Investigation into the effect of data reduction in offloadable task for distributed IoT-fog-cloud computing

Nweso Emmanuel Nwogbaga[1,2]* , Rohaya Latip[1]*, Lilly Suriani Affendey[3] and Amir Rizaan Abdul Rahiman[1]

Abstract

With the increasing level of IoT applications, computation offloading is now undoubtedly vital because of the IoT devices limitation of processing capability and energy. Computation offloading involves moving data from IoT devices to another processing layer with higher processing capability. However, the size of data offloaded is directly proportional to the delay incurred by the offloading. Therefore, introducing data reduction technique to reduce the offloadable data minimizes delay resulting from the offloading method. In this paper, two main strategies are proposed to address the enormous data volume that result to computation offloading delay. First, IoT Canonical Polyadic Decomposition for Deep Learning Algorithm is proposed. The main purpose of this strategy is to downsize the IoT offloadable data. In the study, the Kaggle-cat-and-dog dataset was used to evaluate the impact of the proposed data compression. The proposed method downsizes the data significantly and can reduce the delay due to network traffic. Secondly, Rank Accuracy Estimation Model is proposed for determining the Rank-1 value. The result of the proposed method proves that the proposed methods are better in terms of data compression compared to distributed deep learning layers. This method can be applied in smart city, vehicular networks, and telemedicine etc.

Keywords: Internet of things, Fog computing, Data compression, Offloading, Attribute reduction

Introduction

Today, the interconnectedness of many IoT devices is naturally raising concerns that require research attention and in-depth investigation. With respect to the increasing amount of data that is transferred (offloaded) over networks, especially in IoT systems, it is undoubtedly vital to investigate concepts for downsizing the volume of data that is sent from sensors to the network. The need for the offloading arises because of the limitations of processing capability and battery life of the IoT devices [1, 50, 54] In large IoT systems, where for instance multiple cameras are used as sensors, there might be a need to interpret or analyze the taken images or video by means of neural networks. As sensors or edge devices usually lack in computational power, such images are typically transferred to the fog or cloud, where image recognition, intrusion detection, face recognition algorithms or any other analysis are applied on them with more powerful hardware [13, 43, 45]. The transfer of a large amount of image or video data to the cloud might lead to delay and network congestions. Industry's predictions show that the global IP traffic utilization annually will exceed 3.3 Zettabytes by 2021. This is because of ubiquitous devices connected to the network (such as smart phones, sensors, RFID, cameras, smart TV, PC, etc.) that generate variety of data referred to as big data [41]. The smart devices traffic is quickly exceeding the PC traffic [23]. The large amount of data generated by these IoT devices as well as their transmission to fog or cloud for analysis have been jointly identified as crucial

* Correspondence: nwesonwogbaga@yahoo.com;
gs51878@student.upm.edu.my; rohayalt@upm.edu.my
[1]Department of Communication Technology and Network, Universiti Putra Malaysia, Seri Kembangan, Malaysia
Full list of author information is available at the end of the article

research issues [5, 29, 31, 38, 47];. The IoT devices, on the other hand, cannot process all the data they generate because of their limited storage, processing capability and battery life [6, 14, 49]. To achieve the required results for IoT requests, the data need to be transmitted to the remote fog or cloud with higher processing capability for processing [53, 54], hence the need for the IoT application in cloud Computing environment. Some of the IoT applications (like the Internet of Vehicles (IoV) that use sensors, actuator and cameras) transfer their data to Roadside Units (RSUs) for processing [15, 24, 26]. Offloading task to remote processing node especially with a large data size increases the delay [13, 43, 52, 54, 55]. In order to reduce the delay incurred as result of offloading large data size from IoT to either fog or cloud, there is need for data reduction method to downsize the amount of data to be offloaded.

The traditional mobile cloud computing (MCC) is two-tiered that consist of cloud server and IoT devices. The MCC is no longer capable of handling the IoT request especially in a low latency request from IoT, because of the high volume of data from this avalanche of IoT devices. To improve the network quality of service (QoS), the three-tier architecture is introduced to include cloud server, fog and user equipment (UE) that is the IoT devices. In 2012, the concept of fog computing was introduced by Cisco to overcome the challenges facing IoT applications in traditional Cloud computing [34]. Fog computing is an intermediate layer or bridge, between the user (IoT devices) and cloud computing. The intermediate layer is also known as the cloud at the edge according to Liu et al. [32]. This paradigm that acts as an intermediate layer distributes network, computation, and storage thereby extending cloud-based services closer to the IoT devices [11, 38]. Fog computing complements rather than substitutes cloud computing [42]. The advantages of fog include mobility support, low delay, and a smaller number of devices to serve [2]. The idea behind IoT-fog-cloud computing architecture is to improve the network quality of service by minimizing the processing time of IoT request. This is to enable IoT to be applied in real time application like telemedicine [44], self-driving cars and smart city. Different researchers have attempted different approaches to minimize the response time of IoT request. Such approaches include cooperation computing [18, 25, 29] offloading approach [5, 22, 40], scheduling approach [17, 35, 38, 39, 46] and data reduction [28]. Unfortunately, these attempts are not without drawbacks.

In this paper, we proposed a novel strategy of reducing the size of offloadable data for transmission and maintaining the data accuracy. First, we propose an IoT Canonical Polyadic Decomposition for Deep Learning Algorithm (IoTCP_DL) to be implemented at IoT layer.

We also propose a novel Rank Accuracy Estimation Model (RAEM). This algorithm together with the model will be applied to downsize the volume of offloadable data that the IoT device is expected to offload to a higher processing element without significant effect on the data accuracy, thereby reducing the smart device delay, traffic, and improving the IoT request response time. Consequently, the main contributions of this research are as follow:

- An attribute reduction method is proposed based on canonical Polyadic decomposition (IoT Canonical Polyadic Decomposition for Deep Learning (IoTCP_DL) Algorithm) to be implemented at IoT node to decompose the IoT device offloadable data in the network. This is to downsize the size of the offloadable data to less than 10 % of the original size without affecting the data accuracy significantly. This is important as we strive to ensure that the volume of data being offloaded from the IoT device is reduced thereby reducing the transmission delay and the accuracy is unaffected.
- From the experiment with different rank-1, the size and accuracy of data produced, we propose a novel Rank Accuracy Estimation Model (RAEM) that can be used to estimate the accuracy of a particular R (rank-1) value to be used at IoT node in order to minimize the data size and maintaining the data accuracy within a certain accuracy threshold.
- We establish a relationship between R rank-1 and accuracy, and prove that given a particular R value, the model can estimate the accuracy it will produce, thereby making it possible for it to be used to predict the rank-1 at IoT node. This accomplishes the important task of reducing the data size of the task the IoT will be offloading to the higher processing node.
- Equally important is the fact that the intermediate data (the output of CPD) will also serve to preserve the privacy of the data on the network as the intermediate data from the decomposition process is of a different format from the original format.

The remainder of the paper is organized as follows. The next section (section 2) is related works. This is followed by materials and methods (Section 3) where the introduction to canonical Polyadic decomposition and the proposed methods are discussed. Section 4 presents experiments and evaluation while Section 5 presents Results and discussions. The comparison of attribute reduction of the proposed method and that of CNN layers are presented together with the comparison of classification accuracy of the proposed method with the original images. This, in turn, is followed by the

conclusion in Section 6 while future work is presented in the final section (Section 7).

