Canonical Polyadic decomposition based energy efficient offloading algorithm for mobile edge computing

Nwogbaga, Nweso Emmanuel[1,3*]; Latip, Rohaya[1]; Affendey, Lilly Suriani[2]; Rahiman, Amir Rizaan Abdul[1]; Ituma, Chinagolum[3]; Nweke, Henry Friday[3]; Ogbaga, Ignatius Nwoyibe[3]; and Ikporo, Stephen Chibueze[3]

[1] Department of Communication Technology and Networks, Faculty of Computer Science, Universiti Putra Malaysia, Seri Kembangan, Malaysia

[2] Department of Computer Science, Faculty of Computer Science, Universiti Putra11Malaysia, Seri Kembangan, Malaysia

[3] Department of Computer Science, Ebonyi State University, Abakaliki

Corresponding Author: Nwogbaga, Nweso Emmanuel

Abstract

The application of IoT has given rise to the adoption of mobile edge computing network architecture by many IT-based organizations. However, the problem of limited energy supply, processing capacity, storage of mobile devices, and the inherent problem of bandwidth and transmission delay incurred in offloading tasks to the remote cloud server have been the challenges. In this research, we propose Energy Efficient Canonical Polyadic Decomposition based Offloading Algorithm. This algorithm utilizes canonical Polyadic decomposition based on rank_1 to decompose the input task data to produce the intermediate output with reduced task data size. The reduced data from the mobile layer is offloaded to the remote processing device where the processing capacity, energy, and storage are almost unlimited. The reduction method reduces the tasks deployed at the mobile layer, thereby reducing the mobile devices busy time and energy consumption. The data reduction ratio is used to simulate the computation offloading in mobile edge computing. The result from the simulation shows that the proposed algorithm improved the mobile devices energy consumption by 83.38% than the existing algorithms. The better result is achieved through the reduced data reduction ratio and reduced number of deployed task on mobile devices which are achieved through canonical Polyadic decomposition.

***Index Terms*—** Computation Offloading, Mobile Computing, Energy consumption, Task Scheduling, Resource Allocation

1. INTRODUCTION

The computing resources are under great revolutions these days. The introduction of cloud computing brought about so many technological changes within and around us. Individuals and organizations are leveraging the nearly unlimited storage and processing power of cloud service providers to revolutionize their modus operandi. The unlimited resources benefits of cloud computing instigated explosive growth of data in the cloud [1]. These advantages of the cloud infrastructure encouraged spontaneous application of Internet of Things (IoT) in different facets of life, such as healthcare, education, entertainment, virtual reality, augmented reality, and agricultural monitoring and processing. The increased application of IoT increased bandwidth usage and every device struggles to access the cloud either to send or receive information from the cloud server. This traffic issue introduced delay and energy consumption issues in the IoT devices that are limited in battery life. Mobile edge computing (MEC) aims at reducing the delay incurred in mobile cloud computing (MCC). MEC comes with processing capability near the task generating point as shown in Figure 1.

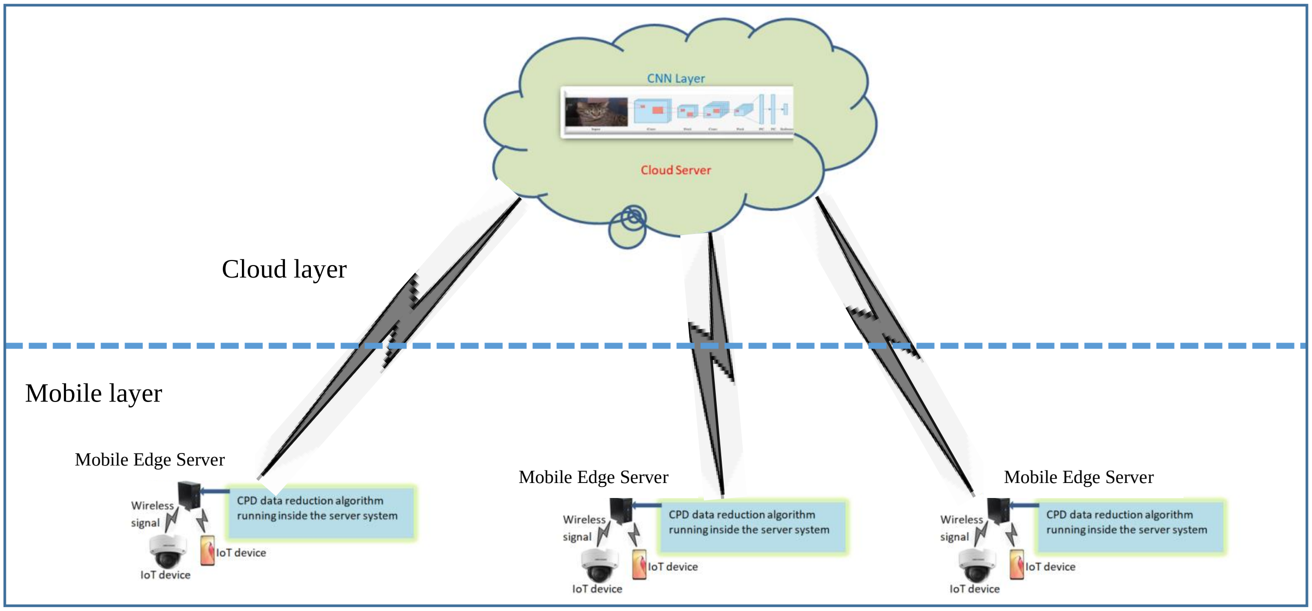

Fig. 1 Mobile Edge Cloud Computing Architecture

MEC is designed to process most of the data generated at the edge devices within the edge of the network. With time, MEC will take over all tasks processing while the cloud will mainly be used for storage if the issue of limited energy and processing capability can be further addressed[2,3]. The energy consumption at the edge is one of the greatest hindrances of the IoT application in MEC because devices are battery-powered and therefore are limited in their energy supply [4]. Energy consumption had been identified as a major issue in IoT application in MEC. This problem can be addressed through hardware or software perspective [1,5]. In software point of view, solutions can either be achieved through task scheduling [6–8] or ressources allocation [9,10].

