

4.1 SLA and SLA template role in cloud markets

SLA makes accurate measurements and enables the described resource parameter values and QoS to be audited. The definition of SLA accurately shows how service supplying is determined. During service implementation, CSP and customers use SLAs to monitor measurable service features and avoid violations to be committed by both parties [8]. The SLA template formally refers to the agreement that the CSP makes available to the market to consider customers' needs and demands. The SLA template also describes the content of the agreement that is acceptable to the CSP during communicating with the customers.

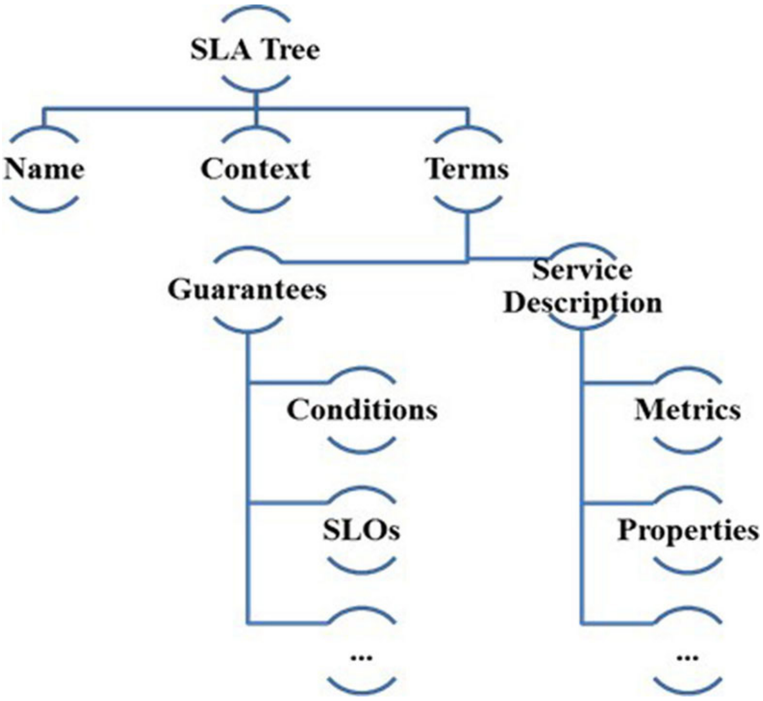

Fig. 7. SLA tree structure

Consequently, the model accurately describes the resource availability, supply plan, and QoS, and the customer will accordingly decide which offer is best suited to meet their needs. Reviewing SLA templates as a service leads to either setting up the agreement or negotiating with one or more of the CSPs [7] [8]. Well-designed SLAs contribute to avoid conflicts and violations, and it can facilitate finding a solution for an issue before it becomes more complicated. SLAs are not seen as end-user documents, but as an automated process that helps resources to be monitored and scheduled. Conversely, cloud markets view SLAs as fixed documents where processing is not allowed. SLAs represent nested tree structures, as shown in Figure 7, including heterogeneous and infinite properties in terms of length and content [8].

The SLA lifecycle goes through three steps: (1) The SLA setting up phase, in which the definitions and specifications are identified, (2) The SLA validation phase, in which the correct actions that are taken to make the cloud service more resilient are demonstrated, and finally (3) The SLA monitoring phase, in which violations are detected and the QoS reliability is verified [3]. Moreover, manual service selection task is considered costly, due to the rapidly increasing number of offered services, and lacking standard specifications and an effective management. This prevents cloud computing to be successfully implemented in a way that meets the specifications set in the SLA in order to achieve QoS objectives [9].

4.2 SLA template construction

According to [8], an SLA consists of three sections: (1) service description, (2) guarantees, and (3) a section where pieces of information regarding the involved parties and/or the provided services are found. The steps to design the SLA template model can be summarized as follows:

1. The original WS-Agreement format comes with XML encoding. Distribute the XML sample to JSON.
2. Use (1) to create an SLA template data model.
3. Create a database schema according to (2).
4. Retrieve service descriptions from the marketplace.
5. Order data from (4) to the types of service.
6. Create fictional information, arranged by (5).
7. Mixing information from (5) and (6) to randomly create lists.
8. Download (7) in CSV files.
9. Upload (8) to the database.

The following example [10] explains that an SLA confirmation is obtained using the XML files that were exchanged between the CSP and the customer. The content of the XML files matches the context of an agreement, the terms of control, and the terms of guarantee. All XML files used in this method follow the pattern of the WS-Agreement requirements [8] [10]. The requirements of the WS-Agreement are specifically designed for the web services in order to define the terms and conditions of using their services. The context section in the agreement explains the details of the CSP, and the start and expiration time refer to the agreement's lifetime. If the customer's request comes after the expiration time, it will be considered a violation, and both parties of the SLA will be notified for action [10].

The term 'monitoring' indicates that abovementioned requirements are the parameters of performance, and based on them, a specific customer application can be selected with the help of the learning system. The section that dictates the terms of guarantee identifies the resources (VM) of the customer application based on the SLA [10]. Because so many CSPs started offering various cloud services, cloud customers are unable to decide which services they should use and what is the basis of their choice; therefore, an SLA is needed to negotiate between cloud parties.

Negotiating an SLA between cloud parties helps in defining the requirements of QoS to have critical operations based on the services. Currently, there is no standard framework that helps customers to rate and classify the services provided by the cloud based on their own needs. However, QoS parameters are seen as the main components of SLAs that work to achieve a set of Service Level Objectives (SLOs) for the aim of satisfying customers' needs and preventing any of SLA violations to happen [11]. So, a CSP must maintain QoS parameters in the SLAs. Moreover, there is no standard number of parameters for QoS as it depends on both the SLOs and customer requirements. These QoS parameters are needed to be measured and monitored throughout the service supply.

5 Quality of service (QoS) in cloud computing

As more and more users share their applications on cloud environments every day, SLAs between customers and CSPs become a more significant component to consider. Because cloud computing is very dynamic, continuous monitoring of QoS features is needed to improve the process of selecting resources, and to reduce the impact of SLA violations [3] [7] [12]. Consequently, Cloud computing faces the challenge of providing appropriate QoS for its cloud services. QoS is a key component for guiding the non-functional attributes of the quality of service, such as response time, price, performance, or safety. Therefore, there is a need to develop structures to appropriately fulfil to QoS requirements.

