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, Mireille Batton-Hubert, Patrick Maillé, Laurent Toutain

***Abstract*—The expanding Internet of Things landscape, combined with a significant reduction in the cost of connected devices, has enabled the widespread deployment of sensors. These sensors, often situated in close proximity to each other, frequently produce overlapping data.**

This paper aims to identify such overlaps in sensor data to effectively cluster similar sensors. The benefits of clustering include enhanced detection of sensor failures and a reduction in data transmissions. We examine a typical scenario where sensors are deployed asynchronously, operate for a limited time within the environment, and emit data that is noisy, irregular, and unsynchronized.

To overcome these challenges, we have developed a similarity metric that employs interpolation techniques to manage noisy, irregular, and unsynchronized data. This metric supports a hierarchical clustering algorithm featuring a novel linkage method tailored to the dynamic nature of sensor deployments. The goal is to cluster sensors that monitor the same phenomenon, regardless of their active periods not coinciding.

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

***Index Terms*—Constrained Devices, Efficient Communications and Networking, Network Architecture, Data Management and Analytics, Smart Cities, Smart Environment, Sensor Phenomenon and Characterization, Low Cost Sensors and Devices.**

I. Introduction

The Internet of Things (IoT) has significantly transformed the landscape of extensive monitoring solutions by providing a plethora of methodologies applicable across various scenarios, such as optimizing resources and flows, managing risks, and facilitating precise tracking [1]. These technologies find utility in a wide array of sectors, including but not limited to agriculture [2], industrial applications [3], and the development of smart cities [4]. Advances in electronic engineering, combined with the advent of networks characterized by stringent constraints, have paved the way for the creation of embedded sensors. These devices are engineered to execute specific, straightforward tasks, such as the measurement of temperature, humidity, or CO2 levels [5]. Owing to their battery-powered nature and affordability, these sensors are readily deployable on a grand scale, potentially being integrated into commonplace objects [6].

Given the extensive deployment of these sensors, it is common for several of them to be situated in proximity to one another, hence transmitting data that are closely related. This phenomenon is advantageous for several reasons: it facilitates the swift identification of malfunctioning sensors through the comparison of their data against that of similar sensors [7], and it provides an opportunity to diminish the volume of transmitted data by leveraging the similarity in the collected data, as investigated in several studies [8-10].

This paper endeavors to establish a methodology for identifying clusters of similar sensors based on the analysis of their data. The environment in question is considered to comprise multiple phenomena, each exhibiting distinct temporal variations in a given physical quantity, with each sensor monitoring a specific phenomenon. The goal is to aggregate sensors that are observing identical phenomena.

This research departs from previous studies [7-10] by proposing new hypotheses that tackle the challenges arising from the extensive deployment of embedded devices. We address issues such as measurement noise, lack of synchronization, and irregular data transmission, which challenge the application of traditional similarity metrics. Additionally, the transient nature of sensor operations, with some only active in the initial stages and others introduced later, complicates the process. Determining whether such sensors are monitoring the same phenomena, especially when their operational periods do not overlap, presents a novel and complex clustering challenge that has yet to be explored.

To address these issues, our proposed solution is structured around two primary components: a metric to quantify the similarity between sensors and a hierarchical clustering mechanism aimed at grouping sensors based on similarity. Initially, we adopt Kriging (a geostatistical technique) to define an interpolator for sensor data. Following this, we ascertain the distance between two sensors over their shared interval by evaluating the difference in the average magnitude of their respective interpolations. Additionally, we introduce an Agglomerative Hierarchical Clustering (AHC) approach. Given the potential variability in the significance of the measurement interval, we propose a methodology for determining the distance between clusters, factoring in the duration of the shared definition interval.

The incorporation of Dynamic Time Warping (DTW) as an alternative similarity metric, alongside a clustering technique rooted in the principles documented in previous research [9,10], underscores the novelty of our approach. Through simulations, the superiority of both our proposed

Gwen Maudet is with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette, Luxembourg (e-mail: gwen.maudet@uni.lu).

Mireille Batton-Hubert is with Mines Saint-Etienne, Univ Clermont Auvergne, UMR CNRS 6158, F-42023, Saint-Etienne, France.