Related work

The IoT-fog-cloud computing architecture is aimed at improving the network quality of service by minimizing the response time of IoT requests in order to enable IoT to be applied in real-time applications [48]. In this section, we reviewed different approaches that researchers have taken to address this problem. Some researchers approached the problem from the point of view of scheduling policy based on resources cooperation, computation offloading, and/or attribute reduction. The problem of choosing the optimal offloading policy, solution to jamming attacks and reducing interference in mobile computing was presented in [52]. Mobile offloading based on Reinforcement learning for Edge Computing was proposed for addressing the Jamming and Interference issues in mobile computing. The paper combines actor-critic and deep Q-learning to solve the high dimensional policies for offloading in mobile computing. The aim is to minimize the response time and energy consumption at the mobile devices [54]. addressed the problem of energy consumption in IoT cloud computing by formulating the inference offloading method to minimize the energy consumption of the devices and average delay. The paper aim at formulating the inference offloading issue within the 5G enabled mobile edge computing for Deep Neural Network-driven artificial intelligent applications to improve on the energy consumption and the delay of the mobile devices.

Workflow scheduling and resource allocation [12, 27, 31] have played a major role in optimizing makespan (response time) and latency of IoT requests in fog-cloud architecture. The efficient algorithm for response time of the IoT request was proposed based on scheduling by prioritizing and distributing tasks among the computing resources [25, 38], presented the difference between resource sharing and cooperation. According to the paper, in resource sharing, IoT devices are scheduled to access resources differently. Because IoT devices access resources differently, resource sharing becomes a scheduling problem. On the other hand, cooperation refers to IoT (agents) contributing their capabilities towards achieving a common goal. They went further to propose a Cooperation Algebra Technique which integrates mobile device processing capabilities based on workflownets. The proposed cooperation method contributed to reducing the task processing time. The paper assumed that all IoT requests must be executed at the IoT levels by the cooperation of different IoT nodes through fog computing. But considering the limitation of IoT processing capability, storage, and battery life, this idea will not be the optimal solution in many heavy IoT tasks

especially those involving image and video data. Another approach based on the cooperation method is a delay-aware online workload allocation and scheduling (DAOWA) that aimed at reducing processing delay. This is achieved using the Lyapunov theory. The cooperation among different sets of fog and cloud is considered in fog paradigm. The computational demand of IoT applications in fog computing is on the increase. The paper investigated task allocation scheme in an IoT-fog-cloud architecture, for minimizing task service delay. This is to satisfy more delay-sensitive tasks [27, 31]. Furthermore, G. Li et al.[29] proposed a fog-cloud cooperation scheduling algorithm to minimize energy usage in IoT-fog-cloud architecture. This was achieved in the fog layer and cloud layer based on queue theory while nonlinear programming was applied to solve the energy problem and minimize the IoT task request delay. Similarly, in [16], the paper presented resources cooperation to minimize IoT request response time for Vehicular Communication using Random Network Coding Technique.

Computational offloading is one of the new techniques that make possible the application of IoT in cloud computing. Computation offloading helps to address the limitations of IoT resources and improves the IoT request response time. A Markov decision process was used in network resources optimization based on unmanned aerial vehicle (UAV). The aim is to improve the long-range, low cost coverage communication problem through offloading approach [9]. Jaddoa et al. [21] modeled the response time and cost of energy for different task offloading options. The offloading can either be to fog or cloud. The model is used at the IoT node to take the decision on where to offload the task or whether to process it at the IoT level. Many offloading policy proposed by researchers considers mobile edge paradigm [33]. The computation offloading issues in heterogeneous devices at the network edge and cloud layer was presented in [36]. They presented computation offloading method to predict the energy consumption and response time of IoT task by investigating edge devices or VMs at the cloud using regression models. Fog computing was introduced to reduce the distance travelled by data from data source to the designated data-center. This is for efficient and quick response to sensor's requests in IoT environment. Bonomi et al. [8] defined the fog computing as the most appropriate environment for applications like internet of smart devices and Vehicular networks. The advantages of fog computing were identified such as efficiency, latency reduction and reliability for IoT task processing. Al-Khafajiy et al. [3] proposed a fog framework to improve quality of service through computation offloading. The computation offloading employs collaboration approach within different fog devices to shared and process data among the devices. Min

et al. [37] presented a deep Reinforcement Learning via computation offloading method for selection of the mobile device and also for the selection of the offloading rate based on available mobile energy. This is to improve the IoT request response time and prolong the battery life usage of the IoT devices. In a similar way, a binary mobile edge computing offloading scheme was proposed in [30]. The paper uses reinforcement learning to decide when to offload to fog or cloud node. The approach can reduce the average delay and energy cost compared to full-offloading or non-offloading schemes. An intelligent offloading method for vehicular network was proposed based on reinforcement learning by [40]. It is hinged on Markov chains and is used to address the smart vehicle issues for travelers' experience and users' well-being. Another researcher [57] also proposed optimized task offloading and scheduling for vehicular networks using game theory approach and langrage multiplier for scheduling to improve the response time of IoT requests. Mobile edge computing (MEC) has a faster data processing speed and data transmission. MEC has passed through several optimizations as a new computing paradigm in internet of connected vehicles (IoV) application [39, 40]. Offloading approach in solving IoT problems in fog-cloud computing can as well be applied in security issues [53]. proposed computation offloading scheme called edge computing offloading (ECO) with privacy preservation in internet of connected vehicles (IoV). It addressed the privacy challenges of smart device applications in internet of connected vehicles. In general, computational offloading means offloading heavy and complex workloads to fog or cloud server where there are higher computing resources. In MEC, the essence of offloading is to move heavy tasks on demand to fog computing to achieve lower latency and energy cost as well as enhance the quality of computing services [4, 51].

Attribute reduction is another approach to improve the IoT request response time. In this approach, the size of data from IoT node for offloading can be reduced through attribute reduction technique, without affecting the meaning (accuracy) of the data. This will reduce the time taken to transfer the data (network traffic) from IoT node to fog or cloud node. From the literature, there is limited research on this area that focused on improving the IoT request response time through minimizing the network traffic as the number of connected smart devices increases. The attempt made by [28] to optimize network performance by distributing Deep Learning layers in edge-cloud computing was to reduce the data size through convolutional layer. The method reduces the input data size, thereby reducing the amount of data flowing in the network. Their model reduces the offloadable data in the network, because some deep learning layers are implemented at the IoT node. The output of

the layers implemented at the IoT node is transmitted to the next level. The transmitted data are reduced in size and are no longer in their original format. The problem with this method is that the transmitted data cannot be converted back to its original format. They are not suitable for other analyses except for deep learning classification. Secondly the implementation of convolutional layers on the IoT devices is not the best option considering their processing, storage and battery life limitations.

Altogether, from this literature review, it is clear that there is insufficient research on how to downsize the data flowing on the network to reduce the IoT offloadable data size which will reduce the data transmission time. Hence, there is the obvious need to research more on attributes reduction on image and video data as a better approach to improve the response time of IoT request, making IoT more applicable in real-time applications.

Materials and methods

In this section, we describe Canonical Polyadic Decomposition as an important foundation/background for our proposed attribute reduction method.

Canonical Polyadic decomposition

Canonical Polyadic Decomposition (CPD) is presented which factorizes an Nth-order tensor $\chi \, \varepsilon \, \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ into $\sum_{r=1}^{R} \searrow_r a_r^{(1)} o \, a_r^{(2)} \, o \ldots o \, a_r^{(N)}$ that produces the eq. (1). If the data is in image form, then N is 3 (representing the width, height, and number of channels). If the data is in video format, then N is 4 (width, height, number of channels, and number of frames). IoT devices receive signals from the environment in different formats. The IoT input can be numbers, images, sound and/or video. Images and videos are read in tensor form.

A tensor is a multi-way array. The order of a tensor is the number of its ways, channels, modes, or dimensions. A vector is a tensor of order one. A matrix is a tensor of order two and any tensor of order greater than two is referred to as higher-order tensor [10].

