International Journal of E-Health and Medical Communications
Volume 12 • Issue 2 • March-April 2021

SLAMMP Framework for Cloud Resource Management and Its Impact on Healthcare Computational Techniques

Vivek Kumar Prasad, Nirma University, India

Madhuri D. Bhavsar, Nirma University, India

ABSTRACT

Technology such as cloud computing(CC) is constantly evolving and being adopted by the industries to manage their data and tasks. CC provides the resources for managing the tasks of the cloud users. The acceptance of the CC in healthcare industries is proven to be more cost-effective and convenient. CC manager has to manage the resources to provide services to the end-users of the healthcare sector. The SLAMMP framework discussed here shows how the resources are managed by using the concept of reinforcement learning (RL) and LSTM (long short-term memory) for monitoring and prediction of the cloud resources for healthcare organizations. The task(s) pattern and anti-pattern scenarios have been observed using HMM (hidden Markov model). These patterns will tune the SLA parameters (service level agreement) using blockchain-based smart contracts (SC). The result discussed here indicates that the variations in the cloud resource demand will be handled carefully using the SLAMMP framework. From the result obtained, it is identified that SLAMMP performs well with the parameter used here.

KEYWORDS

Cloud Computing, Hidden Markov Model, Long Short-Term Memory, Monitoring, Prediction, Reinforcement Learning, Service Level Agreement, SLAMMP, Smart Contract

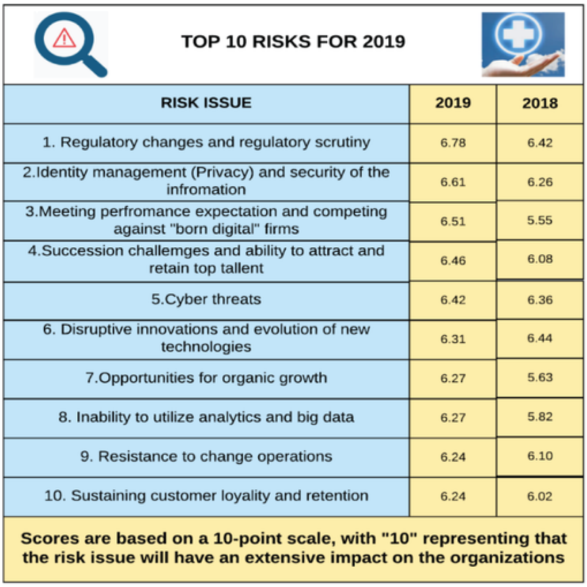
1. INTRODUCTION

The CC technologies are on the upsurge in the industry of Healthcare (Ali Omar et al., 2018). Even the adoption of the CC in healthcare industries is about to rise in the future. It's all because the industries have to deliver better quality medical services, sharing of the medical information, increase the operative efficiency, and increased competition between the different clouds SP (service provider) (Somula et al.,2019).

As per the survey (https://www.protiviti.com/US-en/insights/top-risks,2019) of North Carolina State University EMR Protiviti and EMR initiative. They have come up with the results by suggesting that establishments or organizations in the world have several critical concerns and are mention in Figure 1(a) and Figure 1(b). Amongst the various concerns, the important factor is the performance

DOI: 10.4018/IJEHMC.2021030101

Figure 1. (a) Survey report on top risk for 2019; (b) Graphical representation for the survey report on top risk for 2019

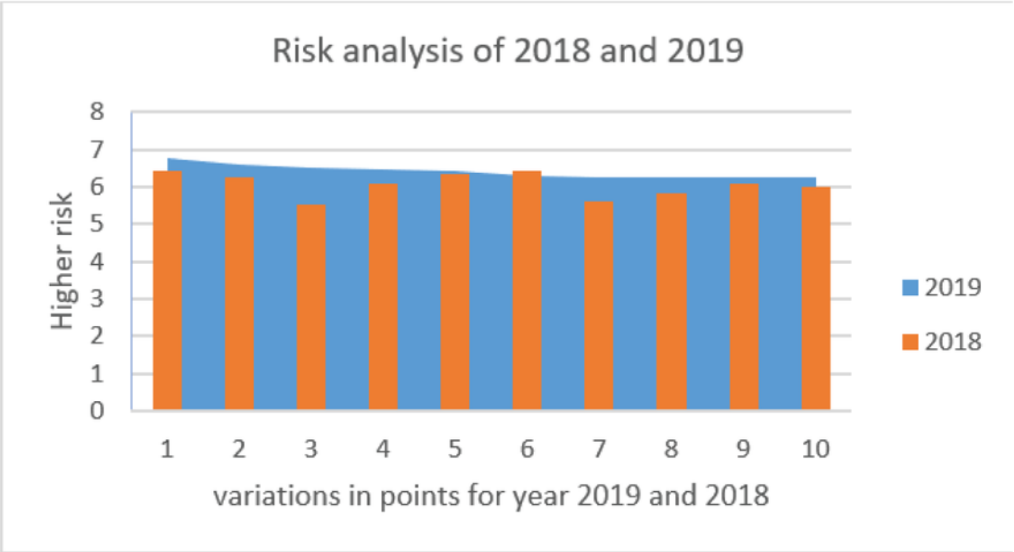


TOP 10 RISKS FOR 2019

RISK ISSUE	2019	2018
1. Regulatory changes and regulatory scrutiny	6.78	6.42
2.Identity management (Privacy) and security of the infromation	6.61	6.26
3.Meeting perfromance expectation and competing against "born digital" firms	6.51	5.55
4.Succession challemges and ability to attract and retain top tallent	6.46	6.08
5.Cyber threats	6.42	6.36
6. Disruptive innovations and evolution of new technologies	6.31	6.44
7.Opportunities for organic growth	6.27	5.63
8. Inability to utilize analytics and big data	6.27	5.82
9. Resistance to change operations	6.24	6.10
10. Sustaining customer loyality and retention	6.24	6.02

Scores are based on a 10-point scale, with "10" representing that the risk issue will have an extensive impact on the organizations

(a)



(b)

expectations. The existing IT infrastructure (legacy systems) and operations might are not capable of meeting with the present performance anticipations.

The CC should provide an environment where the applications must be managed efficiently by fulfilling its concern QoS without the human involvement. Now fulfilling these QoS requirements, maximizing its efficiency, heterogeneity, dispersion, and uncertainty of the infrastructure resources adds challenges to the CC ecosystems, which cannot be efficiently satisfied with the traditional resource allocation policies. Hence the desired qualities of the CC must be: to improves its performance through fault tolerance through avoiding or reducing the impacts of failures on executions, to completely avoid or reduce the under loading and overloading of the resources, and to proactively detect mischievous attacks in the cloud eco-system.

Overloading and underloading of the resources are all due to the instabilities of the workloads. The dispersion and uncertainties of resources cause problems in the allocation of possessions such as CPU utility, RAM, memory, etc. Considering the aforementioned constraints, the SLAMMP framework is proposed in this research paper.The SLAMMP framework discussed here will mitigate the risks factor mentioned with numbers such as 3, 6, 8, 9, and 10 of Figure1(a). As a result, the management of cloud IaaS resources will help health care industries to manage their patient's data and healthcare information in a more cost-effective and precise way, as per their general policies. While the CC can expressively reduce the cost related to IT and intricacies, this also enhances the delivery of the services and utilization of the resources. The e-health cloud services must be accessible continuously without performance degradation and interruptions. The SLAMMP framework discussed herein will focus on performance degradation parameters and take appropriate actions instantly to deliver the services uninterruptedly with a cost-effective approach. The resources will be managed automatically for the medical services at the back end by the CSP using the SLAMMP framework to avoid the conditions of over-provisioning and under-provisioning (Zhao et al.,2016). SLAMMP take cares of the SLA. The SLA explains the roles of each client and the service provider.

A Service level agreement(SLA) is the pledge(bond) for the performance negotiations between the ens user and the cloud service provider. SLA describes the level of the facility expected by the end-users from the CSP, setting out the metrics from which the services can be measured and imposes the penalties to any of the mentioned parties if the agreed-on service levels are not attained (Shah T et al.,2016).

The SLA criteria (Stamou et al.,2013)can be defined as shown in *Table 1*.

The violation in SLA will result in the payment of the penalties (Yuan et al.,2018) to any of the parties involving in service management. The components of the SLAs are QoS (Quality of service) parameters, which have to be monitored to attain the service level objectives (SLOs) and to detect the violations. The different types of violations (Di Martino et al.,2017) are explained in Table 2. As a CSP of the CC eco-system, SLA violations must be avoided to maintain the trust level.

1.1 Open Issues Concerning SLA Management (Hussain et al., 2016), (Singh, S. at al., 2016)

- The existing framework emphasis more on the technical attributes than on the management and security features of services.
- The proposed structures of SLAs do not contain a definition of the association between levels of violation and the cost incurred for the same.
- The studies do not assimilate a framework of trust management of the CSP with the collected information from SLAs monitoring systems.
- The definitions and concepts of service descriptions and service objectives involved in SLAs are not easy to recognize, especially for business decision creators.

Table 1. Criteria of SLA and its example

Various Citeria of SLA	Example
Availability	99.9% for nights/weekends 99.99% during working days
Performance	Less response time
Facility to access the data	information retrievable from a provider in a readable format

Table 2. Types of SLA violations

Type of Violation	Explanation	Solution
Provisioning of all or nothing	All SLOs must have satisfied with the successful delivery of the services or Complete services have been failed	New SLA requires to be identified and negotiated with other CSP
Provisioning partially	Some SLO's has been met	End-user and CSP react differently based upon there significant violated SLA parameters.
Provisioning weighted partial	SLO's satisfies if its value is greater than the threshold	Renegotiate with the service provider for the part of SLA which are violated

1.2 Problem Formulations

- The overall objective of the research paper is to manage the resources of the IaaS CC for healthcare industries as per the need of the end-users demand.
- The CC has a very complex infrastructure and may be challenging to understand and manage.
- As medical health care data are massive and need to be managed properly. For example, transforming healthcare information of the patient to the doctors in real-time and Patient information management data such as claims and billings, etc. Hence any sudden demand for the resources should be managed in real-time by the CSP.

To deal with such problems.The SLAMMP framework has been proposed here, which handles the peak demand and maintain the resources to avoid the conditions of under and over-provisioning.

1.3 Contributions

- A framework SLAMMP has been proposed by combining the approaches of monitoring and prediction methodologies of the cloud resources.
- The anti-patterns identification through HMM has been used to deal with real-time scenarios.
- The Blockchain based Smart Contract has been used to fine-tune the SLA management for uninterrupted services and high availability. This management approach can be used capacity planning, matchmaking algorithms, and task scheduling.
- The prediction of the cloud resources is implemented by LSTM.
- The performance of the proposed framework is validated using the parameters of the QoS and prediction model.

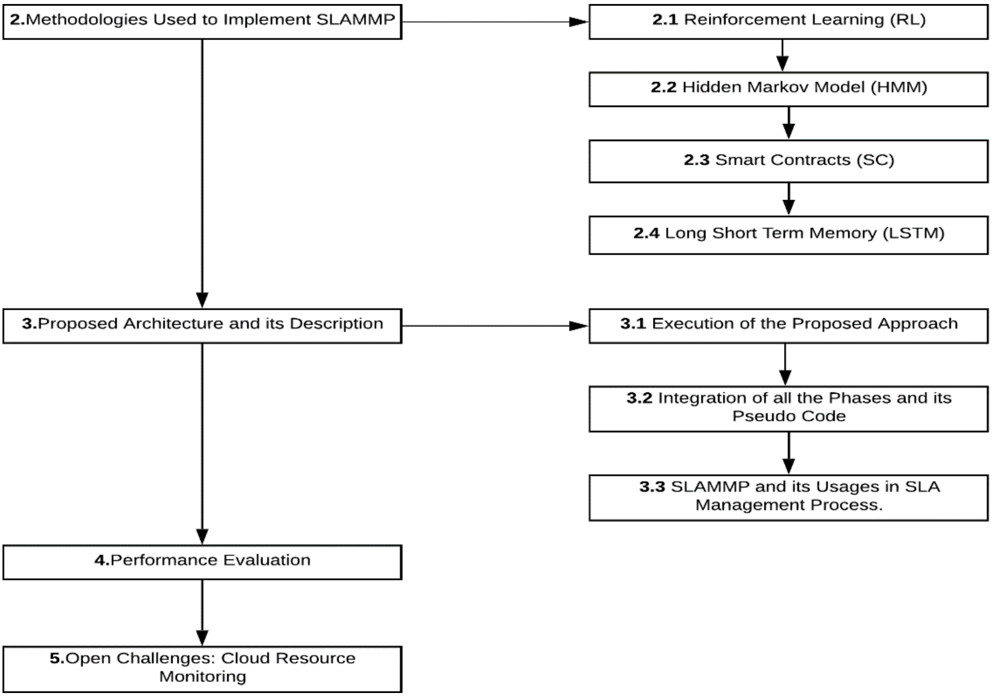
1.4 Organization of the Paper

The next section discusses about the methodologies used to implement the SLAMMP framework, then the architecture has been implemented with every phase-wise working procedures, followed by performance evaluation, conclusion and future work. Figure 2 indicates about the reading map concerning to this research paper.

