# Context-Aware Wireless Connectivity and Processing Unit Optimization for IoT Networks

Metin Ozturk*, Attai Ibrahim Abubakar†, Rao Naveed Bin Rais‡, Mona Jaber§, Sajjad Hussain†, Muhammad Ali Imran† ¶

***Abstract*—A novel approach is presented in this work for context-aware connectivity and processing optimization of Internet of things (IoT) networks. Different from the state-of-the-art approaches, the proposed approach simultaneously selects the best connectivity and processing unit (e.g., device, fog, and cloud) along with the percentage of data to be offloaded by jointly optimizing energy consumption, response-time, security, and monetary cost. The proposed scheme employs a reinforcement learning algorithm, and manages to achieve significant gains compared to deterministic solutions. In particular, the requirements of IoT devices in terms of response-time and security are taken as inputs along with the remaining battery level of the devices, and the developed algorithm returns an optimized policy. The results obtained show that only our method is able to meet the holistic multi-objective optimization criteria, albeit, the benchmark approaches may achieve better results on a particular metric at the cost of failing to reach the other targets. Thus, the proposed approach is a device-centric and context-aware solution that accounts for the monetary and battery constraints.**

***Index Terms*—Constrained Devices, Efficient Communications and Networking, Machine Learning, Context-Awareness, Energy Efficient Devices, Reinforcement Learning.**

## I. Introduction

The world is witnessing a rapid adoption of the Internet of things (IoT) technologies to enhance and augment human endeavors in operating almost every aspect of our lives. IoT is a system of connected devices that sense and actuate signals in the real world. These devices are connected and enable real-time data collection and respective optimized action. However, designing and deployment of IoT networks pose multiple challenges, including energy consumption, connectivity, and data processing [1]–[3].

As previously stated in [4], different communication protocols and standards exist, with each having its pros and cons. Cellular IoT networks, for example, operate in the licensed band and utilize mobile network infrastructure for IoT device inter-connectivity. They are the most suitable option for private data transmission because they provide reliable, secure (due to the presence of eSIM card), and wide range coverage with vast infrastructure. Three cellular technologies have been proposed by 3GPP Rel-13 to support IoT connectivity: narrowband IoT (NB-IoT), enhanced machine type communications (eMTC), and enhanced coverage GSM (EC-GSM) [2]. NB-IoT has been designed to offer low cost, wide coverage, and low data rate transmission with limited mobility support. The short-range solution, on the other hand, comprises Zigbee, Wi-Fi, Bluetooth, etc. which could also be publicly, privately or jointly owned [5].

As argued in [6], the smart port concept is adopted in this research to illustrate context-aware connectivity and processing. Smart ports represent a vast application as they bridge the fleet of vessels to the fleet of trucks and include various aspects of Industry 4.0 technologies. Moreover, cost-efficiency is a prime objective in the application of smart ports in view of the competition between various parties in quality and price. Towards this end, as the industry looks for least complexity in IoT devices to reduce their capital and operating cost and enable cost-effective applications, the design of an IoT system faces the challenge of optimizing the location of the data processing. This could be in the cloud or device but also in an intermediate node (e.g., Wi-Fi gateway) referred to as fog.

Furthermore, the massive deployment of IoT devices would result in the generation of a huge amount of data which would be difficult for traditional processing techniques to handle [7]. In addition, due the massive inter-connectivity of devices, network management and orchestration will become very complex and challenging, hence the need for more intelligence to be included in IoT networks to enable enhanced data processing as well as intelligent and autonomous network operation [8]. In the midst of diverse IoT network standards and protocols, vast IoT applications, varying characteristics of IoT devices (stationary, mobile, battery life, etc.), and several connectivity options, there is the need to carefully select the most suitable method to harness, store, transmit, process, and secure the data obtained from massive IoT networks in an energy efficient and cost effective way. The application of artificial intelligence (AI) and machine learning (ML) would help harness the useful information embedded in the massive data that is generated in order to facilitate decision making while ensuring smart and efficient network operation.

### A. Related Work

There are many challenges associated with IoT networks ranging from energy consumption, security, latency and cost of data processing and transmission, etc. This has led to the introduction various communication standards and protocols, and technologies to ensure that these challenges are addressed.

*Electrical and Electronics Engineering, Ankara Yıldırım Beyazıt University, Turkey.
†Communication, sensing and imaging (CSI) research group, James Watt School of Engineering, University of Glasgow, United Kingdom.
‡Electrical and Computer Engineering, Ajman University, UAE.
§School of Electronic Engineering and Computer Science, Queen Mary University of London, united Kingdom.
¶AI Research Centre (AIRC), Ajman University, UAE.

This work was supported by EPSRC Global Challenges Research Fund the DARE Project: Grant EP/P028764/1, and partially funded by DGSR, Ajman University under grant ID 2019-IRG-ENIT-8.

Some of these communication protocols and standards include long range solutions (NB-IoT, eMTC, EC-GSM) and short range solutions (Zigbee, Wi-Fi, Bluetooth) which may be licensed or non-licensed, private or public networks [5], [9]–[11]. In addition, IoT enabling technologies including enhanced data storage and processing infrastructures such as caching, fog and cloud storage, and mobile edge computing (MEC) [12]–[14] have also been developed. Several techniques (data aggregation, data offloading, computation offloading, etc.,) [15], [16] and optimization algorithms including machine learning and deep learning [17], [18] have been proposed to address the aforementioned challenges either individually or jointly.

The authors in [19] carried out a comparative study of the connectivity based performance of two IoT enabling technologies, namely NB-IoT and LoRa to ascertain their suitability for smart grid applications. In [20], an energy performance analysis of various low power wide area network (LPWAN) technologies was performed in order to determine the type of sensors to select, based on their battery life, that would meet the energy requirement of the desired IoT application. In [21], [22] a survey of various security challenges and solutions in IoT networks was performed. A lightweight authentication mechanism was proposed in [23], to secure IoT devices in order to prevent unwanted access while in [24], a survey of various authentication techniques for the security of IoT devices was performed.

One of the major concerns of IoT networks is the energy consumption of IoT devices, hence the network has to be designed and optimized in such a way that it ensures the longevity of the device battery life. In this regard, various optimization frameworks as well as data aggregation, computation offloading techniques have been proposed. The authors in [25] proposed a smart game algorithm to optimize both data transmission and energy consumption. A novel energy and context-centric scheme based on ML was introduced in [26], which aims at elongating the device battery life while ensuring that the required quality of service (QoS) level is maintained. The use of data aggregators for data collection from IoT devices in order to reduce signalling overhead as well optimize energy consumption of IoT devices was considered in [27], where a drone, which serves as the data aggregator, was used to collect data from a cluster of IoT devices before relaying it to the cellular network. The proposed technique resulted in improved energy efficiency for the IoT devices. The authors in [28] proposed an energy-optimal data aggregation mechanism using mixed integer programming for optimal data routing and aggregation decisions in IoT networks. Even though the use of data aggregators results in less signalling overhead and reduced energy consumption for IoT devices, this approach often results in increased latency and transmission delays, making it unsuitable for critical IoT applications.

To reduce the latency associated with data transmission, various caching techniques have also been introduced. The authors in [29] proposed a caching mechanism using fog nodes where IoT users with similar interest and in close proximity to each other are grouped together and mapped to a fog node, thereby reducing service delays and enhancing the total throughput of the system. Similarly, An efficient caching framework for IoT applications in order to ensure quick content retrieval and reduced latency was introduced in [30]. A review of different machine learning approaches for caching IoT data at fog nodes was carried out in [31].

The choice of where to process data is also one of the decisions that need to be made in IoT networks due to the limited processing power of IoT devices as well as the need to prolong their battery life and increase the data processing speed. In this regard, different computation offloading strategies have been developed. The authors in [32], [33] considered the trade-off between the energy consumed for local data processing at the IoT device and that due to offloading to the edge cloud. The former proposed a theoretical framework for the reduction of the energy consumption of the devices by developing an optimal offloading strategy for all user devices in the network. The latter considered the delays—due to offloading data to the edge cloud for processing—as an optimization constraint while developing a close form expression for determining the optimal offloading strategy for multiple user devices in order to reduce their energy consumption. A data compression mechanism was proposed in [34] to reduce the amount of data offloaded from the IoT device to the edge server for data analysis purposes thereby reducing the energy consumption in the IoT device due to data transmission.

There is also the need to develop proper managements platforms in order to efficiently coordinate the operations of the large number of heterogeneous IoT devices that are connected together in a context-aware framework. This is to ensure that the IoT devices respond rapidly to dynamic changing events in their environment and that QoS is guaranteed when transmitting the information gathered to the required destinations. In this regards, a situation-aware IoT service orchestration framework was proposed in [35], wherein three heuristic algorithms were developed. The first algorithm was to automate the process of event detection based on a predefined event pattern that includes both the event selection and utilization strategy. The second algorithm, which is a process decomposing algorithm, was designed to effectively coordinate the services of the various IoT devices. Then, the final algorithm was developed to detect the occurrence of event mismatch during the process of coordinating the services of the IoT devices. In [36], an intelligent context-aware framework for QoS management during video transmission in mobile and fixed networks was proposed. The proposed context-aware QoS management system comprises three major phases including context discretization, reduction, and QoS assurance. Three separate heuristic algorithms were developed to find the optimal solution in each phase in order to ensure effective end-to-end QoS management.

Deep learning approaches have been exploited for data analysis and network optimization because of their excellent ability to analyse and learn from the large volume of data generated by IoT devices [37]. A deep learning model was proposed in [38] for joint transmission and recognition for IoT devices in order to ensure efficient data transmission to the server for recognition purposes under very low signal-to-noise-ratio (SNR). In [39], a deep learning based framework

for forecasting the energy consumption of IoT networks in an edge computing scenarios was introduced. A novel offloading strategy based on deep learning was developed in [40] to enhance the performance of IoT networks as well as optimize the amount of edge computing tasks that can be performed. Another important aspect of IoT networks is ensuring the confidentiality of the massive data that is collected from IoT devices. To that end, the authors in [41] proposed a deep learning framework for big data analytic in IoT networks that would ensure that the privacy of the data is preserved. Blockchain technologies has also been combined with deep learning solutions in order to ensure that the data generated from IoT devices during model training is secure. In this regards, the work in [42] proposed a blockchain based deep learning model to preserve the privacy of data generated by IoT devices during model training and process of data analysis. A survey of the application of blockchain technology and deep learning for privacy in IoT was carried out in [43], [44].