The architecture must be able to resize the resources available for any hosted program in a dynamic way. That is, it must secure optimal resources, by providing each hosted program with many adequate resources to ensure that the SLA is not violated [13]. This section highlights how important QoS parameters for cloud computing are, specifies the current attributes of qualities according to customers' requirements, and sets standards to measure any service quality deviations from what customers expect and require. Cloud computing services face several QoS challenges, and many different techniques were proposed to solve different types of challenges. However, there is insufficient research done trying to develop a framework that can manage the overall QoS for all cloud computing models [14]. In a nutshell, QoS is a significant feature in many use cases because the CSP must fully satisfy the specified requirements [15].

5.1 QoS and SLA

QoS is crucial in the process of creating services because reducing the QoS in the service can seriously disturb the QoS for a complete configuration. CSPs want to ensure that adequate resources are provided to ensure that QoS requirements for customers, such as price, response time, and budget restrictions, are met and fulfilled. As a result, CSPs aim to get the assurance that violations are avoided by supplying the resources on time and in a dynamic way. The success of cloud computing infrastructures depends on how these infrastructures will discover computing platforms that meet the various requirements of customers' resources and services [13]. These parameters will be defined, identified, and described in the SLAs based on QoS requirements or standards such as scalability, high availability, confidence, pride, and security. Generally, the SLA needs to be carefully evaluated in terms of the characteristics of the required resources. An essential challenge that CSPs face is automating the management of the resources, considering both the resource costs and the requirements of a high level of service quality for hosted programs [13] [14].

5.2 QoS metrics for cloud computing services evaluation

The metrics of cloud computing evaluation depend on the QoS parameters, as the services in the cloud need to be evaluated. According to [15] [16], a recognized evaluation metrics corresponds to match cloud service options. Collected metrics have four aspects of cloud services, mainly: performance metrics (which relate to responding to the time or timeliness), economic metrics (price and elasticity), security metrics (data security, authentication), and general metrics (availability, scalability, reliability).

There are many CSPs that provide many cloud services with different costs and different levels of functionality. With the increasing diversity of cloud services, and with having the opportunity to own and moderate almost unlimited cloud account resources, it is difficult for cloud customers to find the perfect CSPs to meet their QoS requirements. Accordingly, to be able to determine the most suitable provider among different CSPs based on the optimal choice of QoS requirements, customers must have a method or a technique for assessing the critical performance standards for QoS, which is necessary for their programs. To evaluate a computer system, metrics are selected based on the requirements [15] [16]. The following section briefly describes the related work that discusses the methods, techniques, and frameworks for improving the QoS in the cloud.

6 Related work

This survey reviews the previous studies conducted over the past decade that suggest various QoS improvements in cloud computing by implementing deep reinforcement learning and different algorithms that utilize machine learning. The main purpose of the different algorithms, techniques, and methods was to address different QoS parameters related to different cloud-computing models, thus ensuring that each requirement mentioned in the SLA agreement is met via metrics associated with QoS. Moreover, this model aims at increasing trust levels of customers by minimizing the SLA violations while also achieving a considerable profit fir CSP [14].

SLA in the PaaS platform of cloud computing uses variables such as performance, Customers' satisfaction level, cost, security, and SLA violations to analyze and evaluate the models. QoS terms must be specified in the models of cloud computing, while the financial terms must be agreed upon in an SLA [17]. Given that QoS is an essential feature in many instances, the specific requirements of CSPs', such as latency, throughput, among other requirements, must be fulfilled. As a result, QoS is guaranteed to be provided to the cloud customer [18]. However, due to the complex nature of cloud computing, determining the efficacy of QoS in customer service was not possible before the actual implementation of the service. And as a result of the rise in public Cloud Service Providers [CSP], it becomes even more difficult when CSPs choose to meet their QoS needs.

The customer has to choose between many CSPs, each offering very similar services but with different prices and power capacities, and with many different options. The customer does not truly understand the services provided by CSPs and sees them as black boxes. Therefore, evaluating the QoS before deployment is crucial. After evaluating cloud services, it is imperative to choose the right metrics. The choice of metrics plays a vital role in implementation analysis as seen by previous research examining older computing devices. It is, therefore, necessary to use appropriate measures when assessing practices. However, there is no standard interpretation regarding metrics for assessing cloud products and services. QoS metrics are vital for choosing the right CSP as well as improving the efficiency of resources, and it is necessary to establish a set of standardized metrics used to assess QoS in order to ensure a high quality.

One study [19] provided several methods to assess QoS metrics in cloud-computing SLAs, such as selecting the optimal QoS in SLAs, and enhancing SLAs. Typically, QoS sets the standards and provides users with many choices of resources. Other pieces of research on QoS suggest using the QoS engine, information service, SLA, monitoring service, and reservation units and frameworks. Whereas other research sought to offer a framework that focuses on resources for QoS that can optimize application performance. An SLA-based approach was proposed in another study to guarantee CPU performance of cloud services. A resource level measure constitutes part of the SLA rather than throughput, response time, and availability. Another study devised a system for admission control and task assignment; this system assigns the tasks in accordance with the anticipated outcome through a machine-learning algorithm embedded in the framework [10].

This section briefly discusses all the methods, techniques, algorithms, and frameworks used and proposed to improve the QoS and its metrics. The authors of [9] presented an automated system for matching SLAs between parties and, using a variety of machine-learning algorithms, found linguistically equivalent SLA elements through SLAs. Plus, automatic selection of optimal service offers tailored to clients is achieved through this method. The main goal of this paper was to reduce the cost of manual generation of SLA mappings and finding optimal services. In [10], the authors established a Learning Automata QoS (LAQ) framework based on machine learning that can address some of the challenges of different cloud applications. The proposed LAQ cost framework ensures that computing resources are utilized, and customer applications are not overlooked. Guaranteeing service provision is possible through continuous monitoring of suppliers and identification of various QoS that can provide services upon request. The proposed framework helps to establish guarantees by using metrics to provide cloud services that support QoS. LAQ IaaS cloud service are considered useful for computer applications. The LAQ optimizes virtual computing devices and makes sure that all of the customer's requirements are met. The main objective of this study was to boost performance through means of advancing QoS metrics, response time, parallel execution speed, and task prioritization.