2. METHODOLOGIES USED TO IMPLEMENT SLAMMP

To implement the proposed approach, various methodologies have been used, such as RL for monitoring the cloud resources, HMM for identifying the present state of the cloud environ,SC for SLA management and LSTM for predicting the cloud resources. A brief introduction about all the key items (methogologoes) has been represented below.

Figure 2. The reading map


2.Methodologies Used to Implement SLAMMP
2.1 Reinforcement Learning (RL)
2.2 Hidden Markov Model (HMM)
2.3 Smart Contracts (SC)
2.4 Long Short Term Memory (LSTM)
3.Proposed Architecture and its Description
3.1 Execution of the Proposed Approach
3.2 Integration of all the Phases and its Pseudo Code
3.3 SLAMMP and its Usages in SLA Management Process.
4.Performance Evaluation
5.Open Challenges: Cloud Resource Monitoring

2.1 Reinforcement Learning

RL is one of the prominent areas of machine learning (Henderson P et al.,2018). It is about captivating the appropriate actions to maximize the reward in a specific condition. It has been deployed by various machines and software to identify the finest likely behavior it should implement in the specific scenario. There are numerous researchs done to implement RL and is well suited to cloud environs as they do not need a priori information of the application performance. It learns the environment as the task/job runs (Liu B et al.,2019), (Liu H et al.,2018),(Duggan M et al.,2016). To work with RL, the policies are required from which the positive and negative rewards will be generated, which tends to change over time and stops when the goal will be received. RL works with the fundamentals of the Bellman equation, as described below:

$$Q\left(s,a\right)=r\left(s,a\right)+\gamma\max_a Q\left(s',a\right)$$

Q(s, a) is the target, r(s, a) is the reward of taking that action at that corresponding state and $\gamma\max_a Q(s',a)$ is the discounted max Q value among all possible actions from the next state. RL is a process of learning by communications with a dynamic environment like CC, which generates optimal action (or control) policies on an agreed environment state. Also, RL is able to create policies optimizing a long-term goal based on immediate rewards of actions. Starting from an initial policy, the Virtual Machine-Agent would gradually drive the policy to converge to the best one through exploration and exploitation of the CC eco-system. Hence RL is used for monitoring in this research work.

2.2 Hidden Markov Model

Hidden Markov model (HMM) (Wan J et al.,2016) is a part of a statistical Markov model, where the system which is being modeled is presumed as the Markov process with the unobservable states (called a hidden state). This can be considered as a mixture model (generalization) where the hidden variable is used to control the mixture component to be selected for every observation (Wang X et al.,2017). These are related to each other with the Markov process (instead of independence). The HMM will be able to identify the present state of the cloud (heavy loaded, not loaded, etc.) by observing its values (Prasad, V. K et al.,2013), (Xu et al., 2013). HMM are a powerful tool for modeling motifs, especially for a repetitive one forming a pattern. Therefore this is used here for the identification of patterns and anti patterns.

2.3 Smart Contracts

Cloud facilities operate in an extremely dynamic situation. This means that they need to be multifarious with changing SLAs, which explicate how a rich set of QoS guarantees can be implemented. In this way, the cloud users will migrate their processes to the cloud and will trust on CC.The SLAs are assumed to include single states while they are managed mainly in a centralized manner. To manage the dynamics of SLAs, the SLA formalism is transformed into a smart contract.So this concept is used here to manage the SLA. A smart contract (Bhargavan K et al.,2016),(Rimba P et al.,2017) is a PC convention proposed to carefully encourage, confirm, or implement the exchange or execution of an agreement. Smart contracts permit the execution of believable exchanges without outsiders. These exchanges are identifiable and irreversible (Scoca V et al.,2017). A SC not only determines the rules and penalties related to agreements, but it can also automatically enforce those obligations and manages the service in a better way.

Smart contracts are more complex, and their potentials go beyond the simple transfer of assets they can execute a transaction in a very large range. Also, to processes legal agreements to insurance crowdfunding agreements. Its time saving by excluding human participation in transactions and followed by the particular programming structure. In terms of safety, data in the decentralized cannot lose or no chances of cyber-attacks (Nayak. S et al.,2018). By talking about the accuracy and precision, there are no chances at all due to the absence of human input in form filling.

2.4 LSTM

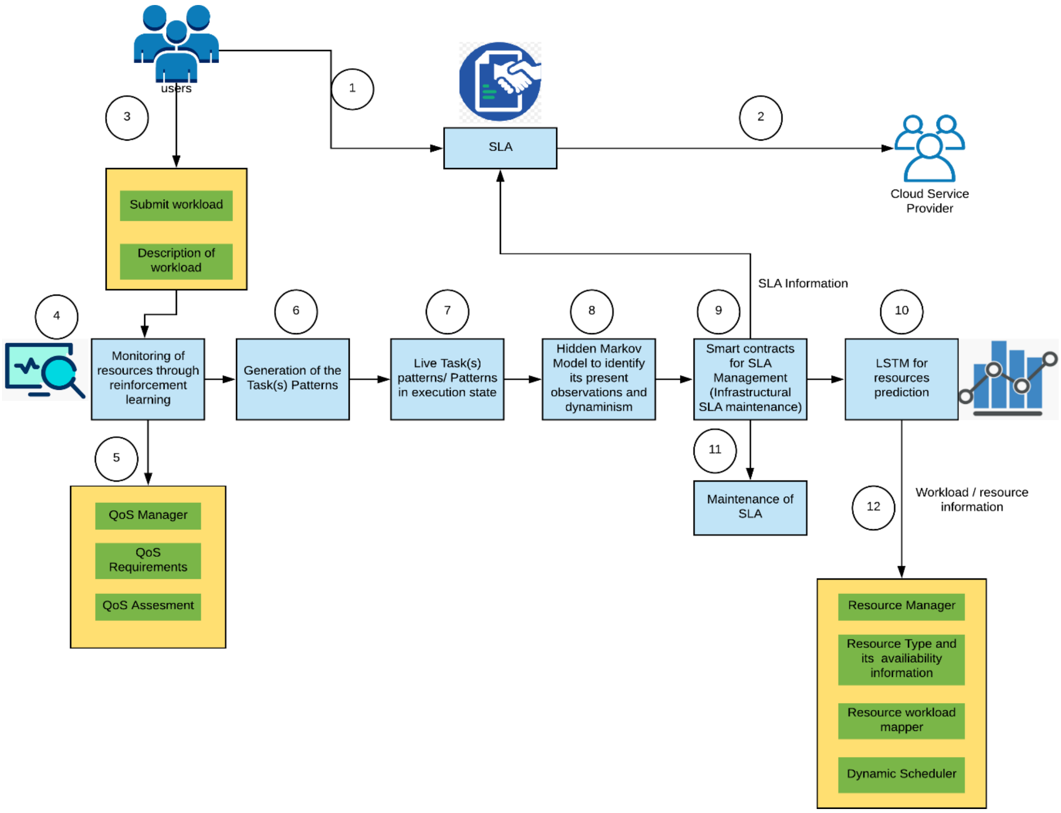
Long short term memory (LSTM) is an extension of the recurrent neural network, which essentially extends their memory (Tan Z et al.,2019), (Jing H et al.,2018). These extended memories are used to store the imperative pieces of knowledge/ experiences that have very elongated time pauses in between. The LSTM’s can delete, read, and write data from its memory. It has three gates, which are named as output, forget and input gate for its memory. These gates regulate whether or not to let fresh input in, this influence the output at the present step, and delete the data as this might not use for the time being (Baughman. M et al.,2018). Artificial neural networks are a computational standard that impersonates the biological neural networks (NN) through a network of linked computational components called neurons, which are arranged as layers. LSTM networks are a class of RNN (Recurrent Neural Network, whose architectural design is well suited for time-series forecasting) that are used to interpret and learn long-term dependencies among the time series forecasting of the cloud resource utilization such as CPU in the data center.Hence is used herein the framework; for prediction of the resources.

3. PROPOSED ARCHITECTURE AND ITS DESCRIPTIONS

To complete the execution of the current workload and reduce the conditions of under-provisioning and over-provisioning of resources in the CC eco-system. For the improvement of the cloud performance via fault tolerance, this section discusses the architecture of the SLAMMP framework and its working. Steps associated with the architecture has been discussed as shown in Figure 3, and the related steps are as below:

Steps 1 and 2: Users connect to the services offered by CSP (Cloud Service Provider) based upon the defined SLA (Service level agreement).
Step 3: User submits the workload to Cloud Computing Environment.
Steps 4 and 5: Monitoring of the resources will be done using RL (Reinforcement Learning); QoS will be observed in Phase-I *(Phase I is discussed in section 3.1).*
Step 6: Generation of various task patterns will be logged.
Step 7: The patterns will be exposed to runtime in a live system; and checked for any inconsistencies (anti-pattern).
Step 8: The inconsistencies will be identified by the HMM (Hidden Markov Model) and is informed to the administrator and operators to manage the system in phase-II. (*Phase II is discussed in section 3.1)*
Step 9: The fine-tunes QoS parameters required to manage the system's performance; will be given as input to the Smart contract phase-III.*(Phase III is discussed in section 3.1)*
Step 10: The output of the previous step can be used for the prediction of resources using LSTM in phase IV. *(Phase IV is discussed in section 3.1)*
Step 11: SLA will be maintained, and CSP will generate revenues.
Step 12: Management of the resources will be achieved.

Figure 3. SLAMMP Architecture of the proposed approach


users
1
2
3
SLA
Cloud Service Provider
Submit workload
Description of workload
SLA Information
4
5
6
7
8
9
10
11
12
Monitoring of resources through reinforcement learning
Generation of the Task(s) Patterns
Live Task(s) patterns/ Patterns in execution state
Hidden Markov Model to identify its present observations and dynaminism
Smart contracts for SLA Management (Infrastructural SLA maintenance)
LSTM for resources prediction
Maintenance of SLA
Workload / resource information
QoS Manager
QoS Requirements
QoS Assesment
Resource Manager
Resource Type and its availiability information
Resource workload mapper
Dynamic Scheduler

3.1 Execution of the Proposed Approach

The proposed scheme has the following phases and has been discussed below.

3.1.1. Phase 1: RL for Creating the Patterns of the Tasks

- **Cloud Applications and Their QoS (Quality of Service) Requirement:** As per the future requirements/demand of the applications, the provisioning for the efficient resources must identify the minimum amount of resources to fulfill the parameters of the QoS; such as CPU, RAM, memory utilization, response time, availability and reliability.
- **Identification of the Threshold:** Depending upon the context of the ongoing job(s) in the cloud, the figured values keep on evolving. The system's manager has to analyze the trend and need to compute the suitable baseline values (Beloglazov et al.,2010), (Buyya R et al, 2010). Hence monitoring plays a vital role in observing the current performance and detecting the irregular patterns. The CC threshold can be classified as static and dynamic. The static threshold doesn't work with the CC, as CC is highly dynamic(Maurya K et al.,2013).
- **Dynamic Threshold:** This helps and sends alerts before the bottleneck arises, used for better visibility, eliminates the occurrences of false alerts, and also decreases the management overhead. Figure 4 indicates the steps associated with the dynamic threshold. Initially, the historical data will be observed; and then the outliers can be removed, identify the maximum value, and then determine the threshold (Beloglazov A. & Buyya R., 2012).
- **Using RL for Creating the Patterns:** Figure 5 shows the working of phase I, i.e. the usages of RL for creating the task patterns in the CC environments.The RL process generates policies for online learning of the resource utility in cloud.This increases the learning adaptability and helps to drive appliances into a good configuration settings quickly. The input for this phase will be the cloud tasks and threshold (how these tasks been executed in the cloud environs using various thresholds selected as per the variations of the load observed). Then the output for this will be the various recorded patterns of workloads related to healthcare systems. The results of this phase have been discussed in the performance evaluation section- 4. The benefits of this phase can be utilized for doing better resource allocation.
- **Patterns and Anti-Patterns (Brabra. H et al., 2016), (Vetter et al., 2016):** Various patterns of the tasks can be classified (for the sake of understanding and simplicity; the CPU parameters has been undertaken) as the CPU utilization is below the threshold, above the threshold, just below the threshold and highly variable. The predefined tasks are kept to analyze its performance at the