Patrick Maillé and Laurent Toutain are with IMT Atlantique, IRISA, UMR CNRS 6074, F-35700, Rennes, France.

metric and clustering methodology over these alternatives is demonstrated. Moreover, the performance of our clustering approach is evaluated across different levels of measurement noise, including a comparison of two distinct stopping criteria for the AHC method.

The structure of the paper is as follows: the discussion commences with an exploration of related works in Section II, followed by the delineation of our assumptions in Section III. The definition of our distance metric is presented in Section IV, with the clustering methodology detailed in Section V. Simulations are subsequently outlined in Section VI, culminating in our conclusions in Section VII.

II. Related works

Studies such as those by [11] introduce methods for assessing similarities between sensors based on geographical proximity, further substantiating these connections with similarity metrics derived from sensor data. Metrics employed include the Jaccard coefficient, cosine similarity, and the Pearson correlation coefficient.

[7] tackles fault detection by leveraging similarities among sensors. Through correlation analysis, potentially overlapping sensor groups are formed. A sensor deviating from its cluster indicates a fault, a method tested in an industrial environment with 17 sensors, showing superior fault detection capabilities over conventional methods.

Strategies proposed by [8-10] aim to minimize data transmissions by employing scheduling based on observations' similarities.

In the method proposed by [8], a transfer function estimates the observation of one sensor from another's measurement. When this function can accurately predict one sensor's data from another's, a directed similarity link is established. Their experiments with 54 temperature and humidity sensors demonstrated the feasibility of significantly reducing message transmissions without compromising precision.

Other proposals emphasize the use of clustering based on sensor similarities to decrease data transmissions. Here, sensors within the same cluster are activated sequentially rather than all at once. Such methodologies align closely with this paper's objectives, as they create distinct clusters of sensors meant to monitor identical phenomena.

[9] sets up links between sensors when their observations stay within a specific threshold of difference, and their trend directions match over a certain period. Clusters are then formed through a graph-based approach, leading to efficient clustering via a greedy algorithm. This method, tested with light sensors, effectively reduce energy consumption while maintaining accuracy.

Similarly, [10] applies these concepts in mesh networks, where a link is formed between two sensors if their Pearson correlation coefficient exceeds a threshold, and their average differences remain within set limits. The clustering approach, reminiscent of [9], employs a greedy algorithm for choosing cluster heads, thus optimizing network performance.

However, the assumptions underpinning these methodologies are challenged by the emergence of the Massive IoT paradigm [6]. The challenges include: (i) Facilitating extremely simplified protocols that do not require synchronization between sensors, with sensors having variable transmission periods. (ii) Accounting for potentially significant measurement errors due to the considerable miniaturization of measuring devices. (iii) Addressing the dynamic nature of Massive IoT environments where sensors may be active only temporarily and new sensors may join over time, unlike traditional settings where all sensors are present from the outset.

III. Hypotheses and Objectives

Massive IoT typically encompasses a vast number of IoT devices dispersed within an environment. These devices are often embedded in everyday objects, allowing for easy deployment without the need for professional installation. This deployment strategy involves the use of miniaturized, low-cost sensors, often battery-powered and capable of moving within the environment. A pertinent example is the management of a logistics platform with temperature control, where temperature sensors integrated into pallets provide real-time monitoring of the goods they carry.

However, such IoT solutions may suffer from measurement uncertainties. Yet, the sheer number of devices can create data redundancy, thereby enhancing reliability.

When a pallet enters a new environment, such as a cold storage area, and finds itself among other similar pallets, identifying relationships between them becomes crucial. This not only helps in assessing the quality of each sensor and potentially identifying faulty ones but also aids in reducing the total number of messages sent, thus conserving battery life.

This section delves deeper into the objectives and hypotheses related to the deployment of sensors and their data collection.

A. Identifying Sensors Observing the Same Phenomenon

In environments exhibiting multiple distinct phenomena, each with unique variations over time, sensors are strategically deployed to monitor these variations. For instance, temperature changes might differ from one room to another. Our objective is to cluster sensors monitoring the same phenomenon into exclusive groups, ensuring each sensor is grouped based on the specific phenomenon it observes.

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 large-scale IoT deployments, sensors are often integrated into mobile everyday objects, allowing them to enter or exit the monitoring environment over time. Additionally, sensors may deactivate due to hardware malfunctions or battery depletion.

Therefore, each sensor's operational life is limited, and the similarity between two sensors can only be evaluated when they coexist in the environment. Notably, this coexistence period can vary or may not occur at all.

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 highlighted in [14], synchronizing transmissions among sensors is challenging due to clock drift, requiring frequent synchronization. With the large number of sensors and limited communication capabilities, maintaining constant synchronization is energy-intensive. However, as demonstrated in [13], lack of synchronization does not inherently compromise monitoring quality, leading us to not assume synchronization between sensor transmissions.

Moreover, we consider a scenario where a sensor does not necessarily send periodic messages. This could be due to significant clock drift, loss of messages during transmission, or the adoption of alternative data collection methods such as trigger-based or model-based approaches [15].

D. Challenges Addressed in This Paper

We consider the scenario presented above, where the only information available is the history of messages sent by sensors from the initiation of the solution until a given time t. By considering all sensors active since the deployment's start, we aim to identify groups of sensors that have observed (or are currently observing) the same phenomenon. The primary challenges include: (i) developing a reliable metric that can assess similarities between sensors despite their data being noisy, transmitted irregularly and without synchronization; (ii) the ability to cluster together all sensors that have tracked the same phenomenon, specifically including those sensors that have not had overlapping periods of operation.

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.

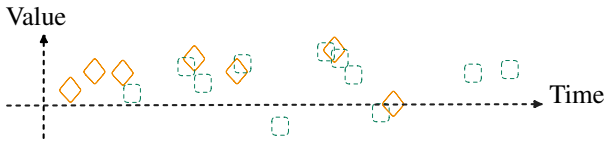

Fig. 1. Representation of two sets of observations. Orange diamonds and dashed green squares represent observations from two sensors, with time on the x-axis and observation values on the y-axis.

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:

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

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.

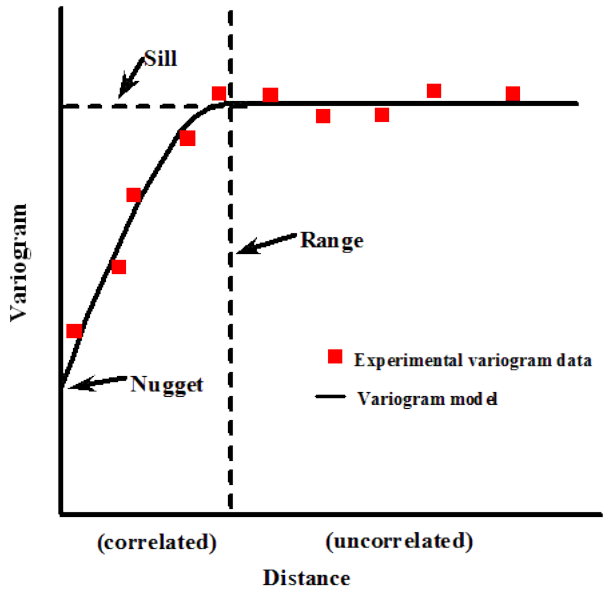

Fig. 2. Illustration of the variogram model based on experimental variogram points. The variogram consists of three parameters: nugget, sill, and range. The closer (temporally) the distance between two observations, the more correlated the values. Beyond a certain threshold, defined by the range, observations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

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:

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

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.

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

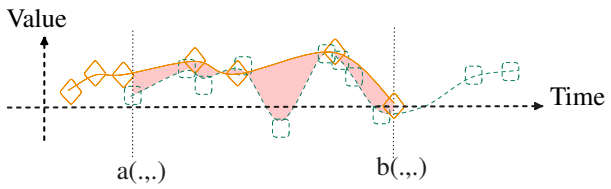

Fig. 3. Interpolations of the sets of observations illustrated in Fig. 1, depicted as solid orange and dashed green lines. The vertical dashed lines indicate the common temporal domain of the two interpolations $[a(.,.), b(.,.)]$. The area between the two interpolations over the common definition interval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

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 AHC 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.

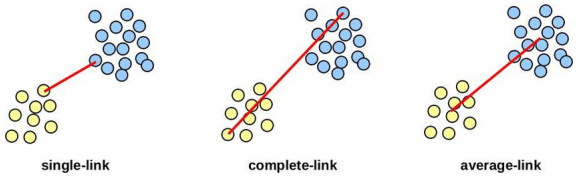

Fig. 4. Examples of single, complete and average linkage methods. from [25].

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:

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

(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:

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

As a reminder of the AHC 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.

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 random noise.

We use Poisson processes to simulate sensor arrivals in the system, and measurement instants. Similarly, the total duration in the system of each sensor is generated using an exponential distribution. 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 AHC 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 summarized in Table I.

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

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.

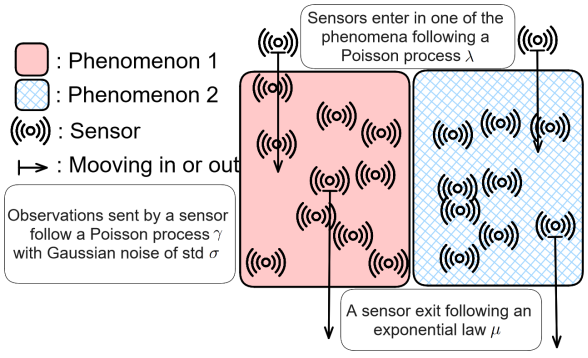

Fig. 5. Representation of the phenomena, sensor inputs and outputs, and the transmitted noisy observations.

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:

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

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.

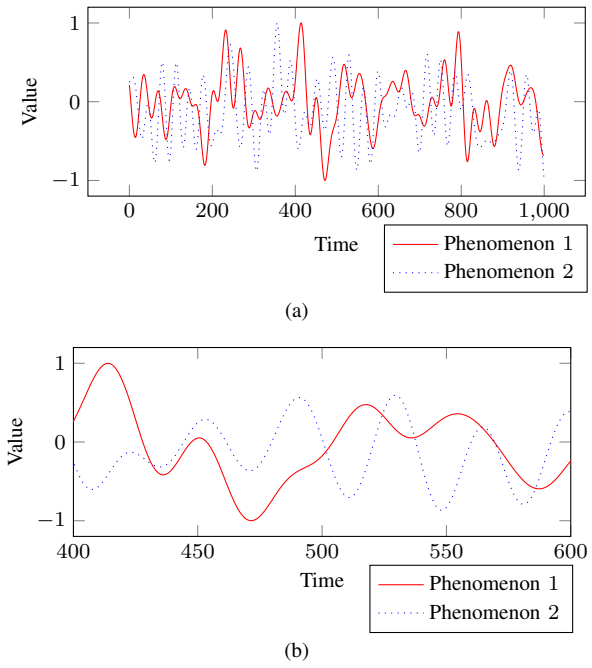

Fig. 6. Value variations of the Phenomena: (a) in their entirety, (b) zoomed between $t = 200$ and 400.

2) Generation of Sensors Observations:

Each sensor follows one of the two phenomena, the same one for its total time in the system, and sends noisy observations of the phenomenon, with Gaussian noise of standard deviation σ. New sensors enter the environment over time, following a Poisson process with a rate of $\lambda = 0.1$ arrival per time unit, 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 process with a parameter of $\gamma = 1$ per time unit.

We terminate the simulation at $t = 1000$. To mitigate 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 sensors that remain active after $t = 200$.

We define an "observation sampling" as the generation of a new dataset from the sensors' observations. Observation sampling follows a random process; to achieve average results, we repeatedly run the simulation, generating new observation samplings each time. When comparing different methods, they are evaluated using the same set of observation samplings to ensure consistency.

An example of observation sampling is shown in Fig. 7, which documents the total presence of 88 sensors. In Fig. 7(a), sensor observations between $t = 400$ and $t = 600$ are displayed, where 25 sensors are active during this interval. A more focused view between $t = 400$ and $t = 420$ shows 7 sensors present. The sampling illustrates significant variations induced by the use of random processes, particularly highlighted through the observations marked with solid squares in yellow and black.

In Fig. 7(a), the sensor depicted in black with solid squares is active only from $t = 413$ to $t = 428$, while the sensor shown in yellow with solid squares remains active from $t = 410$ to $t = 710$. On average, a sensor remains in the environment for a duration of $\frac{1}{\mu} = 100$, which often results in many sensor pairs having no overlapping periods of operation.

Additionally, as shown in Fig. 7(b), sensor messages are not transmitted at regular intervals—for instance, among the 15 messages shown in black, there is less than 0.01 seconds between two messages at $t = 413$ and more than 1.6 seconds between $t = 415$ and $t = 417$. This irregularity reduces the effectiveness of distance metrics based on point-to-point calculations.

The messages are also quite noisy, with values reaching 1.5 at $t = 412$ for the yellow square messages, even though the maximum value of the phenomenon is 1. This level of noise makes distance metrics that evaluate maximum deviations or trends (such as increases or decreases between consecutive points) unreliable.

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 observation sampling, 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 observation sampling, we fix the parameters n, s, and r that will be the same for all interpolations to make.

For one observation sampling, 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 observation sampling as the median value.

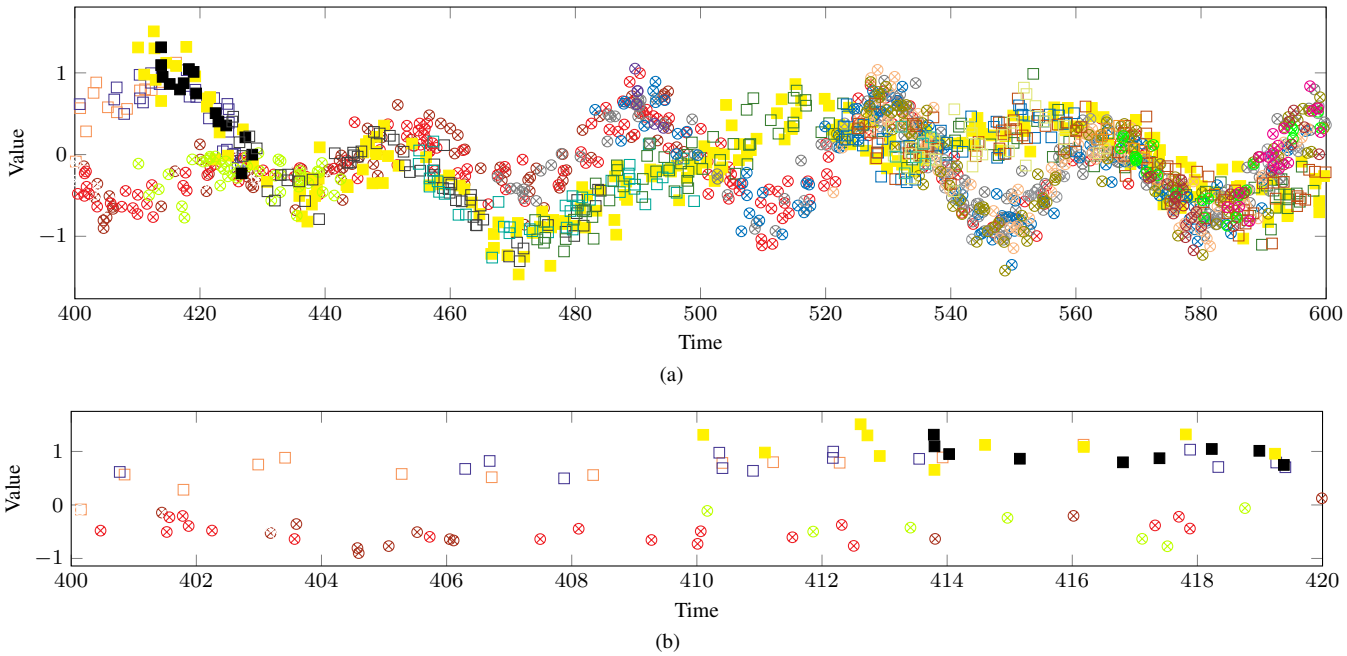

Fig. 7. One observation sampling with a noise level of $\sigma = 0.2$, (a) targets the time interval between $t = 400$ and $t = 600$, and (b) focuses on the shorter span from $t = 400$ to $t = 420$. Each sensor's observations are color-coded for distinction. Sensors tracking phenomenon 1 are represented by square markers, while those monitoring phenomenon 2 are indicated by circles. Solid squares colored in yellow and black, unlike others that have hollow markers, are selected as examples to demonstrate the variabilities encountered in the simulation.

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:

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

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 aim to create groups of similar sensors, the 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 $|T_i|+|T_j| = n_i+n_j$. 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 comparison between our approach and the one proposed in the literature on a common ground, we keep the main clustering algorithm, and change their features. Specifically, we decide to adapt this principle to the AHC 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:

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

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.

For our settings, the ideal number of clusters is 2, 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.

- Occasionally, a phenomenon might not be monitored by any sensors at a specific point in the simulation, resulting in the sensors before and after this point being grouped separately due to the absence of a common definition interval. In such cases, ideally, three clusters would better represent the scenario—two for the disjoint periods of the same phenomenon and one for the other phenomenon.
- When the overlap in the monitoring interval between sensors is minimal, and the phenomena themselves are overlapping, a pair of sensors monitoring different phenomena might end up with a very low measured distance between them. Consequently, these sensors might be mistakenly grouped together. It becomes crucial to isolate such pairs to avoid incorrect clustering, potentially necessitating an additional cluster.
- High noise levels or sensors that transmit very few observations over brief intervals can lead to significantly different readings compared to other sensors. Our clustering method sometimes inadvertently groups sensors tracking different phenomena and leaves such an outlier sensor isolated in its own 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 observation samplings. The average performance along

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**.

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%).

The use of our similarity metric significantly enhances performance, with its application alongside the complete linkage method proving to be the second-most effective configuration in terms of Homogeneity and Completeness. The key advantage of our similarity approach is its comprehensive consideration of all messages sent by a sensor. Unlike traditional time series-based methods like Dynamic Time Warping (DTW), which focus only on the sequence of messages, our interpolation-based method incorporates the dimension of duration. It's important to note, however, that while the interpolation method is robust against irregular and noisy sensor data, it is a parametric method that performs best when the phenomena being monitored do not undergo abrupt changes. Additionally, we have selected the Kriging interpolation method, which is particularly effective at interpolating functions that are combinations of sinusoids.

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

This section evaluates the robustness of our solution against variations in measurement noise.

The stopping criterion plays a crucial role in determining measurement performance. Consequently, we have implemented 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:

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

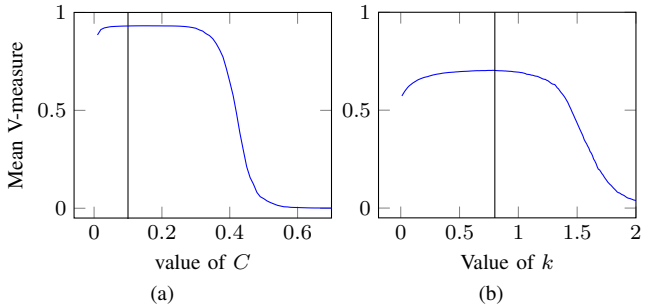

Fig. 8. (a) Average V-measure performances with zero noise, while varying the threshold distance parameter C.
(b) Average V-measure performances with noise $\sigma = 0.2$, while varying the threshold distance parameter k, and with $C = 0.1$.
The vertical line represents the chosen parameter value for C and k, further used.

Setting the Threshold Parameters: We perform simulations to determine the appropriate values for parameters C and k. Initially, to set C in Eq. (5)—the threshold for zero noise—we conduct 1000 observation samplings under the condition of zero noise $\sigma = 0$. We evaluate the clustering performance for various C values, applying our clustering method with the stopping criterion defined in Eq. (5), and assess each simulation's performance by examining the V-measure. The results, showing the mean V-measures in Fig. 8(a), indicate that performance is relatively stable for C values within the range $[0.05, 0.28]$, producing V-measure values between $[0.925, 0.932]$. Based on these findings, we choose $C = 0.1$.

After setting C, we determine k using the same process, this time with noise set at $\sigma = 0.2$ and $C = 0.1$, performing

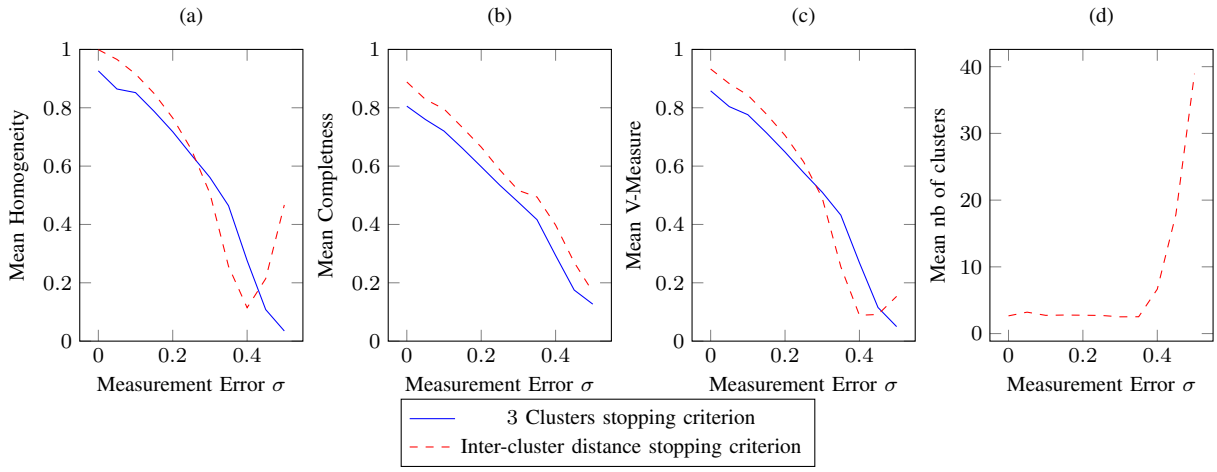

Fig. 9. Performance evaluation of our clustering solution (mean interpolation difference + weighted mean linkage method for AHC) under various levels of measurement noise, with a stopping criterion based on maximum number of sensors (solid blue line) and one based on threshold inter-cluster distance (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.

1000 observation samplings and evaluating the thresholding method's performance for different k values. The V-measure results, presented in Fig. 8(b), suggest a performance plateau for k within $[0.5, 1]$, with scores between $[0.694, 0.702]$. We select $k = 0.8$.

These observations confirm that a broad range of values for C and k yield consistently effective clustering performance.

2) Evaluation of the Clustering Performance for Different Noises: We evaluate the robustness of our clustering method by analyzing its response to varying levels of measurement noise. Our methodology integrates a similarity metric derived from the average amplitude difference between kriging interpolations, along with hierarchical clustering that employs a mean linkage method weighted by the duration of common intervals. We assess the performance of this configuration using two distinct stopping criteria for the clustering: limiting the number of clusters to a maximum of three, and using a maximum inter-cluster distance threshold set at $D_{\text{threshold}} = 0.1 + 0.8\sigma$.

Considering that phenomena values range between -1 and 1, we conduct 1000 observation samplings for each noise level $\sigma = \{0 + i * 0.05, 0 \leq i \leq 10\}$. The evaluation results are illustrated in Fig. 9, where we display metrics such as Homogeneity (a), Completeness (b), and V-measure (c). Additionally, for the method that employs the stopping criterion based on the inter-cluster distance, we document the average number of clusters formed at each noise level in Fig. 9(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 $\sigma = 0.5$). This surge in cluster numbers might appear misleading because it does not correspondingly impact completeness as expected. In the simulations, there is a fluctuation between very low cluster counts (where a single cluster yields perfect completeness) and very high cluster counts (resulting in lower completeness), leading to poorer overall V-measure performance for the distance-based threshold.

VII. Conclusion and Perspectives

In this study, we introduced a novel similarity metric designed to assess transmissions from sensors that are noisy and irregular. Additionally, we developed a clustering method capable of grouping sensors present in the environment for a limited duration, often encountering situations where sensor pairs do not overlap in their periods of operation. We highlighted the effectiveness of our similarity metric and clustering approach in comparison to proposals from the literature, and conducted a robustness study to evaluate the ability of our method to accurately identify groups of sensors under varying levels of measurement noise.

The assumptions made in this paper facilitate the implementation of more flexible IoT solutions, which is crucial given the rapidly increasing number of connected devices.

This research lays the groundwork for future advancements in real-time monitoring and analysis of observed phenomena, potentially aiding in the development of a digital twin.

Future research could explore the next steps for using and expanding these clustering methods. Integrating these strategies into real-world applications or use cases for anomaly detection or message reduction would further highlight the value of this work.

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.