Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 1, 2023, 149-159.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

# A New Adaptive Approach for Efficient Energy Consumption in Edge Computing

Hossein Morshedlou* and Alireza Tajari

*Department of Computer Engineering and Information Technology, Shahrood University of Technology, Shahrood, Iran.*

**Article Info**

**Article History:**
*Received 25 June, 2022*
*Revised 29 August 2022*
*Accepted 08 January 2023*

*DOI:10.22044/jadm.2023.12005.2348*

**Keywords:**
*Edge Computing, Energy Efficiency, ICLA, Compatible Point (CP).*

**Corresponding author morshedlou@shahroodut.ac.ir (H. Morshedlou).*

**Abstract**

Edge computing is an evolving approach for the growing computing and networking demands from end devices and smart things. Edge computing lets the computation to be offloaded from the cloud data centers to the network edge for lower latency, security, and privacy preservation. Although energy efficiency in cloud data centers has been widely studied, energy efficiency in edge computing has been left uninvestigated. In this work, a new adaptive and decentralized approach is proposed for more energy efficiency in edge environments. In the proposed approach, edge servers collaborate with each other to achieve an efficient plan. The proposed approach is adaptive, and considers workload status in local, neighboring, and global areas. The results of the conducted experiments show that the proposed approach can improve energy efficiency at network edges; e.g. by task completion rate of 100%, the proposed approach decreases energy consumption of edge servers from 1053 Kwh to 902 Kwh.

## 1. Introduction

DATA centers are thought to use 200 terawatt hours (TWh) of energy annually, which is more than some nations need. Additionally, they are thought to be responsible for 2% of global CO2 emissions [1]. Edge computing as an evolving approach leads to a greater guarantee of fairness between edge resources. It has the potential to significantly reduce the amount of information moving through the community, freeing up bandwidth. In edge computing, servers are set up at the network's edge so that computation can take place close to the data sources. This has two benefits: edge servers function as cloud service providers for downstream data, putting resources close to end users to reduce service request latency; for upstream data, it helps to enhance network transmission. The positioning of edge servers is vital and crucial since it is the initial stage in the implementation of an edge computing architecture. To reduce request delays, it is ideal to deploy as many edge servers as possible. However, doing so would result in significant power consumption. As a result, there will surely be a significant overall energy usage. Additionally, due to its limited scale and fragmented deployment, it cannot benefit from the high equipment efficiency that comes with scale advantages. Therefore, problem of energy efficiency in edge computing will be a great challenge. In cloud data centers, where there are large volume of requests from different places, wasting time is very low. However, in edge only local requests are processed and at non-peak intervals local resources may waste energy. To decrease power consumption, some edge servers can be switched to sleep mode. But decision-making about switching should be done based on workload status of a server and its neighbor servers at adjacent locations. Irregular Cellular Learning Automata (ICLA) is a distributed learning tool in which a number of learning automata are located in a graph-like structure and cooperate with each other. Each learning automata learns by applying different actions to environment and getting their responses. They consider states of all neighbor learning automata in choosing their actions. Learning automata and ICLA are employed in a wide range of

applications in different areas such as wireless sensor networks [2, 3], social networks [4, 5], and cellular channel assignment [6]. The obtained results show capabilities of ICLA in distributed decision-making problems. Due to the capabilities of ICLA in distributed and decentralized learning problems, we decided to use it for the problem of energy-efficient resource management in edge environments for the first time. The proposed approach is self-adaptive, and changes in environment trigger learning automata to change ICLA behavior. Moreover, workload status in local, neighboring, and global environments are used in our work as an innovation for better handling of energy consumption. Considering local, neighboring, and global load status results in better decisions. Because of complex interactions among edge devices, servers, and cloud data centers, issue of energy efficiency and power consumption in edge servers is not sufficiently addressed. The topic of energy efficiency among edge servers is the main emphasis of this work. To save energy, each edge server in our work has two modes: ON and Standby. At the non-peak time, a node can switch to Standby mode to decrease energy consumption. However, decision-making about switching to standby mode is not simple. A wrong decision may lead to failure in completing offloaded tasks. When an edge server switches to standby mode, its tasks should be accomplished by its neighbors. Therefore, it is necessary to involve status of neighbor nodes in decision-making. Decisions are about putting a server in ON/STANDBY mode. The proposed approach is also adaptive, and any changes in workload may cause the resources switch to another mode. The results of the conducted experiments show that the proposed approach can improve energy efficiency in edge environments. The remainder of the paper is organized as what follows. In section 2, related works are reviewed. Section 3 introduces some preliminary concepts that are used in next sections of the paper. Section 4 presents the proposed approach, and section 5 contains the details of the conducted experiments including platform, data set, simulation details, and the obtained results. Section 6 concludes the paper.

## 2. Related Works

In cloud computing, the placement problem has received extensive attention. For example, a large-scale wireless metropolitan area network, [7] explores the cloudlet placement challenge. They show that the problem is NP-hard, and suggest a fast scalable heuristic solution. Work in [8] attempts to maximize the trade-off between deployment expense and end-to-end delay in their work. They suggest an algorithm to select strategic points in order to decrease the end-to-end delay and reduce the number of edge servers. Mobile edge computing has distinct limitations than cloudlets. The placement of virtual machines (VMs) can be optimized to save energy [9], and this is a common topic in cloud computing. For instance, [10] uses algorithms based on mixed integer programming to address complex VM placement issues. To determine the physical machine energy consumption, they develop a non-linear power consumption model. Work in [11] proposes an energy efficient independent task scheduler using supervised neural networks with the aim to reduce energy consumption, execution overhead, and number of active servers. In [12], problem of energy-aware edge server placement is studied and tried to find a more effective placement scheme with low energy consumption. The problem is formulated as a multi-objective optimization problem, and a particle swarm optimization-based energy-aware edge server placement algorithm is devised to find the optimal solution. Evaluation of the algorithm illustrates that the algorithm can reduce more than 10% energy consumption with over 15% improvement in computing resource utilization, compared to other rival approaches.