However, most of the proposed approaches considered a layered approach to IoT network optimization, where one or two optimization constraints were considered among the many constraints which include energy consumption, connectivity, transmission delays, storage, computing/processing, and monetary considerations, respectively. We argue that a holistic solution approach that considers several optimization constraints is necessary for IoT network optimization in order to fully harness the benefits that such solution has to offer. In this regard, the authors in [3] proposed a context-aware scheme for IoT networks for the joint optimization of computation and connectivity using $Q$-learning, a reinforcement learning (RL) algorithm. A two stage $Q$-learning algorithm was developed where the first stage involved the selection of the connection-processing unit pair, while the second stage was to determine the amount of data to be offloaded to fog or cloud. However, the penalty function developed for the $Q$-learning algorithm is not scalable, and hence difficult to adapt it to changing IoT application requirements. Moreover, in the proposed approach, it is not possible to prioritize the importance of one constraint over another, which helps enhance the context-awareness of the solution.

### *B. Objectives and contributions*

RL is employed in this work in order to manage multiple optimization objectives, such as connectivity, processing, storage, etc., jointly. The following objectives are pursued in this work:

- **energy consumption and monetary cost:** based on the battery conditions of IoT devices, the total energy consumption of a device—incurred by data processing and transmission—along with monetary cost is minimized.
- **quality requirements of IoT devices:** meeting the stipulated quality of the IoT device/application is another objective of this optimization problem. In this work, we focus on security and latency exigencies.

In that regard, in this work, a context-aware connectivity and processing optimization in IoT networks using $Q$-learning is developed for the joint optimization of connection-type, processing unit, percentage of data to be offloaded, response time, security as well as monetary cost. Different from [3], the proposed holistic framework for IoT network optimization involves only one stage of $Q$-learning. A simpler and more generalized penalty function of $Q$-learning—which will be referred to as penalty function hereafter—is also introduced to capture the variations in IoT application demands. In addition, a weighting mechanism is designed that will enable the IoT devices to prioritize any of the optimization constraints, on which basis, the algorithm would change its behaviour. Due to the fact that the work in [3] and the present work have structural differences, it would not be fair to compare them. In other words, the novel weighting mechanism included here considerably distinguishes this work from [3], since IoT devices are allowed to rank their requirements, making the algorithm more dynamic and flexible. In [3], the requirements are considered in a binary form (i.e., 0 for not required and 1 for required), while in this work—albeit still being discritized—the IoT devices can choose the strictness of their requirements.

Having said all this, the differences between this work and [3] can be summarized as follows:

- A weighting mechanism is added in order to consolidate the power of the algorithm, such that IoT devices do not have to make a hard decision between having a requirement or not for a particular parameter, including security and latency. They are rather given the freedom of ranking their requirements, thus making their decision softer and the algorithm more flexible. Indeed, this mechanism makes it very difficult to compare both works as [3] does not have such mechanism.
- The design of $Q$-learning algorithm is made significantly simpler:
  - There are two $Q$-learning stages in [3], making the design more complex and more prone to errors, since ML cannot guarantee the optimal solution and thus there is a possibility of having errors. Therefore, having two stages of ML increases the probability of error. In this work, on the other hand, these two stages are compressed in a single $Q$-learning stage, and thus the complexity and probability of error are dramatically reduced.
  - The penalty function of $Q$-learning is much simplified, making it more understandable and intuitive. This also reduces the need for parameter tuning, as there is a huge number of parameters to be tuned in [3], which is significantly reduced in this work. In other words, the penalty function in [3] is customized for a specific scenario and once network conditions change it would need readjustments. In this work, on the other hand, the penalty function is made much more generic and can be applied to changing network conditions.

The remaining parts of this paper is organized as follows: The system model is defined in Section II, while the considered problem is formulated in Section III. The proposed framework is presented in Section IV, and the performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3152381, IEEE Internet of Things Journal



evaluation via a comprehensive results analysis are discussed in Section V. Lastly, Section VI concludes the article.

## II. System Model

The novel scheme for context-aware connectivity and processing developed in this work may be applied to any IoT application. For clarity purposes, the system model is built based on the smart port environment as depicted in Fig. 1. The IoT devices are powered by batteries of varying capacity and lifespan. They are all equipped with a measure of processing ability required for basic operation and are able to offload their data processing operations to Wi-Fi gateway (fog) or the base station (cloud).

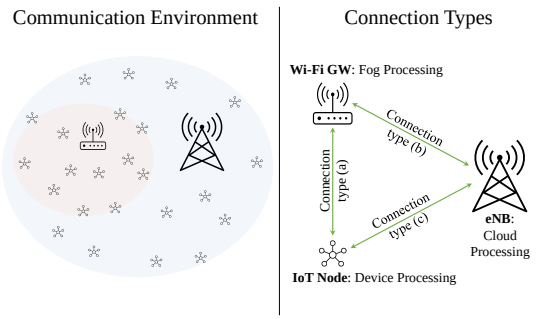


Fig. 1. System modelling. The IoT network scenario is depicted on the left hand side, while the descriptions of the considered wireless connection types are given on the right. GW: gateway

### A. Propagation Model

As seen in Fig. 1, the following wireless connections are modelled: (a) Device-to-Gateway (Wi-Fi), (b) Device-to-eNB (NB-IoT), and (c) Gateway-to-eNB (LTE). On the one hand, as we considered that there are neighbouring Wi-Fi gateways operating at the same frequency—which is a quite likely scenario—, connection type (a) is taken to be interference limited. On the other hand, it is assumed that i) NB-IoT technology is not used in other surrounding eNBs and ii) a scheduler is used for LTE connections, thereby connection types (b) and (c) are considered to be noise limited.

The propagation modelling is performed here in order to obtain the amount of required transmit power for all the connection types, which is subsequently used for energy consumption computations. In general, wireless communication is governed by propagation loss that is a function of distance and frequency. We adopt the log-distance path-loss ($L$) model as follows:

$$L = L_0 + 10\delta \log_{10} \frac{d}{d_0} + \xi_{\mathrm{g}}, \tag{1a}$$

$$L_0 = 20 \log_{10} \left( \frac{4\pi d_0}{\lambda} \right), \tag{1b}$$

where $\delta$ is the path-loss exponent, $d$ (in meter) is the distance between the transmitter and receiver with $d_0$ (in meter) being the reference distance. $\xi_{\mathrm{g}}$ is the shadowing component with a standard deviation, $\sigma$, and zero mean. $L_0$ is the path-loss at the reference distance $d_0$, $\lambda = \dfrac{c}{f_{\mathrm{c}}}$ is the wavelength, where $c$ is the speed of light, and $f_{\mathrm{c}}$ is the transmission frequency.

Signal-to-interference-plus-noise ratio (SINR) is an important measure, since it determines the level of signal power at the receiver, which imposes a certain level of received signal power—referred to as receiver sensitivity, under which the communication link fails. Moreover, the receiver sensitivity varies for different radio access technologies (RATs), thus the communication channel should be designed accordingly. Given that various communication types (e.g., LTE, NB-IoT, and Wi-Fi) with diverse receiver sensitivities are considered in this work, the link margin concept is used to capture the distinctive sensitivity levels.

From Shannon's channel capacity theorem, we know that the achievable data rate is a function of SINR, such that higher SINR values result in higher data rates, and vice versa. As such, by manipulating the well-known Shannon capacity formula, we obtain the receivable data rate as follows [3], [32]:

$$D = TB \log_2 \left( 1 + \frac{P_{\mathrm{r}}}{P_{\mathrm{I}} + \mathcal{N}_0 B} \right), \tag{2}$$

where $P_{\mathrm{r}}$ is the required received power, $D$ (in bits) is the finite-length data to be transmitted, $T$ is the time period, $B$ is the channel bandwidth, and $P_{\mathrm{I}}$ is the cumulative interference power on the given channel during time period $T$. Please note that $P_{\mathrm{I}}$ has no effect for wireless connections of type (b) and (c), since they are noise limited. Next, the required received power—which is needed to achieve the target data transmission—is calculated using (2) and solving for $P_{\mathrm{r}}$ [3], [32]:

$$P_{\mathrm{r}} = \left( 2^{\frac{D}{TB}} - 1 \right) (P_{\mathrm{I}} + \mathcal{N}_0 B). \tag{3}$$

In this regard, the IoT devices are aware of their data rate requirements, which is then used to compute the required transmit power through (3).

### B. Energy consumption model

Similar to [3], two main components of the energy consumption are considered and modelled: wireless transmission and task computation. The energy consumption due to wireless transmission is represented by $E_{\mathrm{tx}}$, while that due to task computation is denoted by $E_{\mathrm{p}}$, hence the total energy consumption in the IoT network is obtained by summing both together. The energy as a result of transmission power is dependent on the transmission route selected by the IoT device. The transmission route can be either one hop from device to NB-IoT ($E_{\mathrm{tx,b}}$) or two hops with the first hop being device to Wi-Fi gateway while the second hop Wi-Fi gateway to LTE ($E_{\mathrm{tx,a}} + E_{\mathrm{tx,c}}$). The energy consumption due to task processing is determined by the amount of data ($D$), computing capability of the processing unit[1] ($X$) as well as the energy consumed in each computation cycle ($\epsilon$); thus $E_{\mathrm{p}} = f(D, X, \epsilon) = \frac{D}{W} X \epsilon$, where $W$ is the number of bits per data element (DE) [32].

[1]It is measured in the number of computational cycle per data element (DE); i.e., higher $X$ yields less computational power

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3152381, IEEE Internet of Things Journal



### C. Response time model

The response time is modelled in this work in a similar fashion to [3]. The response time experienced by an IoT device is a function of the latency experienced during transmission from IoT device to the server. We model the uplink delay in this paper, while assuming that all devices have the same downlink delay. Towards that end, task computation (processing delay, $t_{\mathrm{p}}$) and data transmission (transmission delay, $t_{\mathrm{t}}$) are two primary elements of the uplink delay. Of these two, the processing delay ($t_{\mathrm{p}}$) is a function of the processor's computational power ($X$), such that more delay is experienced with a lower computational power at the processor. A server usually possesses much greater processing power compared to an IoT device, with gateway in between, such that $X_{\mathrm{c}} < X_{\mathrm{f}} < X_{\mathrm{d}}$, where $X_{\mathrm{d}}, X_{\mathrm{f}}, X_{\mathrm{c}}$ are the computational powers of device, fog, and cloud, respectively. Hence, we model $t_{\mathrm{p}}$ with respect to the processing powers of the various location where task computation will be performed as follows: $\frac{t_{\mathrm{p,d}}}{X_d} = \frac{t_{\mathrm{p,f}}}{X_{\mathrm{f}}} = \frac{t_{\mathrm{p,c}}}{X_c}$, with $t_{\mathrm{p,d}}$, $t_{\mathrm{p,f}}$ and $t_{\mathrm{p,c}}$ denoting delays due to task processing at the device, fog, and cloud, respectively. The output from the processing stage is usually compressed data and is of lesser volume compared to the original raw input data. As a result, the compression rate that accounts for the difference in data volume between raw input and processed output data is expressed as $\mho$; $D_{\mathrm{r}} = \mho D_{\mathrm{p}}$ with $D_{\mathrm{r}}$ and $D_{\mathrm{p}}$ denoting the volume of input and output data, respectively.

