Versatile and Generic Monitoring Solution Based on LPWAN Sensors Deployed on the Fly

Gwen Maudet, Mireille Batton-Hubert, Patrick Maillé, Laurent Toutain

Versatile and Generic Monitoring Solution Based on LPWAN Sensors Deployed on the Fly

Gwen Maudet[1], Mireille Batton-Hubert[2], Patrick Maillé[1], and Laurent Toutain[1]

[1]IMT Atlantique, IRISA, UMR CNRS 6074, F-35700, Rennes, France
[2]Mines Saint-Etienne, Univ Clermont Auvergne, UMR CNRS 6158, F-42023, Saint-Etienne, France

***Abstract*—Setting up IoT monitoring solutions requires a significant workforce for deployment and maintenance. This becomes particularly costly when a massive number of sensors are deployed.**

In this paper, we propose a generic solution based on energy-autonomous LPWAN sensors deployed on the fly. On one side, we propose an efficient management strategy that adapts the transmission period of sensors in order to receive a chosen amount of information on each sensor cluster that emits similar information. Moreover, we develop a method that generates real-time estimates based on the sensor message history.

After showing the relevance of our estimation method, we show that our sensor management method is efficient by comparing it to a baseline. Additionally, our tools are robust to the hazards occurring in the IoT: sensor arrivals and departures, noisy measuring in a highly variable environment, misplacement of sensors.

Overall, this paper proposes a way to scale up IoT, breaking the deployment cost barrier and paving the way for universal and versatile monitoring solutions.

I. Introduction

A. Motivations

The IoT has enabled the development of monitoring solutions on a large scale, now providing numerous monitoring methods (resource and flow optimization, risk management, tracking [1]) for different contexts (agriculture [2], industries [3], smart cities [4]). Current solutions are based on a rigid system, where each sensor deployment is calibrated for a specific need, making them efficient but costly to deploy and maintain.

Advances in electronics, along with the development of new high-constraint networks, have enabled the creation of low-cost embedded sensors to accomplish simple tasks: regular measurements of temperature, humidity, or CO2, for example [5,6]. These sensors are powered by batteries and are very inexpensive, allowing them to be deployed on a very large scale on the fly.

Our goal is to develop generic monitoring solutions with minimal human intervention regarding sensor configuration. Such solutions require a high degree of versatility in the management of unpredictable events that can come from the network, the sensors, or the environment. Since we are considering miniature battery-powered sensors, a key aspect is energy efficiency.

B. Existing Literature

Energy management is one main challenge spotted for IoT [7-10]. The aim is to efficiently use the energy of sensors to monitor an environment. This is all the more critical for the miniature, low-cost objects with limited battery capacity envisioned with massive IoT.

Initially, intelligence mechanisms for sensors are proposed to limit emissions, the primary source of battery consumption [11,12]. Messages are only sent if there is a behavior change or adaptively according to variations of the physical quantity studied. These approaches remain energy-consuming since the sensor is always awake to scan its environment.

Several other papers propose solutions to limit sensor emissions, relying on sensors that emit messages periodically and switch to deep sleep mode between each emission [13-16]. The proposed updated emission periods are based on the inter-sensor distance, the standard deviation of the returned values, and the battery level of the sensor. These mechanisms extend the network's life, lengthening the period of use of a sensor whose area is already covered or whose battery is low.

In 2006, the authors of [17] proposed a solution for the efficient management of sensor transmissions, allowing the sleeping of sensors whose measurement can be estimated by other sensors sending similar information. Their approach is composed of two separated phases: similarity analysis between sensors, and generation of sensor subsets so that each subset is capable of estimating the emissions of all the sensors. This work is very promising since it allows, in a dense deployment of sensors, to multiply by at least four the emission period of sensors while keeping the same tracking precision. However, no method is proposed to implement this emission scheduling in an optimal and dynamic way. Plus, the proposed approach considers that all sensors send measurements belonging to a limited set of discrete values, and that all messages are sent at the same time stamps.

In our first works [18,19], we proposed practical ways to dynamically schedule the sensor's emissions. These methods allow receiving a requested quantity of messages per unit of time, distributed between the sensors, by updating their emission period when they emit if needed. For instance, in [19] we propose a period update function that is resilient to the inclusion and departure of sensors, minimizing the number of period modifications to be performed when the sensor field is

modified.

The objective of this paper is to use similar principles for sensor similarity management as those proposed by [17] and to use the transmission management method defined in [19] to build an efficient monitoring solution that is versatile and robust to the hazards that can occur in the massive IoT.

C. Our Contributions

In this paper, we consider battery-powered sensors emitting over LPWANs that are densely deployed into an environment we want to monitor. We don't know the position of the sensors. Most of them follow interesting phenomena, others follow isolated phenomena that are not relevant. Based on these considerations, we want to manage the sensors in an energy-efficient way and provide real-time output estimates.

Our approach has two main components. First, we use splitting/merging mechanisms to dynamically cluster sensors emitting similar information. We then build a **transmission strategy** that adapts the sensor emission period to receive an overall constant amount of messages for each cluster. Second, we perform a hierarchical sensor clustering to propose an **online adaptive estimation** tool that represents the phenomena studied by the largest number of sensors without taking into account the messages of isolated sensors.

Simulations show that our online estimation tool is relevant, as it correctly groups sensors of the same phenomenon and attenuates the noise from imperfect measurements. Moreover, we show that our transmission strategy is energy-saving by comparing it to a classic scheduling strategy, with the drawback of a higher number of period changes. These proposals encompass noise measurement issues, handle that sensors can enter and exit the environment, and identify isolated sensors.

The rest of the paper is organized as follows. The hypotheses and objectives of the paper are exposed in Section II. Section III presents a generic estimation function as well as a distance between sensors that correspond to the characteristics of the sensor messages. The proposed **online adaptive estimation** method is then developed in Section IV, and the **transmission strategy** is presented in Section V. Section VII provides experimental and comparative results for our proposals. Finally, Section VIII concludes by proposing perspectives for future work.

II. Model Assumptions and Objectives

A. The Phenomena

A physical quantity is monitored in an environment, with IoT sensors deployed in massive numbers. For example, one could imagine temperature/CO2/humidity sensors included in false ceiling plates of a building. Another example would be biodegradable sensors dropped in a land to study soil moisture.

We consider that the environment is split into several distinct phenomena, where the physical quantity varies over time differently from one phenomenon to another. According our examples, there may be different phenomena in separate rooms, or on distant land areas. Since these sensors are deployed without knowledge of the variations of physical quantities, it is possible that some of them study localized phenomena (compared to global phenomena which would be more interesting to follow).