In [12], the authors presented a new technique to increase client satisfaction by minimizing violations of the SLA, and that technique is based on learning automata. The costumer's characteristics determine the amount of reduction possible. These characteristics are related to agreed QoS requirements between parties in SLAs. This study explored the possibility of improving the satisfaction level of customers with low willingness to Pay for Service (WTP) and risk avoidance. In [20], a quality model named S3MQual (Service Measurement Metrics Model) was introduced, and the researchers identified quality models for cloud-computing services. The main two factors that drive customer's satisfaction were the quality of the services and the continuity of the operation. The proposed S3MQual model was developed to include the following twelve important attributes, "Accountability, Availability, Maintainability, Scalability Elasticity, Performance, Security Privacy, Usability, Reliability, Features and Functionality, Recoverability, Empathy, and Compliance." The organization needs to take these attributes into account in order to have an efficient QoS. In [21], the researchers proposed a deep learning-based algorithm for service composition (DLSC) that accurately recommended the suitable cloud services based on QoS criteria. The LSTM Deep Learning Network was paired with a Particle Swarm Optimization (PSO) with the goal of utilizing the LSTM capabilities of recording and considering the long-term correlations between the time series of multivariate QoS properties and the PSO algorithm ability to serve many purposes and solve problems. The results of the experiment showed that the proposed framework led to significant improvements in the field of cloud services. This study focused on selecting the best contractors, reducing QoS parameters costs and optimizing response time and throughput.

Other researchers proposed a dynamically evolving model that continuously updated cloud-computing SLA making the SLA highly flexible and adaptable and minimizing the chance for costly violations [22]. Given the dynamic nature of QoS parameters, the desire to continuously change the condition of both CSPs and clients, along with the constantly changing policies of the companies regarding cloud computing, it became increasingly necessary to regularly modify the SLA. Such problems are certain to become more widespread among cloud-computing companies. The three levels of the proposed SLA are the SLA negotiation level, the SLA control level, and the SLA enforcement level. Performance, availability, reliability, and bandwidth are key in determining the efficiency of the QoS in SLAs. In [23], the researchers present a new machine-learning algorithm for reducing energy consumption and improving resource usage. This algorithm monitors the customer's resource requirements alteration to predict a physical device (PM). This algorithm took measurements to prevent server overload such as improving the use of PMs, reducing the number of migrations, suspending idle servers in order to lower power consumption levels, and suggesting efficient resource management for the cloud. Through this machine-learning algorithm, the optimal action is chosen among several options.

In another study, the researchers introduced an integrated DevOps framework responsible for design-time modeling as well as optimization, and runtime control [24], so as to minimize the costs of implementing cloud applications that have embedded QoS guarantee. The modeling of the application chosen to evaluate included important aspects of cloud computing, such as variable workload, congestion due to multiple rents, and performance fluctuations. In [25], the researchers developed a tool to identify and predict scenarios that require corrective action. Bayesian dynamic Network (BDN) collected data to calculate dependencies, and it also used this data to get continuous updates. Correlation values are then recorded into a neural network of Long Short-Term Memory (LSTM) which is used for future forecasting, an example of how SLA for cloud services is improved by processing data. And if problems arose with the SLA, it would compare the incident to similar ones to choose the best course of action. Suitable procedures were, subsequently, determined for each incident using the reinforcement learning (RL) approach. To evaluate the performance of the whole working system, the QoS, response time, and SLA violations were used.

In [26], the authors conducted extensive research regarding Long Short-Term Memory (LSTM) and used it to predict the accurate number of requests that occurred the following time. They also applied it on Reinforcement Learning (RL) to find and follow the optimal procedures for expanding virtual machines. The experiments were performed as part of two real workload tracking processes, as the experiments results proved that this approach ensured consistent operation of virtual machines and reduction of SLA violations. The LSTM network had better accuracy in forecasting orders and allowed the system to allot resources that would be needed in advance. The RL also decided whether the system needed to schedule VMs in order to prevent unnecessary resource scheduling. The RL determined the best course of action based on its experience in processing previous data and the conditions of the current system. To evaluate the performance of the whole working system, the QoS, performance, response time, and CPU utilization were used this time. RL was also implemented in [27] where the authors used it to develop an algorithm that solved the problem of choosing a data center. A mathematical model for the price range for customers was created to arrive at the most efficient solution taking into account the cost of computing resources and network resources. In this study, these prices were considered as part of the QoS.

In [28], the researcher compared the performance of two different methods for machine-learning in predicting violations of an SLA, namely Naive Bayes and Random Forest Classifiers. The classification of the task became challenging, and several reconfiguration methods were used to overcome these challenges, such as Random Over, Under Sampling, SMOTH, Near Miss (1,2,3), etc. Availability was considered to be QoS in this paper, and it was recommended to reduce violations. In [29], the author presented a model for scheduling and admissions management, the goal of which was to maximize profit, utilize resources and guarantee the satisfaction of customers. There were several components of this model including a SaaS provider, users, access control, scheduling system, SaaS providers, and IaaS providers; components of the system included application layer and platform layer functions. Users requested software solutions from SaaS providers through submitting their QoS requirements. Based on the acceptance control, the platform layer could make sense of customers' QoS factors such as cost, refresh time, process time, and availability. It determined whether the request would be accepted or rejected, depending on the capacity, availability, and price of virtual machines. Subsequently, the scheduling procedure followed the acceptance control decision. In [30], the PaaS is considered a key element in the cloud compass, which is the default container. A virtual container is a group of programs with a logical hierarchy of components. Although the Infinite Hierarchy Model provided more flexibility in defining virtual containers, it increased the complexities of model management.

The authors of [30] introduced Cloud Compass, which is the PaaS cloud system responsible for managing the entire lifecycle of resources. This compass featured an extension of the WS-Agreement of the SLA specifications. It was, therefore, designed to meet the specific needs of cloud computing applications in the field of electronic science. It addressed issues that other research projects overlooked. Finally, a comprehensive business model was presented to provide a paradigm for the platform to support flexible deployment. The authors of [31] outlined a framework for admission control and scheduling algorithms to efficiently assign resources and thus enhance profitability and customer satisfaction. The innovative framework for admission control and scheduling algorithms both used public cloud resources to achieve their goals. Simulation results of this framework proved that these algorithms lead to up to 40% cost reduction. As a result of the increasing number of traditional applications moving to the cloud, cloud-computing service providers nowadays face challenges to maintain load balance without sacrificing performance.