In task scheduling point of perspective, Harris Hawks based optimization algorithm is proposed by [7] to optimize the energy usage in IoT network. Though the paper contributed in energy minimization, but the complexity of the method makes it difficult to be applied in systems with low configuration. Binary and partial offloading was proposed by [8], the paper proposed non-convex joint optimization to solve NP-hard energy problem in task offloading scheme. The proposed scheme contributed in energy consumption optimization but the scheme is time consuming because of so many options considered before convergence. Though there are limited researches yet on the energy optimization via input data reduction. [6] adopted local data reduction acquisition algorithm in addressing energy issue in IoT network. The paper proposed Markovian birth death method but the method is not suitable for low configured IoT devices. Convolution's data reduction approach was proposed by [11]. The paper applies part of CNN convolutions at the mobile device while the remaining convolutions for the classification analysis are offloaded to the cloud. The problem with this method is how to determine the number of convolutional layers to be applied at mobile level without increasing the delay above the task required time constrains. Such computational intensive analysis such as CNN classification analysis is better deployed at the cloud with more processing capacity, energy, and storage.

In this paper, the research focused on the energy consumption of mobile devices in IoT applications in MEC architecture. The solution of energy consumption is proposed via task scheduling. Specifically, we tackled the problem of energy consumption through the data reduction method. We proposed Energy Efficient Canonical Polyadic Decomposition based Offloading Algorithm (EECPD) to reduce input tasks data such that the task can be offloaded faster to the remote processor where the processing capacity is almost unlimited. This is achieved through decomposing the input tasks into several rank_1 based on the CPD approach. Consequently, the main contributions of this paper are;

- Input tasks are decomposed via rank_1 based CPD method and the data ratio of the output data to the input data is calculated.
- With the data ratio from the CPD data decomposition, the input tasks are offloaded to the cloud for processing and the reduced number of offloaded tasks is achieved.
- The deployed task which is the total input tasks minus the number of tasks offloaded is determined together with the energy consumed at the level of the mobile devices.
- The results of the proposed EECPD are compared with the online algorithm and low bandwidth first (LBF) algorithm which shows that the EECPD algorithm is performing better than the existing state of the art in terms of energy consumption at the mobile layer, data reduction ratio, and number of offloaded tasks.

The rest of this work is organized as follows. The next section discusses the related work as it concerns the energy consumption problem of MEC. Following is the problem of energy consumption in task scheduling which is formulated. Next, is the method of the proposed solution, followed by experimental settings and performance evaluations, and lastly, we presented the conclusion.

2. RELATED WORK

Energy efficiency is one of the main research issues in both MEC and cloud computing [12–15]. MEC or Fog computing complements cloud computing in addressing task processing challenges especially in latency sensitive cases [16,17]. The energy problem is still limiting some IoT applications in different areas. The energy problem is addressed in [18]. The authors proposed PSO based on Tabu Search to solve the issue of energy consumption in cloud computing, which is viewed as an NP-hard problem. Though the proposed algorithm outperformed simple PSO, Ant Colony Optimization, and Tabu Search, the delay incurred during task transmission was not handled. The algorithm considered only the resource allocation without recourse to task scheduling. According to [19], the problem of energy in MEC is viewed as a mixed integer nonlinear programming problem. They proposed reinforcement learning based on value iteration to achieve the policy of balancing the resource allocation and computation offloading in the system. The proposed method performs well in so many baseline methods except for the exhaustion method. Joint optimization of energy and latency problems for mobile edge computing is proposed in [20]. The multi-objective problem of energy consumption and latency is addressed by proposing a modified non-dominated sorting genetic algorithm (NSGA-II). The modified NSGA-II improved the tradeoff between latency and energy consumption for IoT applications. IoT can be applied in so many areas including in a vehicular network [21]. Though these IoT applications are hampered by mobile cloud computing issues [2]. The application of IoT in unmanned aerial vehicle-assisted MEC networks has energy issue as one of its greatest challenges. This is because both the unmanned aerial vehicle (UAV) and the mobile terminal (MT) are all limited in battery life. The solution to the problem of energy consumption is proposed in [20] which adopts the iterative algorithm with low complexity to solve the non-orthogonal multiple access (NOMA) problem of the network. [22] proposes energy efficient optimization to solve the energy problem in multiuser mobile edge computing based on a machine learning approach. The paper considered three criteria: the first criterion aims at linearly minimizing both energy and latency. The second criterion aims at minimizing the latency while the energy consumption is kept constant. Lastly, energy consumption is minimized while the latency is kept constant. The method improved energy efficiency within the tested parameters. A latency and energy consumption optimization strategy are proposed in [23] to address the problem of energy and latency of a secured MEC for 5G wireless networks. The proposed framework aims at minimizing energy consumption and latency in the computation and communication of the network. The paper proposes a linear combination of energy consumption and latency in solving the multi-objectives optimization issues. The proposed strategy achieved significant system improvement in terms of energy and latency. A queue management system based on Green Cloud for 5G networks is proposed in [24] to address the MEC problem of energy consumption and latency. The proposed method optimized the system energy consumption and latency significantly in a mobile edge computing environment.

Though researchers are intensifying efforts on seeing that the energy problem in IoT applications is ameliorated, more research is still needed to enable IoT to be applied in more sensitive situations like healthcare, traffic, smart city, and self-driving cars[21]. This research focuses on energy consumption minimization at the IoT devices from the perspective of task scheduling. This is achieved by reducing the input task data size to minimize the offloading time (mobile device busy time) thereby minimizing the energy consumption of the mobile device.

3. ENERGY PROBLEM OF TASK SCHEDULING

As many tasks are being generated at the mobile layer, the energy consumption at the mobile layer during task processing or offloading is getting higher [25]. The process of transmission of these tasks to the cloud is highly time-consuming and bandwidth demanding.

To address these problems, CPD based data reduction algorithm is proposed to reduce the task data size before offloading them to reduce the energy usage at the mobile layer.

The problem of the input tasks in MEC is viewed as a tensor decomposition problem. Mobile devices get signals in different formats from their environment. Such signals or tasks can be images videos, sounds, and numbers. Tensors read images and videos inform of matrices in order 3 and 4 respectively.

A multi-way array is known as a tensor. The number of ways, channels, modes, or dimensions of the tensor is referred to as the order of the tensor. A vector is seen as a one-dimensional tensor whereas a matrix is a two-order tensor, and any tensor more than a two-order tensor is referred to as a higher-order tensor [26].