Figure 4. Computing the dynamic threshold value

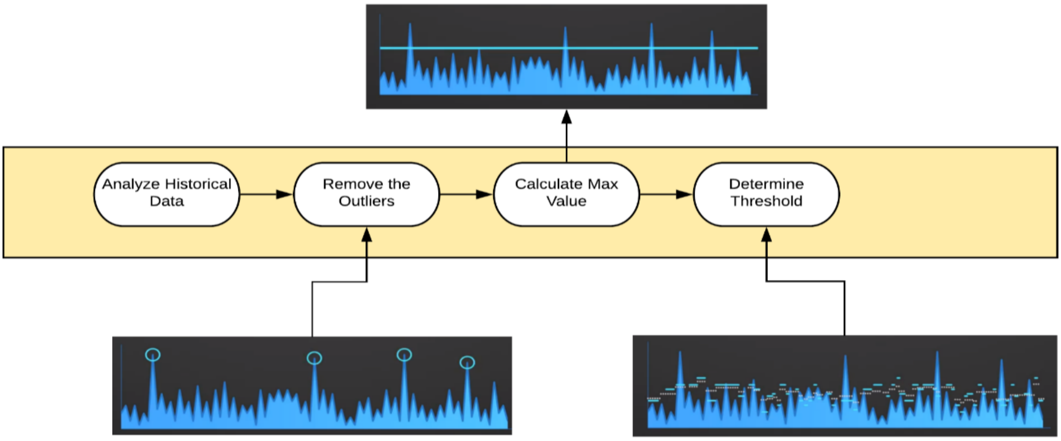


Figure 5. Phase I RL for creating the patterns in the CC environments

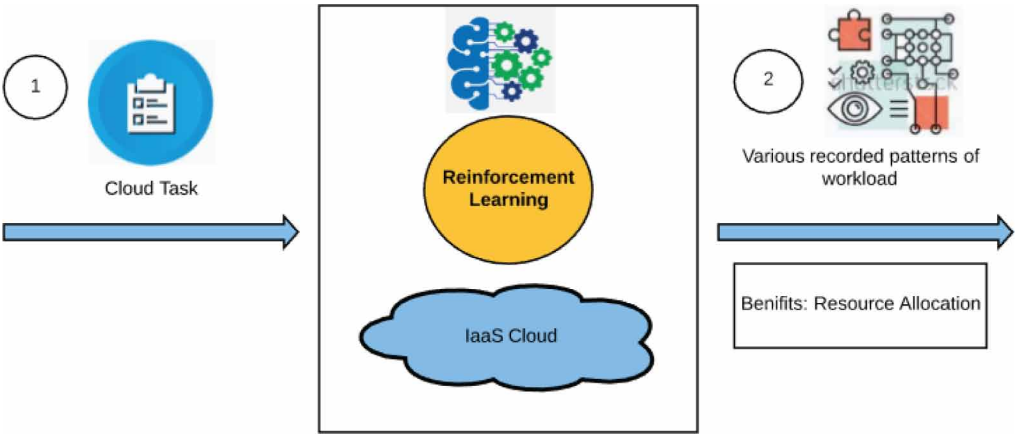
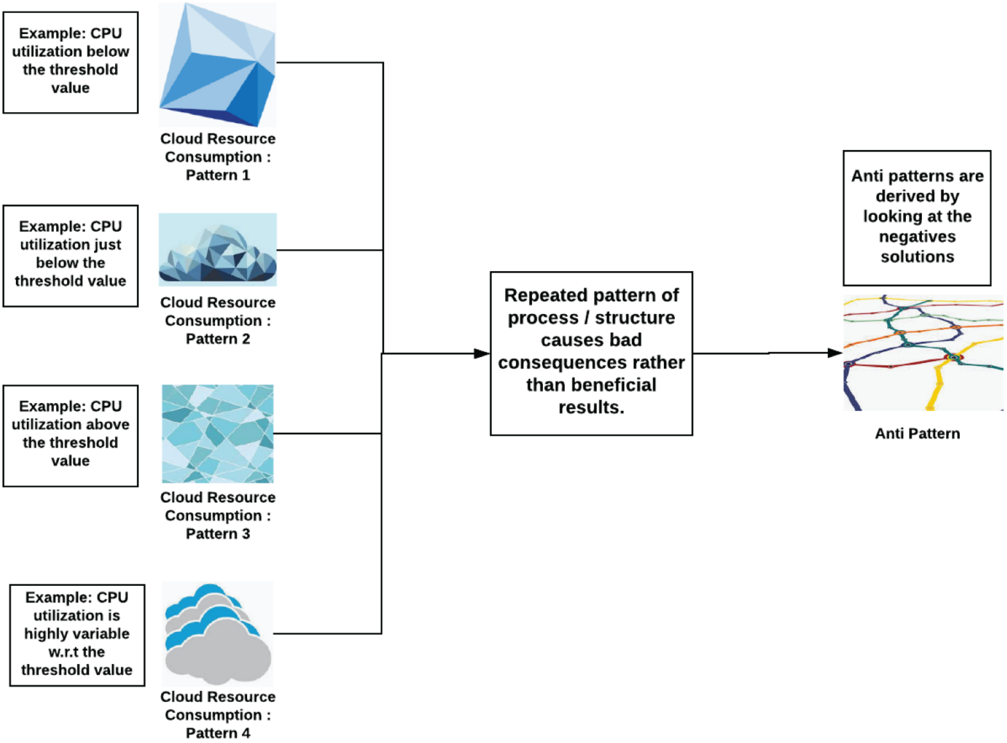

1
Cloud Task
Reinforcement Learning
IaaS Cloud
2
Various recorded patterns of workload
Benifits: Resource Allocation

Figure 6. Patterns and anti-patterns in CC


Example: CPU utilization below the threshold value
Cloud Resource Consumption : Pattern 1
Example: CPU utilization just below the threshold value
Cloud Resource Consumption : Pattern 2
Example: CPU utilization above the threshold value
Cloud Resource Consumption : Pattern 3
Example: CPU utilization is highly variable w.r.t the threshold value
Cloud Resource Consumption : Pattern 4
Repeated pattern of process / structure causes bad consequences rather than beneficial results.
Anti patterns are derived by looking at the negatives solutions
Anti Pattern

runtime in the cloud environments. If the same patterns are used for a long time, and frequently, then these patterns can lead to bad consequences rather than beneficial results. The same has been depicted in Figure 6; hence to observe the present state of the art (IaaS infrastructure usages), the HMM methodologies have been used and are discussed in phase-II. The anti patterns can be identified by the observing the performance degradation in the QoS.

3.1.2. Phase II: Observing the Present Status of the Cloud and Handling Anti-Patterns Using HMM

HMM is a concept that inherits the statistical Markov model, where the system which being modeled will be expected as a Markov process and with unobservable states. This is a stochastic process; by observing the present conditions, its state will be identified.

The applicable window size for the observations needs to be calculated as per the following cases.

Case I: If the load of the submission changes slowly: Long observations will be selected (20 observations).

Case II: If the load of the submissions changes rapidly: Shorter Interval will be selected (7 to 10 observation).

In the present case, the observation is ten and has recorded every 5 minutes, and this constantly updated as the time passes. These observations will lead to the identification of the present state of the cloud as shown in Figure 7, where the current state is identified as "variable peak" in iteration 1, in iteration 2 the state is acknowledged as "Over provisioning" and in iteration 3 the recognized state is "under provisioning".

Figure 7. Identification of the present state

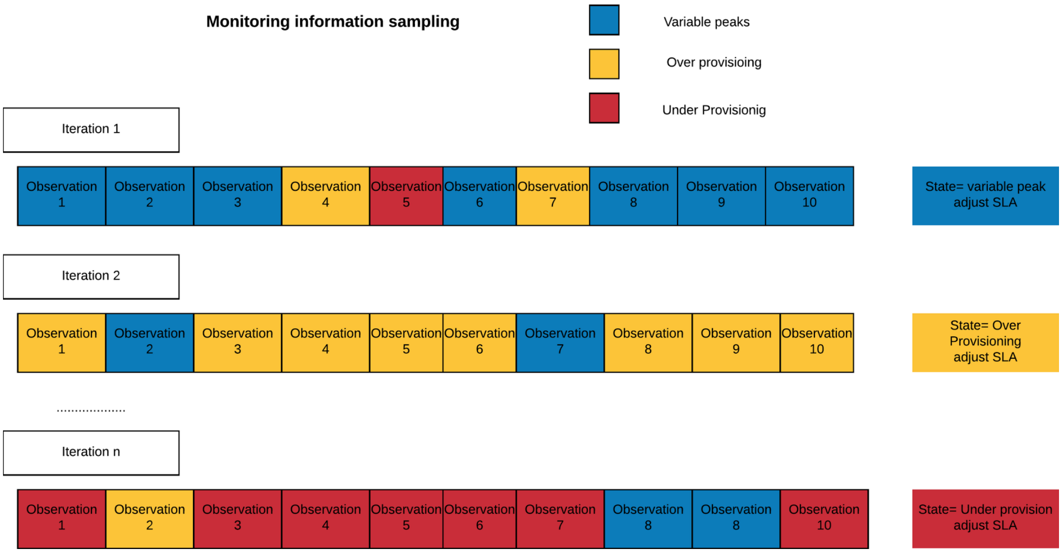
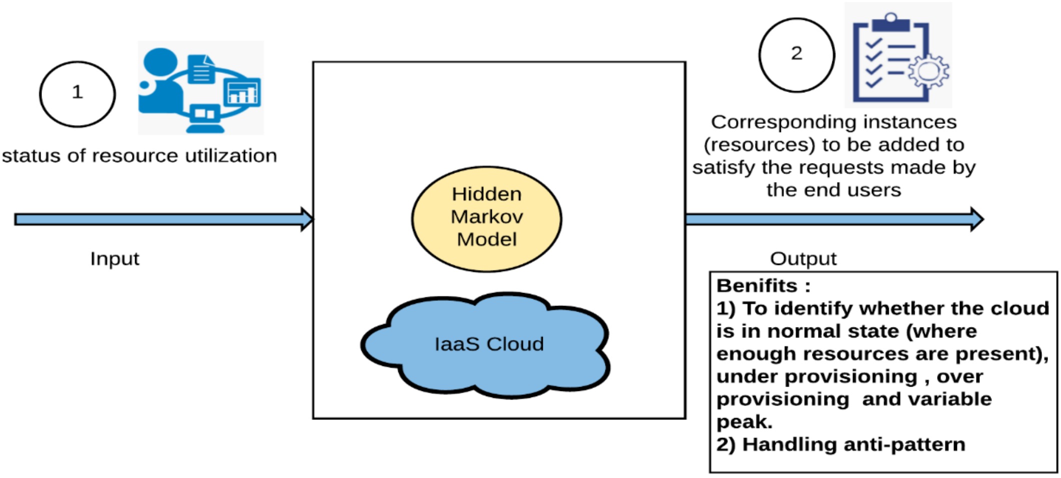


Figure 8 shows the working of phase II. The input for this phase will be the present resource (s) utilization patterns, and after the observation is undertaken, this will respond with the present status of the cloud. And this information can be used to identify which resources are over and under-utilized. This would help the cloud manager to recognize which hardware instances need to be added into the cloud for the smooth conduction of its operations. This phase will also classify that the patterns are behaving properly, and if not, then there are chances for the occurrences of the anti-patterns in the CC.

Figure 8. HMM and identification of the anti-patterns


1
status of resource utilization
Input
Hidden Markov Model
IaaS Cloud
2
Corresponding instances (resources) to be added to satisfy the requests made by the end users
Output
Benifits :
1) To identify whether the cloud is in normal state (where enough resources are present), under provisioning , over provisioning and variable peak.
2) Handling anti-pattern

3.1.3. Phase III: Blockchain-Based Smart Contract for SLA Management

As discussed in phase II, the input for the smart contract can be given as per Figure 9, which is shown below. If the expected range of utility (patterns behavior) is the same as observed by RL and HMM, then the input will be the allotment of the identified resources, which is to be used to fulfill the current demands of the end-user and this will also maintain of the QoS. If the occurrence of the anti-pattern is encountered, then input from HMM will be considered.

Figure 10 shows its working; the output for this phase will be the fine-tuned tasks. The smart contract has been used here for the management of the SLA. The SLA is the manuscripts that define what services and how this is delivered to the end-users by the CSP. The current reparation process is complex because lots of manual efforts are required. Hence the mechanism like Blockchain-based smart contracts can prove to be a better option for maintaining the SLA.

The other important benefits:

- **Expectations are Clear:** All parties involved herein know what to expect from each other. So everybody can proceed to fulfill their responsibilities with well-defined goals in mind.