B. The Sensors

Sensors are deployed in large quantities on the fly: it is assumed that the phenomenon studied by a sensor is unique but not initially known. Real-world experiments from the literature have shown that sensors can be grouped by similarity of their measurements [17]. The measurements sent to the monitoring system are noisy because the sensors are considered to be cheap.

The fleet of sensors can change over time as a consequence of new node activations or sensors exiting the system due to battery discharge or electronic problems. A sensor consumes its battery over time for each message sent, transmitting a measurement just made. Between two emissions, the sensor goes to sleep mode to save energy.

C. The Network

Transmissions are modeled based on the LPWAN standards. Sensors send periodic messages; the transmission period can be redefined by the gateway only during a short listening window after each message sent by the sensor.

These considerations are the most constraining (Aloha networks), which allows adapting them, with some adjustments, to any other network: Bluetooth Low Energy, 5G, for example.

D. Levers of Action and Objectives

The objective of this paper is to define a **transmission strategy** through a period update function that manages the emissions of sensors efficiently in order to track phenomena present in the environment while limiting sensors' consumption.

From this emission stream, we want to provide an **online adaptive estimation** that tracks the most significant phenomena in the environment. One can consider that only the major phenomena should be shown: a user sets the number of estimates representing the phenomena studied by the most sensors.

A pertinent quantity of messages should be given from the transmission strategy to the online adaptive estimation. There is inertia between the actions performed by the transmission strategy (by changing the emission period of a sensor) and the data sent to the monitoring system. The objective for the transmission strategy is to anticipate any change in the behavior of the phenomena (e.g., two initially similar phenomena that become different). On the other hand, the objective of the estimation is to produce an output for the user that is an estimate of the phenomena present over the entire monitoring time. Although both tools we propose have the same goal (i.e., grouping sensors by similarities), we develop two separate methods, each respecting the specificities of their problem. In this way, it would be possible to use either tool without needing the other.

III. Definition of an Estimation Function and a Distance Between Time Series

We propose a noise attenuating estimation function that we use to build a distance between two sets of measurements. This distance will be used to identify similarities between sensor messages in the online adaptive estimation tool as well as in the transmission strategy.

A. An Estimation Function from Message History

We propose a continuous representation with noise attenuation of sensor's measurements. Considering a discrete time series I, each message $e \in I$ is composed of a time e_t and a value e_x. A discrete time series is composed of messages from one or multiple sensors. An *estimation function* E_I is a continuous function by parts from I, aiming to interpolate the measurements with noise attenuation.

faire un lien avec Ordered weighted averaging aggregation operator : papier On ordered weighted averaging aggregation operators in multi-criteria decision making

For this paper, we consider an estimation function defined by the average value of the previous messages, weighted by a freshness function $f(\delta t)$ which decreases according to the age of the data δt [20,21]. Mathematically, our estimation function at time t is:

$$E_I(t) = \frac{1}{\sum_{e \in I, e_t \leq t} f(t - e_t)} \sum_{e \in I, e_t \leq t} f(t - e_t) e_x \quad (1)$$

We could have chosen other methods to build an estimation of a noisy time series (auto-regression, Kalman filters, etc.) [22]. On the other hand, our solution does not rely on any assumption about the physical quantity or the shape of the measurement noise and is simpler to implement.

B. A Distance Between Discrete Time Series Based on the Estimation Function

The clustering tools that will be later developed require notions of distances between measurement sets. The considered discrete time series have some particularities.

- Variable sampling steps: the number of messages can vary in time and can be very different between two elements to compare.
- Different definition intervals: two elements may not be comparable when not defined on overlapping time ranges.
- Imperfect information: measurements are noisy.

For these reasons, the usual distances proposed in the literature for time series are not directly applicable here [23].

We propose a distance between discrete time series relying on the estimation function. We define the distance between two time series I, J as the average distance between their respective estimation function E_I, E_J on their joint definition interval; if they have no joint support, their distance is infinite. Mathematically, with $t_{\min} = \max(\min_{e \in I}(e_t), \min_{e \in J}(e_t))$ and $t_{\max} = \min(\max_{e \in I}(e_t), \max_{e \in J}(e_t))$, the joint definition duration $\delta(I, J)$ and the distance $D(I, J)$ we use can be expressed as:

$$\delta(I, J) = \max(0, t_{\max} - t_{\min})$$
$$D(I, J) = \begin{cases} +\infty & \text{if } \delta(I, J) = 0 \\ \frac{\int_{t=t_{\min}}^{t_{\max}} |E_I(t) - E_J(t)| dt}{\delta(I, J)} & \text{otherwise} \end{cases} \quad (2)$$

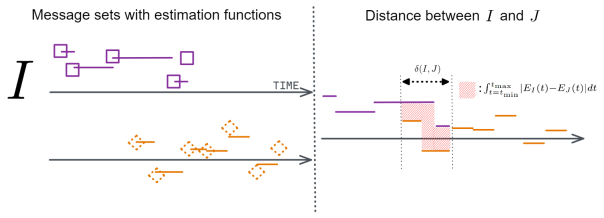


Fig. 1. Representation of the distance built from an estimation function

Abuse of language has been made here: $D(I, J)$ is not a distance since the triangular inequality property is not always respected. Still, this definition is sufficient for its future use.

IV. Online Adaptive Estimation Based on Hierarchical Sensor Clustering

In this section, we develop an **online adaptive estimation** method to display a given number of estimates F, aiming to represent the phenomena containing the most sensors. The goal here is to propose a generic method that takes as input message sets from sensors, and provide estimation of the main observed phenomena.

A global scheme is displayed in Fig. 2: according to sensors time series, we start by grouping sensors returning similar information into clusters, and keep the clusters containing the most elements to build the *estimation*.

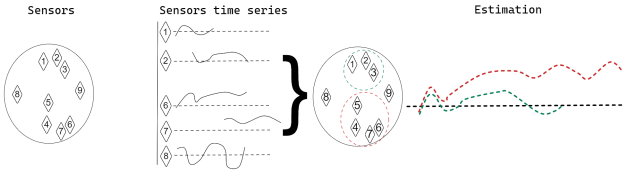


Fig. 2. Online adaptive estimation method scheme

A. Hierarchical Clustering for Sensors Discrete Time Series

We want to cluster sensors represented by their time series and keep the F clusters represented by the largest number of sensors. Time series clustering is a widely studied subject [24-26]. In our context, (i) clusters can have significantly different sizes, and (ii) their number is not known. Moreover, (iii) the distance as defined in Eq. (2) can be infinite.

