Peer-to-Peer Networking and Applications (2021) 14:2340–2350
https://doi.org/10.1007/s12083-020-00924-3

SRAF: Scalable Resource Allocation Framework using Machine Learning in user-Centric Internet of Things

Zafer Al-Makhadmeh [1] · Amr Tolba [1,2]

Received: 17 February 2020 / Accepted: 28 April 2020 / Published online: 21 June 2020

Abstract
Internet of Things (IoT) design focuses on concurrently handling multiple tasks for improving the scalability and robustness of the information sharing platform. Therefore, sophisticated resource allocation and optimization methods are necessary to prevent backlogs in request processing and resource allocation. This paper introduces a scalable resource allocation framework that is designed to maximize the service reliability in IoT because of a large volume of tasks and information. In this process, deep learning is used to assist the effective and scalable framework in allocating the resources to tasks with respective time constraints. The assisted allocation through deep learning balances the density of users, requests, and available resources without replications and overloading. Thus, the proposed deep learning based resource allocation framework helps in reducing the waiting and processing times of the requests under a controlled response time. Besides, the optimal segregation of available resources and request density facilitates failure-less allocation.

Keywords IoT · Machine learning · Request processing · Resource allocation

1 Introduction

Internet of Things (IoT) design facilitates visualization and representation of unprocessed data in digital format. This digital visualization provides easy access to heterogeneous devices, ranging from small sensors to large cloud systems [1, 2]. The design goal of IoT is to provide pervasive access to resources and devices in a distributed communication platform without the need for additional infrastructure or computation units. For this purpose, the common Internet platform is used by the connected devices and service providers [3, 4].

The IoT platform consists of heterogeneous devices, and the communication technologies vary from Zigbee to WiMAX, a wireless local area network (WLAN) [5, 6]. This heterogeneous communication platform is interoperable across all devices that share common computation capabilities and storage [7, 8]. Resource allocation and sharing is preceded with the help of content dissemination and centralized cloud servers through infrastructure and other gateway devices [9, 10]. The fundamental process is the pervasive request processing by the cloud servers for allocating resources to the end-users [11, 12]. Time-constrained request processing and resource allocation improves the quality of service (QoS) and experience of the users over various applications [13].

Resource allocation in IoT is a challenging and demanding task because of its distributed nature, and the necessity for timely access. The interconnection between distributed systems through heterogeneous connectivity and distinct applications increases the demand for available resources [14]. Besides, the allocation interval and response, along with the processing time, are some other QoS constraints when determining the efficiency of an IoT-based system [15].

The primary task in a IoT-coupled resource allocation process is its support for interoperability, along with shared access and service response without delay [16]. Therefore, a resource allocating IoT environment has to balance between the available resources, allocation time interval, and user device requests for improving the QoS of the application [17, 18]. This aids in meeting the user requirements and in-time processing of requests to prevent allocation failures. The

This article is part of the Topical Collection: *Special Issue on Network In Box, Architecture, Networking and Applications*
Guest Editor: Ching-Hsien Hsu

✉ Zafer Al-Makhadmeh
zalmakhadmee@ksu.edu.sa

[1] Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia

[2] Mathematics and Computer Science Department, Faculty of Science, Menoufia University, Shebin El-Kom 32511, Egypt

resource allocation and request processing are balanced for any number of devices to meet the application/ user requirements [19, 20]. Moreover, machine learning techniques are used to select a task from the list of tasks, which helps in reducing the resource allocation complexity. A number of techniques such as neural networks, k-nearest neighbor, support vector machine, and deep neural network are used to provide resources for their respective tasks. These intelligent learning techniques improve the resource allocation process by optimizing the learning process. Therefore, this study utilizes a deep learning technique to allocate resources by overcoming time and computation complexities. The main contributions of the study are as follows:

- To allocate optimal resources for a task by applying deep learning with a scalable resource allocation framework
- To minimize time and computation complexities while allocating resources
- Allocating a resource by resolving its replication and overloading issues

The rest of the manuscript is organized as follows: Section 2 discusses various opinions regarding the resource allocation process. Section 3 analyzes the proposed scalable resource allocation framework (SRAF) with a resource allocation process based on deep learning, and Section 4 evaluates its efficiency. Finally, Section 5 provides the conclusions.

2 Related Works

This section discusses different opinions regarding the resource allocation process. Abedin et al. [21] introduced an effective QoS, formulated joint user association, and resolved the resource allocation problem using analytic hierarchy process (AHP)-matching. AHP-matching examines the priority of QoS requirements in heterogeneous applications. After identifying the priority, an association between an IoT device and fog infrastructure is established. This association helps in minimizing the resource allocation problem by selecting the best resource from a collection of resources. It consists of QoS requirements imposed by ultra-reliable low latency communication (URLLC) and enhanced Mobile Broadband (eMBB) services. Owing to the computation of quality constraints, the resources are allocated easily and effectively. In addition, this also maintains the reliability and scalability of resource allocation.

Mergenci and Korpeoglu [22] have discussed generic resource allocation for heterogeneous cloud infrastructure. They propose two metrics for reflecting the current state of a virtual machine. Thus, their proposed method uses a multi-dimensional resource allocation heuristic algorithm.

Nassar and Yilmaz [23] designed reinforcement learning for resource allocation in fog radio access networks (F-RANs). The limited resources are allocated to IoT applications. For each access, a fog network (FN) decides whether to serve the request from an IoT user locally at the edge by utilizing its own resources, or to refer it to the cloud and conserve its valuable resources for future users with a potentially higher utility to the system.

Efficient resource allocation for the uplink transmission of wireless IoT networks was proposed by Liu et al. [24]. In this study, an efficient channel allocation algorithm (ECAA) of low complexity was designed for user grouping. Then, a Markov decision process (MDP) model was used for unpredicted energy arrival and channel condition uncertainty from each user.

Li et al. [25] proposed an edge-cloud-assisted IoT. They designed an iterative double-sided auction scheme (DSAS) for computing resource trading. Here, the brokers solve an allocation problem, and design a specific price rule for the buyers and sellers of a computing resource to truthfully submit bids. Thus, the proposed system is able to assist different tasks and provide the resources without creating any complexity. However, regular update of price rules is difficult. Nevertheless, this DSAS system is able to manage compatibility, budge balance, and individual rationality.

Li et al. [26] introduced a fog computing node with IoT (FN-IoT) by collecting a large amount of data to make reliable offloading decisions. It transfers the data to the fog computing nodes, thereby supporting a large amount of data with low latency and limited resources. The deployment of non-orthogonal multiple access (NOMA) in an IoT network is used to transmit the data to the same FN in the same time and code domain. The NP-hardness in this process is resolved by applying an improved genetic algorithm. However, an intermediate access may change the task representation because of multiple access, creating difficulties in provisioning of resources. Duplications may occur, thereby reducing the entire system's performance.

A data-driven resource allocation for NFV-based IoT was proposed by Tian et al. [27]. This synthetic approach is based on examining both network processing procedures and stationary users' behaviors. Then, a matrix mapping based dynamic resource allocation mechanism is modeled for the virtualized core mobile networks.