Canonical Polyadic Decomposition factorizes an Nth-order tensor $\chi \, \varepsilon \, \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ into a linear combination of $a_r^{(1)} o \, a_r^{(2)} \, o \ldots o \, a_r^{(N)}$ which is a rank-1 tensor [27,28] such that given an input task as χ the output of the CPD reduction is calculated according to Equation (1)

$$\chi \cong \sum_{r=1}^{R} \lambda_r \; a_r^{(1)} o \, a_r^{(2)} \, o \ldots o \, a_r^{(N)} \tag{1}$$

where χ is the tensor. The objective is to transform all the input task into the form of χ.

4. PROPOSED METHOD

This research proposes Energy Efficient Canonical Polyadic Decomposition-based Offloading Algorithm (EECPD) to address the problem of energy consumption in mobile edge computing. The details of the proposed algorithm are present in this section.

During task offload from the mobile layer to a cloud, the tasks are generated at the mobile devices and preprocessed through the CPD algorithm according to Equation (1) and the original data size (X_s) and the reduced data size (X_{ps}) is calculated according to Equations (2) and (3).

$$X_s = h \times w \times c \tag{2}$$

$$X_{ps} = R(h + w + c) \tag{3}$$

The reduction ratio (R_d) of the output of the CPD is calculated as Equation (4).

$$R_d = \frac{X_{ps}}{X_s} \tag{4}$$

It is assumed that mobile devices have an equal rate of energy consumption at a busy times (E_b) and very negligible energy consumption at idle time. The busy time can be when the mobile device is offloading tasks to the remote server or when it is processing the tasks which were not offloaded. The offloading time (T_f) of original input task without reduction is given in Equation (5).

$$T_f = \frac{data\ size}{bandwidth} = \frac{X_s}{d_{ib}} \tag{5}$$

where d_{ib} is the device *i* allocated uplink bandwidth.

While the processing time (T_p) for tasks that were not offloaded is given in Equation (6).

$$T_p = {}^{X_{pm}}/_{APC} \tag{6}$$

where X_{pm} is the data size processed at the mobile layer, APC is the available processing capacity of the mobile device. If the task passes through the proposed data reduction, the transmission time will not be the same because the data size is reduced. Therefore, the proposed offloading time (T_{pf}) is given in Equation (7);

$$T_{pf} = \frac{data\ size}{bandwidth} = \frac{X_{ps}}{d_{ib}} \tag{7}$$

The X_s energy consumption is calculated in Equation (8);

$$E_c = E_b \times (T_f + T_p) \tag{8}$$

While the proposed method energy consumption (E_{pc}) is calculated using Equation (9);

$$E_{pc} = E_b \times (T_{pf} + T_p) \tag{9}$$

The overall energy consumed by the mobile devices throughout the simulation period for both original tasks average energy consumption ($E_{c(Avarage)}$) and the proposed reduced tasks average energy consumption ($E_{pc(Average)}$) are calculated according to Equation (10) and (11) respectively.

$$E_{c(Average)} = \sum_{j=1}^{J} \sum_{k=1}^{K} (E_c)/J \tag{10}$$

$$E_{pc(Average)} = \sum_{j=1}^{J} \sum_{k=1}^{K} (E_{pc})\ /J \tag{11}$$

where *j* = 1 to a total number of tasks and *k* = 1 to a total number of mobile devices.

Bandwidth is the mobile device allocated bandwidth (uplink bandwidth) used to transfer the task to the remote cloud. With our proposed attribute reduction approach, the transmission time is reduced since the data size is reduced. The data transmission time (T_f)for the original task data size and the proposed data transmission time (T_{pf}) are calculated and compared.

The objective function aims at reducing the energy consumption of mobile devices by minimizing the number of deployed tasks. To determine when to deploy the task on mobile or offload the task for cloud processing, the following objective function $fitness\ (task_i)$is considered;

$$fitness\ (task_i) = min(E_i) \tag{12}$$

where E_i is the energy consumed by a mobile device in processing $task_i$. To achieve this objective, the algorithm estimate and compares the energy consumption of the task according to Equations (8) and (9) to decide whether to deploy the task at the mobile or cloud layer. If the energy consumption will be smaller at the cloud, the algorithm will schedule the task for cloud processing, or else it will be processed on the mobile.

The algorithm summarizing the mobile cloud computing offloading based on the attribute reduction technique is presented in Algorithm (1).

Algorithm 1: Energy Efficient Canonical Polyadic Decomposition ($EECPD$) Task Scheduling Algorithm

Method(Mobile-Layer)
Input: Input Tasks (T)
Output: Data Rduction Ratio, Number of deployed Tasks (d_{task}), mobile device everage energy consumption ($E_{pc(Avarage)}$)
for $j = 1$ *to* J **do**
    get input tasks i from mobile device
    calculate original task energy consumption (E_c) according to Eq(8)
    calculate the proposed task energy consumption (E_{pc}) according to Eq(9)
    if E_{pc} *(Eq(23))* $<$ E_c *(Eq(22))* **then**
        $X_{h,w,c} = \sum_r^R (a_r^h \circ a_r^w \circ a_r^c)$
        Calculate the data reduction ratio (Eq (4))
        Schedule task i
        Use the **Fitness Function** Equation (12) to Schedule the task for cloud processing
    else
        process task i at mobile
calculate the number of deployed tasks (d_{task}) according to Eq(13)
calculate the average proposed task energy consumption ($E_{pc(Avarage)}$) according to Eq(11)
End Mobile-Layer

Algorithm 1: Energy Efficient Canonical Polyadic Decomposition (EECPD) Task Scheduling Algorithm

The algorithm generates an input task and checks whether the input task can be processed on the mobile or should be sent to the cloud. It does the checking by estimating the processing time on the mobile considering the available processing capability of the mobile device. The algorithm uses the estimated processing time to estimate the energy consumption of the task at both mobile and in the cloud. It compares the estimated energy consumption according to Equations (8) and (9) to decide whether to process the task at mobile or cloud. If there is a need to move the task to the cloud, the algorithm will perform attribute reduction based on the proposed CPD algorithm according to Equation (1), and then moves the output of the data reduction process to the cloud following the objective function as presented in Equation (12). The data reduction ratio is calculated according to Equation (4). The time taken to offload the task is calculated according to Equations (5) and (7) together with the energy consumed at the mobile layer according to Equations (8) and (9).

