International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-10 Issue-1, May 2021

Fuzzy Reinforcement Learning Model for Resource Adaptation in IoT

Daneshwari I. Hatti, Ashok V. Sutagundar

Abstract: ***The automation of several applications is creating engrossment in Internet of Things (IoT). The prerequisite for employing IoT in daily life is the ability to interact with devices technologies and process the sensed data. The difficulty to process the sensed data with scarce resources for diverse requests is challenging. Learning the behaviour of changing demand and processing with available resources has resulted in adaptation policy. Adaptation policy by employing Reinforcement learning and Fuzzy logic is proposed to adapt the resources for executing the tasks. Prioritization of task, allocating of resources to the tasks by adapting with available resources with assured Quality of Service (QoS) is performed. Fuzzy Q learning Adaptation Algorithm (FQAA) is designed for evaluating resource adaptation mechanisms to execute the heterogenous tasks. The algorithm with different configuration is simulated using Ifogsim and python. It is compared with traditional method that is without adaptation mechanism, performs better compared to other algorithms in terms of Cost, energy consumption and latency.***

Keywords: Adaptation, Ifogsim, Resource cost, Energy Consumption, Reinforcement Learning, Fuzzy logic.

I. INTRODUCTION

The Internet of Things (IoT) possess objects that are equipped with sensors, actuators and processors which helps in communication with each other [1]. Due to growth in IoT devices, huge data processing at the devices is difficult. Since devices does not have sufficient resources to process the sensed data. Hence processing of this data is done at Cloud, but traffic increases to wards cloud. Delay also increases and cannot serve the critical applications for example healthcare and any disasters. To overcome this fog computing, edge computing is employed so that requests are processed towards the edge devices reducing delay and traffic is distributed among fog and cloud.

Manuscript received on May 08, 2021.
Revised Manuscript received on May 15, 2021.
Manuscript published on May 30, 2021.
* Correspondence Author
Daneshwari I. Hatti*, Assistant Professor, Department of Electronics and Communication Engineering, BLDEA's V. P. Dr. P. G. Halakatti College of Engineering and Technology, Affiliated to VTU Belagavi, Vijayapur, Karnataka, India. Email: daneshwari_hatti@yahoo.co.in
Ashok V. Sutagundar, Assistant Professor, Department of Electronics and Communication Engineering, BEC Bagalkot, Karnataka, India. Email: sutagundar@gmail.com

In IoT, devices have there own requirement of computational resources. Efficient use of resources by allocation, brokering, predicting the resources, provisioning and adapting to the new requests with less latency, high throughput, less energy consumption and efficient utilization of bandwidth is essential.

Adaptation and allocation of resources through multi agents by embedding intelligence in the form of software programs across the devices to obtain optimal resource utilization and better QoS for the end users.

The sensed data is collected and processed at fog layer due to insufficient of resources at the device end. The devices are authenticated and then resources are provisioned by fog if resources are insufficient to process the request it is forwarded to cloud. Fog computing reduces latency and congestion towards cloud as few requests are forwarded to cloud. The provision of resources is based on cost, energy consumption to achieve better QoS without violating SLA requirements of requests.

Due to increase in heterogenous requests and changing workload adaptation of resources plays a vital role. In few cases at some point of time device failure occurs then it has to be monitored even in the absence of human supervision, so to deal such situations self-adaptation is required. The changing environment requires attention of the human to monitor and control the application hence machine learning is employed. Machine learning aids in controlling the environment with less human assistance.

In order to ensure adaptation goals, the adaptation mechanism is used in context of IoT. It adapts to the changing environment and demands by taking proper action with the available resources.

Self-adaptation is an essential property of smart objects in the context of Internet of Things. It enables them to self-configure and adapt to extreme conditions while meeting system goals such as comfort, automation, protection, and safety. Self-adaptation mechanisms driven by adaptation goals with efficient utilization of resources is vital. When adaptation goals change at run-time, both monitoring and self-adaptation mechanisms is essential as it has to deal with changing goals and address monitoring requirements.

In this work, self-adaptation in system for monitoring the leakages in tanks to assure quality-of-service (QoS) is proposed. The proposed work ensures proper utilization of resources meeting adaptation goals of the user. The system ensures the adaptation objectives, monitors and concerns the adaptation strategies with assured QoS.

The paper is organized as follows. Section II presents related works. In Section III, proposed model, RL, Fuzzy logic controller and Fuzzy RL adaptation algorithm is discussed. Section IV presents the results obtained from the proposed model. Section V concludes the work.