Ramezani et al. [28] introduced a single wireless-powered relay for multiple users. An energy-constrained relay assists the information transmission from a number of IoT devices to the access point (AP) using WPC. It maximizes the total network throughput by optimizing the wireless energy transfer (WET) duration and the relay's energy expenditure in each time slot together.

Wireless-powered IoT networks with short packet communication has been proposed by Chen et al. [29]. An effective throughput and an effective amount of information are used to manage the transmission rate and packet error rate (PER). It

cooperates with the optimized transmission time and PER of each user to maximize the total effective throughput or minimize the total transmission time, subject to individual user's information requirements.

Aazam et al. [30] addressed 5G tactile industrial fog computing. The tactile Internet has its own use-cases across a number of application domains, the industrial sector being one of the most popular among them. The objective is the quality of experience (QoE)-awareness for dynamic resource allocation in a tactile IoT application.

Dai et al. [31] presented a game-theoretic approach for QoE-driven, 5G enabled IoT. They introduced an allocation channel problem for the IoT uplink communication in a 5G network. A mean opinion score (MOS) function of transmission delay was used to measure QoE of each smart object.

Gao et al. [32] introduced the expansion of data based on wisdom architecture as an organized approach for modeling both entities and relationship elements. This approach focuses on accessing and processing resources for security protection by exploiting the cost variation in both types of resource conversions and traversals. Here, resources are allocated to a task based on their characteristics, reliability, and scalability. Therefore, this study proposes an effective, scalable, and reliable framework using deep learning techniques. The introduced techniques not only resolve the above-discussed problem but also attain the mentioned contributions. Here, references [21, 25, 26] are chosen for comparison because of their use of massive amount of data, along with reliable resource allocations and scalable allocation processes. Based on the statistical survey, the proposed deep learning based resource allocation framework helps in reducing the waiting and processing times of the requests under a controlled response time. Besides, the optimal segregation of available resources and request density facilitates failure-less allocation. The proposed system is details discussed in the next section as follows,

3 SRAF

An SRAF integrates the user requests and available resources to meet the user requirements. This framework jointly operates in the cloud and IoT layer for improving the resource allocation rate. The concise management of available resources, user requests and allocation lag is aided through deep learning. This deep learning paradigm is responsible for retaining the liveliness of requests and in-time allocation of resources. Therefore, the proposed framework is modeled in three phases: request mapping, resource allocation, and time lag optimization. Here, the interconnection between heterogeneous devices and long-range access support of resources are exploited for improving performance. Besides, the machine learning process for resource allocation and lag optimization is augmented to retain the quality of response. In the following subsections, the three phases will be explained briefly. The proposed SRAF is shown in Fig. 1.

Figure 1 represents that the SRAF structure for allocating the resources according to the user request. The data is collected with the help of IoT devices, which are collected using fog nodes, and stored in the cloud data center. Based on the user request, the requests are mapping with relevant data which is performed by passing the instruction via the data controller and wireless transmission. According to the request, the resources are allocated by performing the resource mapping. The detailed explanation of the Scalable Resource Allocation Framework based resource allocation process is discussed in the following section.

Generally, Fog nodes have an extreme virtualization feature. Here, Each Fog node may be made up of one or more devices and therefore build a virtual network to serve the region of coverage in accordance with base station. Such machines may be routers, switches, gateways or the central base stations where controls run and managed using controllers. IoT Edge is connected to a Fog Computing Layer that is connected in turn to the centralized Cloud Computing Layer, which enables optimized resource, mapping and request of resources. That kind of relation forms a network framework for hierarchical computing architecture.

3.1 Resource Allocation Problem

Resource allocation problem is based on the joint optimization of n IoT users requesting $\Re$ resources that are available with M service providers. Further, the time and type of resource requested by the users depend on the application platform. Here, Based on each resource and service provider as named as $\{r_1, r_2, \ldots r_n\}\Re, m \in M$, The n IoT users accommodated and serviced at a time t_s need not be the same. However, the service provider $M = \{1, 2, \ldots, m\}$ verify the availability of resources $\Re = \{r_1, r_2, \ldots, r_n\}$ at the request time t_r, which is first processed by service provider M to verify if $\{r_1, r_2, \ldots r_n\} : \rightarrow \{t_{s_1}, t_{s_2}, \ldots t_{s_n}\}$, that is, the resources $\{r_1, r_2, \ldots r_n\}$are allocated at the service time $\{t_{s_1}, t_{s_2}, \ldots t_{s_n}\}$. Let ρ_a and ρ_r denote the allocation of R to n and the total R allocated to n in t_s. The allocation function for a user is given as $2^R \rightarrow t_s \forall R \in M$ and $R : \rightarrow t_s$. With reference to the allocation function, the resource allocation problem is framed as follows:

$$\max_{\rho_a, \gamma} \left(\sum_{i=1}^{n} \rho_{r_i} \rho_{a_i} - \sum_{j=1}^{M} \sum_{k=1}^{R} \rho_{r_i} \times \frac{k}{j} \sum_{i=1}^{n} \gamma_i \frac{k}{j} \right) \tag{1a}$$

$$\rho_{a_i} = \begin{cases} 0, if\ R : \nrightarrow t_s \\ 1, if\ R : \rightarrow t_s \end{cases} \tag{1b}$$

$$\sum_{j=1}^{M} \gamma_{ij\frac{k}{j}} = r_{i\frac{k}{j}} \rho_{a_i}, \forall i \in R, \forall k \in M \tag{1c}$$

$$\sum_{i=1}^{n} \gamma_{ij\frac{k}{j}} \leq \frac{r_i}{t_{r_i}}, \forall i \in n\ and\ k \in M\ and\ j \in R \tag{1d}$$

