Journal of Network and Systems Management (2021) 29:6
https://doi.org/10.1007/s10922-020-09572-7

A Robust Energy Optimization and Data Reduction Scheme for IoT Based Indoor Environments Using Local Processing Framework

Sujit Bebortta[1] · Amit Kumar Singh[2] · Bibudhendu Pati[3] · Dilip Senapati[1]

Received: 20 May 2020 / Revised: 15 August 2020 / Accepted: 25 September 2020

Abstract

The extensive growth in popularity of Internet of Things (IoT) has led to the generation of massive amount of data from several heterogeneous sensory devices. This has also led to the increase in energy consumption by these connected devices. Smart buildings are one such platform which are equipped with several micro-controllers and sensors, generating a huge amount of redundant information at their data acquisition level. As a result, real-time applications may not be efficiently executed due to latency delays at the cloud service end. This requires several devices at cloud service end to execute the massive amount of data generated by these sensors, which does not satisfy green computing criteria. In this context, a novel local processing mechanism (LPM) is proposed, which favors an improved IoT service architecture for smart buildings. From the perspective of green computing, the proposed LPM framework facilitates reduction of manifolds at data acquisition level of sensor nodes. This paper also addresses the concept of optimal use of sensors in a wireless sensor network (WSN) and estimates costs corresponding to non-Poisson and Poisson arrival of data packets at local processor using the well-known queuing model. We also provide an efficient algorithm for smart buildings using our expert Markov switching (EMS) model, which is a well known probabilistic model in the field of artificial intelligence (AI) for subjectively validating real sensory data sets (viz., temperature, pressure, and humidity). Further, it has been analyzed that the proposed EMS algorithm outperforms several other algorithms conventionally used for determining the state of large-scale dynamic sensor networks. The service cost of proposed model has been compared with conventional model under various stress conditions viz., arrival rate, service rate, and number of clusters. It is observed that the proposed model operates well by leveraging green computing criteria. Thus, in the aforementioned context, this paper provides thing-centric, data-centric, and service-oriented IoT architecture.

Extended author information available on the last page of the article

Keywords Internet of Things · Wireless sensor network · Markov switching · Cloud computing · Optimal sensor section · Cost optimization · Poisson distribution · Queueing theory

1 Introduction

In recent times, low cost micro-controllers connect almost every physical object to the Internet by employing sophisticated communication protocols which is collectively termed as the Internet of Things (IoT). The interaction between smart objects connected to IoT and management of massive data generated by these devices can be possible through the cloud service mechanism using a high speed active internet [1, 2]. This process generates a massive amount of data which requires more processing time for storing and accessing. As the connected devices scale-up progressively and continually generate data leading to the big data issue in IoT which increases network latency at the cloud end. Various sensors (such as image sensors, temperature sensors, motion sensors, and so forth) can be deployed for sensing different events (such as intrusion detection, environment monitoring, detecting vibrations, and so forth) in a smart environment [3]. The information extracted from the above sensors may contain highly correlated and redundant information from its succeeding to preceding states. As a result, the storage and retrieval of data at cloud end leads to a bottleneck situation. In this context, this paper modifies the existing IoT architecture for smart buildings by introducing the local processing mechanism (LPM). In the local processing architecture, all the local data are processed and the relevant information is transmitted to cloud servers for future access. Consequently, the LPM framework indirectly assists to extract the valuable information from the cloud with minimal time and cost as the data is free from redundancy.

The quantum leap of smart homes and smart infrastructures have made IoT a substantially pervasive juncture for connecting smart objects. There have been momentous growth in areas of embedded systems, network technologies, semantic inter-operability, and perseverance of complexity in information which have gained significant development in the fields of IoT [4–6]. IoT has acquired much amiability in various sectors (*viz.*, industries, healthcare, transportation, and households) over the last decade for which several experimental test beds have been set up by deploying different sensors [7–11]. Thus, IoT can be said to have self-constituting potentials and has become the paradigm for networking of several physical and virtual "things". With persistent progress in integration of several IoT devices, the data generated from the 'things' can be piled up using cloud services [12]. Thus, it is required to consider a non-exhaustive approach for integrating IoT architecture with the cloud services for processing huge data generated from the sensors [13–17].

The optimization of energy is a prime concern for densely deployed sensors in a wireless sensor network (WSN). Thus, certain schemes can be adopted to manage the number of sensor nodes in a WSN for their active participation in sensing an event. In this paper, we formulate the problem for selecting an active sensor node using a probabilistic framework. Therefore, the data sensed by these sensor nodes are uncorrelated which leads to a better utilization of sensed data. In addition to this, the obtained model logically reduces the number of data packets to be transferred and consequently reduces latency delays at the cloud end. Eventually, this becomes an ideal choice for maximum utilization of memory space which also reduces the data transmission costs.

The sensors participating in an IoT application mostly require continuous power supply for processing data and for inter-device communication. Moreover, to obtain accuracy in the perceived data we need to ensure that there is no sensor node failure while sensing data from the surrounding [18–20]. In this view, a more economic, energy-aware, and reliable approach is required to optimize the number of sensors involved in a WSN. In this paper, we present an optimal sensor selection (OSS) model which provides a solution to the aforementioned problem using the economic quantity (EQ) model [21, 22] and determines the optimal number of active sensor nodes for a WSN. We further derive a probabilistic model to test the reliability of sensor nodes pertaining to the Erlang model. This model successfully eliminates passive sensors which minimizes power consumption, and provides more accurate, and reliable acquisition of data. After the completion of sensing phase, the sensed data packets arrive at the local processor in a non-Poisson pattern. Since most of the performance measured models are based on Poisson distribution [23–25], the data packets are converted into a Poisson pattern using LPM framework that captures non-redundant data form the massive collection of correlated data. Finally, these data packets are queued at the gateways and transmitted to cloud storage.

The Markov Switching model [26], is a state switching autoregressive model mostly used to illustrate the transition between distinct states and it is widely used for state-space modeling of spatio-temporal datasets. In this paper, we present a multivariate thresholding model using Markov Switching for smart building monitoring, we collectively acronym it as the expert Markov switching (EMS) model. The information regarding surveillance of various environmental stresses in a smart building can be captured by the sensors and different situations like intrusion, gas leakages, fire, etc., can be identified through various threshold values. We have also provided an algorithm (Algorithm 1) corresponding to the EMS model and have validated it using real datasets [27] with suitable threshold values. Further, the efficiency of EMS algorithm is obtained by comparing the results with four conventionally used sensor state selection algorithms for large-scale dynamic WSNs namely the simplified greedy sensor selection (SGSS) algorithm [28], alternating direction method of multipliers (ADMM) algorithm [29], Lyapunov equation based greedy algorithm (LE-GA) [30], and discrete algebraic Riccati equation based greedy algorithm (DARE-GA) [31]. It was observed that the proposed EMS algorithm provided better computational performance in contrast to the four algorithms considered in this study. Further, in this paper, we present the cost models in compliance with OSS model which provides an estimate of the total expected cost corresponding to deterministic and probabilistic approaches. Finally, we evaluate the processing cost for data transmission using LPM and without using LPM frameworks based on well-known queuing theory models [32, 33]. A comparison for the above models is shown in the succeeding sections.

The major contributions of this paper can be summarized as follows:

- We provide a probabilistic framework for active sensor node selection in a WSN. We further use our OSS model to determine the optimal number of active sensor nodes, which collectively provides a things-centric energy efficient approach for Green-IoT.
- The data packets acquired by active sensor nodes are processed using our proposed LPM framework and generate them in a Poisson pattern.
- We present the EMS model which works on the principle of information thresholding and significantly reduces processing of redundant information for the use case of a smart building.
- The performance of the EMS algorithm is evaluated in convergence with four conventionally used algorithms [28–31]. It is observed that the proposed EMS algorithm considerably outperforms the computational complexities of these algorithms.
- We then provide the derivation of cost models based on LPM and OSS model.
- Finally, we discuss the results obtained by deploying various sensors in a smart building and show the validation of our EMS algorithm.

The remaining part of this paper is classified as follows: In Sect. 2, we discuss various related works. Section 3, presents use case study of the smart building architecture along with its IoT applications and cloud services. In Sect. 4, we provide a probabilistic model for selecting active sensor nodes and derive the OSS model. This section also formulates the data centric EMS model. We then compute total expected cost corresponding to OSS model in Sect. 5. In Sect. 6, we obtain processing cost for transmitting the data packets using LPM and without using LPM framework. Section 7, deals with the results and discussions complying to our proposed framework. Finally, the last section of this paper (i.e., Sect. 8) ends with brief conclusions and relevant future scopes.

2 Related Work

In this section, we present a taxonomy of several related works focusing on the prevailing and emerging technologies for IoT and smart environments. In smart environments, large interconnected networks are involved, this entails superior network configuration and discovery of services as a crucial issue for researchers. Schor et al. [34], presented a web based approach for integrating interconnected sensors and actuators involved in a smart building. They presented a technique for detection of services which reduces the complexities involved in configuring WSNs in a smart building.

In pervasively growing computing environments, the need for computation offloading is actively required to meet inadequacy in the resources. Soliman et al. [35] provided an integration of IoT along with cloud services for smart homes. Their work presented an interactive environment for controlling smart home appliances and their operations which used cloud computing to increase the accessibility of resources. Fan et al. [36] in their work presented a survey regarding challenges and scopes for optimal energy utilization in IoT based smart grids. Their work highlights the techniques for providing an improved and optimized usage of energy levels in distributed and real-time systems. Sembroiz et al. [37] presented a cloud based architecture for IoT based platforms. They used cloud as a middleware for smart buildings which facilitated rapid replication of databases as per the user's demands. Chien et al. [38], provided a service oriented architecture for smart cities. Further, they provided a heuristic based algorithm for mitigating the load and enhancing network performance. Yassine et al. [39], provided an IoT based platform for handling smart home data. This work facilitated operations like storing, processing, and classifying data collected from smart environments, providing a data-driven service to the users.

