International Journal of Communication Networks and Information Security

ISSN: 2073-607X, 2076-0930

*Volume **14** Issue **03 Year 2022** Page **138:149***

Centralized Cloud Service Providers in Improving Resource Allocation and Data Integrity by 4G IoT Paradigm

[1] Rahul Bhatt, [2] Mr. Rishi Shikka, [3] Manjunath C R, [4] Dr. SSPM Sharma, [5] Dr. Arvind Kumar Pandey, [6] Kiran Bala

[1] Assistant Professor, Department of Computer Science and Engineering, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India.

[2] Assistant Professor, Department of Electronics, Sanskriti University, Mathura, Uttar Pradesh, India.
Orcid Id- 0000-0002-6954-1951

[3] Associate Professor, Department of Computer Science and Engineering, Jain (Deemed-to-be University), Bangalore, India.
Orcid Id- 0000-0001-9533-7156

[4] Assistant Professor, Department of Mechatronics, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India.
Orcid Id- 0000-0002-1236-2661

[5] Assistant Professor, Department of Computer Science, ARKA JAIN University, Jamshedpur, Jharkhand, India.
Orcid Id- 0000-0001-5294-0190

[6] Assistant Professor, Department of Computer Science Engineering, Chandigarh Engineering College, Jhanjeri, India.
Orcid Id- 0000-0002-6340-3216

[1] socse.rahulbhatt@dbuu.ac.in, [2] rishisikka.ec@sanskriti.edu.in, [3] cr.manjunath@jainuniversity.ac.in, [4] sspm.sharma8977@paruluniversity.ac.in, [5] arvind.p@arkajainuniversity.ac.in, [6] kiran.j1638@cgc.ac.in

Article History

Received: 13 August 2022
Revised: 24 October 2022
Accepted: 12 November 2022

Abstract

Due to the expansion of Internet of Things (IoT), the extensive wireless, and 4G networks, the rising demands for computing calls and data communication for the emergent EC (EC) model. By stirring the functions and services positioned in the cloud to the user proximity, EC could offer robust transmission, networking, storage, and transmission capability. The resource scheduling in EC, which is crucial to the accomplishment of EC system, has gained considerable attention. This manuscript introduces a new lighting attachment algorithm based resource scheduling scheme and data integrity (LAARSS-DI) for 4G IoT environment. In this work, we introduce the LAARSS-DI technique to proficiently handle and allot resources in the 4G IoT environment. In addition, the LAARSS-DI technique mainly relies on the standard LAA where the lightning can be caused using the overall amount of charges saved in the cloud that leads to a rise in electrical intensity. Followed by, the LAARSS-DI technique designs an objective function for the reduction of cost involved in the scheduling process, particularly for 4G IoT environment. A series of experimentation analyses is made and the outcomes are inspected under several aspects. The comparison study shown the improved performance of the LAARSS-DI technology to existing approaches.

Keywords: ***Internet of Things; 4G environment; Resource scheduling; Cloud computing; Metaheuristics***

1. Introduction

With the fast improvement of the portable Internet, brilliant gadgets have turned into a basic piece of individuals' life. Progressively intricate applications like versatile installments, savvy medical services, portable games, and virtual reality (VR) put higher necessities on the resource limit of shrewd gadgets. Cloud computing (CC) was step by step acknowledged and brought into the versatile climate, which gets through the resource limits of brilliant gadgets and gives profoundly requesting applications to clients. CC is a robust model that gives plentiful applications and administrations when constructing IT the executives are more open and answer clients' requests quicker [1]. The administrations (computing, correspondence, stockpiling, and every single vital help) are conveyed and executed in a worked-on manner: on-request, no matter what the clients' area and the sort of shrewd gadgets. As a consequence of fast advances in basic advancements, the IoT is opening gigantic open doors for countless novel applications [2]. Notwithstanding, as of late, the IoT period has brought greater necessities for communication transfer speed, dormancy, power utilization, application execution, and unwavering quality. In this unique circumstance, because of the restricted data transfer capacity, high dormancy, and high energy utilization in the concentrated handling technique of CC, meeting the superior presentation necessities of users is hard. Fig. 1 represents the overview of centralized CC.