Figure 9. Identification of the input received for smart contracts

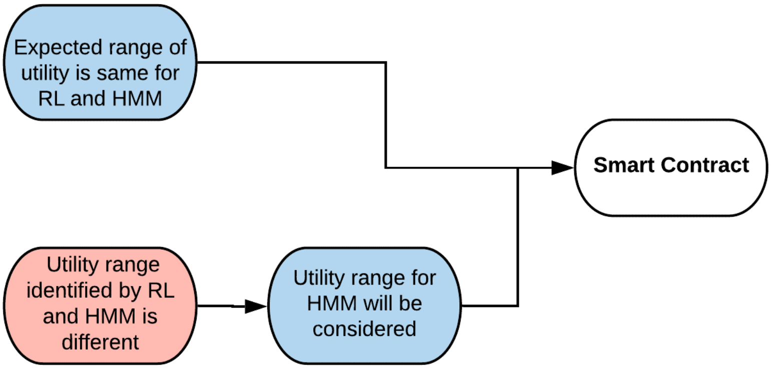
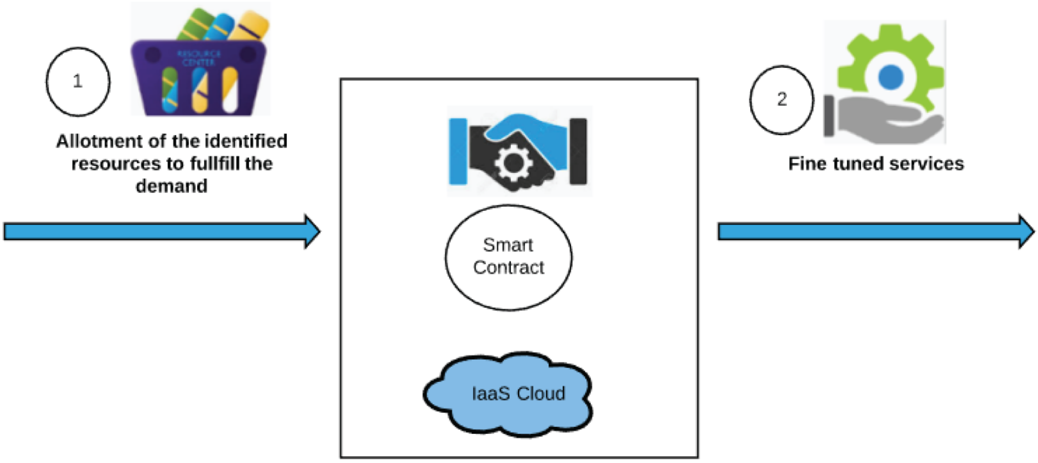

Expected range of utility is same for RL and HMM
Smart Contract
Utility range identified by RL and HMM is different
Utility range for HMM will be considered

Figure 10. Blockchain-based smart contract


1
Allotment of the identified resources to fullfill the demand
Smart Contract
IaaS Cloud
2
Fine tuned services

- **Saves Resources and Time:** When there are contract divergences among the stakeholders involved, SLAs deliver ways to crisscross.
- **Precise SLAs Help Businesses to Resolve Issues Sooner With the Least Resources:** There is no requirement for long meetings and discussions to figure out the root cause of the divergence.

3.1.4. Phase IV: LSTM for Resource Prediction

Figure 11 shows how the output from the last phase, i.e., phase III of Smart Contract, will be given as input to the LSTM to analyze or observe the future scope (prediction) of the said resources. This phase will identify the behavior of the resource demand based on the present input sequences.

3.2. Integration of all the Phases and its Pseudo Code

In this section, the phases (I, II, III, IV) such as RL, HMM, SC and LSTM have been integrated, as shown in Figure 12 and results into a collective framework called SLAMMP. The pseudo-code for

Figure 11. Using LSTM for the prediction of the resources

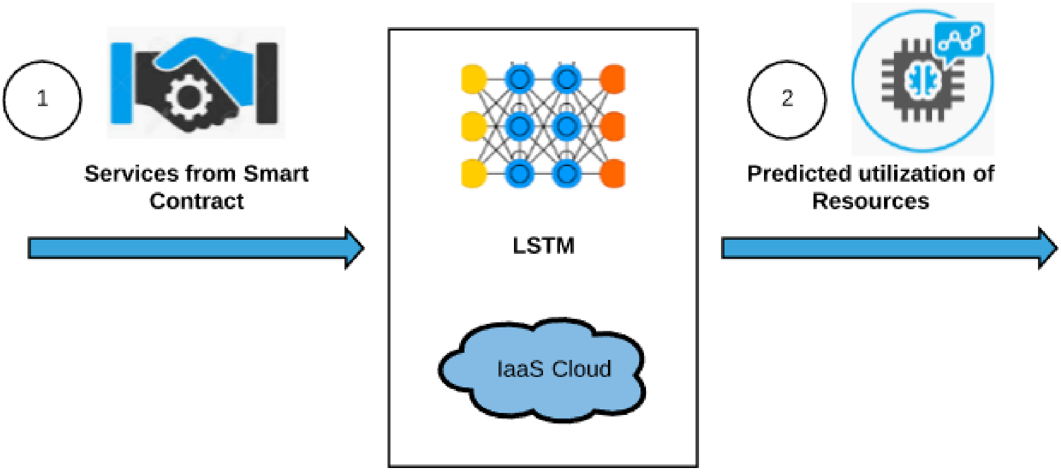
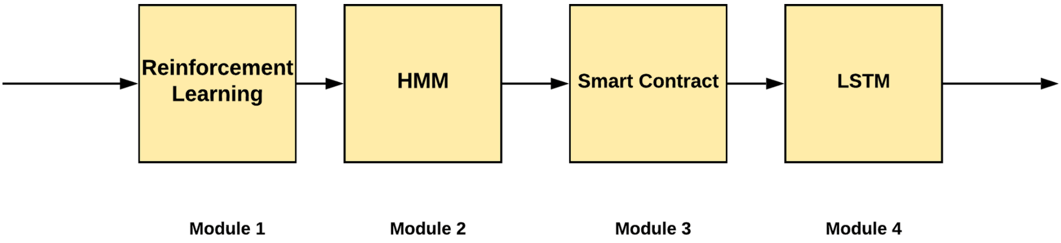

1
Services from Smart Contract
LSTM
IaaS Cloud
2
Predicted utilization of Resources

Figure 12. Integration of all the phases

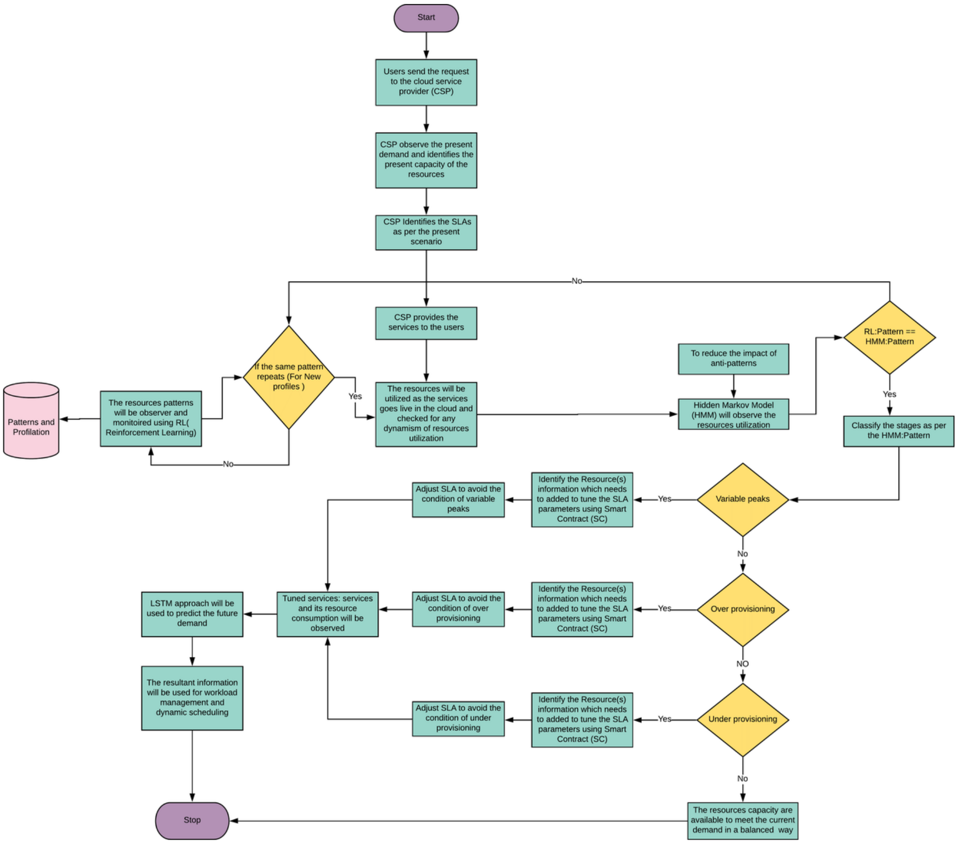


the same is described in pseudo-code 1: Working of SLAMMP, pseudo-code 2: Working of HMM, and pseudo code 3: Identification of the anti-patterns. The discussed framework will be used for capacity planning, matchmaking algorithm, admission control, and SLA process management of cloud computing, as discussed in section 3.3. The working of the entire framework of SLAMMP is described in Figure 13.

3.3 SLAMMP and its Usages in SLA Management Phases

SLA can be well-defined with the help of various phases (Prasad V K et al,2018) such as given below and is indicated in Figure 14. The SLA process management helps to monitor the performance

Figure 13. Flow chart of the SLAMMP


Start
Users send the request to the cloud service provider (CSP)
CSP observe the present demand and identifies the present capacity of the resources
CSP Identifies the SLAs as per the present scenario
No
CSP provides the services to the users
If the same pattern repeats (For New profiles)
Yes
No
The resources will be utilized as the services goes live in the cloud and checked for any dynamism of resources utilization
The resources patterns will be observer and monitoired using RL(Reinforcement Learning)
Patterns and Profilation
To reduce the impact of anti-patterns
Hidden Markov Model (HMM) will observe the resources utilization
RL:Pattern == HMM:Pattern
Yes
Classify the stages as per the HMM:Pattern
Variable peaks
Yes
Identify the Resource(s) information which needs to added to tune the SLA parameters using Smart Contract (SC)
Adjust SLA to avoid the condition of variable peaks
No
Over provisioning
Yes
Identify the Resource(s) information which needs to added to tune the SLA parameters using Smart Contract (SC)
Adjust SLA to avoid the condition of over provisioning
Tuned services: services and its resource consumption will be observed
LSTM approach will be used to predict the future demand
The resultant information will be used for workload management and dynamic scheduling
NO
Under provisioning
Yes
Identify the Resource(s) information which needs to added to tune the SLA parameters using Smart Contract (SC)
Adjust SLA to avoid the condition of under provisioning
No
The resources capacity are available to meet the current demand in a balanced way
Stop

Psuedo code 1. SLAMMP (SLA Management using Monitoring and Prediction)

```
Input: Task(s)
Output: Managed Workload
Initialization:
/* initialization and declaration of the variables*/
Initialize the total number of tasks N
Initialize the total number of resources M
Initialize the completion time $C_{i,j} = 0$
Initialize the execution time $E_{i,j} = 0$
Initialize the ready time $R_j = 0$
for i = 1 to N
  //N denoted the number of tasks to be executed in the cloud
  for j = 1 to M
    //M denotes the availability of the resources of the cloud
    $C_{i,j} = E_{i,j} + R_j$
    // $C_{i,j}$ denotes the completion time of the task
    // $E_{i,j}$ denotes execution time of the task
    // $R_j$ denotes the ready time of task on cloud resources
    RL will observe the $C_{i,j}$
    // RL will observe the completion time and behavior of the tasks in operating conditions for every pattern(s)
  end for
end for
do until the cloud operates in normal conditions for different $C_{i,j}$ patterns observed.
  // the various conditions such as variable peaks, under &over provisioning will be identified using HMM (Hidden Markov model) for identification of anti-patterns, as explained in Pseudo code 2 and 3*
  for each task find whether the observation leads to any of the unexpected conditions
    if there is a variable peak then the SLA will be tuned to adapt with the varying peaks and the predicted value based on LSTM will be used for workload management or dynamic scheduling
    else if
      the observation leads to over-provisioning, the SLA values will be adjusted accordingly and the predicted value based on LSTM will be used for workload management or dynamic scheduling
    else if
      the observation leads to under-provisioning, the SLA values will be adjusted accordingly and the predicted value based on LSTM will be used for workload management or dynamic scheduling
    else
      the cloud is in stable stage and there is no expectation of downtime in the near future
  end for
  update RL
end do
```

*HMM (Hidden Markov Model) steps has been mentioned below

* Algorithm for identification of anti-patterns has been explained after HMM's Algorithm

Pseudo code 2. Working of HMM

```
Input: Task(s) and its resources utilization
Output:Present status of the cloud
Initialization:
Initialize the number of observation =0
Initialize the count of (C1) of Peak observation=0
Initialize the count of (C2) of Under Provisioning=0
Initialize the count of (C3) of Overprovisioning =0

While till the VM's are running do
Read the observations and count the values for each observation of peak observation, Under-provisioning and Over provisioning)

if C1>C2
 if C1>C3
write "C1 is Maximum"
 // the observation indicates the present state is in peak load

else
write "C3 is Maximum"
 // the observation indicates the present state is in over provisioning
end if
else

if C2>C3
 write "C2 is Maximum"
 // the observation indicates the present state is in under-provisioning

else
write "C3 is Maximum"
 // the observation indicates the present state is in over provisioning
end if

Do
obtain the prediction using LSTM for the specific workload (maybe C1, C2, and C3)
end While
```

