Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5379326, 11 pages
https://doi.org/10.1155/2018/5379326

Research Article

Energy-Aware Smart Connectivity for IoT Networks: Enabling Smart Ports

Metin Ozturk,[1] **Mona Jaber**,[2] **and Muhammad A. Imran**[1]

[1] *School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK*
[2] *Fujitsu Laboratories of Europe, Hayes UB4 8FE, UK*

Correspondence should be addressed to Muhammad A. Imran; muhammad.imran@glasgow.ac.uk

Received 7 March 2018; Revised 5 June 2018; Accepted 10 June 2018; Published 28 June 2018

Academic Editor: Manuel Fernandez-Veiga

The Internet of Things (IoT) is spreading much faster than the speed at which the supporting technology is maturing. Today, there are tens of wireless technologies competing for IoT and a myriad of IoT devices with disparate capabilities and constraints. Moreover, each of many verticals employing IoT networks dictates distinctive and differential network qualities. In this work, we present a context-aware framework that jointly optimises the connectivity and computational speed of the IoT network to deliver the qualities required by each vertical. Based on a smart port application, we identify energy efficiency, security, and response time as essential quality features and consider a wireless realisation of IoT connectivity using short range and long-range technologies. We propose a reinforcement learning technique and demonstrate significant reduction in energy consumption while meeting the quality requirements of all related applications.

1. Introduction

The Internet of Things (IoT) is today's buzzword, often coupled with *Big Data* and *Artificial Intelligence* (AI). However, there is a lot of ambiguity of what is meant by that and scepticism about the actual value generated by the IoT. IoT devices have become pervasive but cover a broad range of technologies and standards. Wireless technology is key to connect these devices through gateways or aggregation points; but, similarly, a wide range of wireless protocols and standards are available and competing [1]. Once these devices are connected, they start reporting the sensed or measured data to the platform. Again, multiple choices are possible in this aspect with different strengths and weaknesses. Reporting raw data to the cloud is very costly as every bit gets charged and may also exhaust the battery of the device; this results in *massive* data. On the other hand, running scripts locally in the device and reporting the resulting events to the cloud reduce the cloud service cost but limits the visibility to the actual data; this still results in *big* data. Moreover, local scripts result is real-time actions and do not expose the privacy of the data, whereas cloud computing incurs latency due to the transmission network and requires stringent security measures to protect the data.

An environment, which is rich in IoT devices that are connected to a platform, qualifies as *digitised*, and often as *intelligent*. Analytics, which uses AI, is the added layer that transforms such an environment into a *smart* one. The default application of AI is to draw actionable insights from the data in order to generate value to the given vertical. In this work, we argue that IoT solutions should not be addressed through a layered perspective but, instead, a holistic optimisation approach is needed to generate the desired added value efficiently. In such a holistic approach, AI, among other machine learning tools, is employed in every stage of the solution including connectivity, storage, computing, and analytics.

Since there are many use-cases of the IoT paradigm [2], it should be approached from a given vertical perspective, e.g., smart health, smart cities, smart manufacturing (Industry 4.0), smart transport, etc. Each of these verticals comprises multiple IoT-based applications with various requirements. In [3], for example, signalling measurements and modelling are performed for both static and vehicular machine-to-machine (M2M) applications, as both have different signalling overhead characteristics. As another example, remote monitoring in smart cities requires full compliance with privacy regulations, whereas security-related applications rank response time highest among all key performance indicators (KPI).

In this article, we adopt the smart port use-case to demonstrate the context-aware smart connectivity, since it includes various types of applications and has a determined need for monetisation (as opposed to smart cities that are primary developed for the well-being and productivity of the society). According to figures from the World Trade Organization, 80% of worldwide freight is transported through ports (https://www.wto.org/). The smart port concept entails the use of technologies to transform the different public services at ports into interactive systems with the purpose of meeting the needs of port users with a greater level of efficiency, transparency, and value. European smart port initiatives include the following among many others:

(i) *The port of Rotterdam* where IoT-sensors are used to generate a digital twin and enable augmented intelligence.

(ii) *The port of Hamburg* which exploits 5G networks to enable virtual reality for vital infrastructure monitoring.

(iii) *The port of Antwerp* employs blockchain technology to enable a secure transfer of rights to be exchanged between often competing parties.

(iv) *The port of Seville* through the Tecnoport 2025 project uses mobile network technology for traffic and goods tracking on port and their logistical transfer on land.

Smart ports present a particular challenge due to the necessity of information exchange among competing stakeholders including port authorities, port operators, terminal operators, logistics companies, shipping companies, etc. It is then likely that multiple IoT networks would coexist and would consist of partly private and partly public or shared infrastructure. As described in [4], there are various communication standards, with different strengths and weaknesses, which may be used for connecting IoT networks in the context of smart ports. Mobile IoT, i.e., connectivity over licensed mobile wireless networks, is often the preferred solution for handling private data, since it is reliable, end-to-end secure (owing to the eSIM card), scalable, ubiquitous, and mature. Two main technologies have been introduced by mobile networks to connect IoT devices: eMTC and NB-IoT [5]. Both of these technologies are compatible with LTE (state-of-the-art commercial mobile network technology) which means that a software update suffices to deploy the IoT options. The former is geared towards higher rates (> 1 Mbps) and supports VoIP (Voice over IP based on ITU H.323 protocol (https://www.itu.int/rec/T-REC-H.323/e)) and flexile mobility. The latter is designed for low data rates (20 kbps) and long range (100 km) but with limited mobility. The NB-IoT technology consists of restricting the energy of an LTE normal carrier in a narrow band, hence allowing a maximum coupling loss that is 20 dB higher (164 dB) than LTE [6]. Mobile IoT is a public service enabled by telecom carriers and may be used by any party who subscribes to it. Other long-range and low-power solutions, such as LoRa(https://www.lora-alliance.org/) and Sigfox(https://www.sigfox.com/en), are unlicensed and can reach similar coverage and data rates as NB-IoT and eMTC. These may be privately owned but require the usage of a gateway to connect to the Internet and are often considered less secure. Many short range unlicensed wireless connectivity solutions are available, such as WiFi (IEEE 802.11g), Bluetooth, ZigBee, etc., as described in [7], and may be shared, public, or private.

In the presence of multiple wireless technologies, disparate IoT applications, competing parties, and a broad range of static and moving IoT devices with multiple connectivity options, it is of key importance to identify the best way to collect, store, cache, and process the IoT data. What qualifies as *the best way* depends on the device capabilities (e.g., connectivity options, available battery); the wireless conditions; the security requirements; the processing complexity and availability; the cost of storage/caching/uploading, etc.