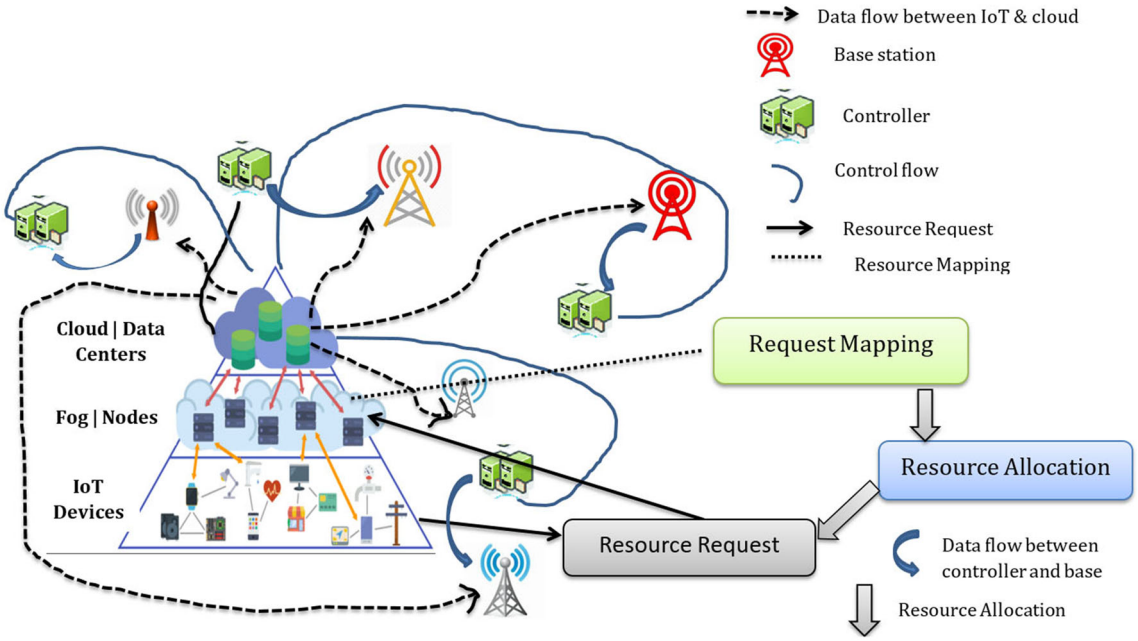

Fig. 1 SRAF structure

In the above formulations, the requirement for resource allocation is defined, where $\gamma = \frac{t_s}{t_r}$. In Equation (1a), the maximization of the $\rho_a = 1$ resource allocation from M to all n in t_s is expected. The conditions in Equations (1b), 1(c), and 1(d) are designed to ensure that the R is high, the resources are mapped for appropriate t_r and the ratio of serviced requests is high. From the above problem formulation, maximizing γ_{ij} and r_i/t_{r_i} for $\rho_{a_i} = 1$ helps achieve optimal resource allocation. The design of the proposed framework considers the above-mentioned constraints. The framework design focuses on improving γ and $\frac{r_i}{t_r}$ for the available R, to ensure $\rho_r = 1$ for n in time $(t_r - t_s)$.

3.2 Request Mapping

In the request mapping phase, the service measures for the input requests at t_r from n are handled and allocated to an active resource. In the request mapping process, the new request has to wait for a time t_a after the existing request, where t_a is the time for allocation of the resource. Similarly, the processing time (t_p) of M needs to be considered when effectively determining the wait time; therefore, $t_w = t_p = s + \left(\frac{count\ \rho_a}{t_a}\right)$ is the required time for the next request to be allocated to the resource. When $M < n$, the incoming requests are queued in the IoT buffer for assigning M with a t_w. As formulated, the wait time of the new request is defined as the processing time for the existing request, and this depends on the processing speed (s) of M. Allocation failures will happen if this time exceeds the waiting time of successive requests. Therefore, request mapping is facilitated on the basis of best-fit M. The best-fit M is identified based on its allocation rate. The optimized resource of $M(f_M)$ is then defined by Equation (2):

$$\left.\begin{aligned} f_M &= \frac{t_p + \frac{\rho_a}{\rho_r}}{s} \\ where, & \\ s &= \left(1 - \frac{t_r}{t_s}\right) + \gamma \end{aligned}\right\} \tag{2}$$

In Equation (2), s, the processing speed, is based on the ratio of balanced r and $\frac{t_r}{t_s}$. This f_M is used by the deep learning process with respect to time (t_p) for assigning r to an appropriate request. Assigning r to the processed request follows the satisfaction of internal and external constraints, that is, $\gamma_{ij_j^k} \leq \frac{r_i}{t_{r_i}}$ and f_M has a maximum value. If these two constraints are satisfied, then the request mapping occurs sequentially. On the other hand, retaining f_M is crucial as $\{r\} \in R \rightarrow \{t\}$ in all t_r accepted for processing. Therefore, a change in the sequence of request mapping degrades the performance of the allocating framework. Based on conditional analysis, the exploited and internal conditions are handled by training the f_M at each sequence. Figures 2 (a) and 2 (b) present the analysis of f_M with respect to the above conditions.

As presented in Fig. 2 (a), the initial mapping of $R \rightarrow \{t_r\}$ until $\gamma \leq \frac{r}{t_r}$. If a change in this condition from prolonged t_s or min $\{f_m\}$ is observed, then f_M based ordering of γ is facilitated. The output of Fig. 2 (a) is refined on the basis of max $\{f_M\}$ condition, following which the reordering takes place.

In the reordering process, the t_r with less t_s is mapped with the R (i.e.) for reordering the request map and analyze the all

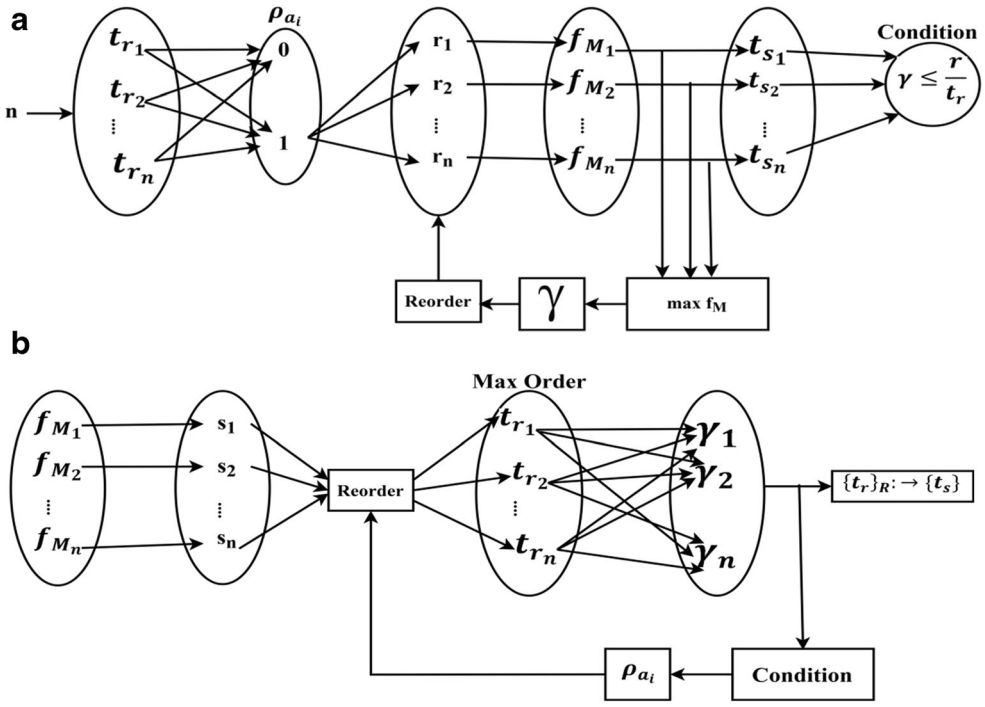

Fig. 2 (a) Analysis of f_M for $\gamma_{ij} \leq \frac{r_i}{t_{r_i}}$ (b) Analysis of max f_M (order of γ)

the mapped and unmapped data. In the successive mapping instances, $R \rightarrow \{t_s\}$ in the diminishing order of γ and f_M. The outputs of processing layers f_M and γ in Figs 2 (a), (b) are represented by Equations (3) and (4).