3 Architecture

In our use case of the smart building, we consider deployment of various sensors such as temperature sensors, humidity sensors, and pressure sensors along with HVAC systems for sensing and supervising prevailing thermal conditions of the smart building environment.

3.1 Clustering of Nodes

The sensors are deployed for administering events prevailing in a smart building. Each dwelling in the building is a smart home equipped with smart appliances, computing devices (*viz.*, PCs, PDAs), HVACs, etc., [40, 41]. The sensors deployed in the building are coalesced to form several clusters and each cluster constitutes of a cluster head [42–44]. The node with minimal residual energy is chosen as the cluster head. The cluster head makes use of Carrier Sense Multiple Access (CSMA) protocol for communicating with all the nodes in a WSN. Clustering of sensor nodes increases the performance of the system by incrementing availability of memory resources which intensifies the scalability [45].

Since, we are considering a heterogeneous sensor network to facilitate acquisition of data from a variety of sources, the information captured by sensors are superfluous. So to eliminate these redundant data, we use the LPM technique. The data packets arriving at the local processor initially follows a non-Poisson arrival pattern. These data packets are then processed using the LPM technique, and queued in a discretized manner following a Poisson distribution. The data packets obtained using this mechanism are forwarded to the cloud.

Figure 1, illustrates a visual representation for the clustering of different sensor nodes deployed in a smart building with each sensor having different functionalities. The cluster head communicates the inferred data to transceiver station from where the data is forwarded to local processors. As the cluster head communicates the data perceived by various sensors in a cluster, the transmitted data may contain redundant information which leads to latency delays at the cloud end. However, by employing our LPM framework, we can attain the reduction of manifolds in information acquired by the local processors.

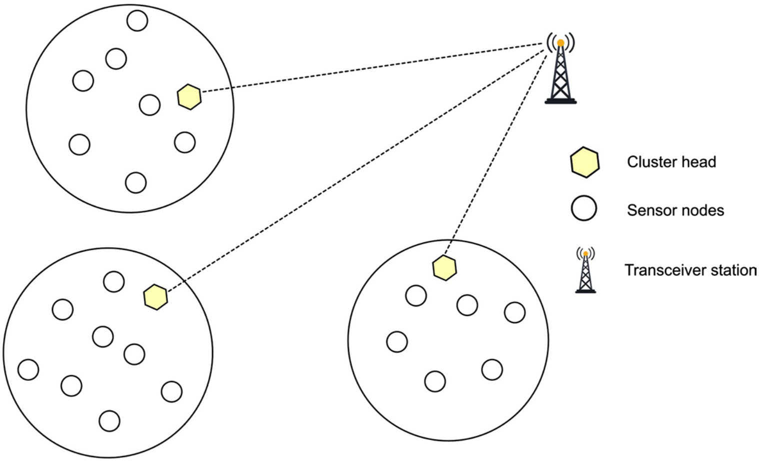

Fig. 1 A pictorial representation depicting cluster formation of the sensor nodes

Figure 2, shows a service oriented IoT architecture for smart buildings along with a pictorial representation of various phases of data transmission. The data perceived by sensors is first locally processed by the local processor using the proposed LPM model. These data are then transmitted to gateways from where they are transmitted over the Internet to cloud storage [46, 47]. This enables the users to achieve a real-time monitoring of smart buildings. The data regarding various conditions of smart buildings can be eventually accessed by the users (or, dwellers of the building) over their smart phones or other connected devices.

In Fig. 3, we present a sketch of the experimental setting for the deployment of sensors in a smart building along with several auxiliary devices. The devices namely, humidifier, dehumidifier, air compressor, pressurizer, room heaters, air conditioners, air filters, etc., constitute the HVAC system, which assists in monitoring air quality of a building. In our setting, the EMS algorithm (Algorithm 1), is used to monitor the smart building's air quality. This algorithm works on principle of information thresholding and simplifies the functioning of HVAC system. When the humidity in a certain zone of the building increases then the EMS algorithm generates a message for activating the dehumidifying equipments. Similarly, when humidity level falls below the comfort level, the humidifier is activated. The HVAC systems activate pressurizer if the pressure in the building's environment falls below a certain threshold. Likewise, if the pressure is too high then the ventilation systems are activated which lets in warmer air from the surrounding and lowers the air pressure. The heaters installed in the building are activated if temperature of the considered area of building drops below the minimum threshold value. Similarly, if temperature of the building exceeds threshold of the thermal comfort level then the air conditioners are activated for cooling down the building's temperature.

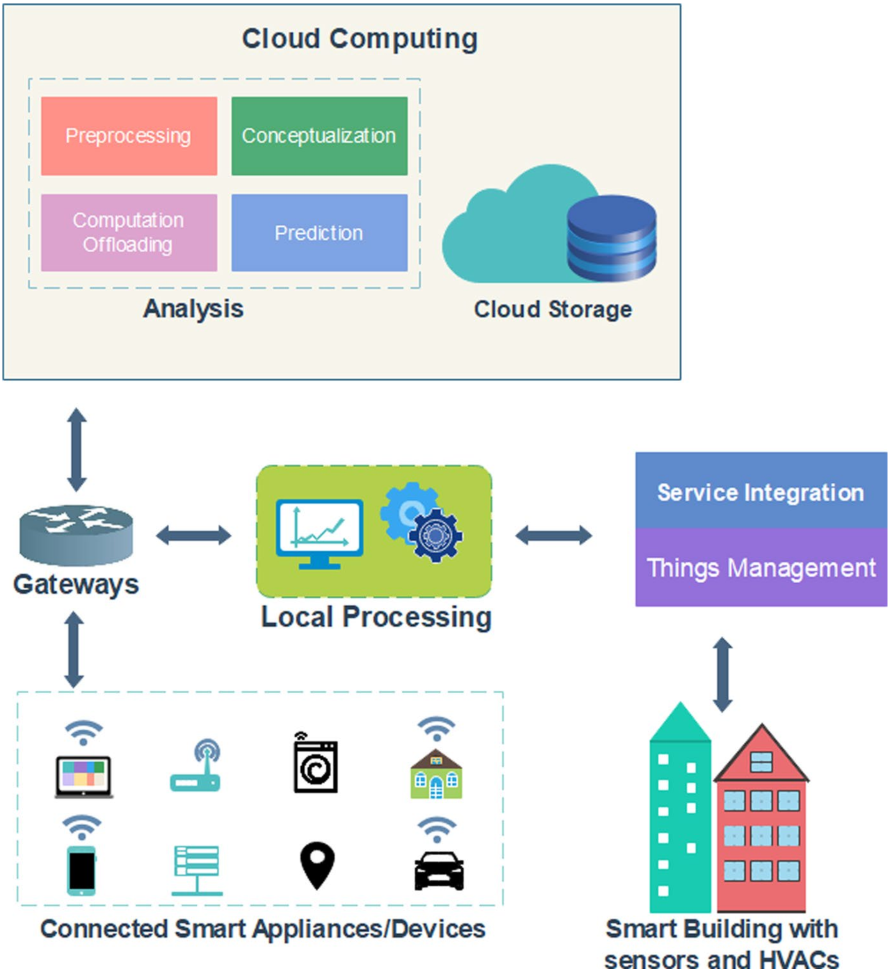

Fig. 2 A functional overview for deployment of various sensors in a smart building and the proposed LPM framework

3.2 Requirements and Set-up

For local execution of the data packets generated from sensors, we do not require the active internet support, or any access to the cloud. This makes it more preferable over various technologies used for smart homes, or buildings. For local processing, we require a multiport hub (like the Hub v2.0) for its ease of installation and maintenance.

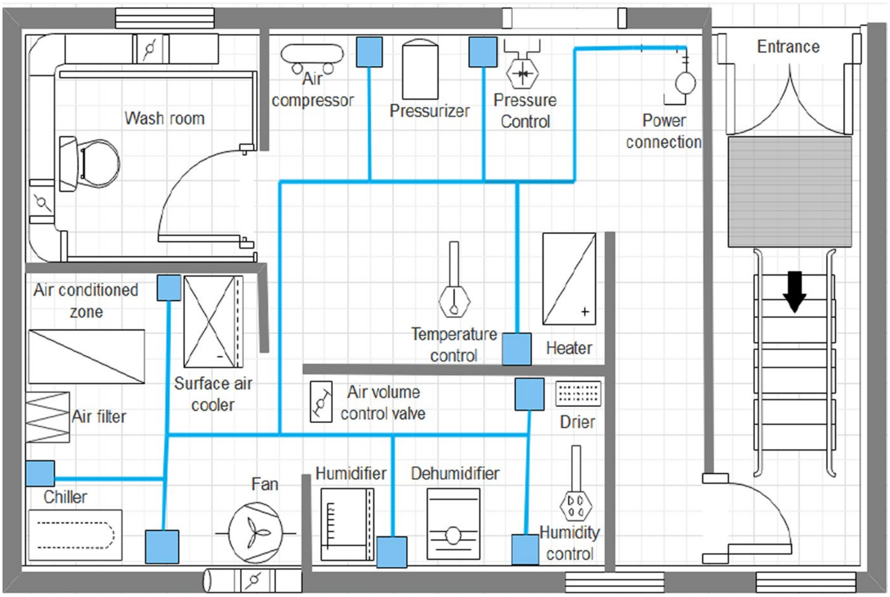

Fig. 3 An overview of smart building depicting the HVAC system equipped with sensors

3.3 Connectivity

Almost all the devices communicating through Zigbee or, Z-wave can communicate locally. In our model, Zigbee an IEEE 802.15.4 standard is used, for connecting the sensors in a WSN. The Zigbee specification is the most acceptable Personal Area Network(PAN) employed for smart homes, smart buildings, and smart Industry networks. It is mostly used for connecting devices with low power requirements (like smart home sensors). It also supports multi-hop transmission of data in a WSN. Although the Zigbee network layer reinforces different topologies, but is mostly associated with mesh network over other network topologies because of its stability and reliability. After all the set-up requirements are successfully implemented, the transmission of data packets can be done by sensor nodes using suitable routing algorithms [48].