2. Related Work

As the energy consumption is one of the challenges for IoT networks [8], recent works, such as [9, 10], study the trade-off between local and cloud computing in terms of device energy consumption. The former proposes an analytical framework that minimises the energy consumption by optimising the offloading decision of multiple user devices. The latter elaborates a theoretical framework for establishing trade-offs in the energy consumption and IoT infrastructure billing comprising cloud computing. Mobile wireless networks are a prime contender in the race to connect IoT networks owing to their well-established and ubiquitous coverage and secure communication based on the subscriber identity module (eSIM card). In [11], authors investigate the connectivity of NB-IoT and LoRa in terms of both area and population coverage in order to highlight the importance of the network deployments. In [12], big data analytics based user-centric smart connectivity is argued by providing corresponding research challenges.

Although data aggregation seems a promising solution to ease the signalling overhead, it is one of the causes of the transmission delay. In [13], authors discuss the trade-off between delay and signalling overhead in order to demonstrate the impacts of data aggregation. Authors in [14] analyse the joint optimisation of caching and task offloading in such networks with mobile edge computing. They present an efficient online algorithm based on Lyapunov optimisation and Gibbs sampling that succeeds in reducing computation latency while keeping the energy consumption low. In [15], a recommendation system is proposed to address the challenge of link selection in a cloud radio access network. A data-driven scheme is introduced that results in optimised classification of link strengths between remote radio heads and IoT devices.

A deep learning algorithm for edge computing is introduced in [16] to boost the learning performance in IoT networks. They also attempt to increase the amount of edge tasks by considering the edge capacity constraints. An open-source database is designed in [17] for the edge computation of Industrial IoT (IIoT) networks. The authors use a time-series analysis for predicting conditions of IIoT machines in order to decrease the amount of condition reports to be sent to the cloud. A holistic view of communication, computation, and caching is presented in [18] using graph-based representations as learning methods for innovative resource allocation techniques. The performance of the edge-caching as well as the energy efficiency and delivery time is investigated in [19] with quality of service (QoS) constraints.

In this work, we employ machine learning techniques, based on reinforcement learning, in order to manage multiple optimisation objectives jointly and to dynamically identify *the best connection* and route for each device. We identify four key quality features that dominate IoT applications in general and smart ports in particular: *security, energy, latency*, and *cost*. This work is the first to address these multiple IoT optimisation objectives jointly using reinforcement learning. We compare our novel approach to the state-of-the-art connectivity solutions and demonstrate significant gains in all aspects (ranging from 95.9% to 283.54%). Moreover, our approach is the only one that is able to meet the context-aware requirements fully, while minimising the cost and the energy consumption. The advantage of the machine learning scheme adopted is primarily its low complexity and its ability to optimise in a dynamic environment such as a smart port.

The rest of the paper is organised as follows. In Section 3 we define the system model of our research. In Section 4, we present our novel machine-learning-based solution for solving the multiobjective problem. Section 5 elaborates the results and analysis, and in Section 6 we conclude the article.

3. System Model

The energy-aware smart connectivity novel approach proposed in this work applies to any IoT network with diverse options of connectivity and processing. For the sake of clarity in the presentation, we build the system model around a smart port scenario such as the one shown in Figure 1. All IoT devices are battery operated and have different battery lives. They all have some processing power to perform basic tasks and can either offload the task to the gateway (or fog), i.e., the WiFi access point or to the evolved node B (eNB or cloud).

Differently from the state-of-the-art research, we propose to decide simultaneously on the best connectivity and the best location for processing the tasks by jointly optimising energy, response time, security, and cost. A two-stage approach, which describes the decision and optimisation processes, is presented in Figure 2. It is assumed that every IoT device is controlled by a given application and they jointly determine the context-aware constraints. Each combination of connectivity option and processing location offers specific characteristics and limitations. Stage 1 consists of optimising these decisions based on the context-aware constraints, while Stage 2 refines the trade-off between energy consumption and cost. In the following paragraphs, we describe the models adopted to capture the propagation loss, energy consumption, and response time for the proposed system. Table 1 lists all the parameters that are pertinent to our simulations.

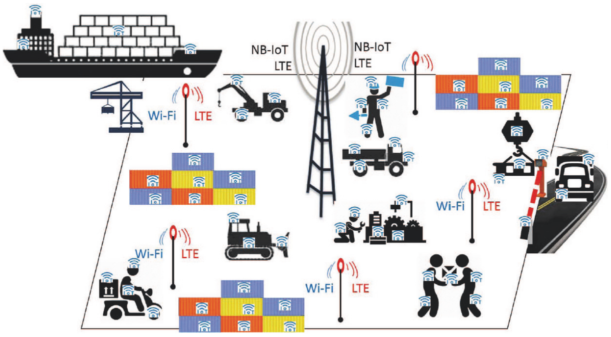


Figure 1: Smart port diagram with two overlapping networks: NB-IoT and WiFi. WiFi access points use LTE for backhauling. All IoT devices are capable of both wireless technologies.

3.1. Propagation Model. There are three wireless connections that require modelling: (a) Device-to-Gateway (WiFi), (b) Device-to-eNB (NB-IoT), and (c) Gateway-to-eNB (LTE). Connections (a) and (c) are often interference limited, as the employed spectrum is likely to be shared by other neighbouring connections. Connections of type (b) are, however, considered to be noise limited, as we assume that there are no other eNB in the surrounding employing NB-IoT technology. The objective of the propagation modelling is to determine the transmission power required to cater for each of the wireless connection types. Accordingly, the energy consumption will be calculated. We start with the propagation loss L which is modelled as a function of two technology-specific parameters, the propagation constant K and the propagation exponent α, and the distance of the wireless hop δ measured in km, as shown below:

$$L = K \cdot \delta^{\alpha}. \tag{1}$$

Moreover, the probability of having line of sight between the device and the gateway is much higher than in the case of the other types of wireless connections; hence the propagation loss per decade is less [20]. On the other hand, NB-IoT connections suffer the same propagation loss per decade as LTE links, however, are successfully received with 20 dB less power (threshold receiver sensitivity is −141 dBm). For all types of links, the received power at a distance d_x from the transmitting device can be expressed as $P_r = P_t/L$ in mWatt. Next, we calculate the required received power P_r (in mWatt) in order to achieve the target data transmission D in bits:

$$D = T \cdot B \cdot \log_2 \left(1 + \frac{P_r}{P_I + N_0 \cdot B} \right), \tag{2}$$

where T is the time period, B is the channel bandwidth, and P_I is the cumulative interference power on the given channel during time period T. Please note that P_I is null for wireless connections of type (b). Using (2) and solving for P_r, we get

$$P_r = \left(2^{D/(T \cdot B)} - 1 \right) \times \left(P_I + N_0 \cdot B \right). \tag{3}$$