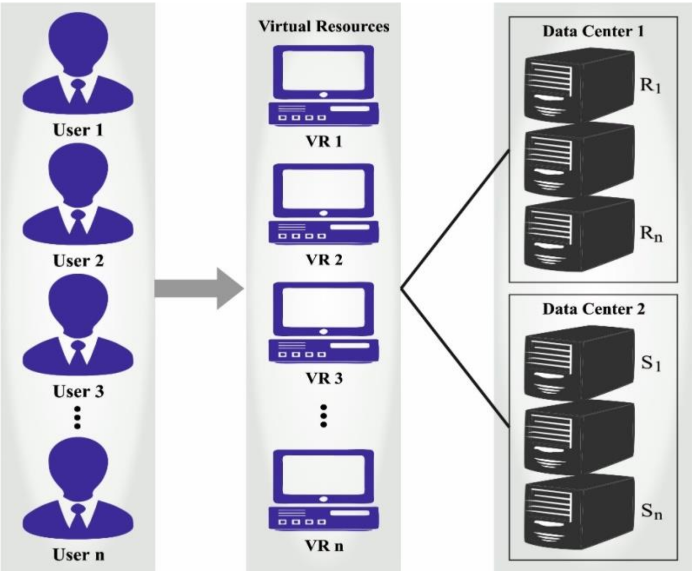

Fig. 1. *Overview of centralized CC*

Scheduling in CC is a cycle or system applied 'to limit squandering restricted resources by proficiently designating them among all dynamic hubs' [3]. Hubs or virtual machines (VMs) are the virtual resources that are relegated to buyers for running assistance and executing errands. Scheduling is an extremely mind boggling activity in CC used to distribute resources, further develop server use, improve administration execution, and execute errands. Scheduling can involve either static or dynamic strategies for scheduling resources in CC. These strategies can give adequate utilization of cloud resources to meet Quality of Service (QoS) prerequisites. Besides, utilizing scheduling procedures can keep away from clashes in assigning dynamic resources. For instance, scheduling can stay away from duplication of assigning a similar virtual resource at one time [4]. Likewise, it can assist with overseeing restricted resources by taking care of popularity of solicitations by utilizing dynamic strategy that can refresh the framework consistently and execute assignments over resources in light of the accessibility of the resources. Notwithstanding, there are a few issues that should be thought of like security, restricted resources, virtual machines, and applications. Executing and running errands over the dispensed resources raises some security gives that should be thought of as like information security, and administration security [5]. Information security incorporates security, trustworthiness, and assurance from any dangers and assaults. Administration security incorporates resource security and protection. Thus, there is a need to think about these issues and the security limitations, including information security, and accessibility to get a streamlined resource scheduling. For this exploration, the principal spotlight will be on resource scheduling components when security is figured into the cloud technique.

Scheduling of resources in the reaction stage is significant because it needs to perform different exercises like clearing individuals and moving them to a protected area on an earnest premise. To deal with calamity related salvage tasks, the ongoing need for different exercises ought to be distinguished appropriately [6]. For example, catastrophe, and different exercises like assembling the data across the debacle places for laying out correspondence networks are fundamental. On the off chance that a correspondence network is IP empowered and assuming it is feasible to associate different resources utilizing this IP empowered network, then, at that point, having data about the need of resources and whereabouts of different resources progressively is conceivable [7]. This data can be used for the dissemination and scheduling of resources for various exercises. Such IP empowered network, having resources with sensor gadgets and RFID tags, is feasible to acknowledge utilizing the IoT. IoT innovation is extremely compelling and works given Internet and can assist to an extraordinary degree with meeting the prerequisites of the post disaster the board. In this paper, we suggest calculations for such IoT based networks. As referenced above with the assistance of IoT, it is feasible to get to the data whenever, wherever, and anyplace which helps in navigation [8].

Contingent upon the kind of action and its effect, different exercises ought to be executed in an ideal and legitimate way. In such situations, it is essential that every action ought to be doled out the need. Given the need for different exercises, they ought to be given more significance while scheduling various resources. For instance, the hospitalization of the harmed individuals is having greater need contrasted with the redistribution of properties or taking care of the monetary misfortune related exercises. Further, the resources are booked so movements of every kind ought to be tended to with least sitting tight time for finishing the individual exercises [9]. For assessing the holding up season of the different exercises, it is investigated to line hypothesis. In many cases, calamity exercises could occur in a covered way regarding time. This present circumstance prompts a bunch of solicitations all the while for the overwhelming majority of resources by various exercises at different areas which brings about gridlock or race conditions. If salvage activities are not completed at right time, the circumstance might break down. From the above conversation, proficient resource scheduling for different exercises is vital, and testing

Cloud based structures were viewed as the base to current IoT empowered arrangements till as of late where endeavors are made so that cloud calculations are pushed to organize limits for which the term haze computing is laid out. Haze computing alludes to a gathering of limited scope server farms introduced as a widely appealing layer between the cloud and the things (IoT gadgets) [10]. These mist hubs are customary gadgets like wired or remote switches.

This manuscript introduces a new lighting attachment algorithm based resource scheduling scheme and data integrity (LAARSS-DI) for 4G IoT environment. In this work, we introduce the LAARSS-DI technique to proficiently handle and allot resources in the 4G IoT environment. In addition, the LAARSS-DI technique mainly relies on the standard LAA where the lightning can be caused using the overall amount of charges saved in the cloud that leads to a rise in electrical intensity. Followed by, the LAARSS-DI technique designs an objective function for the reduction of cost involved in the scheduling process, particularly for 4G IoT environment. A series of experimentation analyses are made and the results are inspected under several aspects.

2. Related Works

Xiao et al. [11] present a developing IoT infrastructure EdgeABC established BC for ensuring the data reliability of resource operation and the earnings of service providers, whereas the TO-RA system was applied to BC during the procedure of smart contracts. Afterward, the structure presented optimized the RA from IoT dependent upon the benefits of BC. Xu et al. [12] examine a distributed secure edge computing (EC) structure utilizing several data stored and BC agents to real-time context data integrity from IoT platform. The structure removes classic centralized server employing an EC structure which demonstrates CC for computer and security problems. Besides, the BC-based EC-compatible IoT plan was maintained for achieving the level of security and scalability needed for data integrity.