3.4 Transmission

Before the data packets are communicated globally to the cloud, they are locally processed using LPM scheme. In our mechanism, we focus on the spatial reduction of manifolds (discussed in Sect. 6.1). This process can be explained in two distinct phases:

3.4.1 Phase-1

In this phase, we first select an optimal set of active sensor nodes using the proposed frameworks (discussed in Sects. 4.1 and 4.2). These active nodes are used in sensing different events in a smart building.

3.4.2 Phase-2

In the second phase, we concentrate on acquisition of data originating from various active sensor nodes. The sensed data obtained from active sensor nodes is processed using LPM framework which reduces data redundancy and can be transmitted to the cloud end for further processing.

4 Problem Formulation

In this section, we present a probabilistic approach to select active sensor nodes in a WSN. We also present our optimal sensor selection (OSS) model subject to the set-up cost and power supply cost. The probabilistic approach for OSS model is also derived in the aspect of reliability of sensor nodes. Finally towards the end of this section, we provide the EMS model and its corresponding efficient EMS algorithm. The working principle of this model is based on partitioning of information sets using various predefined threshold values. Table 1 provides a detailed list of all the variables used in this paper along with their respective illustrations.

Table 1 List of the mathematical notations used

Symbol	Description
S_i	Set of i^{th} sensor nodes
k	Number of clusters in a WSN.
k_a	Number of active clusters in the same network
S	Total number of sensors
C_p	Power supply cost for each sensor node
$F_p(S)$	Total power supply cost for sensors
t	Number of sensors in a cluster
C_s	Set-up cost for each cluster
$\Phi_{j,i}$	Weights of preceding information for j^{th} states
$a_{j,t}$	Additive white Gaussian noises
$T^{\ell}_{j,1}, T^{\ell}_{j,2}$	Two extreme threshold values
T_e	Total expected cost
κ	Redundancy factor for data packets
C_{LPM}, C	Service costs for LPM framework and without using LPM framework
λ_{arr}, μ	Arrival rate and processing rate of the data packets
C_1, C_2	Processing cost and waiting cost per each data packet
L_1, L_2	Data packets generated by local processor and sensor nodes respectively

4.1 Procedure for Selecting an Active Sensor Node

In our framework, we consider a large WSN with densely deployed sensor nodes possessing the ability to sense various events. Here, we formulate a method which probabilistically configures a solution for the selection of active sensor nodes in a WSN. Let us consider a large random network comprising of several sensor nodes having distinct prototypes and varying potentials. These sensors share a scale-invariant connectivity among themselves and generate correlated data packets at any instance of time. Thus, if the WSN is constituted of $\{S_i,\ i = 1, 2, ..., S\}$ sensors, such that the number of clusters formed by clustering S_i sensors in the network be denoted as k clusters, and k_a represents the number of active clusters in the same network. Let X be a random variable which represents number of active sensor nodes among k_a active clusters in a total of k clusters and its probability function is defined as:

$$p(x) = \frac{1}{z}\binom{k}{k_a}\binom{k_a}{x}\frac{e^{-\lambda_{arr}}\lambda_{arr}^{x}}{x!}, \tag{1}$$

where, z is the normalization constant and $\binom{k}{k_a} = \frac{k!}{k_a!(k-k_a)!}$.

If $k = k_a$ that means the number of active cluster is equal to number of total clusters in the WSN. Hence, Eq. (1) becomes the standard Poisson distribution. Now, it becomes necessary to determine number of sensor nodes required for accomplishing the sensing phenomena, which can be resolved by using the OSS model.

4.2 Optimal Sensor Selection Model

In a WSN, determining the optimal selection of active sensor nodes becomes increasingly inevitable due to aperiodic switching between sensors at the time of data acquisition. The devices are densely integrated to meet persistent changes in environment for accomplishing a specific task. In order to meet these demands, we implement our OSS model to quantify the sensor nodes in convergence to their basic traits, *viz.*, capability, functionality, user requirements, and so forth.

4.2.1 Deterministic OSS Model

It is known that the average set-up cost decreases with the increase in the number of sensors. Similarly, the power consumption cost increases if number of sensors increase. So the problem of interest is to provide a balance between the set-up cost and power consumption cost of sensors. Figure 4 depicts the optimal number of sensors subject to set-up cost and power consumption cost.

Let S represents number of sensors, and C_p be the estimated power supply cost for each sensor, hence the total power supply cost for S sensors can be defined as,

$$F_p(S) = SC_p. \tag{2}$$

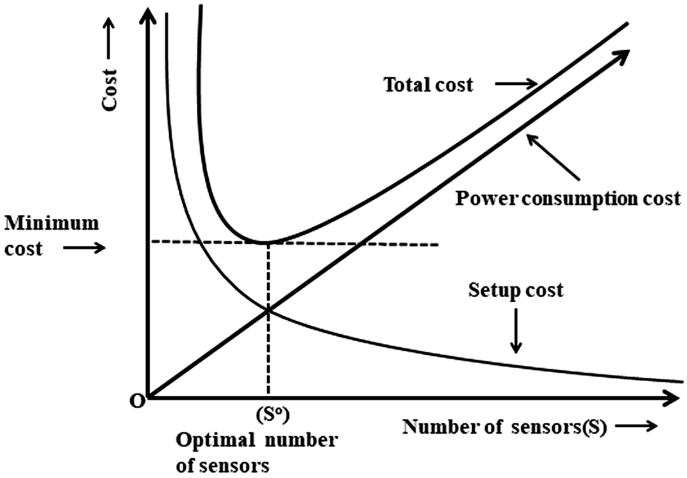

Fig. 4 Quantitative representation of set-up cost and power consumption cost

Let C_s be set-up cost for each cluster, then the total set-up cost for S sensors can be given as,

$$F_s(S) = \frac{t}{S} k C_s, \tag{3}$$

where t denotes the number of sensors contained in a cluster, and k denotes total number of clusters present in the WSN.

The optimal number of sensors can be obtained in Fig. 3, where the total power supply cost is equal to total set-up cost. This relationship can be defined as,

$$SC_p = \frac{t}{S} k C_s. \tag{4}$$

Thus, we obtain the optimum value for S as follows,

$$S^o = \sqrt{\frac{tkC_s}{C_p}}. \tag{5}$$

From Eq. 5, we obtain the deterministic OSS model.

4.2.2 Probabilistic OSS Model

We use probabilistic OSS model for optimal selection of sensor nodes where the required number of sensors in a WSN is subject to uncertainty. Thus, we determine reliability of the number of sensors taken into consideration in order to obtain a delay tolerant and fault resistant perception of data. The reliability of sensor nodes can be characterized by the well-known Erlang distribution [49–51], which is defined as,

$$g(t) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!},\ k > 0,\ \lambda \geq 0, \tag{6}$$

where λ is the rate parameter and represents the failure rate of the sensors.

Hence, the optimal number of sensors is defined as,

$$\hat{S} = \int_0^\infty \sqrt{\frac{tkC_s}{C_p}}\, g(t)\, dt. \tag{7}$$

Following Ryzhik et al. [52], the integral in Eq. (7) can be written as,

$$\hat{S} = \sqrt{\frac{kC_s}{C_p}} \frac{\lambda^{-1/2}}{\Gamma(k)} \Gamma\left(k + \frac{1}{2}\right), \tag{8}$$

where $\Gamma(.)$ represents the Gamma function.

Thus, from Sects. 4.1 and 4.2, we obtain a things-centric approach for green-IoT by reducing the number of "things" (i.e., sensor nodes) connected in a WSN by employing the OSS framework.

4.3 Expert Markov Switching Model

The Markov switching model is usually employed to model aperiodic jumps between various states on the basis of their preceding information sets [26]. Here we have designed an expert Markov switching (EMS) model for smart buildings having three states defined by two threshold values. The EMS model is defined as:

$$x_{j,t} = \begin{cases} x_{j,0} + \sum_{i=1}^{N} \Phi_{j,i} x_{j,t-i} + a_{j,t}, & x_{j,t} < T^{\ell}_{j,1} \\ x_{j,0} + \sum_{i=1}^{N} \Phi_{j,i} x_{j,t-i} + a_{j,t}, & T^{\ell}_{j,1} \leq x_{j,t} < T^{\ell}_{j,2} \\ x_{j,0} + \sum_{i=1}^{N} \Phi_{j,i} x_{j,t-i} + a_{j,t}, & x_{j,t} \geq T^{\ell}_{j,2} \end{cases}, \tag{9}$$

where $\boldsymbol{\Phi}_{j,i}$ are the weights of preceding information for j^{th} states, $a_{j,t}$ are sequence of independent and identically distributed (i.i.d) white Gaussian noise sequence with mean zero and covariance matrix I_3 i.e., $a_{\mathrm{j,t}} \sim N(0, I_3)$, $I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, and $x_{j,t}$ is the threshold variable.

Algorithm 1, shows the working procedure of EMS model. In our algorithm, the cumulative value of $x_{j,t}$ can be obtained from the weights of preceding information sets. We consider two threshold values $T^{\ell}_{j,1}$ and $T^{\ell}_{j,2}$ defined over $x_{j,t}$ as the minimum and maximum threshold values respectively. If threshold value $x_{j,t} < T^{\ell}_{j,1}$, the EMS model generates an alert to activate the HVAC systems. If $T^{\ell}_{j,1} \leq x_{j,t} < T^{\ell}_{j,2}$, the EMS performs no actuation, indicating normal conditions. In contrast, if $x_{j,t} \geq T^{\ell}_{j,2}$, the changes observed are major, and the system recommends some obligatory actions. In our use case of the smart building, the EMS algorithm can be employed for air quality monitoring. This algorithm makes use of the preceding air quality information and suggests the necessary actions to be taken on the observed data. The HVAC systems installed in the building can hence be activated and deactivated upon the recommendations of EMS algorithm.

Algorithm 1: A real-time EMS algorithm for the surveillance of smart buildings.