We choose an ascendant hierarchical clustering since the method can handle these specificities [27]. Initially, each sensor defines a cluster; at each iteration, the two clusters whose inter-distance is minimal are grouped together; the

algorithm is stopped if F clusters remain or if the minimal inter-cluster distance exceeds a threshold d_{esti}.

We denote the distance between clusters i and j by $D_{i,j}$ and the computation duration by $\delta_{i,j}$. We initialize the distance between the clusters i and j from the time series I, J of the respective sensors, using Eq. (2): $\delta_{i,j} = \delta(I, J)$ and $D_{i,j} = D(I, J)$.

We compute the inter-cluster distance using the distances between sensors in order to avoid further distance calculations. We define it as the mean sensor distances weighted by their joint duration interval; this way, more importance is given to distances calculated over longer periods. Mathematically, considering each sensor time series $I \in i$ for cluster i and $J \in j$ for j, the distance between the clusters i and j can be expressed as:

$$D_{i,j} = \frac{\sum_{I \in i} \sum_{J \in j} \delta(I, J) D(I, J)}{\sum_{I \in i} \sum_{J \in j} \delta(I, J)}$$

If I and J have zero joint definition duration, we settle that $\delta(I, J)D(I, J) = 0$.

This clustering method can be implemented by a Lance-Williams algorithm, which is represented by a recursive formula for updating the distances between groups at each step. After merging the clusters i and j that have the closest distance, the distance and computation duration between the merged cluster $i + j$ and any other cluster k is:

$$D_{i+j,k} = \frac{\delta_{i,k}}{\delta_{i,k}+\delta_{j,k}} D_{i,k} + \frac{\delta_{j,k}}{\delta_{i,k}+\delta_{j,k}} D_{j,k}$$
$$\delta_{i+j,k} = \delta_{i,k} + \delta_{j,k}$$

As explained above, we stop merging clusters when there are F clusters (meaning that there are no identified isolated phenomena) or if the minimal inter-cluster distance is higher than a threshold parameter d_{esti}.

B. Online Estimation of the Major Phenomena

At the output of the sensor hierarchical clustering algorithm, we obtain at least F sensor clusters. We keep the F clusters containing the most sensors as the other sensor clusters are supposed to follow isolated phenomena.

For one sensor cluster, we consider one time series containing all the emissions of the sensors belonging to the cluster and display the output of the estimation function defined in Eq. (1). The *estimation* is the set of F continuous functions by part constructed from sensor measurements.

The estimation can regularly be updated as new messages are received. The more messages received, the more data to compare the sensors, and therefore the better we can group them to identify the major phenomena. Whenever an estimation is required, the entire process is repeated from the beginning.

V. Clustering 2-Level Round-Robin Transmission Strategy

In this section, we propose a **transmission strategy** to manage the sensor battery through a period update function that changes the emission period of a sensor that has just transmitted. We want to follow efficiently the message

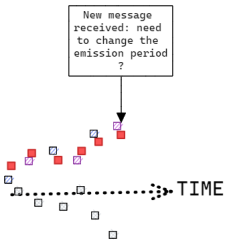


Fig. 3. Principle of the transmission strategy

To efficiently monitor existing phenomena in the environment, we propose a generic framework presented in Algorithm 1, where we successively use two methods:
(i) A *clustering method*, which updates the cluster-ID of a sensor that has just transmitted. The clustering method must dynamically group sensors emitting similar information by dynamically assigning them a similar cluster-ID.
(ii) A *period update function*, which, based on a new message and the updated cluster-ID, gives a period change order if necessary. The period update function must efficiently manage sensor emissions to track the phenomena energy effectively.

Algorithm 1 Transmission strategy framework

Require: new message
1: updated cluster-ID = *clustering method*(new message)
2: period = *period update function*(updated cluster-ID,new message)
3: **if** period is not None **then**
4: **Send period to sensor**
5: **end if**

The transmission strategy requires a new message and provides a new emission period if necessary. Each message is composed of the time, the value of measurement, and the ID of the sending sensor. From one method to another is transmitted the updated cluster-ID of the sending sensor, which is a compact, understandable and standard information format. Moreover, we choose to provide a framework that decorrelates the *clustering method* from the *period update function*. That way, one can propose any other clustering method or period update function that would better fit the needs or specific use cases.

A. Splitting/Merging-Based Clustering Method

The objective is to assign an identical cluster-ID to sensors that emit similar information; assignation is done in real-time when a sensor emits a new message. To group sensors with reasonable certainty, it is important to consider a relatively long observation duration. By contrast, if sensors that were initially grouped together start returning different measurements, it is better to consider the most recent messages to quickly identify that they do not belong to the same phenomenon. To meet these goals, we update the cluster-ID of a sensor

that has just transmitted by asking ourselves the following two questions: based on the most recent messages, is the sensor still well represented by its cluster? During a longer observation period, can this cluster be grouped with another cluster?

We thus propose three mechanisms applied successively:
(i) *Inclusion/exclusion*: include a new sensor or exclude a sensor that leaves the environment. When a new sensor enters the environment, a new cluster is created with it alone. When an expected message is not received, it means that the corresponding sensor is out of service; it is therefore removed from its reference cluster.
(ii) *Splitting*: determine whether it is still relevant that the emitting sensor is identified by this reference cluster. We compute the distance between the sensor measurements and the set of all measurements of sensors belonging to the reference cluster. If the distance exceeds a distance threshold d_s, the sensor is removed from the cluster, and a new cluster is created with it alone.
(iii) *Merging*: determine if the reference cluster can be grouped to another cluster. We calculate the distance between the measurements of all sensors belonging to the reference cluster and any other cluster. If a distance is smaller than a threshold d_m, we merge the reference cluster with the cluster containing the most sensors, by merging the smallest cluster into the largest.

In the *splitting* and *merging* mechanisms, we use the distance defined in Eq. (2), and compute distances by grouping sensor measurements from the same cluster in a single time series. We could have chosen the inter-cluster distance as the average distance between sensors as defined in Section IV-A, but this would have led to much higher computational costs.

We define the observation time windows T_s and T_m so that if a new sensor message is received at t, distances are computed only over messages posterior to $t - T_s$ for *splitting* and $t - T_m$ for *merging*. We choose T_s small in order to react quickly to remove a sensor from a cluster, and T_m bigger to group sensors together over a sufficiently long study time.