The type of RAT as well as the amount of data to be transmitted have a major impact on the level of transmission delays that will be experienced. The use of Wi-Fi technology causes more transmission delays because it uses the unlicensed frequency band, making it more prone to frequent re-transmission due to likely collisions. Hence, in this research, as it is done in [3], we consider this effect by introducing a factor $F > 1$, such that delay experienced due to data re-transmission over Wi-Fi is $F$ times greater than that over LTE or NB-IoT; $t_{\mathrm{t,a}} = t_{\mathrm{t,b}}F = t_{\mathrm{t,c}}F$, where $t_{\mathrm{t,a}}$, $t_{\mathrm{t,b}}$ and $t_{\mathrm{t,c}}$ represents the transmission delays due to connection type (a), (b), and (c), respectively. This is captured in the model in Fig. 1, where the IoT device or the gateway could be the source with either the gateway or cloud being the receiver.

As a result, the total response time per action can be computed as:

$$R = t_{\mathrm{p}}D + \sum_{i=1}^{N_{\mathrm{h}}} t_{\mathrm{t,i}}D_{\mathrm{i}}, \tag{4}$$

where $N_{\mathrm{h}}$ is the number of hops, and $D \in \{D_{\mathrm{r}}, D_{\mathrm{p}}\}$, while $t_{\mathrm{t,i}}$ and $D_{\mathrm{i}}$ denotes the values of $t_{\mathrm{i}}$ and $D$ for the $i^{\mathrm{th}}$ hop respectively. It should be noted that connection type (a) and (b) represent the first hop, while the second hop is depicted by connection type (c).

## III. Problem Formulation

In this section the novel problem formulation will be elaborated. This work takes the advantage of making the IoT network management more context aware, since the requirements of the devices are taken into account while making a decision on the wireless connection type and the data processing unit. Moreover, in this problem formulation, the IoT devices are not only able to indicate their requirements but also prioritize them, meaning that, unlike [3] where the requirements are considered in a binary form, the IoT devices are given the option of ranking their requirements, thus making the solution more accurate. For example, in [3], the IoT devices only indicate that they have security concerns, whereas in this work they are able to indicate how much security is important to them, leading to a more precise decision that makes the devices more satisfied with their requirements.

All the possible connection types and processing unit pairs are given in Fig. 2. Considering the diversity in the available

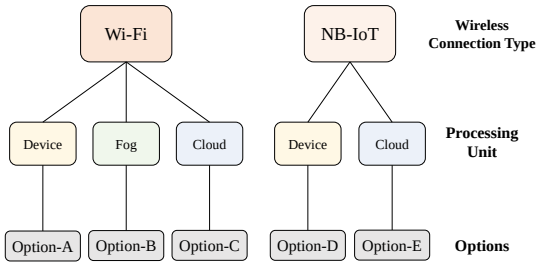


Fig. 2. Possible options for a pair of connection type and processing unit. Note that although device, fog, and cloud processing are all available for Wi-Fi case, NB-IoT includes only device and cloud processing owing to the fact that it does not have/require a Wi-Fi gateway to connect to the Internet.

options provided in Fig. 2, there are multiple components and objectives of the developed optimization problem:

*a) Requirements of IoT devices:* there could be a broad range of requirements based on the use-case, scenario, and conditions, such as data rate, security, latency, etc., to name a few. In this work, response time—as another interpretation of latency—and security are considered as possible requirements of IoT devices. As such, let $K_{\mathrm{r}}$ be the response time requirement of an IoT device, and $\hat{K}_{\mathrm{r}}$ be the response time offered by the selected option. Therefore, in order to satisfy the response time requirement of the IoT device, the following criterion must be met:

$$\hat{K}_{\mathrm{r}} \leq K_{\mathrm{r}}. \tag{5}$$

The security requirement, on the other hand, is captured by eSIM protection, such that IoT devices opt for eSIM protection if data security is of importance to them. In a more formal way, let $K_{\mathrm{s}} \in \{0, 1\}$ be the security requirement of an IoT device, and $\hat{K}_{\mathrm{s}} \in \{0, 1\}$ be the level of data security offered by the selected wireless technology, where 1 indicates the need for eSIM protection and 0 means eSIM protection is unnecessary. In this regard, the following condition is needed in order to meet the security requirement of the IoT device:

$$\hat{K}_{\mathrm{s}} \geq K_{\mathrm{s}}. \tag{6}$$

*b) Energy consumption:* the total energy consumption of the IoT device ($E_{\mathbb{T}}$), which consists of $E_{\mathrm{tx}}$ and $E_{\mathrm{p}}$, should be minimized in order to keep the device alive for a longer time. However, the requirements of IoT devices may undermine this objective, since the number of available options may reduce in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3152381, IEEE Internet of Things Journal



an attempt to satisfy the requirements. Furthermore, the battery level of an IoT device, represented by $\beta$, is also considered in this work provided that it could be another variable in the optimization problem, such that the energy consumption of the device can be prioritized if $\beta$ goes low.

*c) Monetary cost:* it is also important to reduce the monetary costs due to the fact that a smart port scenario, which is more related to trading and business, is targeted in this work. Since lower costs are vital to keep the business sustainable and profitable, the total data processing cost, represented by $M_{\mathbb{T}}$, is supposed to be minimized as well, constituting another objective for the optimization problem. To this end, each processing unit incurs different costs: $M_{\mathrm{d}}$, $M_{\mathrm{f}}$, and $M_{\mathrm{c}}$ which are the processing cost per bit for device, fog, and cloud processing, respectively.

In addition to the requirements and objectives, the optimization problem has strict constraints as well. For example, the available computational capacity of the processing unit is the major constraint, since it can render the selection infeasible. In this regard, let $\hat{K}_{\mathrm{x}} \in \{X_{\mathrm{d}}, X_{\mathrm{f}}, X_{\mathrm{c}}\}$ be the computational capacity of the selected processing unit, where $X_{\mathrm{d}}$, $X_{\mathrm{f}}$, and $X_{\mathrm{c}}$ are the available capacities of device, fog, and cloud, respectively. Then, the following condition must be obeyed to ensure that the selected processing unit has enough capacity for the required amount of data:

$$K_{\mathrm{x}} \leq \hat{K}_{\mathrm{x}}, \tag{7}$$

where $K_{\mathrm{x}}$ is the required data rate of the IoT device. Note that although $K_{\mathrm{x}} = D_{\mathrm{r}}$, the $K_{\mathrm{x}}$ notation is kept here for the sake of consistency.

In addition to the computational capacity, the aforementioned requirements of IoT devices create additional constraints due to the fact that poorly addressed response time and security requirements can make the whole process impractical. For example, meeting the security requirement can be a must for some use-cases, which value data privacy and cannot tolerate data breaches. In this regard, for such use-cases, Options A, B, and C in Fig. 2 are eliminated from the possibilities, since they involve Wi-Fi connectivity that does not offer an eSIM protection.

Moreover, partial offloading—where IoT devices are allowed to offload portions of their raw data to the fog or cloud—is also considered in this work. As such, the amount of data to be offloaded should also be optimized, therefore we propose a joint optimization of connection-processor pair and amount of data to be offloaded. The reasoning behind the partial offloading concept is that it is not sufficient to optimize the best connection-processor pair alone, since this kind of optimization cannot be done properly without considering and optimizing the amount of data to be offloaded given the aforementioned constraints.

In this regard, the overall optimization problem can be written formally as follows:

$$\begin{aligned} \min_{K,\Psi} \quad & E_{\mathbb{T}}(K,\psi), M_{\mathbb{T}}(K,\psi) \\ \text{s.t.} \quad & \hat{K}_{\mathrm{r}} \leq K_{\mathrm{r}}, \\ & \hat{K}_{\mathrm{s}} \geq K_{\mathrm{s}}, \\ & K_{\mathrm{x}} \leq \hat{K}_{\mathrm{x}}, \end{aligned} \tag{8}$$

where $\psi$ is the selected percentage volume of data to be offloaded. $K \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$ is the selected option from Fig. 2, and is a 5-tuple as follows:

$$K = [\hat{K}_{\mathrm{r}}, \hat{K}_{\mathrm{s}}, \hat{K}_{\mathrm{x}}, \hat{E}_{\mathbb{T}}, \hat{M}_{\mathbb{T}}], \tag{9}$$

where $\hat{E}_{\mathbb{T}}$ and $\hat{M}_{\mathbb{T}}$ are the resulting total energy consumption and monetary cost offered by the selected option, $K$, respectively.

## IV. Proposed Scheme

An RL based solution is designed in order to tackle the problem detailed in Section III. The primary motivation of using RL instead of heuristic algorithms, which are also capable of solving such kind of problems, is the computational cost. As such, although heuristics are good options in finding a sub-optimal solution to the many problem in various domains, the nature of wireless communication as well as IoT networks are quite dynamic, and thus the solution process performed by heuristics should be repeated every time when there is a change in the network. In this regard, RL based approaches—if designed appropriately—are generally computationally less demanding which makes them more scalable [45]. In other words, if the environment was static, then then there is a plethora of advanced multi-objective optimization algorithms, including Pareto- and evolutionary-based approaches, etc., that are capable of solving complex problems in different fields; such as finance, electrical, chemical to name a few [46]. However, since the problem-at-hand requires tackling dynamically changing environments—i.e., the physical conditions rapidly change—, such algorithms need to be repeated with a new parameter settings at each change which could be in the order of seconds in terms of time. Given that this kind of implementation would be infeasible in terms of both time/computational complexity and required time for parameter tuning and convergence at each change in conditions, we consider RL as the optimization methodology. Since the experience gained is transferable among different actors and time instances, making the decision process much faster after the initial training process [45], [47], [48].

