Grouping Similar Sensors Based on their Sent Data in a Massive IoT Scenario

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

Grouping Similar Sensors Based on their Sent Data in a Massive IoT Scenario

Gwen Maudet[1], Mireille Batton-Hubert[2], Patrick Maillé[3], and Laurent Toutain[3]

[1]SnT, University of Luxembourg, Esch-sur-Alzette, Luxembourg
[2]Mines Saint-Etienne, Univ Clermont Auvergne, UMR CNRS 6158, F-42023, Saint-Etienne, France
[3]IMT Atlantique, IRISA, UMR CNRS 6074, F-35700, Rennes, France

***Abstract*—The rapid expansion of the Internet of Things (IoT), in parallel with the substantial reduction in the cost of connected devices, have enabled the deployment of sensors in large scale. This massive deployment allows for more comprehensive coverage of the studied area when monitoring a physical quantity over time. As sensors become more densely packed, they often provide similar data due to their proximity.**

In this paper, we look to identify such similarities among sensors based on their returned data, in order to build groups of similar sensors. Groups of similar sensors can have several advantages, such as detecting sensor failures and performing reduction of sensors transmissions. Our primary focus is on a generic scenario that has received limited attention in the existing literature: sensors are deployed at different moments and exist in the environment for a limited duration, transmitting noisy and irregular data over time, without synchronization among them.

To address this, we define a data-driven similarity metric, which is then used for clustering similar sensors. According to the set of messages sent by a sensor, we apply an interpolation method that enable us to evaluate similarity between two sensors as the mean magnitude difference between the interpolations over their common definition interval. The duration of this common definition interval characterizes the trustworthiness of the distance. Hence, in the hierarchical clustering method we propose, we introduce a linkage method that assigns higher weights to distances calculated over longer comparison durations.

Through simulations, we demonstrate the superiority of our method compared to the state-of-the-art Dynamic Time Warping (DTW) distance and a hierarchical clustering with complete linkage inspired by related works. Our results establish a mean improvement of 23% of our approach in terms of V-Measure. Moreover, we provide comprehensive experiments assessing the robustness of our solution under various sensor measurement noise levels and employing different stopping criterion strategies.

I. Introduction

The Internet of Things (IoT) has revolutionized the landscape of large-scale monitoring solutions, offering a multitude of methods for various contexts, including resource and flow optimization, risk management, and tracking [1]. These solutions find applications in diverse domains such as agriculture [2], industries [3], and smart cities [4]. Advancements in electronics, coupled with the emergence of high-constraint networks, have led to the development of embedded sensors capable of performing simple specialized tasks, such as routine temperature, humidity, or CO2 measurements [5]. Powered by batteries and available at low cost, these sensors can be deployed on a large scale with ease. For instance, one can envision temperature sensors integrated into everyday objects [6].

Given the substantial number of deployed sensors, it is common for some of them to be in close proximity to each other, resulting in 'similar' sensors that transmit closely related data.

Exploiting this similarity serves at least two essential purposes. First, if a sensor malfunctions and sends aberrant data, we can promptly detect this failure by comparing its data with those from similar sensors, as studied in [7]. Second, if multiple sensors transmit similar data, there is an opportunity to reduce the volume of messages sent, as studied for instance in [8-10].

This paper aims to develop a method for identifying groups of similar sensors based on their data. The environment is considered to be composed of multiple phenomena, with each phenomenon exhibiting distinct variations in the physical quantity over time. Each sensor tracks one of these phenomena. The objective is to group together sensors that observe the same phenomenon.

In contrast to prior studies [8-10], we tackle a more generalized scenario where the transmission period of sensors is not constant. Furthermore, considering that synchronization of transmissions among sensors can be costly, especially in networks with a large number of nodes susceptible to clock drifts, we opt not to incorporate any synchronization among sensors.

Furthermore, since these sensors can be integrated into everyday objects, they may enter and exit the environment over time. Typically powered by batteries, these sensors are operational for a limited duration within the environment. This situation can give rise to scenarios where two sensors are present in the environment at different times.

Finally, due to the miniaturization of sensors, measurement errors are non-negligible.

The solution we propose can be divided into two main components: a similarity metric that quantifies the closeness between sensors and a hierarchical clustering method that aims to group similar sensors.

Firstly, for a set of data from a sensor, we define its interpolator using Kriging, a geostatistical technique. Subsequently, we establish the distance between two sensors on their

common definition interval, calculated as the difference in the average magnitude between their respective interpolations.

Subsequently, we present an Agglomerative Hierarchical Clustering (AHC) approach. The distance between two sensors can only be measured over their common definition interval, which may vary from being very short (providing limited relevance to the measurement) to very long (instilling more confidence in the measurement). Based on this observation, we define the inter-cluster distance as the mean of distances between sensors from different clusters, with distances being weighted by the common definition duration.

We introduce Dynamic Time Warping (DTW) as a comparative similarity metric and employ a clustering method based on the principles presented in [9,10]. We demonstrate the superiority of both our proposed similarity metric and clustering method compared to these alternatives.

Additionally, we assess the performance of our clustering method across various noise levels, comparing two stopping criteria for the AHC method.

The subsequent sections of the paper are outlined as follows: firstly, we present related works in Section II. Next, we specify the assumptions in Section III. Our similarity metric is introduced in Section IV, and the clustering method is elaborated in Section V. Simulations are then detailed in Section VI, followed by our conclusions in Section VII

II. Related works

Works such as [11] propose a method that evaluate similarities between sensors based on geography and validates these connections using similarity metrics derived from the data returned by the sensors. They employ data-based measures like the Jaccard coefficient, cosine similarity, and the Pearson product correlation coefficient.

[7] propose to address the fault detection problem through the identification of similarities between sensors. Correlation analysis is performed, allowing creation of possibly overlapping groups of correlated sensors. Thus, a fault is identified when a sensor deviates from an assigned cluster. The authors conduct experiments on an industrial process monitored by 17 sensors and achieve better fault detection rates compared to some traditional fault detection methods.

Other approaches, like [8-10], focus on reducing the volume of transmitted messages through scheduling strategies among sensors, assessing similarity based on the returned observations.

In the approach presented by [8], a transfer function is created to estimate one sensor's observation based on another sensor's measurement. A directed similarity link is established when the transfer function can accurately estimate a sensor's measurement from its own reading. In their experiment, 54 temperature and humidity sensors were deployed at distances ranging from 6 to 15 feet. They constructed up to 12 disjoint subsets of sensors, transmitting in round-robin fashion, each able to evaluate the data of all sensors. This approach achieved precision levels close to scenarios where all sensors transmitted at each round.

Other proposals focus on grouping sensors that return highly similar observations in order to reduce transmissions. Instead of activating all sensors in each round, sensors from the same cluster are activated in a round-robin fashion. These papers are most closely related to the problem studied in this paper.

In [9], observations are made simultaneously, and a link between two sensors is established if their observations do not exceed a certain difference threshold, and their trends (increase or decrease in the physical quantity) match for 95% of the time. Clusters are subsequently formed based on the graph representation of sensors, transforming the problem into one of clique partitioning. A greedy method is then employed to generate the clusters. An experiment was conducted with light sensors placed under desk lamps and barriers positioned in certain areas. This method successfully grouped sensors surrounded by barriers, resulting in an average reduction in sensor consumption by a factor of 3 with low precision loss. The authors also present an extended simulation on a larger scale with similarly promising results.

In [10], similar principles are applied, considering information transmission in a mesh network. A similarity link is established between two sensors if their Pearson correlation coefficient surpasses a threshold and the difference in averages exceeds another threshold. The clustering problem, in the form of cliques, resembles that in [9], with then the adoption of a greedy method for the selection of head clusters at the core of the clusters.