$$\left.\begin{aligned} f_{M_1} &= \frac{\rho_{a_1}}{\rho_r} + [s_1 \times (t_{s_1} - t_{r_1})] \\ f_{M_2} &= \frac{\rho_{a_2}}{\rho_r} + [s_2 \times (t_{s_2} - t_{r_2})] + \left(\frac{r_1}{t_{r_1}} - \gamma_1\right) \\ &\vdots \\ f_{M_n} &= \frac{\rho_{a_n}}{\rho_r} + [s_n \times (t_{s_n} - t_{r_n})] - \left(\frac{r_n}{t_{r_n}} - \gamma_n\right) \end{aligned}\right\} \tag{3}$$

In Equation (4), a variation in $\left(\frac{r_n}{t_{r_n}} - \gamma_n\right)$ is observed when Equation (1d) is not satisfied. Therefore, the allocation of R and request mapping is reordered to satisfy the condition in Equation (1d). This reordering is followed by the validation of γ for all mapped and unmapped requests. This validation is given as

$$\left.\begin{aligned} \gamma_2 &= \rho_{a_2} \times \left| \left(\frac{r_1}{t_{s_1}} - \frac{r_2}{t_{s_2}}\right) \right| - \frac{1}{s_1} \\ \gamma_3 &= \rho_{a_3} \times \left| \left(\frac{r_2}{t_{s_2}} - \frac{r_3}{t_{s_3}}\right) \right| - \frac{1}{s_2} \\ &\vdots \\ \gamma_n &= \rho_{a_n} \times \left| \left(\frac{r_{n-1}}{t_{s_{n-1}}} - \frac{r_n}{t_{s_n}}\right) \right| - \frac{1}{s_{n-1}} \end{aligned}\right\} \tag{4}$$

The request mapping is based on the validation of γ_n, that is, the analysis of γ_n prefers $\{t_r\}\ \forall\ R: \rightarrow \{t_s\}$, either in an ordered or unordered manner. Therefore, the mapping order that does not satisfy Equation (1d) is rolled over to next t_s, provided $t_s/t_r \leq \gamma_n$ (as per Equation (4)). If this condition is satisfied, then validation of ρ_a is not necessary, and t_r and t_s are not recorded. This ensures that the request is mapped successfully on either of the orders to R provided by M.

3.3 Resource Allocation

In this phase, we focus on the objective defined in Equation (1a) by satisfying Equations (1c) and (1d). In the request mapping process, the condition in Equation (1d) is satisfied by allocating requests to the appropriate t_s and maximizing γ. Therefore, this resource allocation process focuses on the condition in Equation (1c). In a conventional process, resource allocation is performed on a first come first serve basis, wherein the available resource is mapped to the request processed in t_r. Equation (1c) specifies that the rate of serviced requests γ is equal to the resource mapped for the requests with $\rho_a = 1$ constraint. This means the available requests are mapped with the allocated resource such that $\frac{\rho_a}{\rho_r} = 1$ in time t_s. The s of the M is the deciding factor in handling all requests and their allocated resources. Therefore, the instant allocation process for maximizing γ is defined as

$$f(\gamma) = \begin{cases} \sum_{i=1}^{M} f_{M_i} \cdot \frac{1}{s_{n_i}}, \text{if } M \geq \frac{t_r}{t_s} \\ \sum_{i=1}^{R} f_{M_i} \frac{1}{s_{n_i}} - \sum_{i=1}^{M} \left(1 - \frac{t_{r_i}}{t_{s_i}}\right), \text{if } M < \frac{t_r}{t_s} \end{cases} \tag{5}$$

In Equation (5), $f(\gamma)$ denotes the maximizing function with respect to the available M and R. The case of $M \geq R$ can be neglected as the available resource is sufficient for allocation,

provided $M \geq \frac{t_r}{t_s}$. Instead, if $m < \frac{t_r}{t_s}$, the overloading of R needs to be considered. In this case, the changes in s of a M needs to be verified, and hence, s and $\frac{t_r}{t_s}$ at any time instance is used for analyzing the allocation process. As given in Equation (1c), we can consider the case of $\sum_{i=1}^{M} \gamma_{i\frac{k}{j}} < r_{i\frac{k}{j}}$ as $M < \frac{t_r}{t_s}$ for identifying the possible resource allocation criteria. Therefore, the allocation is determined by max $\{f(\gamma)\}$, for $\frac{t_r}{t_s} > M$ or $\frac{t_r}{t_s} > R$. When $\frac{t_r}{t_s} > R$, the service provider is overloaded based on its s and $\frac{t_r}{t_s}$ rate. The process is differentiated based on s and $\frac{t_r}{t_s}$ rate illustrated in Figs 3(a), (b), respectively.

When handling requests based on s, The following changes has been considered which are listed as follows,

- $M - \frac{t_r}{t_s}$ Requests are allocated with the M to achieve max $f(\gamma)$.
- *if* t_{s_n} for R is high under the condition when s is high, thereby reducing the $(t_s - t_r)$,
- If $\frac{t_r}{t_s}$ is maximized, the allocation time of the requests, on the other hand of the M is considered.

If $t_{r_n} \leq (t_{s_{n-1}} - t_{r_{n-1}})$, then the M with min$\{t_{r_n}\}$ is selected for serving $\left(M - \frac{t_r}{t_s}\right)$ requests. The allocation precedes the M with minimum t_{r_n} such that the remaining requests are allocated with appropriate resources. Therefore, Equation (1c) can be re-written as

$$\sum_{j=1}^{M} \gamma_{ij\frac{k}{j}} = \begin{cases} r_{i\frac{k}{j}} \rho_a + s_i \dfrac{1}{t_{r_i}}, \forall M < \dfrac{t_r}{t_s} and\ i \in M \\ r_{i\frac{k}{j}} \rho_a + s_i . \dfrac{(t_{s_{n-1}} - t_{r_n - 1})}{\min\{t_{r_n}\}.t_{s_n}}, \forall \left(M - \dfrac{t_r}{t_s}\right) \leq R\ and\ i \in M \end{cases} \quad (6)$$

From Equation (6), the objective of resource allocation in Equation (1a) can be redefined as

$$\max_{\rho_{a,\gamma}} \left(\sum_{i=1}^{n} \rho_{r_i}\ \rho_{a_i} - \sum_{j=1}^{M - \frac{t_r}{t_s}} \sum_{k=1}^{R} \rho_{r_i} \times \frac{k}{j} \sum_{i=1}^{n} f_i(\gamma) \right) + \left(\min_{t_{r_n}} \sum_{j=M-\frac{t_r}{t_s}}^{t_r/t_s} \sum_{k=1}^{R} \rho_{r_i} \times \frac{k-M}{j} \sum_{i=1}^{n} \frac{k-m}{t_{r_i}\left(t_{s_{i-1}} - t_{r_{i-1}}\right.} \right) \quad (7)$$

The achievable resource allocation based on s and $\frac{t_r}{t_s}$ analysis differentiates the mapping of R using time and resource availability. Both the factors are verified to improve the rate of γ and $\frac{t_r}{t_n}$, irrespective of the requests in t_r. This also improves the non-overloading functions of M without increasing $(t_s - t_r)$ for any $M < \frac{t_r}{t_s}$. The objective of Equation (1a) is redefined in Equation (7) for satisfying Equation (1c), where $\gamma_{ij\frac{k}{j}}$ is achieved through $\max_{\rho_{a,\gamma}}$ and $\min_{t_{r_n}}$. Therefore, the

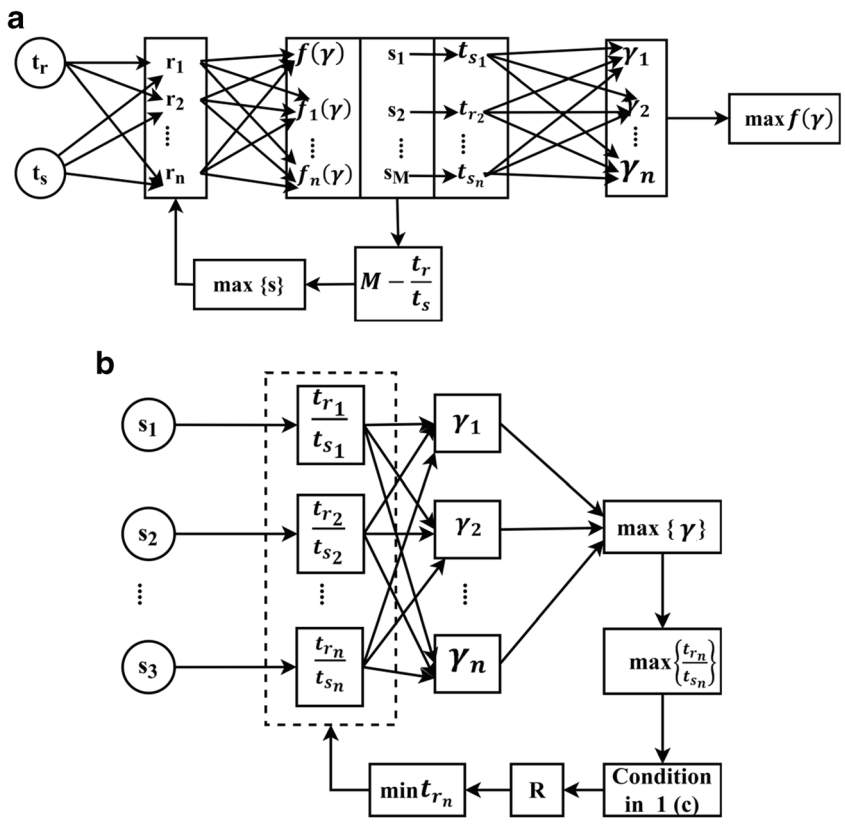

Fig. 3 (a) Analysis based on s (b) Analysis based on $\frac{t_r}{t_s}$

allocation is satisfied by maximizing γ based on $f(\gamma)$ for all R and t_{r_n}, for all $\left(M-\frac{t_r}{t_n}\right)$ requests.

3.4 Time Lag Optimization

The delay in resource allocation is a significant factor in SRAF, as the scalability support for n user devices must not increase $(t_s - t_r)$. If the allocation time of the requests increase, the change in ordering of request mapping and overloading of M requires additional time. Therefore, $(t_s - t_r)$ causes a lag in the allocation of resources for $\left(M-\frac{t_r}{t_s}\right)$ requests. Another factor affecting the regular allocation time is $f(\gamma)$, as the concentration of requests in $\min_{t_{r_n}}$. is high from the remaining requests. This t_{r_n} must be addressed in order to prevent unnecessary wait time of the consecutive requests. Therefore, the time constraints are resolved by controlling the processing and response time of previously queued requests. Different from the objectives in Equations (1a), (1c), and (1d), the time lag for $\left(M-\frac{t_r}{t_s}\right)$ is addressed in this phase. First, the time for processing and response are estimated for their balanced validation such that $t_p = (t_r - t_s)$, and the instances of t_p and t_s are the same. This condition is validated in two constraints, namely $t_p = t_s$ and $t_p > t_s$. The case of $t_s < t_p$ is not feasible as the processed request is dropped when this case is satisfied. Considering that the proposed resource allocation satisfies the condition and constraints in Equations (1a)–(1d), the $t_s < t_p$ condition is discarded. Similarly, when $t_p = t_s$, the request processing and allocation is ideal. On the other hand, if $t_s > t_p$,then $t_w \neq 0$, which results in prolonged serviced time/resource allocation (response) time. In order to confine the process time of resource allocation, t_w needs to be reduced. In some overloaded request-based scenarios, $t_w \neq 0$, but t_w can be shared among the available requests to reduce t_s. This time lag optimization follows the recurrent analysis of f_M and γ_M in the preceded allocation process based on s. The consideration of $\frac{t_r}{t_s}$ and the mapping is not necessary as t_w is relevant for a allocated/processed request. The mapping and maximization of $\frac{t_r}{t_s}$ is achieved through a learning based analysis as derived in Equation (7). The lag optimization is performed for $\left(M-\frac{t_r}{t_s}\right)$ requests that experience t_w. The validation of f_M and γ_M based on available M and s is considered such that

$$\left.\begin{aligned} f_M\left(M-\frac{t_r}{t_s}\right) &= \frac{\left(t_w+\frac{\rho_a}{\rho_r}\right)s}{\left(M-\frac{t_r}{t_s}\right)} \\ &\text{and} \\ \gamma_M\left(M-\frac{t_r}{t_s}\right) &= \left(\frac{M-\frac{t_r}{t_s}}{\sum t_p}-\frac{t_r}{t_s}\times\frac{1}{s}\right)-\frac{\rho_a}{\rho_r} \end{aligned}\right\} \quad (8)$$

In Equation (8), the modified f_M and γ_M for $\left(M-\frac{t_r}{t_s}\right)$ is computed where the existing requests are mapped to s with a high $\gamma_M\left(M-\frac{t_r}{t_s}\right)$. This case is valid until $t_s \leq t_p$; when this condition is not satisfied, M with $\max\left\{f_M\left(M-\frac{t_r}{t_s}\right)\right\}$ is selected for accommodating the request. This means $(t_r - t_s) + t_w \leq t_p$ for $\gamma_M\left(M-\frac{t_r}{t_s}\right)$ constrains $t_s \leq t_p$; else, M is replaced based on $\frac{\rho_a}{\rho_r}$, where $\rho_r > \rho_a$. This helps in allocating all resources expelled by ρ_a to the overloaded requests in time $(t_r - t_s) + t_w \leq t_p$. Therefore, $t_w = t_p - (t_r - t_s)$ when $t_p = t_s$; then, $t_w = 2t_p - t_r$, which is less than $\left(M-\frac{t_r}{t_s}\right) \times t_p$ or $\left(M-\frac{t_r}{t_s}\right) \times t_w + (t_r - t_s)$ time interval. Hence, the delay in processing is optimized by differentiating γ_M and f_M conditions for $\left(M = \frac{t_r}{t_s}\right)$ requests.