In [13], the authors initially expand on the privacy and security problems of EC-allowed IoT, and next exiting the main features of BCs that create BCs appropriate to edge-centric IoT developments. Moreover, the authors present a common structure for BC-related EC-allowed IoT developments which states the step-by-step process of single transaction betwixt IoT end and ECN. Wang et al. [14] presented BalancePIC, a method that tries to reserve a balancing from the 3 features (data reliability from edge-supported IoT devices, computational cost, and user privacy). It obtains the balancing with balanced truth detection method and a presented improved system to data privacy that is utilized from IoT device as well as edge server interfaces. It verifies the IoT user contribution with security from the truth detection procedure with a biometric-ECC-related validation technique.

Li et al. [15] present a new method termed as EdgeWatch for enabling robust and collaborative EDI examination from a decentralized approach dependent upon BC. Under EdgeWatch, edge servers cooperate on EDI study after a new integrity consensus. BC technique includes 3 key elements created as the structure for facilitating integrity consensus: i) an incentive process that stimulates edge servers for participating in EDI examination; ii) a reputation technique that chooses consistent leaders to block consensus, and iii) a leader randomization approach which keeps leader in targeted attack. In [16], the authors offer a trusted RA process dependent upon smart contracts, but a reputation evaluation mechanism (REM) and group-buying pricing mechanism (GBPM) were presented for efficiently addressing the issues present in resource pricing and QoS estimation of the edge servers. During the trusted RA process, end user selects a purchase method in 4 pricing systems concerning actual needs on delay and price.

3. The Proposed Model

This manuscript has introduced an effective LAARSS-DI technique for resource management in the 4G IoT environment. In this work, we introduce the LAARSS-DI technique to proficiently handle and allot resources in the 4G IoT environment. In addition, the LAARSS-DI technique mainly relies on the standard LAA where the lightning can be caused using the overall amount of charges.

3.1. Design of LAA

Lightning can be caused using the overall amount of charges saved in the cloud which leads to a rise in electrical intensity [17]. Precipitate would airstrike, and it might hit numerous times.

The four stages of addition to precipitous method are (1) dominant leader development, (2) eventual hit location, (3) destruction of air at cloud surface, and (4) successor gesture of lightning network.

Creativity. The four procedures of lightning attachment are: (1) the spread and of formation expanding leaders from the ground object, (2) air breakdown on the cloud, (3) descending migration of the precipitous network, and (4) final leap. They are briefly explained in the following.

(1) Cloud Edge Air Breakdown. The perspective among the charge centers raises as the amount of charges raises, and it is potential for deleterious charges to be discrete from the small or huge positive charge components. Consequently, power gradient nearby the cloud edge increases, and an enormous quantity of electric power (negative charge) flows towards the earth. They might create from numerous points, as demonstrated as higher-speed images of genuine lightning strikes.

(2) Efforts of Downhill Leader Headed for Earth. The precipitous approach to the ground in ongoing movement as air failure takes place nearby the cloud edge. It arrives at a stop afterward all the strides, later continue in more than one distinct direction toward the ground.

(3) Fading Branches. It is distributed to innovative divisions when there exist multiple points for the subsequent lightning. A similar method is shadowed for every novel branch, leading to the creation of novel branch.

(4) Propagation of Upward Leaders. Cloud indicates that there exists an enormous undesirable control overhead the ground. Optimistic charge clumps together on the ground or earthed items beneath the cloud.
(5) Final Leap (3.5.1.5). (Striking Point Determination). The final jump occurs when an ascending ground breaker is reached, and the signal point is the mounting leader commenced. Every branch vanishes, and the cloud charge is engrossed.

The steps involved in the LAA are given below.
Step1: Trial spots.
The trial location indicates the downward leader start point as:

$$X^i_{ts} = X^i_{\min} + \left(X^i_{\max} - X^i_{\min}\right) * rand. \tag{1}$$

The opening trial location is represented as X^i_{ts}. The control variable minimum values are $X^i_{\min}$, and maximal values are $X^i_{\min}$.$rand$ indicates an arbitrary value within zero and one in the following equation:

$$F^i_{ts} = obj\left(X^i_{ts}\right). \tag{2}$$

Step2: Determination of the following leap.
The fitness value is evaluated by averaging the original point:

$$X_{avr} = mean(X_{ts}), \tag{3}$$

$$F_{avr} = obj(X_{avr}).$$

The average point is represented as X_{avr}, and the neutral determination of average point is indicated as F_{avr}. An arbitrary result j is selected for updating point I hence $i \neq j$. The attained solution is associated with the potential solution. Consequently, it is expressed as follows:

$$X^i_{ts\_new} = X^i_{ts} + rand\ * \left(X_{avr} + X^j_{PS}\right) IFF_j < F_{avr}, \tag{4}$$

$$X^i_{ts\_new} = X^i_{ts} - rand\ * (X_{avr} + X^j_{PS}) IFF_j > F_{avr}.$$

Step3: Fading of section.
When the critical values are lesser than the electrical domain of the novel test point, the branch stays continuous; or else, it would fade, as follows.

$$X^i_{ts} = X^i_{ts\_new} IFF^i_{ts\_new} < F^i_{ts}. \tag{5}$$

In the process, test point is run, and the initial stage leftover point is pushed down.
Step4: Rising March of the leader.
This spread over the canal considerably, moving the point up in these operations.

$$S = 1 - \left(\frac{t}{t_{\max}}\right) * \ \exp\ \left(-\frac{t}{t_{\max}}\right). \tag{6}$$

If t refers to the iteration count, and $t_{\max}$ represent the maximal amount of iterations, and the following leap can be defined using the channel charge, which is shown below:

$$X^i_{ts}\text{-}new\ = X^i_{ts\_new} + rand\ * S * \left(X^i_{best} - X^i_{worst}\right), \tag{7}$$

Whereas X^i_{best} and X^i_{worst} indicates the better and worsen solutions amongst the population.
Step5: No returns of lightning.