The result of the proposed data reduction is compared with the base paper's data reduction, while the number of deployed tasks for the proposed algorithm is also compared with the number of deployed tasks in the base paper. The energy consumption of the proposed algorithm is compared according to Equations (10) and (11) which show that the proposed algorithm saves energy for mobile devices than the existing algorithm. The experimental settings for the proposed algorithm are presented in the next section.

5. EXPERIMENTAL SETTINGS

This section discusses the experimental settings for the evaluation of the proposed computational offloading to solve the problem of energy at the mobile device level through the CPD data reduction approach. CPD data reduction utilizes the number of rank_1 (unique number of column vectors) to transform the input data (images and videos) into canonical form. The proposed approach increases the number of offloaded tasks from the mobile layer to reduce the busy time of the mobile devices during task offloading. There are two environments for the simulation. First is the implementation of CPD-based data reduction, which reduces the input data size. This was implemented in MatLab 2019a with Intel Core i7. The open-source Kaggle cat and dog dataset which contains 25,000 images of cats and dogs were used. It is an open-source dataset. The images were preprocessed and represented as $\mathbb{R}^{270\times180\times3}$ for every image in the dataset. The ratio of the data reduction is calculated and compared with the benchmark result.

Table 1 illustrates the MatLab experimental parameters.

Table 1 Experimental parameters for the CPD Data Reduction.

Parameter	Description	Value
Image height (h)	The height of the image (pixels)	270
Image width (w)	The width of the image (pixels)	180
Colour channel (c)	RGB	3
R (number of Rank_1)	R rank-1	{1 to 64}
Data size (X_s)	Input image data size (pixels)	Calculated
Reduced data size (X_{ps})	Output image data size (pixels)	Calculated

The second part of the experiment is the offloading process in the scheduling task which was implemented using Python 2.7 together with Networkx. The data reduction ratio was used in the simulation. The edge devices' capability was set to be 290 Gflops each and edge devices in the network was set to be between 20 and 90 devices according to NVIDIA Tegra K1. The uplink bandwidth of the individual edge devices were uniformly distributed between 10 to 1000 Mb/s. The tasks' latency is 0.2s each.

We first simulated the task without a data reduction ratio and measured the number of deployed tasks (Equation 13) on the mobile layer and the energy consumed. Secondly, we simulated with data reduction ratio and compared both results with the online scheduling algorithm and LBF algorithm of the existing study [11]. The results are presented in the next section.

6. PERFORMANCE EVALUATION

In this section, we present the improvement in the performance gained by our proposed attribute reduction for scheduling in a mobile edge computing environment. First, we compared our proposed data reduction ratio according to [11]. The results show that our proposed method reduces the data size more than the CNN layer reduction approach proposed by [11]. Our proposed algorithm reduces every offloadable task's data to a reasonable percentage of its original size before offloading, thereby reducing drastically the added transmission delay as a result of the task offloading. Since the processing capacity in the cloud is higher than that of the mobile devices, and the transmission delay is highly minimized. Furthermore, the response time, delay, and energy consumption are minimized.

6.1 Data reduction ratio

In this subsection, the data reduction ratio of the proposed algorithm is discussed. Table 2 and Figure 2 shows the values and graph of the first five data reduction ratio achieved by the first five (5) rank_1 values.

Table 2: CPD Data Reduction Ratio

R	H	W	C	Input Data Size	Output Data size	Reduction Ratio
1	270	180	3	145800	453	0.003106996
2	270	180	3	145800	906	0.006213992
3	270	180	3	145800	1359	0.009320988
4	270	180	3	145800	1812	0.012427984
5	270	180	3	145800	2265	0.015534979

The data reduction ratio increases linearly with the number of rank_1 values used for the data decomposition. From table 2, it is shown that 2, and 4 numbers of rank_1, produce a data reduction ratio of 0.006213992 and 0.012427984 which represents 0.62% and 1.24% of the original task size respectively.