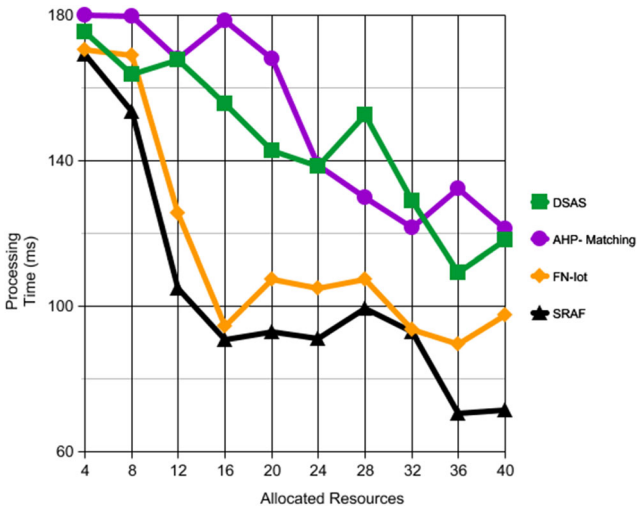

Fig. 4 Processing Time

4 Results and discussion

In this section, the performance of the proposed framework is discussed through suitable experiments. The experiments are carried out using an opportunistic network environment (ONE) simulator [33]. The IoT environment is created with varying number of devices (30, 60, 90, and 120), in which the number of resource servers is fixed. Metrics such as processing time, response time, resource allocation rate, and failure probability are observed through the simulation. The number of resource servers in the experimental simulation is 10, capable of handling multiple resources at the same time. The requests vary from 20 to 200, for 30 to 120 IoT devices. The maximum wait time of the request is set as 2.4 s. The 10 resource servers are configured with 2×2 *Gb* physical memory and 1 TB storage. Besides, the IoT environment is supplied with a shared resource of 1 TB multimedia application.

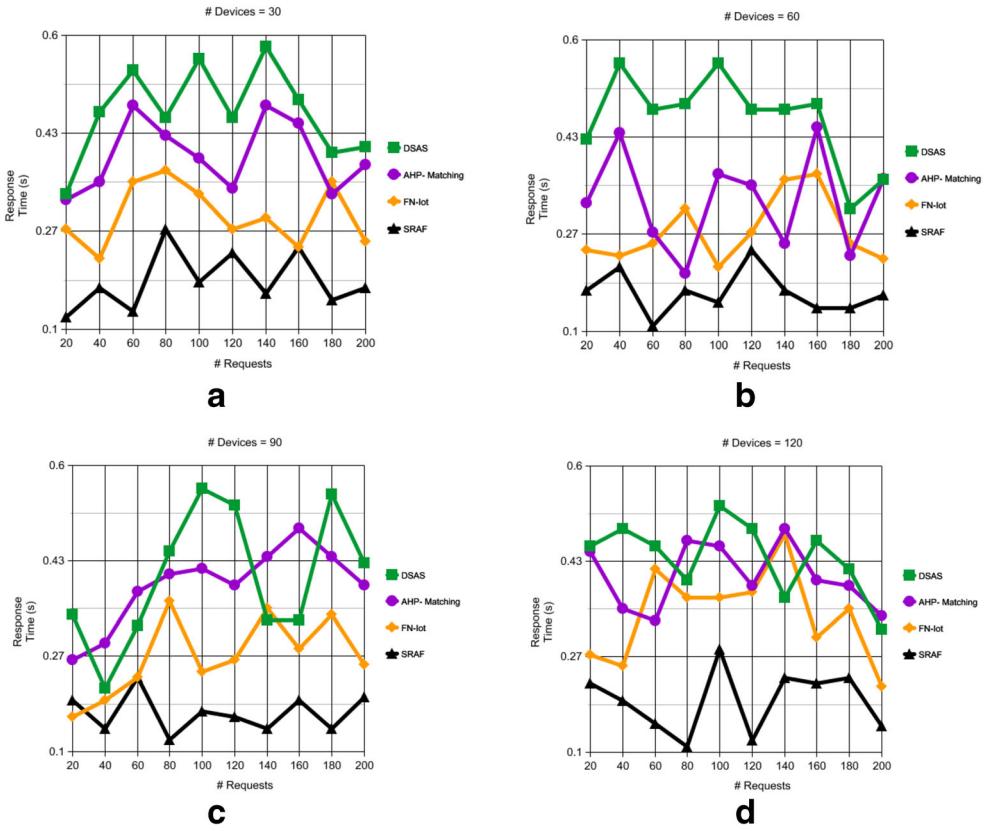

Fig. 5 Response Time

A resource server is configured to handle and serve 40 requests at a particular time instance. To verify the consistency of the proposed framework, the observed metrics are compared with the existing DSAS, AHP-matching, and FN-IoT methods which were discussed in the related work section.

4.1 Processing Time

Figure 4 illustrates the comparative analysis of processing time over the allocated resources. The processing time for the accepted requests is less until $\rho_r > \rho_a$, where the available M accepts the additional requests. On the other hand, if $\rho_r < \rho_a$ or requests, then resource allocation follows $\gamma_{ij\frac{k}{j}} \forall \left(M - \frac{t_r}{t_s}\right)$. The conditional analysis in Figs 2(a) (b) allocate $\left(M - \frac{t_r}{t_s}\right)$ requests to M with max $\{s\}$ and min$\{t_{r_n}\}$. This means there is no additional wait time for the overloading requests. Besides, in the time lag optimization process, $f_M\left(M - \frac{t_r}{t_s}\right)$ and $\gamma_M\left(M - \frac{t_r}{t_s}\right)$ attenuation prevents $t_w > t_p$; in addition, $(t_r - t_s) + t_w \leq t_p$ is retained. Therefore, the wait time for $\left(M - \frac{t_r}{t_s}\right)$ requests is confined within the maximum service time; hence, the processing time is retained for $\left(M - \frac{t_r}{t_s}\right) \times t_p$ in $t_w + (t_r - t_s)$ interval. Therefore, t_p is $(t_s - t_r + t_w)$, for $\left(M - \frac{t_r}{t_s}\right)$ condition, when$t_w = 0$, Then $t_p = (t_s - t_r)$ for the overloaded requests, which helps to reduce the processing time.

4.2 Response Time

The response time of varying requests and devices is compared in Figs 5 (a)– (d). The optimal response time is $(t_r - t_s)$ for the requests that are processed with the condition $\rho_a < \rho_r \forall$