However, the novel assumptions introduced by the emergence of the Massive IoT paradigm [6] constrain the applicability of these methods in this new dynamic context.

In this paper, our goal is to evaluate similarity based on sensor observations. The proposed approaches assume a strict assumption that sensors synchronize their transmissions at regular intervals. We propose to investigate a more generic scenario, not relying on these assumptions of periodic and synchronized transmissions.

Furthermore, the existing clustering solutions presented assume that all sensors are initially present and are grouped based on observations sent during an initialization period. In the context of Massive IoT, characterized by a vast number of sensors, it is more realistic to acknowledge that sensors may participate in the monitoring process for only a limited duration, with new sensors joining over time. In this paper, we attempt to consider this new hypothesis.

III. Hypotheses and Objectives

In this section, we outline the objective and the hypotheses concerning the deployment of sensors and their observations.

A. Identifying Sensors Belonging to the Same Phenomenon

We consider an environment that exhibits multiple distinct phenomena, each demonstrating proper variations in the studied physical quantity over time. For instance, in a building, temperature variations may differ from one room to another. Sensors are deployed in this environment, each tracking one phenomenon.

Our goal is to cluster sensors that observe the same phenomenon, creating disjoint clusters where each sensor belongs to one and only one cluster.

This grouping of similar sensors addresses two well-known challenges in IoT networks. Firstly, with such a cluster structure, we can implement energy-saving mechanisms among sensors. Sensors belonging to the same similar cluster send redundant messages, so it is not essential for all sensors to consume their energy for message transmission. In previous studies [12,13], we presented methods tailored to highly constrained networks that distribute the transmission workload among a cluster of similar sensors. Secondly, it is crucial to assess the failure of an object, especially considering miniature embedded objects. Having groups of similar sensors provides a reliable reference for measurements, enabling the use of robust anomaly detection techniques.

B. Incoming and Outgoing Sensors

In the context of a large-scale IoT deployment involving these resource-constrained devices, it is conceivable that sensors could be integrated into everyday objects. This integration allows them to move in and out of the environment over time. Furthermore, due to the limitations imposed by their batteries, some sensors may become inactive either due to hardware problems or exhausted battery power.

Thus, sensors are defined only for a limited duration. As a result, the similarity between two sensors can only be evaluated when they are coexisting within the environment. Notably, this shared time interval of operation is variable or even non-existent.

C. Observations Sent by a Sensor

Sensors transmit observations over time to the terminal. An observation is defined by a time and a value. The observation value represents the value of the phenomenon that the sensor is following, with added noise due to imprecise measuring devices.

As indicated in [14], scheduling sensors to transmit at the same instants is challenging. Sensors are sensitive to clock drift, necessitating regular synchronization. Given the potential vastness of sensor fleets and the limited communication capacity with these constrained devices, achieving regular synchronization is expensive in terms of sensor consumption. Therefore, we operate under the assumption that sensors are not synchronized.

Furthermore, we assume that sensors send observations irregularly. The specific data collection method employed by a sensor (e.g., trigger-based, model-based [15]) influences the observation period, which tends to vary over time.

IV. Similarity Metric: Mean Difference between Interpolations of Sensors Observations

As part of our assumptions, we consider that a sensor sends unsynchronized observations to other sensors with a variable transmission period. Furthermore, this sensor remains within the environment for a limited duration. An example of the observations sent by two sensors, which we aim to compare, is illustrated in Fig. 1.

In this section, we introduce a distance metric that relies on two key components. Firstly, we utilize an interpolation method named Kriging to convert irregular observations into a continuous representation. Subsequently, we define the distance between two sensors over their common time interval as the mean magnitude difference between their interpolations.

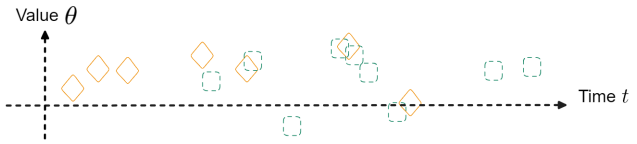

Fig. 1. Representation of two sets of observations. Orange diamonds and dashed green squares represent observations from two sensors, with time on the x-axis and observation values on the y-axis.

A. Kriging-Based Interpolation of an Observation Set

The observations are sent irregularly spaced and noisy, making direct comparisons challenging. Therefore, as an initial step, we propose to employ an interpolation method to transforms a set of observations into a continuous function, facilitating comparisons.

1) Justification of the Kriging Choice: An interpolation function is a mathematical function defined over all time points based on a set of noisy observations. Its objective is to minimize the average discrepancy between the interpolated function and the measured phenomenon. Numerous interpolation methods exist, as documented in [16]. Since the observed data is subject to noise, we aim to relax the constraint of passing through all data points. Consequently, certain methods like Spline are not applicable.

Kriging is an interpolation method based on Gaussian processes governed by prior covariances [17]. This approach is particularly well-suited for various noise reduction applications, as summarized in [18], as it allows the estimation and incorporation of measurement errors into the modeling. For instance, in [19], an experimental study demonstrated the superiority of Kriging over the inverse distance weighting method. Kriging has been applied in the domain of the IoT as well, such as in [20], where it was used to propose a sensor positioning solution based on the data they provide.

2) Principle of the Kriging and the Variogram: Kriging is an interpolation method based on Gaussian processes, where each observation is treated as a random variable. Thus, the *variogram* is a function that measures the variance between two observation values based on their temporal separation. It is employed in the Kriging model to estimate an interpolated value at a target time from known observations that are correlated (temporally close).

Since the true variogram is typically unknown, it is estimated using known observations. This estimation is obtained by initially calculating the experimental variogram. We denote by $\theta = \{\theta_t, t \in T\}$ the set of known observations, where

T represents the set of measurement time instants and θ_t is an observation value made at time t. Then, the experimental variogram γ_θ is computed for each pair of points, so that:

$$\forall (t_1, t_2) \in T^2, \gamma_\theta(|t_1 - t_2|) = 0.5(\theta_{t_1} - \theta_{t_2})^2$$

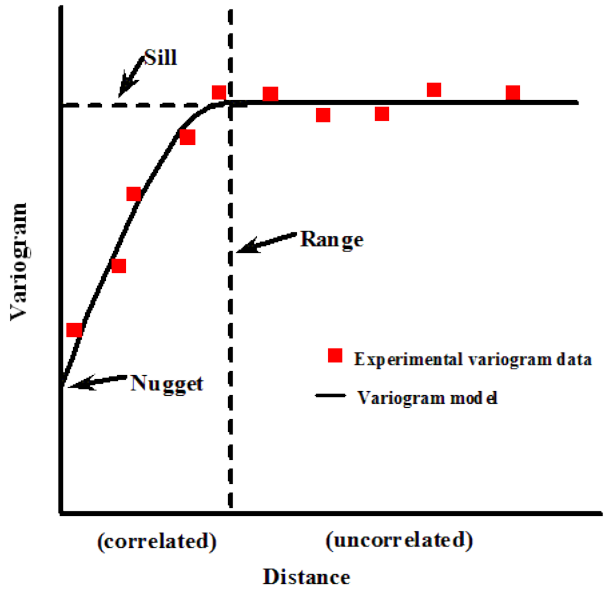

Fig. 2. Illustration of the variogram model based on experimental variogram points. The variogram consists of three parameters: nugget, sill, and range. The closer (temporally) the distance between two observations, the more correlated the values. Beyond a certain threshold, defined by the range, observations that are too distant are no longer correlated.