Load: Sensory data;
Compute: N=Length(data);
Input : Threshold limits $T^{\ell}_{j,1}$ and $T^{\ell}_{j,2}$;
Output : Threshold $x_{j,t}$;
Initialize $\Phi_{j,i}$, where $j = 1, 2, 3$, and $i = 1, 2, ..., N$;
Initialize Gaussian white noise $a_{j,t}$;
Initialize seed x_0 ;
if $x_{j,t} < T^{\ell}_{j,1}$ **then**
  $x_{j,t} = x_{j,0} + \sum_{i=1}^{N} \Phi_{1,i} x_{j,t-i} + a_{j,t}$;
  $Message: \ Activate \ HVAC$;
else if $x_{j,t} \geq T^{\ell}_{j,1}$ *and* $x_{j,t} < T^{\ell}_{j2}$ **then**
  $x_{j,t} = x_{j,0} + \sum_{i=1}^{N} \Phi_{2,i} x_{j,t-i} + a_{j,t}$;
  $Message: \ No \ change$;
else if $x_{j,t} \geq T^{\ell}_{j,2}$ **then**
  $x_{j,t} = x_{j,0} + \sum_{i=1}^{N} \Phi_{3,i} x_{j,t-i} + a_{j,t}$;
  $Message: \ Recommend \ necessary \ actions$;
else
  return $x_{j,t}$;
end

5 Optimal Cost Model

This section deals with derivation of two performance metrics such as total expected cost corresponding to deterministic and probabilistic models. It explicitly characterizes the problem for OSS model in deriving the cost functions pertaining to utilization of sensor nodes and their energy consumption criteria.

5.1 Deterministic Cost Model

The total expected cost corresponding to optimal number of active sensors is given by,

$$T_e(S|C_s, t) = SC_p + \frac{t}{S} C_s k. \tag{10}$$

The necessary condition for optimality can be defined as follows,

$$\frac{dT_e}{dS} = C_p - \frac{t}{S^2} C_s k \equiv 0. \tag{11}$$

Since,

$$\frac{d^2 T_e}{dS^2} = 2\frac{tkC_s}{S^3} > 0. \tag{12}$$

Therefore using Eq. (5), the total optimal expected cost can be obtained as,

$$T_e(S|C_s, t) = 2\sqrt{tkC_s C_p}. \tag{13}$$

5.2 Probabilistic Cost Model

Considering the reliability of sensor nodes in a WSN, we define the probabilistic model for total expected cost as,

$$T(S) = \int_0^\infty T(S|t)\, g(t)\, dt. \tag{14}$$

Using Eq. (6) and following Ryzhik et al. [52], Eq. (14) becomes,

$$T(S) = SC_p + \frac{C_s}{S}\frac{k}{\lambda}. \tag{15}$$

Hence, Eq. (15) provides the probabilistic cost model pertaining to OSS model.

6 Processing Cost

In the smart building architecture, it is worthwhile to calculate cost for transmitting the data packets from sensor nodes to cloud end using the LPM and without using LPM model.

6.1 Service Cost Estimation using Local Processing

The cost for transmitting the data packets from the sensors to the cloud is given by,

$$C_{LPM}(\mu) = \frac{C_1 \mu}{\kappa} + C_2 L_1, \tag{16}$$

where μ is the processing rate of data packets, C_1 is the processing cost per data packet, C_2 is the waiting cost for each data packet before being uploaded to cloud storage, and L_1 denotes number of data packets generated by the local processor.

Figure 5, provides the architecture illustrating local processing criteria. From this architecture, it is observed that the data packets follow a non-Poisson arrival rate at the local processor, and follow a Poisson pattern while being transmitted from the local processors after being locally processed.

From the conventional queuing model, $(M/M/1) : (\infty/FCFS)$ we have [53],

$$C_{LPM}(\mu) = \frac{C_1\mu}{\kappa} + C_2\left(\frac{\lambda_{arr}}{\mu - \lambda_{arr}}\right). \tag{17}$$

where λ_{arr} is the rate at which data packets arrive at the local processor, which follows our proposed distribution Eq. (1), and κ denotes the redundancy factor which is responsible for eliminating redundant data packets generated from the local processor. Thus, the necessary condition for minima is obtained as,

$$\frac{dC_{LPM}}{d\mu} = C_1 - \frac{\lambda_{arr}C_2}{\left(\mu - \lambda_{arr}\right)^2} = 0, \tag{18}$$

which yields minimal value for,

$$\hat{\mu} = \lambda_{arr} + \sqrt{\frac{\lambda_{arr}C_2}{C_1}}. \tag{19}$$

Since, $\frac{d^2C_{LPM}}{d\mu^2} = \frac{2\lambda_{arr}C_2}{(\mu-\lambda_{arr})^2}$, if $\mu > \lambda_{arr}$.

Thus, from Eq. (19), we obtain the optimal rate at which the data packets can be processed before being uploaded to cloud. This technique also offloads the computation demands at the cloud end for processing massive amount of data generated from densely deployed sensor nodes.

6.2 Service Cost Estimation Without Using Local Processing

The cost for transmitting the data packets from the sensors to the cloud without employing LPM is given as,

$$C(\mu) = C_1\mu + C_2L_2, \tag{20}$$

where L_2 denotes the number of data packets generated by various sensors, and using $\left(M/E_k/1\right) : (\infty/FCFS)$ we have,

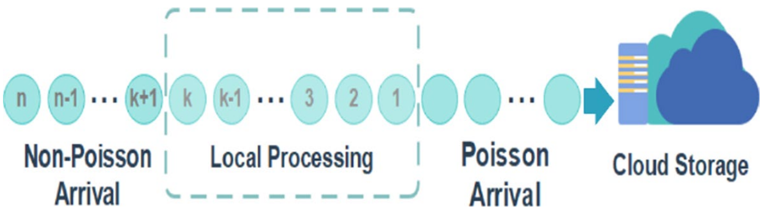

Fig. 5 An architecture for Local processing corresponding to Poisson and non-Poisson arrival pattern of data packets

$$L_2 = \frac{\kappa + 1}{2\kappa} \frac{\lambda_{arr}^2}{\mu\left(\mu - \lambda_{arr}\right)} + \frac{\lambda_{arr}}{\mu}. \tag{21}$$

From Eqs. (20) and (21), we get,

$$C(\mu) = C_1\mu + C_2\left[\frac{\kappa + 1}{2\kappa} \frac{\lambda_{arr}^2}{\mu\left(\mu - \lambda_{arr}\right)} + \frac{\lambda_{arr}}{\mu}\right]. \tag{22}$$

Now from Eq. (22), we obtained as,

$$\frac{dC}{d\mu} = C_1 + C_2\left(-\frac{\lambda_{arr}}{\mu^2} - \frac{(\kappa + 1)(2\mu - \lambda_{arr})\lambda_{arr}^2}{2\kappa\left(\mu(\mu - \lambda_{arr})\right)^2}\right). \tag{23}$$

Thus, the necessary condition for Eq. (23) to obtain the extremal point is given by,

$$\frac{dC}{d\mu} = 0, \;\; \mu = \hat{\mu}. \tag{24}$$

Differentiating the Eq. (23), we get,

$$\frac{d^2C}{d\mu^2} = C_2\left(\frac{2\lambda_{arr}}{\mu^3} - \frac{(\kappa + 1)\lambda_{arr}^2(3\mu^2 + \lambda_{arr}^2 - 3\mu\lambda)}{\kappa\mu^3(\mu - \lambda_{arr})^3}\right). \tag{25}$$

As $\left.\frac{d^2C}{d\mu^2}\right|_{\mu=\hat{\mu}} > 0$, Eq. (20) attains its minimal value at $\mu = \hat{\mu}$.

7 Results and Discussions

7.1 Performance Evaluation of EMS Algorithm

The versatility of any algorithm can be realized by evaluating its computational complexity. The computational complexity is highly reliant on the way an algorithm performs for dynamically changing requirements and the amount of memory occupied for executing a task. Here, the performance of proposed EMS algorithm (presented in Algorithm 1) has been evaluated in convergence with four other popularly used algorithms namely the SGSS algorithm [28], ADMM algorithm [29], LE-GA [30], and DARE-GA [31]. The SGSS algorithm is used in dynamical systems for estimation of sensor states in constrained systems. The ADMM algorithm is popularly used for large-scale systems to address the sensor selection problem. The other two algorithms [30, 31], are based on greedy algorithm for solving the sensor selection problem. They leverage capabilities of Lyapunov equation and DARE for accomplishing optimal solution in terms of cost for selection of sensors. The numerical experiments show that the complexity of the proposed EMS algorithm depicts a slow increase with the number of states as compared with other algorithms. Thus, the contributions of our EMS algorithm are considerably much novel in contrast to the studies made earlier.

We address the issue of device state transition problem for constrained environments by employing the proposed EMS algorithm. Here, each recursive value corresponding to the threshold variable $x_{j,t}$ implements an independent state for respective values of $T^{\ell}_{j,1}$ and $T^{\ell}_{j,2}$. This value further endorses in determining the state of proposed indoor environment monitoring system for optimal management of resources. Thus, following the Master theorem [54], complexity of the proposed algorithm corresponding to each state can be expressed as the relation $T\left(x_{j,t}\right) = T\left(x_{j,t-1}\right) + O(c)$, for any constant c. Hence, exploiting the Markov switching model, the complexity corresponding to EMS algorithm can be conjectured to be $O(N)$. Thus, it can be deduced that complexity of the proposed algorithm is quite optimistic as it is much smaller than the complexities of SGSS, ADMM, LE-GA, and DARE-GA algorithms.

In Fig. 6, we provide an analysis of performance for the proposed EMS algorithm along with SGSS, ADMM, LE-GA, and DARE-GA algorithms for 1000 iterations with the y-axis considered in logarithmic scale. Hence, the computational complexity of our proposed EMS algorithm is obtained as $O(N)$, which outperforms the performance of algorithms proposed in [28–31].