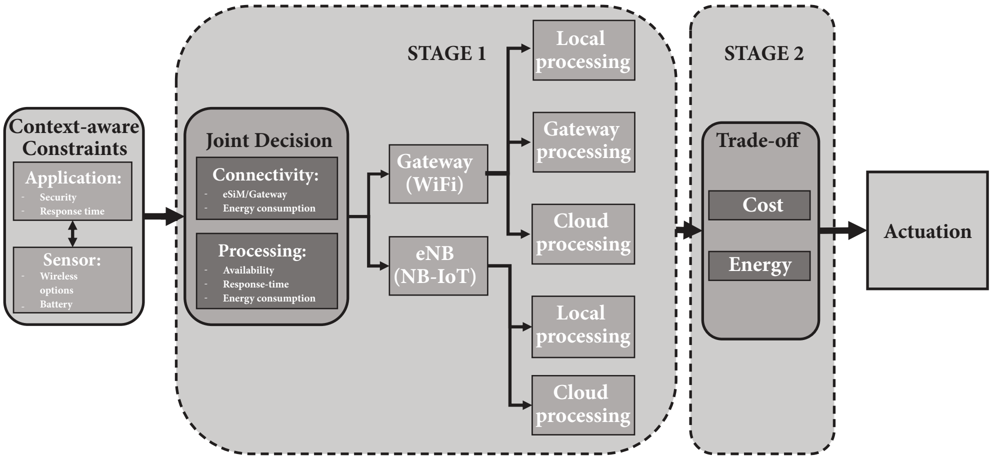


Figure 2: Decision and optimisation processes in a two-stage approach to optimise four performance criteria: energy, response time, security, and cost.

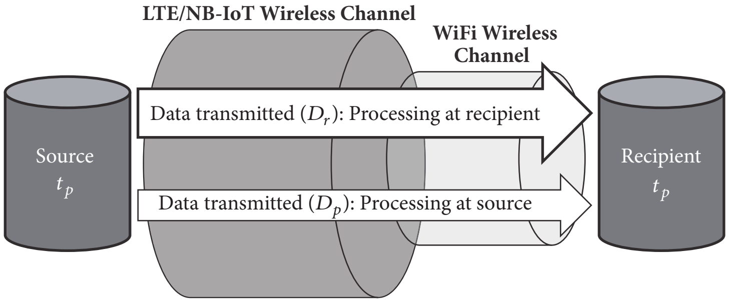

LTE/NB-IoT Wireless Channel
WiFi Wireless Channel
Source
t_p
Data transmitted (D_r): Processing at recipient
Data transmitted (D_p): Processing at source
Recipient
t_p

Figure 3: Uplink delay model capturing the factors affecting both processing and transmission delays over any hop in our system.

3.2. Energy Consumption Model. There are two major processes that consume energy in an IoT network: wireless transmission and task computation. The energy consumption of the former is E_t and the latter is E_p; thus the total energy consumption is the sum of both. Depending on the route of communication taken by the device, the energy consumed due to transmission power can be a result of either one hop using NB-IoT (E_{t_b}) or two hops using WiFi for the first link and LTE for the second ($E_{t_a} + E_{t_c}$). The energy consumed for processing the task is a function of the data rate requirement of device d, θ_d, and the computational power of the processor, $E_{p_i}\ \forall i = \{d, f, c\}$ (see Table 1), and is expressed as $E_p = \theta \cdot E_{p_i}$.

3.3. Response Time Model. The response time perceived by the IoT device is the combination of the uplink and downlink delays between the IoT device and the server. In this work, the uplink delay is modelled, while the downlink delay is assumed the same for all devices.

The uplink delay is caused by two phenomena: task processing (processing delay, t_p) and data transmission (transmission delay, t_t). The processing delay depends on the processor's computational power, which is measured in the number of computational cycle per data element (η); i.e., the higher η, the less computational power. Naturally, a server has higher computational power than a small gateway and much higher than a simple IoT device ($\eta_c < \eta_f < \eta_d$). Thus, in this work, t_p is modelled based on the computational powers of the processing locations: $t_{p_d} = 10 \times t_{p_f} = 100 \times t_{p_c}$. In addition, while the input to the task processing stage is large raw data, the output is compressed data with comparably less volume. To that end, the compression rate between the input and output data volumes is given as C; $D_r = C \cdot D_p$, where D_r and D_p are the volumes of raw and processed (compressed) data, respectively.

The transmission delay is affected by the type of radio access technology and the volume of data to be transmitted. Since WiFi access employs the unlicensed frequency bands, it often suffers from higher retransmission rates, which results in increased transmission delays, due to frequent collisions. Therefore, in this work, this effect is captured by the factor $F > 1$ whereby the delay incurred for transmitting the same volume of data over WiFi is F times higher than that over LTE or NB-IoT; $t_{t,a} = F \cdot t_{t,b} = F \cdot t_{t,c}$. This model is represented in Figure 3, in which the source could be either the IoT device or the gateway, and the recipient could be either the gateway or the cloud.

Consequently, the overall response time for each action is calculated for $C = 200$ and $F = 2$ as follows:

$$R = t_p + \sum_{i=1}^{N_h} t_{t_i} \cdot D_i, \qquad (4)$$

Table 1: System model parameters and simulation values.