Pseudo code 3. Identification of anti-patterns

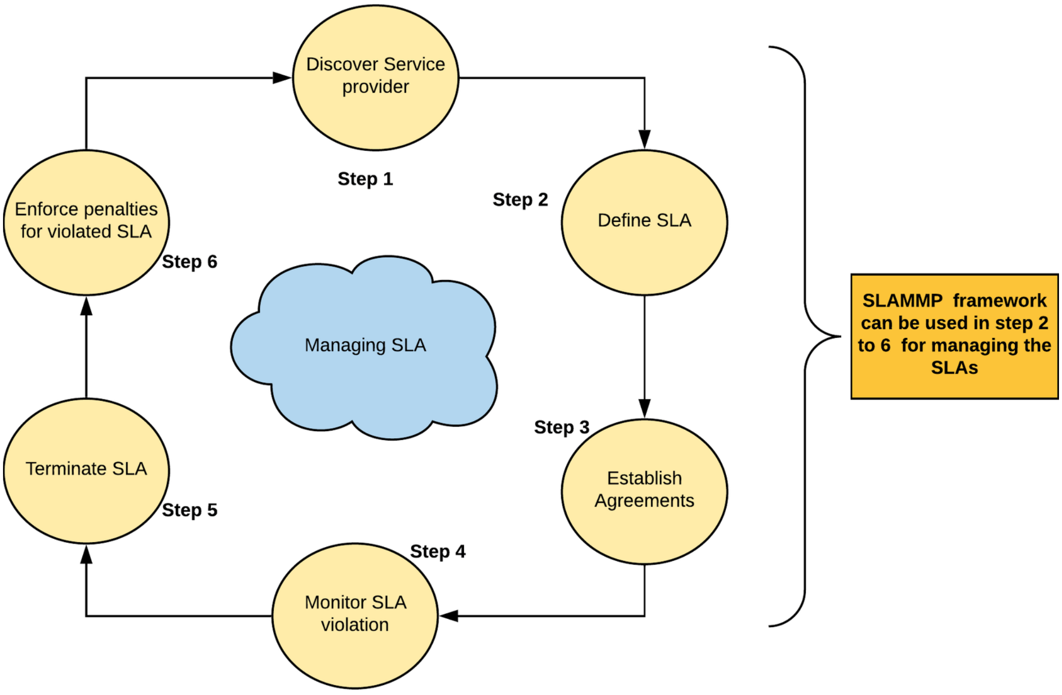
```
Input: Recorded pattern(s) using RL
Output: Identification of antipatterns and patterns
Initialization:
Initialize the pattern identified by RL
RL:Patterns=0

Initialize the pattern identified by HMM
HMM:Patterns=0
for each observation of RL: Patterns and HMM: Patterns identify

if (RL:Patterns== HMM:Patterns)

Classify the stages as per the HMM ;
Rest of the steps will be the same as pseudo-code 2

else
update the pattern in the RL: Patterns
Rest of the steps will be the same as pseudo-code 1

end for
```

of the achievement of the services against the targets within the SLA. It improves, measures, and collates the customer's satisfaction. It also determines the expectations of the customer and business by evaluating the capabilities and resources of the IT CSP. Figure 14 indicates the steps associated with this i.e discover service provider, defining the SLAs, establishing the agreement, monitoring the SLA violation, terminating the SLA and enforcing penalties for SLA violation:

1. **Discover the Service Provider:** This involves selecting the possible associates to interact with the known registry of the provider as per the customer requirements.
2. **Defining the SLAs:** This includes the definition of the services, QoS parameters, policies, and penalties, at this phase, it is possible to negotiate between the two parties (end-users and CSP).
3. **Establishing the Agreement:** The SLA template will be recognized and are made available, the groups will reach to the mutual agreements.
4. **Monitoring the SLA Violation:** The provider's service parameters are measured against the SLAs.
5. **Terminating the SLA:** The SLA terminates because of violations or any other means such as timeouts etc.
6. **Enforcing Penalties for SLA Violation:** If there is any party violating SLAs, the conforming penalty items are invoked and accomplished.

The SLAMMP framework that has been discussed here will be well suited for the steps numbers 2 and 6 of conventional SLA process management. Thus the proposed technique will automate the entire process and will avoid the conditions of breaching SLA and as a result, the downtime in the cloud will be avoided.

Figure 14. SLAMMP and SLA management phases


Discover Service provider
Step 1
Step 2
Define SLA
Enforce penalties for violated SLA
Step 6
Managing SLA
SLAMMP framework can be used in step 2 to 6 for managing the SLAs
Step 3
Establish Agreements
Terminate SLA
Step 5
Step 4
Monitor SLA violation

4. PERFORMANCE EVALUATION

Experimentations have been carried out by making use of the dataset of 1750 VMs in the distributed datacentre environment, with highly dynamic scenarios, and are used by various commercialized services. In Phase I, the RL has been used for creating the patterns. The parameters used for the experimentations w.r.t RL has been mentioned in *Table 3*.

Table 3. Parameters for RL

Parameters	Values
Discount (γ)	0.8
Tau	0.001
Batch size	64
layers	(50,50)
Learning rate	0.0001
Epsilon decay fraction	0.4
Memory faction	0.80
Memory Type	Deque
Process_observation	Standardized
Process_target	Normalizer

The results are shown in Figure 15 and Figure 16, which make use of the CPU utilization. Here the workload has been classified as less than and greater than the threshold defined i.e. less than and greater than 70%. The threshold is defined based upon the elastic nature of the cloud resources. An RL based approach for dynamic decision making in resource adjustment is self-management without human intervention. This approach enables a cloud request to guarantee its performance by learning the consequences of its behaviour (as what can be the optimal state using rewards and episodes) and by dynamically changing its plans based on the knowledge in the presence of environmental variations. This condition computes whether the best-chosen actions in this learning cycle equals the last stage or not.

Here we are assuming that there are no anti-patterns, and the pattern recognized by the RL and HMM are the same, and the System identifies the respective resources (instances), which need to be added in Phase-II. In Phase III the smart contract results are based on the parameters, as discussed below:

- Ether (ETH) is the Ethereum, which is based on the network’s native cryptocurrency.
- Gas is the unit of calculation that indicates the fee/charges for a specific act or contract.
- The Gas Limit is the extreme amount of Gas that an end-user(s) is ready to pay for the execution of an action or approving a transaction.
- The price of Gas (Gas Price) is the amount in terms of Gwei; that the user is agreeable to spend on each unit of Gas.

4.1. Communicating Ethereum

The operator instantiates the contract with the Ethereum in the distributed VMs (Virtual Machines); after this, the interested customers register themselves to the contract instance. After this, both the

Figure 15. Graph for rewards generation for the cloud environment policy is to manage the resources in less than 70% threshold. X-Axis: Episodes and Y-Axis: Rewards.

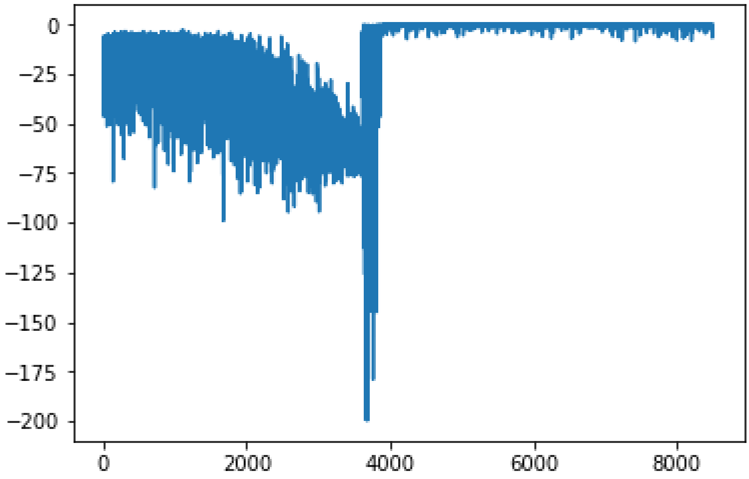
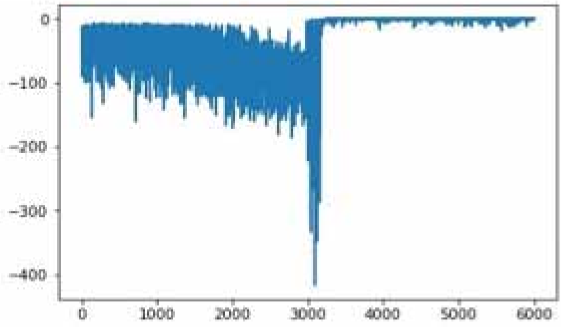


Figure 16. The CPU utilization is very high, and more than the threshold defined i.e. 70%. X-Axis: Episodes and Y-Axis: Rewards.



customer and operator will start posting their metrics records. In our case here, a transfer of 1 Ethereum is used with three conditions:

1. If the records don't match: the operator will transfer 2 ETH and the customer, and the smart contract will be temporarily disabled.
2. The record matched, and QoS is also above to the defined SLA: both the operator and customer will get back the respective stakes.
3. The records are matched but the QoS is found below the SLA: SLA violation occurs; a customer receives the stake from the operator:

[Note 1 Ether= 1000000000 Gwei]

The Blockchain-Based SC results have been depicted in Figure 17, 18, and 19.Here the upper limit and lower limit of the CPU utilization (utility range) has been classified (different patterns requirements) as 20 to 45 (Pattern A), 40 to 60 (Pattern B) and 60 to 80 (Pattern C). For the sake of visualization and clarity (of the variations); only a few of the iterations have been showcased here.

Figure 17. SLA tunedas per the demand, 20 to 45 percent, [X-axis = Iterations and Y-axis=Metrics/Ether]: Pattern A

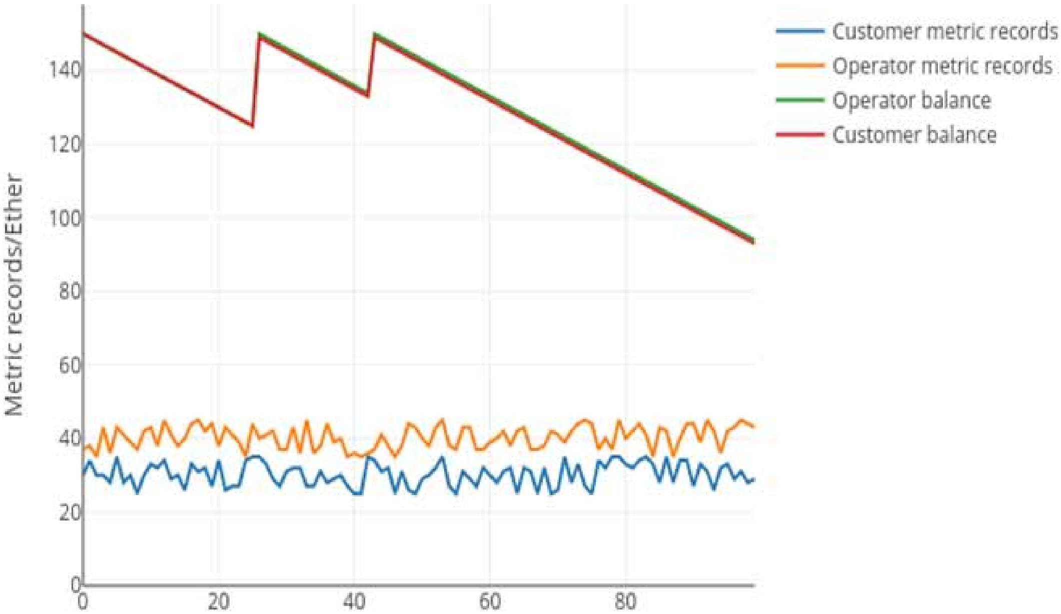


Figure 18. SLA tuned as per the peak demand, 40 to 60 percent, [X-axis = Iterations and Y-axis = Metrics/Ether]: Pattern C