In [32], the authors presented a framework for online task scheduling consisting of three components, namely task queue, country monitoring, and task scheduling. The main purpose of task scheduling was to perform substantial and dynamic tasks to reduce resources, according to the SLA requirements. Moreover, online task scheduling was formulated as an ideal dynamic issue with some minor limitations. Using DRL decision-making abilities, the DDPG network was adopted to find the optimal routing. It dynamically adapted to uncertainty and volatile workloads, and it reduced average task-response time while maintaining balance between VMs and workloads. The algorithm was evaluated in two real-world workload scenarios and compared to other solutions. In this paper, a design was proposed to enable QoS-aware configuration, enabling the system to smoothly operate with different types of cloud middleware. It achieved QoS through virtualization on software-defined networks (SDN) on the clouds. The interface was based on SLA and it allocated resources by the highest layer in accordance with QoS parameters that were enforced by the bottom layer. This model was described as the "receive which you pay." In cases of unreliable resources, the model identified ways to balance the costs against the reliability. The approach proposed in this study was called the negotiation approach, and it intended to increase cloud providers' profitability by maximizing time utilization; with this approach, the provider analyzed the consumer's choices and managed deadlines; this approach also provided both parties with a negotiation model for to discuss their SLA requirements.

To identify the cloud's access-control requirements, the researchers of [35] devised a model. However, the authentication mechanism and risk engine were implemented before evaluating the model. They also implemented a risk engine as one of its components; the risk engine supported adaptive behaviors and implemented an authentication technique that handled space and time complexities. As for cloud computing SLAs, many papers were published that evaluated and defined QoS in SLAs. Several QoS parameters were considered in the scope of customers' requirements.

In [36], researchers presented a Markov Chain-based monitoring model that took into account memory, CPU, and storage. Other researchers developed a novel and flexible representation-based forecasting method for predicting QoS response time and throughput based on ANN [37]. For an evaluation of the Weight Service Rank Approach (W-SR), paper [38] evaluated accountability, agility, cost, performance, assurance, security, and usability. In [39], researchers set forth an optimal technique for risk-management strategy selection for cloud services; this strategy used these QoS parameters: execution time, availability, cost, reputation. Whereas [40] employed three generic QoS parameters, namely availability, response time. It used throughput in cloud computing to examine several approaches that advocated the employment of machine learning for better resource management, energy efficiency, and security.

The authors of [41] proposed a testbed system for obtaining service data from cloud-hosted software services. Five QoS parameters were examined in this study: response time, availability, throughput, reliability, and latency. [42] Presented a new model that summarized the QoS standards that cloud consumers could use to select cloud services with ease. These parameters were security, user-friendliness, quality, availability and technology. The authors discussed availability and cost as QoS parameters in paper [43]. Paper [44] introduced CloudExp; an environment for modeling and simulating cloud computing. It can be used to evaluate a wide spectrum of cloud components, one of which is element processing. CloudExp took many variables into account including users, availability, cost, performance, and SLA violation. There was a model in [45] with QoS and metrics that targeted general cloud services. CLOUDQUAL contained six QoS parameters i.e., usability, availability, reliability, responsiveness, security, and elasticity. An ANN-based machine-learning method was proposed in paper [46] for determining current application workload and expected QoS requirements for resource allocation. QoS parameters in this study included performance, cost, and the number of violations.

In [47], a collaborative filtering methodology was described for predicting both QoS and cloud service ratings, and it used the parameters of throughput, efficiency, scalability, availability, security, and effectiveness. Authors of [48] proposed heterogeneous measures combining the quantitative and qualitative dimensions of QoS-based cloud service ranking, and the study considered the elements of response time, availability, cost, usability, security and flexibility. Researchers of [49] developed a service-rate control system that is based on reinforcement learning (RL). The system gave probabilistic upper bounds for end-to-end system delays while preventing exploitation of service resources. Performance and violations were considered as QoS parameters in this study. In [50], a coordinator was introduced. This coordinator plays the role of managing the network and accounting for the resources and applications hosted in the cloud. This paper evaluated QoS parameters based on ML algorithms, bandwidth, and response time.

In [51], the authors suggested a DRL-based service composition model for cloud computing that was also QoS-aware, considering logistics, reliability, cost, and time as the QoS parameters. The authors of [52] took a slightly different approach and recommended improving these metrics using a Hybrid Multiple Parallel Queuing Approach. The arrival rate, service rate, and the number of servers were the QoS parameters in this study. In the following section, we review the proposed models in relation to the techniques and list most of the QoS parameters that have been used in prior related works.

Table 1. A summary of QoS parameters and percentages in related work

No.	QoS Parameter	References	Percentage of QoS Used in Related Work
1	Cost	[9],[12],[21],[24],[27],[29],[30],[31],[34],[35],[38],[39],[43],[44],[46],[48],[51],[52]	16%
2	Time	[10],[21],[25],[26],[29],[32],[35],[37],[39],[40],[41],[45],[47],[48],[50],[51],[52]	15%
3	Computation	[10],[21],[23],[26],[33],[34],[36],[37] [40],[41],[47]	10%
4	Communication	[10],[22],[27],[41],[50]	4%
5	Customer Satisfaction	[10],[12],[20],[38],[42],[44]	5%
6	Decreasing SLA Violation	[12],[25],[46],[49]	4%
7	High profit for Providers	[12],[44],[45]	3%
8	Availability	[20],[22],[28],[29],[31],[34],[39],[40],[41],[42],[43],[44],[45],[47],[48]	13%
9	Composability	[20],[29],[30],[31],[38],[45],[47],[48]	7%
10	Performance	[20],[22],[23],[26],[30],[38],[44],[46],[47],[49]	9%
11	Security & Privacy	[20],[35],[38],[42],[45],[47],[48]	6%
12	Reliability	[20],[22],[41],[45],[51]	4%
13	Technology	[33],[34],[36],[39],[42]	4%
Total		114	100%

7 Analysis models

This section discusses all the SLA models proposed in cloud computing for improving QoS. Table 1 lists most of the QoS parameters, their percentages, and references that have been used in the related work over the last decade. Additionally, Figure 8 shows the percentage of QoS in the related work over the last decade.

Table 1 summarizes the QoS parameters, their percentages, as well as the references in which each QoS parameter has occurred parameter in the related works in the previous decade. It is worth mentioning that some references are mentioned more than once because they have used more than one QoS parameter(s).