Parameter	Value	Description
r_n	200 m	eNB cell radius
r_w	30 m	WiFi cell radius
N_G	10	Number of IoT devices per gateway
χ_d	30 Kbps	Computational capacity (device)
χ_f	10^2 Kbps	Computational capacity (fog)
χ_c	10^3 Kbps	Computational capacity (cloud)
ϵ	5×10^{-9} Joule	Energy consumption per computational cycle
η_d	10^2	Required amount of computational cycle per data element (device)
η_f	10	Required amount of computational cycle per data element (fog)
η_c	1	Required amount of computational cycle per data element (cloud)
N_0	-204 dBW/Hz	Noise density
B	180 kHz	Bandwidth
$\overline{P_{t,d}}$	10^{-8} W	Average transmit power of the IoT devices in the gateways #2, #3, #4, and #5
T	1 s	Time period
λ	0.5	Q-table update parameter
ϕ	0.9	Q-table update parameter
ε_1	0.8	Action selection parameter for Stage 1
ε_2	10^4	Action selection parameter for Stage 2
ρ	0.8	Decaying rate for ε_1 and ε_2
S	8	Number of bits in each data element
ϑ	$10^3/S$	Conversion of kbps data rates to number of data elements
E_{p_d}	$\epsilon \cdot \eta_d \cdot \lambda$	Data processing energy consumption per data rate in kbps (device)
E_{p_f}	$\epsilon \cdot \eta_f \cdot \lambda$	Data processing energy consumption per data rate in kbps (fog)
E_{p_c}	$\epsilon \cdot \eta_c \cdot \lambda$	Data processing energy consumption per data rate in kbps (cloud)
Γ_d	10^{-4}	Cost of processing per kbps (device)
Γ_f	10^{-1}	Cost of processing per kbps (fog)
Γ_c	1	Cost of processing per kbps (cloud)
b	20	Budget
β_1	10^2	Constant coefficient for penalty comparison.
β_2	10^{12}	Constant coefficient for penalty comparison.
$K_w = K_l = K_n$	128.1 dB	Propagation loss constant for all wireless connection types (a, b, and c).
$\alpha_l = \alpha_n$	3.76	Propagation loss exponent for NB-IoT and LTE wireless connection types (b and c).
α_w	3	Propagation loss exponent for Wi-Fi (802.11g) wireless connection type (a).

where $N_h = \{1, 2\}$ is the number of hops and $D = \{D_r, D_p\}$. Besides, t_{t_i} and D_i represent the values of t_t and D for the i^{th} hop, respectively. Then, the calculated values populate Table 2 after the application of feature scaling into the range of [0, 1] using the function given as

$$f(x) = \frac{x - \min(X)}{\max(X) - \min(X)}, \qquad (5)$$

where X is the set of x. Note that both (a) and (b) type connections constitute the first hop, while the connection type (c) is the second hop.

4. Machine Learning-Based Solution

In this work, we propose to employ reinforcement learning (RL), a machine learning technique based on a goal-seeking approach. It is a trial and error approach in which the agent (or learning device) learns to take the correct action by interacting with its surroundings and being rewarded or penalised in each iteration. RL is selected in this work due to its great applicability to the presented problem. For example, IoT devices need to interact with its environment in order to assess the circumstances and to take subsequent actions, which is determination of the connection type and the data processing location. Therefore, RL maps to this requirement very well, since it allows optimisation with environmental interactions.

Being one of the most prominent reinforcement learning techniques, Q-learning aims to find the optimum policy for a given problem, that is, the best action to take at any given state. To do this, the agent takes an action and evaluates the subsequent reward/cost of taking that action given that it was

Table 2: Stage one action list.

Action	Connection	Processor	Tuple
A_1	Wi-Fi	Device	$A_1 = [0.004, 1, \chi_d, (E_{t_a} + E_{t_c} + E_{p_d} \cdot \theta), \Gamma_d]$
A_2	Wi-Fi	Fog	$A_2 = [0.62, 1, \chi_f, (E_{t_a} + E_{t_c} + E_{p_f} \cdot \theta), \Gamma_f]$
A_3	Wi-Fi	Cloud	$A_3 = [1, 1, \chi_c, (E_{t_a} + E_{t_c} + E_{p_c} \cdot \theta), \Gamma_c]$
A_4	NB-IoT	Device	$A_4 = [0, 0, \chi_d, (E_{t_b} + E_{p_d} \cdot \theta), \Gamma_d]$
A_5	NB-IoT	Cloud	$A_5 = [0.2, 0, \chi_c, (E_{t_b} + E_{p_c} \cdot \theta), \Gamma_c]$

in a certain state. This reward/cost is then used to update a look-up-table known as the Q-table, which is later utilised by the agent to select the best action. Further, the agent calculates the Q-value for every possible state/action pair. Therefore, a simple implementation can result in the agent learning online the best actions, regardless of the policy.

Moreover, Q-learning offers two key features which enable an efficient solution to our problem. First, as it is a model-free learning approach [21, 22], it is (1) capable of operating in dynamically changing environments, (2) a low-complexity algorithm which does not require a lot of power, thus reducing the energy consumption of the IoT network. Second, Q-learning is known to converge in most cases [23], which has also been demonstrated in multiagent noncooperative environments [24], as are IoT networks.

We propose a two-stage approach to solve the energy-aware smart IoT connectivity where each of the stages employs Q-learning.

4.1. First Stage Learning. Stage 1 consists of learning the best combination of connectivity and processing location in view of the device and application requirements and the limitations offered by each of these options. Thus, there are five possible actions that may be taken by each device as described in Table 2. As a side note, all the variables in Table 2 are the feature scaled values (into the range of [0, 1]) calculated through (5). The tuples shown represent the limitations of each action, e.g., $A_i = [R, \Sigma, \chi_l, E_t + E_p, \Gamma_l]$, where R and $E_t + E_p$ are described in Sections 3.3 and 3.2, respectively, χ_l is the available processing capacity, and Γ_l is the processing cost where $l = \{d, f, c\}$ as defined in Table 1. The parameter $\Sigma = \{1, 2\}$ refers to the level of data security offered by the wireless technology, whereby, the value 1 indicates eSIM protection (only provided by NB-IoT) and 2 the absence of that. Moreover, each device may be in four different states, as shown in Table 3, depending on the context-aware constraints defined jointly by the device and application. These constraints are R', Σ', and χ' which represent the response time, security level, and computational power requirements, respectively.

4.1.1. Penalty Function Determination. Each device will estimate the penalty function associated with each possible action it is able to take, following the system shown in Table 3, where $\varphi_p = \{R - R', \Sigma - \Sigma', \chi - \chi' \mid p = 1, 2, 3\}$ is the difference between the available and required characteristics. The fourth penalty is $\varphi_4 = \chi' \cdot A_i^{(5)} - b$, where $A_i^{(5)}$ is the fifth index of i^{th} action and the parameter b is the available budget.