7.2 Numerical Results

7.2.1 Optimal Energy Consumption Model

In order to analyse the validity of our proposed scheme, we rely on the numerical results computed for each model. We consider a network initially consisting of 1000 sensors such that each cluster consists of 5 sensor nodes. Considering the energy

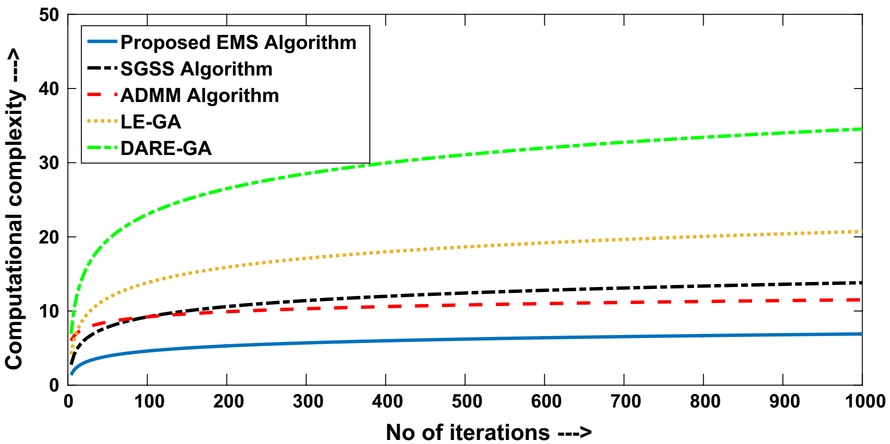

Fig. 6 Performance analysis of proposed EMS algorithm

required for setting up each cluster to be $C_s = 20$J and the amount of energy supplied to each sensor node for performing the sensing activities is $C_p = 3.5$J. Now using [55, 56], we consider the network subject to some random node failures such that 57 nodes suffer some type of failure due to outages, or hardware issues over a period of 1 year, hence the failure rate of the entire network is given as $\lambda = 0.057$, to show the performance of both the deterministic and probabilistic models. Figure 7, illustrates the total expected energy consumption for both deterministic and probabilistic models corresponding to the number of clusters. It is clear from the figure that total expected energy consumption increases when the number of cluster increases, but the probabilistic model provides better results than the deterministic model to meet dynamic requirements of the network.

7.2.2 Processing Cost

It is well-known that the events triggered at sensor nodes follow Poisson distribution. However, the data packets generated from densely populated WSNs obey non-Poisson distribution due to continuous acquisition of sensory data from the surrounding. These data are highly redundant, or correlated and hence are computationally intensive for the local engines. Therefore, the proposed LPM framework filters out the redundant information collected from these sensors to reduce workload on local engines by discretizing the data packets. As a result, data packets generated using the proposed LPM framework are free from the limiting effects of redundancy and follow Poisson distribution [23, 24]. This mechanism can be observed from the proposed high level architecture provided in Figs. 2 and 5. Here, we consider the cost for processing data packets in the system as $C_1 = 7$, and the packets waiting in the queue to be processed at cost $C_2 = 8$. The redundancy factor in this setting is considered as $\kappa = 16$. Considering the system to have a deterministic arrival rate

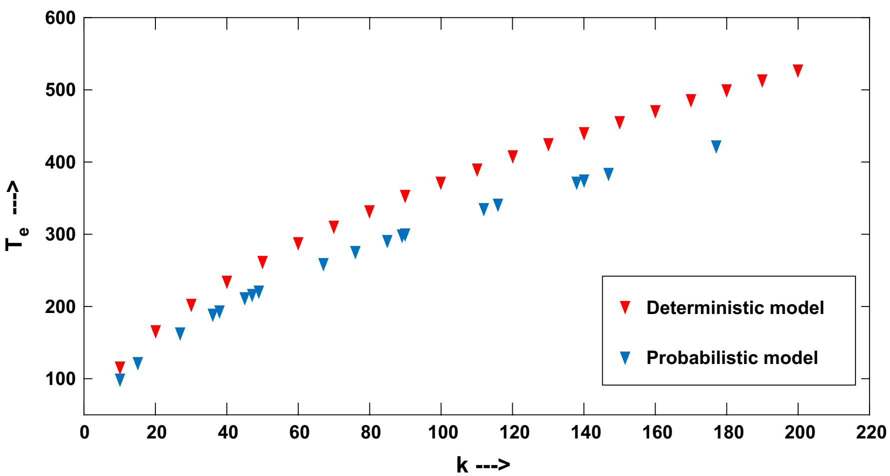

Fig. 7 Representation of the total expected energy consumption T_e for both deterministic and probabilistic models corresponding to k clusters with $C_s = 20$J, $C_p = 3.5$J, and $\lambda = 0.057$

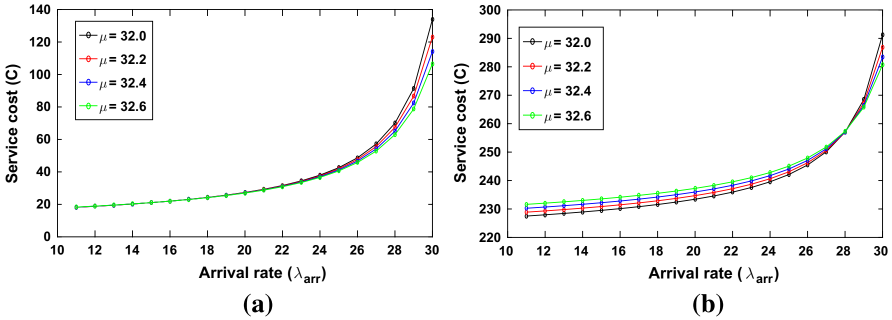

Fig. 8 The service costs for data packet transmission using LPM and without using LPM model with different values of μ. **a** Service cost for LPM corresponding to different values of μ i.e., $\mu = 32.0, 32.2, 32.4, 32.6$. **b** Service cost for the system without using LPM corresponding to different service rates i.e., $\mu = 32.0, 32.2, 32.4, 32.6$

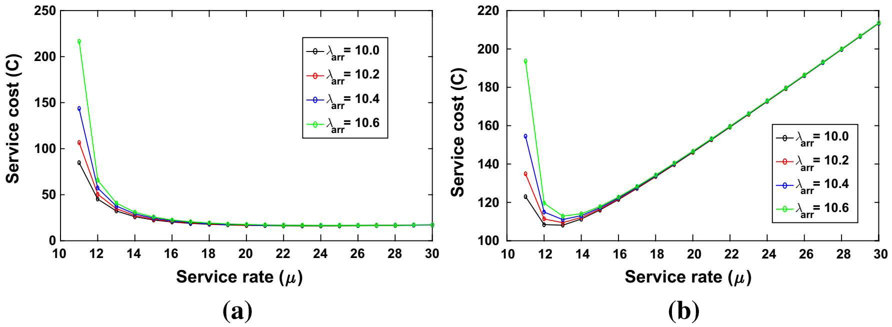

Fig. 9 The service cost for processing data packets using LPM and without LPM model with different arrival rates. **a** The service cost using LPM for different values of λ_{arr} i.e., $\lambda_{arr} = 10, 10.2, 10.4, 10.6$. **b** The service cost without LPM model for $\lambda_{arr} = 10.0, 10.2, 10.4, 10.6$

within the range $\mu = [11 : 30]$, the service cost of data packets for different service rates can be observed in Fig. 8. Figure 8(a) provides the service cost for data packets corresponding to different service rates viz., $\mu = 32.0, 32.2, 32.4, 32.6$ using the LPM framework. Figure 8(b) illustrates service cost for processing data packets without using LPM framework corresponding to different service rates i.e., for $\mu = 32.0, 32.2, 32.4, 32.6$.

The service costs for processing the data packets with a deterministic service rate i.e., for $\mu = [11 : 30]$, corresponding to different arrival rates $\lambda_{arr} = 10.0, 10.2, 10.4, 10.6$, is provided in Fig. 9. Here, Fig. 9(a) represents the service cost for processing the incoming data packets using proposed LPM framework. Figure 9(b) represents service cost for processing the data packets without using LPM model. It would be significant to observe that service cost for processing the data packets without using the LPM framework experiences an exponential growth. Hence, the proposed LPM framework provides a better choice for minimizing the processing costs associated with massive IoT datasets.

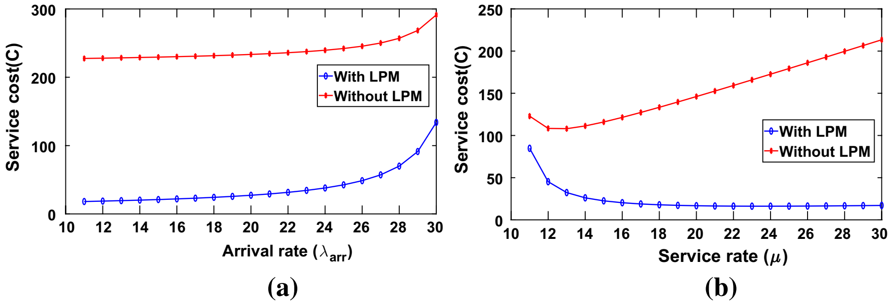

Fig. 10 **a** The service cost for processing the data packets using LPM and without using LPM models corresponding to parameters (i.e., $\lambda_{arr} = [11 : 30]$), $\kappa = 16$, $\mu = 32$, $c1 = 7$, and $c2 = 8$) and (**b**) Service cost of both models corresponding to parameters (i.e., $\mu = [11 : 30]$), $\kappa = 16$, $\lambda_{arr} = 10$, $c1 = 7$, and $c2 = 8$

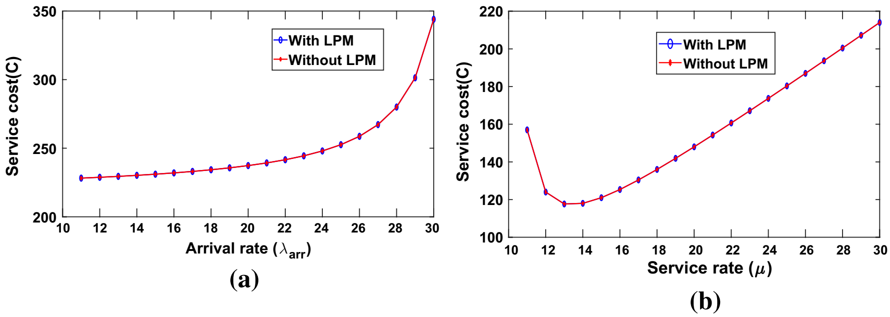

Fig. 11 For $\kappa = 1$, (**a**) the service cost of LPM and without LPM models corresponding to its arrival rates (i.e., $\lambda_{arr} = [11 : 30]$), $\mu = 32$, $c1 = 7$, and $c2 = 8$) and (**b**) the service cost of both models corresponding to its service rates (i.e., $\mu = [11 : 30]$), $\lambda_{arr} = 10$, $c1 = 7$, and $c2 = 8$

Figure 10 provides performance trade-offs between the proposed LPM model and the approach without using LPM model. The stable parameters considered in this case are as follows: $\kappa = 16$, $c1 = 7$, and $c2 = 8$. Figure 10(a) provides service costs for processing the data packets using LPM and without using LPM model for $\lambda_{arr} = [11 : 30]$ and service rate $\mu = 32$. Figure 10(b) provides the service costs corresponding to the LPM and without using LPM model for dynamic parameters $\mu = [11 : 30]$ and arrival rate $\lambda_{arr} = 10$.