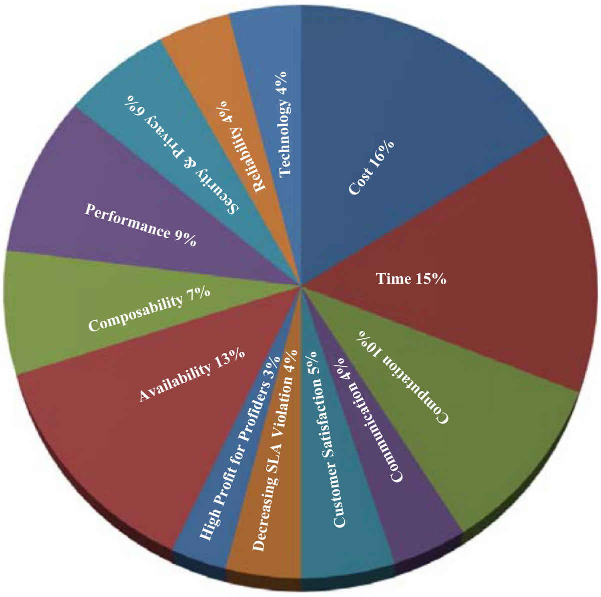

Fig. 8. QoS parameters percentage

8 The proposed model

In cloud computing, as it is very dynamic, the features mentioned in the SLA must be continually monitored to guarantee the QoS and to reduce the impact of SLA violations. SLAs specify the QoS parameters that a CSP must maintain (e.g., Response time, bandwidth, storage, reliability, deadline, throughput, delay, violations, security, and price). For QoS, there is no specific number of parameters, but it depends on the SLOs and on the requirements of the customer. In order to detect SLA violations, these QoS parameters must be measurable and monitored during the service supply process. SLAs must also define penalty clauses when CSPs fail to provide the services agreed to for the customer.

All in all, Scaling and automating SLA can help in coping with dynamic changes in the environment, and this can be achieved through SLA management [11] [53]. To address different types of challenges, different techniques have been proposed; however, there is not enough research done to manage QoS from a general perspective for all cloud computing models in one framework. Common resources in SLAs would be maximally utilized by choosing the optimal ones [8] [54] [55].

Finally, there are a variety of SLAs introduced by different CSPs; therefore, customers need help to find the best QoS parameters and metrics and choose to the appropriate cloud device provider that would fulfil their requirements. With our proposed model, we aim to identify the most common QoS metrics and parameters for both parties in the cloud, while at the same time taking into account the customer's interest and the way in which the agent will manipulate those parameters to identify the optimal resource for the customer by which his/her requirements will be fully fulfilled.

Our proposed model is based on Reinforcement Learning (RL) and Deep Neural Network (DNN) or Deep Learning. RL is one of the most critical research trends for machine learning and has major implications for the development of Artificial Intelligence (AI) [56]. RL is a learning process that allows an agent to make decisions, monitor results, and then automatically align his strategy to achieve the optimal policy based on the best reward. For this learning process to converge, it needs much time to determine the best approach because the whole system needs to be explored and fully understood.

Deep learning has recently been introduced as an important new technology. As the limitations of RL are possible to be resolved, a new era for the development of RL can be moved to, and this is what Deep Reinforcement Learning (DRL) is mainly about [57] [58] [59] [60]. In our proposed model for dynamic SLA model with an enhanced DRL agent (EDRLA), as shown in Figure 9, we used an enhanced deep Q-learning to select the optimal QoS for cloud computing parties. EDRLA combines the proposed DRL and GANs with some improvements using KWOA to optimize and enhance the loss function of GANs. Using the enhanced Baikal loss function for GANs, the proposed model is deep reinforcement learning with GANs EDRLA. This combination can overcome the limitations of DRL, consequently, open a new era for the development of deep reinforcement learning. Thus, it will improve the learning process for the agent to select the optimal QoS in SLAs without violations. The proposed model consists of ten steps explained as follows:

Step 1: A cloud customer prepares a list of potential QoS requirements, and the customer looks through the list and identifies a set of potential cloud QoS.

Steps 2, 3: The customer submits the list of potential QoS requirements selected from the QoS list, and the CSP introduces the QoS offers to the SLA manager. Managing SLAs will involve negotiating and compromising customer's QoS requirements with the offers provided by the CSP, and collecting both the QoS from customers, and the QoS offers from CSP. After negotiating and compromising with the customer, the manager prepares an SLA and submits it to the DRL agent.

Steps 4, 5: The EDRLA, as it uses the enhanced deep Q-learning agent, will select the optimal action with the maximum rewards representing the selection of the QoS that meets the customer's requirements. Then, the initial SLA will be initiated.

Step 6: During this process, needed modifications can be made by the three parties. The customer or the CSP can request any needed modifications to be made, or new QoS parameters to be introduced due to the monitoring process.

Step 7: After making the modifications, if one of the parties disagrees on any of them, the process will restart all over again, from the point in which the SLA manager negotiates and compromise between the two parties.

Step 8: If both parties agree on the modifications, the monitoring process will start comparing the QoS values with the SLA that both parties have agreed upon.

Step 9: If the monitoring process detects any QoS violations, they will be recorded in the database, and the penalty will be imposed according to the SLA, and the new QoS parameters will be submitted to be modified based on the agreed SLA.

Step 10: In the monitoring process, a QoS report about QoS violations and values will be sent to the customer and the CSP to be modified based on the agreed SLA. A QoS report will also be sent to them if the agreement expires or needs to be renewed. In the database of SLA violations, the customer will find the new QoS values corresponding to the SLA violations. Lastly, the new QoS will be provided to the SLA list as a potential QoS for the customer.

In our proposed model, we modify the loss function of the deep Q- learning algorithm with a new loss function called the Baikal loss function [60] [61] [62] and increasing the agent's training speed on selecting the optimal action greatly improves his/her ability to select the optimal QoS parameters in cloud computing environments. In the proposed method, an auxiliary task is added to the neural network, the loss function and the Baikal loss function are applied, and convergence speed is improved.