The penalty function determination policy aims to satisfy the optimisation objective by including the elements that are desired to be minimised. As seen from Table 3, the penalty functions consist of three main elements: constant term, dissatisfaction level, and energy consumption. The constant value is the cost of being in the states and it decreases while the level of state increases. This element compels the agent try to achieve the highest possible level of states, as it is one of the objectives of the optimisation problem. The element of dissatisfaction level, as a supportive of the constant value, incurs cost for not satisfying the device requirements in order to improve the satisfaction levels. Lastly, the energy consumption element provides minimisation in the end-to-end energy consumption (connection and data processing). The parameter $0 \leq \nu \leq 1$ is the battery level, where 0 represents an empty battery and 1 represents the full charge. In the expressions in Table 3, the parameter ς specifies the priority level of the energy consumption. For instance, low values of ς prioritise the energy consumption once the battery level, ν, is very low (e.g., 5%), while high values prioritise the energy consumption even when the battery level is high (e.g., 50%).

In addition to all these, normally, the algorithm tends to select an option with a cloud processing, as it is the most energy efficient one. However, some amount of data will not be offloaded due to budget constraints, and will then be processed locally, which is the most energy consuming option. Note that this amount is evaluated by the second stage learning. Thus, the selected option by the first stage would be more energy consuming than the fog processing-included option, as the processing will be the combination of the cloud and device. Therefore, the last parts of the penalty functions (inside the square brackets) prevent the algorithm from making blind decisions, which ignores the budget availability, by including an average energy consumption of the actions with the device processing. The reason of taking the average value is that the final action is yet to be taken during the learning process. The coefficients of these three elements are determined empirically. However, they can be used to prioritise any element that is desired to be minimised more.

The Q-table entries are then updated according to the following expression, where s, s', P, and a are the current state, next state, penalty function, and action under evaluation:

$$Q(s,a) \longleftarrow Q(s,a) + \lambda \left(P(s) + \phi \min \left(Q\left(s', a\right) \right) - Q(s,a) \right). \tag{6}$$

Table 3: List of possible states of each device in Stage one and corresponding penalty calculation.

State	Description	Penalty function (P)
σ_1	None of the constraints are satisfied	$10^4 + \sum_{p=\{1-3\}} \varphi_p + 10^{\varsigma/\nu} \cdot \beta_1 \cdot A_i^{(4)} \cdot \left[\left(1 - \frac{\varphi_4}{\chi' \cdot A_i^{(5)}}\right) + \frac{\varphi_4 \cdot \left(\left(A_1^{(4)} + A_4^{(4)}\right)/2\right)}{\chi' \cdot A_i^{(5)}}\right]$
σ_2	One constraint is satisfied	$5 \times 10^3 + 0.8 \sum_{p=\{1-3\}} \varphi_p + 10^{\varsigma/\nu} \cdot \beta_1 \cdot A_i^{(4)} \cdot \left[\left(1 - \frac{\varphi_4}{\chi' \cdot A_i^{(5)}}\right) + \frac{\varphi_4 \cdot \left(\left(A_1^{(4)} + A_4^{(4)}\right)/2\right)}{\chi' \cdot A_i^{(5)}}\right]$
σ_3	Two constraints are satisfied	$2 \times 10^3 + 0.6 \sum_{p=\{1-3\}} \varphi_p + 10^{\varsigma/\nu} \cdot \beta_1 \cdot A_i^{(4)} \cdot \left[\left(1 - \frac{\varphi_4}{\chi' \cdot A_i^{(5)}}\right) + \frac{\varphi_4 \cdot \left(\left(A_1^{(4)} + A_4^{(4)}\right)/2\right)}{\chi' \cdot A_i^{(5)}}\right]$
σ_4	Three constraints are satisfied	$0.8 \sum_{p=\{1-2\}} \varphi_p + \varphi_3 + 10^{\varsigma/\nu} \cdot \beta_1 \cdot A_i^{(4)} \cdot \left[\left(1 - \frac{\varphi_4}{\chi' \cdot A_i^{(5)}}\right) + \frac{\varphi_4 \cdot \left(\left(A_1^{(4)} + A_4^{(4)}\right)/2\right)}{\chi' \cdot A_i^{(5)}}\right]$

Table 4: List of possible states of each device in Stage two and corresponding penalty calculation.

State	Description	Penalty function
$\ddot{\sigma}_1$	No availability in cloud or fog for χ'	$10^3 + \beta_2 \left(A_i^{(4)} \cdot \left(1 - \ddot{A}_i\right) + \chi' \cdot \ddot{A}_i \cdot A_i^{(5)}\right)$
$\ddot{\sigma}_2$	Enough availability in cloud or fog but no budget for χ'	$10^3 + \beta_2 \left(A_i^{(4)} \cdot \left(1 - \ddot{A}_i\right) + \chi' \cdot \ddot{A}_i \cdot A_i^{(5)}\right)$
$\ddot{\sigma}_3$	Enough availability and budget for χ'	$\beta_2 \left(A_i^{(4)} \cdot \left(1 - \ddot{A}_i\right) + \chi' \cdot \ddot{A}_i \cdot A_i^{(5)}\right)$

4.2. Second Stage Learning. The second stage aims to find the best policy for task offloading by considering the budget and availability of the fog or cloud. To this end, the second stage is activated only when the action taken in Stage 1 does not result in local processing (i.e., A_1 and A_4). In Stage 2, Q-learning is also employed with 21 possible actions = $[0 : 0.05 : 1]$, and the constraints are the available budget b and the availability of the fog and/or cloud. The resulting states and penalty functions for this stage are listed in Table 4.

4.2.1. Penalty Function Determination. The penalty function of this stage is determined with a similar procedure to the first stage; hence, there are three cost elements: constant term, energy consumption, and monetary cost. Similar to the first stage, the constant value ensures ending up with the highest possible level of state. Having the energy consumption and monetary cost elements simultaneously provides finding the best trade-off between the two. However, unlike the first stage, these elements are calculated for a piece of data that is planned to be transferred, as specifying the best amount is the objective of this stage learning. Similarly, the coefficients are obtained empirically.

The interaction between Stage 1 and Stage 2 in the learning process is depicted in Algorithms 1 and 2, respectively.

5. Results and Analysis

In this section, we implement the proposed reinforcement learning approach in a simulation environment, as shown in Figure 4, using the parameter values defined in Table 1. We consider that half of the IoT devices connect with NB-IoT in view of the data privacy and related security requirements; these represent Group A. The remaining devices connect to the eNB through the WiFi gateway, hence over two wireless hops, and represent Group B. Consequently, there are six possible fixed scenarios that may be formed by selecting the processing location of each group of devices; these are listed in Table 6. A total of 100 iterations is conducted and, in each, random battery levels are allocated to each of the devices.