In the ideal case, the phenomena are initially sufficiently different: for sensors measuring the same phenomenon, the distance between a sensor and its reference cluster is at most d_s and between clusters of different phenomena at least d_m. In this case, sensors should be well-grouped thanks to *merging*. However, if during the initial instants or a duration T_m, different phenomena have similar physical quantity values, the sensors of these phenomena will be grouped (with *merging*). It will then be necessary to wait for these phenomena to be sufficiently different for a duration T_s so that the misrepresented sensors come out one by one (by *splitting*) and then be correctly grouped by *merging*. The *splitting* mechanism may also identify a sensor that starts returning outlier values by putting it outside the cluster.

B. 2LRR-Based Period Update Function

We propose a period update function ensuring that, for each cluster, a constant chosen amount of messages is received per time unit to maintain a chosen measurement precision while conserving energy.

We base our solution on the 2-Level Round-Robin period update function (2LRR) developed in [19]. The method allows receiving messages at chosen average time intervals τ distributed among all the present sensors. Concretely, the sensors are represented as leaf nodes of a perfectly balanced binary tree, and the period of a sensor is set to $2^d\tau$, where d is the depth of the sensor in the tree. When a sensor enters or leaves the system, dynamic period modification mechanisms allow keeping the constant time interval reception property while minimizing the number of period change modifications: two changes for inclusion of a new sensor (including the period definition of the incoming sensor), and one or two changes for the exclusion of a sensor. Fig. 4 illustrates the evolution of the tree as sensors enter and exit the system.

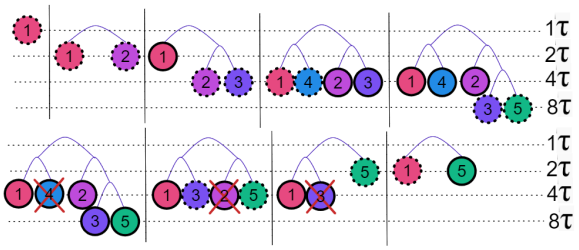


Fig. 4. Evolution of the binary tree representation as sensors enter *(top)* or leave *(bottom)* the system; each sensor is represented with a colored circle with an ID, and horizontal dotted lines represent the emission periods of the sensors at that depth. A dotted line around a sensor means that its position (and height) was changed in the tree (hence a period change order is needed). The top part represents the successive arrivals of sensors indexed from 1 to 5; the bottom one shows the successive departure of sensors 4, 2, and 3 (departures are symbolized by a cross) [19].

We extend this method by proposing a period update function that applies 2LRR for each cluster in order to receive messages at an average rate of τ on each cluster. We create a binary tree for each new cluster index returned by the *clustering method*. If the cluster-ID of a sensor is changed, we take it out of its binary tree to place it in the tree of its new label, by the inclusion and exclusion processes presented in detailed in [19] and briefly shown in Fig. 4.

Sensors removed from their reference cluster by the splitting mechanism are placed in a new binary tree. When merging two clusters, since only the updated cluster-ID is transferred, the period update function does not initially know the total number of sensors to move from one tree to another. Thus, the sensors that change their cluster-ID are included in the new tree one by one for their new emission. Since there are fewer sensors to add than sensors already in the tree, this results in the same period changes as including the sensors in the tree all at once and giving the period change orders when they emit.

In the end, if the depth of a sending sensor in a tree didn't change, none is sent. Otherwise, a period change order is sent, arising from a change in the clustering or the arrival/departure of a sensor.

VI. Comparison of the different clustering methods

Complexité algorithmique:

Cout de calcul de la distance :

Cout d'une estimation d'un ensemble de message de taille k: $O(k) = k$ si le temps entre message est superieur a fraishness

Cout du calcul de distance entre ensembles de messages avec k_1 messages pour $T1$ et k_2 message pour T_2 en commun: $O(k_1 + k_2)$.

On considère qu'on a effectué une durée de monitoring T, avec N capteurs. Chaque capteur a émit $k_i, i \in [1, N]$ messages. On note $K = \sum k_i$ le nombre total d'émissions des capteurs.

On considère le faceur entre le temps de monitoring et la durée de temps merging p_{T,T_m} et le facteur entre le temps de merg et le temps de split p_{T_m,T_s}. Sur ces fenetres de temps, on peut considerer que les capteurs on émit $p_{T,T_m} k_i$ pour merg et $p_{T,T_m} p_{T_m,T_s} k_i$.

Ascendant hierarchical clustering: distance entre tous les capteurs pairs a pair : on calcul a chaque fois la distance entre des messages de taille k_i, k_j. On a donc $O(\sum_{i,j}(k_i + k_j)) = O(K^2)$. On merge de facon itérative ensuite ce qui correspond a au plus N itérations en temps constant $O(N) = o(K^2)$.

Splitting merging: Merging, on compare un cluster avec tout autre cluster. Finalement, avec les notations prises, on obtient $O(p_{T,T_m} K)$.

VII. Simulations

In this section we study, through simulations, the performance of the solutions developed in the paper.

A. Simulation Setting

For all our simulations, we consider four temporal phenomena $(F_i)_{i\in[1,4]}$, whose physical quantity is simulated by a random walk of similar parameters. Each time step ΔT, the physical quantity can increase of an amplitude step ΔV with a probability 50%, or decrease of ΔV with the same probability.

Sensors enter the environment during a duration T following a Poisson process with an arrival rate of λ_i for F_i; this simulates the installation time of the IoT monitoring solution. In this simulation, we choose to include more sensors related to the phenomena F_1, F_4 than F_2, F_3 (i.e., $\lambda_1, \lambda_4 > \lambda_2, \lambda_3$). We fix the number of estimates to display to $F = 2$, implying that the phenomena F_2, F_3 should not appear because they must be considered isolated phenomena.

Sensors can get out of the environment for two main reasons. They consume their energy for each message sent until being out of battery; we assume that their initial energy follows an exponential law of mean $\frac{c_e}{\gamma}$ with c_e the energy consumed at each emission and γ the variability of the initial energy at the entry in the environment. Moreover, the maximum lifetime of a sensor follows an exponential law of parameter μ, representing an exceptional event (hardware problem, for example).

Sensor's measurements are noisy with Gaussian noise of standard deviation σ.

Simulation parameters are available in Table I; we use h as the time unit, although the hour is not the temporal reference to consider.