A *Our Contributions*

The proposed model ensures adaptation of resources to meet the user requirement. According to survey RL adapts according to the changes in the workload but increases number of state and converges after several iterations. To overcome the problem Fuzzy logic is used. The main objective of the work is to process the request by adapting with network resources and adapt the services according to the user resources availability. Self-adaptation by employing learning model and decentralized architecture for monitoring leakages in tanks is carried out. Adaptation at the user and devices to process the tasks assuring Quality of Service (QoS) is performed through Fuzzy RL model.

II. RELATED WORK

Resource management in IoT have several sub problems namely resource discovery, selection, provision, monitor, scheduling, allocation and adaptation of resources to the services based on their requirement. Apart from this managing the dynamic workload to fulfil the heterogenous requirement of the user is essential. Machine learning plays a vital role in understanding the behaviour of the devices and incoming requests. Several work for adaptation in IoT is discussed in this section.

In [2] authors have developed self-adaptive fuzzy logic controller with two reinforcement learning (RL) approaches for reducing application cost and guaranteeing service-level agreements (SLAs) to react to workload fluctuations and changing resources demand.

In [3] different perspectives of self-adaptive systems is discussed. In [4] authors have presented DeltaIoT, an exemplar for enhancing the IoT network with self-adaptation to automate the tasks.

In [5] authors concentrated on proactively adapting the devices to align with context fluctuations.

Authors [6] modelled IAS-QN that is Queuing Networks (QN) for modelling IoT Architectural Self-adaptation (IAS) to control the levels and their interaction for performing both architectural and environmental adaptations.

In [7] authors proposed adaptive framework that uses Reinforcement Learning to combine adaptation mechanisms, specified in the development phase, during runtime in order to optimize the performance of applications and services.

In [8] authors have proposed QoS-Driven Self-adaptation for Critical IoT-Based Systems that is adapting to the changes in the context and providing functional and non-functional requirements to the user.

In [9] authors discussed the possible ways of adapting at IoT devices level, microservices level, due to dynamic resource demands and at application level due to the changing user goals.

According to several works addressed in context of self-adaptation, RL and fuzzy play an important role in monitoring dynamic changes in the environment. In this work Fuzzy RL model is proposed to self-adapt the system and provide services meeting user criteria as well as adaptation in placement of modules to process the request so as to reduce delay, resource cost, energy consumption and network usage.

III. MATERIALS AND METHODS

The proposed work detects the leakage in tanks, bridges using IoT and action is taken by embedding the intelligence in the devices using Reinforcement learning and Fuzzy logic. The application module to be used and allocating the network resources for various application modules is decided by monitoring the resources available at user and devices in the network environment. Adaptation of resources is done according to the changes in the environment for executing the tasks within their deadline. This section discusses network model, Fuzzy RL model, example scenario.

A *Network Model*

It comprises three layers IoT, Fog and Cloud as depicted in Fig. 1. IoT comprising high end cameras and sensors, fog layer comprising router, switches, micro servers and cloud constituting high end servers. The sensed data is processed at IoT layer if resources are available. Else it will be offloaded to fog device and cloud. Maximum data is processed at fog instead of sending for cloud to avoid traffic and delay [10]. Adaptation according to the end user capability and adaptation with respect to placement of application modules on devices to process the tasks is performed.