Once the up and down leader gets together, and the striking spot is allocated, the lightning process comes to a stop.
CC was introduced and gradually accepted into the mobile environments that breakthrough source limitation of smart devices and offers extremely demanding applications for the user. CC is a cost-efficient technique providing plentiful services and applications when creating IT management highly available and responds to user demand. The services (communication, storage, computing, etc.) are implemented and delivered: on-demand, irrespective of the user position and the kind of smart device. Because of fast developments in fundamental technology, the IoT pays the way for enormous amounts of applications that are promising to increase the quality of day to day life.

3.2. Proposed Resource Scheduling Process

Consider that there exist n IoT devices and several delay-sensitive processes at UAV or IoT or cloud levels should be locally tackled [18]. U_i task generated in an IoT scheme (D_i), whereas $i = 1,2,\dots,n$, has the subsequent features: task size (U_{szi}), $(E_{i,l})$ energy consumed or CPU cycle, computation capability f_i) concerning deadline $(dl_i)$and CPU cycles or seconds.

$$T_{i.l} = \frac{U_{szi}}{f_i} \tag{8}$$

Here, the computing delay $d_{i.l}$ for device D_i is

$$d_{i.l} = ST_{i.l} + T_{i.l} \tag{9}$$

$ST_{i.l}$ characterizes the starting time of U_i for device D_i and it is given as follows:

$$E_{i.l} = U_{szi} \times U_{Ei} \tag{10}$$

$$T_{tr.i} = \frac{U_{size}}{f_{i.link}} \tag{11}$$

$f_{i.link}$ indicates the communication ability, and the execution time $T_{i.j}$ of U_i on $VMi \in ES$ as

$$T_{i.j} = \frac{U_{szi}}{f_j} \tag{12}$$

Consequently, the computation delay $d_{i.j}$ of Ui because of offloading tasks as

$$d_{i.j} = T_{t.i} + \left(ST_{i.j} + T_{i.j}\right) \tag{13}$$

Now, $ST_{i.j}$ signifies the beginning time of U_i in VMj, suppose E_t as the energy utilized for transmitting data for all the CPU cycles:

$$E_{tr.i} = E_t + U_{szi} \tag{14}$$

The computation energy $E_{i.j}$ of U_i in VMj as:

$$E_{i.j} = E_j + U_{szi} \tag{15}$$

$$tot\_E_{i.j} = E_{tr.i} + E_{i.j} \tag{16}$$

Consider that h_j as the binary predictor that specifies either a task is implemented locally as follows:

$$h_j = \begin{cases} 0 \; Local\ case \\ 1 \; Edge\ case\ or\ Cloud\ case \end{cases} \tag{17}$$

$$d_{i.j} = (1 - h_i) \times d_{i.l} + h_i \times edge\_d_{i.j} \tag{18}$$

$$E_{i.j} = (1 - h_i) \times E_{i.l} + h_i \times tot\_E_{i.j} \tag{19}$$

Furthermore, the subsequent constraint is given as:

$$d_{i.j} \leq dl_i \; and \; E_{i.j} \leq E_{\max} \tag{20}$$

In Eq. (20), $E_{\max}$ indicates the maximal energy consumed as follows:

$$y_{i.j} = \infty \times \frac{d_{i.j}}{d_{\max}} + (1 - \infty) \times \frac{E_{i.j}}{E_{\max}} \tag{21}$$

$$y_t = \sum_{vmj \epsilon Q_E} \sum_{i=1}^{n} y_{i.j} \times x_{i.j} \tag{22}$$

Here, $x_{i.j}$ indicates a binary parameter that illustrates either U_i is implemented on VMj or not. Fig. 2 exhibits the common architecture of scheduling process.