Parameter	Meaning	Value
	Physical quantity parameters of F_i	
ΔT	Time between two variations	$0.01h$
ΔV	Amplitude step	0.1
	Sensors field fluctuation and characteristics	
T	Duration of the sensor inclusion phase	$100h$
λ_1, λ_4	Sensor inclusion rate in F_1, F_4	$1h^{-1}$
λ_2, λ_3	Sensor inclusion rate in F_2, F_3	$0.1h^{-1}$
$\frac{1}{\mu}$	Maximum mean sensor life span	$10000h$
$\frac{1}{\gamma}$	Mean number of emissions per sensor	100
σ	Standard deviation of the Gaussian noise	1

TABLE I
Simulation parameters

B. Setting of the Estimation Function

We are looking for an estimation function that provides a good estimate of discrete time series with noise attenuation. For this, we choose the exponential freshness function family described as $f_l(\delta t) = e^{-\frac{\delta t}{l}}$ to use the estimation function defined in Eq. (1); we are looking for parameters l that provide an accurate estimation.

We consider several scenarios where the average number of received messages changes, which we model by Poisson laws of different rates. First, we perform one simulation for each message arrival rate with a new drawing of the physical quantity with the parameters defined in Table I, during a simulation duration of $100h$. Next, we choose different parameters l and look for the one that minimizes the mean squared error (MSE) between the estimation and the real phenomenon: Fig. 5 draws the MSE (on the y-axis) according to the choice of l (on the x-axis).

The estimation function we built strongly attenuates the noise: 0.077 of MSE is reached from highly noisy measurements ($\sigma = 1$) following a physical quantity of significant variability (jumps of 0.1 every $0.01h$ time step), with a message reception rate of 200.

Moreover, the choice of the optimal parameter l is conditioned by the arrival rate of the messages, varying from $l = 0.07$ for an arrival rate of 200 to $l = 0.9$ for an arrival rate of 1. Choosing the wrong l can have negative effects on the accuracy, especially for high arrival rates. According to this, for further simulations, the parameter l is chosen according to the *average message arrival rate* of the time series to estimate, which we define as: $\frac{\text{number of messages}}{\text{duration of the time series}}$. Thus, when constructing an estimation function (e.g. for distance calculations), we will refer to Table II for the choice of the appropriate freshness parameter l.

C. Setting of the Transmission Strategy and the Online Adaptive Estimation

We show that our monitoring solutions are suitable for many scenarios without rescaling by setting parameters that will remain for all the simulations.

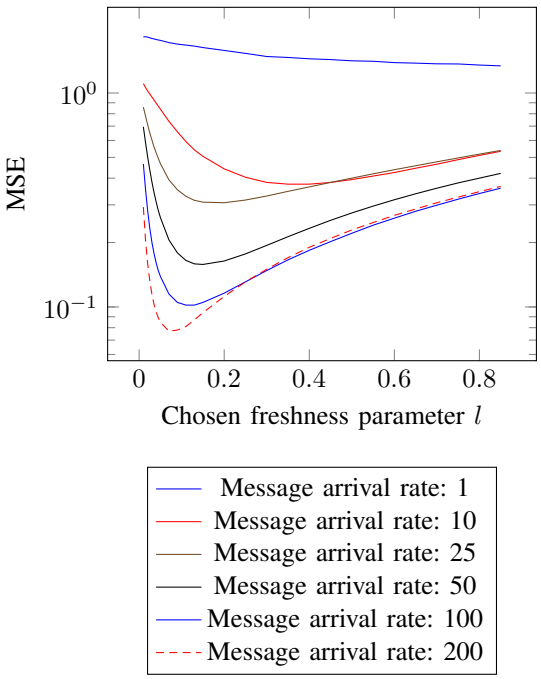


Fig. 5. MSE between the estimation function and the physical quantity as a function of the freshness parameter, for several message arrival rates.

average message arrival rate		l
lower bound	upper bound	
100		$0.07h^{-1}$
50	100	$0.11h^{-1}$
25	50	$0.15h^{-1}$
10	25	$0.25h^{-1}$
1	10	$0.35h^{-1}$
	1	$0.9h^{-1}$

TABLE II
CHOICE OF THE FRESHNESS PARAMETER l ACCORDING TO THE AVERAGE MESSAGE ARRIVAL RATE

First, in Section VII-E we will perform the *C2LRR* (Clustering 2-Level Round Robin) transmission strategy with small to large target message reception τ, which translates into more or less precise monitoring. Thus, the transmission strategy must handle time series with small and large message arrival rates. Moreover, the shapes of the time series are different according to the different mechanisms: for two time series to be compared during merging, their message arrival rates are both τ (in a non-changing context); using splitting, for a sensor in a cluster of n sensors, the ratio of arrival rates between the two time series to be compared is about $\frac{1}{n}$. According to these considerations, we fix d_s, d_m parameters to obtain globally good results; the parameters are summarized in Table III. We choose rather small analysis time windows T_s, T_m, mainly for computational time constraints.

In addition, for the online adaptive estimation, we set the threshold d_{esti} in Table III in order to identify phenomena with the same initial value (in Section VII-D) and to perform well when considering spaced initial values but with possible large time steps for the sensor emissions (in Section VII-E).

Parameter	**Meaning**	**Value**
	Splitting/merging mechanisms	
T_s	Max observation duration for splitting	$100h$
T_m	Max observation duration for merging	$200h$
d_s	minimum distance for splitting	10
d_m	maximum distance for merging	3
	Sensor hierarchical clustering	
d_{esti}	maximum inter-cluster distance	5

TABLE III
SETTINGS FOR THE MONITORING SOLUTIONS

D. An Illustrative Example

In this part of the simulation, we consider phenomena with the same initial value; the four phenomena are displayed in Fig. 6(a). We apply the *C2LRR* transmission strategy defined in Section V and we compare it to a baseline strategy. We did not find in the literature any method of sensor message management that are directly applicable in the context proposed here. Therefore, we define the transmission strategy *Static* that gives the same transmission period p to any new sensor entering the environment.

Considering the *estimation* defined in Section IV performed when all sensors are dead, we define the beginning and end of the monitoring as the minimum and maximum time when the two main phenomena are displayed. Our energy efficiency metric is the monitoring duration, defined by the duration between the beginning and end of the monitoring.