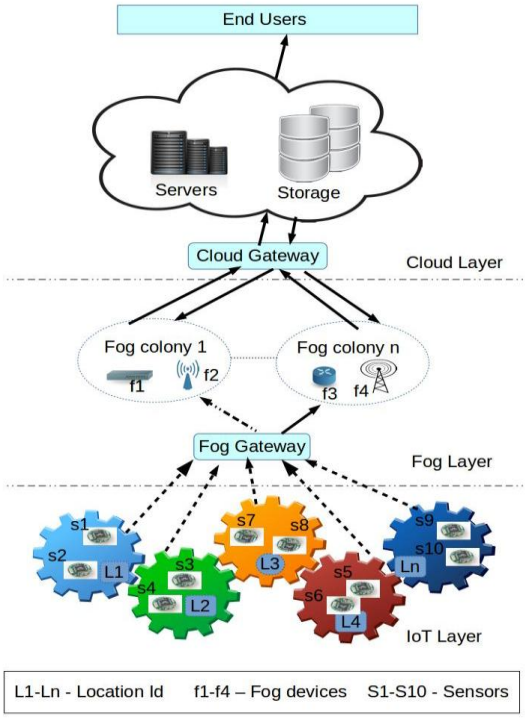

Fig. 1. Network Environment

B Fuzzy Reinforcement Learning based Adaptation of Resources in IoT

This section discusses adaptation of resources and services by proposed Fuzzy RL model. Q learning is employed for learning the states of the environment and act accordingly. The action or services offered is adapted by monitoring the resources at the user and processing of tasks is distributed among various devices and adapted by monitoring the devices resources. RL is type of machine learning [11] and uses off policy for updating the table. The updating of table is decided by several factors that is learning rate, reward discount factor, number of epochs.

Reinforcement Learning (RL): It employs feedback mechanism wherein agent learns from environment and take appropriate actions resulting in reward. The reward is positive if action taken is satisfactory to the environment. Dynamic changes in the environment are monitored by the RL model by employing agents. RL updates Q table with state action pair for various states through FLC as shown in Fig. 2. It learns with learning rate varying from 0.1 to 0.9. Policy is required for finding the action for current state through Fuzzy Logic Controller (FLC). As states occurring in RL approaches to infinity to overcomes the infinite states fuzzy is used. Fuzzy Logic reduces to finite states and aids in fast update of Q table. For every state the action obtained is updated in Q table. The action initiated by user helps in monitoring the tank, if user is satisfied and goal is achieved the reward is positive. If the user is not satisfied the reward is negative, hence new action is taken for next time.

Fuzzy Logic Controller (FLC): It has capability of reasoning and decision making for vague inputs [10]. It tries to obtain decision according to human thinking. In this context FLC is employed for reducing number of states in RL model. FLC applies set of rules for classifying input sensed data, processes and allocates the required resources. FLC uses *If then* rules for evaluating set of conditions for single state. FLC is of Mamdani type and aggregates all the rules using centre of gravity method. The output decides the type of action in terms poor, good and high-quality video with duration. FLC is used for applying policy to help RL model to learn and find the action. FLC act as decision maker that is in deciding type of request to process with how much resources. Resource requirement is adapted according to the resource availability at the user and processing of task is decided by monitoring the resources at IoT device and Fog device. Adaptation takes place at device and User based on the decision of FLC. RL learns and updates the Q table with state-action pair for every new state. FLC constitutes classifying input data and output data in form of membership function as depicted in Fig. 3. Set of rules are framed and aggregated for finding optimal decision that is duration of video and quality of video to monitor the tank.

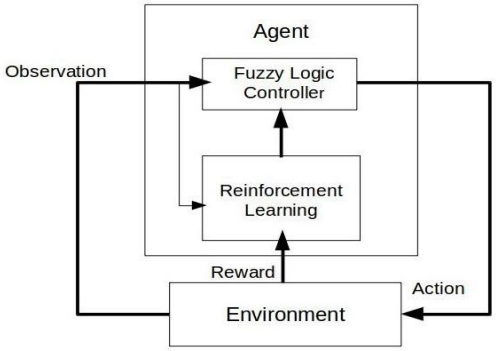

Fig.2. Fuzzy RL model

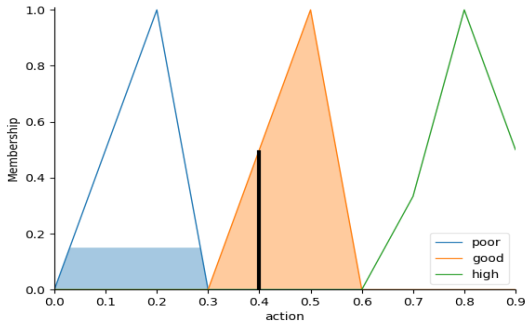

Fig.3. Membership function for output

C Algorithm

Algorithm: Fuzzy Q Learning Adaptation Algorithm

Input: Tank level, No of requests/tasks, Location id, Battery level, Internet speed of user

Output: Quality of video with specified duration

Begin

1. Deploy the sensors, sense the tank level periodically after x min.
2. Collect the user resources available
3. Check resources of the devices is greater than the required resources for processing
4. **If** $R_{dev} > R_{procRL}$
5. Place Fuzzy RL module on device
6. Assign priority for the tasks
7. Processes the requests through FLC considering the user resources availability
8. Update of Q table with new state action pair using equation 1

$$Q(S,A) = Q(s,a) + \alpha * (r + \gamma * \max Q(s+1,a) - Q(s,a)) \quad (1)$$

9. Obtain reward R
   a. **If** R=+1
   b. Process next request
   c. Else
   d. Update with different action
   e. Endif
10. **Else**
11. Find appropriate fog device for processing and if not available process at cloud
12. **Endif**

End

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication

The proposed algorithm is applied for monitoring the leakages in tanks which is discussed in case study.