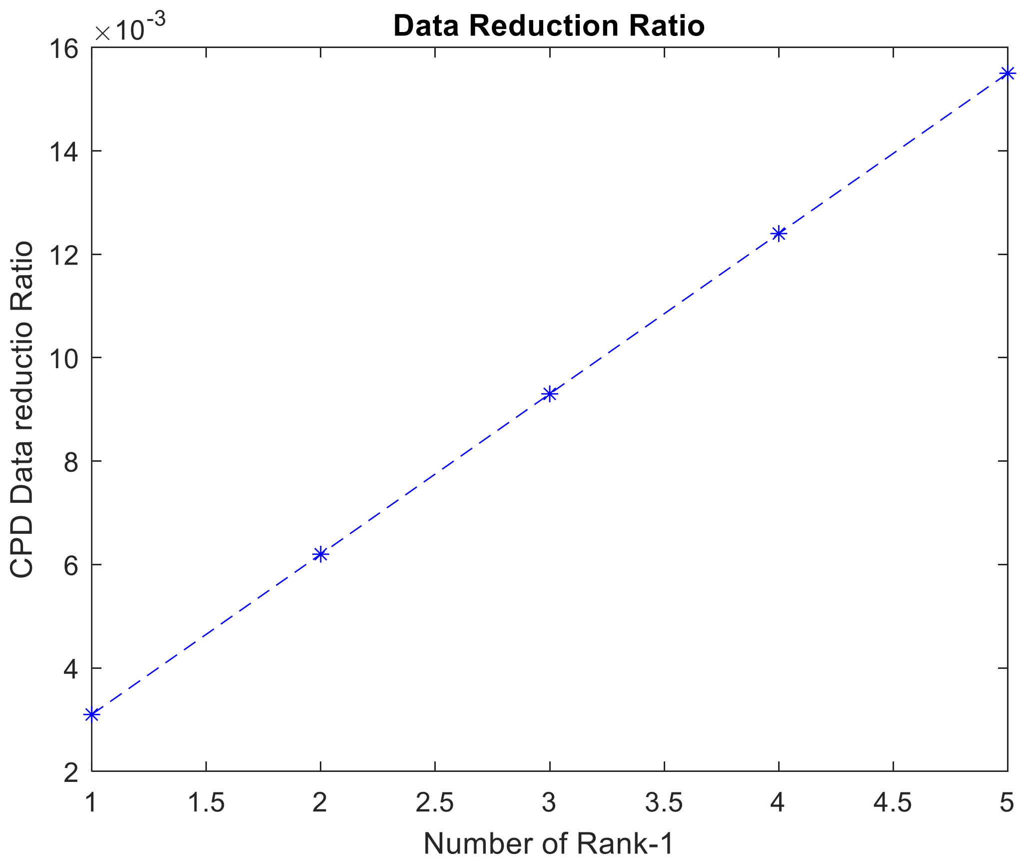

Fig. 2 CPD data reduction ratio

We compared the proposed attribute reduction of the canonical Polyadic decomposition method with the existing attribute reduction of CNN convolutions. Table 3 and Figure 3 present the comparison of data reduction of [11] with the data reduction of the proposed method. From Figure 3, the CNN data reduction ratio reduces as the CPD data reduction ratio increases. For instance, at convolution one (1), the CNN data reduction produced 0.245390947 reduction ratio while at Rank_1 one (1) the CPD data reduction ratio is 0.003106996. But at convolution 4 and Rank_1 four (4), their data reduction ratios are 0.002932099 and 0.012427984 respectively.

The results show that the reduction ratio of the CPD base approach reduces with an increase in the value of rank_1 while the reduction ratio of CNN based approach increases with the number of convolutions. Since mobile devices are limited in processing capability, it is not suitable to implement several CNN convolutional layers on mobile devices considering the computational complexity of each CNN layer. Therefore, the CPD approach is better in terms of the attribute reduction to be applied at the mobile device layer to minimize energy consumption and improve the quality of service in IoT application since it applies all the data reduction at the mobile layer.

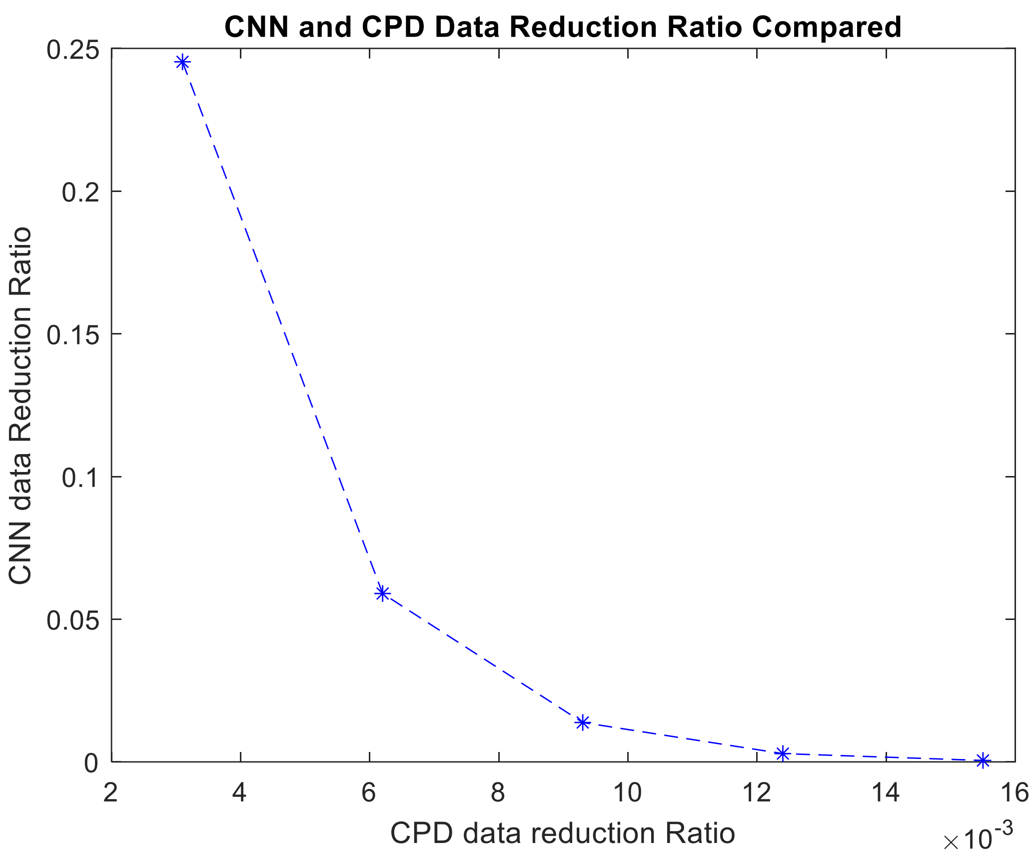

Fig. 3 CNN and CPD Data Reduction Ratio Compared

Table 3 CNN and CPD data reduction Ratio compared

CNN convolutions/CPD Rank_1	CNN Data Reduction Ratio	CPD Data Reduction Ratio
1	0.245390947	0.003106996
2	0.059074074	0.006213992
3	0.01382716	0.009320988
4	0.002932099	0.012427984
5	0.000534979	0.015534979

6.2 Number of the deployed task at the mobile layer

The number of deployed tasks is important in this research since the amount of energy consumed by the mobile device is determined by the duration of usage. Reducing the number of the deployed task means offloading more of the tasks to the remote server. This will reduce device usage and therefore, reduce energy consumption. In the simulation, the input tasks simulated are 200, 400, 600, 800, and 1000 tasks as shown in Table 4. The number of tasks that were deployed at the mobile layer is recorded. These deployed tasks are those tasks that couldn't be offloaded to the remote server. The objective of the proposed algorithm is to reduce the number of deployed tasks at the mobile layer because of the mobile device's limited battery life, processing capacity, and storage.

Table 4 Number of deployed tasks at the Mobile devices

Input Tasks (Number)	Number of Deployed Tasks		
	LBF	Online Algorithm	Proposed Algorithm
200	200	120	20
400	380	270	45
600	460	400	66
800	510	530	88
1000	560	660	110

Figure 4 and Table 4 present the values and graph of the comparison of the number of deployed tasks by the proposed algorithm and the LBF and Online algorithm of the existing work. From Figure 4, the proposed algorithm reduces the number of deployed tasks more than the existing algorithms. For instance, when the input tasks are 400, the LBF deployed 380 tasks at the mobile level and the Online algorithm deployed 270 whereas the proposed EECPD algorithm deployed only 45 tasks representing 95%, 67.5%, and

11.25% of the input tasks respectively. Likewise, when the input tasks are 1000, the LBF deployed 560 tasks and the online algorithm deployed 660 tasks while EECPD deployed 110 tasks representing 56%, 66%, and 11% of the input tasks respectively.