$$Minimize \; y_t \tag{23}$$

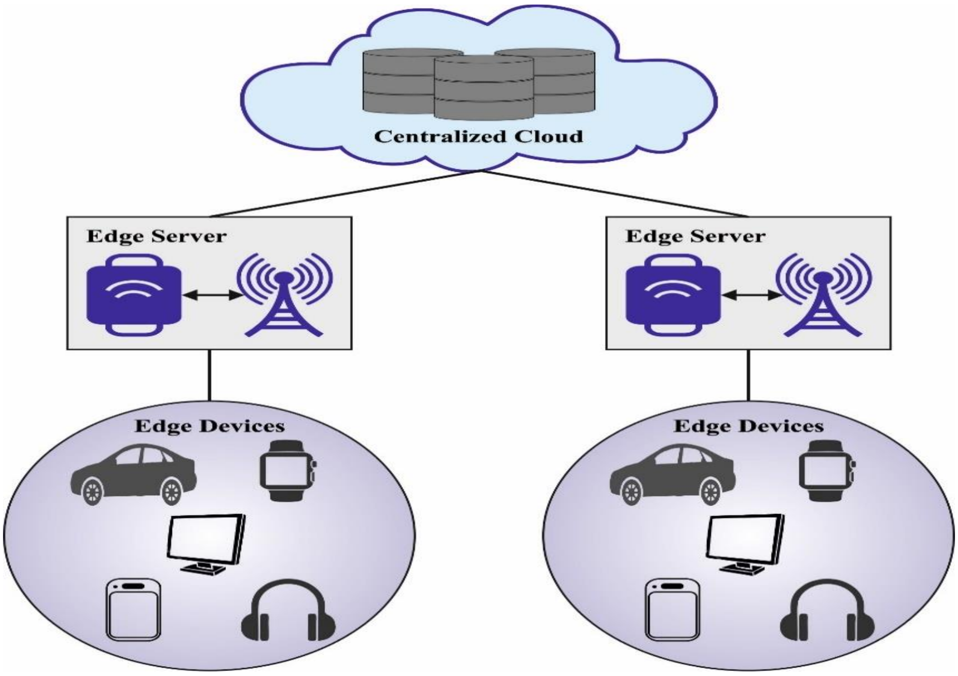

Fig. 2. *General Structure of Scheduling Process*

4. Experimental Evaluation

This section assesses the scheduling performance of the LAARSS-DI technique.

Table 1 *DEL analysis of LAARSS-DI approach with existing methods under varying tasks*

Delay (ms)				
Number of Tasks	**Local**	**Offloading**	**DEC-RASC-IoT**	**LAARSS-DI**
0	0.000	0.000	0.000	0.000
20	1.147	0.638	0.279	0.159
40	2.135	0.998	0.728	0.369

60	3.033	1.686	0.968	0.668
80	3.991	2.345	1.447	0.788
100	5.009	3.033	1.776	1.387
120	6.057	3.662	2.165	1.327
140	6.656	3.991	2.584	1.746
160	8.182	4.620	3.003	2.105
180	9.320	5.368	3.482	2.524
200	10.428	6.027	3.991	2.944

Table 1 and Fig. 3 report a DEL examination of the LAARSS-DI and existing techniques under varying tasks [18, 19]. The results inferred that the LAARSS-DI system has achieved effectual results with minimal DEL values. For example, on 20 tasks, the LAARSS-DI method has provided lower DEL of 0.159ms while the local, offloading, and DEC-RASC-IoT techniques have reached improved DEL of 1.147ms, 0.638ms, and 0.279ms correspondingly. Meanwhile, on 200 tasks, the LAARSS-DI technique has given lowest DEL of 2.524ms while the local, offloading, and DEC-RASC-IoT methods have accomplished improved DEL of 10.428ms, 5.027ms, and 3.991ms correspondingly.

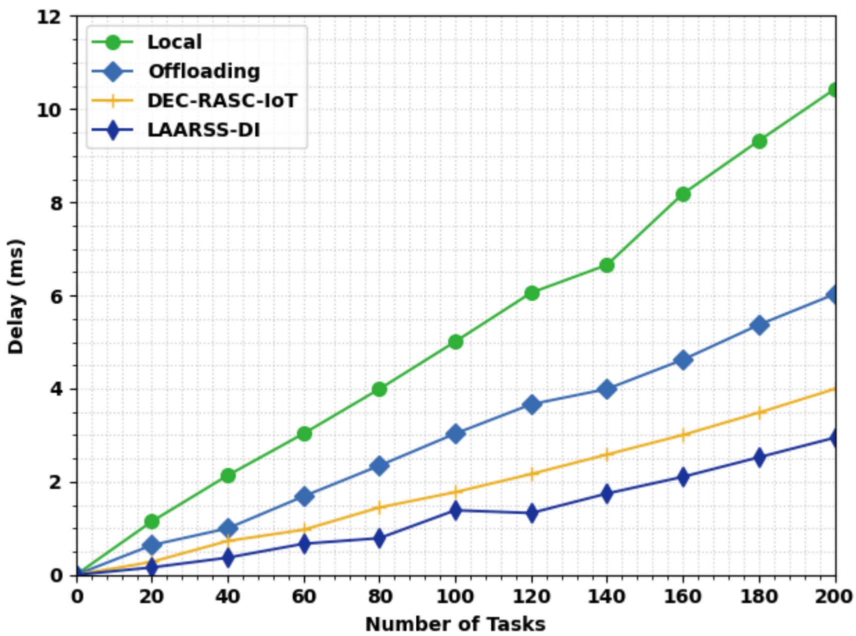

Fig. 3. *DEL analysis of LAARSS-DI approach under varying tasks*

Table 2 *ECON analysis of LAARSS-DI approach with existing methods under varying tasks*

Energy Consumption (mj)				
Number of Tasks	**Local**	**Offloading**	**DEC-RASC-IoT**	**LAARSS-DI**
0	0.00	0.00	0.00	0.00
20	16.99	6.19	2.99	1.79
40	31.80	9.39	6.19	1.39
60	49.00	14.99	8.59	2.99
80	63.01	16.99	10.99	2.99

100	79.82	23.39	16.19	6.99
120	95.42	30.20	16.19	7.39
140	112.63	35.00	22.19	6.59
160	128.63	39.80	26.20	9.79
180	149.85	45.00	29.00	10.19
200	160.00	48.20	32.60	12.19