Figure 11 provides the service cost estimation corresponding to LPM and without LPM approach for transmission of data packets considering unit value for the redundancy factor i.e., for $\kappa = 1$. In Fig. 11(a) the parameters considered are $\lambda_{arr} = [11 : 30]$, $\mu = 32$, $c1 = 7$, and $c2 = 8$. Further, Fig. 11(b) considers the parameters $\mu = [11 : 30]$, $\lambda_{arr} = 10$, $c1 = 7$, and $c2 = 8$, for obtaining the service costs. It would be striking to observe that the previous results considered a cluster size of $k = 16$, thus resulting in considerable performance trade-offs between the two models (as observed in Fig. 10(a) and (b)). However, when a single WSN cluster is responsible for generating the data packets i.e., $\kappa = 1$, the two

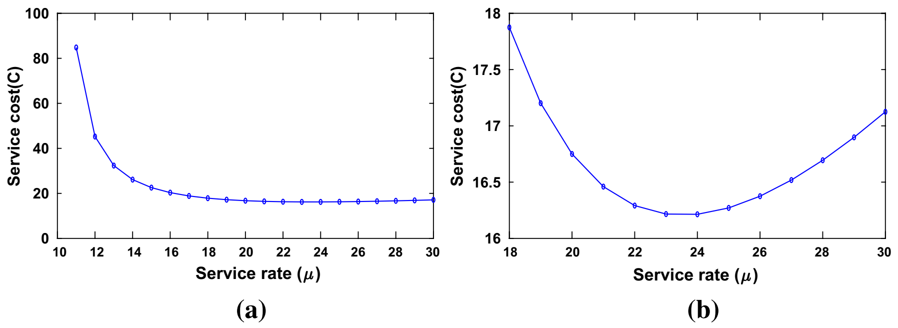

Fig. 12 (**a**) The service cost of LPM corresponding to parameters (i.e., $\mu = [11 : 30]$), $\kappa = 16$, $\lambda_{arr} = 10$, $c1 = 7$, and $c2 = 8$) and (**b**) Shows the windowed view of (**a**) when it attains optimal value at $\mu = 24$

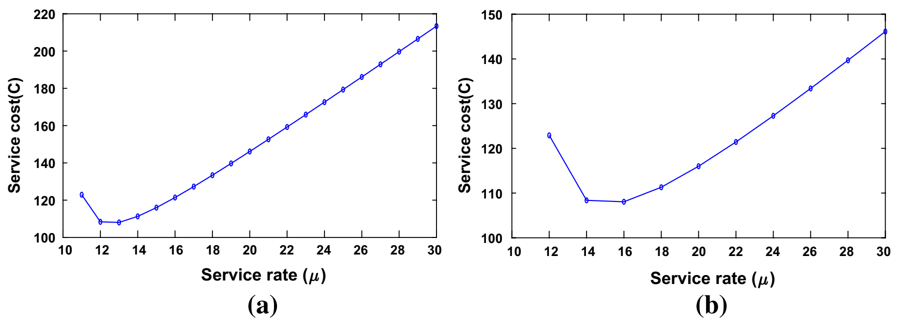

Fig. 13 **a** The service cost for without LPM approach corresponding to service rates (i.e., $\mu = [11 : 30]$), $k = 16$, $\lambda_{arr} = 10$, $c1 = 7$, and $c2 = 8$) and (**b**) Shows the windowed view where (**a**) attains its optimal value for $\mu = 13$

models (i.e., the LPM and without LPM models) converge hence indicating parallel performance.

In Fig. 12(a) the service costs for the proposed LPM model for different parameters viz., $\mu = [11 : 30]$, $k = 16$, $\lambda_{arr} = 10$, $c1 = 7$, and $c2 = 8$ are presented. Further, Fig. 12(b) represents a windowed view for optimal value obtained from Fig. 12(a). It is observed that the model provides an optimal value when service rate $\mu = 24$ resulting in a service cost of $C_{LPM} = 16.214$.

In Fig. 13(a) the service cost for processing the data packets without using LPM model has been provided. The parameters considered for obtaining the service cost corresponding to the without LPM model are as follows: $\mu = [11 : 30]$, $\kappa = 16$, $\lambda_{arr} = 10$, $c1 = 7$, and $c2 = 8$. Figure 13(b) provides the windowed view for the optimal value of Fig. 13(a) which is obtained at $\mu = 13$, and the corresponding service cost is $C = 108.051$. The service cost for this continues to grow after this point. Hence it is observed that the LPM model obtains much optimal service cost i.e., $C_{LPM} = 16.214$, for service rate $\mu = 24$. Table 2 provides a detailed summary for the service cost of LPM model obtained for different values of μ. In Table 3, the

Table 2 Service Cost for LPM Model

μ	Service cost	μ	Service cost	μ	Service cost	μ	Service cost
11	84.812	16	20.333	21	16.460	26	16.375
12	45.250	17	18.866	22	16.291	27	16.518
13	32.354	18	17.875	23	16.216	28	16.694
14	26.125	19	17.201	**24**	**16.214**	29	16.898
15	22.562	20	16.750	25	16.270	30	17.125

respective values for service cost of the model without using LPM is provided corresponding to different values of service rate μ. Here, the estimated optimal service costs for corresponding μ values have been represented in bold letters.

7.3 Model Validation

In this section, we discuss the results obtained in previous sections and validate with the real-time data obtained using IoT devices [27]. In smart buildings, the evaluation of air quality and thermal comfort levels has become a major concern. Several distinctive estimations (like the ASHRAE 55 standards) have been adopted to assess the air quality of residential or, industrial buildings [57, 58]. In our framework, we independently estimate the thermal comfort levels subject to three measures *viz.*, temperature (T), relative humidity (RH), and pressure (P_s). We consider a network equipped with three types of sensors namely, PM 1, PM 2.5, and PM 10, for sensing the temperature, relative humidity, and pressure respectively and validate our model in a smart building environment.

Table 4, represents the comfort levels (CL) and the respective thresholds corresponding to different air quality attributes. The attributes represent the temperature, relative humidity, and pressure along with their standard measurement units. The comfort level represents the thermal comfort zone (TCZ) of the attributes and their thresholds are defined for monitoring indoor environments. Thus, we assign certain threshold values to $T^{\ell}_{j,1}$ and $T^{\ell}_{j,2}$, which represent the minimum and maximum thresholds for activating the HVAC system. This paper uses the empirical data for 2 years (i.e., from 2016–2017) recorded in one minute

Table 3 Service Cost for without LPM

μ	Service cost	μ	Service cost	μ	Service cost	μ	Service cost
11	122.909	16	121.427	21	152.649	26	186.098
12	108.375	17	127.277	22	159.246	27	192.888
13	**108.051**	18	133.395	23	165.899	28	199.700
14	111.303	19	139.695	24	172.598	29	206.529
15	116.000	20	146.125	25	179.333	30	213.375

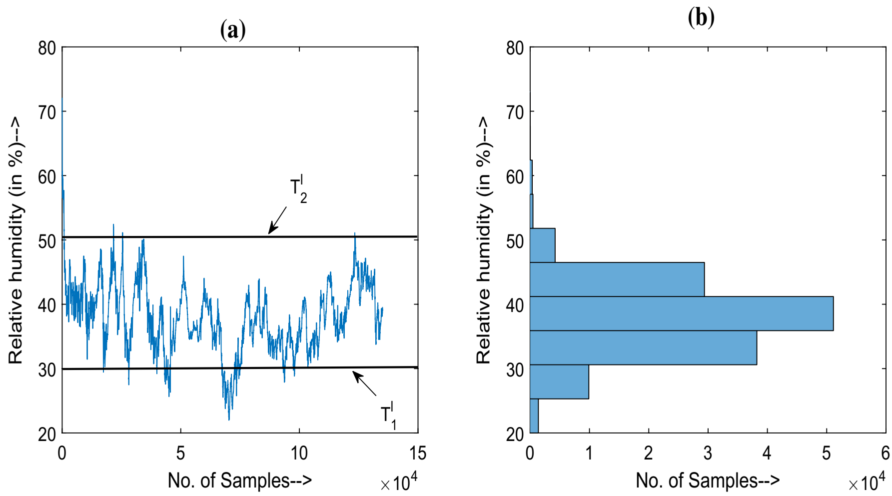

Fig. 14 (**a**) shows the minimum and maximum thresholds ($T^{\ell}_{j,1}$ & $T^{\ell}_{j,2}$) for humidity; and (**b**) shows the corresponding histogram representation for humidity

Table 4 The attributes for smart building's air quality monitoring

Attributes	Comfort level (CL)	$T^{\ell}_{j,1}$	$T^{\ell}_{j,2}$
T (in °C)	20–26	$20 \le x_{1,t} \le 26$	$x_{1,t} < 20$, or $x_{1,t} > 26$
RH (in %)	30–50	$30 \le x_{2,t} \le 50$	$x_{2,t} < 30$, or $x_{2,t} > 50$
P_s (in Pa)	10–50	$10 \le x_{3,t} \le 50$	$x_{3,t} < 10$, or $x_{3,t} > 50$

interval, corresponding to the relative humidity, pressure, and temperature respectively.Figure 14, represents the data collected by PM 2.5 sensors for measuring the relative humidity of a building. The HVAC systems can be activated by incorporating the threshold values along with the EMS algorithm to retain the TCZ. The values lying above the maximum threshold value $T^{\ell}_{j,2}$ represent the thermal discomfort zone (TDZ). For humidity exceeding these thresholds, the EMS algorithm recommends the HVAC systems of the building to get activated for preserving the comfort level of the building. Further, if humidity level lies below the minimum threshold $T^{\ell}_{j,1}$, then the air quality within the building becomes dry and EMS algorithm recommends the HVAC system to activate humidifiers.