D Case study: Monitoring and controlling leakage in Irrigation Tanks

Adaptation is achieved by placement of application modules according to the availability of resources. The application includes monitoring of leakages in irrigation tanks consists of modules for pre-processing the data, applying reinforcement learning module, fuzzy logic controller for updating the policy, reward estimator and end user for controlling the leakages in tanks.

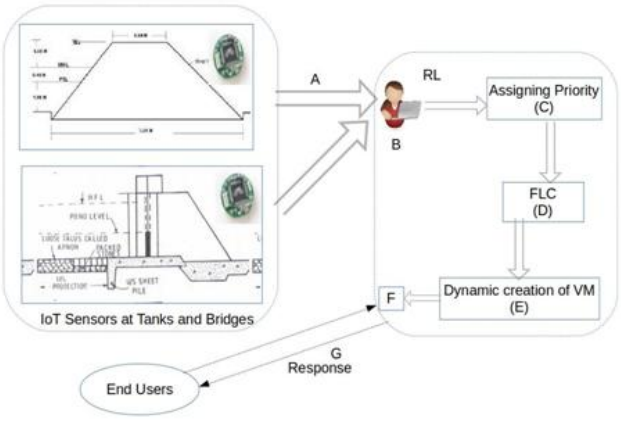

Fig. 4. Example scenario

Fig. 4 shows the example scenario in which water level of irrigation tank is sensed by sensor deployed across the guard stone of tanks. The sensed data is collected by RL agent that is step (A), processing of data that is if it is above Full tank level is done in step (B) it is forwarded to prioritize in step (C), after prioritizing it is processed by FLC in step (D), FLC decide the action to be taken for monitoring the tank if it above or below FTL, in step (E) VM is created for processing the request as user is scarce of resources. In step (F) user uses the resources of VM and monitors the tank and controls the leakage through video conferencing or by using CCTV footages sent by the camera placed at tank area. In step (G) if user is satisfied the reward is positive and the state occurred is updated with action taken by the user or else the process is repeated till the reward is positive.

IV. RESULTS AND DISCUSSION

The simulation of the proposed model is performed for various cases and the results are obtained from Ifogsim and python.

Ifogsim [12] ensures resource management s in two levels that is placement and scheduling. The application modules are placed on the fog device for processing the requests. The modules placement is done according to the proposed adaptation policy. After placement the scheduling of resources is performed by assigning the priority of the task in updateAllocatedMips method of class fog device. After placing and scheduling the application modules are placed on the fog device for execution of task.

AppModule class of iFogSim is used to create the application modules. As illustrated in Fig. 5, the modules have data dependencies and the dependences are modelled using AppEdge class in iFogSim. AppLoop class helps in controlling the control loop of application. The application is the tank level sensed by sensor across tanks and an end user acting as actuator displays the action taken. The specifications of the edges of these modules are tabulated in table I.

Table I: Configuration of devices

Sl.No	Configuration and Specifications	
A	**Hardware units of servers of Fog and Cloud**	
1	Architecture	X86
2	VM	Xen
3	Operating System	Linux
B.	**Configuration of data centres and fog device**	
1	Memory Size (Mb)	204800
2	Number of processors	4
3	Storage (Mb)	1000000
4	Available bandwidth (Kbps)	10000
5	Memory cost ($/s)	0.05
6	Cost per VM ($/Hr)	0.1
7	Storage cost ($/s)	0.001
8	Network length (KB)	500-1000
9	CPU Length (MIPS)	100-10000
10	Video quality	140p-1040p
11	Data transfer cost ($/Gb)	0.01
12	Latency(ms)	5-30
13	Power (W) (idle) cloud; fog	83.433; 82.44
14	Power (W) cloud; fog	107.339; 87.53