Table 2 and Fig. 4 report an ECON examination of the LAARSS-DI and existing techniques under varying tasks. The outcomes inferred that the LAARSS-DI method has accomplished effectual results with least DEL values. For example, on 20 tasks, the LAARSS-DI method has provided lowest ECON of 1.79mj while the local, offloading, and DEC-RASC-IoT systems have accomplished improved ECON of 16.99mj, 6.19mj, and 2.99mj correspondingly. Meanwhile, on 200 tasks, the LAARSS-DI method has provided lowest ECON of 12.19mj while the local, offloading, and DEC-RASC-IoT techniques have attained increased ECON of 160mj, 48.20mj, and 32.60mj correspondingly.

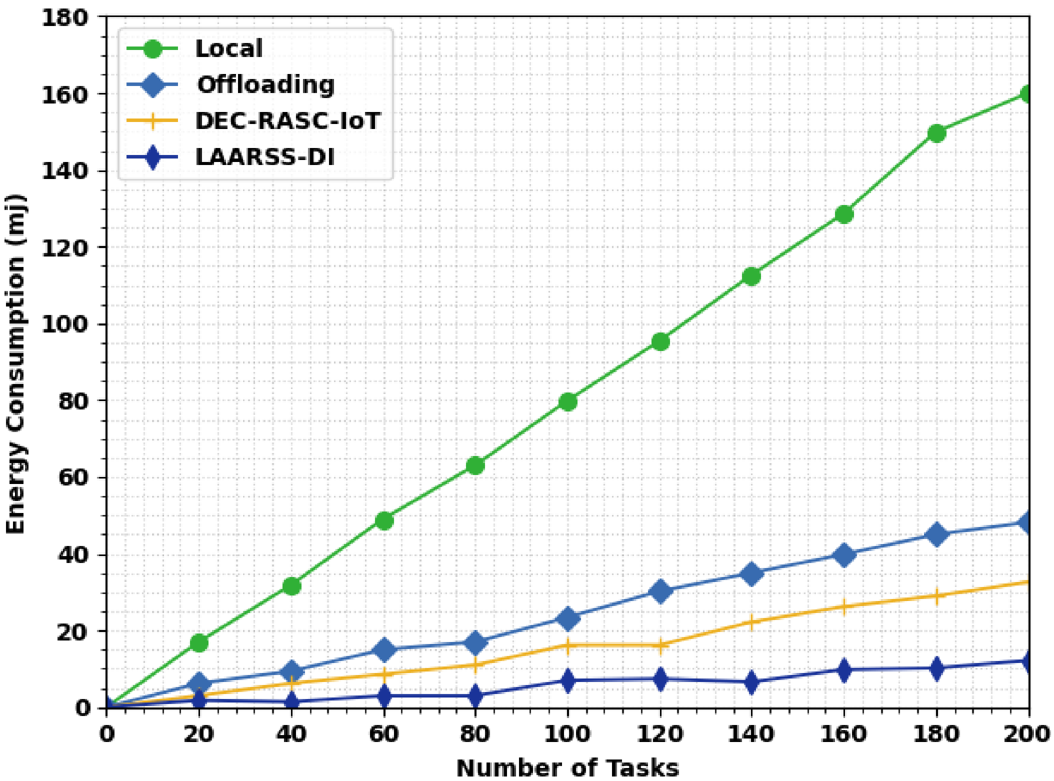

Fig. 4. *ECON analysis of LAARSS-DI approach under varying tasks*

Table 3 *DEL analysis of LAARSS-DI approach with existing methods under varying VMs*

Delay (ms)				
No. of VMs	**Local**	**Offloading**	**DEC-RASC-IoT**	**LAARSS-DI**
30	0.1425	0.0773	0.0513	0.0302
60	0.0713	0.0513	0.0279	0.0140
90	0.0472	0.0249	0.0163	0.0091

Table 3 and Fig. 5 report a DEL inspection of the LAARSS-DI and existing systems under distinct VMs. The outcomes inferred that the LAARSS-DI procedure has accomplished efficient results with minimal DEL values. For example, on 30 VMs, the LAARSS-DI method has provided lower DEL of 0.0302ms while the local, offloading, and DEC-RASC-IoT techniques have attained increased DEL of 0.1425ms, 0.0773ms, and 0.0513ms correspondingly. Meanwhile, on 90 VMs, the LAARSS-DI method has provided lowest DEL of 0.0091ms while the local, offloading, and DEC-RASC-IoT techniques have accomplished improved DEL of 0.0472ms, 0.0249ms, and 0.0163ms respectively.