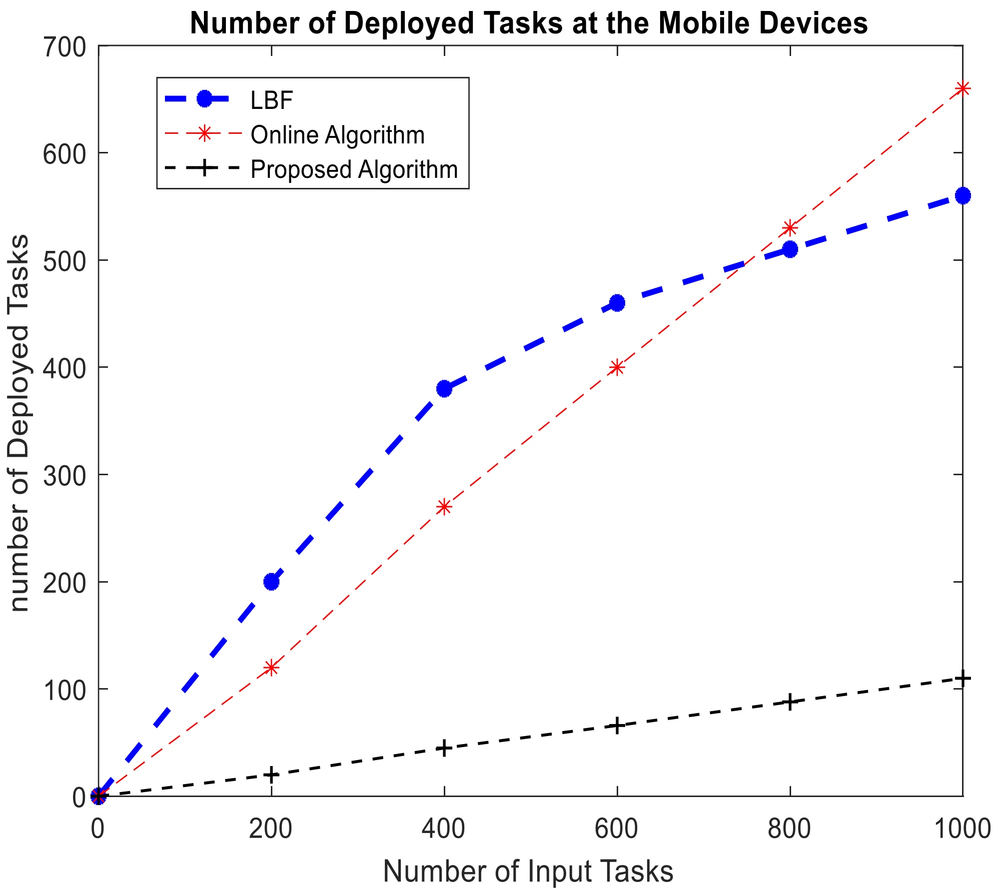

Fig. 4 Number of Deployed Tasks at the Mobile Layer

6.3 Energy Consumption at the mobile layer

This objective aims to minimize the mobile devices energy consumption in MEC. This is achieved by introducing rank_1 based canonical Polyadic decomposition at the mobile device level to reduce the input task data size. This enables the network to offload more tasks to the remote server since; energy, bandwidth, processing power, and storage are always the problems of the mobile devices. The task scheduling simulation using the EECPD algorithm is implemented with the input tasks of 200, 400, 600, 800, and 1000. The energy consumed at the mobile device level was measured.

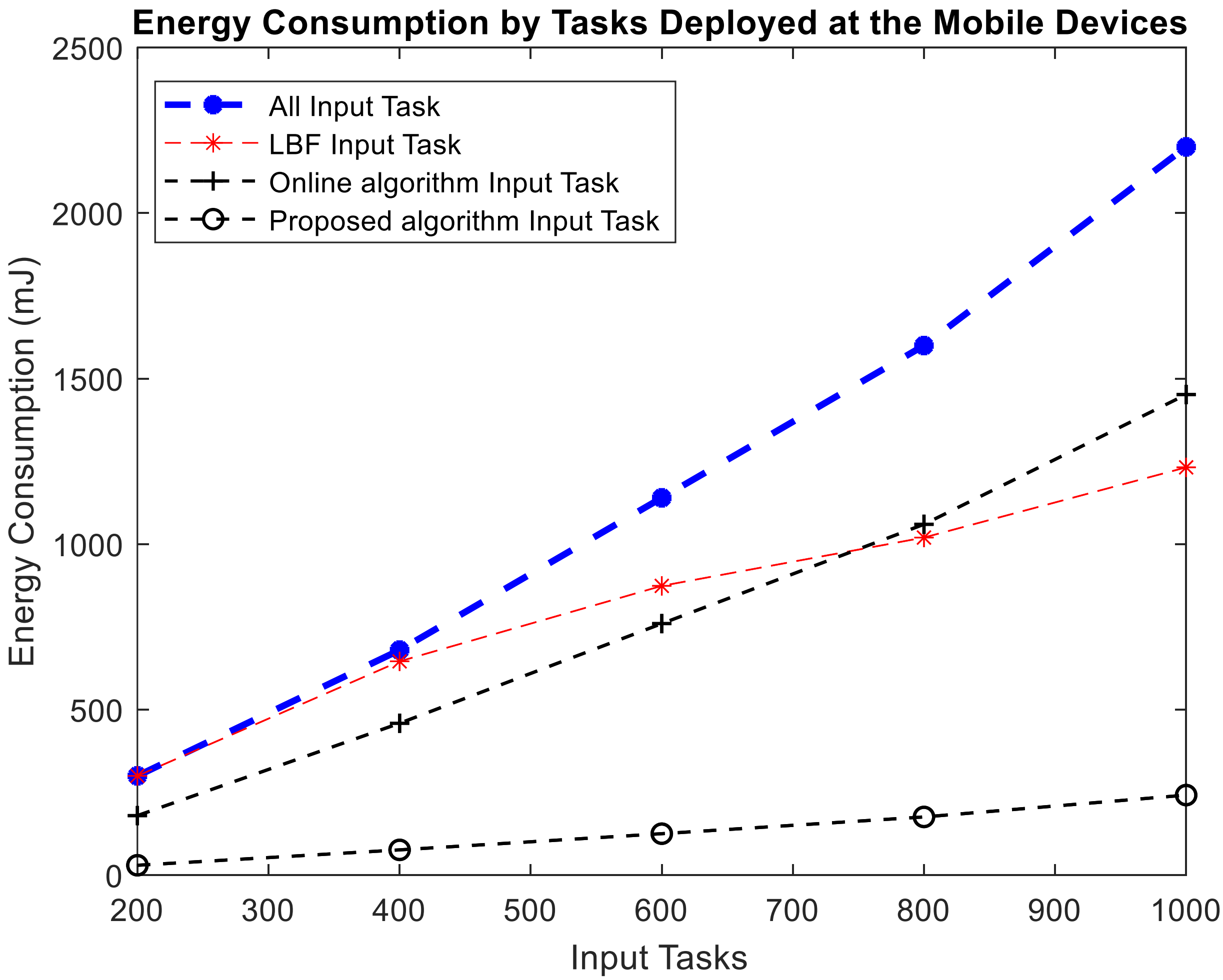

Fig. 5 Energy Consumption at the Mobile devices

Table 5 Energy Consumed by the deployed Tasks at Mobile

Number of input tasks	Energy Consumption by the Deployed Tasks			
	All Input (mJ)	LBF (mJ)	Online Algorithm (mJ)	Proposed Algorithm (mJ)
200	300	300	180	30
400	680	646	459	76.5
600	1140	874	760	125.4
800	1600	1020	1060	176
1000	2200	1232	1452	242

Figure 5 and Table 5 present the values and graphical representation of the energy consumed by the proposed algorithm and the existing algorithms. From Figure 5, it is clear that the proposed algorithms reduced the energy consumption of mobile devices more than the existing algorithms. For instance, when the simulated input tasks are 200, the energy consumption for deploying all input tasks on the mobile level is 300mJ, the LBF algorithm consumed 300mJ, the online algorithm consumed 180mJ, and the proposed EECPD consumed 30mJ representing 83.33% improvement from the online algorithm. Furthermore, when the number of input tasks is 1000, the energy consumed when all tasks are deployed, is 2200mJ. LBF algorithm consumed 1232mJ, and online consumed 1452mJ, while the proposed algorithm consumed 242mJ representing 83.33% energy enhancement for the proposed algorithm.

7. CONCLUSION

This research improved the energy usage of mobile devices in MEC. This was achieved by reducing the task data size before offloading the tasks to the cloud. This data reduction is achieved through the novel idea of rank_1 based CPD approach. The reduction ratio is used to simulate task processing in MEC and the data reduction ratio, number of deployed tasks, and energy consumption at the mobile level are measured and compared with LBF and online algorithms of the existing work. Experimental results proved the superiority of the proposed reduction approach in terms of data ratio, meaning it is capable of reducing the data size better than the existing algorithm. In terms of the number of deployed tasks, the proposed method's results also performed better. And lastly, the results of the energy consumption of the proposed algorithm are also better than the existing work by an average of 83.33% which proves that the proposed algorithm is better for energy efficiency for IoT applications in mobile edge computing.

Authors Contributions

Nwogbaga, Nweso Emmanuel: Formal analysis, Conceptualization, Software, Resources, Methodology, Writing – original draft, Writing – review & editing. **Latip, Rohaya:** Project administration, Supervision, Resources, Formal analysis, Conceptualization. **Affendey, Lilly Suriani:** Supervision, Validation, Conceptualization, Formal analysis. **Rahiman, Amir Rizaan Abdul:** Writing – review & editing, Investigation, Validation. **Ituma, Chinagolum:** Validation, Resources, Project administration. **Nweke, Henry Friday:** Formal analysis, Investigation, Resources, Software, Resources, Writing – review & editing. **Ogbaga, Ignatius Nwoyibe:** Investigation, Validation, Software, Resources. **Ikporo, Stephen Chibueze:** Validation, Software, Resources, review & editing.

Declarations

No funding was received for conducting this study.

Financial interests

The authors have no relevant financial or non-financial interests to disclose.

Declaration of Competing Interest The authors declare that they have no known competing financial interest.

REFERENCE

[1] K. Banka, K. Suo, Y. Shi, S. Baidya, A study of state-of-the-art energy saving on edges, Proc. 2021 ACMSE Conf. - ACMSE 2021 Annu. ACM Southeast Conf. (2021) 224–228. https://doi.org/10.1145/3409334.3452079.

[2] N.E. Nweke, F. Henry, Michael, J. S and Nwogbaga, Main Challenges Hampering Cloud Computing Adoption By Business Organizations, Res. J. Inf. Technol. 2 (2015) 1–11. http://www.researchjournali.com/pdf/1225.pdf.

[3] N.E. Nwogbaga, B.M. Emewu, I.N. Ogbaga, Critical Analysis Of Cloud Computing And Its Advantages Over Other Computing Techniques, J. Multidiscip. Eng. Sci. Technol. 3 (2016) 3955–3960.

[4] W. Li, W. Deng, R. She, N. Zhang, Y. Wang, W. Ma, Edge Computing Offloading Strategy Based on Particle Swarm Algorithm for Power Internet of Things, 2021 IEEE 2nd Int. Conf. Big Data, Artif. Intell. Internet Things Eng. ICBAIE

2021. (2021) 145–150. https://doi.org/10.1109/ICBAIE52039.2021.9389919.

[5] H. Yuan, G. Tang, D. Guo, K. Wu, X. Shao, K. Yu, W. Wei, BESS Aided Renewable Energy Supply Using Deep Reinforcement Learning for 5G and Beyond, IEEE Trans. Green Commun. Netw. 6 (2022) 669–684. https://doi.org/10.1109/TGCN.2021.3136363.

[6] S. Bebortta, D. Senapati, C.R. Panigrahi, B. Pati, An Adaptive Modeling and Performance Evaluation Framework for Edge-Enabled Green IoT Systems, IEEE Trans. Green Commun. Netw. 6 (2022) 836–844. https://doi.org/10.1109/TGCN.2021.3127487.

[7] K. Dev, P.K.R. Maddikunta, T.R. Gadekallu, S. Bhattacharya, P. Hegde, S. Singh, Energy Optimization for Green Communication in IoT Using Harris Hawks Optimization, IEEE Trans. Green Commun. Netw. 6 (2022) 685–694. https://doi.org/10.1109/TGCN.2022.3143991.