Figure 15, represents the pressure level obtained by PM 10 sensors. The pressure levels lying above the maximum threshold value $T^{\ell}_{j,2}$ indicate extreme pressure levels in the building, while the pressure levels lying below $T^{\ell}_{j,1}$ indicate lower pressure levels. The EMS algorithm activates the HVAC system, if the comfort level of pressure exceeds its minimum and maximum threshold limits.

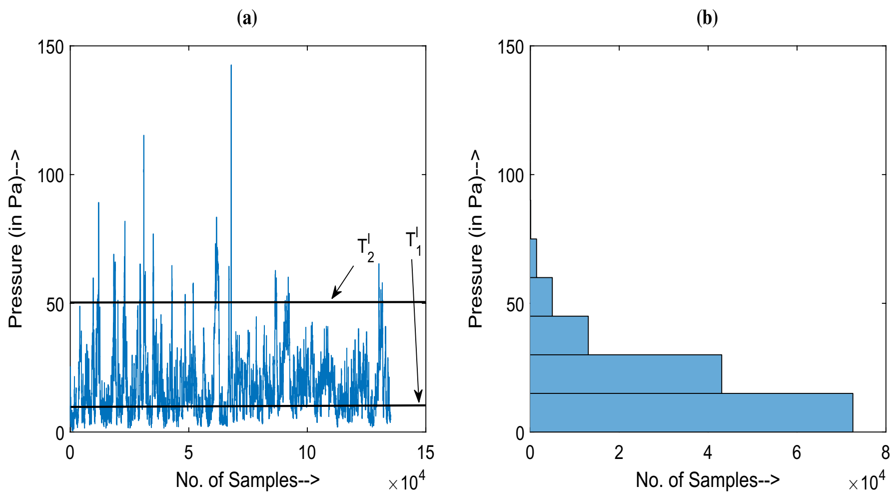

Fig. 15 **a** shows the minimum and maximum thresholds ($T^{\ell}_{j,1}$ & $T^{\ell}_{j,2}$) for pressure; and (**b**) shows the corresponding histogram representation for pressure

Similarly, the temperature of the indoor environment can be obtained by employing PM 1 sensors. Figure 16, shows the minimum and maximum threshold values for temperature. The HVAC system controls the indoor temperature, if the threshold value exceeds standard comfort level.

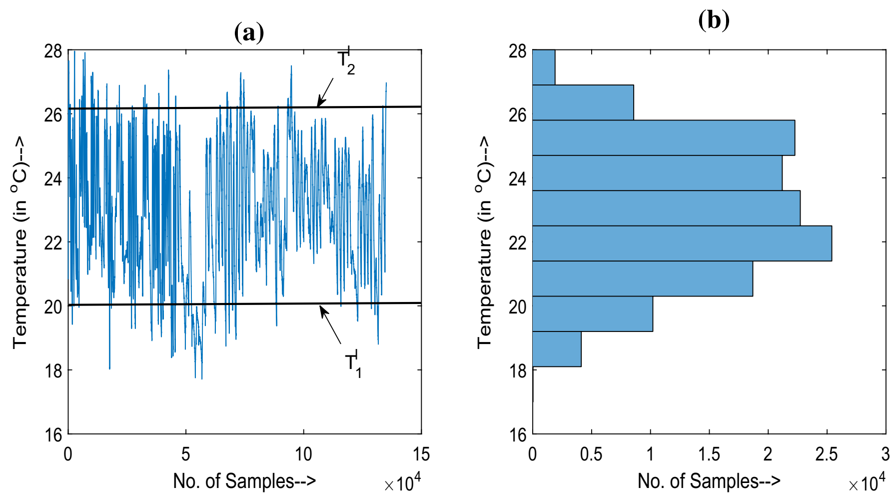

Fig. 16 (**a**) shows the minimum and maximum thresholds ($T^{\ell}_{j,1}$ & $T^{\ell}_{j,2}$) for temperature; and (**b**) shows the corresponding histogram representation for temperature

8 Conclusion and Future Work

In this paper, we presented a service oriented IoT architecture for adaptively monitoring the air quality of smart buildings using our expert Markov switching (EMS) model. This model has reduced the manifold redundancy at data acquisition level of the sensor nodes using the EMS algorithm. We have also shown the validation of our model in compliance with empirical data acquired from various sensors (viz., temperature, humidity, and pressure). In order to proclaim the validity of the proposed EMS algorithm, we have analysed its performance in convergence with four conventionally used algorithms for modelling large-scale dynamic sensory systems. It has been observed that the EMS algorithm outperforms the conventional algorithms greatly by achieving a time complexity of $O(N)$. The latency delay at the cloud end for real-time applications have also been reduced using the proposed local processing mechanism (LPM) in context of computational offloading based on the well-known queuing model. We have also proposed a probabilistic model for selecting active sensor nodes and the optimal sensor selection (OSS) model, which subsequently minimize the overall cost and power consumption corresponding to non-Poisson and Poisson arrival of data packets. The validity of each proposed model has been illustrated using the respective numerical results. Further, in the aforementioned context this paper has also addressed things-centric and data-centric approaches towards service oriented IoT architecture.

In future, we will propose a two-tier service architecture for IoT, which optimizes both storage and retrieval of information at cloud end satisfying the green computing constraints.

Acknowledgements The authors would like to thank OURIIP, Odisha, India, for extending their support to initiate the research and provide the adequate research environment. The authors are also thankful to the anonymous reviewers for providing their valuable comments which assisted in improving the overall quality of the paper.

References

1. Rao P.B.B., Saluia, Paval, S., Neetu, S., Ankit, M., Sharma S.V.: Cloud computing for internet of things & sensing based applications. In Sensing Technology (ICST), 2012 Sixth International Conference on, pp. 374–380. IEEE, New York (2012)
2. Atzori, L., Iera, A., Morabito, G.: From ”smart objects” to” social objects”: the next evolutionary step of the internet of things. IEEE Commun. Magazine **52**(1), 97–105 (2014)
3. Cook, D.J., Das, S.K.: How smart are our environments? an updated look at the state of the art. Pervasive Mobile Comput. **3**(2), 53–73 (2007)
4. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. Computer Networks **54**(15), 2787–2805 (2010)
5. Chen, S., Hui, X., Liu, D., Bo, H., Wang, H.: A vision of iot: applications, challenges, and opportunities with china perspective. IEEE Internet Things J **1**(4), 349–359 (2014)
6. Bebortta, S., Singh, A.K., Mohanty, S., Senapati, D.: Characterization of range for smart home sensors using tsallis entropy framework. In Advanced Computing and Intelligent Engineering, pp. 265–276. Springer, Berlin (2020)
7. Xu, L.D., He, W., Li, S.: Internet of things in industries: a survey. IEEE Trans. Ind. Inform. **10**(4), 2233–2243 (2014)