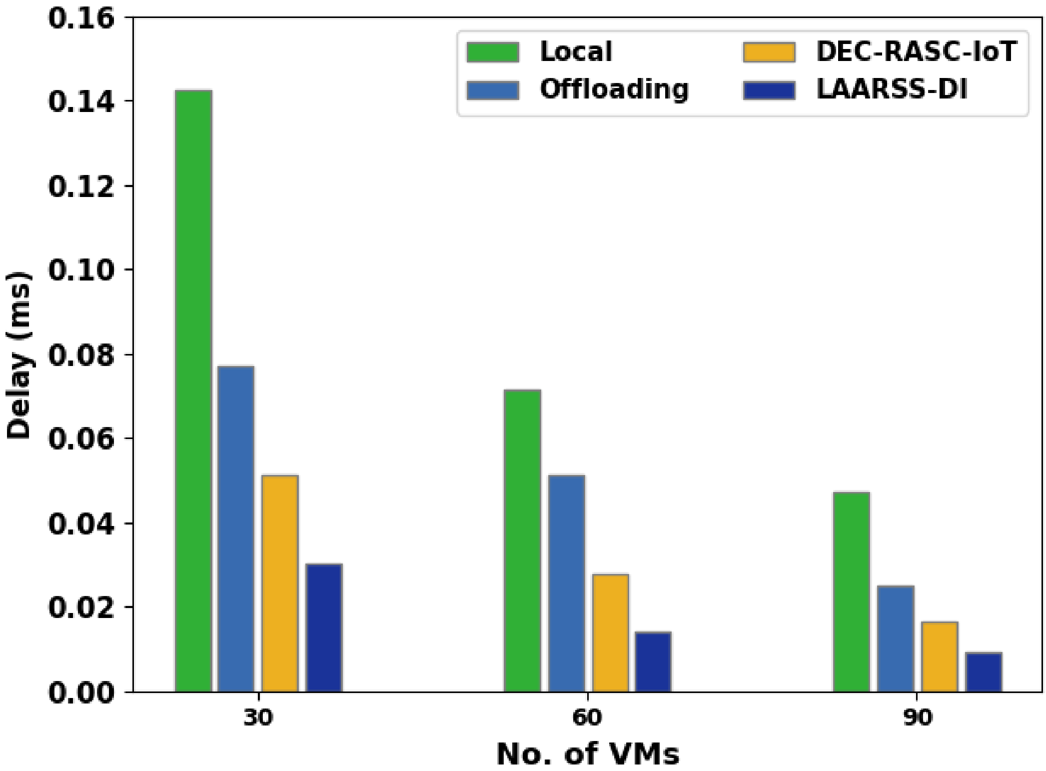

Fig. 5. *DEL analysis of LAARSS-DI approach under varying VMs*

Table 4 *ECON analysis of LAARSS-DI approach with existing methods under varying VMs*

Energy Consumption (mj)				
No. of VMs	**Local**	**Offloading**	**DEC-RASC-IoT**	**LAARSS-DI**
30	1968.86	695.84	362.43	119.95
60	3939.00	2999.39	1984.01	423.05
90	5727.28	4636.13	3332.80	483.67

Table 4 and Fig. 6 report an ECON inspection of the LAARSS-DI and existing methods under varying VMs. The outcomes inferred that the LAARSS-DI system has accomplished effectual results with minimal DEL values. For example, on 30 VMs, the LAARSS-DI method has provided lowest ECON of 119.95mj while the local, offloading, and DEC-RASC-IoT techniques have accomplished improved ECON of 1968.86mj, 695.84mj, and 362.43mj correspondingly. Meanwhile, on 90 VMs, the LAARSS-DI method has provided lower ECON of 483.67mj while the local, offloading, and DEC-RASC-IoT systems have achieved improved ECON of 5727.28mj, 4636.13mj, and 3332.80mj correspondingly.

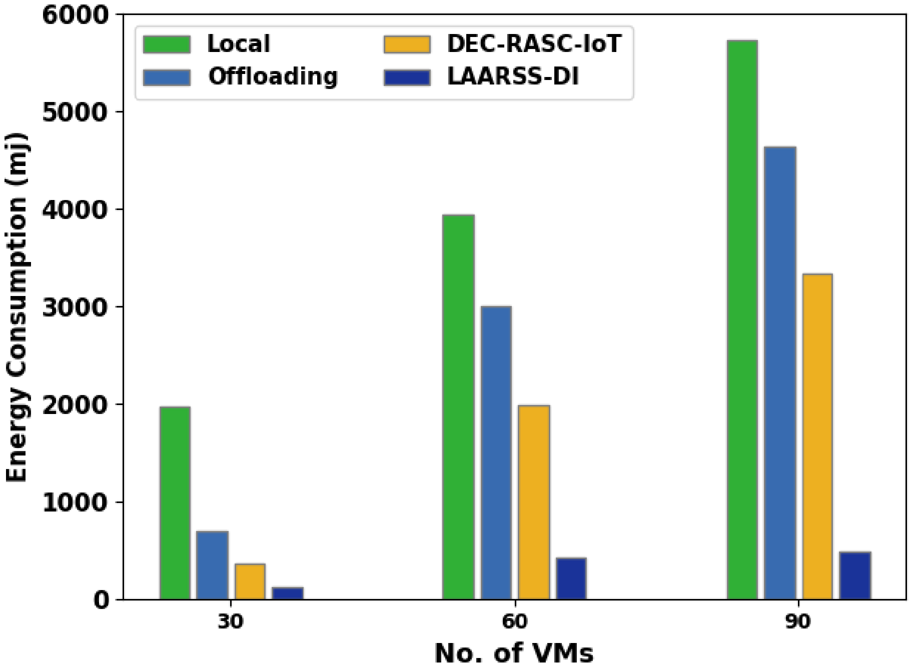

Fig. 6. *ECON analysis of LAARSS-DI approach under varying VMs*

5. Conclusion

This manuscript has introduced an effective LAARSS-DI technology for resource management in the 4G IoT environment. In this work, we introduce the LAARSS-DI technique to proficiently handle and allot resources in the 4G IoT environment. In addition, the LAARSS-DI technique mainly relies on the standard LAA where the lightning can be caused using the overall amount of charges saved in the cloud that leads to a rise in electrical intensity. Followed by, the LAARSS-DI technique designs an objective function for the reduction of cost involved in the scheduling process, particularly for 4G IoT environment. A series of experimentation analyses are made and the outcomes are examined under several aspects. The comparison study shows the improved performance of LAARSS-DI technology compared to existing approaches.