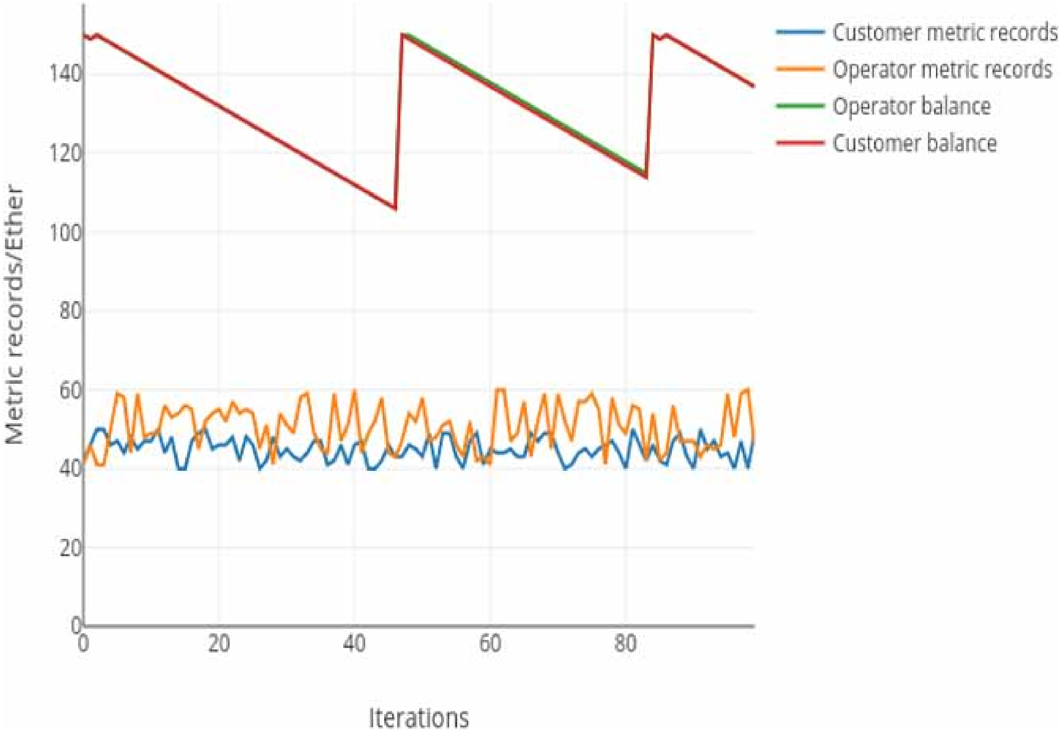
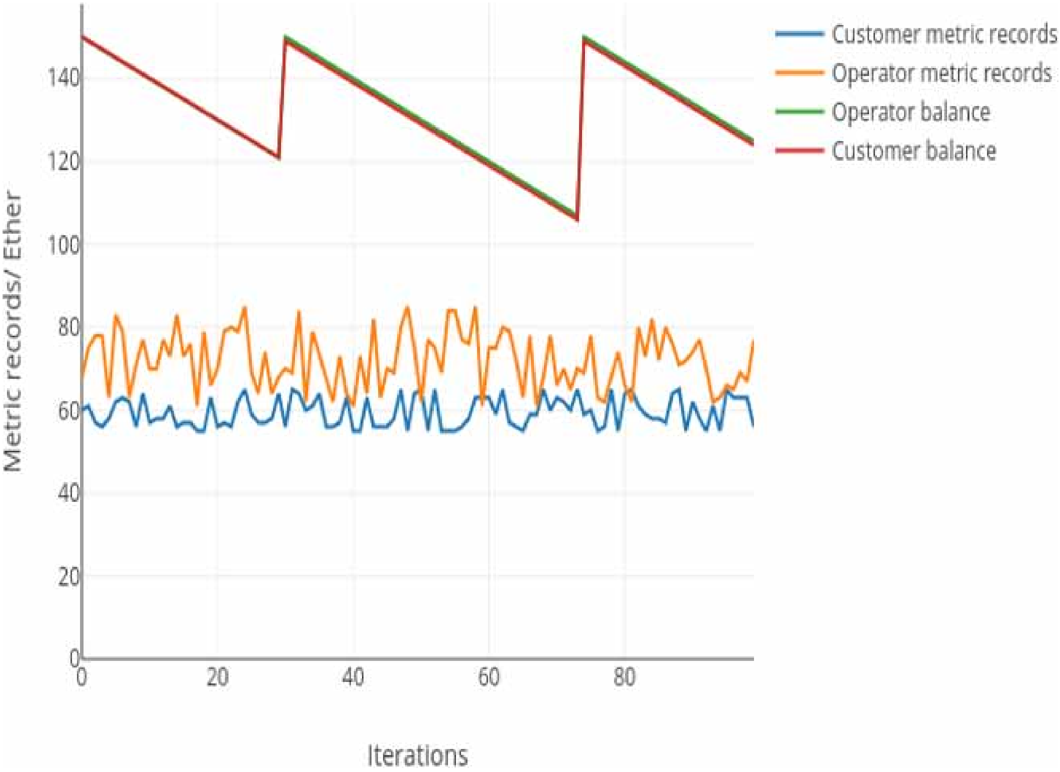


Figure 19. SLA tuned as per the peak demand, 60 to 80 percent, [X-axis = Iterations and Y-axis = Metrics/Ether]: Pattern C



These patterns will be treated as workloads and used as a performance evaluator for the CC applications. It also investigate the effect on application performance to measure the application sensitivity related to the CPU utility parameter.

Once the SLA is adjusted,the quantitative values of operators (CSP) and customer (Cloud users) traces will be generated and Prediction will be done on these traces to satisfy the future demands.

The prediction of pattern A, B, and C have been implemented using LSTM. The parameters used for LSTM has been depicted in *Table 4* and for selecting the hyperparameters grid search mechanism have been used.

The results using LSTM will determine the future demand of the resources for a particular type of workloads (both for customers and the operators); which has been mentioned as Pattern A, pattern B and Pattern C, shown in Figure 20, Figure 21, Figure 22, Figure 23, Figure 24 and Figure 25 for different ranges of CPU utilization and their respective MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) has been calculated and have been mentioned.

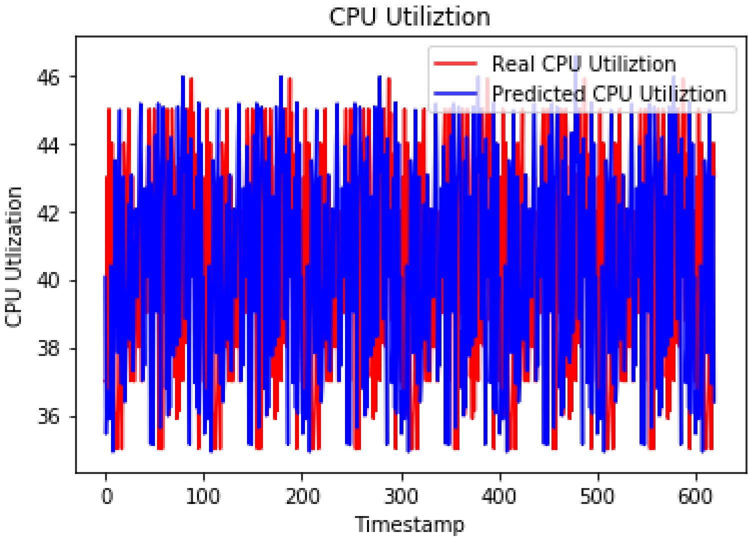
The experimental values are displayed in Table 5 with respect to various CPU utilization range, such as up to 40% utilization, from 40% to 60% utilization and above 60%. Remember that the right-hand side Figure 21, 23, and 25 are the graph of customers (Cloud-user) value, and its CPU utilization is less than the left-hand side Figures of the operators (CSP) such as Figure 20, 22, and 24. The values of the Operators are more than the customers and is adjusted in the Smart contract phase mentioned earlier and as per Figure 17, 18, and 19.

In the case where the utilization is up to 40%, the operator's MAE value is 0.79, and for RMSE, the value is 0.91, and for the customers' value, the MAE is 0.81 and RMSE is 0.98. Similarly, when the utilization is 40% to 60%, the operator's MAE value is 0.78, and the RMSE value is 0.92. For

Table 4. Parameters used for LSTM

Parameters	Values
Batch Size	64
Epochs	120
Time steps	10
Input layer	10 nodes
Output layer	10 nodes
Parameters for input layer	4 * LSTM output size * (weights of LSTM output size +1 Bias variable)
Parameters for output layer	4 * LSTM output size * (weights of LSTM output size +1 Bias variable)
Optimizer	Adam

Figure 20. Prediction of the CPU utilization using LSTM for pattern A of Operator, MAE= 0.79, RMSE= 0.91



the customer, the MAE value is 0.83, and RMSE is 0.99. For above 60%, the operator's MAE value is 0.92, and the RMSE value is 1.01, and for customers, the MAE value is 0.93, and RMSE is 1.02. These predicted values will help the CSP to manage their resources at runtime and helps to avoid the conditions of under and over provisioning proactively.

SLA can be well-defined with the help of various phases (Muthusamy V et al.,2009) (Debusmann. M et al., 2003) (Prasad V K et al.,2017) such as given below and is indicated in Figure 14. The SLA process management helps to monitor the performance of the achievement of the services against the targets within the SLA. It improves, measures, and collates the customer's satisfaction. It also determines the expectations of the customer and business by evaluating the capabilities and resources of the IT CSP. Figure 14 indicates the steps associated with this i.e discover service provider, defining the SLAs, establishing the agreement, monitoring the SLA violation, terminating the SLA and enforcing penalties for SLA violation.

Figure 21. Prediction of the CPU utilization using LSTM for pattern A of Customer, MAE=0.81, RMSE= 0.98

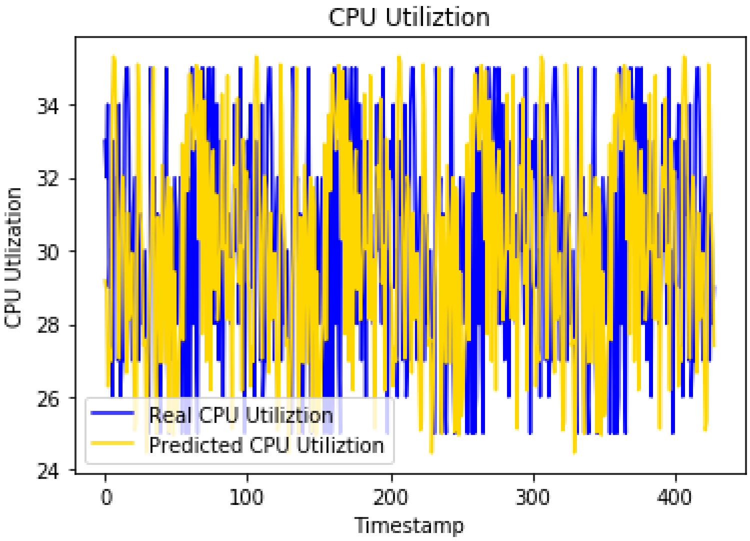
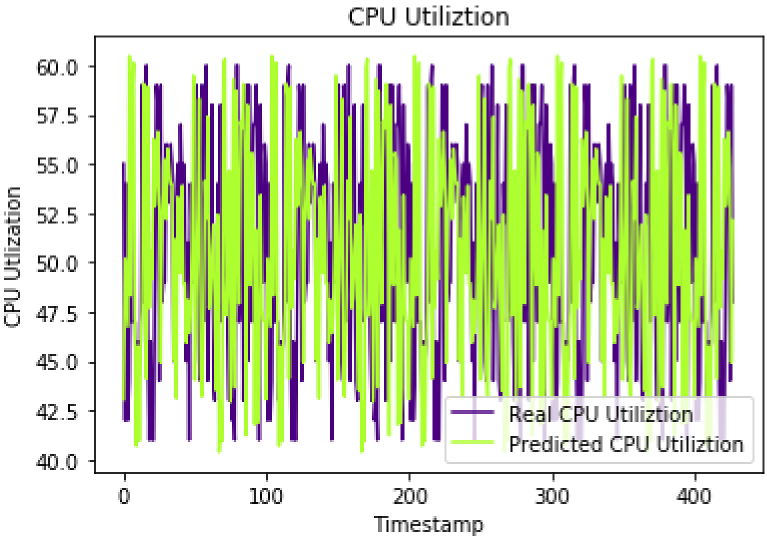


Figure 22. Prediction of the CPU resource using LSTM for pattern B of the operator.MAE =0.78,RMSE=0.92



The results discussed can be well suited for capacity management of the cloud resources and are based on real-time monitoring and prediction features, which usages the RL, HMM, smart contract and LSTM to support this framework. The monitoring results will identify the possible scenarios in which the type of tasks (workloads) can be generated in ideal situations with maximized enactment.

Key properties of several well-known open-source cloud monitoring platforms are (Paschke A et al., 2006) scalability, elasticity, adaptability, timeline ness, autonomic, comprehensiveness, extensibility, intrusiveness, resilience, reliability, availability, and accuracy. The mapping of these properties with the proposed approach and the existing open source has been discussed in *Table 6*. Some of the important open issues of resource monitoring and prediction of cloud computing is mentioned in section 5.