There are many issues that effect on consumed energy of a data center, data center operations, and computational load management [13]. The existing works for improvement of energy efficiency concentrate on workload allocation so that the requirements of the application can be met with a minimum number of machines. Task allocation, service migration, and energy scheduling issues to decrease energy consumption are considered in Gu et al. [14]. They propose an approach that formulates virtual machines (VM) migration and power planning issues with the goal of minimizing power costs. Next, to solve this problem, a low-complexity algorithm is designed. A collaborative cloud and edge data analytics platform is proposed by Stefan et al. [15] that extends the idea of server-less computing to the network edge. Delay-sensitive jobs are processed at the platform's edge for real-time response, while tasks that need a lot of computing power are transferred to the cloud for processing and storing. Understanding and improving energy efficiency becomes difficult [16-19], and there are two common ways to fix the issue: either by consolidating the workload or by shutting down the extra servers. There have been several relevant

studies in the recent years to improve energy efficiency in data centers using machine learning-based methods [11,20-22]. Numerous factors including power delivery, heat generated by data center operations, and associated cooling costs, and computational load control, all have an impact on the energy usage of data centers [13]. A comprehensive overview of works regarding the energy-aware Edge service and applications in literature is included in Table 1.

## 3. Preliminary Concepts

In this section, we present some preliminary concepts. For details about learning automata, ICLA and its applications, we refer the readers to [23].

**Neighbor Resource:** Resource A is neighbor of B, if it is possible to delegate tasks of resource A to resource B and vice versa. For example, A can decide to switch to standby mode and B handle the tasks in buffers of A. We used fixed neighboring for edge nodes. E.g., Figure 2 shows the neighboring relations by edges between nodes.

**Neighbor-based Agreement (NBA)**: It is an agreement over a period of time that determines states of a resource and its neighbors. Here, set of states is {ON, Standby}*. Energy usage of a resource in Standby state is very low, and according to agreement, other neighbors that are in ON mode offer service to tasks of the resource in standby mode.

**Compatible Point (CP):** In ICLA structure, it is equivalent to Nash Equilibrium (NE) point, if all nodes in an ICLA to be neighbors. However, when neighboring relation is defined just between some nodes, CP is used instead of NE. In CP, each node cares about the actions of its neighbors only. The selected action of a non-neighbor node has indirect effect on utility of a node.

**Notations:** some notations that are used in the following sections of the paper, are listed below:

$ME_G$: Maximum possible energy consumed by all resources on the edge over a predefined interval T.

$ME_{N_i}$: Maximum possible energy consumed by resource I and its neighbors over a predefined interval T.

$ME_{L_i}$: Maximum possible energy consumed by resource i over a predefined interval T.

$TCR_{GE}$: Task Completion Rate is percentage of completed tasks by all resources in the edge over T.

$TCR_{NE_i}$: Percentage of completed tasks by neighbors of resource i over T.

$TCR_{LE_i}$: Percentage of completed tasks by resource i over T.

$CE_G(t)$: Total consumed energy by all resources on edge over interval t

$CE_{L_i}(t)$: Total consumed energy by resource i over interval t

$CE_{N_i}(t)$: Total consumed energy by neighbors of resource i over interval

## 4. Proposed Approach

In the proposed approach, a learning automaton with two actions (ON/Standby) is assigned to each node that decides about the resource modes. Neighboring relations are also defined between nodes according to resources with similar capabilities. The defined neighboring relations forms the ICLA structure. Figure 1 shows mapping of edge nodes neighboring relation to an ICLA.

**Table 1. Literature works on energy aware edge services and applications.**

| Contributions | | | Description |
|---|---|---|---|
| Energy-aware services and applications | Application specific [15, 24-27] | | Application-specific approaches |
| | Services placement | Services placement [28] | Provide an optimal mapping between IoT applications and computing resources |
| | | Data placement [14,29] | Provide an optimal data placement strategy with minimal cost |
| | Energy-aware service and application based on machining learning | Methods for data center [11,20,21] | Forecasting, consolidating the resource and shutdown the servers by putting spare servers into sleep |
| | | Methods for edge computing [22, 30-55] | Enabling the machine learning applications available at the edge |

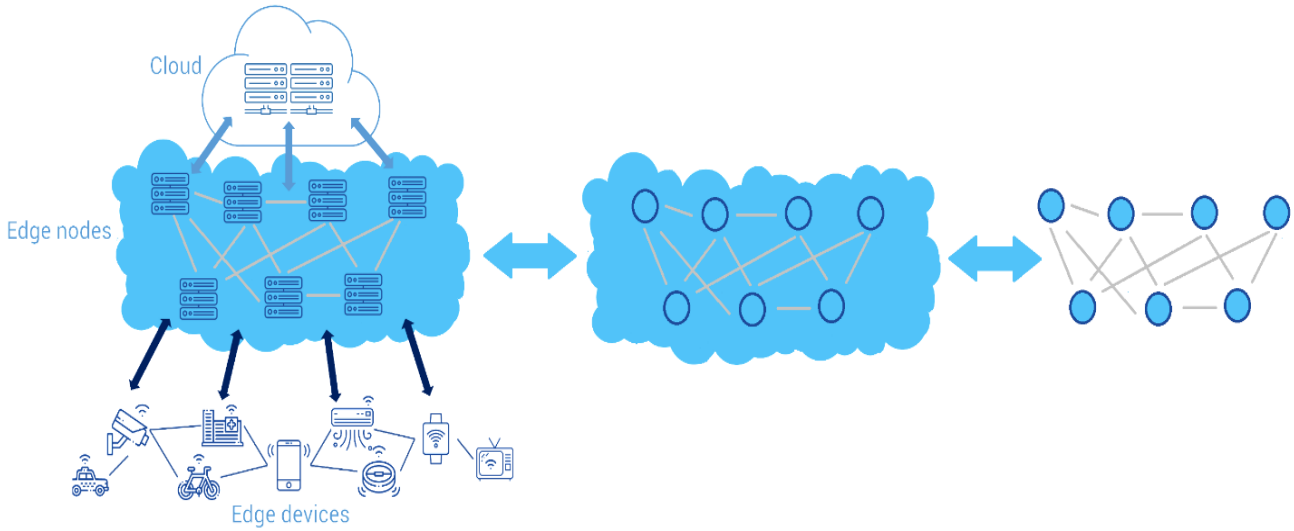



**Figure 1. Mapping of Edge nodes neighboring relation to ICLA structure.**

A learning automaton on resource i receives information about traffic load (buffer status) of resource i, its neighbours, and total traffic load of whole nodes in the edge, as illustrated in Figure 2.