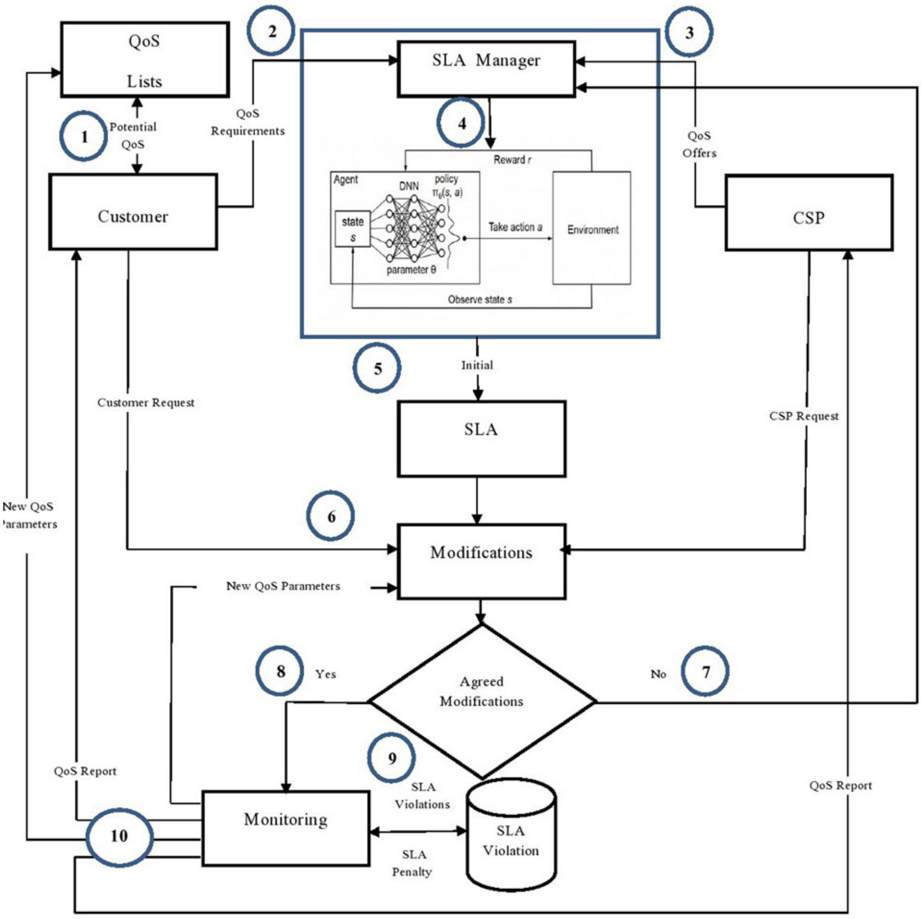

Fig. 9. The proposed model with an enhanced deep Q-learning

9 Conclusion

This paper briefly surveyed and described various models, frameworks, and methods used for improving QoS in SLA in a cloud computing environment. While some of the models provide high-level security measures for consumer data, other models provide penalizations on SLA violations. Additionally, other models increase the satisfaction level of the customers, while the rest of them, compared to other models, raise their level of performance. Moreover, some models improve several SLA QoS parameters. It is crucial to understand the role of the CSPs in providing all essential QoS services in the SLA, in order to design the SLAs between customers and CSPs, as it is expected that customers will receive all required services.

In this paper, we suggested an enhanced model-based Deep Reinforcement Learning Agent (EDRLA). The agent of the proposed model will train the EDRLA to perform the best action and improve the learning process in the deep Q-learning algorithm. This improvement will be effective in the deep learning process of the agent's training that focuses on the agent's performance in selecting the best of the cloud computing requirements according to QoS, with having maximum rewards.

The proposed model EDRLA improves the selection of SLAs according to QoS parameters and selects the optimal CSP that offers the required services for the customer without any violations and predicts any violation for QoS in future. The future work is to implementing the new proposed EDRLA which is a combining model between deep reinforcement learning with GANs using KWOA to optimize the loss function for GANs after enhancing the loss function for GANs [63], and we will compare the experimental results between EDRLA with standard models containing the standard loss function for GANs.

10 References

[1] Al-Roomi, M., Al-Ebrahim, S., Buqrais, S., & Ahmad, I. (2013). Cloud computing pricing models: a survey. *International Journal of Grid and Distributed Computing, 6(5), 93–106.* https://doi.org/10.14257/ijgdc.2013.6.5.09

[2] Hassan, W., Chou, T. S., Tamer, O., Pickard, J., Appiah-Kubi, P., & Pagliari, L. (2019). Cloud computing survey on services, enhancements and challenges in the era of machine learning and data science. *Journal of Computer Engineering & Information Technology, 8(3)*.

[3] Labidi, T., Mtibaa, A., & Brabra, H. (2016). CSLAOnto: a comprehensive ontological SLA model in cloud computing. *Journal on Data Semantics, 5(3), 179–193.* https://doi.org/10.1007/s13740-016-0070-7

[4] Abel, A. (2017). An Effective Framework to Monitor Service Level Agreements in Cloud Computing Environment (Doctoral dissertation, ASTU).

[5] Jadeja, Y., & Modi, K. (2012, March). Cloud computing concepts, architecture, and challenges. *In 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET) (pp. 877–880). IEEE.* https://doi.org/10.1109/ICCEET.2012.6203873

[6] Saha, S., Joshi, K. P., & Gupta, A. (2017, May). A deep learning approach to understanding cloud service level agreements. *In Fifth International IBM Cloud Academy Conference.*

[7] Keserwani, P. K., & Samaddar, S. G. (2017, December). An SLA design with digital forensic capabilities. *In 2017 Ninth International Conference on Advanced Computing (ICoAC) (pp. 109–113). IEEE.* https://doi.org/10.1109/ICoAC.2017.8441460

[8] Stamou, K., Kantere, V., & Morin, J. H. (2013). SLA template filtering: a faceted approach. *In 4th Int. Conf. on Cloud Computing, GRIDs, and Virtualization.*

[9] Redl, C., Breskovic, I., Brandic, I., & Dustdar, S. (2012, September). Automatic SLA matching and provider selection in grid and cloud computing markets. *In 2012 ACM/IEEE 13th International Conference on Grid Computing (pp. 85–94). IEEE.* https://doi.org/10.1109/Grid.2012.18

[10] Misra, S., et al. (2014). Learning automata-based QoS framework for cloud IaaS. *IEEE Transactions on Network and Service Management, 11(1), 15–24.* https://doi.org/10.1109/TNSM.2014.011614.130429

[11] Sahal, R., Khafagy, M. H., & Omara, F. A. (2016). A survey on SLA management for cloud computing and cloud-hosted big data analytic applications. *International Journal of Database Theory and Application, 9(4), 107–118.* https://doi.org/10.14257/ijdta.2016.9.4.10

[12] Morshedlou, H., & Meybodi, M. R. (2014). Decreasing impact of SLA violations: a proactive resource allocation approach for cloud computing environments. *IEEE Transactions on Cloud Computing, 2(2), 156–167.* https://doi.org/10.1109/TCC.2014.2305151

[13] Chana, I., & Singh, S. (2014). Quality of service and service level agreements for cloud environments: Issues and challenges. *In Cloud Computing (pp. 51–72). Springer, Cham.* https://doi.org/10.1007/978-3-319-10530-7_3