Figure 23. Prediction of the CPU resource using LSTM for pattern B of Customers. MAE=0.83,RMSE=0.99

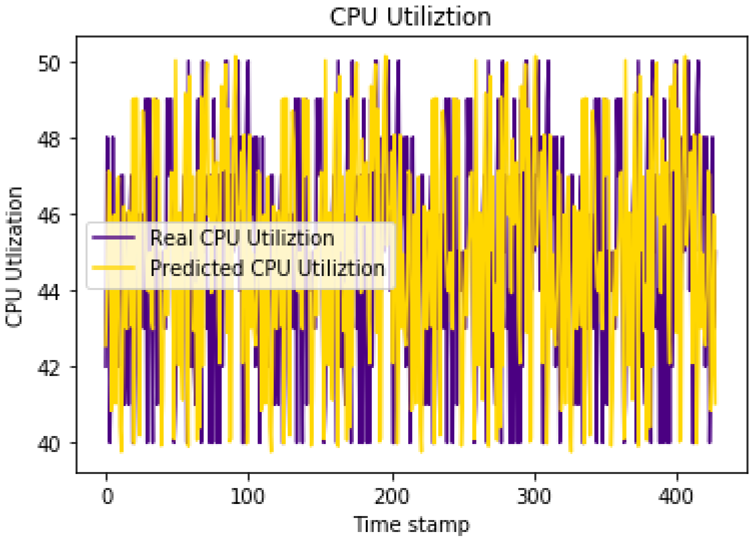
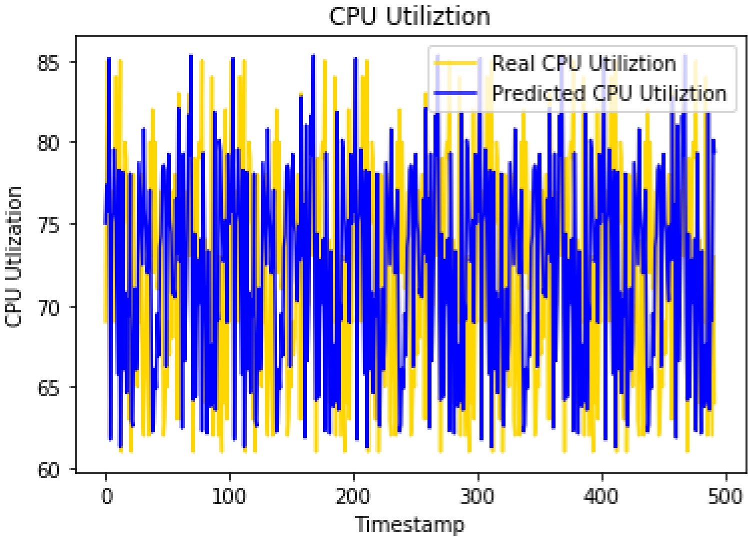


Figure 24. Prediction of the CPU resource using LSTM for the pattern of the operator. MAE = 0.92, RMSE= 1.01



5. OPEN CHALLENGES WITH RESPECT TO CLOUD RESOURCE MONITORING

Figure 26 demonstrates the various open research challenges for the CC Resource(s) monitoring. The key aspects for the same have been described below:

- Effectiveness
- Efficiency
- Cost and energy-efficient monitoring
- Cross-layer monitoring

Figure 25. Prediction of the CPU resource using LSTM for the pattern of the Customers. MAE=0.93, RMSE =1.02

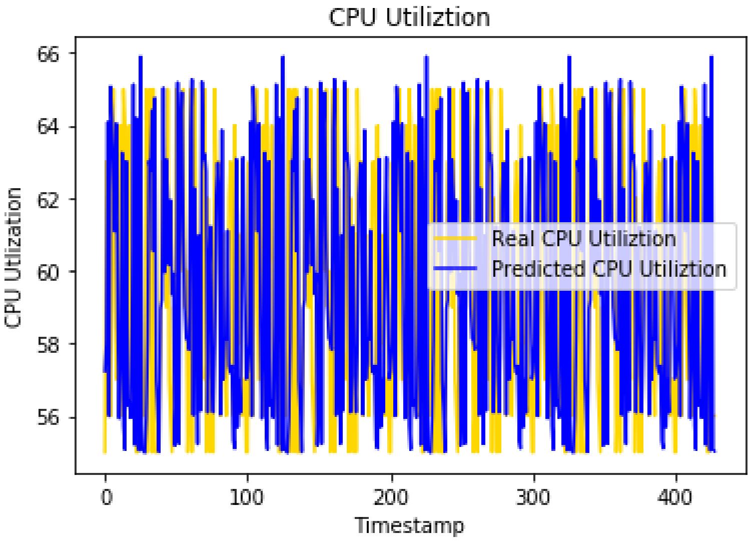


Table 5. Computational values for different workloads

Sr. No	CPU Utilization [in Percentage]	Operator		Customer	
		MAE	RMSE	MAE	RMSE
1	Upto 45%	0.79	0.91	0.81	0.98
2	40% to 60%	0.78	0.92	0.83	0.99
3	Above 60%	0.92	1.01	0.93	1.02

- Federated cloud monitoring
- The novel network architecture of cloud computing
- Workload generator for different cloud scenario

These points are explained briefly:

1. **Effectiveness (A. Botta et al., 2011):** Used to identify the actual causes besides any of the phenomenon happened in cloud environs, such as the root cause analysis is used to identify the main triggering event that is required for an outcome.
2. **Efficiency (Pocatilu Paul et al., 2010):** Efficiency is needed for data management, as large numbers of data will be generated as a log file. It has to be managed without too much burden in cloud infrastructure.
3. **Cost and Energy-Efficient Monitoring (Lee et al., 2012):** Activities related to the monitoring will always be associated with computing and resources related to communications. Hence cost and energy both will be incurred and need to be managed.
4. **Cross-Layer Monitoring (Zeginis C et al., 2013):** The cloud structure is complex and consists of several layers. These layers allow the separation of function and modularity. These different layers impose various limits on the monitoring systems, which can be reduced.

Table 6. Comparative analysis with existing schemes

Platform	Scalability	Elasticity	Adaptability	Timeliness	Automaticity	Comprehensiveness	Extensibility	Intrusiveness	Resilience	Reliability	Availability
Nagios (Aceto, G et al.,2013)							Y				
OpenNebula (http:// opennebula. org.2019)	Y		Y						Y		
Could stack(http:// www. cloudstack. org/.,2019)				Y							
Nimbus (https://www. nimbusproject. org/,2019)					Y						
PCMONS (De Chaveset al.,2011)							Y				
DARGOS (Povedano et al.,2013)			Y				Y	Y			
Hyper-HQ (A. Corradiet al.,2012)	Y					Y					
Sensu (Pavlik, J et al.,2014)	Y					Y					
Proposed Approach: SLAMMP	Y	Y	Y	Y	Y	Y			Y	Y	Y

Note: Y stands for Yes

5. **Federated Cloud Monitoring (Massonet P et al., 2011):** In dissimilar Cloud monitoring infrastructures, there is a great heterogeneity of tools, systems, and swapped data. Hence monitoring of Federated Clouds is part of ongoing research.
6. **The Novel Network Architecture of Cloud Computing (Wan, J et al., 2014):** The cloud grounded network is used for distributed enterprise networks, with the help of highly resilient and multi-tenant applications. Monitoring results should be improved and adapted to calculate and control these new network set-ups.
7. **Workload Generator for Different Cloud Scenario (Bahga et al., 2011):** The workload generation and modeling can be done via synthetic and real workloads, the important challenge is how specifically the workloads acts in the real situations/ cloud scenarios.

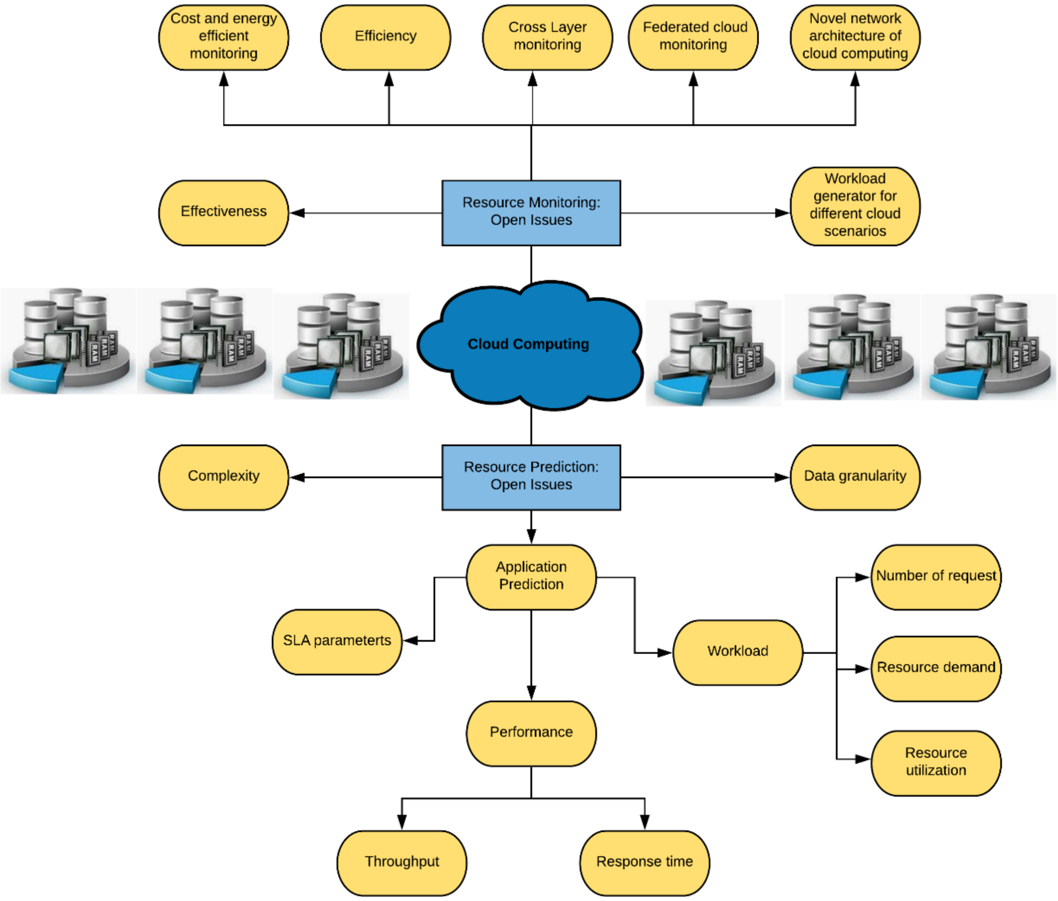
5.1 Issues Concerning the Cloud Resource Prediction

- Complexity
- Data granularity
- Application prediction

These points are explained briefly in Figure 26:

1. **Complexity (Azodolmolky et al., 2013):** Every model for prediction requires the estimation of future conduct. Hence the space and time complexities of the prediction model should be judicious.
2. **Data Granularity(Pedrycz et al., 2014):** This emphasis on the concept of which resources should be monitored, the length of intervals for the sampling data, coarse-grained and to identify the reason that leads to loss of dynamism (system behavior in various situations).
3. **Application Prediction (Li. et al, 2011):** This is prediction w.r.t the various applications that are executed in the cloud; this can be classified as SLA parameters, performance, and workload. SLA parameters have been discussed before and can be classified as application SLA and infrastructural SLA(Prasad V K & Bhavsar M, 2019). Performance indicates the behavior of the system and will be calculated via throughput and response time. The workload is calculated in terms of several requests, resource demand, and resource utilization (Prasad V K & Bhavsar M, 2020).

Figure 26. Open issues in cloud resource monitoring and prediction


Cost and energy efficient monitoring
Efficiency
Cross Layer monitoring
Federated cloud monitoring
Novel network architecture of cloud computing
Effectiveness
Resource Monitoring: Open Issues
Workload generator for different cloud scenarios
Cloud Computing
Complexity
Resource Prediction: Open Issues
Data granularity
Application Prediction
SLA parameters
Workload
Number of request
Resource demand
Resource utilization
Performance
Throughput
Response time

6. CONCLUSION AND FUTURE DIRECTIONS

We believe in the approaching future; CC will impact the medical field infrastructure for the organizations to make a great move/contribution. To satisfy the end-users demand, SLA violations should be avoided by the CSP. Most of the research proposed the solutions or explanations of violations after they have occurred, this research paper solves this by making use of a proactive approach using the mechanism of monitoring and prediction. RL and LSTM have been used to implement the same. The taxonomy of the SLAMMP framework work has been presented based on four different perceptions: (1) Workload generation (2) Identification of the present status of the CC resources (3) Management of the SLA (4) Prediction of the utilized resources. After this, every point (phase) is implemented in dynamic cloud environments. The parameters that have been used here is CPU Utilization for the management of the resources. The performance metrics show that the SLAMMP works fine under dynamic resource handling scenarios, and the obtained results indicate the supremacy of the proposed framework w.r.t the CPU utilization parameter. As future work, the concept of hierarchical reinforcement learning (HRL) can be used for monitoring of the resources and to identify the present status of the cloud.