Canonical Polyadic Decomposition (CPD) also known as CandeComp or ParaFac 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 rank-1 tensor [7, 19, 20] given by

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

$$= \wedge \times_1 A^{(1)} \times_2 A^{(2)} \ldots \times_N A^{(N)}$$

$$= \left[\left[\wedge; A^{(1)}, A^{(2)}, \ldots, A^{(N)} \right] \right] \quad (1)$$

where $\mathbb{R}$ is a set of real numbers, $\searrow_r$ is a non-zero entry

of diagonal core tensor, "o" is the outer product, "R" is the number of rank-1 . "N" is the order of the tensor, where n = 1, 2, 3, …, N. "a" is the value representing each pixels. Each rank-1 of the N order tensor can be defined by the outer product of N vectors as $a_r^{(1)} o\, a_r^{(2)}\, o...o\, a_r^{(N)}$ denoting rth rank-1 tensor in eq. (2) [56].

Therefore, every element $X_{h,\,w,\,c} \in X$ (where $X_{h,\,w,\,c}$ is the tensor of order 3 of element X) can be equivalently represented in canonical Polyadic decomposition as

$$X_{h,w,c} = \sum_{r=1}^{R} a_r^{(h)} o\, a_r^{(w)}\, o...o\, a_r^{(c)} \qquad (2)$$

where h is the image height, w is the image width and c is the number of channels.
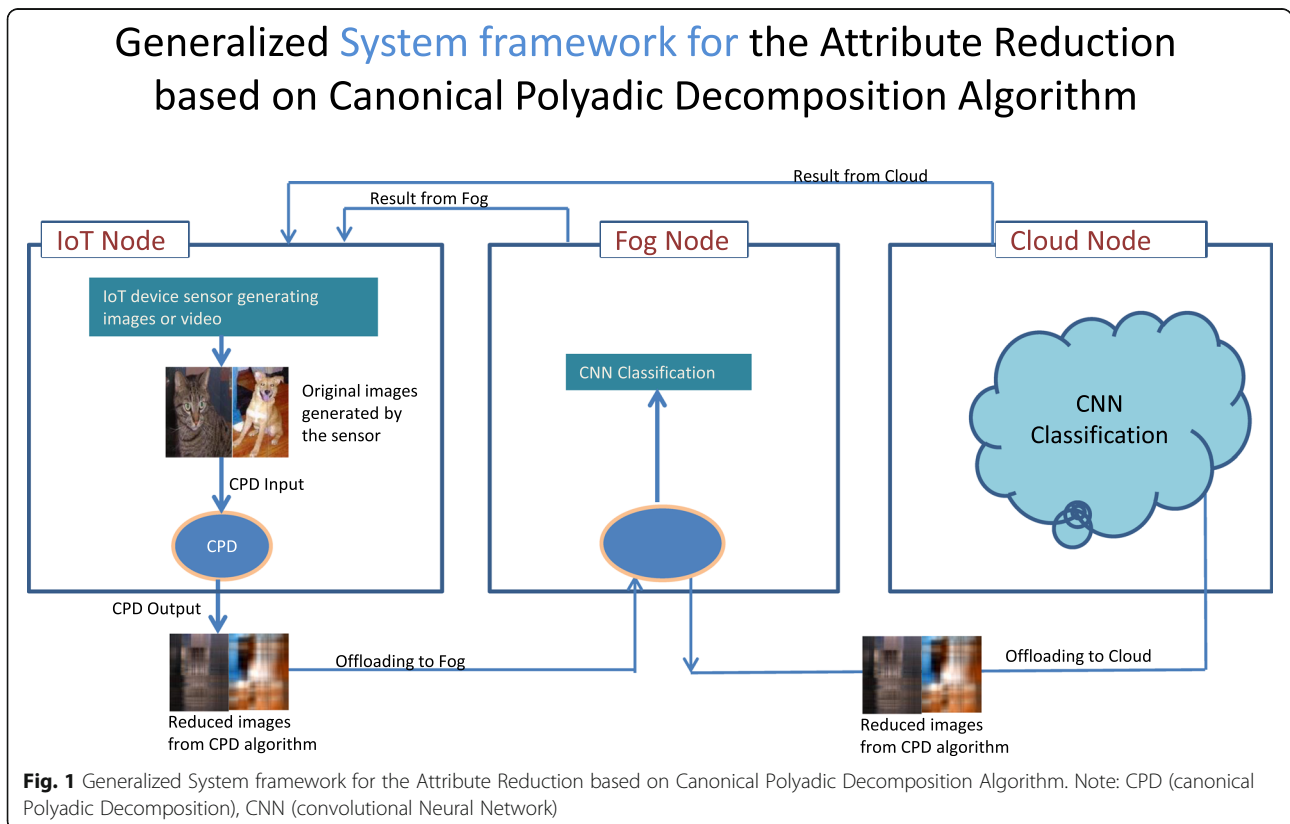
Computation cost of CPD

The computational cost of CPD is negligible because the IoT device will be sending the data as they are generated. Decomposing single image as they arrive without storing them to have a volume that will take time to be decomposed makes the decomposition time to be insignificant. Therefore, the infrastructure of IoT-Fog-Cloud computing will not significantly influence the efficiency of the proposed CPD algorithm.

Proposed method

In this research, we consider the infrastructure of IoT-fog-cloud computing designed to offer a distributed computing at the network as shown in Fig. 1. The focus of this study is to investigate the effect of canonical Polyadic decomposition as an attribute reduction technique and determine its applicability in IoT computation offloading to minimize network traffic and transmission time.

Attribute reduction results in data compression. There are different attribute reduction techniques and each has a different rate of data reduction. The focus in this research is to reduce the data size (data compression). We achieved this through canonical Polyadic decomposition which is an attribute reduction technique. CPD is suitable for this because it is capable of reducing the attribute for the purpose of offloading to a higher processing node and still regenerates the data's original format at the destination or point of analysis. Details of the CPD are presented in section 3.1. The CPD is applied to any offloadable data from IoT. The aim is to downsize the data and to reduce the amount of data that will be sent out from IoT device. The transmission time of a given task is given as the ratio of data size and available bandwidth [5]. When the data size is reduced through the proposed CPD method, the transmission time which is given as data-size/ bandwidth will be reduced. To

Fig. 1 Generalized System framework for the Attribute Reduction based on Canonical Polyadic Decomposition Algorithm. Note: CPD (canonical Polyadic Decomposition), CNN (convolutional Neural Network)

illustrate the CPD attribute reduction, Kaggle cat and dog dataset, which is an open sources dataset, was used. The data size of the dataset is reduced through CPD method and the size of the intermediate data is calculated. This intermediate result is the actual data size to be sent through the network. When the data gets to where it will be processed, it will be regenerated back to its original size and format. But for the purpose of this investigation, the research focused on:-.

(1). determining the reduction ratio of this method using eq. (5).
(2). using the intermediate data to regenerate the original image and use the regenerated image to do data classification.

The classification is done using Convolutional Neural Network (CNN) to determine the effect of applying CPD on the data. This effect is evaluated by comparing the classification result of the regenerated images with the classification results of the original images. The result of the reduction rate of the proposed method and the effect on classification accuracy are presented.

Therefore, this research proposes IoT Canonical Polyadic decomposition for deep learning (IoTCP_DL) algorithms and also proposes a novel Rank Accuracy Estimation model (RAEM) to determine the number of rank-1 (R) to be used in IoTCP_DL without affecting data accuracy. The step by step approaches to achieve the proposed methods are outlined in the subsequent subsections below.

IoT canonical Polyadic decomposition for deep learning (IoTCP_DL) algorithm

Given the IoT input object (say an image) $X_j \in \mathbb{R}^{h \times w \times c}$, the total attributes in each object will be h x w x c in the original tensor format. However, the data size can be reduced to $(h + w + c)$ R in the canonical Polyadic format as follow

$$X_j = \sum_{r=1}^{R} a_{jr}^{(h)} o\, a_{jr}^{(w)} \, o...o \, a_{jr}^{(c)}$$

$$= \left[\left[A^{(h)}, A^{(w)}, ..., A^{(c)} \right]\right] \tag{3}$$

Where o = outer product. Alternatively, each element X_j will be converted into canonical Polyadic format via eq. (4)

$$X_{j(h,w,c)} = \sum_{r=1}^{R} A_{jr}^{(h)}, A_{jr}^{(w)}, ..., A_{jr}^{(c)} \tag{4}$$

where $X_{j(h, w, c)}$ is image j with height (h), width (w) and colour channels (c). j = 1 to number of images. R is rank_1 and r is an instance of R.