The data points of the experimental variogram are shown in Fig. 2 as red squares. Here, the horizontal axis represents the temporal distance between two observations, while the vertical axis displays the measurement of the experimental variogram between these two observations.

To create a continuous representation from this discrete experimental variogram, we fit these data points to a mathematical function known as the *variogram model*, denoted as $\hat{\gamma}_\theta$, and visualized in Fig. 2 by the black curve. This model serves to evaluate the correlation between two observations based on their temporal separation.

For example, spherical, exponential, and Gaussian models are characterized by three parameters and illustrated in Fig. 2:

- The nugget n: Signifies the variogram value when there is zero temporal distance between observations. It quantifies the amount of short-range variability in the data, essentially capturing measurement noise.
- The sill s: Represents the variogram value when the temporal distance becomes extensive enough that observation values are no longer correlated.
- The range r: Denotes the temporal distance at which the variogram reaches the sill value.

The generic version of the Gaussian variogram is for example given by:

$$\hat{\gamma}(t_1, t_2) = n + s\left(1 - e^{-\frac{(t_1 - t_2)^2}{r^2}}\right) \quad (1)$$

3) Calculations for the Simple Kriging: Kriging is an interpolation method rooted in statistical modeling. It assumes that each observation is a random variable with a finite mean and variance.

We present the result for the simple Kriging. The strong assumption here is that the mean expectation of values at all time instances is the same and known, assumed to be zero. In the case of ordinary Kriging (another Kriging modeling), the expectation is similar across all points and unknown; for universal Kriging, a polynomial trend model is incorporated.

Here, $\theta = (\theta_t)_{t \in T}$ constitutes the vector representing the set of known observations. Under the given assumptions, we assume $E[\theta_t] = 0$. The covariance matrix of the observation history vector is defined using the variogram model $\hat{\gamma}_\theta$ as follows: $K = E[\theta\theta^\top] = (\hat{\gamma}_\theta(t_1, t_2))_{t_1, t_2 \in T}$.

Our objective is to evaluate the value at the point $\hat{t}$. Let $\Theta_{\hat{t}}$ denote the random variable representing the value at $\hat{t}$ (with $E[\Theta_{\hat{t}}] = 0$). The covariance vector between the observation value to evaluate at $\hat{t}$ and the set of known observations is defined based on the variogram model: $k_{\hat{t}} = E[\theta\Theta_{\hat{t}}] = (\hat{\gamma}_\theta(\hat{t}, t))_{t \in T}$.

The core principle of Kriging is that interpolation at a point is defined as a linear combination of the observation values. Hence, the estimator at the point $\hat{t}$, denoted by $\hat{\theta}_{\hat{t}}$, is the sum of observation values weighted by the coefficient vector $\psi_{\hat{t}} = (\psi_{t,\hat{t}})_{t \in T}$:

$$\hat{\theta}_{\hat{t}} = \sum_{t \in T} \psi_{t,\hat{t}} \theta_t = \psi_{\hat{t}}^\top \theta$$

From the definition of $\hat{\theta}_{\hat{t}}$, we can already establish through its expectation calculation that it is unbiased: $E[\hat{\theta}_{\hat{t}}] = \sum_{t \in T} \psi_{t,\hat{t}} E[\Theta_t] = 0$.

The weights are defined to minimize the expectation of the squared difference between the estimator and the quantity to predict at this new point $\hat{t}$: $\Delta(\hat{t}) = E[(\hat{\theta}_{\hat{t}} - \Theta_{\hat{t}})^2]$

By expanding this squared difference, we have:

$$\begin{aligned}\Delta(\hat{t}) &= E[(\psi_{\hat{t}}^\top \theta - \Theta_{\hat{t}})^2] \\ &= E[\psi_{\hat{t}}^\top \theta\theta^\top \psi_{\hat{t}} - \Theta_{\hat{t}}\theta^\top \psi_{\hat{t}} - \psi_{\hat{t}}^\top \theta\Theta_{\hat{t}} + \Theta_{\hat{t}}^2] \\ &= \psi_{\hat{t}}^\top E[\theta\theta^\top]\psi_{\hat{t}} - 2E[\Theta_{\hat{t}}\theta^\top]\psi_{\hat{t}} + E[\Theta_{\hat{t}}^2] \\ &= \psi_{\hat{t}}^\top K \psi_{\hat{t}} - 2k_{\hat{t}}^\top \psi_{\hat{t}} + \sigma_{\hat{t}}^2\end{aligned}$$

Where $\sigma_{\hat{t}} = \sqrt{E[\Theta_{\hat{t}}^2]}$, independent of $\psi_{\hat{t}}$.

We aim to find the vector $\psi_{\hat{t}}$ that minimizes $\Delta(\hat{t})$. The derivative with respect to each $\psi_{t,\hat{t}}$ is zero, resulting in:

$$\begin{aligned}\frac{\partial \Delta(\hat{t})}{\partial \psi_{\hat{t}}} &= 2K\psi_{\hat{t}} - 2k_{\hat{t}} = 0 \\ \Leftrightarrow \quad \psi_{\hat{t}} &= K^{-1}k_{\hat{t}}\end{aligned}$$

K is a symmetric matrix, so K^{-1} is a symmetric matrix, leading us to the expression of the estimation $\hat{\theta}_{\hat{t}}$:

$$\hat{\theta}_{\hat{t}} = k_{\hat{t}}^\top K^{-1} \theta$$

Therefore, for the computation of $\hat{t}$, it is necessary to define K and $k_{\hat{t}}$ based on the variogram model γ_θ and invert the matrix K. For any new estimate of observation, it is only necessary to redefine k.

B. Distance Based on Mean Magnitude Difference

Let the sets of observations from sensors i and j defined by $i : \{\theta_{i,t}, t \in T_i\}$ and $j : \{\theta_{j,t}, t \in T_j\}$, so that $\hat{\theta}_i(t)$ and $\hat{\theta}_j(t)$ be the interpolations obtained using the Kriging-based interpolation. We use the mean magnitude difference to evaluate the distance between two interpolations over their common definition interval, as schematically represented in Fig. 3.

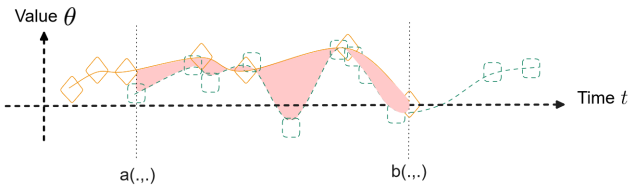

Fig. 3. Interpolations of the sets of observations illustrated in Fig. 1, depicted as solid orange and dashed green lines. The vertical dashed lines indicate the common temporal domain of the two interpolations $[a(.,.), b(.,.)]$. The area between the two interpolations over the common definition interval is represented by the red filling.

Firstly, the interpolations can only be compared over their common definition interval. If there exists a common definition interval between i and j, we denote it by $[a(i,j), b(i,j)]$. This interval begins at the time of the sensor that arrived the latest and ends at the time of the sensor that leaves the earliest:

$$a(i,j) = \max\{\min\{t \in T_i\}, \min\{t \in T_j\}\}$$
$$b(i,j) = \min\{\max\{t \in T_i\}, \max\{t \in T_j\}\}$$

Hence, the duration of the common definition interval, denoted by $\delta(i,j)$, is defined by:

$$\delta(i,j) = \max\{0, (b(i,j) - a(i,j))\} \tag{2}$$

Furthermore, since the interpolation method aims to minimize the average difference between the ground truth and the estimation, we define the distance $d(i,j)$ as the mean magnitude difference between the interpolations. If the duration of the common definition interval is not zero, it can be mathematically expressed as:

$$d_{\text{interp-mean}}(i,j) = \frac{1}{\delta(i,j)} \int_{a(i,j)}^{b(i,j)} |\hat{\theta}_i(t) - \hat{\theta}_j(t)| dt \tag{3}$$

V. Weighed Mean Linkage Hierarchical Clustering

In this section, we propose a method that relies on the presented similarity measure to cluster together sensors that are considered similar, using a hierarchical clustering approach.

