Applying network analysis for extracting knowledge about environment changes from heterogeneous sensor data streams

Francesco Cauteruccio[1], Paolo Lo Giudice[2], Giorgio Terracina[1], and Domenico Ursino[3]

[1] DEMACS, University of Calabria
{cauteruccio,terracina}@mat.unical.it,
[2] DIIES, University "Mediterranea" of Reggio Calabria
paolo.lo.giudice@unirc.it
[3] DICEAM, University "Mediterranea" of Reggio Calabria
ursino@unirc.it

Abstract. Sensor network analysis has become a challenging task. The detection of sensor anomalies is one of the most prominent topics in this research area. In the past, researchers mainly focused on the detection and analysis of single-sensor anomalies. In this paper, we shift the focus from a local approach, aimed to detect anomalies on single sensors, to a global one, aiming at detecting and investigating the consequences, on the whole sensor network and/or its subnetworks, of anomalies present in one or more (heterogeneous) sensors.

Keywords: Wireless Sensor Networks, Anomaly Detection, Network Analysis, Dashboard

1 Introduction

In the last few years, research on Wireless Sensor Networks (WSNs) has been ignited by important advances in various technological areas, such as wireless communications, digital electronics and micro-electro-mechanical systems. These improvements allowed for an easy development of low-power and low-cost multi-functional sensors and networks thereof. Sensor networks usually include a large number of nodes, each of which may sense several measures. Cooperation among nodes is usually sought for in such networks. Sensor nodes are usually positioned either inside or very close to observed events, and the main objective is to provide users with a better understanding of the environment in which sensors are deployed, thus giving the opportunity to acquire new information and intelligence. While the management of sensor networks and the development of robust data acquisition layers received much attention in the literature, one big open challenge in this research area is anomaly detection [4, 5]. Anomalies can be generated by either malfunctioning sensors or changes in the monitored environment. In most cases, being able to distinguish between the two scenarios is a challenging task. Most of the past approaches for anomaly detection focused on the analysis of data produced by each single device [16]. The most notable approaches in this setting can be grouped in four categories, namely: *(i)* rule-based detection [8], *(ii)* statistical techniques [12], *(iii)* graph-based techniques [13], and *(iv)* data mining and computational intelligence-based techniques [15]. Instead, network-based approaches for anomaly detection in WSNs received less attention [3, 9, 2, 17]. In fact, in spite of a strict complementarity and correlation between network analysis and WSNs, only in the latest years, researchers have begun to apply network analysis-based techniques to WSNs. However, they have only proposed the application of classical network analysis parameters to this context. Indeed, most of the proposed approaches employ centrality measures [14], which allow the detection of anomalies of only one node at a time.

In this paper, we aim at introducing new solutions for the analysis of heterogeneous sensors organized as a network. In particular, our techniques will be based on the evaluation of the connectivity of the whole WSN and its subnetworks (instead of on node centrality), and are mainly focused on potential anomalies involving more sensors located therein. They adopt a metric capable of uniformly handling measures provided by heterogeneous sensors, as well as a dashboard of network analysis parameters. This way, they allow the detection of anomalies involving more (heterogeneous) sensors, and the evaluation of the impact of these anomalies on the whole sensor network and its subnetworks. The plan of this paper is as follows. In Section 2, we introduce our model used to represent WSNs and our anomaly detection approach. In Section 3, we present some preliminary results on tests carried out on a sensor network, along with some discussions. Conclusions and future work are illustrated in Section 4.

2 Methods

2.1 Network construction

Let $\mathcal{W}$ be a WSN. Without loss of generality, assume that the corresponding sensors can be partitioned along two orthogonal dimensions[4]. In the scenario considered here, these dimensions are location and physical quantities to evaluate (in particular, we consider $p = 3$ physical quantities, i.e., temperature, lightness and humidity). Assume that the WSN covers l locations (in particular, we consider $l = 3$ locations, named A, B and C in the following) and that one location contains n devices, each measuring p physical quantities. As a consequence, the overall number of sensors is $s = pln$.

A network $\mathcal{N} = \langle V, E \rangle$ can be associated with $\mathcal{W}$. Here, V is the set of the nodes of $\mathcal{N}$. Each node $v_i \in V$ corresponds to a sensor and has associated a label $\langle l_i, p_i \rangle$, where l_i represents its location and p_i denotes the physical quantity it measures. E is the set of the edges of $\mathcal{N}$. Each edge e_{ij} connects the nodes v_i and v_j. It can be represented as $e_{ij} = (v_i, v_j, w_{ij})$. Here, w_{ij} is a measure of

[4] Actually, the number of dimensions could be greater than two, without requiring any change of the approach.

"distance" between v_i and v_j. It is an indicator of the non-correlation level of the sensors associated with v_i and v_j. Actually, each parameter representing this feature could be adopted in our model. In the experiments presented in this paper we adopted Multi-Parameterized Edit Distance (MPED) [1] for its capability of measuring the non-correlation level of sensors regarding heterogeneous physical quantities, characterized by different units of measure and possible data shifts.