We compare the results obtained with our method to the six listed scenarios in terms of five different parameters: *energy*, *cost*, *dissatisfaction*, *number of out of budget devices*, and *joint penalty*. First, *energy* represents the end-to-end energy consumption caused from both connection and data processing. Second, *cost* is the overall monetary cost incurred by the use of the data processing locations, such as fog and cloud. Third, *dissatisfaction* is a measure of the total number of device requirements that are not satisfied. Fourth, *number of out of budget devices* reflects the count of devices that exceed their available monetary budgets during performing their tasks. Finally, *the joint penalty* indicates the cumulative combination of previous four parameters (*energy, cost, dissatisfaction,* and *number of out of budget devices*).

The results in terms of gain (positive values) and loss (negative values) are shown in Figure 5. Note that the values for parameters *energy*, *cost*, *dissatisfaction*, and *joint penalty* are obtained as follows:

$$g(x) = \frac{p_s - p_q}{p_q} \times 100, \qquad (7)$$

where p_s and p_q are the values from Table 5 for Scenarios A-F and Q-learning, respectively.

On the other hand, the gain/loss values for the parameter of *number of out of budget devices* in Figure 5 is calculated using the function given as

$$o(x) = \frac{\#Out\ of\ Budget\ Devices}{N_G} \times 100. \qquad (8)$$

It is worth noting that the results provided in Figure 5 are evaluated using the average values given in Table 5 along with 95% confidence intervals. Moreover, the joint cost parameter in Table 5 is calculated by summing them. However, before the summation, other four parameters (energy consumption,

```
Data: Context-aware constraints, available computational capacity in gateway and eNB, budget
Result: Combination of connectivity route and processing venue
initialization;
for all IoT devices do
    Determine the current state using Table 3;
    Evaluate all the actions;
    Calculate the penalty using Table 3;
    Select the best action;
    Jump to the next state;
    Update the Q-table;
    if the selected action includes fog(gateway) or
      cloud (eNB) processing then
        go to Algorithm 2
    end
end
```

Algorithm 1: First stage learning.

```
Data: Action selected by the first stage, available computational capacity in gateway and eNB, budget
Result: Share of data to be offloaded
initialization;
for all IoT devices do
    Determine the current state using Table 4;
    Evaluate all the actions;
    Calculate the penalty using Table 4;
    Select the best action;
    Jump to the next state;
    Update the Q-table;
end
```

Algorithm 2: Second stage learning.

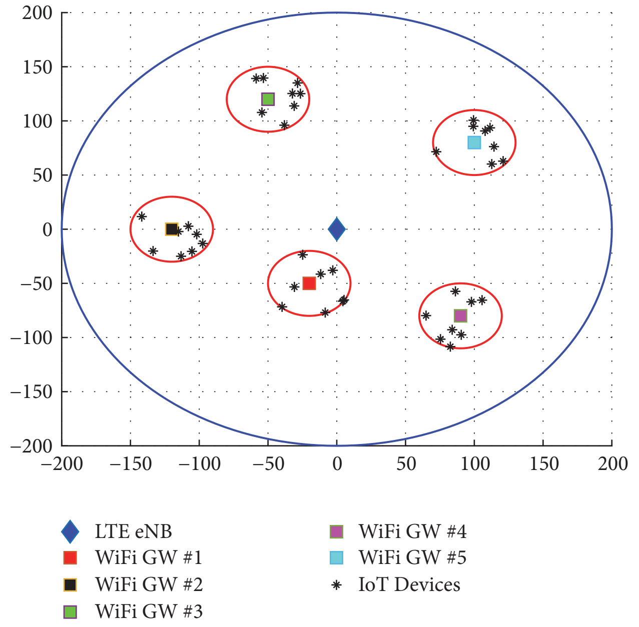


Figure 4: Sample snapshot of the simulation environment. IoT devices are located randomly, while positions of the gateways are fixed.

cost, dissatisfaction, and number of out of budget devices) are feature scaled into the range of [0, 1] using the function in (5) in order to keep their impacts in the same scale.

Our method outperforms any fixed combination when examining the joint or holistic gain, with values ranging from 95.9% to 283.54%. Similarly, the reinforcement learning technique results in better matching between the context-aware constraint and the availability of the IoT network compare to any other scenario, with gains varying from 183.33% to 344.44%. Although the processing cost of our proposed method is higher than that of Scenario A, the resulting gain in energy saving is even more important as well as the context-aware constraint compliance. The closest contender to reinforcement learning, with respect to the generated results, is Scenario C, in which the processing of Group A IoT devices is locally conducted while that of Group B occurs in the gateway. Nonetheless, the reinforcement learning allows for a device-driven context-aware connectivity that improves the compliance criteria by more than two times while saving 43.22% of energy, resulting in a holistic gain of 58.52%. Scenario D manages to reduce the energy consumption more than our proposed approach at the same total cost; however, 30.3% of the devices are out of budget resulting in incomplete or interrupted computational tasks.

Table 5: Results on various metrics for *Q*-learning and the scenarios.

	Energy Consumption (mJ)	Cost	Dissatisfaction	#Out of Budget Devices	Joint Cost
Q-Learning	5.69 ± 0.322	96.77 ± 4.01	1.8 ± 0.291	0 ± 0	0.7822
Scenario A	14.88 ± 0.385	$0.24 \pm 6.15e^{-3}$	5.1 ± 0.28	0 ± 0	1.5323
Scenario B	7.55 ± 0.24	118.57 ± 4.49	5.29 ± 0.181	3.03 ± 0.217	2.0679
Scenario C	8.16 ± 0.284	12.07 ± 0.383	5.81 ± 0.208	0 ± 0	1.2399
Scenario D	0.83 ± 0.025	130.41 ± 4.54	6 ± 0	3.03 ± 0.217	1.7756
Scenario E	7.48 ± 0.281	119.68 ± 3.83	7.81 ± 0.208	2.97 ± 0.213	2.4643
Scenario F	$0.15 \pm 4.59e^{-3}$	238.02 ± 6.16	8 ± 0	6 ± 0.339	3.0000

Table 6: List of fixed scenarios with connection types and locations of data processing.