A. Specification of the Clustering Problem

In a typical clustering problem, objects are considered with n variables, and the goal is to group together objects that are close when represented in a space where each variable constitutes a dimension. Commonly, standard similarity metrics based on vectors are employed for such clustering tasks [21-23].

In our specific context, an object represents a sensor, its set of observations, and its interpolation based on Kriging defined over a specific time interval. Here, the calculation of distance is not as straightforward, which is why we have dedicated a specific section to it. Thus, we were able to define a distance (which can be $None$) $d(.,.)$ and a common definition interval duration $\delta(.,.)$ between two sensors.

This change implies specific considerations in devising a clustering solution:

- Some pairs of sensors may have an unknown distance: they are defined over disjoint intervals, making it impossible to determine their proximity,
- The duration of the common definition interval is an essential indicator for defining the quality of the distance measure: a distance calculated over a longer period carries more significance than one computed over a very short duration.

B. Agglomerative Hierarchical Clustering Basics

Algorithm Principles: For this problem, we choose to focus on solutions based on AHC. This clustering method involves iteratively merging clusters together [24].

Initially, each object (sensor) is considered as its own cluster. At each iteration, the two closest clusters are merged to form a new cluster. Consequently, in each iteration, we obtain one less cluster than in the previous iteration. The merging process terminates when the stopping criterion is met; this stopping criterion can be the final number of clusters or based on intra-cluster and inter-cluster distances.

Linkage Method: An essential aspect here is the definition of the distance between clusters. The method that relies on inter-object distances to determine the inter-cluster distance is referred to as the *linkage method*. In Fig. 4, we illustrate several linkage methods: Simple-link defines the distance between clusters as the smallest distance between any pair of objects from a different cluster; complete-link uses the largest distance between any pair of objects from a different cluster; average-link calculates the average of all pairwise distances between objects from a different cluster.

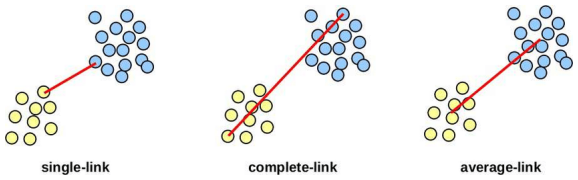

Fig. 4. Examples of linkage methods, from [25].

C. Weighted Mean Linkage Method

In the literature, various common linkage methods exist, all of which involve linear combinations of distances between the elements of the clusters being compared. Here, we choose to adapt the average-link to better suit our problem. We weigh the distances by the duration of the common definition interval

to give more importance to distances calculated over longer periods.

Let $d(i,j)$ be the distance between sensors i and j calculated using the method described in Eq. (3), and $\delta(i,j)$ be the duration of their common definition interval, as defined in Eq. (2). When two sensors are not directly comparable, $\delta(i,j) = 0$, and $d(i,j) = None$, and our convention dictates $\delta(i,j)d(i,j) = 0$.

We define the distance between two clusters as the mean of distances between pairs of objects from different clusters, weighted by their common definition interval duration. Considering $i \in I$ as the set of sensors included in cluster I, and $j \in J$ for J, the distance between clusters I and J is given by:

$$D(I,J) = \frac{\sum_{i\in I}\sum_{j\in J}\delta(i,j)d(i,j)}{\sum_{i\in I}\sum_{j\in J}\delta(i,j)} \tag{4}$$

(If all distances between i and j are unknown, then by convention, we will have $D(I,J) = None$, and we will not merge I and J.)

For this linkage method, we employed the Lance-Williams algorithm as a reference for hierarchical clustering implementation [26]. This algorithm updates the distance between clusters at each merging step. First, we extend the notation $\delta(.)$, with $\delta(I,J)$ being the sum of the duration of the common definition interval between each sensor from I and from J. Mathematicaly, this means: $\delta(I,J) = \sum_{i\in I}\sum_{j\in J}\delta(i,j)$.

Denoting the cluster composed of elements from clusters I and J by $I+J$, after this merging, we update its distance with another cluster K. The update formulas are as follows:

$$\begin{aligned} D(I+J,K) = &\ \tfrac{\delta(I,K)}{\delta(I,K)+\delta(J,K)}D(I,K) \\ &+\tfrac{\delta(J,K)}{\delta(I,K)+\delta(J,K)}D(J,K) \\ \delta(I+J,K) = &\ \delta(I,K)+\delta(J,K) \end{aligned}$$

As a reminder of the hierarchical algorithm, in each round, we choose to merge clusters with the smallest distance D based on this distance definition.

D. Stopping Criterion

We will delve into the stopping criterion for this AHC method in the simulation section, as this criterion plays a crucial role in the performance of such methods. We will consider two types of stopping criteria.

Firstly, since we will introduce in the simulation part a comparative clustering method to evaluate the performance of the proposed approach in this paper, we aim to compare these methods fairly. Therefore, the first stopping criterion will be the maximum number of clusters.

On the other hand, arbitrarily defining the final number of clusters is not always the best option for achieving optimal performance [26]. Therefore, we also propose a stopping criterion that fix a threshold to the maximum distance between clusters. This threshold is specific to our distance definition and is therefore not relevant for the comparative clustering method.

symbol	Meaning	Value(s)
	Phenomena Parameters Section VI-A1	
ω_i, ϕ_i	Frequencies of signal i	$\mathcal{U}(0, \frac{2\pi}{30})$
α_i, β_i	Amplitudes of signal i	$\mathcal{U}(-100, 100)$
	Rescaling of the phenomena values	$[-1, 1]$
	Sensors Observations Section VI-A2	
λ	Sensor arrival rate	0.1
$1/\gamma$	Average number of sent observations	1
μ	Sensor existing time rate	0.01
	End of Simulation	$t = 1000$
	Sensors Considered in simulation	Alive at $t = 200$
	V-measure Parameter Section VI-C	
a	Weight given for Homogeneity	1
	Evaluation Using a Comparative Method Section VI-D	
	Max nb of clusters	3
σ	Std of Gaussian noise	0.2
	Robustness to Noise Variations Section VI-E	
C	Zero noise threshold	0.1
k	Noise dependent threshold	0.8
σ	Std of Gaussian noise	$\{0 + 0.05i,$ $0 \leq i \leq 10\}$

TABLE I
PARAMETERS OF THE SIMULATION

VI. Simulations

In this section, we perform simulations by generating two distinct continuous phenomena, each sensor consistently following one of the two phenomena. Specifically, an observation is the value of the corresponding phenomenon at the time of measurement, with added noise. We model the characteristics of sensors observations using exponential laws, such as entrance of new sensors and their duration, and the observations made over time. We vary the measurement noise to study the extent to which our solution can identify similarities and group sensors following the same phenomenon.

To assess the performance of our solution, we construct alternative propositions. We leverage DTW as an algorithm that computes distance, taking into account the peculiarities of the considered time series. Additionally, we implement a hierarchical clustering algorithm based on the principle of clique partitioning. We demonstrate the superiority of our approach over this competing solution. Additionally, we explore the performance of our solution across a range of measurement noises, examining its robustness using various stopping criterion strategies.

The parameters of all the simulation part are sum-up in Table I.