Furthermore, heuristics are often model-based, meaning that they require an initial knowledge about the environment-of-interest, which is an issue in the context of wireless communication networking as there are multiple random effects in the equation. Even though there are model-based RL approaches available, we designed a model-free RL algorithm in order to resolve the aforementioned challenge of acquiring the model of the environment in advance. Besides, as mentioned in [45]–[48], heuristic implementations would require expertise in tuning the hyperparameters of the algorithms when the environmental conditions change, bringing the cost of qualified

labour from the perspective of mobile network operators. In RL, on the other hand, after the initial design process, there is no need for parameter tuning when the network conditions change as the algorithm learns the experience and transfers accordingly. Lastly, this work is an attempt to demonstrate and prove the applicability of ML in optimizing such problems, opening a new path in automating the IoT networks by eliminating the need for traditional optimization algorithms which might be outdated when the scale of IoT networks and latency requirements are jointly considered.

Given its promising convergence features and capabilities of working in dynamically changing environments, $Q$-learning—one of the most popular RL algorithms—is employed in this work [49]–[51]. In general, $Q$-learning is an algorithm, where an agent interacts with its environment by taking actions and evaluating consequent outcomes—referred to as reward or penalty. In particular, there are four main components of $Q$-learning [49], [50]. 1) *environment:* anything that produces an output for any taken action; 2) *agent:* the entity that takes the actions within the given environment; 3) *action:* the movements that the agent performs to observe the output; 4) *state:* the condition of the agent according to the action taken.

It is a model-free algorithm, meaning that it does not require a prior knowledge about the environment; it rather learns the environment through continuous interactions. In addition to the aforementioned convergence and adaptability features, this model-free characteristics of $Q$-learning also makes it a good candidate for the problem defined and modelled in Section III. Furthermore, it is already proven to perform well in a similar scenario and problem tackled in [3].

The sequence diagram explaining how the proposed methodology can be implemented is depicted in Fig. 3, wherein three main entities—namely, device, fog, and cloud—are included with their subsystems. Starting from the subsystems that are common to all the entities, the processor is primarily responsible for data processing, meaning that it tries to extract the meaningful information from the data. Transceivers are for wireless communications, that is, they are used to transmit and receive data from other entities and convey the data to the remote controller, which analyzes the data received from the IoT device and takes the required action. In the case of wild fire, for example, the IoT device may sense[2] some physical parameters—say corbonmonoxide—from the environment (forest in this case) and transmit it to the remote controller (e.g., firefighters or local authorities), which then decides the kind of action to take in order to prevent and/or fight the fire.

The local controller in the IoT device orchestrates all the decision making process by training the designed RL algorithm and taking the final decision. The local controller, as seen in Fig. 3, collects all the necessary information from all the entities and subsystems in order to train the algorithm. It is worth noting the the QoS requirements and their corresponding weights are acquired from the remote controller, since it is the one determining such parameters. In other words, the same IoT device can be used for various use cases, and each use case might have distinctive requirements, which are set by the remote controller according to different variables, including the characteristics of the use case, environmental conditions, etc.

[2]Note that the sensing unit in the IoT device is not depicted in Fig. 3 in order not to pollute the figure as it is not related to the discussion.

### A. Actions and States

In the considered scenario, the IoT devices are supposed to choose one of the options presented in Fig. 2 to conduct their connection and data processing tasks. In this regard, these options could also be treated as an action set for the developed $Q$-learning algorithm. However, Options B, C, and E include either fog or cloud processing, meaning that IoT devices are supposed to offload their collected data to the fog or cloud for processing if they choose one of these options. Provided that partial offloading is also captured in this work, the amount of data to be offloaded should also be optimized, and thus considering the options in Fig. 2 alone as the action set would not be adequate for this objective. Therefore, the action set is determined as follows:

$$\mathbb{A} = \mathbb{K} \times \Psi, \tag{10}$$

where $\times$ represents a Cartesian product, $\mathbb{K}$ is the set of all the possible options included in Fig. 2, such that $\mathbb{K} = \{A, B, C, D, E\}$, and $\Psi$ is the set of all the possible options for offloading percentage, such that

$$\Psi = \{\tau m \mid m \in \{0, 1, 2, ..., 20\}\}, \quad \psi \in \mathbb{R}^{+}, \tag{11}$$

where $m$ is discretisation factor that is used to discretise the continuous values from $0\%$ to $100\%$, and $\tau$ is the resolution of the discretisation process. $\tau$ can take any value in $\mathbb{R}^{+}$, but with a trade-off: the smaller it gets, the higher the resolution is, resulting in a more precise decision. However, smaller $\tau$ creates an additional computational burden, since it increases $|\mathbb{A}|$, which is the cardinality of $\mathbb{A}$. Without loss of generality, $\tau = 5$ is taken in this work, since it provides a sufficient resolution without significantly increasing the computational complexity. Note that Options A and D do not have $\Psi$ parameter, because they do not perform any offloading at all. Using this phenomena, the actions set in (10), can be rewritten as

$$\mathbb{A} = \begin{cases} \mathbb{K} \times \Psi, & \text{if } K \in \{\mathrm{B}, \mathrm{C}, \mathrm{E}\} \\ \mathbb{K}, & \text{if } K \in \{\mathrm{A}, \mathrm{D}\}. \end{cases} \tag{12}$$

Since each of these actions also defines the state of the agent, the state space, denoted by $\mathbb{S}$ is designed to be the same with the action space, such that $\mathbb{S} = \mathbb{A}$.

### B. Penalty Function

In the proposed $Q$-learning algorithm, two novel prioritization concepts are adopted:

*a) Prioritization of requirements:* IoT devices are allowed to prioritize their requirements using a weighting mechanism, such that $w = \{w_{\mathrm{r}}, w_{\mathrm{s}}\}$, where $w_{\mathrm{r}} \in \mathbb{R}$ and $w_{\mathrm{s}} \in \mathbb{R}$ are the weight parameters for response time and security requirements, respectively. IoT devices are asked to rate the strictness of their requirements, such that lower values indicate that the requirement is loose, while higher values yield a stricter requirement.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3152381, IEEE Internet of Things Journal



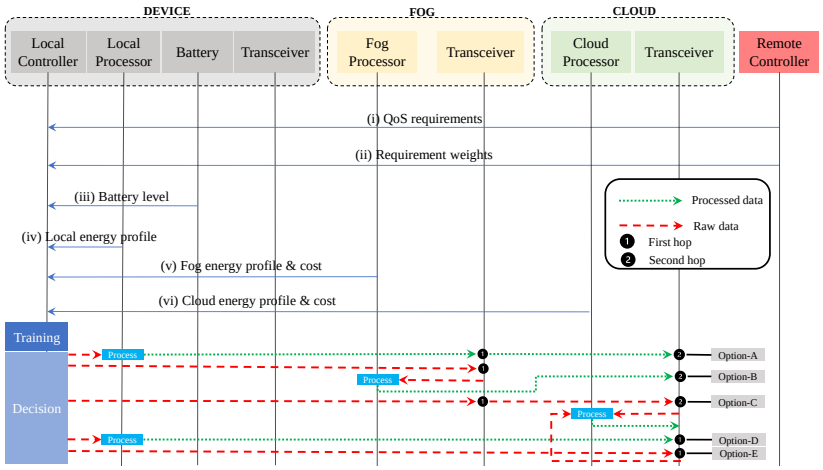


Fig. 3. Sequence diagram showing the information flow between an IoT device, fog, and cloud. Note that the data is always transmitted or received by the transceiver of the concerned entity before it is conveyed to the processor internally.

*b) Prioritization of energy consumption and monetary cost:* the total energy consumption and monetary cost are subject to prioritization as well. However, unlike the requirement prioritization case, where IoT devices control their weights ($w_\mathrm{r}$ and $w_\mathrm{s}$), the energy consumption and cost prioritization are triggered by the network. Moreover, this mechanism is linked to the battery level of an IoT device, $\beta$. More specifically, a certain threshold, denoted by $\beta_T \in \mathbb{R}$, is determined for the battery level, and

- if the battery level of an IoT device $i$ is above or equal to the threshold, such that

$$\beta_i \geq \beta_T, \tag{13}$$

the monetary cost is prioritized.

- if, on the other hand, the battery level of an IoT device $i$ is less than the threshold, such that

$$\beta_i < \beta_T, \tag{14}$$

then the energy consumption is prioritized.

Based on that, the overall penalty function for the developed $Q$-learning algorithm is formulated as follows:

$$\mathfrak{C}_Q = \Theta_\mathrm{r} + \Theta_\mathrm{s} + \Theta_\mathrm{c} + \Theta_\mathrm{m} + w_\mathrm{e}\hat{E}_\mathbb{T}, \tag{15}$$

where $\Theta_\mathrm{r}$, $\Theta_\mathrm{s}$, $\Theta_\mathrm{c}$, and $\Theta_\mathrm{m}$ are the penalty elements for response time, security, computational capacity, and monetary cost, respectively:

$$\Theta_\mathrm{r} = \begin{cases} \Omega^{w_\mathrm{r}} + \hat{K}_\mathrm{r}, & \text{if } \hat{K}_\mathrm{r} > K_\mathrm{r} \\ 0, & \text{otherwise,} \end{cases} \tag{16a}$$

$$\Theta_\mathrm{s} = \begin{cases} \Omega^{w_\mathrm{s}}, & \text{if } \hat{K}_\mathrm{s} < K_\mathrm{s} \\ 0, & \text{otherwise,} \end{cases} \tag{16b}$$

$$\Theta_\mathrm{c} = \begin{cases} \Omega^{w_\mathrm{c}}, & \text{if } K_\mathrm{x} > \hat{K}_\mathrm{X} \\ 0, & \text{otherwise,} \end{cases} \tag{16c}$$

$$\Theta_\mathrm{m} = w_\mathrm{m} M_{\mathbb{T},S} \psi K_\mathrm{x}, \tag{16d}$$

where $\Omega \in \mathbb{R}$ is the global penalty factor, and $w_\mathrm{c} \in \mathbb{R}$ is the penalty factor incurred when the computational capacity is exceeded. Note that $w_\mathrm{c} > \max\{w_\mathrm{r}, w_\mathrm{s}\}$, since the computational capacity is a physical constraint that cannot be breached.

$w_\mathrm{e}$ and $w_\mathrm{m}$ are the weights for energy consumption and monetary cost set by the network, such that

$$w_\mathrm{e} = \begin{cases} w_\mathrm{e}^+, & \text{if } \beta_i < \beta_T \\ w_\mathrm{e}^-, & \text{otherwise,} \end{cases} \tag{17a}$$

$$w_\mathrm{m} = \begin{cases} w_\mathrm{m}^+, & \text{if } \beta_i \geq \beta_T \\ w_\mathrm{m}^-, & \text{otherwise,} \end{cases} \tag{17b}$$

where $x^+$ and $x^-$ represents the high and low values of $x$, respectively.