[14] Wazir, U., Khan, F. G., & Shah, S. (2016). Service level agreement in cloud computing: a survey. *International Journal of Computer Science and Information Security, 14(6), 324.*

[15] Bardsiri, A. K., & Hashemi, S. M. (2014). QoS metrics for cloud computing services evaluation. *International Journal of Intelligent Systems and Applications, 6(12), 27–33.* https://doi.org/10.5815/ijisa.2014.12.04

[16] Li, Z., O'brien, L., Zhang, H., & Cai, R. (2012, September). On a catalogue of metrics for evaluating commercial cloud services. *In 2012 ACM/IEEE 13th International Conference on Grid Computing (pp. 164–173). IEEE.* https://doi.org/10.1109/Grid.2012.15

[17] Macías, M., & Guitart, J. (2011, December). Client classification policies for SLA negotiation and allocation in shared cloud datacenters. *In International Workshop on Grid Economics and Business Models (pp. 90–104). Springer, Berlin, Heidelberg.* https://doi.org/10.1007/978-3-642-28675-9_7

[18] Zdraveski, D., Janeska, M., & Taleska, S. (2020). Evaluating cloud computing services.

[19] Ellens, W., Akkerboom, J., Litjens, R., & van den Berg, H. (2012, June). Performance of cloud computing centers with multiple priority classes. *In 2012 IEEE Fifth International Conference on Cloud Computing (pp. 245–252). IEEE.* https://doi.org/10.1109/CLOUD.2012.96

[20] Hourani, H., Abdallah, M., & Tamimi, A. A. A, (2019). Proposed Cloud Computing Quality Model: S3MQual (Service Measurement Metrics Model).

[21] Haytamy, S., & Omara, F. (2020). A deep learning-based framework for optimizing cloud consumer QoS-based service composition. *Computing, 1–21.* https://doi.org/10.1007/s00607-019-00784-7

[22] Al-Ghuwairi, A. R., Khalaf, M. N., Al-Yasen, L., Salah, Z., Alsarhan, A., & Baarah, A. H. (2016, March). A dynamic model for automatic updating cloud computing SLA (DSLA). *In Proceedings of the International Conference on Internet of things and Cloud Computing (pp. 1–7).* https://doi.org/10.1145/2896387.2896442

[23] Ranjbari, M., & Torkestani, J. A. (2018). A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers. *Journal of Parallel and Distributed Computing, 113, 55–62.* https://doi.org/10.1016/j.jpdc.2017.10.009

[24] Guerriero, M., Ciavotta, M., Gibilisco, G. P., & Ardagna, D. (2015, September). A model-driven DevOps framework for QoS-aware cloud applications. *In 2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) (pp. 345–351). IEEE.* https://doi.org/10.1109/SYNASC.2015.60

[25] Vakilinia, S., Truchan, C., Kempf, J., & Elbiaze, H. (2018, July). Automated enforcement of SLA for cloud services. *In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 49–56). IEEE.* https://doi.org/10.1109/CLOUD.2018.00014

[26] Zhong, J., Duan, S., & Li, Q. (2019, June). Auto-scaling cloud resources using LSTM and reinforcement learning to guarantee service-level agreements and reduce resource costs. *Journal of Physics: Conference Series, 1237(2), 022033.* https://doi.org/10.1088/1742-6596/1237/2/022033

[27] Li, Q., Peng, Z., Cui, D., He, J., Chen, K., & Zhou, J. (2019, December). Data center selection based on reinforcement learning. *In 2019 4th International Conference on Cloud Computing and Internet of Things (CCIOT) (pp. 14–19). IEEE.* https://doi.org/10.1109/CCIOT48581.2019.8980333

[28] Reyhan, A. H. SLA Violation Prediction in Cloud Computing*: A Machine Learning Perspective [Electronic resource]. arXiv.–2016.–Available at:* https://arxiv. org/pdf/1611.10338. pdf

[29] Wu, L. (2014). SLA-based resource provisioning for management of Cloud-based Software-as-a-Service applications (Doctoral dissertation).

[30] García, A. G., Espert, I. B., & García, V. H. (2014). SLA-driven dynamic cloud resource management. *Future Generation Computer Systems, 31, 1–11.* https://doi.org/10.1016/j.future.2013.10.005

[31] Wu, L., Garg, S. K., & Buyya, R. (2012). SLA-based admission control for a software-as-a-service provider in cloud computing environments. *Journal of Computer and System Sciences, 78(5), 1280–1299.* https://doi.org/10.1016/j.jcss.2011.12.014

[32] Ran, L., Shi, X., & Shang, M. (2019, August). SLAs-Aware Online Task Scheduling Based on Deep Reinforcement Learning Method in Cloud Environment. In 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; *IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 1518–1525). IEEE.* https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00209

[33] Stanik, A., Koerner, M., & Lymberopoulos, L. (2014). SLA-driven federated cloud networking: quality of service for cloud-based software-defined networks. *Procedia Computer Science, 34, 655–660.* https://doi.org/10.1016/j.procs.2014.07.093

[34] Dastjerdi, A. V., & Buyya, R. (2012, May). An autonomous reliability-aware negotiation strategy for cloud computing environments. *In 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (pp. 284–291). IEEE.* https://doi.org/10.1109/CCGrid.2012.101

[35] Younis, Y. A., Kifayat, K., & Merabti, M. (2014). An access control model for cloud computing. *Journal of Information Security and Applications, 19(1), 45–60.* https://doi.org/10.1016/j.jisa.2014.04.003

[36] Chandrasekar, A., Chandrasekar, K., Mahadevan, M., & Varalakshmi, P. (2012, May). QoS monitoring and dynamic trust establishment in the cloud. *In International Conference on Grid and Pervasive Computing (pp. 289–301). Springer, Berlin, Heidelberg.* https://doi.org/10.1007/978-3-642-30767-6_25

[37] Yahyaoui, H., Own, H. S., Agwa, A., & Maamar, Z. (2020). A novel scalable representative-based forecasting approach of service quality. *Computing, 1–30.* https://doi.org/10.1007/s00607-020-00802-z

[38] Jahani, A., Khanli, L. M., & Razavi, S. N. (2014). W_SR: a QoS based ranking approach for cloud computing service. *Computer Engineering and Applications Journal, 3(2), 55–62*. https://doi.org/10.18495/comengapp.v3i2.76