A. Generation of Phenomena and Sensor Observations

The assumptions regarding the phenomena, sensor inputs and outputs, as well as the transmitted observations, are presented here in detail, and visible in Fig. 5.

1) Generation of Phenomena:

We define a phenomenon using a continuous function over time. In this study, we consider two phenomena, each generated in the same way. Specifically, the generic function is given by:

$$f(t) = \sum_{i=1}^{30}(\alpha_i \cos\omega_i t + \beta_i \sin\phi_i t)$$

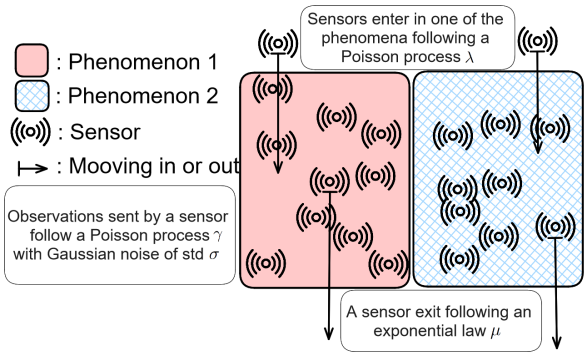

Fig. 5. Representation of the phenomena, sensor inputs and outputs, and the transmitted noisy observations.

For each $i \in \{1, 30\}$ and for each of the two phenomena, the constants α_i and β_i are chosen from a uniform distribution $\mathcal{U}(-100, 100)$, and the frequencies ω_i and ϕ_i are chosen from a uniform distribution $\mathcal{U}(0, \frac{2\pi}{30})$ (ensuring a minimum oscillation period of 30, limiting the variability). Then, we rescale the function to the range $[-1, 1]$, compressing the phenomena values into a small value segment. We keep the same phenomena for all the simulation parts, and they are depicted in Fig. 6.

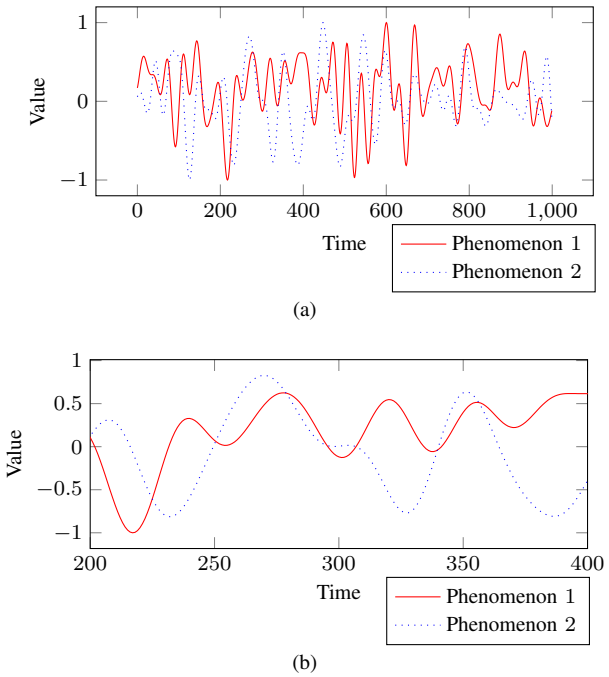

Fig. 6. Phenomena: (a) in their entirety, (b) zoomed between $t = 200$ and 400.

2) Generation of Sensors Observations:

Each sensor follows one of the two phenomena, always the same one, and sends a noisy observation of the phenomenon, with Gaussian noise of standard deviation σ. The arrivals, departures, and observation times of the sensors are generated according to statistical laws. New sensors enter the environment over time, following a Poisson distribution with a parameter of $\lambda = 0.1$, and each of them follows one of the two phenomena with equal probability. The duration of a sensor's stay in the environment follows an exponential distribution with a parameter of $\mu = 0.01$. While in the environment, a sensor transmits observations following a Poisson distribution with a parameter of $\gamma = 1$.

We terminate the simulation at $t = 1000$. To mitigate too much with cases where a phenomenon ceases to be tracked by a sensor, we initiate the evaluation when a sufficient number of sensors have entered the environment. Specifically, we consider only those sensors that remain active after $t = 200$.

Given that the average duration of a sensor in the environment is $\frac{1}{\mu} = 100$, it is noteworthy that there are a considerable number of pairs of sensors with zero overlapping definition intervals.

We define a *run* as the generation of a new set of sensors' observations, which is then used for all the compared methods.

B. Kriging Parameter Settings

The Kriging requires fitting the experimental variogram to the variogram model. We have chosen the Gaussian model defined in Eq. (1). In the survey [18], it was established that the choice of variogram model is relatively unimportant compared to the parameters associated with this model. Hence, we propose a robust method for fixing the variogram parameters.

In our simulations, we used the Pykrige package in Python, which we utilized to create Kriging interpolations. This module can estimate the parameters of nugget n, sill s, and range r based on a given variogram model. However, since the sensor observations are randomly generated with random measurement noise, the parameter estimation was not always accurate. In some cases, the parameter estimation led to very strong variations in the interpolation (e.g., small range r), while in other cases, it resulted in a nearly linear interpolation (e.g., very large range r).

To address this issue, for a given run, we assume that sets of observations from each sensor have the same underlying form since they are generated using the same random laws; therefore, they should be interpolated with the same variogram model. To achieve this, for a given run, we fix the parameters n, s, and r that will be the same for all interpolations to make.

For one run, for each sensor i, we estimate the triplet of parameters n_i, s_i, and r_i using the fitting function provided by the PyKrige package. Consequently, for each parameter, we define the value for the variogram model across all sensors in the run as the median value.

C. Using V-measure to Evaluate Clustering Performance

To evaluate the performance of a clustering solution, we assess the clustering results in comparison to the true membership of sensors to their corresponding phenomenon. A so-called class is defined by one phenomenon and its related sensors, and we compare this set of classes with the set of clusters formed by the evaluated clustering method.

A method to evaluate the performance of a clustering algorithm when true labels are known relies on two measures:

Completeness and Homogeneity, forming the V-measure. These measures are based on conditional entropy and provide a score ranging from 0 to 1; the mathematical expressions are developed in [27].

On the one hand, Homogeneity evaluates the proportion of a cluster containing elements from the same class. In the extreme case, a clustering with perfect Homogeneity would involve constructing a cluster for each object.

On the other hand, Completeness evaluates the proportion of a class being grouped into the same cluster. In the extreme case, a clustering with perfect Completeness would involve constructing a single cluster containing all objects.

A score of 1 corresponds to perfect Completeness (respectively Homogeneity), while 0 indicates null Completeness (Homogeneity).

These two metrics characterize two main aspects of a clustering performance. The weighted harmonic mean by a (that we choose $a = 1$), known as V-measure, is defined as:

$$\text{V-measure} = \frac{(1+a)\text{Homogeneity} \times \text{Completeness}}{a * \text{Homogeneity} + \text{Completeness}}$$

D. Evaluation Using a Comparative Method

We begin by assessing our proposal in comparison to alternatives found in the literature, introducing a comparative similarity metric and clustering algorithm.

Subsequently, we evaluate the various possible combinations, opting for either our proposed method or the comparative one, for each of the two components of the clustering methodology. We demonstrate the effectiveness of each proposal compared to the alternative ones.

1) Comparison Similarity - Dynamic Time Warping: Due to the non-synchronicity of observations, conventional distance metrics for time series, which rely on observations at identical instances, are not directly applicable. In [9,10], which also aims to create groups of similar sensors, their similarity metric between two sets of observations is based on the maximum difference between pairs of observations made at the same instants and on similar trends (rise or fall). However, observations between sensors are not synchronized, and with measurement noise, neither of these metrics seems to be suitable.