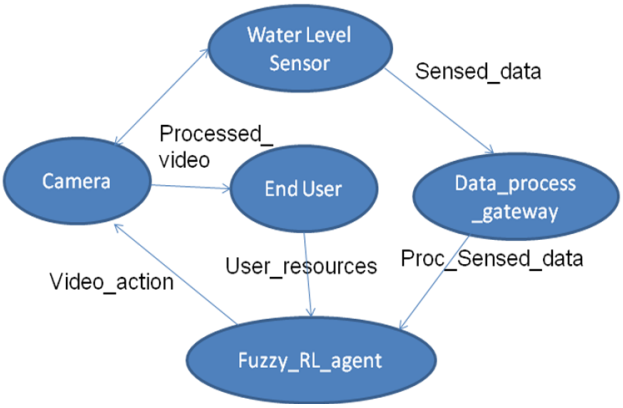

Fig. 5. Application Model of Adaptation mechanisms

Fig. 6 shows the topology used for simulation of proposed algorithm in Ifogsim. It has 4 levels, and at the bottom level each gateway is connected with two sensors. Similar configuration is used by varying number of requests and simulated. The parameters used for connecting those modules and edges is shown in Fig. 7.

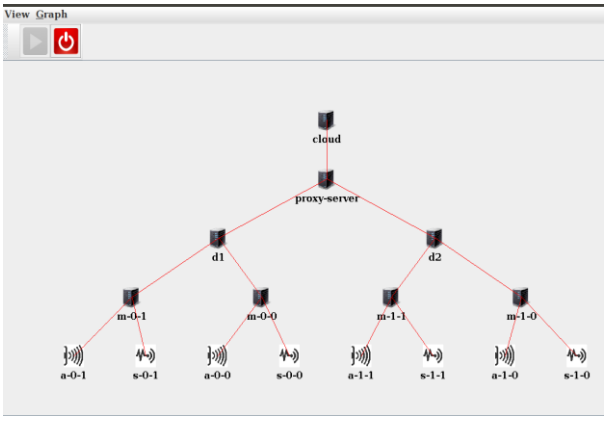

Fig. 6. Topology

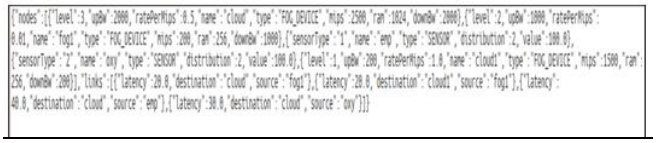

Fig.7. Network topology parameters

A Simulation procedure

The simulation procedure for evaluating the performance is as follows.

1. Generate the network environment deployed with sensors, camera and set the parameters.
2. Create application modules in Ifogsim
3. Apply adaptation policy to place modules on various fog devices
4. Process the requests according to the decision obtained from FLC.
5. Vary number of requests, place on fog device, place on cloud and compare both.
6. Estimate performance parameters namely latency, energy consumption, cost and execution time for both configurations.

Fig. 8 depicts Fuzzy RL Adaptation algorithm processed at device end performs many requests compared to traditional approach.

Fig. 9 depicts Fuzzy RL Adaptation algorithm processed at fog device performs many requests compared to traditional approach and processing at IoT device. It incurs less delay and network usage compared to traditional approach.