In the experiment, the IoT data is processed using CPD as shown in eq. (4) at the IoT device. The result of this reduction technique is the intermediate data that should be transmitted as illustrated in Method_(IoT) of algorithm 1. In this study, the effect of attribute reduction is investigated and the reduction ratio is calculated as illustrated in Table 2, Fig. 3. After that, the intermediate data is regenerated to its original size and format at the destination and CNN classification is done to determine the accuracy of the classification. The result of the classification is compared with the classification result of the original images without CPD. The aim is to determine whether the application of CPD has any negative effect on the data. If not, it means that the reduced intermediate data from the CPD result will be a better option for transmission to the fog or cloud instead of offloading the heavy images as they are generated from the sensor/mobile devices. Eq. (5) and (6) calculate the total pixels ($X_{j(h, w, c)}$ *original*) of the original image and the reduced size ($X_{j(h, w, c)}$ *CP*) from the CPD method respectively.

$$X_{j(h,w,c)} \; original = h \times w \times c \tag{5}$$

$$X_{j(h,w,c)} \; CP = R(h + w + c) \tag{6}$$

The ratio of the output of CPD to the input data (original data) is calculated as in eq. (7).

$$\begin{aligned} \text{Reduction Ratio} &= \frac{X_{j(h,w,c)} \; CP}{X_{j(h,w,c)} \; original} \\ &= \frac{R(h + w + c)}{h \times w \times c}. \end{aligned} \tag{7}$$

where image j ranges from j = 1 to J. Eq. (7) shows the ratio of attribute reduction based on the Canonical Polyadic decomposition on the IoT data which will be transmitted over the network to the fog or cloud for classification. At the fog or cloud, the decomposed data will be regenerated to its original format via eq. (3) before Convolutional Neural Network will be applied to classify the object. The process of CPD data compression and conversion back to its original format at the destination (point of analysis) is illustrated in Algorithm 1.

The algorithm has three methods outlined as follow:

- Method (IoT) present what happed at the IoT layer, where the task is generated and the data size (ds) is reduced through CPD.
- Method (Fog) receives reduced data size (reduced ds) and checks if it is faster to process it at the fog or cloud. If it is faster at the fog, reduced ds is converted back to its original size and format and

processed by applying CNN to do classification on the images.

- Method (cloud), reduced ds is converted back to its original size and format and processed by applying CNN to do classification on the image data.

Algorithm 1: IoT Canonical Polyadic Decomposition for Deep Learning (IoTCP_DL) Algorithm.

Algorithm 1 illustrates the process of the proposed data reduction method. At the point of data generation, the method is applied to decompose the images into an intermediate form which is smaller in size. The intermediate data is no longer in the image form. The intermediate data will be converted back to its original form at the point of the analysis. The main purpose of the intermediate data is to reduce the volume of data that IoT devices will be offloading to the fog or cloud because of the limited bandwidth. At the destination, the intermediate data will be converted back to its original format. In this study, the 3 tier architecture is not simulated to illustrate the transmission. The study focused on the data reduction and whether the reduction affects the data accuracy.

Rank accuracy estimation model (RAEM)

Rank Accuracy Estimation Model is proposed in this study. The model represents the relationship between R rank-1 values and accuracy of the classification result. The proposed novel model estimates the R rank-1 that can give certain accuracy and reduce the data size to minimize the volume of data being offloaded from IoT device. The R rank-1 and accuracy (AC) follows a logarithmic model which we call Rank Accuracy Estimation model (RAEM) as in eq. (8)

$$AC = \alpha \ln(R) + \beta \qquad (8)$$

where α and β denote the constant used to estimate accuracy (AC) from any given R. The constant are problem-specific. This model determines the value of R to be used at IoT node to decompose the data being generated, such that the size will be reduced according to Fig. 5. The model will reduce the data size which indirectly will minimize transmission delay and the overall request delay without violating the accuracy of the information. With this model, the data size that will minimize total delay will be determined and the accuracy it will produce at the fog or cloud estimated. The model will be used in the scheduling policy to determine the value of R to be used at IoT to minimize the size of offloadable data. In this research, using Kaggle cat and dog dataset, we preprocessed each object and represented it as $\mathbb{R}^{270 \times 180 \times 3}$. With this, the values of α and β are 3.3968 and 63.06 respectively.

Experiment and evaluation

The IoTCP_DL algorithm is implemented with Mathlab. We carried out convolutional neural network classification of images which is implemented with Mathlab as well. To evaluate the performance of our proposed strategies, Kaggle cat and dog dataset dataset is used for the IoTCP_DL algorithm and subsequent classification analysis. Kaggle cat and dog dataset is an open-source dataset with 25, 000 images. First we used the dataset to do classification with Convolutional Neural Network. Then the dataset is processed for attribute reduction using the proposed method as presented in Section 3.3.1. We preprocessed and represented every object in the data set as $\mathbb{R}^{270 \times 180 \times 3}$ then applied the reduction method according to Eq. (4). Different values of R (R = 1, 2, 4, 8, 16, 32) are used to decompose the images based on the proposed algorithm and the reduced data size calculated using Eq. (6). The reduced data size and percentages of the reduced data to the original data size are shown in Table 2. The data reduction achieved as a result of the attribute reduction is compared with the benchmark result from convolutional layer data

Table 1 Experimental parameters

Parameter	Description	Value
Image height (h)	The height of the image in pixels	270
Image width (w)	The width of the image in pixels	180
Colour channels (c)	The number of colours used to read in the image. RGB is used	3
R	Number of rank-1 for CPD reduction	{1, 2, 4, 8, 16, 32}
Data size (ds)	Total pixels of an image	Calculated based on eq. (5)
Reduced-ds	Total pixels of an image after CPD reduction	Calculated based on eq. (6)

Table 2 Result of IoTCP_DL algorithm with different values of R Rank_1

R rank_1	Height (H)	Width (W)	Channels (C)	original size	CPD reduced size	Percentage of the original size generated by CPD algorithm
1	270	180	3	145,800	453	0.310699588
2	270	180	3	145,800	906	0.621399177
4	270	180	3	145,800	1812	1.242798354
8	270	180	3	145,800	3624	2.485596708
16	270	180	3	145,800	7248	4.971193416
32	270	180	3	145,800	14,496	9.942386831

reduction presented in [28]. To check for the effectiveness of the proposed algorithm, the original data classification result is compared with the classification results of different data regenerated from the intermediate CPD result of different R values. The aim of the comparison is to determine the effect of the data compression on the accuracy of the analysis done with the data, which passed through the proposed attribute reduction method. We presented the reduction ratio of our proposed system at different R rank-1. This shows how the data moving into the network can be reduced with different R rank-1 to decongest the network traffic. The process will reduce the delay experienced by the IoT devices in getting request responses especially for image and

video data requests that are needed in real-time applications. The decomposed data will be transmitted over the network for analysis. At the destination, the data will be re-converted to its original format and size as illustrated in algorithm 1. Table 1 show the parameters used in the experiment.

The image height, width and colour channels were measured so that the total pixels used to represent the image can be calculated. This total pixel represents the size of data the mobile device will be sending out for each image.

Results and discussion

In this section, the result of the proposed data reduction method and the result of the novel model for the
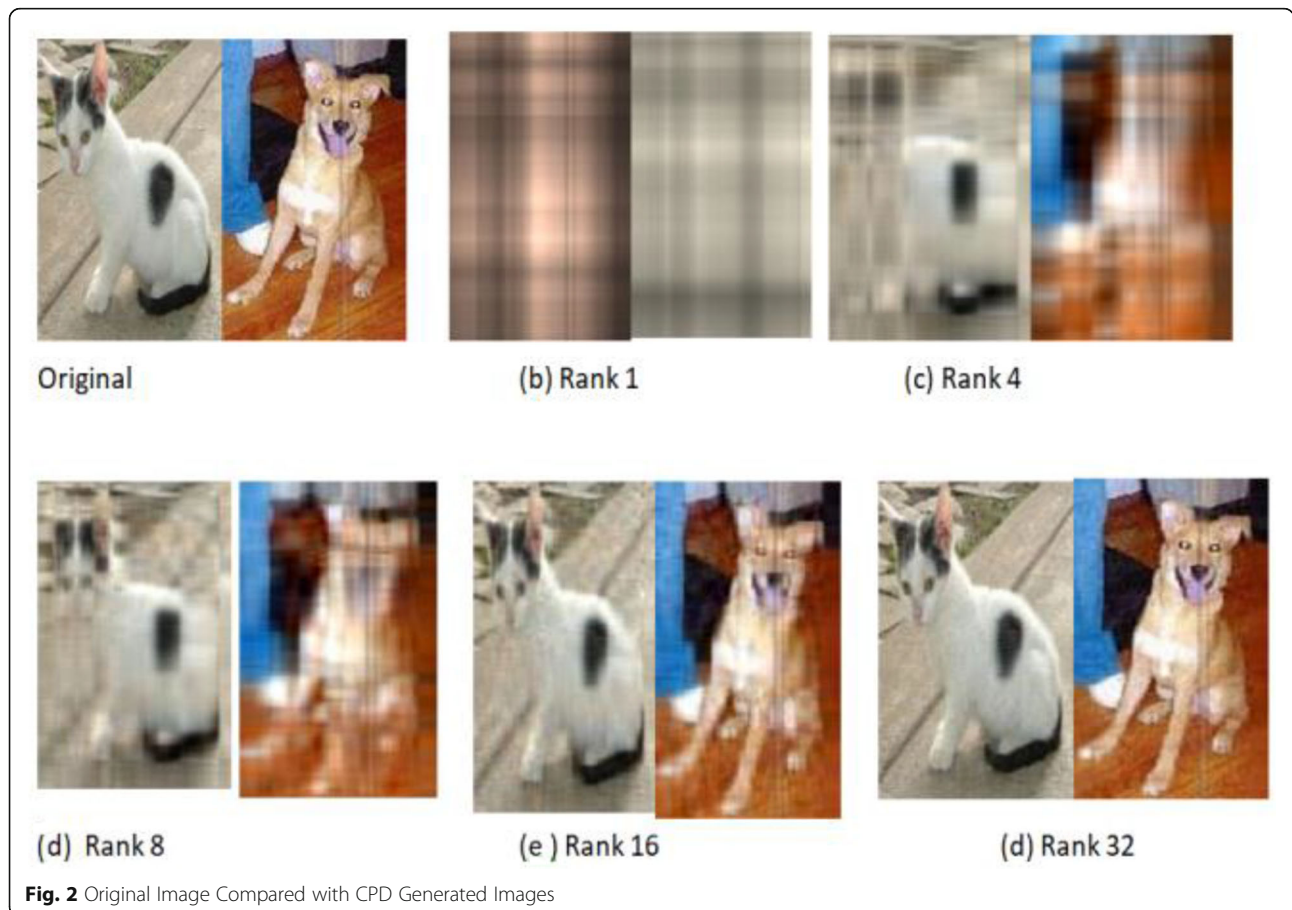

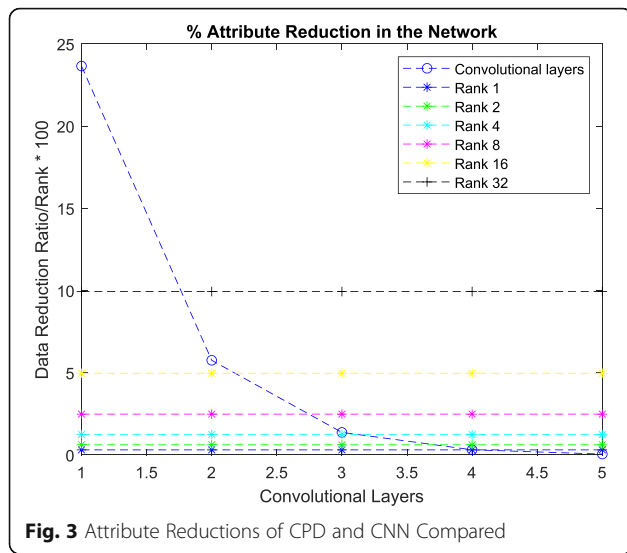**Fig. 2** Original Image Compared with CPD Generated Images

Fig. 3 Attribute Reductions of CPD and CNN Compared

estimation of R value are presented. First the data reduction results are presented followed by the classification accuracy of both the original images and the images processed through the reduction method. The comparison of the reduction rate and accuracy is also presented followed by the result of the RAEM that estimates the R value to be used in the proposed algorithm.

CPD data reduction by different R rank-1

The results of the IoTCP_DL algorithm with different values of R shows that the size of the intermediate data generated by the IoTCP_DL algorithm increases with the increase in the values of R. for instance, when R is 1 the size reduces from 145,800 to 453 while when R is 32, the size reduces from 145,800 to 14,496 as shown in Table 2. Likewise the percentage of the original size generated by the IoTCP_DL algorithm also increases as the R value increases. For example, when R is 2, the percentage of the original size achieved by the algorithm is 0.621% of the original size, while the when R is 16, the algorithm intermediate data size is 4.97% of the original size.

Comparison of attribute reduction of the proposed method with CNN layers data reduction

We compared the attribute reduction of our proposed method with the attribute reduction of CNN convolutional layers [28]. The attribute reduction by CPD is calculated as in Eq. (7) presented in Table 2. From Fig. 3, the data reduction reduces with an increase in the rank of the CPD. The lower the percentage of the reduced data, the lower the reduction rate by proposed method. At R rank_1 = 1, the reduced data is 0.31% of the original data while at R rank_1 = 32, the reduced data is 9.94% of the original data. This is far lower than the percentage reduction achieved by the first convolutional layer according to H. Li [28]. The image
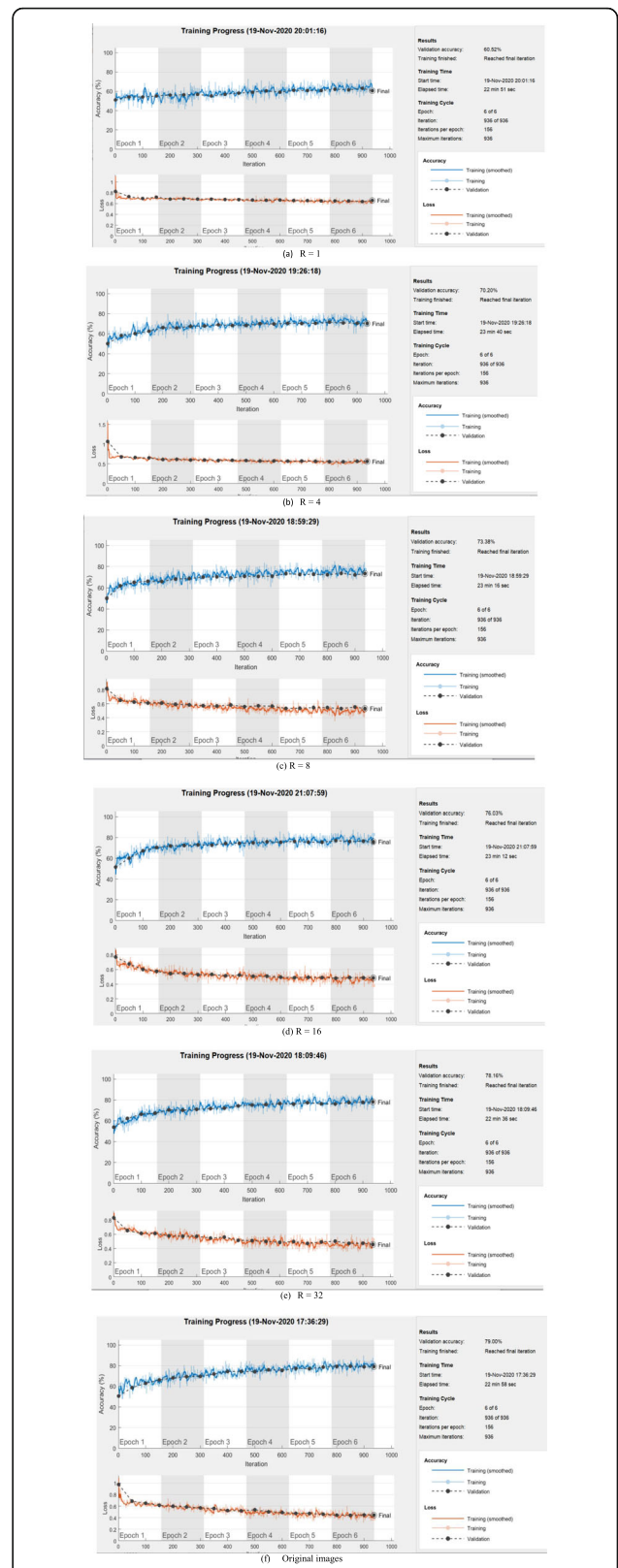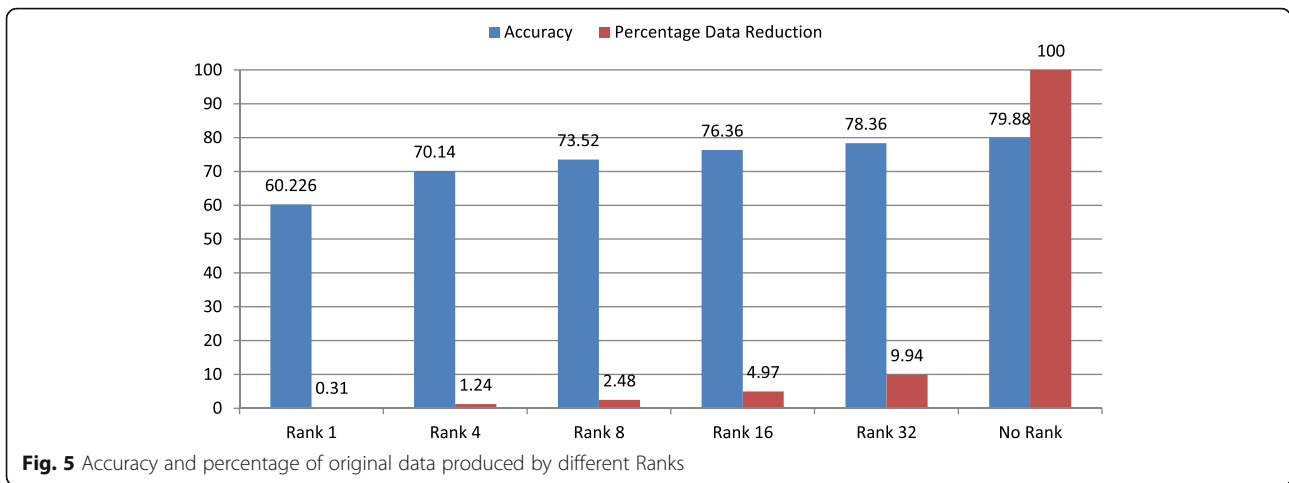

Fig. 4 Accuracy of the CNN classification result at R Rank-1 and Original images and R = 1, 4, 8, 16 and 32

Fig. 5 Accuracy and percentage of original data produced by different Ranks

generated by rank R = 1, 2, 4, 8, 16, and 32 is compared with the original image as illustrated in Fig. 2. The Fig. 2 shows the effect of attribute reduction on the images. Figure 3, shows that the attribute reduction of Convolutional layers increases with the increased number of convolutions. The lower the percentage of original data produced after convolutions, the higher the reduction effect. For instance, at first convolution, the original data is reduced to 24.5% while at three convolutions; it is reduced to 1.38% of the original data. But, at the IoT layer, it will not be suitable to implement multiple layers of convolutions (where it is possible to implement at all) because of storage, processing capability, and energy limitations. Because of such limitations at the IoT layer, implementing deep learning at the IoT level will delay the response time of the tasks for many complex and large data emanating from IoT devices.

Comparison of classification accuracy of the proposed method

To evaluate the proposed method in terms of the classification accuracy of data processed through the proposed attribute reduction method, we performed classification with CNN. This is to prove that the method can be applied in IoT to reduce the size of offloadable data in the network without affecting the accuracy of data significantly. We compared the accuracy of the original data and the accuracy at R = 1, 4, 8, 16, and 32. Figure 4 shows CNN classification result for R = 1, 4, 8, 16, 32 and the original images. The result shows that the data processed through CPD attribute reduction has little accuracy drop compared to the reduction made in the size of the data. Figure 5 shows the comparison of accuracy at different ranks and the percentage of original data produced by the rank. Figure 5 shows that as the R value increases, the reduced data size (output of the reduction method) increases and the classification accuracy increases. Figure 6 shows the relationship between accuracy (AC) and R value used for the data decomposition.

The difference between the accuracy achieved from the experimental values of different rank-1 and the accuracy calculated from our proposed model is illustrated in Table 3. The results shows that our model can be used to predict the accuracy of any R value to be used at IoT level since their differences are very small in each case.
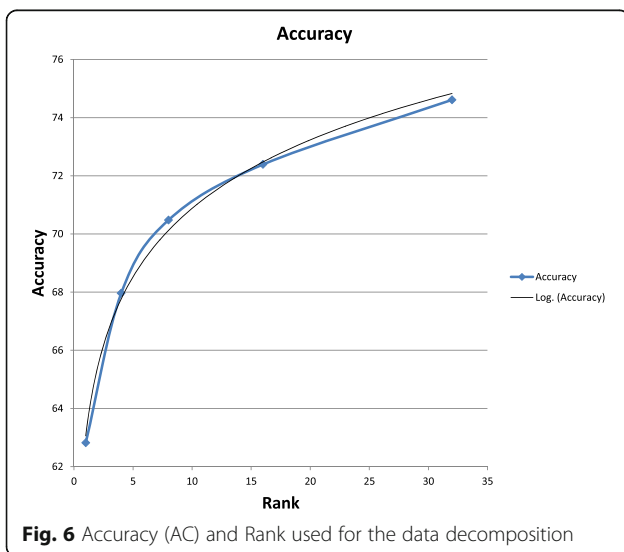

Fig. 6 Accuracy (AC) and Rank used for the data decomposition

Table 3 Comparison between the experimental values and RAEM values

Rank	Experimental Values	RAEM Values	Difference
1	62.82	63.06	−0.24
4	67.96	67.76896	0.19
8	70.478	70.12345	0.35
16	72.39	72.47793	−0.08
32	74.612	74.83241	−0.22

From Table 3, the minimum difference is at $R = 1$ where the difference is -0.24 and the maximum is at $R = 8$ where the difference is 0.35. Therefore, we can reliably say that our model can predict the accuracy of R to be produced with the error at ±0.35 in the accuracy.

Conclusions

The problem of heavy data emanating from the avalanche of interconnected IoT in a distributed computing is addressed. The paper proposed IoT Canonical Polyadic Decomposition for Deep Learning Algorithm in Fog computing architecture to reduce the size of IoT offloadable data. RAEM was modeled for determining the value of R to be applied at the IoT level without violating the accuracy required of the data. The result proved that the proposed attribute reduction method is better than the state of the art. Secondly, the proposed RAEM model can be reliable to determine R-value for the required accuracy that can reduce the data size to less than 10 % of the original size. The proposed CPD based data reduction is compared with the data reduction of the convolutional layer and the proposed method performed better than convolutional layers attribute reduction in terms of reducing size of offloadable data in the network.

Future work

In our future work, we plan to simulate the 3-tier user device (IoT), fog and cloud network and propose a scheduling algorithm for deep learning based on data reduction. Using RAEM model, the scheduler will determine the R-value that will give the best QoS based on the characteristics of a given task. The characteristics of the task include but not limited to the latency, response time, and computational requirement.

Authors' contributions
Conceptualization, Nweso Emmanuel Nwogbaga; Data curation, Nweso Emmanuel Nwogbaga; Formal analysis, Nweso Emmanuel Nwogbaga; Funding acquisition, Rohaya Binti Latip and Amir Rizaan Abdul Rahiman; Investigation, Nweso Emmanuel Nwogbaga; Methodology, Nweso Emmanuel Nwogbaga; Project administration, Rohaya Binti Latip and Lilly Suriani Affendey; Resources, Lilly Suriani Affendey and Amir Rizaan Abdul Rahiman; Software, Nweso Emmanuel Nwogbaga; Supervision, Rohaya Binti Latip, Lilly Suriani Affendey and Amir Rizaan Abdul Rahiman; Writing – original draft, Nweso Emmanuel Nwogbaga; Writing – review & editing, Nweso Emmanuel Nwogbaga, Rohaya Binti Latip, Lilly Suriani Affendey and Amir Rizaan Abdul Rahiman.

Declarations

Author details
[1]Department of Communication Technology and Network, Universiti Putra Malaysia, Seri Kembangan, Malaysia. [2]Department of Computer Science, Faculty of Science, Ebonyi State University, Abakaliki, Nigeria. [3]Department of Computer Science, Faculty of Computer Science, Universiti Putra Malaysia, Seri Kembangan, Malaysia.

References
1. Aggarwal A (2020) Self adaptive fruit fly algorithm for multiple workflow scheduling in cloud computing environment. Kybernetes. https://doi.org/10.1108/K-11-2019-0757 ahead-of-print, ahead-of-print
2. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun Survey Tutorials 17(4):2347–2376. https://doi.org/10.1109/COMST.2015.2444095
3. Al-khafajiy M, Baker T, Waraich A, Al-jumeily D, Hussain A (2018) IoT- F og optimal workload via fog offloading. In: 2018 IEEE/ACM international conference on utility and cloud computing companion (UCC companion), pp 359–364. https://doi.org/10.1109/UCC-Companion.2018.00081
4. Alam KM, Saini M, El Saddik A (2015) Toward social internet of vehicles: concept, architecture, and applications. IEEE Access 3:343–357. https://doi.org/10.1109/ACCESS.2015.2416657
5. Alli AA, Alam MM (2019) SecOFF-FCIoT: machine learning based secure offloading in fog-cloud of things for smart city applications. Int Things 7(2019):100070. https://doi.org/10.1016/j.iot.2019.100070
6. Asghari S, Navimipour NJ, Club E, Branch T, Branch T, Expressway P (2019) A cost-aware method for cloud services composition using a hybrid algorithm. J Soft Comput Inform Technol (JSCIT) 8(2):17–26
7. Battaglino C, Ballard G, Kolda TG (2018) A practical randomized CP tensor decomposition. SIAM J Matrix Analysis Appl 39(2):876–901. https://doi.org/10.1137/17M1112303
8. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. MCC'12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop, (august 2012), pp 13–15. https://doi.org/10.1145/2342509.2342513
9. Chen Z, Xiao N, Han D (2020) Multilevel task offloading and resource optimization of edge computing networks considering UAV relay and green energy. Applied Sciences (Switzerland)
10. Cichocki A (2014) Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions, pp 1–30 doi: abs/1403.2048
11. Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: principles, architectures, and applications. Internet of Things: Principles and Paradigms, pp 61–75. https://doi.org/10.1016/B978-0-12-805395-9.00004-6
12. Deng R, Liang H (2018) Whether to charge or discharge an electric vehicle? An optimal approach in polynomial time. IEEE Vehicular Technology Conference, 2017-Septe (September 2017), pp 1–5. https://doi.org/10.1109/VTCFall.2017.8288324
13. Derhab A, Belaoued M, Guerroumi M, Khan FA (2020) Two-Factor mutual authentication offloading for Mobile cloud computing. IEEE Access 8:28956–28969. https://doi.org/10.1109/ACCESS.2020.2971024
14. Ebadi, Y., & Navimipour, N. J. (2018). An energy-aware method for data replication in the cloud environments using a T abu search and particle swarm optimization algorithm. Concurrency and Computation: Practice and Experience, (July 2017), 1–10. doi: https://doi.org/10.1002/cpe.4757, 31, 1
15. Eze EC, Sijing Z, Liu E, Nwogbaga NE, Eze JC (2016a) Timely and reliable packets delivery over internet of vehicles (IoVs) for road accidents prevention : a cross-layer approach. IET Networks 5(5):127–135. https://doi.org/10.1049/iet-net.2015.0112
16. Eze EC, Zhang S, Liu E, Nwogbaga NE, Eze JC (2016b) RECMAC: reliable and efficient cooperative cross-layer MAC scheme for vehicular communication based on random network coding technique. In: 2016 22nd International Conference on Automation and Computing, ICAC 2016: Tackling the New

Challenges in Automation and Computing, (September), pp 342–347. https://doi.org/10.1109/IConAC.2016.7604943

17. Feng J, Liu Z, Wu C, Ji Y (2017) AVE: autonomous vehicular edge computing framework with ACO-based scheduling. IEEE Trans Veh Technol 66(12):10660–10675. https://doi.org/10.1109/TVT.2017.2714704

18. Goudarzi P (2020) Joint customer / provider evolutionary multi-objective utility maximization in cloud data center networks. Iran J Sci Technol Transact Electrical Eng 0123456789(2):479–492. https://doi.org/10.1007/s40998-020-00381-x

19. Harshman RA (1970) Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis. UCLA Working Papers Phonetics 16(4):1–84. https://doi.org/10.1134/S0036023613040165

20. Hitchcock FL (1927) Multiple invariants and generalized rank of a P-way matrix or tensor. J Math Phys 7(1–4):39–79. https://doi.org/10.1002/sapm19287139

21. Jaddoa A, Sakellari G, Panaousis E, Loukas G (2019) Dynamic decision support for resource offloading in heterogeneous internet of things environments

22. Jahantigh MN, Rahmani AM, Navimirour NJ, Rezaee A (2019) Integration of internet of things and cloud computing : a systematic survey. IET Commun 14(2):165–176. https://doi.org/10.1049/iet-com.2019.0537

23. Klaine PV, Imran MA, Onireti O, Souza RD (2017) A survey of machine learning techniques applied to self-organizing cellular networks. IEEE Commun Surveys Tutorials 19(4):2392–2431. https://doi.org/10.1109/COMST.2017.2727878

24. Kong L, Khan MK, Wu F, Chen G, Zeng P (2017) Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: overview, design, and challenges. IEEE Commun Mag 55(1):62–68. https://doi.org/10.1109/MCOM.2017.1600422CM

25. Kotb Y, Al Ridhawi I, Aloqaily M, Baker T, Jararweh Y, Tawfik H (2019) Cloud-based multi-agent cooperation for IoT devices using workflow-nets. J Grid Comput 17(4):625–650. https://doi.org/10.1007/s10723-019-09485-z