Still, algorithms based on time series that could address such variability exist, with DTW being a notable example. DTW aims to measure the similarity between two time series, accommodating temporal shifts or differences in sampling between the compared time series [28].

Considering sets of observations from sensors i and j as θ_i and θ_j, assumed without loss of generality to be defined over the same interval (otherwise, we constrain T_i, T_j to their common definition set), DTW relies on the distance matrix between all pairs of observation values $(d(\theta_{i,t_i}, \theta_{j,t_j}))_{t_i \in T_i, t_j \in T_j}$. In this simulation, we choose the distance function as the absolute difference between the two compared values $d(\theta_{i,t_i}, \theta_{j,t_j}) = |\theta_{i,t_i} - \theta_{j,t_j}|$.

A path is defined in that matrix, starting from the earliest instants of both historical observations (top left corner of the matrix) and progressing in proximity (vertical, horizontal, diagonal, always forward) until reaching the opposite end of the matrix (bottom right corner). The value associated with this path is the sum of the matrix values it traverses. In this matrix representation, for example, the Manhattan distance is defined thanks to the path along the diagonal of the matrix when the matrix is square. The DTW chooses the path with the smallest value - and in its normalized form, divided by the sum of the matrix sides $n_i + n_j$. The pseudocode of this algorithm is presented in Algorithm 1.

Algorithm 1 Normalized DTW algorithm. Abuse have been made, representing observation times with indexes respectively $[1..n_i]$ and $[1..n_j]$ in order to facilitate the understanding.

Require: $\theta_i = (\theta_{i,k})_{k \in 1..n_i}$, $\theta_j = (\theta_{j,l})_{l \in 1..n_j}$
DTW:= array $k \in 1..n_i, l \in 1..n_j$, $DTW[k,l] = |\theta_{i,k} - \theta_{j,l}|$
for $k \in [2..n_i]$ **do**
  $DTW[k,1] = DTW[k,1] + DTW[k-1,1]$
end for
for $l \in [2..n_j]$ **do**
  $DTW[1,l] = DTW[1,l] + DTW[1,l-1]$
end for
for $k \in [2..n_i]$ **do**
  for $l \in [2..n_j]$ **do**
    $DTW[k,l] = DTW[k,l] + \min\{DTW[k-1,l], DTW[k,l-1], DTW[k-1,l-1]\}$
  end for
end for
return $\frac{DTW[n_i,n_j]}{n_i+n_j}$

2) Comparison Clustering - AHC with Complete Linkage: Talking about the clustering method, we propose to compare our solution to an approach extracted from the literature, specifically the solution proposed in [9,10]. In these references, the sensors transmit observations at exactly the same time points. Two sensors are defined as similar if the maximum amplitude difference between their observations does not exceed a threshold. The problem is thus formulated as a sensor graph where the edges represent similarity links. They have developed an algorithm that performs clique partitioning, meaning a partition of the sensor set such that each group contains sensors that are all mutually similar.

To enable a meaningful comparison between our approach and the one proposed in the literature, we retain certain aspects of our methodology. Specifically, we aim to evaluate this solution on a common ground, which is why we decide to adapt this principle to the hierarchical clustering algorithm. Drawing an analogy with the clique partitioning method, we choose a complete linkage method [29]. This linkage method defines the distance between two clusters as the maximum existing distance between each pair of objects from different clusters:

$$D(I,J) = \max\{d(i,j), d(i,j) \neq None, i \in I, j \in J\}$$

Similarity metric	Linkage method	Homogeneity		Completness		V-measure	
		Mean	Std	Mean	Std	Mean	Std
Mean interpolation difference	**Weighed mean**	0.72	0.23	0.60	0.20	0.65	0.22
Mean interpolation difference	Complete	0.70	0.21	0.52	0.16	0.59	0.18
Dynamic Time Warping	**Weighed mean**	0.53	0.34	0.50	0.27	0.50	0.31
Dynamic Time Warping	Complete	0.60	0.22	0.43	0.17	0.50	0.19

TABLE II

CLUSTERING PERFORMANCE COMPARISON USING A SIMILARITY METRIC AND A LINKAGE METHOD FOR AHC FROM BOTH OUR PROPOSED SOLUTION AND THE COMPARATIVE APPROACH, WITH A PREDEFINED NUMBER OF FINAL CLUSTERS SET TO 3 AND SENSOR MEASUREMENT NOISE $\sigma = 0.2$. PRESENTATION OF AVERAGE VALUES AND STANDARD DEVIATIONS OF HOMOGENEITY, COMPLETENESS, AND V-MEASURE. HIGHLIGHTING OUR CONTRIBUTIONS IN **BOLD**.

Thus, at each stage, we merge the two clusters that have the lowest distance, hence restricting the maximum distance between two sensors that belong to the same cluster.

3) Setting the Maximum Number of Clusters: As mentioned in Section V-D, to ensure a fair comparison between the two comparison methods, we need to choose a stopping criterion that is not dependent on the distance, hence the choice of the maximum number of clusters.

The ideal number of clusters is 2, in the best case, one cluster containing the sensors following the first phenomenon, and the second containing those following the second phenomenon. However, due to simulations driven by random variables, the created objects exhibit significant variability. We conduct a substantial number of simulations, consistently regenerating sets of sensor observations, revealing instances where the decision to have two clusters proved suboptimal. We identified cases where choosing two clusters yields poor clustering results.

- A phenomenon may, at a time in simulation, be followed by no sensors, leading to the grouping of sensors before and after this cut without being able to group them due to a null common definition interval. Thus, two clusters should represent the two disjoint part of the same phenomenon.
- When the common interval duration is short and when the phenomena overlap, a pair of sensors following different phenomena may have a very low distance. In such cases, these sensors might be grouped together, even though they follow different phenomena. Isolating this pair as much as possible is thus essential, leading to an additional cluster.
- If the noise is substantial, and a sensor transmits very few observations over a short time interval, it can provide observations significantly different from other sensors. We observed that our clustering method might prioritize grouping sensors following different phenomena and leave such a sensor alone in its cluster.

For these reasons, we opt to set 3 clusters. In this case, this choice is not always optimal, but it is a compromise to obtain sufficiently consistent groups and comparable results.

4) Simulation Settings: We aim to assess the relevance of our choices for the similarity metric and linkage method. With the comparative method we have just presented, we have the option to choose between two similarity metrics and two clustering methods. Firstly, for the similarity metric, we can opt for our proposal – which calculates the average difference between interpolations – or the DTW method. Secondly, for the clustering method, the two proposed approaches involve AHC, either using our weighted mean linkage or the complete linkage method.

By selecting a similarity metric and a clustering method, we obtain four different methods, allowing us to investigate the performance impact of altering one component of the methodology.

We set the measurement noise to $\sigma = 0.2$ and conduct 1000 runs. The average performance along with the standard deviation of Homogeneity, Completeness, and V-measure can be observed in Table II.

5) Discussion of the Results: Globally, we achieve a 23% improvement in terms of V-measure performance compared to the method we have chosen for comparison, demonstrating its superiority, which is evident in both Homogeneity ($+28\%$) and Completeness ($+16\%$).

It can be observed that the use of our similarity method has the most positive impact on performance. Employing this metric with the complete linkage method constitutes the second-best solution, whether in terms of Homogeneity or Completeness. Therefore, it seems preferable to use interpolation methods followed by traditional time series distance measures. This approach demonstrates greater effectiveness compared to time series-based algorithms, such as DTW, which solely consider the order of arrivals rather than the observation times. It is crucial to note, however, that although the interpolation method is robust to disturbances in sensors observations (irregular and noisy), it remains a parametric method that requires the phenomena being tracked not to have too abrupt changes.