8. Rohokale, V.M., Prasad, N.R., Prasad, R.: A cooperative internet of things (iot) for rural healthcare monitoring and control. In Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on, pp. 1–6. IEEE, New York (2011)
9. Zhang, K., Ni, J., Yang, K., Xiaohui L., Ju, R., Shen, X.S.: Security and privacy in smart city applications: challenges and solutions. IEEE Commun. Magazine **55**(1), 122–129 (2017)
10. Bebortta, S., Panda, M., Panda, S.: Classification of pathological disorders in children using random forest algorithm. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), pp. 1–6. IEEE, New York (2020)
11. Fadi, A.T., David, D.B.: Seamless authentication: For iot-big data technologies in smart industrial application systems. IEEE Trans. Ind. Inform. (2020)
12. Sharma, Y., Javadi, B., Si, W., Sun, D.: Reliability and energy efficiency in cloud computing systems: survey and taxonomy. J Network Computer Appl. **74**, 66–85 (2016)
13. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Syst. **28**(5), 755–768 (2012)
14. Lu, R., Lin, X., Liang, X., Shen, X.-S.: Secure provenance: the essential of bread and butter of data forensics in cloud computing. In Proceedings of the 5th ACM symposium on information, computer and communications security, pp. 282–292. ACM, (2010)
15. Li, H., Yang, Y., Luan, T.H., Liang, X., Zhou, L., Shen, X.S.: Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data. IEEE Trans. Dependable Secure Comput. **13**(3), 312–325 (2016)
16. Hakiri, A., Berthou, P., Gokhale, A., Abdellatif, S.: Publish/subscribe-enabled software defined networking for efficient and scalable iot communications. IEEE Commun. Magazine **53**(9), 48–54 (2015)
17. Li, Fei, Vögler, Michael, Claeßens, Markus, Dustdar, Schahram: Efficient and scalable iot service delivery on cloud. In Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on, pp. 740–747. IEEE, New York (2013)
18. Zenia, N.Z., Aseeri, M., Ahmed, M.R., Chowdhury, Z.I., Kaiser, M.S.: Energy-efficiency and reliability in mac and routing protocols for underwater wireless sensor network: a survey. J Network Computer Appl. **71**, 72–85 (2016)
19. Ayub, Q, Rashid, M.S., Zahid, Abdullah, S.M., Hanan A.: Contact quality based forwarding strategy for delay tolerant network. J. Network Computer Appl. **39**, 302–309 (2014)
20. Li, Y., Bartos, R.: A survey of protocols for intermittently connected delay-tolerant wireless sensor networks. J. Network Computer Appl. **41**, 411–423 (2014)
21. Meiqin, Mao, Meihong, Ji, Wei, Dong, Chang, Liuchen: Multi-objective economic dispatch model for a microgrid considering reliability. In The 2nd International Symposium on Power Electronics for Distributed Generation Systems, pp. 993–998. IEEE, New York (2010)
22. Nezami, F.G., Heydar, M.: Energy-aware economic production quantity model with variable energy pricing. Operational Res. **19**(1), 201–218 (2019)
23. Itti, L., Baldi, P.: A principled approach to detecting surprising events in video. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol 1, pp 631–637. IEEE, New York (2005)
24. Baldi, P., Itti, L.: Of bits and wows: a bayesian theory of surprise with applications to attention. Neural Networks **23**(5), 649–666 (2010)
25. Bebortta, S., Senapati, D., Rajput, N.K., Singh, A.K., Rathi ,V. K. Pandey, H.M., Jaiswal, A.K., Qian, J., Tiwari, P.: Evidence of power-law behavior in cognitive iot applications. Neural Computing and Applications, pp. 1–13, New York (2020)
26. Tsay, R.S.: Analysis of financial time series, vol. 543. Wiley, Amsterdam (2005)
27. Liu, J.: Gams indoor air quality dataset, (2017). www.measureofquality.com
28. Shamaiah, M., Banerjee, S., Vikalo, H.: Greedy sensor selection: Leveraging submodularity. In 49th IEEE conference on decision and control (CDC), pp. 2572–2577. IEEE, New York (2010)
29. Dhingra, N.K., Jovanović, M.R., Luo, Z.Q.: An admm algorithm for optimal sensor and actuator selection. In 53rd IEEE Conference on Decision and Control, pp. 4039–4044. IEEE, New York (2014)
30. Zhang, H., Ayoub, R., Sundaram, S.: Sensor selection for optimal filtering of linear dynamical systems: Complexity and approximation. In 2015 54th IEEE Conference on Decision and Control (CDC), pp. 5002–5007. IEEE, New York (2015)

31. Zhang, H., Ayoub, R., Sundaram, S.: Sensor selection for kalman filtering of linear dynamical systems: complexity, limitations and greedy algorithms. Automatica **78**, 202–210 (2017)
32. Singh, A.K., et al.: Power law behavior of queue size: maximum entropy principle with shifted geometric mean constraint. IEEE Commun. Lett. **18**(8), 1335–1338 (2014)
33. Singh, A.K., Singh, H.P., et al.: Analysis of finite buffer queue: maximum entropy probability distribution with shifted fractional geometric and arithmetic means. IEEE Commun. Lett. **19**(2), 163–166 (2015)
34. Schor, L., Sommer, P., Wattenhofer, R.: Towards a zero-configuration wireless sensor network architecture for smart buildings. In Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, pp. 31–36. ACM (2009)
35. Soliman, M., Abiodun, T., Hamouda, T., Zhou, J., Lung, C.H.: Smart home: Integrating internet of things with web services and cloud computing. In 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 317–320. IEEE, New York (2013)
36. Fan, Z., Kulkarni, P., Gormus, S., Efthymiou, C., Kalogridis, G., Sooriyabandara, M., Zhu, Z., Lambotharan, S., Chin, W.H.: Smart grid communications: overview of research challenges, solutions, and standardization activities. IEEE Commun. Surveys Tutorials **15**(1), 21–38 (2013)
37. Sembroiz, D., Ricciardi, S., Careglio, D.: A novel cloud-based iot architecture for smart building automation. In Security and Resilience in Intelligent Data-Centric Systems and Communication Networks, pp. 215–233. Elsevier, Amsterdam(2018)
38. Chien, W.C., Lai, C.F., Cho, H.-H., Chao, H.C.: A sdn-sfc-based service-oriented load balancing for the iot applications. J. Network Computer Appl. **114**, 88–97 (2018)
39. Yassine, A., Singh, S., Hossain, M.S., Muhammad, G.: Iot big data analytics for smart homes with fog and cloud computing. Future Generation Computer Syst. **91**, 563–573 (2019)
40. Li, W., Kara, S.: Methodology for monitoring manufacturing environment by using wireless sensor networks (wsn) and the internet of things (iot). Procedia CIRP **61**, 323–328 (2017)
41. Carreira, P., Costa, A.A., Mansu, V., Arsénio, A.: Can hvac really learn from users? a simulation-based study on the effectiveness of voting for comfort and energy use optimisation. Sustainable Cities Soc. **41**, 275–285 (2018)
42. Abuarqoub, A., Hammoudeh, M., Adebisi, B., Jabbar, S., Bounceur, A., Al-Bashar, H.: Dynamic clustering and management of mobile wireless sensor networks. Computer Networks **117**, 62–75 (2017)
43. Noel, A.B., Abdaoui, A., Elfouly, T., Ahmed, M.H., Badawy, A., Shehata, M.S.: Structural health monitoring using wireless sensor networks: a comprehensive survey. IEEE Commun. Surveys Tutorials **19**(3), 1403–1423 (2017)
44. Estrin, D., Govindan, R., Heidemann, J., Kumar, S.: Next century challenges: Scalable coordination in sensor networks. In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp. 263–270. ACM (1999)
45. Liu, X.: A survey on clustering routing protocols in wireless sensor networks. Sensors **12**(8), 11113–11153 (2012)
46. Javed, A., Larijani, H., Ahmadinia, A., Emmanuel, R., Mannion, M., Gibson, D.: Design and implementation of a cloud enabled random neural network-based decentralized smart controller with intelligent sensor nodes for hvac. IEEE Internet Things J. **4**(2), 393–403 (2017)
47. Gupta, V., Gill, H.S., Singh, P., Kaur, R.: An energy efficient fog-cloud based architecture for healthcare. J. Stat. Manag. Syst. **21**(4), 529–537 (2018)
48. Salama, H.F., Reeves, D.S., Viniotis, Y.: Evaluation of multicast routing algorithms for real-time communication on high-speed networks. IEEE J. Selected Areas Commun. **15**(3), 332–345 (1997)
49. Walpole, R.E., Myers, S.L., Ye, K., Myers, R.H.: Probability Stat. Engin. Scientists. Pearson, London (2007)
50. Senapati, D., et al.: Generation of cubic power-law for high frequency intra-day returns: maximum tsallis entropy framework. Digital Signal Processing **48**, 276–284 (2016)
51. Mukherjee, T., Singh, A.K., Senapati, D.: Performance evaluation of wireless communication systems over weibull/q-lognormal shadowed fading using tsallis entropy framework. Wireless Personal Commun. **106**(2), 789–803 (2019)
52. Gradshteyn, I.S., Ryzhik, I.M.: Table of integrals, series, and products. Academic press, Cambridge (2014)
53. Gross, D.: Fundamentals of queueing theory. Wiley, New Jersey (2008)
54. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT press, Cambridge (2009)

55. Inst Tools. Control valve failure rate calculation. retrieved from, https://instrumentationtools.com/control-valve-failure-rate-calculation/
56. Tian, E., Yue, D.: Reliable h_∞ filter design for t-s fuzzy model-based networked control systems with random sensor failure. Int. J. Robust Nonlinear Control **23**(1), 15–32 (2013)
57. De Dear, R.J.: A global database of thermal comfort field experiments. ASHRAE Trans. **104**, 1141 (1998)
58. Fountain, M., Brager, G., de Dear, R.: Expectations of indoor climate control. Energy Buildings **24**(3), 179–182 (1996)

Sujit Bebortta Received his M.Sc. degree from Ravenshaw University, India and is currently pursuing his M. Tech. degree in Computer Science and Engineering from the College of Engineering and Technology, Bhubaneshwar, India. His current research interests include Optimal resource allocation, Ad-hoc networks, IoT, Blockchain, Machine Learning, and Randomized algorithm design.

Amit Kumar Singh Received his M.Tech. and Ph.D. degree from Jawaharlal Nehru University, New Delhi. He was a Postdoc fellow in National Tsing Hua University, Taiwan. He is currently an assistant professor with the department of Computer Science at Ramanujan College, India. His research interests include wireless networks, and queueing systems.

Bibudhendu Pati Is an Associate Professor and Head in the Department of Computer Science at Rama Devi Women's University, Bhubaneswar, India. His areas of research interests include Wireless Sensor Networks, Cloud Computing, Internet of Things, and cutting edge technology of Advanced Network Technologies. He completed his Ph.D. from IIT Kharagpur, India.

Dilip Senapati Is an Assistant Professor in the Department of Computer Science at Ravenshaw University, Odisha. He received his M.Tech. and Ph.D. degree in Computer Science and Technology from School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi. His research interests include wireless sensor networks, IoT, and entropy optimization.

Affiliations

Sujit Bebortta[1] · Amit Kumar Singh[2] · Bibudhendu Pati[3] · Dilip Senapati[1]

✉ Dilip Senapati
dilipsenapati.cs@ravenshawuniversity.ac.in

Sujit Bebortta
sujitbebortta.cs@ravenshawuniversity.ac.in

Amit Kumar Singh
aksingh@ramanujan.du.ac.in

Bibudhendu Pati
patibibudhendu@rdwu.ac.in

[1] Department of Computer Science, Ravenshaw University, Odisha 753003, India

[2] Department of Computer Science, Ramanujan College, University of Delhi, New Delhi 110019, India

[3] Department of Computer Science, Rama Devi Women's University, Bhubaneswar 751022, Odisha, India