## V. Performance Evaluation

In this section, the proposed RL approach is implemented in a simulation environment, as illustrated in Fig. 1, using the simulation parameters defined in Table I.

### A. Benchmarking

In this subsection, we introduce the benchmark scenarios that we compared our proposed approach with. It is worth noting that the methods mentioned here are some fixed combinations of wireless connectivity and data processing unit, since other methodologies, such as heuristics, are hard to implement for the problem detailed in Section III due to the fact that they need the model of the environment in advance. As such a model cannot be available for IoT networks because of multiple random effects that change the environmental parameters very abruptly and frequently (refer to Section IV for a more detailed discussion on this), hence the use of heuristics as a benchmark is not appropriate. Such kind of benchmarking makes a weak assumption of having the model of the environment in advance, which would violate the reality of the work.

Moreover, given that the primary objective of this work is to demonstrate the feasibility and compatibility of RL to the problem-at-hand, we do not deem it necessary to compare different RL algorithms, such as $Q$-learning and SARSA. In

TABLE I
SIMULATION PARAMETERS

| Parameter | Value | Description |
|---|---|---|
| | **Communication** | |
| $\delta$ | 3 | Path loss exponent |
| $d_0$ | 10 m | Reference distance |
| $\sigma$ | 8 dB | Standard deviation |
| $c$ | $3 \times 10^8$ m/s | Speed of light |
| $f_{\text{c,a}}$ | 2.4 GHz | Carrier frequency for IEEE 802.11g |
| $f_{\text{c,b}}$ | 1700 MHz | Carrier frequency for NB-IoT |
| $f_{\text{c,c}}$ | 1800 MHz | Carrier frequency for LTE |
| $\mathcal{N}_0$ | $-204$ dbW/Hz | Noise density |
| $T$ | $1/N$ s | Time period |
| $B$ | 180 kHz | Bandwidth |
| $F$ | 2 | IEEE 802.11g retransmission rate |
| $r_{\text{enb}}$ | 1 km | Coverage radius of eNB |
| $r_{\text{wifi}}$ | 30 m | Coverage radius of eNB |
| | **General** | |
| $N_{\text{iot}}$ | 10 | Number of IoT devices |
| $\epsilon$ | $5 \times 10^{-6}$ J | Energy per computation cycle |
| $W$ | 8 | Number of bits per DE |
| $\chi_{\text{d}}$ | 30 kbps | Computational capacity of device |
| $\chi_{\text{f}}$ | 100 kbps | Computational capacity of fog |
| $\chi_{\text{c}}$ | 10 Mbps | Computational capacity of cloud |
| $X_{\text{d}}$ | 100 | (Device) Comp. cycles per DE |
| $X_{\text{f}}$ | 10 | (Fog) Comp. cycles per DE |
| $X_{\text{c}}$ | 1 | (Cloud) Comp. cycles per DE |
| $M_{\text{d}}$ | $10^{-4}$ AC | (Device) Cost of processing per bps |
| $M_{\text{f}}$ | $10^{-1}$ AC | (Fog) Cost of processing per bps |
| $M_{\text{c}}$ | 1 AC | (Cloud) Cost of processing per bps |
| $\mho$ | 200 | Data compression rate |
| $\beta_T$ | 30% | Threshold for battery level |
| | **$Q$-learning** | |
| $\alpha$ | 0.5 | Learning rate |
| $\varphi$ | 0.9 | Discount factor |
| $\varepsilon$ | 0.8 | Chance of choosing random action |
| $N_{\text{ep}}$ | $10^3$ | Number of episodes |
| $N_{\text{it}}$ | $10^3$ | Number of iterations per episode |
| $\Omega$ | 10 | Global penalty factor |
| $w_{\text{c}}$ | 5 | Penalty of exceeding comp. capacity |
| $w_{\text{e}}^+$, $w_{\text{e}}^-$ | 10, 1 | High, low values of $w_{\text{e}}$ |
| $w_{\text{m}}^+$, $w_{\text{m}}^-$ | 10, 0 | High, low values of $w_{\text{m}}$ |

other words, the motivation of this work is not comparing RL algorithms; instead, revealing that RL can be employed for this type of problems, mitigating the computational burden and model dependence of other optimization methods. However, it is worth mentioning the rationale behind choosing $Q$-learning over SARSA in this work: since $Q$-learning is an off-policy algorithm and it acquires straightforwardly the optimal policy, its analysis is more simplified and it converges early [52]. Furthermore, considering the cliff walking example in [52], SARSA is more conservative in taking actions, and thus it is preferable in problems whose costs cannot be tolerated. Given that the cost of choosing a sub-optimal or undesired option would not make a serious consequence in the case of the problem formulated in Section III, it is considered safe to select $Q$-learning algorithm. Nonetheless, it should be noted that such selection would be trivial for this problem, as SARSA and $Q$-learning are quite similar algorithms other than being on-policy and off-policy respectively, SARSA would also be expected to produce similar results as $Q$-learning [52], [53].

The implementation of RL with value-function approximation (VFA), including deep RL and linear VFA, is beyond the scope of this work as it can be implemented for more complex scenarios and requires a completely different approach. Besides, albeit its proven strong capabilities—especially in large-scale problems—RL with VFA brings an extra implementational complexity on the grounds that the inclusion of either linear function approximation or a neural network (in the case of deep RL) necessitates more hyperparameter tuning as the feature selection, which plays a crucial role in the performance of such algorithms [52], will also be an issue. In this regard, intuitively, conventional RL methods can be preferable if the search space (e.g., number of states) or complexity of the problem is limited, whereas RL with VFA would be a choice in the case of increase state space in the design of the algorithm.

Lastly, even though this work originates from [3], it is quite difficult to use it as a benchmark given that there are structural changes made in this current work. Although the reasoning behind this is detailed in Section I-B, it is better to have a brief summary in order to ensure the completeness of this subsection. The main reason is that a weighting mechanism—discussed in Section IV—is included in this work, making the comparison of both works very hard; such that although the work in [3] considers the requirements of IoT devices in a binary fashion, this work allow them to prioritize one requirement over another. Therefore, the comparison would not be possible and fair.

There are some relevant and strong works in the literature, such as [35] and [36]. Although those works are similar to our work from a top-level view, as the context-awareness is investigated in both [35] and [36], the characteristics and focus of such works are quite different than our work. For example, in [35], the authors tried to automate the process of event detection, and they developed a detection mechanism for the occurrence of event mismatch during the process of coordinating the services of the IoT devices. The authors in [36], on the other hand, investigates an intelligent context-aware framework for QoS management during video transmission. In addition to the characteristics, the formulated and investigated problem of our work is distinct, making it hard to compare it with such exiting works.

Having said all this, in order to obtain the benchmark scenarios, first, the IoT devices are categorized into two groups—with equal number of members—based on their wireless connection type as follows: the devices with NB-IoT connection and the ones with Wi-Fi connection. The former group is called Group-X, while the latter group represents Group-Y. Then, these groups are mapped to the available processing units, and all the possible combinations are considered as benchmark scenarios, as given in Table II.

### *B. Performance Metrics*

The obtained results are evaluated with five different metrics, namely energy consumption, monetary cost, response time, security dissatisfaction, and a novel joint metric that is specifically developed for this work. Moreover, these metrics are presented in a comparative fashion, where the performances of the benchmark methods—provided in Table II—are compared to the proposed RL-based method.

The performance metrics are elaborated as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3152381, IEEE Internet of Things Journal



TABLE II
LIST OF FIXED BENCHMARK SCENARIOS WITH CONNECTION TYPES AND DATA PROCESSING UNIT

| Scenario | Group-X | Group-Y |
|---|---|---|
| $\text{Sc}_\text{A}$ | Device | Device |
| $\text{Sc}_\text{B}$ | Cloud | Device |
| $\text{Sc}_\text{C}$ | Device | Fog |
| $\text{Sc}_\text{D}$ | Cloud | Fog |
| $\text{Sc}_\text{E}$ | Device | Cloud |
| $\text{Sc}_\text{F}$ | Cloud | Cloud |

*a) Energy consumption:* the accumulated energy consumption of all the IoT devices, which is caused by data processing and transmission, is calculated by

$$\dot{E}_{\mathbb{T}} = \sum_{i=1}^{N_{\text{iot}}} E_{\mathbb{T},i}, \tag{18}$$

where $N_{\text{iot}}$ is the number of IoT devices, and $E_{\mathbb{T},i}$ is the total energy consumption of $i^{\text{th}}$ IoT device.

*b) Monetary cost:* the aggregated monetary cost that the IoT devices are charged for data processing:

$$\dot{M}_{\mathbb{T}} = \sum_{i=1}^{N_{\text{iot}}} M_{\mathbb{T},i}, \tag{19}$$

where $M_{\mathbb{T},i}$ is the total monetary cost for $i^{\text{th}}$ IoT device.

*c) Response time:* the accumulated response time for all the IoT devices:

$$\dot{R}_{\mathbb{T}} = \sum_{i=1}^{N_{\text{iot}}} R_{\mathbb{T},i}, \tag{20}$$

where $R_{\mathbb{T},i}$ is the total response time for $i^{\text{th}}$ IoT device, and given by

$$R_{\mathbb{T},i} = \begin{cases} \hat{K}_\text{r} - K_\text{r}, & \text{if } \hat{K}_\text{r} > K_\text{r} \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

*d) Security dissatisfaction:* the number of IoT devices, whose security requirements are not satisfied:

$$\dot{N}_{\text{dis},\mathbb{T}} = \sum_{i=1}^{N_{\text{iot}}} v_{\text{dis},i}, \tag{22}$$

where $v_{\text{dis},i}$ is the security dissatisfaction variable for the IoT device $i$, such that

$$v_{\text{dis}} = \begin{cases} 0, & \text{if } \hat{K}_\text{s} \geq K_\text{s} \\ 1, & \text{otherwise.} \end{cases} \tag{23}$$

*e) Joint metric:* the combination of all the aforementioned metrics, such that

$$J = \gamma \hat{\ddot{E}}_{\mathbb{T}} + \eta \hat{\ddot{M}}_{\mathbb{T}} + \zeta \hat{\ddot{R}}_{\mathbb{T}} + \kappa \hat{\ddot{N}}_{\text{dis},\mathbb{T}}, \tag{24}$$

where $\hat{\ddot{E}}_{\mathbb{T}}$, $\hat{\ddot{M}}_{\mathbb{T}}$, $\hat{\ddot{R}}_{\mathbb{T}}$, and $\hat{\ddot{N}}_{\text{dis},\mathbb{T}}$ are the normalised versions of $\ddot{E}_{\mathbb{T}}$, $\ddot{M}_{\mathbb{T}}$, $\ddot{R}_{\mathbb{T}}$, and $\ddot{N}_{\text{dis},\mathbb{T}}$, respectively. The normalisation (feature scaling) operation is performed here in order to keep the scale of the each metric in the same range, thus preventing one from dominating another. $\gamma$ (unitless), $\eta$ in (Joule/AC), $\zeta$ in (Joule/s), and $\kappa$ in (Joule), where $\gamma = \eta = \zeta = \kappa = 1$, are coefficients used to make the units of the elements of $J$ in (24) the same. Note that AC in the unit of $\eta$ stands for arbitrary currency.