Regarding different linkage methods, our linkage method enhances performance by 9% when combined with the mean interpolation difference similarity metric, although, interestingly, when using the DTW metric, applying either the complete linkage or the weighted mean linkage results in similar overall V-measure performance. Our objective was to consider the duration of the common comparison between compared sensors, giving more weight to pairs of sensors defined over a longer common definition interval. In contrast, complete linkage only retains the most significant distance and does not incorporate the duration of the common definition interval into its distance calculation.

E. Robustness of the Solution to Noise Variations

In this section, we assess the robustness of our solution to variations in measurement noise.

The stopping criterion can significantly impact measurement performance. Therefore, we introduce another thresholding method based on the distance between clusters.

1) Setting of the Threshold Based on Inter-cluster Distance: As explained in Section VI-D3, the number of "optimal" clusters can vary, ranging from a minimum of 2 clusters to a potentially higher number due to the strong variability inherent in the considered simulation.

Hence, we propose a stopping criterion that is a threshold for the maximum distance between clusters.

Firstly, with zero noise, since the distance is based on interpolations over sets of irregular observations, when sensors belong to the same phenomenon, the distance is non-zero. The threshold for zero noise distance must, therefore, have a non-zero value.

Furthermore, as the noise increases, the distance between two sensors following the same phenomenon becomes larger. Analogous to confidence interval definitions, we set the threshold distance proportionally to the intensity of the measurement noise σ.

Thus, we define our threshold in a generic form:

$$D_{\text{threshold}} = C + k\sigma \tag{5}$$

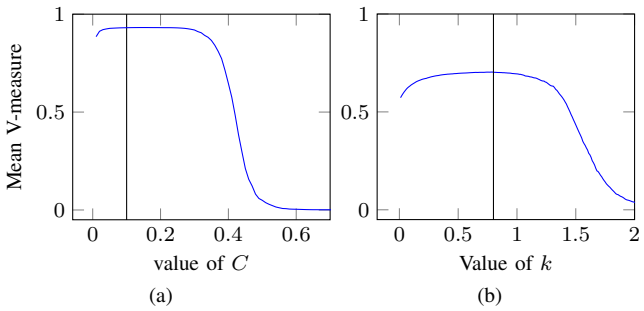

Fig. 7. (a) Average V-measure performances with zero noise, while varying the threshold distance parameter C.
(b) Average V-measure performances with noise $\sigma = 0.2$, while varying the threshold distance parameter k, and with $C = 0.1$.
The vertical line represents the chosen parameter value for C and k, further used.

Setting the Threshold Parameters: To define the parameters C and k, we analyze the impact of each parameter on clustering performance by measuring the V-measure.

First, we set the value of C in Eq. (5), corresponding to the threshold chosen for zero noise. Thus, we conduct 1000 runs considering sensors with zero noise $\sigma = 0$ and, for each run, apply the clustering method for various values of $C \in [0, 1]$. We then measure the V-measure for each clustering result, with the mean results shown in Fig. 7(a). It is observed that the performance remains relatively stable when $C \in [0.05, 0.28]$, with values ranging between $[0.925, 0.932]$. Based on this result, we choose $C = 0.1$.

After fixing this parameter, we set k in Eq. (5). Here, we set the noise to $\sigma = 0.2$, and with $C = 0.1$, perform 1000 runs, with the average V-measure results presented in Fig. 7(b). Similarly, we observe a plateau where the performance varies minimally with the choice of k, for $k \in [0.5, 1]$ (with values within $[0.694, 0.702]$). Consequently, we choose $k = 0.8$.

These stable phases indicate that a wide range of values for C and k contribute to close clustering performance.

2) Evaluation of the Clustering Performance for Different Noises: We assess the robustness of our clustering method by evaluating its sensitivity to measurement noise. Our approach combines a similarity metric based on the average amplitude difference between kriging interpolations and hierarchical clustering with a mean linkage method weighted by the duration of common intervals. We evaluate the performance of this solution using two stopping criteria: a maximum of 3 clusters and a maximum inter-cluster distance threshold exceeding $D_{\text{threshold}} = 0.1 + 0.8\sigma$.

Considering that phenomena values range between -1 and 1, we conduct 1000 runs for each noise level $\sigma = \{0 + i * 0.05, 0 \leq i \leq 10\}$. The results are displayed in Fig. 8, showcasing Homogeneity (a), Completeness (b), and V-measure (c). For the method with the stopping criterion defined by the inter-cluster distance, we present the average number of final clusters according to the measurement noise in Fig. 8 (d).

Overall, for both stopping criteria, noise significantly impacts clustering performance, with an average V-measure decrease of 34% from zero noise to $\sigma = 0.25$ when using the distance-based stopping criterion, and a decrease of 32% when fixing the final number of clusters. It's worth noting that, overall, the formed clusters are more homogeneous than complete, given that there are only two classes to cluster.

Comparing the two stopping criteria, when noise is low ($\sigma \leq 0.25$), the distance-based stopping criterion outperforms, both in terms of average Homogeneity and Completeness. On average, the final number of clusters is below 3 (2.7 clusters for $\sigma = 0.25$), which is advantageous compared to the maximum cluster number stopping criterion. Thus, for $\sigma < 0.25$, there is a difference of at least 7.8% in terms of mean V-measure in favor of the distance-based stopping criterion.

However, as noise increases, the distance-based stopping criterion becomes more sensitive. Indeed, with relatively high noise levels ($\sigma > 0.35$), the average number of final clusters increases significantly (6.6 for $\sigma = 0.4$, 17.9 for $\sigma = 0.45$, and 39.0 for $\sigma = 0.5$). The figure may seem confusing because the average number of clusters increases significantly without influencing completeness as it should. In reality, although not visible in these figures, in the simulation results, there is an alternation between very low cluster numbers (when one cluster is formed, completeness is 1) and very high cluster numbers (lower completeness).

VII. Conclusion and Perspectives

In this study, we proposed a method for grouping similar sensors based on their observations, addressing the challenges of a scenario where sensors exhibit irregular and noisy behavior over time.

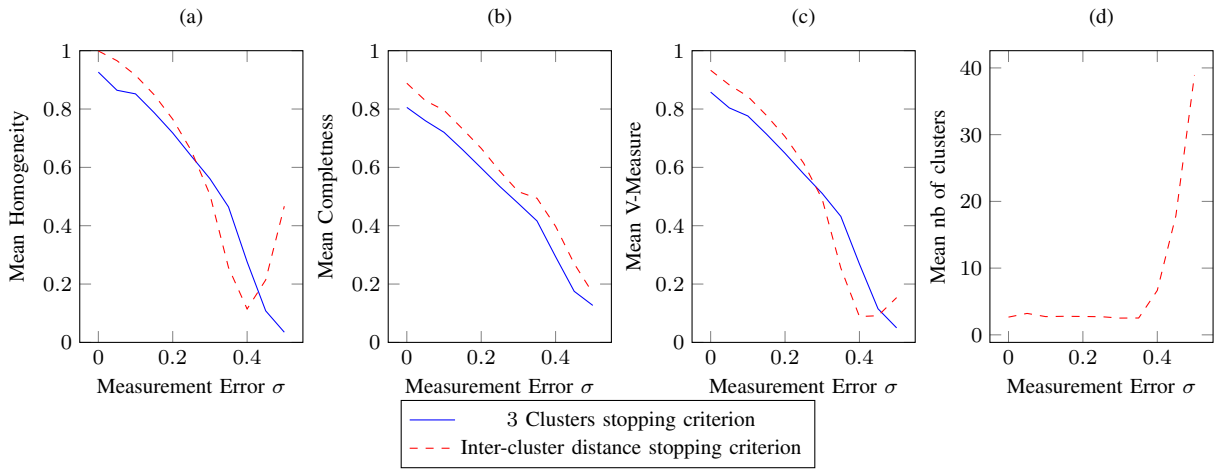

Fig. 8. Performance evaluation of our clustering solution (mean interpolation difference + weighted mean linkage method for AHC) under various levels of measurement noise, with a stopping criterion based on maximum number of sensors (solid blue line) and one based on threshold inter-cluster distance (dashed red line).
(a) Mean Homogeneity, (b) Mean Completeness, and (c) Mean V-measure of clustering results for the two compared solutions. (d) Average number of final clusters when using the distance-based inter-cluster stopping criterion.

We developed a similarity metric based on Kriging interpolation and utilized a hierarchical clustering method, introducing a novel linkage approach tailored to the problem. Through simulations, we compared our approach with existing methods from the literature, showcasing its superiority in the considered scenario with a 23% increase in V-measure. Furthermore, we explored the robustness of our solution under different measurement noise levels, employing two strategies for the stopping criterion for our AHC.

This work paves the way for further analyses, delving deeper into potential challenges within the Massive IoT context. Specifically, it is relevant to consider scenarios where sensors remain in the environment but no longer track the intended phenomenon, either due to physical sensor movement or sending of aberrant data if the sensor get corrupted.

Viewed as a stepping stone within a broader vision, the ultimate goal of such deployments is real-time assessment of observed phenomena, supporting a digital twin for instance. Subsequent steps could involve evaluating the precision of reconstructing tracked phenomena using this clustering method, while varying the number of messages sent to the digital twin. In our prior work, as an example, we developed methods to distribute the load among similar sensor groups, ensuring a fixed quantity of messages for each cluster, regardless of cluster size [12,13].

REFERENCES

[1] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, "Data Mining for Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 77–97, 2014.

[2] S. L. Ullo and G. R. Sinha, "Advances in Smart Environment Monitoring Systems Using IoT and Sensors," *Sensors*, vol. 20, p. 3113, May 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, pp. 891–902, Feb. 2021.

[4] Y. Liu and K. Yang, "Communication, sensing, computing and energy harvesting in smart cities," *IET Smart Cities*, p. smc2.12041, Sept. 2022.

[5] A. Ikpehai, B. Adebisi, K. M. Rabie, K. Anoh, R. E. Ande, M. Hammoudeh, H. Gacanin, and U. M. Mbanaso, "Low-Power Wide Area Network Technologies for Internet-of-Things: A Comparative Review," *IEEE Internet of Things Journal*, vol. 6, pp. 2225–2240, Apr. 2019.

[6] N. H. Motlagh, E. Lagerspetz, P. Nurmi, X. Li, S. Varjonen, J. Mineraud, M. Siekkinen, A. Rebeiro-Hargrave, T. Hussein, T. Petaja, M. Kulmala, and S. Tarkoma, "Toward Massive Scale Air Quality Monitoring," *IEEE Communications Magazine*, vol. 58, pp. 54–59, Feb. 2020. Conference Name: IEEE Communications Magazine.

[7] Y. Yoo, "Data-driven fault detection process using correlation based clustering," *Computers in Industry*, vol. 122, p. 103279, Nov. 2020.

[8] F. Koushanfar, N. Taft, and M. Potkonjak, "Sleeping Coordination for Comprehensive Sensing Using Isotonic Regression and Domatic Partitions," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, (Barcelona, Spain), pp. 1–13, IEEE, 2006.

[9] C. Liu, K. Wu, and J. Pei, "An Energy-Efficient Data Collection Framework for Wireless Sensor Networks by Exploiting Spatiotemporal Correlation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, pp. 1010–1023, July 2007. Conference Name: IEEE Transactions on Parallel and Distributed Systems.

[10] Z. Liu, W. Xing, Y. Wang, and D. Lu, "Hierarchical Spatial Clustering in Multihop Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 9, p. 528980, Nov. 2013. Publisher: SAGE Publications.

[11] B. Q. Ali, N. Pissinou, and K. Makki, "Identification and Validation of Spatio-Temporal Associations in Wireless Sensor Networks," in *2009 Third International Conference on Sensor Technologies and Applications*, (Athens, Greece), pp. 496–501, IEEE, June 2009.

[12] G. Maudet, M. Batton-Hubert, P. Maille, and L. Toutain, "Emission Scheduling Strategies for Massive-IoT: Implementation and Performance Optimization," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–4, Apr. 2022. ISSN: 2374-9709.

[13] G. Maudet, M. Batton-Hubert, P. Maille, and L. Toutain, "Energy Efficient Message Scheduling with Redundancy Control for Massive IoT Monitoring," in *2023 IEEE Wireless Communications and Networking, 2023. WCNC 2023.*, 2023.

[14] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks," in *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 200–209 vol.1, June 2002. ISSN: 0743-166X.

[15] G. M. Dias, B. Bellalta, and S. Oechsner, "A Survey About Prediction-Based Data Reduction in Wireless Sensor Networks," *ACM Computing Surveys*, vol. 49, pp. 1–35, Sept. 2017.

[16] C. Caruso and F. Quarta, "Interpolation methods comparison," *Computers & Mathematics with Applications*, vol. 35, pp. 109–126, June 1998.
[17] J. P. Kleijnen, "Kriging metamodeling in simulation: A review," *European Journal of Operational Research*, vol. 192, pp. 707–716, Feb. 2009.
[18] V. Picheny, T. Wagner, and D. Ginsbourger, "A benchmark of kriging-based infill criteria for noisy optimization," *Structural and Multidisciplinary Optimization*, vol. 48, pp. 607–626, Sept. 2013.
[19] D. Zimmerman, C. Pavlik, A. Ruggles, and M. P. Armstrong, "An Experimental Comparison of Ordinary and Universal Kriging and Inverse Distance Weighting," *Mathematical Geology*, vol. 31, no. 4, pp. 375–390, 1999.
[20] C. C. Castello, J. Fan, A. Davari, and R.-X. Chen, "Optimal sensor placement strategy for environmental monitoring using Wireless Sensor Networks," in *2010 42nd Southeastern Symposium on System Theory (SSST 2010)*, (Tyler, TX, USA), pp. 275–279, IEEE, Mar. 2010.
[21] S. Déjean, P. G. P. Martin, A. Baccini, and P. Besse, "Clustering Time-Series Gene Expression Data Using Smoothing Spline Derivatives," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2007, pp. 1–10, 2007.
[22] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, "Time-series clustering – A decade review," *Information Systems*, vol. 53, pp. 16–38, Oct. 2015.
[23] T. W. Liao, "Clustering of time series data-a survey," *Pattern Recognition*, Nov. 2005.
[24] 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, Jan. 1967.
[25] P. Guevara, *Inference of a human brain fiber bundle atlas from high angular resolution diffusion imaging*. PhD thesis, University of Concepción, Oct. 2011.
[26] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *WIREs Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.53.
[27] A. Rosenberg and J. Hirschberg, "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure," Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2007.
[28] S. Salvador and P. Chan, "Toward Accurate Dynamic Time Warping in Linear Time and Space," in *KDD Workshop on Mining Temporal and Sequential Data*, vol. 11, pp. 70–80, Jan. 2004.
[29] A. Großwendt and H. Röglin, "Improved Analysis of Complete-Linkage Clustering," *Algorithmica*, vol. 78, pp. 1131–1150, Aug. 2017.