In this example, we parametrize C2LRR to a target period reception of $\tau = 0.1h$. Fig. 6(b) shows the emissions of sensors with their respective cluster-ID given by the splitting/merging-based clustering method. The bound of monitoring are the vertical lines in Fig. 6(b): we obtain a monitoring duration of $983h$. In order to compare the two strategies according to a similar monitoring duration, we set the individual period parameter of Static to $p = 2$ to obtain a monitoring duration of $986h$. Emissions of the sensors by the use of the Static method and the bound of monitoring are displayed in Fig. 6(c).

First, we see in Fig. 6(c) that the sensors from F_1, F_4 stay alive longer (almost until $1000h$) than those from F_2, F_3 (around $500h$) because a greater amount of sensors is in F_1, F_4. This is even more visible in Fig. 6(b) using C2LRR where F_2, F_3 are no more represented after $350h$.

Focusing on the C2LRR clustering performance, we see in Fig. 6(b) that at the beginning of the monitoring, the splitting/merging-based tool incorrectly groups the sensor since phenomena have similar values: the method groups all sensors belonging to F_1, F_2, F_4. Then, as the phenomena become more distinguishable, they are more easily identified. As a result, at $270h$, sensors related to the same phenomenon are grouped under a similar cluster-ID.

Conclusions on the efficient grouping of sensors can also be transposed in terms of message reception rates. In Fig. 6(d) is depicted the sum of the inverse of the period of sensors present in F_1, F_4 for Static and C2LRR, characterizing message arrival rates in these main phenomena. Our proposed period update

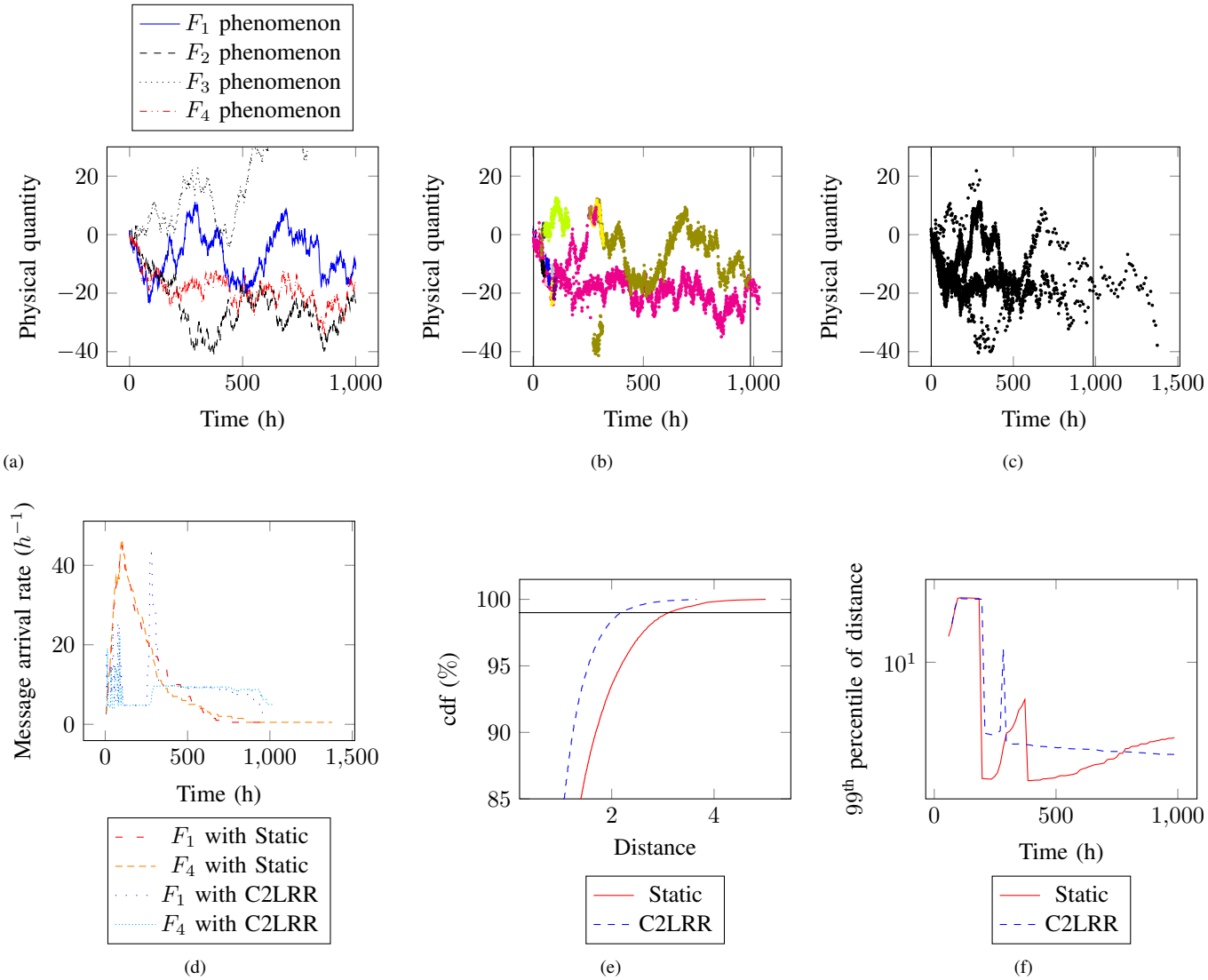


Fig. 6. (a) : physical quantities over time between 0 and 1000.
(b): all the emissions and their cluster-ID given by C2LRR with $\tau = 0.1h$, with bounds of monitoring in vertical lines.
(c): all the emissions of sensors by the use of the Static transmission strategy with $p = 2h$, with bounds of monitoring in vertical lines.
(d): message arrival rate of sensors belonging to F_1, F_4 phenomena over time while using either Static or C2LRR.
(e): last 15% of the cumulative distribution function (cdf) of the distance between real phenomena F_1, F_4 and the estimation performed at the end of the monitoring. The horizontal line draws the 99$^{\text{th}}$ percentile of distance.
(f): 99$^{\text{th}}$ percentile of the distance between real phenomena and the estimation over time.

function is tuned so that the sensors of a cluster send messages to receive a constant message arrival rate of $\frac{1}{\tau}$. Thus, C2LRR allows to receive on average 10 messages per time unit on the two main phenomena, starting from $350h$. For the Static transmission strategy, the number of messages per time unit starts at a high rate (46 messages per time unit in each phenomenon) and decrease steadily over time to zero.

Fig. 6(e) shows the last 15% of the cumulative distribution function of the distance between real phenomena and the estimation done at the end of the monitoring. The 99$^{\text{th}}$ percentile of distance (represented by a horizontal line) defines our precision metric: according to this figure, in 99% of the monitoring duration, the distance between the estimation and the real phenomena doesn't exceed 2.21 for C2LRR and 2.91 for Static.