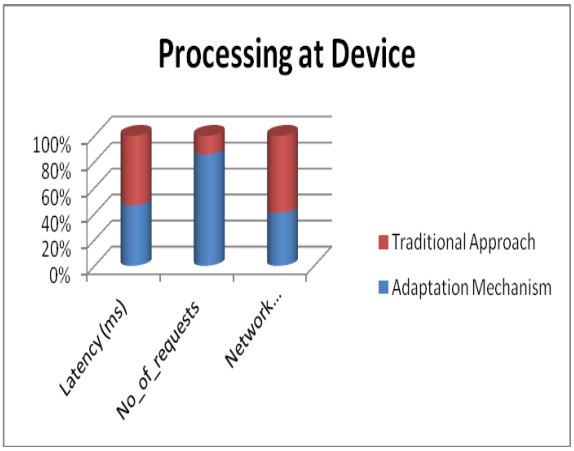

Fig. 8. Fuzzy RL Adaptation algorithm at device

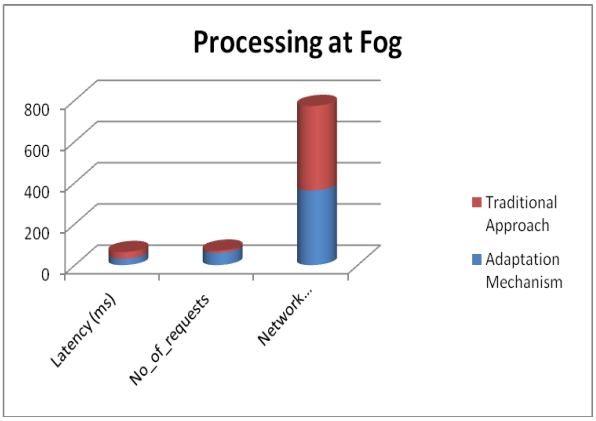

Fig. 9. Fuzzy RL Adaptation algorithm at Fog device

Fig. 10 depicts Fuzzy RL Adaptation algorithm processed at cloud that is IoT-Cloud framework uses more network resources and incurs more delay compared to processing at fog and IoT device. Compared to traditional approach adaptation mechanism performs better.

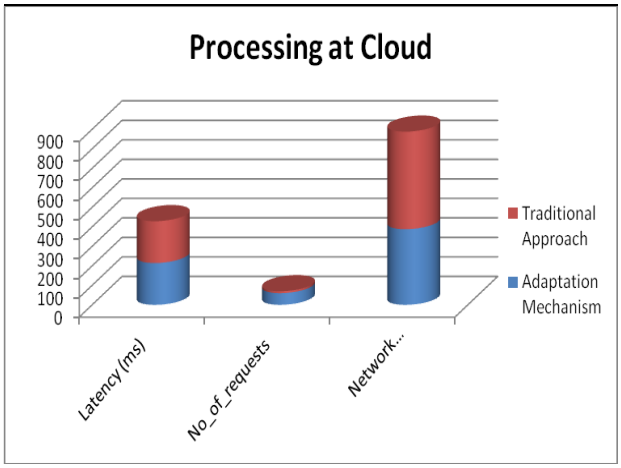

Fig. 10. Fuzzy RL Adaptation algorithm at Cloud

Fig. 11 depicts the execution time obtained for processing at IoT device, fog device and Cloud datacenter for various configurations. Config 1 indicates requests are obtained through two gateways, config 2 shows three requests, config 3 has 4 requests and config 4 has 6 requests obtained from different locations. As number of requests increases the execution time also increases because queuing of requests occurs. Comparatively execution of task at IoT device is faster compared to fog device and cloud datacenter. Hence proposed adaptation policy processes many requests at device then forward few requests to fog device if resources are insufficient at device end incurring reduced execution time and delay.

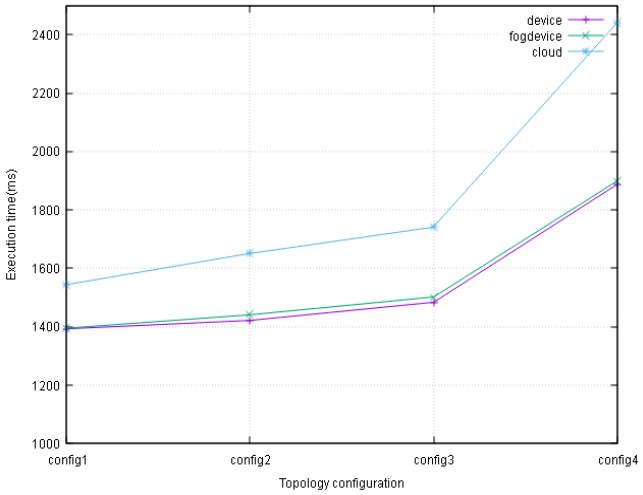

Fig.11: Simulation of various topologies with execution time (ms)

V. CONCLUSION AND FUTURE WORK

Resource adaptation and allocation mechanisms in IoT enhances the usage of resources. To utilize scarce resources of the devices and ensure QoS to the users, resource adaptation mechanism using Fuzzy RL algorithm is performed. Adaptation in two ways is fulfilled that is adaptation of services according to the availability of resources at the user and adaptation of processing the requests considering the resources of IoT devices. The proposed algorithm gives better results compared to traditional approach that is without RL and Fuzzy logic in terms of latency, energy consumption, resource cost and network usage. With adaptation algorithm number of requests with assured QoS is achieved. The extension of the work includes the evaluation of the scheme with other performance measures and implementing on testbed.