REFERENCES

Aceto, G., Botta, A., De Donato, W., & Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks*, *57*(9), 2093–2115. doi:10.1016/j.comnet.2013.04.001

Ali, O., Shrestha, A., Soar, J., & Wamba, S. F. (2018). Cloud computing-enabled healthcare opportunities, issues, and applications: A systematic review. *International Journal of Information Management*, *43*, 146–158. doi:10.1016/j.ijinfomgt.2018.07.009

Azodolmolky, S., Wieder, P., & Yahyapour, R. (2013). Cloud computing networking: Challenges and opportunities for innovations. *IEEE Communications Magazine*, *51*(7), 54–62. doi:10.1109/MCOM.2013.6553678

Bahga, A., & Madisetti, V. K. (2011). Synthetic workload generation for cloud computing applications. *Journal of Software Engineering and Applications*, *4*(07), 396–410. doi:10.4236/jsea.2011.47046

Baughman, M., Haas, C., Wolski, R., Foster, I., & Chard, K. (2018, June). Predicting Amazon spot prices with LSTM networks. In *Proceedings of the 9th Workshop on Scientific Cloud Computing* (p. 1). ACM. doi:10.1145/3217880.3217881

Beloglazov & Buyya. (2010). Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. *MGC@ Middleware, 4*.

Beloglazov, A., & Buyya, R. (2010, May). Energy efficient allocation of virtual machines in cloud data centers. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* (pp. 577-578). IEEE. doi:10.1109/CCGRID.2010.45

Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation*, *24*(13), 1397–1420. doi:10.1002/cpe.1867

Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., & Zanella-Béguelin, S. et al. (2016, October). Formal verification of smart contracts: Short paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security* (pp. 91-96). ACM. doi:10.1145/2993600.2993611

Botta, A., Pescapè, A., Guerrini, C., & Mangri, M. (2011). A customer service assurance platform for mobile broadband networks. *IEEE Communications Magazine*, *49*(10), 101–109. doi:10.1109/MCOM.2011.6035823

Brabra, H., Mtibaa, A., Sliman, L., Gaaloul, W., Benatallah, B., & Gargouri, F. (2016, October). Detecting cloud (anti) patterns: OCCI perspective. In *International Conference on Service-Oriented Computing* (pp. 202-218). Springer. doi:10.1007/978-3-319-46295-0_13

Corradi, A., Foschini, L., Povedano-Molina, J., & Lopez-Soler, J. M. (2012). DDSenabled Cloud management support for fast task offloading. *Computers and Communications (ISCC), 2012 IEEE Symposium on*, 67–74.

De Chaves, S. A., Uriarte, R. B., & Westphall, C. B. (2011). Toward an architecture for monitoring private clouds. *IEEE Communications Magazine*, *49*(12), 130–137. doi:10.1109/MCOM.2011.6094017

Debusmann, M., & Keller, A. (2003, March). SLA-driven management of distributed systems using the common information model. In IFIP/IEEE Eighth International Symposium on Integrated Network Management, 2003 (pp. 563-576). IEEE. doi:10.1109/INM.2003.1194211

Di Martino, C., Sarkar, S., Ganesan, R., Kalbarczyk, Z. T., & Iyer, R. K. (2017). Analysis and diagnosis of SLA violations in a production SaaS cloud. *IEEE Transactions on Reliability*, *66*(1), 54–75. doi:10.1109/TR.2016.2635033

Duggan, M., Flesk, K., Duggan, J., Howley, E., & Barrett, E. (2016, August). A reinforcement learning approach for dynamic selection of virtual machines in cloud data centres. In *2016 Sixth International Conference on Innovative Computing Technology (INTECH)* (pp. 92-97). IEEE. doi:10.1109/INTECH.2016.7845053

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018, April). Deep reinforcement learning that matters. *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hussain, W., Hussain, F. K., & Hussain, O. K. (2016). SLA Management framework to avoid violation in cloud. In *International Conference on Neural Information Processing*, (pp. 309-316). Springer. doi:10.1007/978-3-319-46675-0_34

Jing, H., Zhang, Y., Zhou, J., Zhang, W., Liu, X., Min, G., & Zhang, Z. (2018, July). LSTM-Based Service Migration for Pervasive Cloud Computing. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 1835-1840). IEEE.

Lee, Y. C., & Zomaya, A. Y. (2012). Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, *60*(2), 268–280. doi:10.1007/s11227-010-0421-3

Li, A., Zong, X., Kandula, S., Yang, X., & Zhang, M. (2011, August). CloudProphet: Towards application performance prediction in cloud. *Computer Communication Review*, *41*(4), 426–427. doi:10.1145/2043164.2018502

Liu, B., Wang, L., Liu, M., & Xu, C. (2019). *Lifelong federated reinforcement learning: a learning architecture for navigation in cloud robotic systems*. arXiv preprint arXiv:1901.06455

Liu, H., Liu, S., & Zheng, K. (2018). A reinforcement learning-based resource allocation scheme for cloud robotics. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 17215–17222. doi:10.1109/ACCESS.2018.2814606

Massonet, P., Naqvi, S., Ponsard, C., Latanicki, J., Rochwerger, B., & Villari, M. (2011, May). A monitoring and audit logging architecture for data location compliance in federated cloud infrastructures. In *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum* (pp. 1510-1517). IEEE. doi:10.1109/IPDPS.2011.304

Maurya, K., & Sinha, R. (2013). Energy conscious dynamic provisioning of virtual machines using adaptive migration thresholds in cloud data center. *International Journal of Computer Science and Mobile Computing*, *2*(3), 74–82.

Muthusamy, V., Jacobsen, H. A., Chau, T., Chan, A., & Coulthard, P. (2009, November). SLA-driven business process management in SOA. In *Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research* (pp. 86-100). IBM Corp.

Nayak, S., Narendra, N. C., Shukla, A., & Kempf, J. (2018, July). Saranyu: Using smart contracts and blockchain for cloud tenant management. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)* (pp. 857-861). IEEE.

Paschke, A., & Schnappinger-Gerull, E. (2006). A Categorization Scheme for SLA Metrics. *Service Oriented Electronic Commerce, 80*(25-40), 14.

Pavlik, J., Sobeslav, V., & Horalek, J. (2014, July). Statistics and analysis of service availability in cloud computing. In *Proceedings of the 18th International Database Engineering & Applications Symposium* (pp. 310-313). ACM. doi:10.1145/2628194.2628222

Pedrycz, W., & Chen, S.-M. (Eds.). (2014). *Information granularity, big data, and computational intelligence* (Vol. 8). Springer.

Pocatilu, P., Alecu, F., & Vetrici, M. (2010). Measuring the efficiency of cloud computing for e-learning systems. *WSEAS Transactions on Computers*, *9*(1), 42–51.

Povedano-Molina, J., Lopez-Vega, J. M., Lopez-Soler, J. M., Corradi, A., & Foschini, L. (2013). DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant Clouds. *Future Generation Computer Systems*, *29*(8), 2041–2056. doi:10.1016/j.future.2013.04.022

Prasad, V. K. (2013). Method and system for detecting fraud in credit card transaction. *International Journal of Innovative Research in Computer and Communication Engineering*, *1*(5).

Prasad, V. K., & Bhavsar, M. (2017, August). Efficient Resource Monitoring and Prediction Techniques in an IaaS Level of Cloud Computing: *Survey. In International Conference on Future Internet Technologies and Trends* (pp. 47-55). Springer.

Prasad, V. K., & Bhavsar, M. (2019). Preserving SLA Parameters for Trusted IaaS Cloud: An Intelligent Monitoring Approach. *Recent Patents on Engineering*, *13*, 1. doi:10.2174/1872212113666190315162646

Prasad, V. K., & Bhavsar, M. D. (2020). Monitoring IaaS Cloud for Healthcare Systems: Healthcare Information Management and Cloud Resources Utilization. *International Journal of E-Health and Medical Communications*, *11*(3), 54–70. doi:10.4018/IJEHMC.2020070104

Prasad, V. K., Shah, M., Patel, N., & Bhavsar, M. (2018). Inspection of Trust Based Cloud Using Security and Capacity Management at an IaaS Level. *Procedia Computer Science*, *132*, 1280–1289. doi:10.1016/j.procs.2018.05.044

Rimba, P., Tran, A. B., Weber, I., Staples, M., Ponomarev, A., & Xu, X. (2017, April). Comparing blockchain and cloud services for business process execution. In *2017 IEEE International Conference on Software Architecture (ICSA)* (pp. 257-260). IEEE. doi:10.1109/ICSA.2017.44

Scoca, V., Uriarte, R. B., & De Nicola, R. (2017, June). Smart contract negotiation in cloud computing. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)* (pp. 592-599). IEEE. doi:10.1109/CLOUD.2017.81

Shah, T., Yavari, A., Mitra, K., Saguna, S., Jayaraman, P. P., Rabhi, F., & Ranjan, R. (2016). “Remote health care cyber-physical system: Quality of service (QoS) challenges and opportunities.” IET Cyber-Physical Systems. *Theory & Applications*, *1*(1), 40–48.

Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing*, *14*(2), 217–264. doi:10.1007/s10723-015-9359-2

Somula, R., & Anilkumar, C. (2019). B. Venkatesh, Aravind Karrothu, CS Pavan Kumar, and R. Sasikala. “Cloudlet services for healthcare applications in mobile cloud computing. In *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology* (pp. 535–543). Springer. doi:10.1007/978-981-13-1610-4_54

Stamou, K., Kantere, V., & Morin, J.-H. (2013). SLA data management criteria. In *2013 IEEE International Conference on Big Data*, (pp. 34-42). IEEE. doi:10.1109/BigData.2013.6691769

Tan, Z. X., Goel, A., Nguyen, T. S., & Ong, D. C. (2019, May). A multimodal LSTM for predicting listener empathic responses over time. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)* (pp. 1-4). IEEE. doi:10.1109/FG.2019.8756577

Vetter, A. (2016). Detecting operator errors in cloud maintenance operations. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, (pp. 639-644). IEEE, 2016. doi:10.1109/CloudCom.2016.0110

Wan, J., Lin, K., Zeng, D., Li, J., Xiang, Y., Liao, X., . . . Liu, Z. (2016). Cloud computing, security, privacy in new computing environments. In *Proceedings of 7th international conference, cloudcomp 2016 and first international conference*. SPNCE.

Wan, J., Zhang, D., Sun, Y., Lin, K., Zou, C., & Cai, H. (2014). VCMIA: A novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing. *Mobile Networks and Applications*, *19*(2), 153–160. doi:10.1007/s11036-014-0499-6

Wang, X., Xu, W., & Jin, Z. (2017, February). A hidden Markov model based dynamic scheduling approach for mobile cloud telemonitoring. In *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)* (pp. 273-276). IEEE. doi:10.1109/BHI.2017.7897258

Xu, D. Y., Yang, S. L., & Liu, R. P. (2013). A mixture of HMM, GA, and Elman network for load prediction in cloud-oriented data centers. *Journal of Zhejiang University SCIENCE C*, *14*(11), 845–858. doi:10.1631/jzus.C1300109

Yuan, S., Das, S., Ramesh, R., & Qiao, C. (2018). Service Agreement Trifecta: Backup Resources, Price and Penalty in the Availability-Aware Cloud. *Information Systems Research*, *29*(4), 947–964. doi:10.1287/isre.2017.0755

Zeginis, C., Kritikos, K., Garefalakis, P., Konsolaki, K., Magoutis, K., & Plexousakis, D. (2013, September). Towards cross-layer monitoring of multi-cloud service-based applications. In *European Conference on Service-Oriented and Cloud Computing* (pp. 188-195). Springer. doi:10.1007/978-3-642-40651-5_16

Zhao, Y., Calheiros, R. N., Bailey, J., & Sinnott, R. (2016). SLA-based profit optimization for resource management of big data analytics-as-a-service platforms in cloud computing environments. In *2016 IEEE International Conference on Big Data (Big Data)*, (pp. 432-441). IEEE. doi:10.1109/BigData.2016.7840634

Vivek Kumar Prasad is pursuing Ph.D. in Computer Science and Engineering, Nirma University, Ahmedabad, Gujarat, India. He received Master of Technology in Computer Science and Engineering from MVJCE, under VTU, Bangalore, India in 2010. His research area includes resource management in Cloud Computing, Trust Management, and Software Engineering.

Madhuri D. Bhavsar is working as Professor and Head in the Computer Science and Engineering department at Institute of Technology, Nirma University, Ahmedabad, Gujarat, India. She received Ph.D. in 2012 from Nirma University, Ahmedabad, Gujarat, India. She has authored more than 30 technical research papers published in leading international conferences and peer-reviewed international journals. She has also published three books and have five projects approved from government organizations such as SAC-ISRO, BRNS, and DST, etc. She has guided many students leading to M.Tech and guiding students leading to Ph.D. Her current area of interest includes Cloud Computing, High-Performance Computing, Advanced Computing, and Software Engineering.