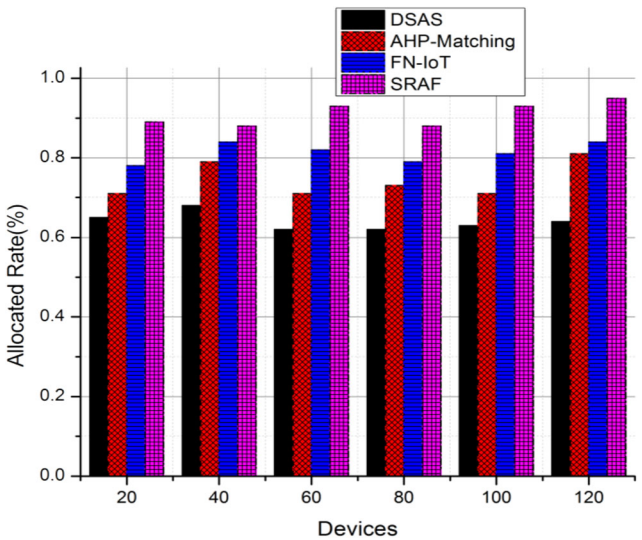

Fig. 6 Allocation Rate

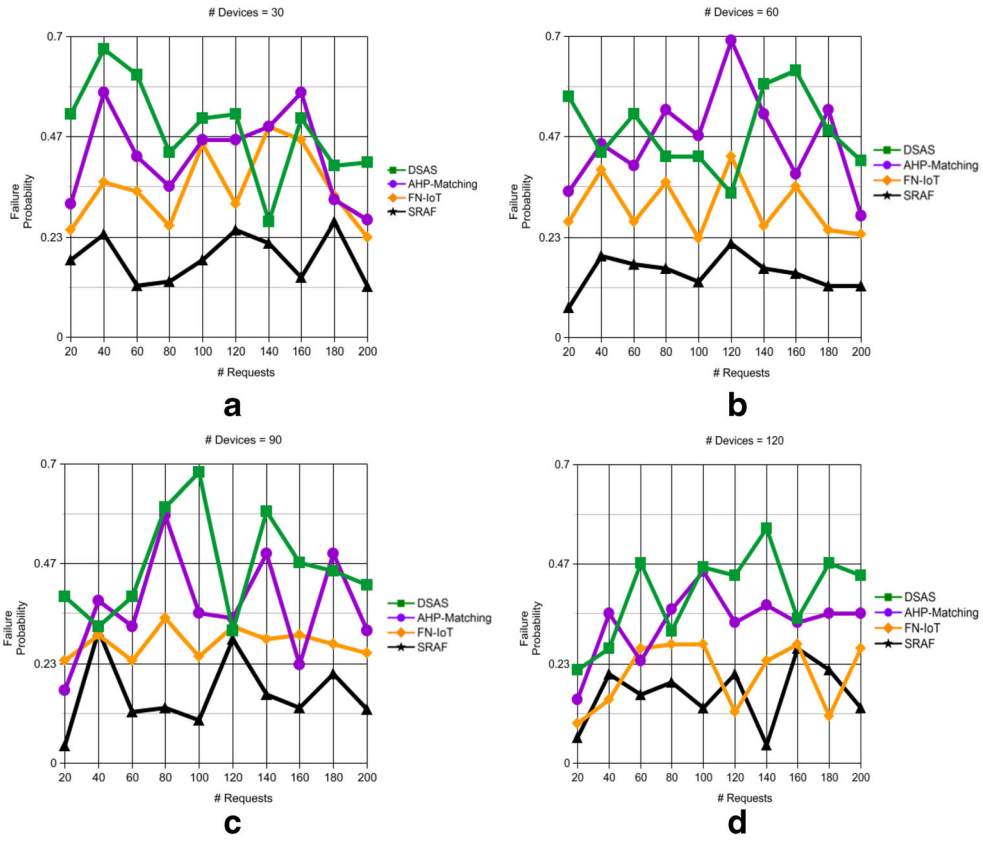

Fig. 7 Failure Probability

M. Response time increases if the number of ρ_r is less than the incoming requests. Therefore, the response time is $(t_r - t_s) + t_w$, where t_r is determined based on $\rho_a = 1$ condition. In the proposed framework, request mapping and resource allocation rely on the condition $\rho_a < \rho_r$ for all incoming requests handled by the service providers. These two processes limit the resource allocation response time. Contrarily, the response time for $\left(M - \frac{t_r}{t_s}\right)$ requests are to be limited by reducing t_w; this is done by selecting M based on $\max\{s\}$ and $\min\{t_{r_n}\}$ such that $\gamma_{ij\frac{k}{j}}$ and $\gamma_M\left(M - \frac{t_r}{t_s}\right)$ jointly satisfy the resource allocation objective in Equation (7). Therefore, based on s and t_{r_n}, the remaining requests are assigned to M for which $t_r = 2t_p - t_w$ or $2t_s - t_w$, provided the overall response time is $\frac{s}{2}(t_r - t_w)$, satisfying the maximum limit of $\left(M - \frac{t_r}{t_s}\right)\frac{t_p}{s}$. Hence, the processing time is less than $\left(M - \frac{t_r}{t_s}\right) \times t_w + (t_r - t_s)$.

4.3 Resource Allocation Rate

The resource allocation rate in the proposed framework is high, depending on ρ_a and s of the available M. In the request mapping process, f_M and $\{t_r\} : \rightarrow \{t_s\}$ based allocations are formed, where f_M and γ_n are the balancing factors for assigning n requests to M resource servers. Different from this allocation process, $\left(M - \frac{t_r}{t_s}\right)$ requests are mapped with M satisfying the s and t_{r_n} constraints. Besides, M must also meet f_M and γ_M (as defined in Equation (8)) modeled using s and t_w. Therefore, in the resource mapping and allocation phase, $\frac{t_r}{t_s}$ is the maximum requests served, which implies the resource allocation is performed for this rate of processed requests. Similarly, in the mapping of $\left(M - \frac{t_r}{t_s}\right)$, $\frac{\rho_a}{\rho_r}$ is the achievable resource allocation rate. Here, $\rho_a < \rho_r$ as the additional requests are allocated to M with $\min\{t_{r_n}\}$. Therefore, $\frac{t_r}{t_s}$ and $\frac{\rho_a}{\rho_r}$ achieve maximum resource allocation in the proposed framework (Fig. 6).

4.4 Failure Probability

The chances of failed resource allocation in the request mapping phase is less as the condition $\frac{t_r}{t_s}$ is satisfied for all $\rho_a \leq \rho_r$ of M. In order to reduce the failure probability of $\left(M - \frac{t_r}{t_s}\right)$ requests, the selection of M is based on $\min\{t_{r_n}\}$. If t_w or t_p exceeds $(t_r - t_s)$, then the resource allocations is unsuccessful, reducing the success rate of the request. The time allocated for processing prolongs the delay for the consecutive requests. Therefore, assigning M based on γ_M and f_M (as per Equation (8)) helps retain the concurrent processing and mapping of $\frac{t_r}{t_s}$

requests. Therefore, the $(t_r - t_s)$ time of the previous requests is the t_w for the new requests. In particular, the successive time for two t_r is t_w, and hence, the processing experiences a delay. Besides, allocating $\left(M - \frac{t_r}{t_s}\right)$ requests within the defined time interval helps reduce the failure in request processing and resource allocation (refer Figs 7(a)–7(d)). This case is unanimous for varying requests and user densities.

5 Conclusion

This paper proposes an SRAF for a user-focused IoT paradigm. The aim of this framework is to improve the quality of response for available users with in-time resource allocation and swift request processing. Deep learning aids the concise management of available resources, user requests, and the lag in allocation. This deep learning paradigm is responsible for retaining the liveliness of requests and in-time allocation of the resources. Therefore, allocation is performed by balancing processed requests and available resources. The optimal performance and delay in response is attuned using a time lag optimization process for the overloaded requests, based on the processing speed and time of the resource providers. The joint process flow helps improve the resource allocation rate, and reduce the processing and response time and failure probability. Future studies can include meta-heuristic techniques to further improve the resource allocation process.