$\mathcal{N}$ can be partitioned along one or both dimensions. We indicate by $\mathcal{N}_p = \langle V_p, E_p \rangle$ the subnets obtained by taking only the nodes that correspond to the sensors measuring the physical quantity p. Here, $p \in \{l, t, h\}$ can denote lightness, temperature and humidity, respectively. Analogously, we indicate by $\mathcal{N}_q = \langle V_q, E_q \rangle$ the subnets obtained by taking only the nodes that correspond to the sensors operating at the location q. Here, $q \in \{A, B, C\}$. Finally, we denote by $\mathcal{N}_{pq} = \langle V_{pq}, E_{pq} \rangle$ the subnet obtained by considering only the nodes corresponding to the sensors that measure the physical quantity p and operate in the location q, along with the edges linking them.

2.2 Network parameters

As pointed out in the Introduction, we use several parameters to construct our dashboard supporting the extraction of knowledge about environment changes. The first four parameters are derived from classical network theory; the fifth is derived from a particular centrality measure proposed in [10]; the last is introduced by us. In this section, we present an overview of these parameters. In the following, we define all of them on a reference network $\mathcal{N} = \langle V, E \rangle$. The first parameter is the *Characteristic Path Length*, also known as the *Average Shortest Path Length*. It is defined as the average length of the shortest paths connecting all possible pairs of network nodes. More formally, let $l(v_i, v_j)$ be the length of the shortest path between v_i and v_j. The Characteristic Path Length $\mathcal{L}_{\mathcal{N}}$ of $\mathcal{N}$ is defined as: $\mathcal{L}_{\mathcal{N}} = \frac{1}{|V|(|V|-1)} \sum_{v_i \in V} \sum_{v_j \in V, v_j \neq v_i} l(v_i, v_j)$. The second parameter is the *Average Node Connectivity*. Given two nodes v_i and v_j, their connectivity $c(v_i, v_j)$ represents the minimum number of edges that need to be removed to disconnect them. The Average Node Connectivity $\mathcal{C}_{\mathcal{N}}$ is defined as: $\mathcal{C}_{\mathcal{N}} = \frac{1}{|V|(|V|-1)} \sum_{v_i \in V} \sum_{v_j \in V, v_j \neq v_i} c(v_i, v_j)$. The third parameter is the *Average Number of Simple Paths*. Given two nodes v_i and v_j, we indicate by $p(v_i, v_j)$ the number of simple paths (i.e., paths with no repeated nodes) between them. Then, we define the Average Number of Simple Paths $\mathcal{P}_{\mathcal{N}}$ as: $\mathcal{P}_{\mathcal{N}} = \frac{1}{|V|(|V|-1)} \sum_{v_i \in V} \sum_{v_j \in V, v_j \neq v_i} p(v_i, v_j)$. The fourth parameter is the *Average Clustering Coefficient*. In order to define it, we must preliminarily introduce the neighborhood $nbh(v_i)$ of a node v_i as follows: $nbh(v_i) = \{v_j | e_{ij} \in E\}$. Then, we define the Clustering Coefficient of a node v_i as: $s(v_i) = \frac{2 \cdot |\{e_{jk} | v_j, v_k \in nbh(v_i), e_{jk} \in E\}|}{|nbh(v_i)| \cdot (|nbh(v_i)| - 1)}$. Finally, we define the Average Clustering Coefficient as: $\mathcal{S}_{\mathcal{N}} = \frac{1}{|V|} \sum_{v_i \in V} s(v_i)$. The fifth parameter is the *Average Closeness Vitality*. Given a node v_i, the closeness vitality $t(v_i)$ represents the increase in the sum of distances between all the pairs of nodes of $\mathcal{N}$, when v_i is excluded from $\mathcal{N}$ [10]. The Average Closeness Vitality $\mathcal{T}_{\mathcal{N}}$ is defined as: $\mathcal{T}_{\mathcal{N}} = \frac{1}{|V|} \sum_{v_i \in V} t(v_i)$.

The sixth parameter (i.e., the one introduced by us) is the *Connection Coefficient*. It starts from the observation that, in network analysis, one of the most powerful tools for investigating the connection level of a network is the concept of clique. As a consequence, it is reasonable to adopt this concept to evaluate the cohesion of a network. This coefficient takes the following considerations into account: *(i)* both the dimension and the number of cliques are important as connectivity indicators; *(ii)* the concept of clique is intrinsically exponential; in other words, a clique of dimension $n+1$ is exponentially more complex than a clique of dimension n.