### C. Battery Regimes

Two different battery regimes, namely low and high, are considered in this work, and results are produced separately in order to observe the behaviours of the proposed method[3].

*a) Low-battery regime:* when the energy level of the battery of a particular IoT device is under a certain threshold ($\beta_i < \beta_T$), the device is considered to be in a low-battery regime, and the proposed $Q$-learning algorithm starts to prioritize the total energy consumption of the device along with meeting the response time and security requirements. It is worth noting that, the priority between the requirements and the energy consumption is determined by the weights of the requirements ($w$), such that if the requirements are strict (i.e., with high weights), then they become more important than the energy consumption, and vice versa. The underlying idea here is that if the device is strict in any requirement, it means that it is unwilling to compromise on that. Moreover, the monetary cost is completely discarded, and therefore IoT devices are expected to be charged more when they are in this regime.

*b) High-battery regime:* to be in the high-battery regime, the remaining energy in the battery of an IoT device should be above the aforementioned threshold for the battery level ($\beta_i \geq \beta_T$). In this high-battery regime, the monetary cost is valued significantly, and—similar to the low-battery regime—the importance of the requirements are determined by their correspondent weights, and energy consumption is loosely prioritized[4].

### D. Results and Discussions

Fig. 4 demonstrates the performances of all the methods including the proposed one and the benchmarks when the entire set of IoT devices are in the low-battery regime. Moreover, both response time and security requirements of the IoT devices are prioritized in a rigid way, where both $w_\text{r}$ and $w_\text{s}$ are ranked as 3. In the following paragraphs, there will be individual discussions on the results for each performance metric:

*a) Energy consumption:* owing to the low-battery regime, it was expected that the proposed method will perform well in minimizing the energy consumption. This is due to the fact that, when the battery level is under the threshold, the energy consumption component in the penalty function in (15) is prioritized through its weight ($w_\text{e}$) by setting it to its high value ($w_\text{e}^+$), as seen in (17a). This is a reasonable behaviour, since the energy consumption becomes more crucial when the battery is about to be depleted, which in turn interrupts the communication until the battery is recharged or replaced.

The energy consumption results for the benchmark scenarios are also worth discussing. There are two main components of the overall energy consumption, namely data processing and transmission. Based on the channel conditions and distance

[3]It is worth noting here that the benchmark methods do not consider the remaining battery level of IoT devices, and thus their behaviours are not expected to change with the battery level.

[4]It is important to mention that unlike the low-energy regime, where the monetary cost is completely discarded, the energy consumption is still considered in the high-battery regime albeit with much less importance, since energy consumption of an IoT device should always be in the equation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3152381, IEEE Internet of Things Journal



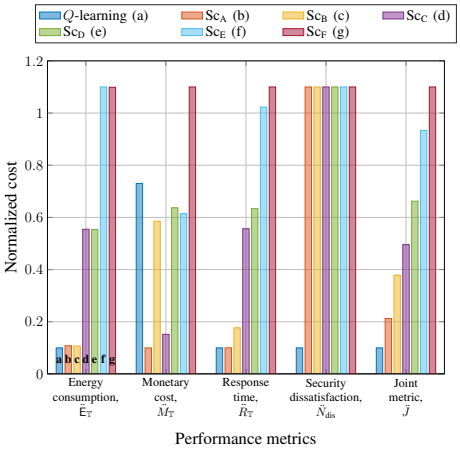


Fig. 4. Performances of the proposed method (named as $Q$-learning in the legend) and the benchmark scenarios in terms of considered metrics when all the IoT devices are in the low-battery regime. The response time and security requirements are strictly prioritized, such that $w_{\mathrm{r}} = w_{\mathrm{s}} = 3$. Note that the shown results are normalised values in the range of $[0, 1]$ with an offset of 0.1, which is used only due to visualisation purposes. However, the results are discussed in the text without considering the offset value.

between a transmitter and a receiver, the transmission energy consumption can prevail over the processing energy consumption, or vice versa [32]. However, receiver sensitivity, which is captured by the link margin in this work, also plays an important role provided that it is directly correlated to the required received power, which in turn affects the required transmit power. Therefore, the connection type (i.e., Wi-Fi, NB-IoT, and LTE) is also involved in the breakdown of the total energy consumption. In this work, due to the random distribution of the IoT devices and the Wi-Fi gateways at each repeat[5] in the simulations, it is avoided for one of the aforementioned two components of energy consumption to dominate the other[6]. Nonetheless, owing to the link margin assumption, the NB-IoT connection happened to be the least energy consuming connection type in most of the cases. Besides, in terms of the energy consumption, the descending order of the tasks is as follows: Wi-Fi connection, data processing, NB-IoT connection. Note that the energy consumption difference between the Wi-Fi connection and data processing happened to be much more than the difference between the data processing and NB-IoT connection. This may seem counter-intuitive given that the Wi-Fi gateway is much closer to the IoT devices than the eNB. Nonetheless, since the receiver sensitivity of NB-IoT is less than Wi-Fi and LTE, it happens to result in less energy consumption owing to the less required transmit power, which is caused by the less path-loss.

In this regard, since all the options include the same number of IoT devices with NB-IoT (Group-X) and Wi-Fi (Group-Y), there is no difference in terms of the number of connection types. However, the point that matters here is the processing unit. On one hand, when an IoT devices is connected through Wi-Fi, device processing is expected to consume less energy than cloud processing, with fog processing in between. On the other hand, when the device is connected through NB-IoT, device processing consumes more energy than cloud processing, with fog processing in between. In addition, the maximum energy consumption with Wi-Fi connection is expected to be more than the maximum energy consumption with NB-IoT connection. Based on that, $\mathrm{Sc_B}$ resulted in the least energy consumption due to the fact that the devices with Wi-Fi connection process the data locally, while the devices using NB-IoT connection performs cloud processing. $\mathrm{Sc_B}$ is followed by $\mathrm{Sc_A}$, since it also processes the data locally for Wi-Fi connections. The small difference between $\mathrm{Sc_A}$ and $\mathrm{Sc_B}$ comes from the difference between the cloud and device processing for NB-IoT connections. In a similar fashion, $\mathrm{Sc_E}$ consumed the maximum energy among the benchmark methods, since it employs cloud processing for Wi-Fi connected devices and device processing for NB-IoT connected devices. In summary, cloud processing is less energy consuming for NB-IoT, whereas it is more energy consuming for Wi-Fi. Moreover, as mentioned earlier, the maximum energy consumption with Wi-Fi is much more than that of NB-IoT.

*b) Monetary cost:* as a result of the low-battery regime, the proposed algorithm inclines towards being much looser in monetary cost, and thus it is not expected to be competitive in this metric. Based on that, as expected, the proposed method performed worse than all the benchmark methods other than $\mathrm{Sc_F}$, which purely includes cloud processing.

Similarly, the results of the benchmark methods are obtained as expected: the cloud is the most expensive means of data processing, followed by the fog, and the device (local), respectively, such that $M_{\mathrm{d}} < M_{\mathrm{f}} < M_{\mathrm{c}}$. Thus, the scenarios with device processing (e.g., $\mathrm{Sc_A}$) resulted in less amount of monetary cost, whereas the scenarios with cloud and/or fog processing (e.g., $\mathrm{Sc_D}$ and $\mathrm{Sc_F}$) become the most expensive ones. The results obtained in Fig. 4 confirms this statement.

*c) Response time:* the proposed method performed quite well in response time by outperforming all the benchmark methods. Provided that $w_{\mathrm{r}} = 3$, which yields a strict prioritization of response time, it was expected that the proposed method will reduce the response time. Considering (15) and (16a) together, the effect of response time in the penalty function in (15) increases with growing $w_{\mathrm{r}}$. As such, the primary objective of the developed $Q$-learning algorithm is to minimize the overall penalty, hence, the response time satisfaction becomes key for this objective.

Similar to the previous metrics, the benchmark methods also performed as anticipated. As discussed in Section II-C, response time is a function of the number of hops ($N_{\mathrm{h}}$), the computational power ($X$), retransmission rate ($F$), the data volume ($D \in \{D_{\mathrm{r}}, D_{\mathrm{p}}\}$), and compression rate ($\mho$). Given that NB-IoT connection has only one hop and lower retransmission rate than Wi-Fi connection, the scenarios with NB-IoT resulted in a comparatively less response time. Similarly, from (4), albeit suffering from a higher computational time, device processing is also preferable due to the processed data

[5]The simulations are repeated for 25 times to avoid random effects.

[6]This is ensured via random process; the path-loss can sometimes be huge, which results in the transmission energy consumption surpassing the data processing energy consumption, or vice versa. However, this effect is minimized by averaging out the simulation repeats.

transmission, which entails $\mho$ times less data volume[7]. Thus, for example, a cloud processing with a Wi-Fi connection would result in the highest response time owing to: 1) Wi-Fi connection, which has higher retransmission rate; 2) two hops taken; and 3) raw data transmission. In this regard, $\text{Sc}_\text{A}$ resulted in the least response time, whereas $\text{Sc}_\text{F}$ caused the highest response time among all the methods.

*d) Security dissatisfaction:* similar to the response time case, security requirement is also strictly prioritized in these simulation campaigns by setting $w_\text{s}$ to 3. Considering (15) with (16b), higher values of $w_\text{s}$ incurs more cost by increasing $\Theta_\text{s}$, which in turn inflates the penalty function, $\mathfrak{C}_Q$ in (15). Given that the objective of the designed $Q$-learning algorithm is to minimize $\mathfrak{C}_Q$, satisfying the security requirement of IoT devices becomes crucial for the proposed algorithm, as $\Theta_\text{s}$ returns 0 when the requirement is met. To this end, the developed $Q$-learning algorithm achieved a significant reduction in terms of the security dissatisfaction when compared to the benchmark methods.