Scenario	Group A	Group B
A	Device	Device
B	Cloud	Device
C	Device	Fog
D	Cloud	Fog
E	Device	Cloud
F	Cloud	Cloud

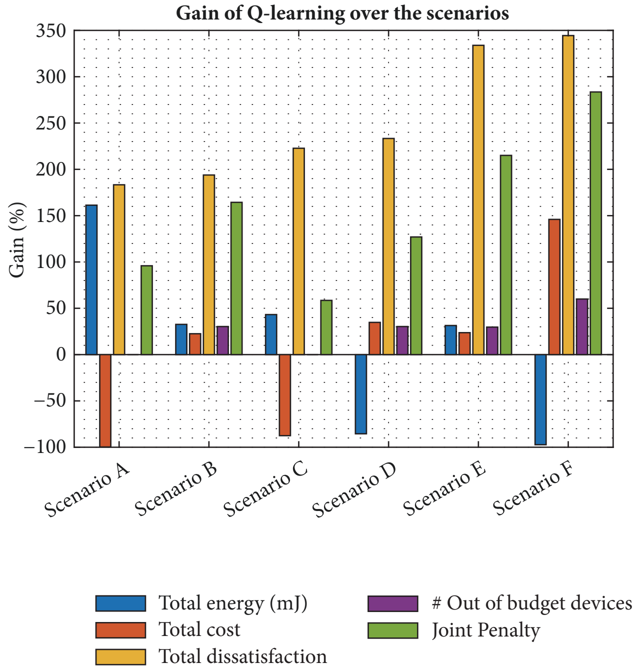


Figure 5: Summary of results for ς = 0.1. Positive and negative values reflect gain and loss, respectively. Gain/loss occurs when the *Q*-learning/scenarios is better than the scenarios/*Q*-learning.

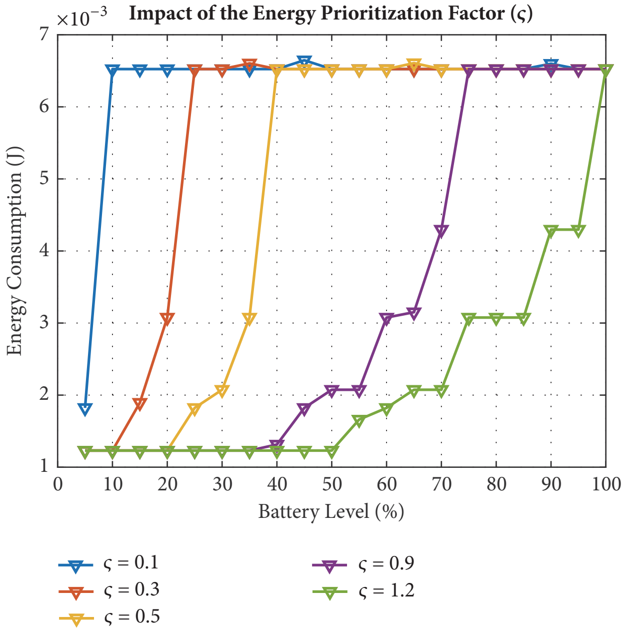


Figure 6: Impact of energy prioritisation factor ς.

Moreover, in this scenario, connected devices are more than two times more likely to be dissatisfied with one or more of the context-aware requirements.

Next, we examine the impact of the battery priority factor, ς, on the energy efficiency. As shown in Figure 6, low values of ς result in almost neglecting the battery life of the device in the optimisation process until it drops below 10%. Very high values of ς prioritise the reduction of energy consumption for all devices except those that have higher than 70% battery life. To this end, it is possible to tune this parameter depending on the scenario at hand and in a device-specific manner. For instance, some devices may be part of a moving vehicle with the possibility of agile and low cost battery replenishment. Such devices may benefit from low settings of ς to allow more flexibility in meeting the remaining constraints. Other devices may be in hard-to-reach places and would require skilled force, special equipment, and hence high cost to replace the dead battery. In this case, higher settings of ς are more suitable and would result in better cost to quality ratio.

The simulation results achieved in this work are very promising, as they indicate a large margin for improvement that is not possible in fixed connection schemes. The proposed reinforcement learning method relies on centralised intelligence, which has access to all the constraints and requirements of all devices, gateways, and connections. Hence, the *Q*-learning-based method selects the best action (connection type/processing location pair in the first stage, and amount of data to be transmitted in the second stage) after the convergence. We appreciate that such a deployment is not realistic and propose to explore the feasibility and corresponding gains of multiagent and distributed reinforcement learning, as adopted in [24], in our future work. Nonetheless, this work is undoubtedly the first to highlight the importance of context-aware connectivity in the IoT context that addresses jointly security, energy, and computational power as well as cost. We present a new application, Smart Ports, and quantify the potential margin for improvement by employing the novel scheme and highlight its effects on the application.

6. Conclusion

In this work, we have presented novel approach for energy-aware and context-aware IoT connectivity that jointly optimises the energy, security, computational power, and response time of the connection. The proposed scheme employs reinforcement learning and manages to achieve a holistic gain of up to 283.54% compared to deterministic routes. Although some deterministic scenarios may result in lower computational cost or lower energy consumption, none is able to meet the holistic context-aware performance target. In addition, we presented an analysis of the impact of the energy prioritisation factor in which we demonstrated the importance of tuning this parameter in a device-centric manner in order to achieve better optimisation of the whole system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partly funded by EPSRC Global Challenges Research Fund—the DARE Project—EP/P028764/1. The first author was supported by the Republic of Turkey Ministry of National Education (MoNE-1416/YLSY).

References

[1] S. Andreev, O. Galinina, A. Pyattaev et al., “Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap,” *IEEE Communications Magazine*, vol. 53, no. 9, pp. 32–40, 2015.

[2] L. Atzori, A. Iera, and G. Morabito, “The internet of things: a survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] 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, 2017.

[4] Y. Yang, M. Zhong, H. Yao, F. Yu, X. Fu, and O. Postolache, “Internet of things for smart ports: technologies and challenges,” *IEEE Instrumentation Measurement Magazine*, vol. 21, no. 1, pp. 34–43, 2018.

[5] GSMA, “3GPP low power wide area technologies,” GSMA, White paper, Oct 2016.

[6] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); LTE coverage enhancements,” 3GPP Thechnical Report 36, Jun 2012.

[7] Technologies Keysight, “The menu at the IoT cafe: a guide to IoT wireless technologies,” *Application Note*, 2017.