26. Kumar N, Rodrigues JJPC, Chilamkurti N (2014) Bayesian coalition game as-A-Service for Content Distribution in internet of vehicles. IEEE Internet Things J 1(6):544–555. https://doi.org/10.1109/JIOT.2014.2374606

27. Li G, Yan J, Chen L, Wu J, Lin Q, Zhang Y (2019b) Energy consumption optimization with a delay threshold in cloud-fog cooperation computing. IEEE Access 7:159688–159697. https://doi.org/10.1109/ACCESS.2019.2950443

28. Li H, Ota K, Dong M (2018) Learning IoT in edge: deep learning for the internet of things with edge computing. IEEE Netw 32(1):96–101. https://doi.org/10.1109/MNET.2018.1700202

29. Li G, Yan J, Chen L, Wu J, Lin Q, Zhang Y (2021). Energy Consumption Optimization With a Delay Threshold in Cloud-Fog Cooperation Computing. IEEE Access, 7. p. 159688–97. https://doi.org/10.1109/ACCESS.2019.2950443

30. Li J, Gao H, Lv T, Lu Y (2018a) Deep reinforcement learning based computation offloading and resource allocation for MEC. IEEE Wireless Communications and Networking Conference, WCNC, 2018-April, pp 1–6. https://doi.org/10.1109/WCNC.2018.8377343

31. Li L, Guo M, Ma L, Mao H, Guan Q (2019a) Online workload allocation via fog-fog-cloud cooperation to reduce IoT task service delay. Sensors (Basel, Switzerland) 19(18). https://doi.org/10.3390/s19183830

32. Liu L, Chang Z, Guo X, Mao S, Ristaniemi T (2018) Multiobjective optimization for computation offloading in fog computing. IEEE Internet Things J 5(1):283–294. https://doi.org/10.1109/JIOT.2017.2780236

33. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surveys Tutorials 19(3):1628–1656. https://doi.org/10.1109/COMST.2017.2682318

34. Mahmud R, Kotagiri R, Buyya R (2018) Fog computing: a taxonomy. Survey Future Direct:103–130. https://doi.org/10.1007/978-981-10-5861-5_5

35. Manasrah AM, Ali HB (2018) Workflow scheduling using hybrid GA-PSO algorithm in cloud computing. Wirel Commun Mob Comput 2018:1–16. https://doi.org/10.1155/2018/1934784

36. Meurisch C, Gedeon J, Nguyen TAB, Kaup F, Mühlhäuser M (2017) Decision support for computational offloading by probing unknown services. 2017 26th International Conference on Computer Communications and Networks, ICCCN 2017. https://doi.org/10.1109/ICCCN.2017.8038406

37. Min M, Xiao L, Chen Y, Cheng P, Wu D, Zhuang W (2019) Learning-based computation offloading for IoT devices with energy harvesting. IEEE Trans Veh Technol 68(2):1930–1941. https://doi.org/10.1109/TVT.2018.2890685

38. Name HAM, Oladipo FO, Ariwa E (2017) User mobility and resource scheduling and management in fog computing to support IoT devices. In:

39. Naseri A, Jafari Navimipour N (2019) A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm. J Ambient Intell Humaniz Comput 10(5):1851–1864. https://doi.org/10.1007/s12652-018-0773-8

40. Ning Z, Dong P, Wang X, Rodrigues JJPC, Xia F (2019) Deep reinforcement learning for vehicular edge computing: an intelligent offloading system. ACM Trans Intell Syst Technol 10(6):1–24. https://doi.org/10.1145/3317572

41. Nwogbaga NE (2020) A review of big data clustering methods and research issues. Int J Sci Res (IJSR) 9(5):253–264. https://doi.org/10.21275/SR20502183559

42. Sarkar S, Misra S (2016) Theoretical modelling of fog computing: a green computing paradigm to support IoT applications. IET Networks 5(2):23–29. https://doi.org/10.1049/iet-net.2015.0034

43. Shakarami A, Ghobaei-Arani M, Masdari M, Hosseinzadeh M (2020) A survey on the computation offloading approaches in Mobile edge/cloud computing environment: a stochastic-based perspective. J Grid Comput 18(4):639–671. https://doi.org/10.1007/s10723-020-09530-2

44. Sharma G (2018) A lightweight user authentication scheme for cloud-IoT based healthcare services. Iran J Sci Technol Transact Electrical Eng 43(S1):619–636. https://doi.org/10.1007/s40998-018-0146-5

45. Sujana JAJ, Revathi T, Rajanayagam SJ, Jennifa JA, Revathi T, Fuzzy-, S. J. R. (2018) Fuzzy-based security-driven optimistic scheduling of scientific workflows in cloud computing Fuzzy-based security-driven optimistic scheduling of scientific workflows in. IETE J Res 66(2):1–18. https://doi.org/10.1080/03772063.2018.1486740

46. Tien CM, Lee CJ, Cheng PW, Lin YD (2005) SOAP request scheduling for differentiated quality of service. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 3807 LNCS, pp 63–72. https://doi.org/10.1007/11581116_7

47. Tuli, S., Basumatary, N., & Buyya, R. (2019). EdgeLens: deep learning based object detection in integrated IoT, Fog and Cloud Computing Environments. Cloud computing. Retrieved from http://arxiv.org/abs/1906.11056

48. Vivekrabinson K, Muneeswaran K (2021) Fault-tolerant based group key servers with enhancement of utilizing the contributory server for cloud storage applications. IETE J Res 1–16.

49. Wan S, Zhao Y, Wang T, Gu Z, Abbasi QH, Choo KKR (2019) Multi-dimensional data indexing and range query processing via Voronoi diagram for internet of things. Futur Gener Comput Syst 91(September 2018):382–391. https://doi.org/10.1016/j.future.2018.08.007

50. Wang C, He Y, Yu FR, Chen Q, Tang L (2018) Integration of networking, caching, and computing in wireless systems: a survey, some research issues, and challenges. IEEE Commun Surveys Tutorials 20(1):7–38. https://doi.org/10.1109/COMST.2017.2758763

51. Wang X, Yang LT, Liu H, Deen MJ (2017) A big data-as-a-service framework: state-of-the-art and perspectives. IEEE Transactions Big Data 4(3):325–340. https://doi.org/10.1109/tbdata.2017.2757942

52. Xiao L, Lu X, Xu T, Wan X, Ji W, Zhang Y (2020) Reinforcement learning-based Mobile offloading for edge computing against jamming and interference. IEEE Trans Commun 68(10):6114–6126. https://doi.org/10.1109/TCOMM.2020.3007742

53. Xu X, Xue Y, Qi L, Yuan Y, Zhang X, Umer T, Wan S (2019) An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles. Futur Gener Comput Syst 96(March):89–100. https://doi.org/10.1016/j.future.2019.01.012

54. Xu Z, Zhao L, Liang W, Rana OF, Zhou P, Xia Q, Xu W, Wu G (2021) Energy-aware inference offloading for DNN-driven applications in Mobile edge clouds. IEEE Transact Parallel Distributed Syst 32(4):799–814. https://doi.org/10.1109/TPDS.2020.3032443

55. Cheng Y-M (2021) Can tasks and learning be balanced? A dual-pathway model of cloud-based e-learning continuance intention and performance outcomes. Kybernetes. https://doi.org/10.1108/K-07-2020-0440

56. Zhang Q, Yang LT, Chen Z, Li P (2018) High-order possibilistic c-means algorithms based on tensor decompositions for big data in IoT. Inform Fusion 39:72–80. https://doi.org/10.1016/j.inffus.2017.04.002

57. Zhao J, Li Q, Gong Y, Zhang K (2019) Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in 68(8):7944–7956

Publisher's Note