One can question the equal results of the benchmark methods, but there is a rationale behind it: half of the IoT devices are connected with NB-IoT (Group-X in Table II), while the other half communicates through Wi-Fi (Group-Y in Table II), and—as discussed in Section III—the security requirement is captured by the need for an eSIM card, which is only available for NB-IoT connections. Thus, those connected with NB-IoT do not have any issue with the security dissatisfaction, since they always meet the requirements due to their eSIM card availability. Those connected through Wi-Fi, on the other hand, cannot respond to the eSIM card requirement. Based on that, the number of IoT devices with security dissatisfaction always equals to the number of IoT devices that: 1) is connected through Wi-Fi and 2) requires eSIM protection. Thus, the number of dissatisfied devices is the same for all the benchmark methods.

*e) Joint metric:* while each individual previous metric reflects the behaviours of the methods in a specialized manner, this joint metric summarizes the overall performances. Therefore, this metric can be seen as a holistic cost of each method, which demonstrates how they perform when all the previous metrics are combined. The proposed method performed quite well and outperformed all the benchmark methods in different scales. The reasoning behind this is that there are four metrics in total other than joint metric, and the proposed method outperformed all the benchmark methods in three of them. Therefore, it is quite reasonable that the proposed method performed the best in terms of the joint metric. Similarly, the benchmark methods responded to the joint metric according to their results in each individual metric, namely: energy consumption, monetary cost, response time, and security dissatisfaction.

The comparison between the cases of requirement deprioritization ($w_\text{r} = w_\text{s} = 0$)—which will be referred to as low-battery requirement-aware (LBRA) hereafter—and prioritization ($w_\text{r} = w_\text{s} = 3$)—which will be referred to as low-battery requirement-unaware (LBRU) hereafter—under the low-battery regime is demonstrated in Fig. 5. The loss calculations for each metric is performed as follows:

$$L = \frac{\Delta_{Q,\text{LBRU}} - \Delta_{Q,\text{LBRA}}}{\Delta_{Q,\text{LBRU}}}, \tag{25}$$

where $\Delta_{Q,\text{LBRA}}$ and $\Delta_{Q,\text{LBRU}}$ are the differences between the values obtained via the proposed method and the minimum value obtained with the benchmark methods for LBRA and LBRU, respectively, such that

$$\Delta_{Q,\text{LBRA}} = V_{Q,\text{LBRA}} - \min(V_\text{b}), \tag{26a}$$

$$\Delta_{Q,\text{LBRU}} = V_{Q,\text{LBRU}} - \min(V_\text{b}), \tag{26b}$$

where $V_\text{b} \in \{V_{Sc_\text{A}}, V_{Sc_\text{B}}, V_{Sc_\text{C}}, V_{Sc_\text{D}}, V_{Sc_\text{E}}, V_{Sc_\text{F}}\}$ are the obtained values in the aforementioned metrics with the benchmark methods. $V_{Q,\text{LBRA}}$ and $V_{Q,\text{LBRU}}$ are the obtained values in the aforementioned metrics with the proposed method for LBRA and LBRU, respectively.

There is an important caveat to note here: positive values of $L$ calculated through (25) indicate loss, where the LBRU performed worse than LBRA, and the negative values of $L$ indicate gain, where LBRU performed better than LBRA. From the findings in Fig. 5, it is obvious that LBRA outperformed LBRU in all the metrics other than the energy consumption. Starting from response time and security dissatisfaction, the results are quite expected provided that both of the requirements are prioritized in LBRA with weight values of $w_\text{r} = w_\text{s} = 3$. Hence they are given a special care in LBRA when (15) is considered together with (16a) and (16b). Although the scale difference between the response time and security dissatisfaction is worth discussing, however, it is better to first analyze the energy consumption results, which will then be more beneficial to explain such difference.

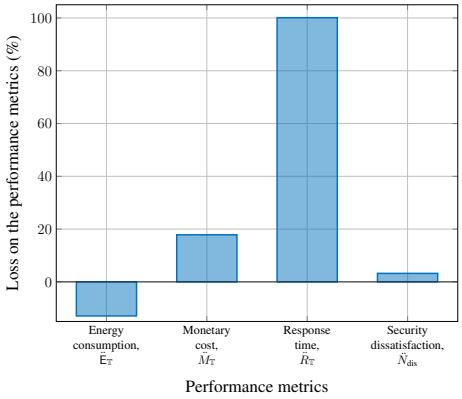


Fig. 5. Performances of the proposed method in terms of considered metrics when all the IoT devices are in the low-battery regime. The results show the percentage loss when LBRU ($w_\text{r} = w_\text{s} = 0$) is compared to LBRA ($w_\text{r} = w_\text{s} = 3$). Positive values yield loss; i.e., the superiority of LBRA, while negative values yield gain; i.e., the superiority of LBRU.

As seen from Fig. 5, the energy consumption is the only metric that LBRU performed better than LBRA. The rationale behind this is that energy consumption is the only focus of LBRU given that: 1) the IoT devices are in the low-battery regime, and 2) both $w_\text{r}$ and $w_\text{s}$ are set to 0. Thus, LBRU does

[7]The response time of the options (connection and processing) would alter for different values of $\mho$, $F$, and $X$. Therefore, the response time of the benchmark methods would change accordingly, but the discussions here are based on the current assumptions for $\mho$, $F$, and $X$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3152381, IEEE Internet of Things Journal



not consider any other metric other than energy consumption, which is the reason why it managed to outperform LBRA.

Considering this rationale as a base, it would be more straightforward to explain the obtained results from the other metrics, as the reasoning for all the results are linked to each other and they all arise from this base. There are additional supporting facts as follows:

- LBRA aims at minimizing the energy consumption, but with the response time and security constraints. In other words, the energy consumption is minimized after the response time and security requirements of the devices are satisfied. Provided that Options A, D, and E happen to result in minimal response time, meaning that they could be the options for the devices with low response time requirements.
- Similarly, Options D and E are the ones that provide the eSIM protection, which means an IoT device should select one of these if it has security concerns.

Based on that, Options D and E followed by A are the intersection ones that are most likely to be selected when both response time and security is prioritized. For example, the response time requirement of an IoT device can only be satisfied with Option A and D, but Option E can result in less energy consumption. In such cases, LBRA would select Option A or D that has the least monetary cost due to device processing, whereas LBRU goes for Option E that results in the highest monetary cost due to cloud processing. As such, LBRA is more likely to perform better in terms of monetary cost, while LBRU is better in energy consumption. These explain the performance differences between LBRA and LBRU in terms of energy consumption and monetary cost.

It is now better to turn back to the discussion on the scale difference between the response time and security dissatisfaction. There are two points to consider:

- Switching among Wi-Fi options (i.e., A, B, and C) and among NB-IoT options (i.e., D and E) does not change the security dissatisfaction results, but it changes the response time. In other words, selecting a different option from Fig. 2 definitely changes the response time behaviour, whereas the security behaviour might remain the same. In this regard, there is more room for response time to alter than that of the security behaviour.
- Options D and E have NB-IoT connection, which was already discussed in this section as being the least energy consuming one in majority of the cases, and thus the agent would be more prone to stick with them. Owing to the fact that Options D and E are with NB-IoT connection, both LBRA and LBRU would more possibly select one of these options, which would have an impact on the response time but the security behaviour remains unaffected.

Due to these reasons, it is quite reasonable that LBRA outperforms LBRU more significantly in response time than that in security dissatisfaction.

Table III reveals the performance comparison between LBRA and the case when the IoT devices are in the high-battery regime and their requirements are fully prioritized ($w_{\mathrm{r}} = w_{\mathrm{s}} = 3$)—which will be referred to as high-battery requirement-aware (HBRA) hereafter. The results show the percentage loss when HBRA is compared to LBRA. Positive values yield loss; i.e., the superiority of LBRA, while negative values yield gain; i.e., the superiority of HBRA. Note that only energy consumption and monetary cost results are presented in Table III, since LBRA and HBRA performed equally well in response time and security dissatisfaction given that they both fully prioritize the device requirements.

TABLE III
LOSS ON THE GIVEN PERFORMANCE METRICS

| **Energy consumption** $\ddot{\mathrm{E}}_{\mathbb{T}}$ | **Monetary cost** $\ddot{M}_{\mathbb{T}}$ |
|---|---|
| 54.1525% | -96.7937% |

The findings in Table III show that LBRA reduced the energy consumption of HBRA by around $53\%$, while HBRA managed to decrease the monetary cost of LBRA by around $97\%$. These results are quite expected because LBRA focuses only on the energy consumption, while completely ignoring the monetary cost reduction[8]. HBRA, on the other hand, aims at minimizing the monetary cost rather than the energy consumption, since the battery levels of IoT devices are high.

Although these explanations are adequate to understand why they surpass each other in the two considered metrics, there is still room for clarification for the question of why the scales of the outperformance are quite different from each other. Considering (15) together with (16d), (17b), and (17a), it is obvious that both energy consumption and monetary cost have their own impacts in $\mathfrak{C}_Q$. On one hand, when the IoT devices are in the low-battery regime, the weight for energy consumption ($w_{\mathrm{e}}$) takes its higher value ($w_{\mathrm{e}}^{+}$), which is set to 10, while the weight for monetary cost ($w_{\mathrm{m}}$) takes its lower value ($w_{\mathrm{m}}^{-}$), which is set to 0. On the other hand, when the IoT devices are in the high-battery regime, $w_{\mathrm{e}}$ is set to its lower value as $w_{\mathrm{e}}^{-} = 1$, while $w_{\mathrm{m}}$ is set to its higher value as $w_{\mathrm{e}}^{-} = 10$. This means that

- when the IoT devices are in the low-battery regime, the algorithm fully focuses on the energy consumption minimization while completely discarding the monetary cost reduction; but
- when the IoT devices are in the high-battery regime, the algorithm mainly takes care of the monetary cost reduction, but without completely ignoring the energy consumption minimization.

This is done because energy consumption is always important for an IoT device regardless of the battery level, which would only change the degree of importance. As such, HBRA still tries to conserve some energy while focusing primarily on the monetary cost reduction, and thus this puts a barrier for HBRA's loss in energy consumption. Nonetheless, LBRA does not take the monetary cost reduction into account at all, as a

[8]Energy consumption minimization and/or monetary cost reduction are the secondary objectives for both LBRA and HBRA, since they fully prioritize the device requirements.

result, the scale of HBRA's gain in monetary cost is more than HBRA's loss in energy consumption.