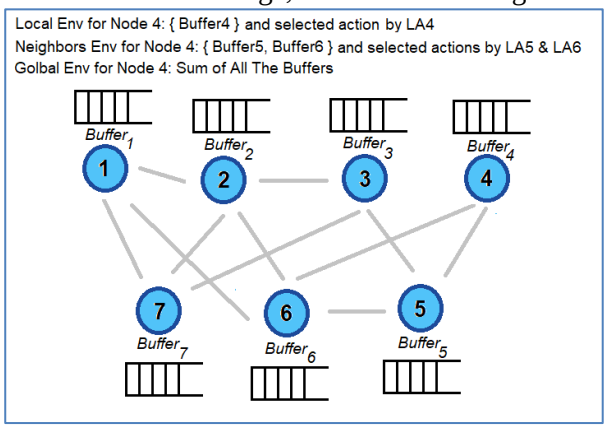



**Figure 2. Local, neighbors, and global enviroments for edge nodes.**

In our work, the available capacities of an edge node and user task requirements are represented by vector <cpu, memory>. The vectors are unitized for simplicity. Assume the capacity needs of the tasks in the buffers to be [Buffer1: {<2,3>, <5,4>}, Buffer2: {<1,1>, <3,1>, <5,2>, <2,7>}, Buffer3:{<6,3>, <7,1>, <2,5>}, Buffer4:{<1,1>, <3,2>, <2,3>, <5,4>, <3,3>, <3,2>}, Buffer5:{<4,6>}, Buffer6:{<1,7>, <2,3>, <5,4>}, Buffer7:{<3,2>, <7,5>}]. Therefore, Global Env is (21, 72, 69) that means there are 21 tasks in system with aggregate capacity needs of <72,69>. As shown in Figure 3, using this information and selected actions of neighbours, Reinforcement Rule of each LA generate reward signal for LA. LA updates its probability vector according to the reward signal.

**4.1. Reinforcement Rule Algorithm:** Current status of task buffers in edge network define a state for resource i in our approach. For example, a state for node i includes information about ($Buffer_G(t)$, $Buffer_{N_i}(t)$, $Buffer_{L_i}(t)$), where $Buffer_G(t)$ is Global Env. $Buffer_{N_i}(t)$ is buffer status of neighbour nodes (Neighbours Env), and $Buffer_{L_i}(t)$ shows status of node i 's buffer (Local Env). Learning automaton of each node learns a Q-values for each state separately and make decision for switching to ON/Standby modes using the learned Q-Values. The learning of q-values is done using (1) where $a^1$, …, $a^{\overline{m}_i}$ are the selected actions (ON/Standby) by resource i and its neighbours.

$$Q^i_{t+1}(a^1,...,a^{\overline{m}}) = (1-\alpha)\times Q^i_t(a^1,...,a^{\overline{m}}) + \alpha\times[Er^t_i(a^1,...,a^{\overline{m}}) + \beta.\sum_{a^1}\sum_{a^2}...\sum_{a^{\overline{m}}} \underline{p}^{br}_i(\underline{\hat{p}}^t, Q^i_t)(a^i)\prod_{\substack{j=1\\ j\neq i}}^{\overline{m}} \underline{\hat{p}}^t_j(a^j).Q^i_t(a^1,...,a^{\overline{m}})] \quad (1)$$

The local rule of ICLA to generate reward signals is presented below. $\underline{p}^{br}_i$ is the best response of LA to the selected actions of its neighbours (here, the concept of best response is equivalent to the concept of best response in the game theory). Using this local rule for generating reward signals and updating Q-values using (1), we formally proofed that ICLA will converge to a CP point.

$Er^t_i(a^1,...,a^{\overline{m}})$ in (1) should be defined based on context of application.

For state ($Buffer_G(t)$, $Buffer_{N_i}(t)$, $Buffer_{L_i}(t)$), we have:

$$Er_i^t(a^1,...,a^{\overline{m}}) = \left[(t-1)\times Er_i^{t-1}(a^1,...,a^{\overline{m}}) + r_i^t\right]/t$$

Here, $r_i^t$ is defined as (2):

$$r_i^t = w_{Energy}\times[w_G\times\frac{CE_G(t)}{ME_G} + w_L\times\sum_{i=0}^{n}\frac{CE_{L_i}(t)}{ME_{L_i}(t)} + w_N\times\sum_{i=0}^{n}\frac{CE_{NE_i}(t)}{ME_{N_i}(t)}]^{-1} + w_{TCR}\times[v_G\times TCR_{GE} + v_N\times\sum_{i=1}^{n}TCR_{NE_i} + v_L\times\sum_{i=1}^{n}TCR_{LE_i}] \quad (2)$$

In (2), $w_{Energy}$ determines weight of consumed energy in edge against percentage of completed tasks in edge ($w_{TCR}$). The weights $w_G$, $w_L$ and $w_N$ define importance of total consumed energy in edge resources, resource i and the neighbours of resource i, respectively.

## 5. Experiments and Results

In this section, we have conducted some experiments to evaluate the proposed approach. Specification of platform used for running the experiments is Windows 10 + Core i7-8559U CPU with 4 cores, Turbo-Boost disabled at a base clock of 2.70 GHz and with 16 GB memory. Implementation tool is EdgeCloudSim. For simulation of energy-aware edge server, PowerDatacenter class is used. Load Generator Module of EdgeCloudSim is employed for generating the user tasks. Each user device generates tasks by following a given distribution. Each user device has a task type, e.g. health app and generate tasks according to the given type. Convergence of ICLA is not restricted to a specific distribution, and it is capable to learn and handle repeatable patterns with different distributions [23]. However, the times at which tasks are generated are defined by a Poisson process.