Acknowledgements The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through Research Group No. RG-1438-027.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Lee I (2019) The internet of things for enterprises: an ecosystem, architecture, and IoT service business model. Internet of Things 7: 100078
2. Asghari M, Yousefi S, Niyato D (2019) Pricing strategies of IoT wide area network service providers with complementary services included. J Netw Comput Appl 147:102426
3. Al-Makhadmeh Z, Tolba A (2019) Utilizing IoT wearable medical device for heart disease prediction using higher order Boltzmann model: a classification approach. Measurement 147:106815
4. Alqahtani F, Al-Makhadmeh Z, Tolba A, Said O (2020) TBM: a trust-based monitoring security scheme to improve the service authentication in the internet of things communications. Comput Commun 150:216–225
5. Read J, Bifet A, Fan W, Yang Q, Yu P (2019) Introduction to the special issue on big data, IoT Streams and Heterogeneous Source Mining. *International Journal of Data Science and Analytics* 8(3): 221–222
6. Simiscuka AA, Markande TM, Muntean G-M (2019) Real-virtual world device synchronization in a cloud-enabled social virtual reality IoT network. IEEE Access 7:106588–106599
7. Tolba A, Al-Makhadmeh Z (2020) A recursive learning technique for improving information processing through message classification in IoT–cloud storage. Comput Commun 150:719–728
8. Said O, Al-Makhadmeh Z, Tolba A (2020) EMS: an energy management scheme for green IoT environments. IEEE Access 8: 44983–44998
9. Metzger F, Hobfeld T, Bauer A, Kounev S, Heegaard PE (2019) Modeling of aggregated IoT traffic and its application to an IoT cloud. Proc IEEE 107(4):679–694
10. Ghanbari Z, Navimipour NJ, Hosseinzadeh M, Darwesh A (2019) Resource allocation mechanisms and approaches on the internet of things. Clust Comput 22(4):1253–1282
11. Said O, Tolba A (2018) Design and performance evaluation of mixed multicast architecture for internet of things environment. J Supercomput 74(7):3295–3328
12. Tolba A, Elashkar E (2019) Soft computing approaches based bookmark selection and clustering techniques for social tagging systems. Clust Comput 22(2):3183–3189
13. Kim H-W, Park JH, Jeong Y-S (2019) Adaptive job allocation scheduler based on usage pattern for computing offloading of IoT. Futur Gener Comput Syst 98:18–24
14. Elgendy IA, Zhang W, Tian Y-C, Li K (2019) Resource allocation and computation offloading with data security for mobile edge computing. Futur Gener Comput Syst 100:531–541
15. Alarifi A, Tolba A, Al-Makhadmeh Z, Said W (2018) A big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks. J Supercomput. https://doi.org/10.1007/s11227-018-2398-2
16. Tolba A (2019) Content accessibility preference approach for improving service optimality in internet of vehicles. Comput Netw 152:78–86
17. Wang Y, Liang Y, Tian W, Zeng P, Zhao Q, Tan J, Chai J, Feng L (2019) Paging-Efficient NB-IoT Resource Allocation for Massive-Connectivity-Enabled Communications in Smart Grid. *2019 IEEE International Conference on Energy Internet (ICEI)*
18. Sun H, Yu H, Fan G, Chen L (2019) Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture. Peer-to-Peer Networking and Applications
19. Alarifi A, Tolba A (2019) Optimizing the network energy of cloud assisted internet of things by using the adaptive neural learning approach in wireless sensor networks. Comput Ind 106:133–141
20. AlFarraj O, AlZubi A, Tolba A (2018) Trust-based neighbor selection using activation function for secure routing in wireless sensor networks. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-018-0885-1
21. Abedin SF, Alam MGR, Kazmi SMA, Tran NH, Niyato D, Hong CS (2019) Resource allocation for ultra-reliable and enhanced Mobile broadband IoT applications in fog network. IEEE Trans Commun 67(1):489–502
22. Mergenci C, Korpeoglu I (2019) Generic resource allocation metrics and methods for heterogeneous cloud infrastructures. J Netw Comput Appl 146:102413
23. Nassar A, Yilmaz Y (2019) Reinforcement learning for adaptive resource allocation in fog RAN for IoT with heterogeneous latency requirements. IEEE Access 7:128014–128025
24. Liu X, Qin Z, Gao Y, Mccann JA (2019) Resource allocation in wireless powered IoT networks. IEEE Internet Things J 6(3):4935–4945
25. Li Z, Yang Z, Xie S (2019) Computing resource trading for edge-cloud-assisted internet of things. IEEE Transactions on Industrial Informatics 15(6):3661–3669

26. Li X, Liu Y, Ji H, Zhang H, Leung VCM (2019) Optimizing resources allocation for fog computing-based internet of things networks. IEEE Access 7:64907–64922
27. Tian X, Huang W, Yu Z, Wang X (2019) Data driven resource allocation for NFV-based internet of things. IEEE Internet Things J 6(5):8310–8322
28. Ramezani P, Zeng Y, Jamalipour A (2019) Optimal resource allocation for multiuser internet of things network with single wireless-powered relay. IEEE Internet Things J 6(2):3132–3142
29. Chen J, Zhang L, Liang Y-C, Kang X, Zhang R (2019) Resource allocation for wireless-powered IoT networks with short packet communication. IEEE Trans Wirel Commun 18(2):1447–1461
30. Aazam M, Harras KA, Zeadally S (2019) Fog computing for 5G tactile industrial internet of things: QoE-aware resource allocation model. IEEE Transactions on Industrial Informatics 15(5):3085–3092
31. Dai, H., Zhang, H., Wu, W., Wang, B.: A game-theoretic learning approach to QoE-driven resource allocation scheme in 5G-enabled IoT. *EURASIP Journal on Wireless Communications and Networking*, 2019 (1), (June 2019)
32. Gao H, Duan Y, Shao L, Sun X (2019) Transformation-based processing of typed resources for multimedia sources in the IoT environment. Wirel Netw
33. Prodhan AT, Das R, Kabir H, Shoja GC (2011) TTL based routing in opportunistic networks. J Netw Comput Appl 34(5):1660–1670

Zafer Al-Makhadmeh received the M.Sc. and Ph.D. degrees from the Department of Computer Engineering, Faculty of Information and Computer Engineering, Kharkov National Technical University of Ukraine, in 1998 and 2001, respectively. He is currently an Associate Professor with the Department of Computer Science, Community College, King Saud University, Saudi Arabia. His main research interests include cloud computing, image processing, computer vision, and intelligent systems.

Amr Tolba received the M.Sc. and Ph.D. degrees from Mathematics and Computer Science Department, faculty of science, Menoufia University, Egypt, in 2002 and 2006, respectively. He is currently an Associate Professor at the Faculty of Science, Menoufia University, Egypt. He is currently on leave from Menoufia University to the Computer Science Department, Community College, King Saud University (KSU), Saudi Arabia. Dr Tolba serves as a technical program committee (TPC) member in several conferences. He has authored/coauthored over 65 scientific papers in top ranked (ISI) international journals and conference proceedings. His main research interests include socially aware networks, vehicular ad-hoc networks, Internet of Things, intelligent systems, Big Data, recommender systems, and cloud computing. (atolba@ksu.edu.sa).