## VI. Conclusion

A novel context-aware approach is presented in this work for IoT networks, and connectivity-processor pair is jointly optimized in order to meet the objectives in terms of the energy consumption, monetary cost, security, and response time. More specifically, this work is an attempt to determine the wireless connection type and data processing unit along with the amount of data to be offloaded, which is the case where data can be processed at a unit other than the device. In that regard, IoT devices come with diverse requirements in terms of response time and security, and they are allowed to prioritize their requirements. In addition, the proposed scheme also takes the battery level of a device into account, such that the minimization of energy consumption becomes the focus if the battery level is under a certain threshold, while the reduction of monetary cost is mainly targeted in case the battery level is above that threshold. The proposed scheme employs $Q$-learning algorithm, and manages to achieve significant gains compared to deterministic benchmark routes. Results demonstrate that the proposed method outperforms all the benchmark methods in the novel joint metric, combining all the objectives of the formulated problem.

## References


[1] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[2] S. Popli, R. K. Jha, and S. Jain, "A survey on energy efficient narrowband internet of things (NBIoT): Architecture, application and challenges," *IEEE Access*, vol. 7, pp. 16 739–16 776, 2019.

[3] M. Ozturk, M. Jaber, and M. A. Imran, "Energy-aware smart connectivity for IoT networks: Enabling smart ports," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[4] N. Kouzayha, M. Jaber, and Z. Dawy, "Measurement-based signaling management strategies for cellular IoT," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1434–1444, Oct 2017.

[5] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of things in the 5G era: Enablers, architecture, and business models," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 510–527, March 2016.

[6] R. Castellano, U. Fiore, G. Musella, F. Perla, G. Punzo, M. Risitano, A. Sorrentino, and P. Zanetti, "Do digital and communication technologies improve smart ports? a fuzzy dea approach," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5674–5681, Oct 2019.

[7] W. Sun, J. Liu, and Y. Yue, "AI-enhanced offloading in edge computing: When machine learning meets industrial IoT," *IEEE Network*, vol. 33, no. 5, pp. 68–74, Sep. 2019.

[8] X. Ma, T. Yao, M. Hu, Y. Dong, W. Liu, F. Wang, and J. Liu, "A survey on deep learning empowered IoT applications," *IEEE Access*, vol. 7, pp. 181 721–181 732, 2019.

[9] W. Kassab and K. A. Darabkh, "A–z survey of internet of things: Architectures, protocols, applications, recent advances, future directions and recommendations," *Journal of Network and Computer Applications*, vol. 163, p. 102663, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520301375

[10] M. Elsaadany, A. Ali, and W. Hamouda, "Cellular LTE-A technologies for the future internet-of-things: Physical layer features and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2544–2572, Fourthquarter 2017.

[11] A. Čolaković and M. Hadžialić, "Internet of things (iot): A review of enabling technologies, challenges, and open research issues," *Computer Networks*, vol. 144, pp. 17–39, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128618305243

[12] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Computer Networks*, vol. 130, pp. 94–120, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128617303778

[13] S. Alam, S. T. Siddiqui, A. Ahmad, R. Ahmad, and M. Shuaib, "Internet of things (iot) enabling technologies, requirements, and security challenges," in *Advances in Data and Information Sciences*, M. L. Kolhe, S. Tiwari, M. C. Trivedi, and K. K. Mishra, Eds. Singapore: Springer Singapore, 2020, pp. 119–126.

[14] S. Krishnamoorthy, A. Dua, and S. Gupta, "Role of emerging technologies in future iot-driven healthcare 4.0 technologies: a survey, current challenges and future directions," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–47, 2021.

[15] K. Gasmi, S. Dilek, S. Tosun, and S. Ozdemir, "A survey on computation offloading and service placement in fog computing-based iot," *The Journal of Supercomputing*, pp. 1–32, 2021.

[16] I. D. I. Saeedi and A. K. M. Al-Qurabat, "A systematic review of data aggregation techniques in wireless sensor networks," *Journal of Physics: Conference Series*, vol. 1818, no. 1, p. 012194, mar 2021. [Online]. Available: https://doi.org/10.1088/1742-6596/1818/1/012194

[17] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[18] I. U. Din, M. Guizani, J. J. Rodrigues, S. Hassan, and V. V. Korotaev, "Machine learning in the internet of things: Designed techniques for smart cities," *Future Generation Computer Systems*, vol. 100, pp. 826–843, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X19304030

[19] S. Persia, C. Carciofi, M. Barbiroli, M. Teodori, V. Petrini, A. Garzia, and M. Faccioli, "IoT enabling technologies for extreme connectivity smart grid applications," in *2019 CTTE-FITCE: Smart Cities Information and Communication Technology (CTTE-FITCE)*, Sep. 2019, pp. 1–6.

[20] R. K. Singh, P. P. Puluckul, R. Berkvens, and M. Weyn, "Energy consumption analysis of lpwan technologies and lifetime estimation for iot application," *Sensors*, vol. 20, no. 17, p. 4794, 2020.

[21] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[22] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on internet of things security: Requirements, challenges, and solutions," *Internet of Things*, p. 100129, 2019.

[23] M. Dammak, O. R. M. Boudia, M. A. Messous, S. M. Senouci, and C. Gransart, "Token-based lightweight authentication to secure iot networks," in *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2019, pp. 1–4.

[24] S. Agrawal and P. Ahlawat, "A survey on the authentication techniques in internet of things," in *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, 2020, pp. 1–5.

[25] J. Abegunde, H. Xiao, and J. Spring, "A smart game for data transmission and energy consumption in the internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 528–543, Jan 2020.

[26] A. Biason, C. Pielli, M. Rossi, A. Zanella, D. Zordan, M. Kelly, and M. Zorzi, "EC-CENTRIC: An energy- and context-centric perspective on IoT systems and protocol design," *IEEE Access*, vol. 5, pp. 6894–6908, 2017.

[27] G. Hattab and D. Cabric, "Energy-efficient massive cellular IoT shared spectrum access via mobile data aggregators," in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2017, pp. 1–6.

[28] E. Fitzgerald, M. Pióro, and A. Tomaszwski, "Energy-optimal data aggregation and dissemination for the internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 955–969, April 2018.

[29] N. Gupta, S. K. Dhurandher *et al.*, "Efficient caching method in fog computing for internet of everything," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 439–452, 2021.

[30] K. Hasan and S.-H. Jeong, "Efficient caching for data-driven iot applications and fast content delivery with low latency in icn," *Applied Sciences*, vol. 9, no. 22, p. 4730, 2019.

[31] R. Tapwal, N. Gupta, and Q. Xin, "Data caching at fog nodes under IoT networks: Review of machine learning approaches," apr 2020. [Online]. Available: https://doi.org/10.36227%2Ftechrxiv.12091410.v1

[32] S. Tayade, P. Rost, A. Maeder, and H. D. Schotten, "Device-centric energy optimization for edge cloud offloading," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[33] S. Tayade, P. Rost, A. Maeder, and H. D. Schotten, "Delay constrained energy optimization for edge cloud offloading," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2018, pp. 1–6.

[34] J. Azar, A. Makhoul, M. Barhamgi, and R. Couturier, "An energy efficient iot data compression approach for edge machine learning," *Future Generation Computer Systems*, vol. 96, pp. 168–175, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18331716

[35] B. Cheng, M. Wang, S. Zhao, Z. Zhai, D. Zhu, and J. Chen, "Situation-aware dynamic service coordination in an IoT environment," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2082–2095, 2017.

[36] B. Cheng, M. Wang, X. Lin, and J. Chen, "Context-aware cognitive QoS management for networking video transmission," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1422–1434, 2021.

[37] T. J. Saleem and M. A. Chishti, "Deep learning for internet of things data analytics," *Procedia Computer Science*, vol. 163, pp. 381–390, 2019, 16th Learning and Technology Conference 2019Artificial Intelligence and Machine Learning: Embedding the Intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919321593

[38] C. Lee, J. Lin, P. Chen, and Y. Chang, "Deep learning-constructed joint transmission-recognition for internet of things," *IEEE Access*, vol. 7, pp. 76 547–76 561, 2019.

[39] S. H. Lee, T. Lee, S. Kim, and S. Park, "Energy consumption prediction system based on deep learning with edge computing," in *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, May 2019, pp. 473–477.

[40] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, Jan 2018.

[41] M. Guo, N. Pissinou, and S. S. Iyengar, "Privacy-preserving deep learning for enabling big edge data analytics in internet of things," in *2019 Tenth International Green and Sustainable Computing Conference (IGSC)*, Oct 2019, pp. 1–6.

[42] S. Rathore, Y. Pan, and J. H. Park, "Blockdeepnet: a blockchain-based secure deep learning for iot network," *Sustainability*, vol. 11, no. 14, p. 3974, 2019.

[43] N. Waheed, X. He, M. Ikram, M. Usman, S. S. Hashmi, and M. Usman, "Security and privacy in iot using machine learning and blockchain: Threats and countermeasures," *ACM Comput. Surv.*, vol. 53, no. 6, Dec. 2020. [Online]. Available: https://doi.org/10.1145/3417987

[44] Y. Du, Z. Wang, and V. Leung, "Blockchain-enabled edge intelligence for iot: Background, emerging trends and open issues," *Future Internet*, vol. 13, no. 2, p. 48, 2021.

[45] H. Khadilkar, "A scalable reinforcement learning algorithm for scheduling railway lines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 727–736, 2019.

[46] A. Rodríguez-Molina, E. Mezura-Montes, M. G. Villarreal-Cervantes, and M. Aldape-Pérez, "Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem," *Applied Soft Computing*, vol. 93, p. 106342, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620302829

[47] Y. Chen, L. K. Norford, H. W. Samuelson, and A. Malkawi, "Optimal control of hvac and window systems for natural ventilation through reinforcement learning," *Energy and Buildings*, vol. 169, pp. 195–205, 2018.

[48] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," *Computers & Operations Research*, vol. 134, p. 105400, 2021.

[49] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[50] E. M. Russek, I. Momennejad, M. M. Botvinick, S. J. Gershman, and N. D. Daw, "Predictive representations can link model-based reinforcement learning to model-free mechanisms," *PLOS Computational Biology*, vol. 13, no. 9, pp. 1–35, Sep. 2017.

[51] E. Even-Dar and Y. Mansour, "Convergence of optimistic and incremental Q-learning," in *Advances in neural information processing systems*, 2002, pp. 1499–1506.

[52] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. MIT press, 2018.

[53] M. Z. Asghari, M. Ozturk, and J. Hämäläinen, "Reinforcement learning based mobility load balancing with the cell individual offset," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–5.