[8] Z. Li, N. Zhu, D. Wu, H. Wang, R. Wang, Energy-Efficient Mobile Edge Computing Under Delay Constraints, IEEE Trans. Green Commun. Netw. 6 (2022) 776–786. https://doi.org/10.1109/TGCN.2021.3138729.

[9] T. Zhang, S. Mao, Energy-Efficient Federated Learning with Intelligent Reflecting Surface, IEEE Trans. Green Commun. Netw. 6 (2022) 845–858. https://doi.org/10.1109/TGCN.2021.3126795.

[10] M. Diamanti, P. Charatsaris, E.E. Tsiropoulou, S. Papavassiliou, The Prospect of Reconfigurable Intelligent Surfaces in Integrated Access and Backhaul Networks, IEEE Trans. Green Commun. Netw. 6 (2022) 859–872. https://doi.org/10.1109/TGCN.2021.3126784.

[11] H. Li, K. Ota, M. Dong, Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing, IEEE Netw. 32 (2018) 96–101. https://doi.org/10.1109/MNET.2018.1700202.

[12] Z. Ali, L. Jiao, T. Baker, G. Abbas, Z.H. Abbas, S. Khaf, A deep learning approach for energy efficient computational offloading in mobile edge computing, IEEE Access. 7 (2019) 149623–149633. https://doi.org/10.1109/ACCESS.2019.2947053.

[13] R. Aldmour, S. Yousef, M. Yaghi, G. Kapogiannis, MECCA offloading cloud model over wireless interfaces for optimal power reduction and processing time, 2017 IEEE SmartWorld Ubiquitous Intell. Comput. Adv. Trust. Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov. SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017 - . 2021 (2018) 1–8. https://doi.org/10.1109/UIC-ATC.2017.8397639.

[14] S.U.R. Malik, H. Akram, S.S. Gill, H. Pervaiz, H. Malik, EFFORT: Energy efficient framework for offload communication in mobile cloud computing, Softw. - Pract. Exp. 51 (2021) 1896–1909. https://doi.org/10.1002/spe.2850.

[15] R. Aldmour, S. Yousef, T. Baker, E. Benkhelifa, An approach for offloading in mobile cloud computing to optimize power consumption and processing time, Sustain. Comput. Informatics Syst. 31 (2021) 100562. https://doi.org/10.1016/j.suscom.2021.100562.

[16] H. Liazid, M. Lehsaini, A. Liazid, Data transmission reduction using prediction and aggregation techniques in IoT-based wireless sensor networks, J. Netw. Comput. Appl. 211 (2023). https://doi.org/10.1016/j.jnca.2022.103556.

[17] M. Songhorabadi, M. Rahimi, A.M. MoghadamFarid, M. Haghi Kashani, Fog computing approaches in IoT-enabled smart cities, J. Netw. Comput. Appl. 211 (2023). https://doi.org/10.1016/j.jnca.2022.103557.

[18] Y. Ebadi, N.J. Navimipour, An energy-aware method for data replication in the cloud environments using a T abu search and particle swarm optimization algorithm, Concurr. Comput. Pract. Exp. (2018) 1–10. https://doi.org/10.1002/cpe.4757.

[19] V.C.M.Z. Huan, J. Kai, L. Xuxun, L. Xiuhua, Leung, Deep Reinforcement Learning for Energy-Efficient Computation Offloading in Mobile-Edge Computing, IEEE Internet Things J. 9 (2022) 1517–1530.

https://doi.org/10.1109/JIOT.2021.3091142.

[20] L. Cui, C. Xu, S. Yang, J.Z. Huang, J. Li, X. Wang, Z. Ming, N. Lu, Joint optimization of energy consumption and latency in mobile edge computing for internet of things, IEEE Internet Things J. 6 (2018) 4791–4803. https://doi.org/10.1109/JIOT.2018.2869226.

[21] E.C. Eze, S. Zhang, E. Liu, N.E. Nwogbaga, J.C. Eze, RECMAC: Reliable and efficient cooperative cross-layer MAC scheme for vehicular communication based on random network coding technique, in: 2016 22nd Int. Conf. Autom. Comput. ICAC 2016 Tackling New Challenges Autom. Comput., 2016: pp. 342–347. https://doi.org/10.1109/IConAC.2016.7604943.

[22] Y. Guo, R. Zhao, S. Lai, L. Fan, X. Lei, G.K. Karagiannidis, Distributed Machine Learning for Multiuser Mobile Edge Computing Systems, IEEE J. Sel. Top. Signal Process. 16 (2022) 460–473. https://doi.org/10.1109/JSTSP.2022.3140660.

[23] S. Lai, R. Zhao, S. Tang, J. Xia, F. Zhou, L. Fan, Intelligent secure mobile edge computing for beyond 5G wireless networks, Phys. Commun. 45 (2021) 101283. https://doi.org/10.1016/j.phycom.2021.101283.

[24] S. Gopi, R; Suganthi, S. T; Rajadevi, R; Johnpaul, P; Bacanin, N; & Kannimuthu, An enhanced green cloud based queue management (GCQM) system to optimize energy consumption in mobile edge computing, Wirel. Pers. Commun. 117 (2021) 3397–3419.

[25] N.E. Nwogbaga, R. Latip, L.S. Affendey, A.R.A. Rahiman, Attribute reduction based scheduling algorithm with enhanced hybrid genetic algorithm and particle swarm optimization for optimal device selection, J. Cloud Comput. 11 (2022) 1–17. https://doi.org/10.1186/s13677-022-00288-4.

[26] A. Cichocki, Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions, (2014) 1–30. https://doi.org/abs/1403.2048.

[27] R.A. Harshman, FOUNDATIONS OF THE PARAFAC PROCEDURE: MODELS AND CONDITIONS FOR AN "EXPLANATORY" MULTIMODAL FACTOR ANALYSIS, UCLA Work. Pap. Phonetics. 16 (1970) 1–84. https://doi.org/10.1134/S0036023613040165.

[28] F.L. Hitchcock, Multiple Invariants and Generalized Rank of a P-Way Matrix or Tensor, J. Math. Phys. 7 (1927) 39–79. https://doi.org/10.1002/sapm19287139.