$\underline{p}_i^c$ = current probability vector of $LA_i$

$\underline{p}_i^{br}$ = best response probability vector accuired from (1)

$d_i$ = vector of reward probabilities

*Begin*

$$\underline{p}_i^{br}(\underline{\hat{p}}^t, Q_t^i) = \arg\max_{\underline{p}_i^{br}}\sum_{a^1}\sum_{a^2}...\sum_{a^{\overline{m}}}\underline{p}_i^{br}(\underline{\hat{p}}^t, Q_t^i)(a^i)\prod_{\substack{j=1\\ j\neq i}}^{\overline{m}}\underline{\hat{p}}_j^t(a^j).Q_t^i(a^1,...,a^{\overline{m}});$$

$$\Delta\underline{p}_i^{RS} = \underline{p}_i^{br} - \underline{p}_i^c;$$

$$j_{\max} = \arg\max_j \Delta\underline{p}_i^{RS}(j);$$

$$j_{\min} = \arg\min_j \Delta\underline{p}_i^{RS}(j);$$

$$d_i = \left(\frac{\Delta\underline{p}_i^{RS}(1) - \Delta\underline{p}_i^{RS}(j_{\min})}{\Delta\underline{p}_i^{RS}(j_{\max}) - \Delta\underline{p}_i^{RS}(j_{\min})},...,\frac{\Delta\underline{p}_i^{RS}(\overline{m}_i) - \Delta\underline{p}_i^{RS}(j_{\min})}{\Delta\underline{p}_i^{RS}(j_{\max}) - \Delta\underline{p}_i^{RS}(j_{\min})}\right)$$

if (selectedaction $== a^k \in \{a^1,...,a^{\overline{m}_i}\})$ then

set $\beta = 1$ with probability $d_i(k)$ and $\beta = 0$ with probability $(1 - d_i(k))$;

update Q - values using (1);

return $\beta$;

End

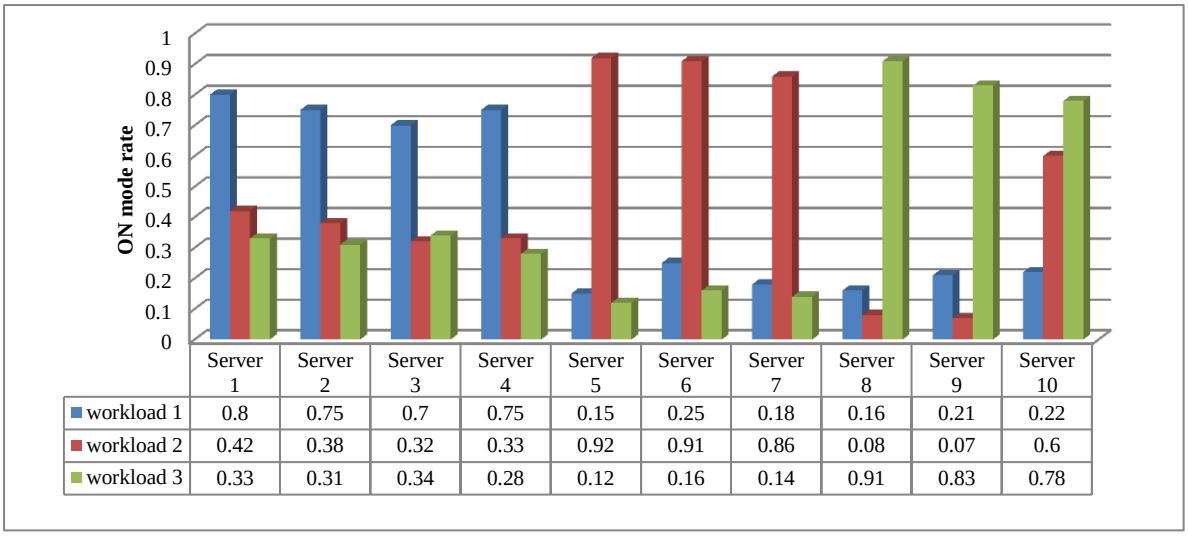
**Figure 3. Local rule of ICLA.**

Each task type has a certain expected value. The time interval in which two tasks of a device are generated follows a random process by the exponential distribution with this expected value. The number of edge servers and user devices are 10 and 20, respectively. The triplex of (Poisson parameter, CPU capacity needs, memory capacity needs) for task type 1, 2 and 3 are (1,5,5), (1,8,2) and (1,2,8) respectively. For simplicity, all 20 user devices during a workload generate same task types in experiments. Thus in workload 1, workload 2, and workload 3, we have task type 1,

task type 2, and task type 3, respectively. We have used location information of servers and users from EUA dataset (10 servers and 20 users). It contains datasets of edge server locations and datasets of user location. The locations of edge servers (base stations) are in the Melbourne central business district area in Australia. Following the Gaussian Distribution N(u, σ), users are distributed in different ways in this area to simulate six different real-world TD-EUA scenarios with different user distributions. In our work, each edge server has a pre-determined capacity, and in the ON mode can serve sent tasks by user apps according to the required capacity of them. Table 2 illustrates energy consumption, CPU, and memory capacity of edge servers in the conducted experiments. However, an edge server may buffer the received tasks and fetch them from buffer for processing. Some tasks will not be completed if the required capacity of the buffered tasks exceed the server capacity. When a server switches to StandBy mode, tasks in its buffer are moved to buffers of its neighbours. At first experiment, fixed workloads (Workload 1~3) over 2000 iterations are used to evaluate ICLA behaviour. Charts of Figure 4 illustrate the rate of being in the ON mode for edge servers during various workloads. As shown in this figure, over workload 1, servers 1~4 are mostly on ON mode (80%, 75%, 70%, and 75% of the time). Task type in workload 1 is task type 1 that needs moderate CPU and memory requirements. For such tasks, servers 1~4 are the best choices. Similarly, over workload 2 and 3, servers 5~7, and servers 8~10 are mostly on the ON mode. This means that ICLA has the capability to converge to an appropriate point according to task requirements.