[39] Devgan, M., & Dhindsa, K. S. (2013). A study of different QoS management techniques in cloud computing. *International Journal of Soft Computing and Engineering, 3, 37–41.*

[40] Fiala, J. (2015). *A survey of machine learning applications to cloud computing.*

[41] Rahman, M. S., Ding, C., Liu, X., & Chi, C. H. (2016, June). A testbed for collecting QoS data of cloud-based analytic services. *In 2016 IEEE 9th International Conference on Cloud Computing (CLOUD) (pp. 236–243). IEEE.* https://doi.org/10.1109/CLOUD.2016.0040

[42] Eisa, M., Younas, M., & Basu, K. (2018, May). Analysis and representation of QoS attributes in cloud service selection. *In 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA) (pp. 960–967). IEEE.* https://doi.org/10.1109/AINA.2018.00140

[43] Alkalbani, A. M., Ghamry, A. M., Hussain, F. K., & Hussain, O. K. (2015, November). Blue pages: software as a service data set. *In 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA) (pp. 269–274). IEEE* https://doi.org/10.1109/BWCCA.2015.83

[44] Jararweh, Y., Jarrah, M., Alshara, Z., Alsaleh, M. N., & Al-Ayyoub, M. (2014). CloudExp: a comprehensive cloud computing experimental framework. *Simulation Modelling Practice and Theory, 49, 180–192.* https://doi.org/10.1016/j.simpat.2014.09.003

[45] Zheng, X., Martin, P., Brohman, K., & Da Xu, L. (2014). Cloudqual: a quality model for cloud services. *IEEE transactions on industrial informatics, 10(2), 1527–1536.* https://doi.org/10.1109/TII.2014.2306329

[46] Bouras, I., Aisopos, F., Violos, J., Kousiouris, G., Psychas, A., Varvarigou, T. A., & Stavroulas, Y. (2019). Mapping of quality of service requirements to resource demands for IaaS. *In CLOSER (pp. 263–270).* https://doi.org/10.5220/0007676902630270

[47] Zheng, X., Da Xu, L., & Chai, S. (2017). Qos recommendation in cloud services. *IEEE Access, 5, 5171–5177.* https://doi.org/10.1109/ACCESS.2017.2695657

[48] Ezenwoke, A., Daramola, O., & Adigun, M. (2018). QoS-based ranking and selection of SaaS applications using heterogeneous similarity metrics. *Journal of Cloud Computing, 7(1), 15.* https://doi.org/10.1186/s13677-018-0117-4

[49] Raeis, M., Tizghadam, A., & Leon-Garcia, *A. Queue-Learning: A Reinforcement Learning Approach for Providing Quality of Service (2021) arXiv preprint arXiv:2101.04627.*

[50] Moreira, R., Silva, F. D. O., Rosa, P. F., & 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.* https://doi.org/10.1504/IJGUC.2020.10025642; https://doi.org/10.1504/IJGUC.2020.103969

[51] Liang, H., Wen, X., Liu, Y., Zhang, H., Zhang, L., & Wang, L. (2021). Logistics-involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning. *Robotics and Computer-Integrated Manufacturing, 67, 101991.* https://doi.org/10.1016/j.rcim.2020.101991

[52] Afzal, S., & Kavitha, G. (2020). A hybrid, multiple parallel queuing model to enhance QoS in cloud computing. *International Journal of Grid and High Performance Computing (IJGHPC), 12(1), 18–34.* https://doi.org/10.4018/IJGHPC.2020010102

[53] Matiisen, T., Oliver, A., Cohen, T., & Schulman, J. (2019). Teacher-student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2019.2934906

[54] Ranjbari, M., & Torkestani, J. A. (2018). A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers. *Journal of Parallel and Distributed Computing, 113, 55–62.* https://doi.org/10.1016/j.jpdc.2017.10.009

[55] Serrano, D., Bouchenak, S., Kouki, Y., de Oliveira Jr, F. A., Ledoux, T., Lejeune, J., & Sens, P. (2016). SLA guarantees for cloud services. *Future Generation Computer Systems, 54, 233–246.* https://doi.org/10.1016/j.future.2015.03.018

[56] Zahour, O., Benlahmar, E., Eddaouim, A., & Hourrane, O. (2020). A comparative study of machine learning methods for automatic classification of academic and vocational guidance questions. *(iJIM), 14(8)*. https://doi.org/10.3991/ijim.v14i08.13005

[57] Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y. C., & Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials, 21(4), 3133–3174.* https://doi.org/10.1109/COMST.2019.2916583

[58] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). *A brief survey of deep reinforcement learning. arXiv preprint arXiv:1708.05866*. https://doi.org/10.1109/MSP.2017.2743240

[59] Kuutti, S., Bowden, R., Jin, Y., Barber, P., & Fallah, S. (2020). A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2019.2962338

[60] https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html 12/4/2020.

[61] Shaikh A, (2020). Advances in deep learning in mobile interactive algorithms and learning. *International Journal of Interactive Mobile Technologies(iJIM), 14(10)*. https://doi.org/10.3991/ijim.v14i10.15369

[62] Gonzalez, S., & Miikkulainen, R. (2020, July). Improved training speed, accuracy, and data utilization through loss function optimization. *In 2020 IEEE Congress on Evolutionary Computation (CEC) (pp. 1–8). IEEE.* https://doi.org/10.1109/CEC48606.2020.9185777

[63] Biyanto, T. R., Irawan, S., Febrianto, H. Y., Afdanny, N., Rahman, A. H., Gunawan, K. S., … & Bethiana, T. N. (2017). Killer whale algorithm: an algorithm inspired by the life of killer whale. *Procedia Computer Science, 124, 151–157*. https://doi.org/10.1016/j.procs.2017.12.141

11 Authors

Afaf Edinat is currently pursuing a Ph.D. from the University of Jordan, King Abdullah II School for Information Technology, Amman (Jordan). E-mail: Afaf_Ediant@yahoo.com

Professor Rizik Al-Sayyed is a Jordanian Prof. of Networks, Databases, and Data Science at the University of Jordan, King Abdullah II School for Information Technology, Department of Information Technology, Amman (Jordan). E-mail: r.alsayyed@ju.edu.jo

Professor Amjad Hudaib is a Jordanian professor of Software Engineering, King Abdullah II School for IT, University of Jordan. Amman (Jordan). E-mail: Ahudaib@ju.edu.jo

Article submitted 2021-08-02. Resubmitted 2021-09-06. Final acceptance 2021-09-07. Final version published as submitted by the authors.