The online adaptive estimation tool allows returning an estimation of the phenomena through time; the estimation changes as new messages are received. Fig. 6(f) shows the precision of the estimation along the time until the end of the monitoring: a value at t is the 99$^{\text{th}}$ percentile of the distance between the real phenomena and the estimation considering only messages prior to t. Before $400h$, for both Static and C2LRR, the estimation tool misidentifies the phenomena, which results in very high values of the 99$^{\text{th}}$ percentile of distance. After this time, as the reception rate is constant for C2LRR, the accuracy remains relatively stable until the

end of the monitoring. For Static, since there is no effective management method, many messages are received, and the 99^{th} percentile of distance drops. However, as the sensors die due to overuse, it increases to be overall less precise than C2LRR at the end of the monitoring.

E. Larger Scale Comparison of the Transmission Strategies

We perform here a more important simulation campaign. We want to evaluate our C2LRR transmission strategy to the baseline strategy Static, for different precision requirements. Here, the initial instants of each phenomenon are spaced: $F_1(0) = 0, F_2(0) = 50, F_3(0) = 100, F_4(0) = 150$. This way, no clustering error will occur for the estimation part, and the transmission strategy should perform well.

We run the simulation multiple times for different parameters: 50 simulations for each strategy, with reception rates of $\tau \in [0.05, 0.55]$ with steps of 0.01 for C2LRR and sensor emission period of $p \in [0.20, 6.2]$ with steps of 0.12 for Static. For each simulation, we build the estimation at the end of the monitoring (using Section IV), which we evaluate through the monitoring duration (energy efficiency metric) and the 99^{th} percentile of distance between the estimation and the phenomena (precision metric). Moreover, we count the number of period change orders.

In Fig. 7(a), each simulation is represented as a point with the 99^{th} percentile of distance on the x-axis and the monitoring duration on the y-axis. Overall, the C2LRR strategy provides more efficient solutions than Static, as the points representing its performance metrics in Fig. 7(a) are more at the top left of the figure. By performing linear regression on the performance of each strategy, we obtain the following linear equations: $y = 1168x - 988$ for C2LRR and $y = 945x - 1654$ for Static. Based on these regression results, for a given precision, there is at least an absolute monitoring duration difference of $1000h$ between the two strategies (comparison for $x > 1.5$). We can also say that we lengthen the monitoring duration by at least 64% using C2LRR compared to the baseline and the result is better for more precise requirements (comparison for $x < 4.5$).

Regarding the performance in terms of sensor period management, Fig. 7(b) displays the number of period change orders for the different simulations performed. Since Static assigns the emission period to sensors only once, it is the best method in terms of period changes. By contrast, because C2LRR includes sensors dynamically from the beginning of monitoring, groups them as new messages are received, and reacts to the death of each sensor, the number of period change orders is greater. As a result, there are, on average, 217 period changes for Static, and 1033 for C2LRR.

Qualitatively, choosing the optimal parameter of a transmission strategy is a key point for its implementation. However, it is impossible to fix the parameter p for the Static proposition if we don't know in advance the number of deployed sensors. Instead, although this result is subject to the efficiency of the clustering (here, our splitting/merging method), the τ parameter of C2LRR represents the average time between two message receptions on each identified phenomenon. This parameter can be requested by a user who wants to follow phenomena over time.

Overall, C2LRR efficiently manages sensor emissions by attempting to have the desired number of messages per time unit received on each phenomenon, as long as there are still active sensors; however, this has a significant management cost represented by the number of period change orders.

VIII. Conclusion

This paper proposes a new approach based on IoT battery-powered sensors deployed on the fly for the monitoring of multiple phenomena present in an environment. We developed a method that dynamically identifies phenomena present in the environment and efficiently manages sensor transmissions accordingly. In addition, we propose an estimation method that tracks the phenomena followed by most sensors over time. These solutions are resilient to various uncertainties considered in Massive IoT.

More globally, we show in this paper that low-cost wireless sensors can be deployed in any way and the monitoring system can automatically and energy-efficiently manage sensors while providing relevant and accurate estimation. This marks a new step towards an even larger scale deployment of monitoring solutions using IoT.

A lot of work remains to be done. Among other things, the next step is the development of a test bed to study the considerations taken in this paper: (i) can a real environment be divided into distinct phenomena? (ii) To what extent the No Ack on the uplink and downlink can degrade the proposed solution?

On the other hand, it is interesting to expand this solution by considering other parameters like uncertainties of the network and heterogeneity of the returned information. This knowledge about sensors is another milestone for a better understanding of the data and its integration into more efficient management policies.

References

[1] C. Tsai, C. Lai, M. Chiang, and L. Yang, "Data mining for internet of things: A survey," *IEEE COMMUNICATIONS SURVEYS and TUTORIALS*, 2014.

[2] S. Ullo and G. Sinha, "Advances in smart environment monitoring systems using iot and sensors," *Sensors*, vol. 20, no. 11, 2020.

[3] H. H. R. Sherazi, L. A. Grieco, M. A. Imran, and G. Boggia, "Energy-efficient lorawan for industry 4.0 applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 891–902, 2021.