When all learning automata converge to one of their actions, this means that ICLA has converged to a CP point. Now to evaluate the CP point, we change the action that a LA has converged to it unilaterally without any changes in actions of its neighbours. When ICLA converges during Workload1, total energy consumption is 902 Kwh with task completion rate of 100%. Table 3 illustrates the total energy consumption and task completion rates when each edge server unilaterally deviates from strategy determined by the reached CP point. For example, when node 1 changes its action from ON to StandBy, total energy consumption by all edge servers decreases from 902 Kwh to 823 Kwh but task completion rate falls down to 0.92 (92%). For node 5, when it switches from the converged action StandBy to the ON action, the energy consumption increases from 902 Kwh to 1053 Kwh and task completion rate remain 1 (100%). The obtained results of Table 3 show that task completion rate of 100% is accessible only with higher energy consumption (1053 and 1217 Kwh when node 5 and node 10 unilaterally deviate from CP point in comparison to energy consumption in CP point which is 902 Kwh with task completion rate of 100%). This means that unilaterally deviation is not profitable and CP point offer a better energy consumption with task completion rate of 100%. Figure 5 compares the obtained reward by each edge server when strategy of CP point is followed by all servers versus condition a node unilaterally deviates from the CP point strategy. As illustrated in this figure, unilateral deviation is not profitable, and this means that it is better for each server to follow proposed strategy of CP point to earn maximum possible Reward.

Now to evaluate our approach against the existing works in the literature, we selected one ANN-based approach [11] and one PSO-based approach [12] to compare. For details of these approaches, please see Section 3. The evaluation criteria used for comparison are total energy consumption and TCR over all nodes of the edge. Figure 6 shows the total energy consumed by edge servers during 1000 iterations. TCR over these iterations are illustrates in figure 7. Over initial iterations, ANN-based approach has a faster learning rate, and shows a better performance than the proposed approach but after iteration 200, the proposed approach outperforms it. This illustrates that ANN-based approach is better than ICLA in short term but considering effect of workloads in neighbouring areas leads to reaching better results by ICLA in long term. PSO-based approach has a poor performance in comparison to both ANN and ICLA. This is due to heuristic strategy of this approach. Sometimes, requests with high CPU requirement and low memory needs are assigned to servers with low CPU capacity and high memory capacity. This causes a higher energy consumption and a lower TCR for PSO-based approach.




| | Server 1 | Server 2 | Server 3 | Server 4 | Server 5 | Server 6 | Server 7 | Server 8 | Server 9 | Server 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| workload 1 | 0.8 | 0.75 | 0.7 | 0.75 | 0.15 | 0.25 | 0.18 | 0.16 | 0.21 | 0.22 |
| workload 2 | 0.42 | 0.38 | 0.32 | 0.33 | 0.92 | 0.91 | 0.86 | 0.08 | 0.07 | 0.6 |
| workload 3 | 0.33 | 0.31 | 0.34 | 0.28 | 0.12 | 0.16 | 0.14 | 0.91 | 0.83 | 0.78 |

**Figure 4. Rate of being in ON mode for edge servers during various workloads.**

**Table 2. Energy consumption, CPU, and memory capacity of edge servers used in experiments**

| Edge server | Server 1 | Server 2 | Server 3 | Server 4 | Server 5 | Server 6 | Server 7 | Server 8 | Server 9 | Server10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CPU capacity | 100 | 100 | 100 | 100 | 150 | 150 | 150 | 50 | 50 | 50 |
| Memory capacity | 100 | 100 | 100 | 100 | 50 | 50 | 50 | 150 | 150 | 150 |
| Energy consumption (ON mode) | 100 | 100 | 100 | 100 | 120 | 120 | 120 | 80 | 80 | 80 |

**Table 3. Total energy consumption and task completion rates when each edge server unilaterally deviates from strategy determined by the reached CP point.**

| Node | Node1 | Node2 | Node3 | Node4 | Node5 | Node6 | Node7 | Node8 | Node9 | Node10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Energy | 823 | 824 | 815 | 818 | 1053 | 758 | 754 | 1120 | 708 | 1217 |
| TCR | 0.92 | 0.85 | 0.84 | 0.88 | 1 | 0.72 | 0.75 | 1 | 0.8 | 1 |

.