References

[1] Lu, N., Zhang, Y., Shi, W., Kumari, S. and Choo, K.K.R., 2020. A secure and scalable data integrity auditing scheme based on hyperledger fabric. *Computers & Security*, *92*, p.101741.

[2] Wang, H., Zhang, J., Lin, Y. and Huang, H., 2021, May. ZSS signature based data integrity verification for mobile edge computing. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)* (pp. 356-365). IEEE.

[3] Cheng, Y., Zhu, C. and Peng, F., 2020, October. A Resource Allocation Algorithm in Energy Harvesting Aided Cellular IoT With Full-Duplex Backhauls. In *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)* (pp. 867-871). IEEE.

[4] Narayanan, E. and Muthukumar, B., 2022. A machine learning framework for providing data integrity and confidentiality for sensitive data cloud applications. *International Journal of System Assurance Engineering and Management*, pp.1-12.

[5] Li, C., Sun, H., Tang, H. and Luo, Y., 2019. Adaptive resource allocation based on the billing granularity in edge-cloud architecture. *Computer Communications*, *145*, pp.29-42.

[6] Khalid, A., ul Ain, Q., Qasim, A. and Aziz, Z., 2021. QoS based optimal resource allocation and workload balancing for fog enabled IoT. *Open Computer Science*, *11*(1), pp.262-274.

[7] Zhang, X. and Li, W., 2020, December. Research on data integrity protection technology of industrial control network. In *2020 5th international conference on mechanical, control and computer engineering (ICMCCE)* (pp. 755-758). IEEE.

[8] Hang, L. and Kim, D.H., 2019. Design and implementation of an integrated iot blockchain platform for sensing data integrity. *Sensors*, *19*(10), p.2228.

[9] Gangadevi, K. and Devi, R.R., 2021, March. A survey on data integrity verification schemes using blockchain technology in Cloud Computing Environment. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1110, No. 1, p. 012011). IOP Publishing.

[10] Li, B., He, Q., Chen, F., Jin, H., Xiang, Y. and Yang, Y., 2020. Auditing cache data integrity in the edge computing environment. *IEEE Transactions on Parallel and Distributed Systems*, *32*(5), pp.1210-1223.

[11] Xiao, K., Gao, Z., Shi, W., Qiu, X., Yang, Y. and Rui, L., 2020. EdgeABC: An architecture for task offloading and resource allocation in the Internet of Things. *Future Generation Computer Systems*, *107*, pp.498-508.

[12] Xu, R., Hang, L., Jin, W. and Kim, D., 2021, August. Distributed Secure Edge Computing Architecture Based on Blockchain for Real-Time Data Integrity in IoT Environments. In *Actuators* (Vol. 10, No. 8, p. 197). MDPI.

[13] He, Y., Wang, Y., Qiu, C., Lin, Q., Li, J. and Ming, Z., 2020. Blockchain-based edge computing resource allocation in IoT: a deep reinforcement learning approach. *IEEE Internet of Things Journal*, *8*(4), pp.2226-2237.

[14] Wang, T., Bhuiyan, M.Z.A., Wang, G., Qi, L., Wu, J. and Hayajneh, T., 2019. Preserving balance between privacy and data integrity in edge-assisted Internet of Things. *IEEE Internet of Things Journal*, *7*(4), pp.2679-2689.

[15] Li, B., He, Q., Yuan, L., Chen, F., Lyu, L. and Yang, Y., 2022, August. EdgeWatch: Collaborative investigation of data integrity at the edge based on blockchain. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 3208-3218).

[16] Cheng, H., Hu, Q., Zhang, X., Yu, Z., Yang, Y. and Xiong, N., 2021. Trusted Resource Allocation Based on Smart Contracts for Blockchain-Enabled Internet of Things. *IEEE Internet of Things Journal*, *9*(11), pp.7904-7915.

[17] Sun, S., Ye, Z., Yan, L., Su, J. and Wang, R., 2018, September. Wrapper feature selection based on lightning attachment procedure optimization and support vector machine for intrusion detection. In *2018 IEEE 4th international symposium on wireless systems within the international conferences on intelligent data acquisition and advanced computing systems (IDAACS-SWS)* (pp. 41-46). IEEE.

[18] Mahmood, O.A., Abdellah, A.R., Muthanna, A. and Koucheryavy, A., 2022. Distributed Edge Computing for Resource Allocation in Smart Cities Based on the IoT. *Information*, *13*(7), p.328.

[19] Kim, S., 2019. Novel resource allocation algorithms for the social internet of things based fog computing paradigm. *Wireless Communications and Mobile Computing*, *2019*.