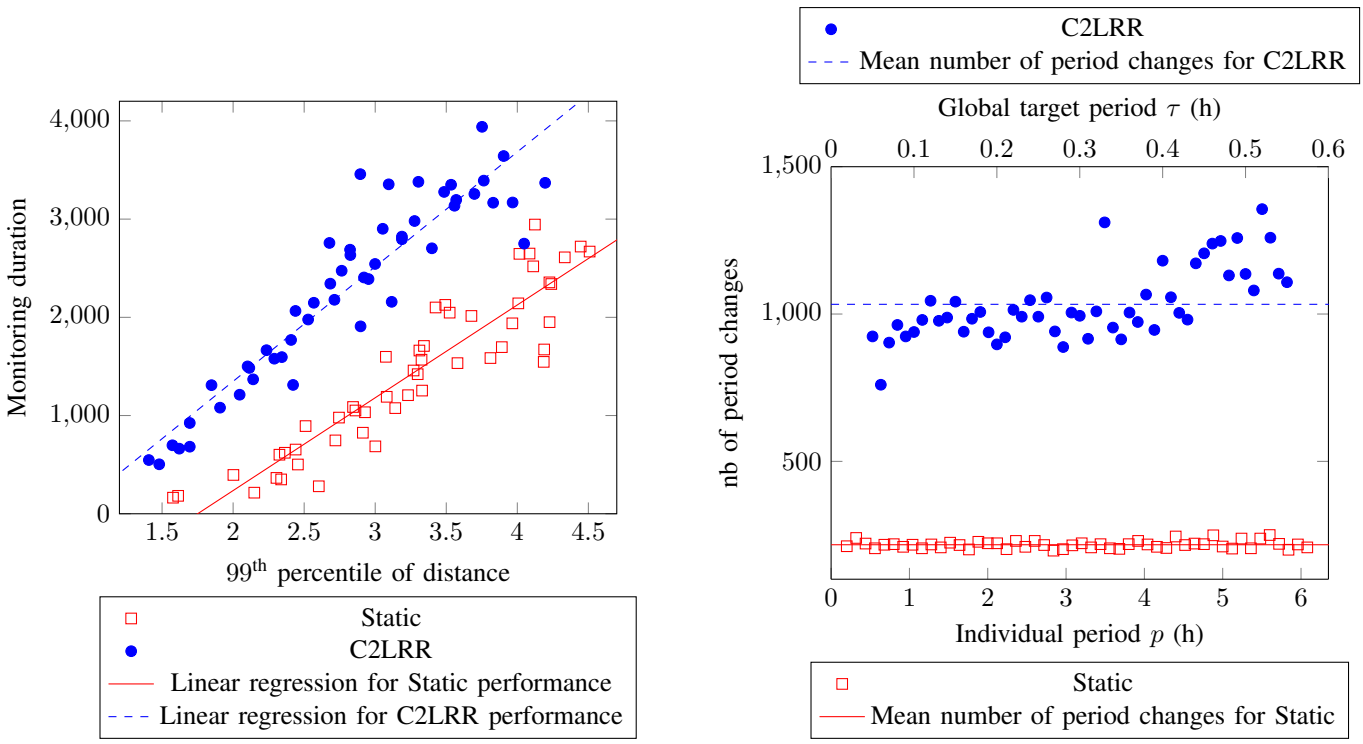
[4] Y. Liu and K. Yang, "Communication, sensing, computing and energy harvesting in smart cities," *IET Smart Cities*, pp. n/a–n/a, 09 2022.

[5] B. Chaudhari, M. Zennaro, and S. Borkar, "Lpwan technologies: Emerging application characteristics, requirements, and design considerations," *Future Internet*, vol. 12, p. 46, 03 2020.

[6] A. Boulogeorgos, P. Diamantoulakis, and G. Karagiannidis, "Low power wide area networks (lpwans) for internet of things (iot) applications: Research challenges and future trends," *CoRR*, vol. abs/1611.07449, 2016.

[7] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–31, 2005.

[8] G. Hurlburt, J. Voas, and K. Miller, "The internet of things: A reality check," *IT Professional*, vol. 14, pp. 56–59, 05 2012.



(a) (b)

Fig. 7. Performance evaluation of transmission strategies: 50 simulations for each strategy with $\tau \in [0.05, 0.55]$ for C2LRR and $p \in [0.20, 6.2]$ for Static
(a): 99$^{\text{th}}$ percentile of distance between real phenomena and the estimation performed at the end of the monitoring (in x-axis) and monitoring duration (in y-axis), with linear regressions.
(b): number of period change orders according to the parameter choice of the transmission strategy: τ for C2LRR on the top x-axis, and p for Static on the bottom. Mean number of period change orders for both transmission strategies.

[9] J. Gubbi, J. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[10] Y. Cheng, M. Lim, and K. Hui, "Impact of internet of things paradigm towards energy consumption prediction: A systematic literature review," *Sustainable Cities and Society*, vol. 78, p. 103624, 2022.

[11] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, "Energy management in wireless sensor networks with energy-hungry sensors," *IEEE Instrumentation Measurement Magazine*, vol. 12, no. 2, pp. 16–23, 2009.

[12] V. Raghunathan, S. Ganeriwal, and M. Srivastava, "Emerging techniques for long lived wireless sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 108–114, 2006.

[13] O. Akgul and B. Canberk, "Self-organized things (sot): An energy efficient next generation network management," *Computer Communications*, vol. 74, 07 2014.

[14] N. Kaur and S. Sood, "An energy-efficient architecture for the internet of things (IoT)," *IEEE Systems Journal*, vol. 11, no. 2, pp. 796–805, 2017.

[15] U. Gupta, Y. Tripathi, H. Bhardwaj, S. Goel, A. Kaur, and P. Kumar, "Energy-efficient model for deployment of sensor nodes in iot based system," in *International Conference on Contemporary Computing*, pp. 1–5, 2019.

[16] H. Vo, "Implementing energy saving techniques for sensor nodes in iot applications," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 5, p. 156076, 12 2018.

[17] F. Koushanfar, N. Taft, and M. Potkonjak, "Sleeping coordination for comprehensive sensing using isotonic regression and domatic partitions," in *IEEE International Conference on Computer Communications*, pp. 1–13, 2006.

[18] G. Maudet, M. Batton-Hubert, P. Maille, and L. Toutain, "Emission scheduling strategies for massive-iot: implementation and performance optimization," in *IEEE/IFIP Network Operations and Management Symposium*, 2022.

[19] G. Maudet, M. Batton-Hubert, P. Maille, and L. Toutain, "Energy Efficient Message Scheduling with Redundancy Control for Massive IoT Monitoring." working paper or preprint, Oct. 2022.

[20] M. Bouzeghoub and V. Peralta, "A framework for analysis of data freshness," in *International Workshop on Information Quality in Information Systems*, pp. 59–67, 06 2004.

[21] A. Even and G. Shankaranarayanan, "Utility-driven assessment of data quality," *SIGMIS Database*, vol. 38, p. 75–93, May 2007.

[22] J. Kostelich, "The analysis of chaotic time-series data," *Systems and Control Letters*, vol. 31, no. 5, pp. 313–319, 1997. Control of Chaos and Synchronization.

[23] M. Basseville, "Distance measures for signal processing and pattern recognition," *Signal Processing*, vol. 18, no. 4, pp. 349–369, 1989.

[24] V. Kavitha and M. Punithavalli, "Clustering time series data stream - a literature survey," 2010.

[25] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-series clustering – a decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[26] T. Warren Liao, "Clustering of time series data—a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[27] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies: Ii. clustering systems," *The Computer Journal*, vol. 10, pp. 271–277, 01 1967.