REFERENCES

1. P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *Journal of Electrical and Computer Engineering*, Jan. 26, 2017. https://www.hindawi.com/journals/jece/2017/9324035/ (accessed Feb. 11, 2021).
2. H. Arabnejad, C. Pahl, P. Jamshidi, and G. Estrada, "A Comparison of Reinforcement Learning Techniques for Fuzzy Cloud Auto-Scaling," *ArXiv170507114 Cs*, May 2017, Accessed: Feb. 07, 2021. [Online]. Available: http://arxiv.org/abs/1705.07114.
3. C. Krupitzer, F. M. Roth, S. VanSyckel, G. Schiele, and C. Becker, "A survey on engineering approaches for self-adaptive systems," *Pervasive Mob. Comput.*, vol. 17, pp. 184–206, Feb. 2015, doi: 10.1016/j.pmcj.2014.09.009.
4. M. U. Iftikhar, G. S. Ramachandran, P. Bollansee, D. Weyns, and D. Hughes, "DeltaIoT: A Self-Adaptive Internet of Things Exemplar," in *2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, Buenos Aires, Argentina, May 2017, pp. 76–82, doi: 10.1109/SEAMS.2017.21.
5. A. Achtaich, N. Souissi, R. Mazo, C. Salinesi, and O. Roudies, "Designing a Framework for Smart IoT Adaptations," in *Emerging Technologies for Developing Countries*, vol. 206, F. Belqasmi, H. Harroud, M. Agueh, R. Dssouli, and F. Kamoun, Eds. Cham: Springer International Publishing, 2018, pp. 57–66.
6. M. T. Moghaddam, E. Rutten, P. Lalanda, and G. Giraud, "IAS: An IoT Architectural Self-adaptation Framework," in *Software Architecture*, vol. 12292, A. Jansen, I. Malavolta, H. Muccini, I. Ozkaya, and O. Zimmermann, Eds. Cham: Springer International Publishing, 2020, pp. 333–351.
7. M. F. Argerich, "Learning based Adaptation for Fog and Edge Computing Applications and Services," p. 65, 2018.
8. A. Gatouillat, Y. Badr, and B. Massot, "QoS-Driven Self-adaptation for Critical IoT-Based Systems," in *Service-Oriented Computing – ICSOC 2017 Workshops*, vol. 10797, L. Braubach, J. M. Murillo, N. Kaviani, M. Lama, L. Burgueño, N. Moha, and M. Oriol, Eds. Cham: Springer International Publishing, 2018, pp. 93–105.
9. M. D. Sanctis, H. Muccini, and K. Vaidhyanathan, "Data-driven Adaptation in Microservice-based IoT Architectures," in *2020 IEEE International Conference on Software Architecture Companion (ICSA-C)*, Salvador, Brazil, Mar. 2020, pp. 59–62, doi: 10.1109/ICSA-C50368.2020.00019.
10. K. S. Awaisi, A. Abbas, S. U. Khan, R. Mahmud, and R. Buyya, "Simulating Fog Computing Applications using iFogSim Toolkit," p. 29.
11. R. Dhaya, R. Kanthavel, F. Algarni, P. Jayarajan, and A. Mahor, "Reinforcement Learning Concepts Ministering Smart City Applications Using IoT," in *Internet of Things in Smart Technologies for Sustainable Urban Development*, G. R. Kanagachidambaresan, R. Maheswar, V. Manikandan, and K. Ramakrishnan, Eds. Cham: Springer International Publishing, 2020, pp. 19–41.
12. H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments," p. 22, 2016.

AUTHORS PROFILE

Daneshwari I. Hatti, working as Assistant Professor in BLDEA's V.P. Dr P.G.Halakatti College of Engineering and Technology, Vijayapur, Karnataka, India. She completed B. E and M.Tech from VTU Belagavi, and currently pursuing PhD in field of IoT at research centre BEC Bagalkot affiliated to VTU Belagavi. She has published in various international journals, conferences and book chapters in field of IoT. Her area of interest includes IoT, signal processing, VLSI, etc

Ashok V. Sutagundar, completed his M.Tech from VTU Belgavi, Karnataka. He has pursued PhD in the area of Content Based Information Retrieval in wireless Networks using Mobile Agents. Presently he is serving as Associate Professor, Department of Electronics and Communication Engineering, Basaveshwar Engineering College, Bagalkot, Karnataka. His areas of interest include Signal and system, Digital Signal Processing, Digital Image Processing, Multimedia Networks, Computer communication networks, Wireless networks, Mobile ad-hoc networks, Agent technology. He has published 27 papers in referred National/International Conferences and Journals.