[8] L. Farhan, S. T. Shukur, A. E. Alissa, M. Alrweg, U. Raza, and R. Kharel, “A survey on the challenges and opportunities of the Internet of Things (IoT),” in *Proceedings of the 2017 Eleventh International Conference on Sensing Technology (ICST)*, pp. 1–5, December 2017.

[9] S. Tayade, P. Rost, A. Maeder, and H. D. Schotten, “Device-centric energy optimization for edge cloud offloading,” in *Proceedings of the 2017 IEEE Global Communications Conference (GLOBECOM 2017)*, pp. 1–7, Singapore, December 2017.

[10] F. Renna, J. Doyle, V. Giotsas, and Y. Andreopoulos, “Query processing for the internet-of-things: coupling of device energy consumption and cloud infrastructure billing,” in *Proceedings of the 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 83–94, Berlin, Germany, April 2016.

[11] S. Persia, C. Carciofi, and M. Faccioli, “NB-IoT and LoRA connectivity analysis for M2M/IoT smart grids applications,” in *Proceedings of the 2017 AEIT International Annual Conference*, pp. 1–6, Cagliari, September 2017.

[12] A. Mihovska and M. Sarkar, “Smart connectivity for internet of things (IoT) applications,” in *New Advances in the Internet of Things*, vol. 715 of *Studies in Computational Intelligence*, pp. 105–118, Springer International Publishing, Cham, 2018.

[13] N. Kouzayha, M. Jaber, and Z. Dawy, “M2M data aggregation over cellular networks: signaling-delay trade-offs,” in *Proceedings of the 2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 1155–1160, December 2014.

[14] J. Xu, L. Chen, and P. Zhou, “Joint service caching and task offloading for mobile edge computing in dense networks,” ArXiv e-prints 1801.05868, Jan 2018.

[15] O. Y. Bursalioglu, Z. Li, C. Wang, and H. Papadopoulos, “Efficient C-RAN random access for IoT devices: learning links via recommendation systems,” ArXiv e-prints 1801.04001, Jan 2018.

[16] 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, 2018.

[17] E. Oyekanlu, “Predictive edge computing for time series of industrial IoT and large scale critical infrastructure based on open-source software analytic of big data,” in *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data)*, pp. 1663–1669, Boston, MA, USA, December 2017.

[18] S. Barbarossa, S. Sardellitti, E. Ceci, and M. Merluzzi, “The edge cloud: a holistic view of communication, computation and caching,” ArXiv e-prints 1802.00700, Feb 2018.

[19] T. X. Vu, S. Chatzinotas, and B. Ottersten, “Edge-caching wireless networks: performance analysis and optimization,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2827–2839, 2018.

[20] ITU-R, “Propagation data and prediction methods for the planning of short-range outdoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz,” *International Telecommunication Union—Radiocommunication Sector, Geneva*, 2017, Recommendation ITU-R P.1411-9.

[21] 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.

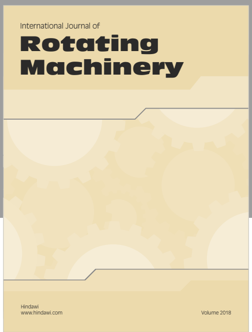
[22] E. M. Russek, I. Momennejad, M. M. Botvinick, S. J. Gershman, and N. D. Daw, “Predictive representations can link model-based reinforcement learning to model-free mechanisms,” *PLoS Computational Biology*, vol. 13, no. 9, Article ID e1005768, 2017.

[23] E. Even-Dar and Y. Mansour, “Convergence of optimistic and incremental q-learning,” in *Advances in Neural Information Processing Systems*, pp. 1499–1506, 2002.

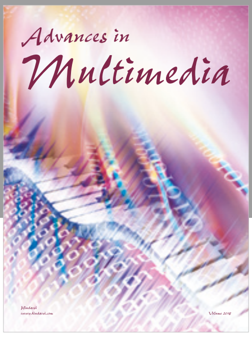
[24] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, “A distributed SON-based user-centric backhaul provisioning scheme,” *IEEE Access*, vol. 4, pp. 2314–2330, 2016.

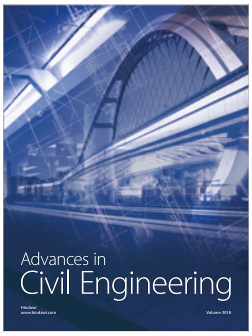
Journal of
Engineering

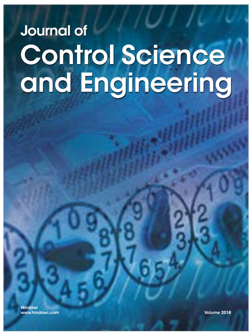
The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

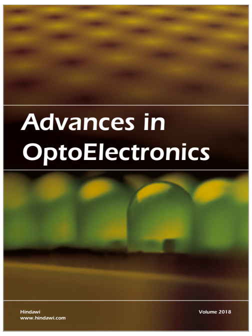

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

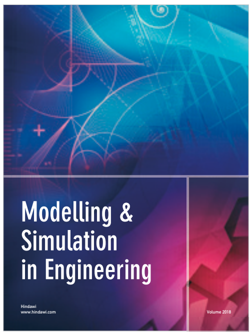
Journal of
Robotics

Journal of
Electrical and Computer Engineering

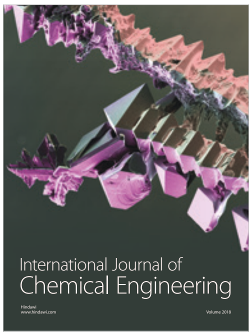

Advances in
OptoElectronics

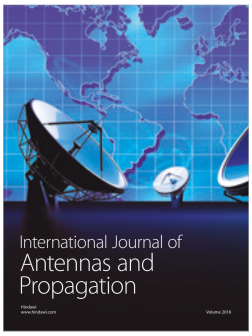
VLSI Design

International Journal of
Navigation and Observation

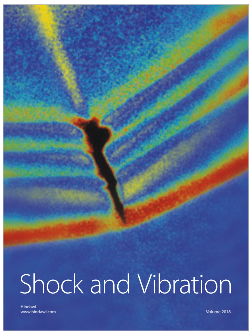

Modelling & Simulation in Engineering

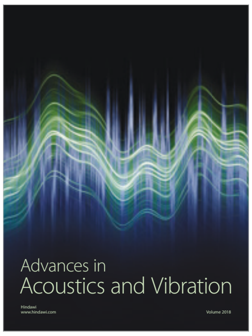
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration