Distributed Clustering Strategies in Industrial Wireless Sensor Networks

Angelo Cenedese, *Member, IEEE*, Michele Luvisotto, *Student Member, IEEE,* and Giulia Michieletto

***Abstract*—Wireless sensor networks (WSNs) can provide numerous benefits in industrial automation. By removing the cable infrastructure, the wireless architecture enables the possibility for nodes in a network to dynamically and autonomously group into clusters according to the communication features and the data they collect. This capability allows to leverage the flexibility and robustness of industrial WSNs in supervisory intelligent systems for high level tasks such as, for example, environmental sensing, condition monitoring and process automation. In this paper, a clustering strategy is studied that partitions a sensor network into a non–fixed number of non–overlapping clusters according to communication network topology and measurements distribution: to this aim, both a centralized and a distributed algorithm are designed that do not require a cluster–head structure or other network assumptions. As a validation, these strategies are tested on a real dataset coming from a structured environment and the effectiveness of the clustering procedure is also investigated to perform anomalies detection in an industrial production process.**

***Index Terms*—Industrial Wireless Sensor Networks (IWSNs), network clustering, distributed algorithms, anomaly detection, environment monitoring.**

I. Introduction

In the competitive industrial marketplace, Industrial Wireless Sensor Networks (IWSNs) have emerged as a key technology to improve the efficiency of production processes while limiting implementation costs. Recent developments have led to the realization of tiny and low–cost sensor nodes equipped with data processing and communication capabilities and IWSNs incorporate networks of up to thousands of these autonomous devices to facilitate the realization of highly reliable and self–healing intelligent systems for heterogeneous applications in the industrial context [1], [2], [3].

The collaborative nature, rapid deployment and flexibility of IWSNs bring several advantages over traditional wired industrial systems. Specifically, the lack of cables, the support for mobility and the low power maintenance make these solutions suitable for harsh environmental conditions [4]. The existing and potential applications of IWSNs can be classified according to the taxonomy in [5] into three main categories, namely *environmental sensing*, *condition monitoring*, and *process automation*, and they regard a wide and heterogeneous range of specific scenarios that include building automation [6], process control [7], utility automation [8], precision agriculture [9], to cite a few.

In all these contexts, there is typically an intelligent unit, be it centralized or distributed, that realizes the principal task through monitoring and control actions, thus requiring real–time information delivery over very large–scale systems. In this sense, locality in the communication can, on the one hand, ensure a rapid spreading of the information and, on the other, reduce the data interpretation complexity by filtering out the unnecessary information that regards non neighboring nodes. Hence, grouping nodes into local clusters arises as a fundamental tool to enhance the network self-organization capabilities and improve the system autonomicity towards the fulfillment of collective goals.

Besides real–time performance, IWSNs differ from other wireless sensor networks because of an increased attention towards maintainability, reliability, and safety [1], [4]. In the industrial environment the networks should be able to guarantee robust operations in often critical scenarios and to ensure the safety of personnel, machinery and propriety as well as fast detection and recovery from malicious external attacks. Moreover, IWSNs should operate autonomously for such process and service monitoring. To this aim, fault detection algorithms must be developed that are able to identify sensors and actuators whose operating conditions are different from those expected [10]. In this venue, network partitioning strategies able to group nodes that exhibit similar behaviors can serve as a useful tool.

The development of effective clustering strategies specifically tailored for the industrial environment is hence a key research area towards the realization of self–organizing, real–time, robust and secure wireless sensor networks to be deployed in industrial applications [11], [12], [13], [14].

A. Related works

The literature on clustering in sensor networks is quite vast and heterogeneous: it involves different concepts of *clustering* and covers several disciplines. It is therefore difficult to draw an even fairly complete state of the art: in this respect and with no aim of being exhaustive, a brief overview of clustering results is presented in the following.

As a general distinction, clustering problem in a wireless sensor network can be tackled by considering the topology (*network decomposition*) or the data gathered from the environment (*data clustering*). The first approach is generally

This work was supported in part by the Italian MIUR Project SEAL - Smart&safe Energy-aware Assisted Living (SCN-00398).

The authors are with the Department of Information Engineering, University of Padova, Via Gradenigo 6/B, 35131, Padova, Italy, e-mail: {angelo.cenedese, giulia.michieletto}@unipd.it, michele.luvisotto@dei.unipd.it.

considered in the design of communication algorithms and protocols, where the formation of clusters can lead to higher energy efficiency and reduced communication delays [15]. On the other hand, data–based network partitioning allows to reduce the computational load by taking into account similarities among nodes measurements in applications that deal with large amounts of information [16].

Several partitioning algorithms have been proposed to accomplish the clustering task. In particular, the network decomposition problem can be tackled via well known procedures, which can be divided into heuristic, weighted, hierarchical and grid schemes. A comprehensive overview can be found in [17] and the references therein. Differently, in data clustering the focus is moved from the topology of the information sources to the information flow itself. Most techniques are based on the spatio–temporal correlation properties of the data gathered by the sensors [18]. In all these works, one solution appears as particularly popular, in which nodes are partitioned into clusters and some of them, called *cluster-heads* (CHs), take the crucial rôle of aggregating the data coming from the other elements in the cluster and forwarding them to the base station.

The interest around clustering strategies for general purpose wireless sensor networks is well established; conversely, only recently the application of these techniques to IWSNs has been considered. The work in [11], for example, presents two clustering approaches to realize tracking of mobile nodes in IWSNs, a static one where clusters have a predefined number of elements and a dynamic one, which achieves the task more efficiently. The application of clustering strategies to solve environmental sensing and conditional monitoring tasks is the subject of [12], where a real–time clustering algorithm is applied to an industrial monitoring network to achieve risk assessment in an energy–efficient way. The detection of fault and anomalies in industrial sensing or process applications is another problem that can be tackled via clustering strategies. The work in [13] first adopts a well known network decomposition algorithm, Low Energy Adaptive Clustering Hierarchy (LEACH) [17], then implements a fault detection algorithm that exploits the presence of clusters. The authors of [14], instead, develop an original clustering algorithm (Distributed Matching–based Grouping Algorithm, DMGA) to partition the network into strongly correlated groups of at least a predefined number of nodes; on this basis, then, a General Anomaly Detection (GAD) distributed procedure is developed that exploits data correlation for real–time recognition of anomaly conditions. Again, all these approaches rely on the presence of CHs to facilitate the clustering task.

B. Contribution of the Paper

This paper deals with clustering strategies in IWSNs. Differently from other approaches proposed in the literature, network decomposition and data clustering are here considered together, since both these aspects are crucial in ensuring the performance standards of industrial communications. Moreover, the proposed approach is fully distributed and does not rely at all on the presence of CHs, yielding considerable benefits in terms of scalability and robustness.

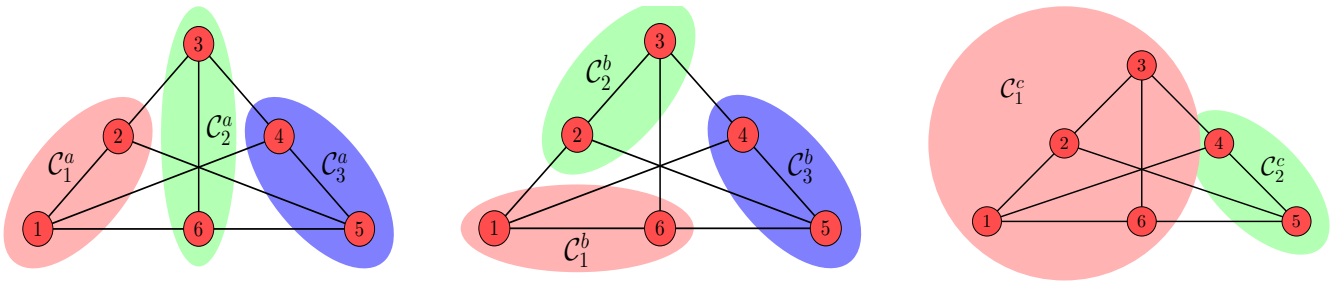
The first contribution of this paper is to provide a formal definition of the clustering task over a given IWSN, stated as the determination of the unique partition that simultaneously abides by three different criteria. Firstly, for each pair of nodes in a cluster there must exist at least one (communication) path connecting them composed exclusively by elements included in the same cluster (*connectivity criterion*). Secondly, for each node in a cluster there must exist at least another one in the same cluster such that the two nodes measurements are similar according to a defined metric (*measurements similarity criterion*). Finally, the network must be partitioned into the minimal number of non–overlapping clusters (*maximality criterion*).

The most important and original contribution of this work is to provide algorithms which solve the aforementioned clustering task. Specifically, the partitioning problem is solved in the first instance through a centralized policy and then the distributed paradigm is considered to provide an algorithm capable of partitioning the network only by local exploration of measurements. In both cases, no CHs assignment is considered, and no assumptions are made on the structure of the emerging clusters (other than those implied by the partitioning criteria), which are uniquely determined by the proposed algorithms. The convergence of the distributed solution to the centralized one is showed through a numeric example and the efficiency of the proposed approach is confirmed by comparing it with other clustering techniques, i.e., a classical k-means strategy [19] and the most recent DMGA algorithm [14]. Finally, the applicability of the proposed procedure to real industrial use cases is demonstrated by experimental and simulated scenarios. As a first example, the distributed clustering algorithm is employed to perform anomaly detection in a simulated industrial production process. Subsequently, the performance of a fault detection algorithm based on this procedure are evaluated on an environmental sensing dataset and compared with those offered by the state–of–the–art GAD algorithm, that serves the same purpose.

The remainder of the paper is organized as follows. Section II formally describes the task to achieve. Section III introduces both the centralized and the distributed algorithms to solve the clustering problem, with a first validation against k-means and DMGA. Then, Section IV validates the proposed distributed clustering strategy in real–world industrial use cases. Finally, Section V presents the main conclusions of this work and some future directions of research.

II. Clustering in an IWSN: Definition and Notation

The clustering procedure proposed in this paper is defined over an IWSN composed by N nodes, whose topology is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (vertexes) and $\mathcal{E}$ is the set of communication links (edges) among them. In this framework, each node v_i is associated with a measurement m_i, gathered from the environment and stacked into the measurement vector $\mathbf{m} = \begin{bmatrix} m_1 & \dots & m_N \end{bmatrix}^\top$.



(a) Partition $\mathcal{C}^a$ with 3 clusters (C1-C2)

(b) Partition $\mathcal{C}^b$ with 3 clusters (C1-C2)

(c) Partition $\mathcal{C}^c$ with 2 clusters (C1-C2-C3)

Fig. 1. Example 1: clustering on a 6–nodes network. The vector measurement for nodes $\{v_1, \dots, v_6\}$ is $m = [10 \quad 12 \quad 13 \quad 20 \quad 22 \quad 11]^\top$.

Given $\mathcal{G}$ and $\mathbf{m}$, the clustering task consists in identifying the node clusters $\{\mathcal{C}_\ell\}$ that constitute the unique partition of $\mathcal{V}$, defined as $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_k\}$ satisfying the following criteria:

C1. **Connectivity:** $\forall\ \mathcal{C}_\ell \in \mathcal{C},\ \forall\ v_i, v_j \in \mathcal{C}_\ell\ \exists$ *path* $p = \{v_1, \dots, v_h, \dots, v_n\}$ such that $v_1 = v_i$ and $v_n = v_j$, $(v_h, v_{h+1}) \in \mathcal{E}$ and $v_h \in \mathcal{C}_\ell\ \forall\ h \in [1, n-1]$.

C2. **Measurements similarity:** $\forall\ \mathcal{C}_\ell \in \mathcal{C},\ \forall\ v_i \in \mathcal{C}_\ell\ \exists\ v_j \in \mathcal{C}_\ell$ such that $\|m_i - m_j\| \leq b$, according to some norm; $b \in \mathbb{R}$ is named as *clustering bound*.

C3. **Maximality:** let $\mathcal{C}^{(i)} = \{\mathcal{C}_1^{(i)}, \dots, \mathcal{C}_{k_i}^{(i)}\},\ i \in \mathbb{N}$ be a generic partition of the network satisfying criteria C1–C2, the maximal partition is $\mathcal{C}^\star = \{\mathcal{C}_1, \dots, \mathcal{C}_{k^\star}\}$, where

$$k^\star = \arg\min_{i \in \mathbb{N}} [k_i]$$

The elements of the obtained partition $\mathcal{C}_1, \dots, \mathcal{C}_{k^\star}$ are denoted as *optimal clusters* and the function $\mathcal{F} : \mathcal{V} \to \mathcal{C}^\star$ is introduced that maps each node to the optimal cluster it belongs to in the optimal partition.

Remark 1. *(Optimal) clusters are non–overlapping and cover the entire network, i.e., any node in $\mathcal{V}$ belongs to one and only one cluster following C1-C2-C3. This results in the function $\mathcal{F}$ to be bijective.*

Looking at the three criteria, criterion C1 requires that each cluster forms a connected subgraph: in practice, in industrial applications, this means that the measurements and more in general the data obtained from nodes within the same cluster are shared. Criterion C2, instead, states that a sort of similarity exists among the measurements of nodes in a cluster: from the application point of view, the *clustering bound* b is a setup parameter related to the expected variance in the measurements range. Finally, since there can be several network partitions which fulfill C1 and C2, condition C3 is introduced to select the partition composed by the minimum number of clusters, i.e. the one wherein the cardinality of each cluster is maximal. This ensures that the defined partition is unique, apart from pathological cases.

Example 1. *In order to provide an example of the presented clustering task and to show the uniqueness of the defined partition, a network composed by 6 nodes is considered in Fig. 1. The following (scalar) measurement vector is assumed,* $\mathbf{m} = \begin{bmatrix}10 & 12 & 13 & 20 & 22 & 11\end{bmatrix}^\top$*, with a clustering bound chosen as $b = 2$, which corresponds to the admitted standard deviation for a* correct *measurement. Figs. 1a–1b show two partitions of the network, $\mathcal{C}^a$ and $\mathcal{C}^b$, both composed by three clusters, that satisfy criteria C1 and C2. Nevertheless, a further partition $\mathcal{C}^c$ with only two clusters can be identified, as reported in Fig. 1c, which also abides by C1 and C2, and maximizes the cardinality of the clusters (C3).*

Interestingly, with respect to C1 and C2 only, all these partitions show a cluster that includes exclusively nodes v_4 and v_5 due to the similarity between their measurements and the dissimilarity with the other nodes values; conversely, the remaining nodes show a higher level of measurement similarity among them and can be grouped according to different connectivity graphs that are identified in the network. From a building intelligence perspective, this suggests the possibility of detecting and isolating faulty nodes (in this case v_4 and v_5) or anomalous events through the clustering procedure [20].

III. Clustering Algorithms

In this section two clustering algorithms are presented: the former is designed through a centralized approach, while the latter is achieved according to the distributed paradigm. Both strategies converge to the same solution providing an optimal network partition with respect to the established criteria.

A. Centralized Clustering Algorithm (CCA)

In the framework introduced in Sec. II, it is possible to solve the clustering task through the centralized procedure reported in Alg. 1. The inputs of the algorithm are the measurement vector $\mathbf{m}$, the clustering bound b and the (symmetric) adjacency matrix $\mathbf{E} \in \mathbb{R}^{N \times N}$ of the network, which provides information about the communication links derived from $\mathcal{E}$, i.e., $(v_i, v_j) \in \mathcal{E} \Leftrightarrow \mathbf{E}[i, j] = 1$. The outcome of the algorithm is the set $\{\mathcal{C}(v_i)\}_{i=1}^N$ which is related to the optimal partition $\mathcal{C}^\star$ defined in Sec. II, through the neighborhood of each node v_i, $\mathcal{N}_i = \{v_j \in \mathcal{V} \mid \mathbf{E}[i, j] = 1\}$, according to the relation $\mathcal{C}(v_i) = \mathcal{N}_i \cap \mathcal{F}(v_i)$.

The proposed solution relies on the dynamic update of the terms B_i^ℓ and B_i^u that indicate a lower and upper bound, respectively, associated to each node v_i. The initial part of the algorithm is devoted to the setup phase (rows 2-5): for each node v_i, $\mathcal{C}(v_i) = \{v_i\}$, while the two bounds B_i^ℓ and B_i^u are defined basing on the initial node measurement m_i and

the imposed clustering bound b. The remaining part of Alg. 1 consists of two steps that are repeated iteratively.

1) *Inclusion of nodes in clusters* (rows 8-16): for each node v_i, all the neighbors not already included in $\mathcal{C}(v_i)$, i.e., $v_j \notin \mathcal{C}(v_i)$ such that $\mathbf{E}[i,j] = 1$, are explored. Neighbor v_j is added to cluster $\mathcal{C}(v_i)$ only if the *measurements similarity criterion* (C2) is fulfilled.
2) *Update of bounds* (rows 17-32): in the second step, for each cluster $\mathcal{C}(v_i)$ whose cardinality is larger than 1, two quantities are computed, namely the minimum lower bound B^{ℓ}_{min} and maximum upper bound B^{u}_{max}

$$B^{\ell}_{min} = \min_{v_k \in \mathcal{C}(v_i)} B^{\ell}_k, \qquad B^{u}_{max} = \max_{v_k \in \mathcal{C}(v_i)} B^{u}_k.$$

Then, every node in the cluster updates its bounds accordingly, i.e., $B^{\ell}_k = B^{\ell}_{min}$, $B^{u}_k = B^{u}_{max}$ $\forall\, v_k \in \mathcal{C}(v_i)$.

The procedure ends when the second step does not produce any update of node bounds and, therefore, the cardinality of each cluster coincides with that of the previous iteration step.

Remark 2. *It can be observed that by adopting CCA, criteria C1, C2 and C3 are actually fulfilled. Indeed, a generic node v_j is inserted in cluster $\mathcal{C}(v_i)$ through the instructions in rows 10-14. Specifically, this happens only if v_j is the neighbor of another node v_h in the same cluster, i.e., $\mathbf{E}[j,h] = 1$, $v_h \in \mathcal{C}(v_i)$. At the same time, it is ensured that v_j is inserted in the cluster $\mathcal{C}(v_i)$ only if $\exists\, v_h \in \mathcal{C}(v_i)$, s.t. $B^{\ell}_h \leq m_j \leq B^{u}_h$, thus satisfying C2. Finally, Alg. 1 stops only when there are no more bounds update ensuring that the corresponding partition has the lowest possible number of clusters, fulfilling C3.*

It can be proved that the proposed procedure has complexity $O(N^3)$. Indeed, the instructions inside the *while* loop require $O(N^2)$ computations, due to the two nested *for* loops, whereas the *while loop* is executed N times in the worst case. This situation happens when the sensors form a line graph, i.e. $\mathbf{E}[i,j] = 1 \Leftrightarrow |i-j| \leq 1$, and are provided with evenly spaced measurements at distance b, i.e., $m_i = (i-1)\cdot b$. In this scenario, the upper bound of node v_1 (which is the last one to converge together with v_N) at the k-th iteration is given by $B^{u}_1 = (k+1)\cdot b$ and reaches the maximum value of $N \cdot b$ only at iteration $k = N-1$. The algorithm hence converges only at the subsequent iteration, thus yielding an overall $O(N^3)$ complexity. However, it should be noted that this worst–case scenario is quite uncommon to find in practice and therefore the complexity is generally lower.

Algorithm 1 CCA

```
1:  procedure CCA(m,b,E)
2:      term ← false
3:      C(v_i) ← {v_i} ∀ i
4:      B^ℓ_i ← m_i − b ∀ i
5:      B^u_i ← m_i + b ∀ i
6:      while not term do
7:          update ← false
8:          for i ← 1 to N do
9:              for j ← 1 to N do
10:                 if E[i,j] = 1 and v_j ∉ C(v_i) then
11:                     if ∃ v_h ∈ C(v_i) : B^ℓ_h ≤ m_j ≤ B^u_h then
12:                         C(v_i) ← v_j
13:                     end if
14:                 end if
15:             end for
16:         end for
17:         for i ← 1 to N do
18:             if |C(v_i)| > 1 then
19:                 B^ℓ_min ← min_{v_k ∈ C(v_i)} B^ℓ_k
20:                 B^u_max ← max_{v_k ∈ C(v_i)} B^u_k
21:                 for each v_j ∈ C(v_i) do
22:                     if B^ℓ_j > B^ℓ_min then
23:                         B^ℓ_j ← B^ℓ_min
24:                         update ← true
25:                     end if
26:                     if B^u_j < B^u_max then
27:                         B^u_j ← B^u_max
28:                         update ← true
29:                     end if
30:                 end for
31:             end if
32:         end for
33:         term ← not update
34:     end while
35: end procedure
```

B. Distributed Clustering Algorithm (DCA)

The centralized approach of CCA is based on the assumption that all nodes measurements are available at the same time together with the network topology at a single location. Although this statement might be true for some network configurations, it is not verified in a generic IWSN, and a distributed paradigm that relies only on local communication exchange is generally preferable. Moreover, the decentralized strategy results to be more robust to node failures and dynamic network topology modifications, which is a valuable feature in an industrial environment characterized by noise sources that may impair communication and faults due to the application to critical operational scenarios.

The proposed distributed algorithm (DCA) is reported in Alg. 2 where the iterative nature of the procedure regards the execution of the same instructions by each node v_i of the network until the unique partition $\mathcal{C}^{\star}$ that fulfills criteria C1, C2 and C3 is determined. To this aim, a label c_i is associated to each node v_i to specify the cluster to which it belongs to (at the beginning $c_i = i$). This variable is updated during the algorithm execution, so that the output of the whole procedure is a set of labels, one for each node, describing the partition of the network, in the sense that $c_i = c_j \Leftrightarrow \mathcal{F}(v_i) = \mathcal{F}(v_j)$. As in the centralized solution, a node v_i is also associated to a lower and an upper bound, B^{ℓ}_i and B^{u}_i respectively, that are initialized as in Alg. 1.

DCA is again based on an iterative exploration of the neighbors, performed by each node in a distributed manner. In detail, node v_i checks the measurement of each node v_j belonging to its neighborhood $\mathcal{N}_i$ under the constraint $c_j \neq c_i$, meaning that they do not belong to the same cluster. If the two measurements are similar, according to criterion C2, then both the labels c_i and c_j are set to $\min(c_i, c_j)$ (rows 6-8). Moreover,

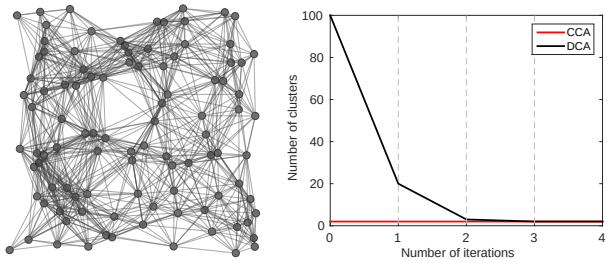
Algorithm 2 DCA

1: **procedure** DCA(m_i,b,$\mathcal{N}(i)$)
2: $active(v_i) \leftarrow false$
3: $B^{\ell}_{min} \leftarrow \min_{v_h:c_h=c_i} B^{\ell}_h$
4: $B^{u}_{max} \leftarrow \max_{v_h:c_h=c_i} B^{u}_h$
5: **for each** $v_j \in \mathcal{N}(i),\ c_j \neq c_i$ **do**
6: **if** $B^{\ell}_i \leq m_j \leq B^{u}_i$ **then**
7: $c_j \leftarrow min\,(c_i, c_j)$
8: $c_i \leftarrow c_j$
9: **if** $B^{\ell}_{min} > B^{\ell}_j$ **then**
10: $B^{\ell}_{min} \leftarrow B^{\ell}_j$
11: **end if**
12: **if** $B^{u}_{max} < B^{u}_j$ **then**
13: $B^{u}_{max} \leftarrow B^{u}_j$
14: **end if**
15: $active(v_i) \leftarrow true$
16: $active(v_j) \leftarrow true$
17: **end if**
18: **end for**
19: **if** $B^{\ell}_{min} < B^{\ell}_i$ **or** $B^{u}_{max} > B^{u}_i$ **then**
20: $B^{\ell}_i \leftarrow B^{\ell}_{min}$
21: $B^{u}_i \leftarrow B^{u}_{max}$
22: $active(v_i) \leftarrow true$
23: **end if**
24: **for each** $j \in \mathcal{N}(i),\ c_j = c_i$ **do**
25: **if** $B^{\ell}_{min} < B^{\ell}_j$ **then**
26: $B^{\ell}_j \leftarrow B^{\ell}_{min}$
27: $active(v_j) \leftarrow true$
28: **end if**
29: **if** $B^{u}_{max} > B^{u}_j$ **then**
30: $B^{u}_j \leftarrow B^{u}_{max}$
31: $active(v_j) \leftarrow true$
32: **end if**
33: **end for**
34: **end procedure**

node v_i keeps track of the minimum and maximum values assumed by B^{ℓ}_k and B^{u}_k respectively (rows 10-15) for any compatible neighbor v_k, in order to update its own bounds at the end of neighbors exploration (rows 19-23). Subsequently, a further exploration of all the neighbors of v_i that belong to its same cluster, i.e. $c_j = c_i$, is performed to update their bounds accordingly (rows 24-33). The node v_i stops to perform the iterative execution of this algorithm when its flag *active* becomes false. This label is initialized to false at the beginning of the procedure and set to true in three possible cases: a new compatible neighbor is found, the node is included in another cluster or its bounds are updated.

The rationale behind DCA resides in the iterative bounds update and nodes inclusion into clusters, similarly to that of CCA (Alg. 1); remarkably, though, DCA exploits only local information to attain the network partition and it is executed locally by each node, without requiring a central controller. Notably, for a given network and set of measurements, the two algorithms produce the same partition, the one defined in Sec II, as confirmed by extensive numerical simulations.

Example 2. *Consider a synthetic IWSN composed of* $N = 100$ *nodes whose communication graph is described by a random geometric graph [21], whereas measurements are randomly drawn from a uniform distribution with range between* 0 *and* 100, $m_i \in \mathcal{U}([0, 100])$, *and the clustering bound is set to* $b = 5$. *One realization of such scenario is reported in Fig. 2 where both the network topology and the convergence behavior of DCA in terms of number of clusters are shown. The initialization of DCA corresponds to the creation of one cluster per node and the iterative procedure makes the number of clusters decrease as nodes are merged together, according to the update of the node bounds. In this context, a step is defined as the full execution of Alg. 2 by every node in the network. As expected, the result of the distributed procedure converges to the centralized solution after three steps. Specifically, in the reported case, the network optimal partition* $\mathcal{C}^{\star}$ *is constituted by* 2 *clusters.*



(a) (b)

Fig. 2. Example 2: (a) topology of the IWSN as a random geometric graph; (b) convergence of the DCA solution to the CCA one.

C. First Assessment of DCA

Here, the performance of the distributed partitioning strategy DCA is compared with two other procedures: the k–means clustering approach (hereafter, KMC) and the DMGA strategy, described in Sec. IV-A of [14].

The evaluation is conducted on a network made of N nodes ranging from 8 to 100, organized into N_C clusters of fixed size equal to 4 (hence, N_C spans from 2 to 25). The measurement of a generic node in the i–th cluster is uniformly distributed in the range $[T_i - b/2, T_i + b/2]$, where T_i is the cluster average measurement and it is generated as $T_i = T_0 + (i-1)\cdot 2b,\ i = 1, \ldots, N_C$ (b is the clustering bound). Two different network topologies are considered: in one case, a random geometric graph is chosen within each cluster and any two clusters are connected by one link between two CH nodes with probability $p_L = 0.9$ (Fig. 3a); in the other case, instead, the communication network is given as a complete graph (Fig. 3b). In both cases, the number of misclassified nodes is considered versus the network size. Indicating with $\mathcal{C}^{\star}$ the optimal partition yielded by CCA and with $\mathcal{C}$ a generic partition, the misclassification cost function is given by

$$d\,(\mathcal{C}, \mathcal{C}^{\star}) = \sum_{i=1}^{N} \chi_i,$$

where χ_i is a function that is equal to zero if the set of cluster elements of node v_i in partitions $\mathcal{C}$ and $\mathcal{C}^{\star}$ coincides (node v_i is correctly classified) and it is equal to one otherwise (node v_i is misclassified).

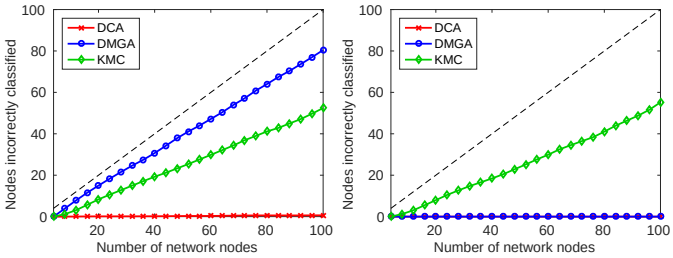


Fig. 3. Node classification comparison: (a) not fully connected network topology; (b) fully connected network topology.

For KMC, the parameter k is set equal to N_C and the centroids are randomly inizialized. For DMGA, instead, the correlation coefficient c_{ij} is defined as the ratio between the distance of two measurements and the measurements range, i.e.,

$$c_{ij} = 1 - \frac{|m_i - m_j|}{b\,(2 \cdot N_C - 1)}.$$

Also, the DMGA parameters have been chosen equal to $N_{min} = 4$ and $c_{min} = 1 - (2 \cdot N_C - 1)^{-1}$.

Fig. 3 shows how the performance of KMC are totally independent of the network topology as such strategy is exclusively based on the similarity of the measurements and does not take into account the communication links. Specifically, the fact that the algorithm neglects the structure of the network and the consequent totally random initialization of centroids cause KMC to get stuck on local optima, which results in the significant percentage of misclassified nodes observable in Fig. 3. On the contrary, applying the DMGA procedure, the set of nodes that are incorrectly classified is empty when the network is modeled through a complete graph, while a significant number of nodes is not classified appropriately when a less connected communication structure is adopted. Instead, and most remarkably, the partition provided by DCA algorithm coincides with the optimal one in both cases, thanks to the fact that the proposed procedure considers at the same time both the IWSN connectivity properties and the measurements similarity. Moreover, as explained in Secs. III-A and III-B, CCA and DCA always yield the same clustering solution, which is equal to the optimal one defined in Sec. II.

IV. Application to the Industrial Scenario

Many practical applications of the clustering procedure can be envisaged within the context of IWSNs. They range from the fault detection along a production line to the monitoring of a structured environment in building or factory automation, from the tracking of mobile nodes in a productive industrial area to the optimization of energy resources for a surveillance system. In this sense, two different real world scenarios have been considered for validation of the proposed DCA, specifically:

#1 a factory process line, where an item undergoes several production stages on its way from raw material to end product;

#2 a structured indoor environment, wherein the task is that of indoor monitoring, since building energy management issues may arise and anomaly detection is also important.

Scenario #1 concerns condition and monitoring of highly dynamic processes in modern factory facilities, where the timing behaviors of the control variables and of the quantities of interest need to be accurately monitored in order to ensure efficiency, performance and quality to the process/service. A factory intelligence unit should manage to follow the product/process chain, to identify the different stages, and to detect possible faults and anomalies that may occur.

Indeed, this issue can be experienced in a large variety of production plants and processes. Just to provide a couple of examples: in the context of the food industry, the traceability of the product and the proper management of the ambient conditions throughout the whole supply chain are of paramount importance to guarantee the quality and safety of food products and to extend their shelf life [22]. In semiconductor manufacturing, the development of intelligent monitoring systems based on IWSN solutions can increase the automation and maintainability of such complex process and equipments, thus leading to real–time problem diagnostics and production optimization procedures [23].

Scenario #2 is characteristic of many industrial, commercial, and public service installations, and refers to environmental sensing and service monitoring. In particular, service monitoring aims at offering to the end–users a designed (or expected) quality of service, and proposing to the providers a tool to control and optimize the use of resources and increment the awareness of their employment. These instances are strictly related to building automation, which has received a surge of attention in the last few years towards the deployment of green building solutions in the industry [5], [8]. Environmental sensing, instead, is referred to the task of measuring quantities that can be only partially controlled but are fundamental for the efficiency of equipment and operators, to detect pollution, avoid hazard and ensure security, and also yield comfort in the workspace [18].

In detail, the focus in this section is posed on the validation of the distributed strategy DCA, since it has already been shown to converge to the centralized solution, and it represents the most interesting approach for practical applications due to its intrinsic robustness and flexibility. Thus, the performance of the proposed partitioning procedure is studied with respect to different kind of faults in the first scenario, while it is compared with those of KMC and of DMGA/GAD algorithms in the second case.

A. Numerical Validation: Process Monitoring Application

This scenario is schematically represented in Fig. 4, where a plant with multiple production stages is considered, and a number of items move along the production line. Each stage is assumed to be characterized by a specific value of an observable physical quantity (e.g., temperature, pressure, vibration, or a combination of heterogeneous variables): in this respect, a gradient of such quantity can be measured across

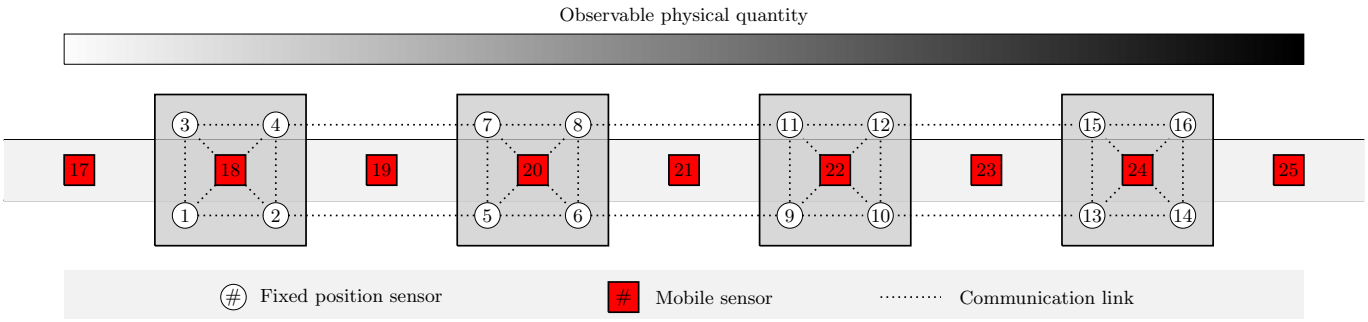


Fig. 4. Application scenario #1: sample industrial line–production process characterized by $L = 4$ stages (gray boxes) each monitored by $N_l = 4$, $l = 1, \ldots, L$ sensors (white numbered circles), and several moving items (depicted as red squares), which are associated to measurement sensors.

the different production phases and a monitoring IWSN is installed, made up by different subnetworks, each corresponding to a distinct production phase. Also, the item that is undergoing the production process is equipped with or accompanied by a wireless sensor, so that its own local measurement can be taken along the production line.

In correctly working conditions, the fixed sensors would be grouped into four clusters, corresponding to the different production stages, and the mobile node associated to the item can be inserted into a specific cluster, meaning that it is undergoing the corresponding stage, or grouped by itself, when traveling from one stage to another. It follows that, since the expected outcome of the clustering procedure for a mobile node is a priori known, any deviation from such behavior can be seen as an anomaly and detected by an intelligent supervisory system, which may be distributed as well. In this context, DCA is able to keep track of the item state and detect possible anomalies, given the fact that it takes into account both communication network topology and measurements distribution. This detection can be actually based on a simple *cluster label* comparison, by opportunely choosing the IDs for the fixed and the mobile nodes. With reference to the example in Fig. 4, all fixed sensors have a lower ID with respect to the mobile nodes: since any node can retrieve the cluster it is included in by looking at its the *cluster label* (that assumes the value of the lowest node ID in the same cluster), a mobile node can recognize if it is grouped alone (cluster label equal to its ID) or grouped in a specific cluster (cluster label lower than its ID). Consequently, since the evolution of its cluster label over time can be a priori stored in the node's memory, the node itself can autonomously detect an anomaly when the actual evolution differs from the expected one.

The application of DCA to this framework allows to identify several different anomalies. In particular, in this context the following ones are considered:

- **Measurement fault:** the value of the observed quantity measured by the mobile node at one stage is significantly different from that expected. In this case, the node clusters by itself when it should be clustered with the stage nodes.
- **Timing mismatch:** the mobile node reaches a certain stage earlier or later than expected. In this case, the evolution of the cluster label is anticipated or delayed with respect to the nominal trend.
- **Communication fault:** the mobile node experiences communication loss and is no longer able to exchange messages with other nodes. In this case, the node is always grouped by itself.

The intelligent supervisory system behavior in presence of these types of anomalies has been simulated for different scenarios, with multiple fault instances and increasing network complexity ranging from tens to thousands of sensor nodes: the results of these numerical experiments consistently show that the proposed method always allows to detect the different kinds of anomalies with no occurrence of false positives.

An example in this sense is reported in Fig. 5 for a mobile node that experiences a failure, in the scenario represented in Fig. 4: here, the observed physical quantity is the process temperature and $L = 4$ production stages are characterized by temperature ranges $\Delta T_1 = [5, 10]^\circ$ C, $\Delta T_2 = [15, 20]^\circ$ C, $\Delta T_3 = [25, 30]^\circ$ C and $\Delta T_4 = [35, 40]^\circ$ C, while the initial temperature measured by the mobile node is below $0\,^\circ$C (temperature of the raw material before the process). A value of $b = 4\,^\circ$C has been selected as the clustering bound. The resulting system behaviors (actually, referring to one realization of such scenario) are shown in terms of measurements trends, clusters evolution, and fault detection signal. This latter signal is obtained as the mismatch (computed in practice via logical XOR) between the actual clusters evolution value with the reference expected one. Interestingly, in the case of communication fault, namely loss of signal from the monitoring mobile station, the measurement evolution of the mobile node is statistically close to the reference one, to the point that no anomaly can be identified by looking only at the measurements, while the cluster label comparison promptly reveals the anomaly.

B. Experimental Dataset: Environmental Sensing Application

The DCA approach has also been applied to a dataset coming from a sparse sensor network deployed in a public indoor environment characterized by heterogeneous usage zones, with reference to Fig. 6a–6b, respectively Area 1 to 4. The considered monitoring network is composed by

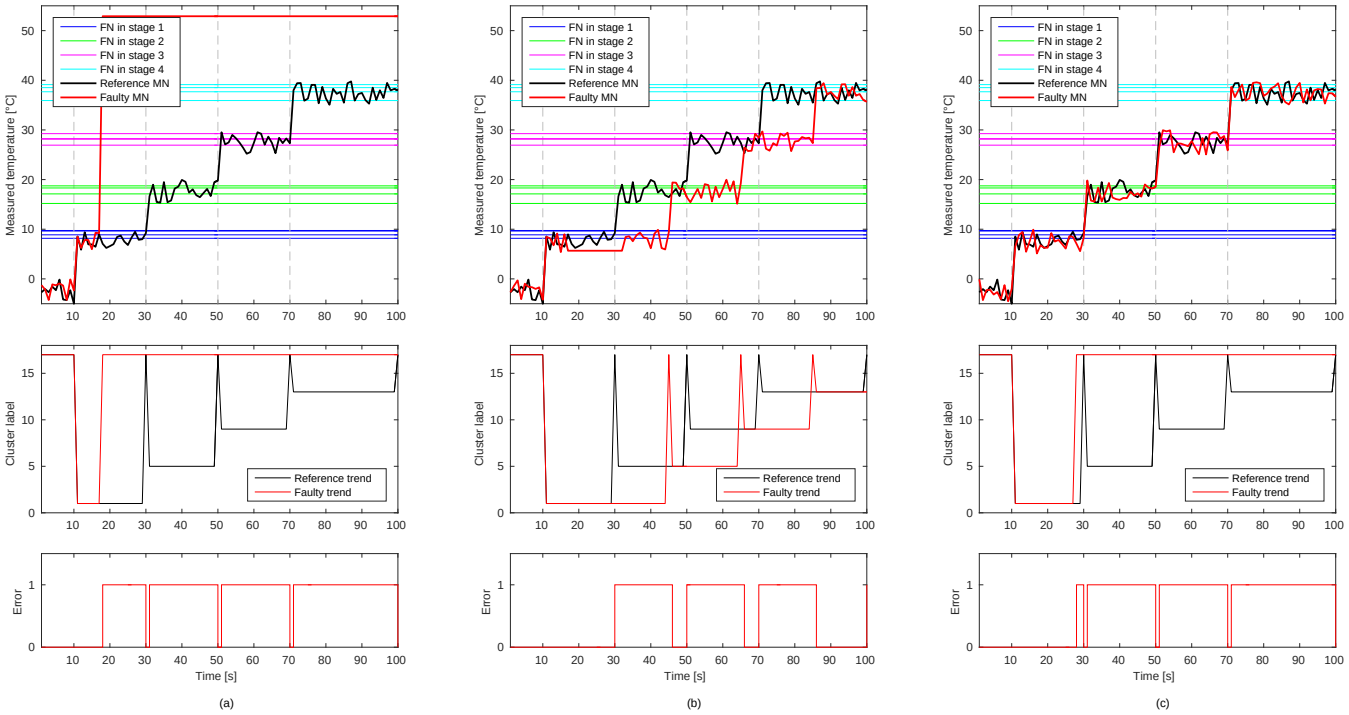


Fig. 5. Application scenario #1: system behavior in pres++ence of anomalies of different kind, i.e. (a) measurement fault, (b) timing mismatch and (c) communication error. For each anomaly, it is reported the measurements of fixed nodes (FN) and mobile node (MN) in the top row, the clustering trends in the middle row, and the fault detection signal in the bottom row.

$N = 17$ wireless t-mote nodes [24] allocated in 4 different areas composed of multiple rooms and a sample connection is assumed as shown in the graph reported in Fig. 6a. This dataset has been derived from a 4 months monitoring period that includes weekends and holidays; in detail, each sensor measures a temperature with a fixed sampling interval of five minutes.

In such a context, the application of DCA to a set of static measurements collected at a specific time instant (e.g. at 12 a.m. of a generic weekday) with a suitable clustering bound b yields a network partition such that there is a two–way correspondence between clusters and areas, as reported in Fig. 6a. Indeed, this cluster–area correspondence is not achieved by partitioning the network through KMC and DMGA strategies (see Fig. 6b). On the one hand, even if with $k = 4$ KMC identifies four clusters, they do not coincide with the different areas of the building, mainly because of the reduced signal variability in the whole environment. On the other hand, the implementation of DMGA with $N_{min} = 2$ to allow for the identification of small groups, leads to seven clusters because of the network sparsity. Remarkably, DCA strategy can handle both of these aspects correctly detecting the four areas, which are characterized by a specific measurement behavior and, hence, may be managed in a dedicated way by an intelligent environment controller.

Static data processing, however, may be not informative enough for a building management system with the aim of energy profile optimization and efficiency. In this sense, the application of the clustering algorithm during a large

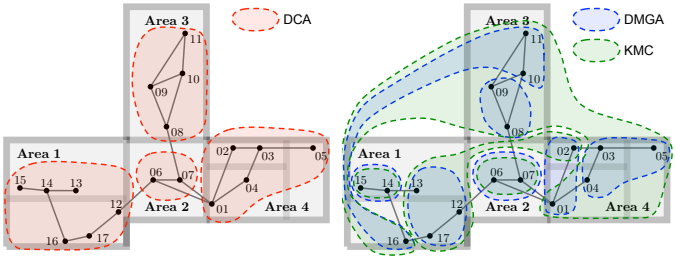


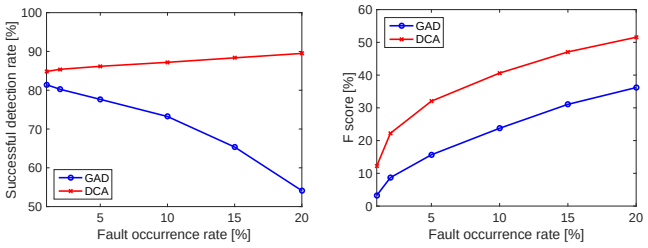
Fig. 6. Application scenario #2: (a) DCA network partition; (b) KMC (green) and DMGA (blue) network partitions.

observation interval allows to extract general trends and to build an effective anomaly detection strategy. In order to validate this claim, anomalous values are artificially inserted in the measurements collected by the network during 8 weeks of operations. In particular, two anomaly models are here considered [25]:

- **constant**: the sensor reports a constant value that corresponds to the measurement at the beginning of the fault period multiplied by a factor $\gamma = 2$;
- **noise** : the sensor measurement is affected by an additive Gaussian noise with zero mean and standard deviation $\sigma = 10°\text{C}$.

Different anomaly occurrence rates have been considered and the duration of each fault is a uniform random variable with an average value of 12 samples (1 hour)

In this framework, a simple yet effective fault detection



(a) (b)

Fig. 7. Application scenario #2: (a) anomaly detection accuracy; (b) F–score.

strategy based on DCA is adopted: the algorithm runs in a supervised manner during the first week of operation, without any fault, and, at each sample, every node stores the list of its neighbors, i.e., the nodes in its same cluster. Then, in the following weeks, DCA is applied at any new measurements (with a possible presence of faults) and each node compares the detected neighbors with those stored for the corresponding instant of the training week: if less than 50% of current neighbors are not in the list, the node self–declares as faulty.

To evaluate the performance of this strategy against a state–of–the–art anomaly detection scheme, GAD algorithm [14] is applied to the same dataset. The anomaly detection phase starts after the first week and the following parameters are used: $N_{min} = 2$, $c_{min} = 0.5$ and $\Delta_t = 90$ samples (7.5 hours). It has to be noted that the choice of setting the minimum number of cluster elements to 2 is imposed by the considered scenario, characterized by relatively few nodes, sparsely connected and with a significant measurement variability. However, this choice strongly affects the detection capabilities of GAD, which generally requires denser clusters. To cope with this issue, the algorithm has been slightly modified, labeling as faulty all the nodes whose status is uncertain. This conservative choice is motivated by the significant robustness required by industrial sensing applications, which translates into the urge to detect as many faults as possible, enduring a possible high occurrence of false positives. Following a similar reasoning, the bound of the DCA algorithm is set to an adequately low value ($b = 2°$C), so as to privilege detection of faults with respect to false positives.

The performance comparison between the two anomaly detection strategies is shown in Fig. 7 for several values of occurrence rate. Fig. 7a reports the percentage of anomalies detected on the total number of generated ones. The detection strategy based on DCA outperforms GAD for every value of anomaly rate, yielding an accuracy always greater than 85%, while using the other strategy it drops below 60% for a fault rate of 20%. To highlight how both strategies suffer from the presence of false positives due to the significant irregularity of the measurement trends, the F–score, a metric that sums up accuracy and precision, is depicted in Fig. 7b. It can be observed that this metric is low, especially for low anomaly occurrence rates, where the incidence of false positives is significant. However, also from this point of view, the anomaly detection strategy based on DCA confirms its validity, outperforming GAD by a wide margin.

V. Conclusions and Future Works

Within the framework of distributed intelligent systems, this paper aims at designing strategies to effectively partition an IWSN into non–overlapping clusters of nodes.

To this purpose, three clustering criteria are proposed, that take into account both communication network topology and the measurements gathered by the sensor nodes. Indeed, these features results to be important in noisy industrial environments, where both the network connectivity and the measurement consistency concur to guarantee IWSN performance in terms of timeliness, reliability and security. In order to accomplish this task, two clustering strategies are proposed following either a centralized or a distributed approach, the former (CCA) relying on the presence of a central coordinating unit, the latter (DCA) employing the network itself as a computational grid, without the need of identifying CHs.

Effectively, the distributed solution converges to the centralized one after some iterations and hence emerges as the preferred one for IWSNs thanks to the inherent properties of autonomicity, scalability and robustness. The proposed DCA procedure is then tested both in numerical simulations and on a real–world dataset, to provide an assessment of its performance in environmental monitoring and fault detection applications employed in building and process automation. In this evaluation, the performance of DCA are compared with both a classical k-means approach and a most recent procedure, leading to conclude that the proposed algorithm outperforms other approaches in accomplishing the clustering task and can be used, for example, as a basis to develop efficient anomaly and fault detection strategies to be employed in intelligent industrial monitoring applications.

Future research directions will be focused on the real–time implementation of the proposed clustering algorithms in a testbed wireless sensor network to investigate the practical challenges that could arise from their adoption. Moreover, future developments include the theoretical and simulative comparison of the proposed methods in more specific scenarios, as well as the deployment of such procedures in actual IWSN applications.

References

[1] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, Oct 2009.

[2] G. Hancke, V. Gungor, and G. Hancke, "Guest Editorial Special Section on Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 762–765, Feb 2014.

[3] P. Leitao, V. Marik, and P. Vrba, "Past, present, and future of industrial agent applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2360–2372, Nov 2013.

[4] J. Akerberg, M. Gidlund, and M. Bjorkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *9th IEEE International Conference on Industrial Informatics (INDIN)*, July 2011, pp. 410–415.

[5] V. C. Gungor and G. P. Hancke, *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2013.

[6] F. Osterlind, E. Pramsten, D. Roberthson, J. Eriksson, N. Finne, and T. Voigt, "Integrating building automation systems and wireless sensor networks," in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept 2007, pp. 1376–1379.

[7] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun, "Distributed collaborative control for industrial automation with wireless sensor and actuator networks," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 4219–4230, Dec 2010.

[8] E. Fadel, V. Gungor, L. Nassef, N. Akkari, M. A. Maik, S. Almasri, and I. F. Akyildiz, "A Survey on Wireless Sensor Networks for Smart Grid," *Computer Communications*, vol. 71, no. C, pp. 22–33, Nov. 2015.

[9] S. Ivanov, K. Bhargava, and W. Donnelly, "Precision farming: Sensor analytics," *IEEE Intelligent Systems*, vol. 30, no. 4, pp. 76–80, July 2015.

[10] J. Neuzil, O. Kreibich, and R. Smid, "A distributed fault detection system based on IWSN for machine condition monitoring," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1118–1123, May 2014.

[11] M. Gholami and R. W. Brennan, "Comparing two clustering-based techniques to track mobile nodes in industrial wireless sensor networks," *Journal of Systems Science and Systems Engineering*, vol. 25, no. 2, pp. 177–209, 2016.

[12] X. Ding, Y. Tian, and Y. Yu, "A Real-Time Big Data Gathering Algorithm Based on Indoor Wireless Sensor Networks for Risk Analysis of Industrial Operations," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 1232–1242, June 2016.

[13] W. Zhang, G. Han, Y. Feng, L. Cheng, D. Zhang, X. Tan, and L. Fu, "A Novel Method for Node Fault Detection Based on Clustering in Industrial Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Jan. 2015.

[14] P.-Y. Chen, S. Yang, and J. McCann, "Distributed real-time anomaly detection in networked industrial sensing systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3832–3842, June 2015.

[15] M. M. Afsar and M.-H. Tayarani-N, "Clustering in sensor networks: A literature survey," *Journal of Network and Computer Applications*, vol. 46, pp. 198 – 226, 2014.

[16] I. Eyal, I. Keidar, and R. Rom, "Distributed data clustering in sensor networks," *Distributed Computing*, vol. 24, no. 5, pp. 207–222, 2011.

[17] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 1415, pp. 2826 – 2841, 2007.

[18] Y. Ma, Y. Guo, X. Tian, and M. Ghanem, "Distributed clustering-based aggregation algorithm for spatial correlated sensor networks," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 641–648, March 2011.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, ser. Springer Series in Statistics. Springer, 2009.

[20] G. Bianchin, A. Cenedese, M. Luvisotto, and G. Michieletto, "Distributed fault detection in sensor networks via clustering and consensus," in *54th IEEE International Conference on Decision and Control (CDC)*, Dec 2015, pp. 3828–3833.

[21] M. Penrose, *Random geometric graphs*, ser. Oxford studies in probability. Oxford, New York: Oxford University Press, 2003.

[22] S. Piramuthu and W. Zhou, *RFID and Sensor Network Automation in the Food Industry: Ensuring Quality and Safety Through Supply Chain Visibility*. Wiley, 2016.

[23] G. A. Susto, S. Pampuri, A. Schirru, A. Beghi, and G. De Nicolao, "Multi-step virtual metrology for semiconductor manufacturing: A multilevel and regularization methods-based approach," *Computers & Operations Research*, vol. 53, pp. 328–337, 2015.

[24] Moteiv Corporation, "T-mote Sky: Low power wireless sensor module," June 2006. [Online]. Available: http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf

[25] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Transactions on Sensor Networks*, vol. 6, no. 3, pp. 23:1–23:39, Jun. 2010.