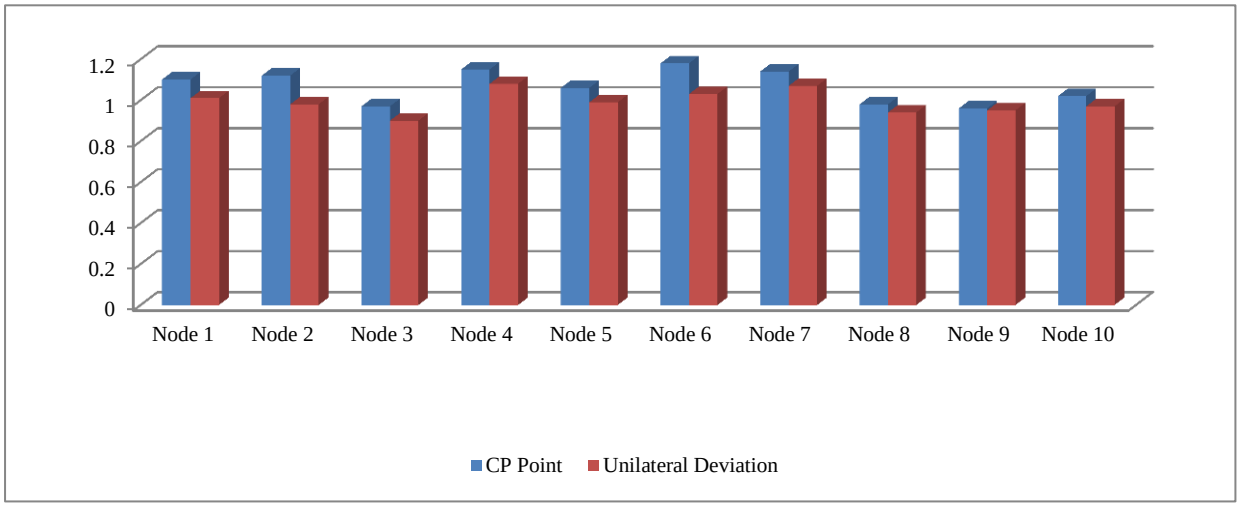



**Figure 5. Value of r(t) in CP point and CP point with unilateral deviation of one node.**

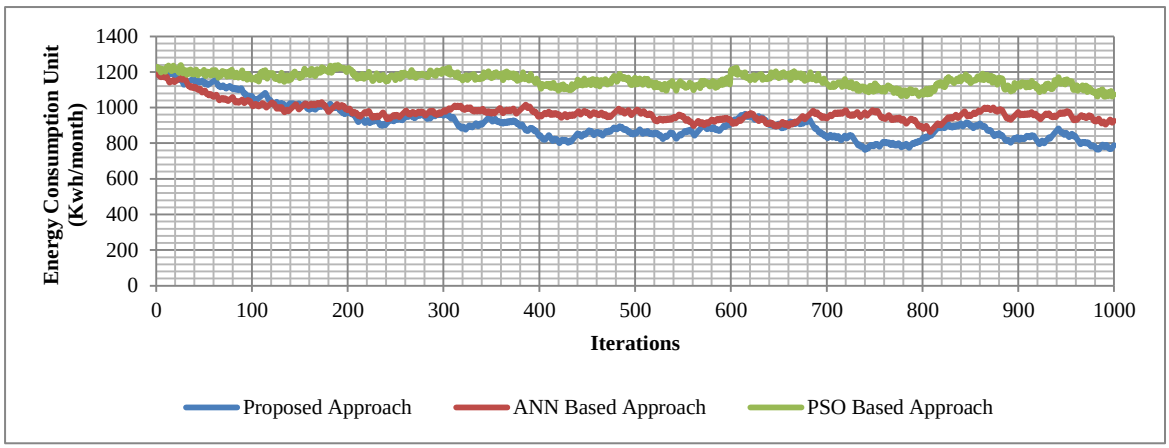



**Figure 6. Comparison of consumed energy by rival approaches**

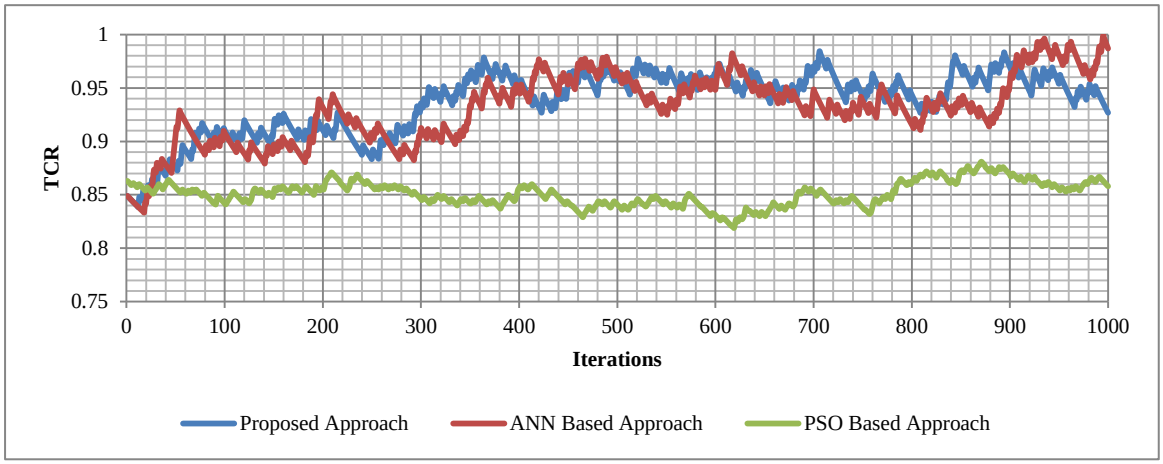



**Figure 7. Comparison of TCR measure of proposed approach and rival approaches.**

Although in terms of TCR, ANN and ICLA based approaches have similar results, but ICLA-based approach completes same volume of tasks by consuming less energy. In general, the results of conducted experiments show ICLA-based approach has notable advantages compared to the existing methods.

## 6. Conclusion

In this work, a new adaptive and decentralizes approach was proposed for more energy efficiency in edge environments. For the first time, we used ICLA for the energy efficiency problem in edge computing. Using this tool, we proposed an adaptive approach that considered local, neighbouring, and global workload status together for decision-making about putting a server in ON/STANDBY mode. As an innovative approach, considering local, neighbouring, and global status makes us capable to take better decisions for reaching higher level of energy efficiency. In the presented approach, edge resources collaborate each other to reach a more efficient plan for their active/sleep scheduling. The result of the conducted experiments shows capabilities of this approach for handling and serving user tasks using less energy consumption. For example, reaching task completion rate of 100%, the proposed approach decreased energy consumption of edge servers from 1053 Kwh to 902 Kwh. Because of complex interactions between edge devices, servers, and cloud centers, problem of energy efficiency in edge computing is challenging. Focus of the current work was on energy efficiency issue among edge servers. As a future work, we aim to extend this work to include interactions among edge and cloud servers and edge devices as well. These extensions need employment of efficient and flexible learning models to be capable of handling complexity in such networks.

## References

[1] M. Avgerinou, P. Bertoldi, and K. Castellazzi, "Trends in data centre energy consumption under the european code of conduct for data centre energy efficiency." *Energies*, vol. 10, no. 10, p. 1470, 2017.

[2] R. Ghaderi, M. Esnaashari, and M. R. Meybodi, "A Cellular Learning Automata-based Algorithm for Solving the Coverage and Connectivity Problem in Wireless Sensor Networks." *Adhoc & Sensor Wireless Networks*, vol. 22, 2014.

[3] M. Esnaashari and M. R. Meybodi. "Deployment of a mobile wireless sensor network with k-coverage constraint: a cellular learning automata approach." *Wireless networks*, Vol. 19, No. 5, pp. 945-968, 2013.

[4] M. K. Manshad, M. R. Meybodi, and A. Salajegheh. "A variable action set cellular learning automata-based algorithm for link prediction in online social networks." *The Journal of Supercomputing*, vol. 77, no. 7, pp. 7620-7648, 2021.

[5] M.D. Khomami, A. R. Rezvanian, and M. R. Meybodi. "A new cellular learning automata-based algorithm for community detection in complex social networks." *Journal of computational science*, vol. 24, pp. 413-426, 2018.

[6] M. Jahanshahi, M. Dehghan, and M. R. Meybodi. "On channel assignment and multicast routing in multi–channel multi–radio wireless mesh networks." *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 12, no. 4, pp. 225-244, 2013.

[7] X. Zichuan, W. Liang, W. Xu, M. Jia, and S. Guo. "Capacitated cloudlet placements in wireless metropolitan area networks." in *2015 IEEE 40Th conference on local computer networks (LCN)*, 2015, pp. 570-578.

[8] F. Qiang and N. Ansari. "Cost aware cloudlet placement for big data processing at the edge." *in 2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1-6.

[9] A.H. Safari-Bavil, S. Jabbehdari and M. Ghobaei-Arani, "An Efficient Approach to Solve Software-defined Networks based Virtual Machines Placement Problem using Moth-Flame Optimization in the Cloud Computing Environment." *Journal of AI and Data Mining*, vol. 9, no. 3, pp. 309-320, 2021.

[10] W. Yi and Y. Xia. "Energy optimal VM placement in the cloud." in *2016 IEEE 9th international conference on cloud computing (CLOUD)*, 2016, pp. 84-91.

[11] M. Sharma and R. Garg, "An artificial neural network based approach for energy efficient task scheduling in cloud data centers", *Sustainable Computing: Informatics and Systems*. vol. 26, 2020.

[12] L. Yuanzhe and S. Wang. "An energy-aware edge server placement algorithm in mobile edge computing." in *2018 IEEE International Conference on Edge Computing (EDGE)*, 2018, pp. 66-73.

[13] M. Demirci, "A survey of machine learning applications for energy efficient resource management in cloud computing environments," in *14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 1185-1190.

[14] L. Gu, J. Cai, D. Zeng, Y. Zhang, H. Jin, and W. Dai, "Energy efficient task allocation and energy scheduling in green energy powered edge computing." *Future Generation Computer Systems*, vol. 95, pp. 89-99. 2019.

[15] S. Nastic, T. Rausch, O. Scekic, S. Dustdar, M. Gusev, B. Koteska, M. Kostoska, B. Jakimovski, S. Ristov, and R. Prodan, "A serverless real-time data analytics platform for edge computing." *IEEE Internet Computing*, vol. 21, No. 4, pp. 64-71, 2017.

[16] C. Jiang, Y. Qiu, H. Gao, T. Fan, K. Li, and J. Wan, "An edge computing platform for intelligent operational monitoring in internet data centers." *IEEE Access*, vol. 7, 2019.

[17] C. Jiang, D. Ou,Y. Wang, Y. Li, J. Zhang, J. Wan, B. Luo, and W. Shi, "Energy efficiency comparison of hypervisors," *Sustainable Computing: Informatics and Systems*, vol. 22, pp. 311-321, 2019.

[18] J. Gao, "Machine learning applications for data center optimization," 2014.

[19] H. Momeni and N. Mabhoot, "An Energy-aware Real-time Task Scheduling Approach in a Cloud Computing Environment." *Journal of AI and Data Mining*, vol. 9, no. 2, pp. 213-226, 2021.

[20] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Energy efficient cloud resource management," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2014, pp. 386–391.

[21] I. AlQerm and B. Shihada, "Enhanced machine learning scheme for energy efficient resource allocation in 5G heterogeneous cloud radio access networks," in *28th Annual International IEEE Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1-7.

[22] S. Jiang, S. R. Priya, N. Elango, J. Clay, and R. Sridhar, "An Energy Efficient In-Memory Computing Machine Learning Classifier Scheme," in *32nd International Conference on VLSI Design and 18th International Conference on Embedded Systems (VLSID), Delhi*, NCR, India, 2019, pp. 157-162.

[23] R. Vafashoar, H. Morshedlou, A. Rezvanian, and M.R. Meybodi, Cellular Learning Automata: Theory and Applications, Vol. 307, Springer, 2021. [E-book] Available: https://link.springer.com/book/10.1007/978-3-030-53141-6.

[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks." In *IEEE conference on Computer Vision and Pattern Recognition, Columbus*, OH, 2014, pp. 1725-1732.

[25] J. Yue-Hei, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification." In *IEEE conference on computer vision and pattern recognition, Boston*, MA, 2015, pp. 4694-4702.

[26] H. Cao, M. Wachowicz, and S. Cha, "Developing an edge computing platform for real-time descriptive analytics." In *IEEE International Conference on Big Data*, 2017, pp. 4546-4554.

[27] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos, "Edge analytics in the internet of things." *IEEE Pervasive Computing*, vol. 14, no. 2, pp. 24-31, 2015.

[28] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards qos-aware fog service placement." in *IEEE first international conference on Fog and Edge Computing (ICFEC).*, 2017, pp. 89-96.

[29] M. I. Naas, P. R. Parvedy, J. Boukhobza, and L. Lemarchand, "iFogStor: an IoT data placement strategy for fog infrastructure." in *IEEE 1st International Conference on Fog and Edge Computing (ICFEC).*, 2017, pp. 97-104.

[30] C. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, and B. Jia, "Machine learning at facebook: Understanding inference at the edge," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2019, pp. 331-344.

[31] T. Chen, T. Moreau, Z. Jiang, L. Zheng, E. Yan, M. Cowan, H. Shen, L. Wang, Y. Hu, L. Ceze, C. Guestrin, and A. Krishnamurthy, "TVM: An automated end-to-end optimizing compiler for deep learning," [Online]. Available: https://arxiv.org/abs/1802.04799. [Accessed 2018].

[32] N. Rotem, J. Fix, S. Abdulrasool, S. Deng, J. H. Roman Dzhabarov, R. Levenstein, B. Maher, S. Nadathur, J. Olesen, J. Park, A. Rakhov, and M. Smelyanskiy, "Glow: Graph lowering compiler techniques for neural networks," [Online]. Available: https://arxiv.org/abs/1805.00907. [Accessed 2018].

[33] Google, "XLA is a compiler that optimizes TensorFlow computations." [Online]. Available: https://www.tensorflow.org/performance/xla/

[34] Apple Core ML, "Core ML: Integrate machine learning models into your app." [Online]. Available: https://developer.apple.com/documentation/coreml?changes=_8.

[35] NNPACK, "Acceleration package for neural networks on multi-core cpus." [Online]. Available: https://github.com/Maratyszcza/NNPACK.

[36] M. Dukhan, Y. Wu, and H. Lu, "QNNPACK: open source library for optimized mobile deep learning." [Online]. Available: https://code.fb.com/mlapplications/qnnpack/.

[37] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy consumption in mobile phones: A measurement study and implications for network applications,” in *ACM SIGCOMM Conf. Internet Meas. Conf.*, 2009, pp. 280–293.

[38] A. Sharma, V. Navda, R. Ramjee, V. N. Padmanabhan, and E. M. Belding, “Cool-Tether: Energy efficient on-the-fly wifi hot-spots using mobile phones,” in *ACM Emerging Netw. Exp. Technol.*, 2009, pp. 109–120.

[39] Z. Tang, S. Guo, P. Li, T. Miyazaki, H. Jin, and X. Liao, "Energy-Efficient Transmission Scheduling in Mobile Phones Using Machine Learning and Participatory Sensing," in *IEEE Transactions on Vehicular Technology*, vol. 64, no. 7, pp. 3167-3176, July 2015.

[40] A. Kumar, S. Goyal, and M. Varma, "Resource-efficient machine learning in 2 KB RAM for the internet of things," in *34th International Conference on Machine Learning*, vol. 70, 2017, pp. 1935-1944.

[41] X. Zhang, A. Ramachandran, C. Zhuge, D. He, W. Zuo, Z. Cheng, K. Rupnow, and D. Chen, "Machine learning on FPGAs to face the IoT revolution," in *Proceedings of the 36th International Conference on Computer-Aided Design*, 2017, pp. 819-826.

[42] G. Anastasi, M. Conti, M.D. Francesco, A. Passarella, "Energy conservation in wireless sensor networks: A survey", *Ad Hoc Networks*. vol. 7, pp. 537–568, 2009.

[43] M. A. Razzaque, C. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, pp. 1-44, 2013.

[44] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Network*, vol. 32, pp. 96-101, 2018.

[45] N.D. Lane, P. Georgiev, L. Qendro, "Deepear: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments Using Deep Learning", in *Proc. 2015 ACM Int'l. Joint Conf. Pervasive and Ubiquitous Computing*, 2015, pp. 283-94.

[46] H. Harb, A. Makhoul, and C. A. Jaoude, "En-route data filtering technique for maximizing wireless sensor network lifetime," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2018, pp. 298-303.

[47] J. Azar, A. Makhoul, R. Darazi, J. Demerjian, and R. Couturier, "On the performance of resource-aware compression techniques for vital signs data in wireless body sensor networks," in *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, 2018, pp. 1-6.

[48] J. Azar, R. Darazi, C. Habib, A. Makhoul, and J. Demerjian, "Using DWT lifting scheme for lossless data compression in wireless body sensor networks," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2018, pp. 1465-1470.

[49] J. Azar, A. Makhoul, M. Barhamgi, and R. Couturier, "An energy efficient IoT data compression approach for edge machine learning," *Future*

*Generation Computer Systems*, vol. 96, pp. 168- 175, 2019.

[50] Y. Wang, X. Dai, J. M. Wang and B. Bensaou, "A Reinforcement Learning Approach to Energy Efficiency and QoS in 5G Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1413-1423, June 2019.

[51] Q. Zeng, Y. Du, KK. Leung, and K. Huang, "Energy-efficient radio resource allocation for federated edge learning," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1-6.

[52] Y. Liu, C. He, X. Li, C. Zhang and C. Tian, "Power Allocation Schemes Based on Machine Learning for Distributed Antenna Systems," *IEEE Access*, vol. 7, pp. 20577-20584, 2019.

[53] C. He, Y. Zhou, G. Qian, X. Li, and D. Feng, "Energy Efficient Power Allocation Based on Machine Learning Generated Clusters for Distributed Antenna Systems," *IEEE Access*, vol. 7, pp. 575-584, 2019.

[54] K. Thangaramya, K. Kulothungan, R. Logambigai, M. Selvi, S. Ganapathy, and A. Kannan, "Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in IoT," *Computer Networks*, vol. 151, pp. 211-223, 2019.

# یک رویکرد تطبیقی جدید برای مصرف انرژی کارآمد در محاسبات لبه

**حسین مرشدلو* و علیرضا تجری**

**گروه نرم افزار، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی شاهرود، شاهرود، سمنان، ایران.**

ارسال ۲۰۲۲/۰۶/۲۵؛ بازنگری ۲۰۲۲/۰۸/۲۹؛ پذیرش ۲۰۲۳/۰۱/۰۸

**چکیده:**

محاسبات لبه یک رویکرد در حال تکامل برای نیازهای رو به رشد محاسبات و شبکه از دستگاه ها و اشیاء هوشمند است. محاسبات لبه برای پردازش های حجیم و سنگین این امکان را فراهم می کند تا این پردازش ها در راستای تأخیر کمتر، امنیت و حفظ حریم خصوصی از مراکز داده ابری به لبه شبکه بارگذاری شوند. هر چند مصرف بهینه انرژی در مراکز داده ابری تا کنون به طور گسترده ای در تحقیقات مختلف مورد مطالعه قرار گرفته است، اما این مبحث در بحث محاسبات لبه تا کنون چندان مورد بررسی قرار نگرفته است. در این مقاله، یک رویکرد تطبیقی و غیرمتمرکز جدید برای بهره وری بیشتر در زمینه انرژی در محیط های لبه پیشنهاد شده است. در رویکرد پیشنهادی، سرورهای لبه برای دستیابی به یک طرح کارآمد جهت مصرف انرژی در کل لبه شبکه با یکدیگر همکاری می کنند. رویکرد پیشنهادی یک رویکرد تطبیقی است و وضعیت بار کاری را در مناطق محلی، همسایه و کل لبه در نظر می‌گیرد. نتایج آزمایش‌های انجام شده نشان می‌دهد که این رویکرد قادر است کارایی انرژی را در لبه شبکه تا میزان قابل توجهی بهبود بخشد. به عنوان مثال، با نرخ تکمیل کار ۱۰۰ درصد، رویکرد پیشنهادی مصرف انرژی سرورهای لبه را از ۱۰۵۳ کیلووات ساعت به ۹۰۲ کیلووات ساعت کاهش می‌دهد.

**کلمات کلیدی:** محاسبات لبه، عامل، مراکز داده، مجازی سازی، مصرف کارآمد انرژی، اتوماتای یادگیر سلولی نامنظم.