In order to define the Connection Coefficient it is necessary to introduce a support network $\mathcal{N}^\pi = \langle V, E^\pi \rangle$, obtained by removing from $\mathcal{N}$ the edges with an "excessive" weight; observe that the nodes of $\mathcal{N}^\pi$ are the same as the nodes of $\mathcal{N}$. To formally define E^π, we employ the distribution of the weights of the edges of $\mathcal{N}$. Specifically, let max_E (resp., min_E) be the maximum (resp., minimum) weight of an edge of E. It is possible to define a parameter $step_E = \frac{max_E - min_E}{10}$, which represents the length of a "step" of the interval between min_E and max_E. We can define $d^k(E)$, $0 \leq k \leq 9$, as the number of the edges of E whose weights belong to the interval between $min_E + k \cdot step_E$ and $min_E + (k+1) \cdot step_E$. All these intervals are closed on the left and open on the right, except for the last one that is closed both on the left and on the right. E^π can be defined as: $E^\pi = \{e_{ij} \in E | e_{ij} \in \bigcup_{k \leq th_{max}} d^k(E)\}$. We have experimentally set $th_{max} = 6$. We are now able to define the Connection Coefficient $\mathcal{Q}_\mathcal{N}$ of $\mathcal{N}$. In particular, let C be the set of the cliques of $\mathcal{N}^\pi$; let C_k be the set of cliques of dimension k of $\mathcal{N}^\pi$; finally, let $|C_k|$ be the cardinality (i.e., the number of cliques) of C_k. Then, $\mathcal{Q}_\mathcal{N}$ is defined as: $\mathcal{Q}_\mathcal{N} = \sum_{k=1}^{|V|} |C_k| \cdot 2^k$.

2.3 Approach to knowledge extraction

The idea underlying our approach is that, if some changes occur on sensor data streams, then some variations can be observed in some or all the dashboard parameters, when measured on the whole network, and/or on some of its sub-networks, depending on the number, the kind and the location of involved sensors. Our approach consists of a training phase and a testing phase. To carry out them, we employed available data (see Section 3.1) and, according to the holdout technique, we partitioned these data in such a way as to use 2/3 of them for the training phase and 1/3 of them for the testing phase. As for the training phase, we considered the following situations: *(1)* all sensors behaved correctly; *(2)* two sensors in location A and two sensors in location B were perturbed, in such a way as to decrease humidity; *(3)* two sensors in location B and two sensors in location C were perturbed, in such a way as to decrease lightness; *(4)* two sensors in location A and two sensors in location C were perturbed, in such a way as to increase lightness. Obtained results, along with the corresponding discussion, are presented in Section 3. After the training phase, we started the testing phase. In this case, we considered the following situations: *(1)* all sensors behaved correctly; *(2)* two sensors in location B and two sensors in location C were perturbed, in such a way as to decrease humidity; *(3)* two sensors in location A and two sensors in location C were perturbed, in such a way as to decrease lightness; *(4)* two sensors in location A and two sensors in location B were perturbed, in such a way as to increase lightness. Obtained results, along with the corresponding discussion, are presented in Section 3. Here, we simply point out that our approach behaved very well and was capable of correctly identifying all perturbations.

Finally, we applied our approach to the following situations: *(1)* one sensor in the location A and one sensor in the location B were perturbed, in such a way as to decrease humidity; *(2)* one sensor in the locations A and C was perturbed, in such a way as to increase lightness, and one sensor in the locations B and C was perturbed, in such a way as to decrease the same physical quantity; *(3)* three sensors in the location A and one sensor in the location B were perturbed, in such a way as to decrease humidity; *(4)* one sensor in the location A was perturbed, in such a way as to increase humidity; *(5)* one sensor in the location B was perturbed, in such a way as to increase lightness. Obtained results, along with the corresponding discussion, are presented in Section 3. Here, we anticipate that our approach showed its suitability to detect almost all perturbations.

3 Results

3.1 Testbed

To collect data for the experiments introduced in Section 2.3, we built a WSN by following specific guidelines. In particular, we organized devices in a multi-hop Wireless Sensor Area Network (WSAN) and managed them through the Building Management Framework (BMF) [6]. This is a framework for domain-specific networks, which offers an efficient and flexible management of WSANs deployed in indoor areas by allowing users to take advantage of sensing/actuation intelligent techniques and fast prototyping of WSAN applications. BMF enabled the use of heterogeneous WSANs through a base station, which acted both as data collector and network configurator. Communication between base station and devices was carried out by means of the BMF Communication Protocol, an application level protocol built on top of multi-hop network protocols [11, 7]. We composed the WSAN using MICAz sensor devices, providing 128 kB for program storage, 512 kB for data storage, and 4 kB of RAM. Devices were powered mainly by means of external power. They were configured to communicate with the base station, sending data every minute. To test our approach, we synthetically injected several anomalies at pre-determined time slots. In particular, to increase lightness, we employed artificial sources of lightness with controlled intensity, whereas to reduce lightness, we applied artificial lightness filters. Finally, humidity was controlled by chemicals. Our network consisted of 9 devices labeled by increasing numbers. Each device included 3 sensors, which retrieved values for humidity, lightness and temperature. Devices 1, 2 and 3 have been positioned in location A, devices 4, 5 and 6 operated in location B, devices 7, 8 and 9 were situated in location C. A, B and C were three different rooms on the same floor of a building. Finally, we collected data for 24 days without perturbations and other 36 days with several perturbations, as described in Section 2.3.

3.2 Obtained results and Discussion

In this section, we report the results obtained by performing all the experiments mentioned in Section 2.3. Preliminarily, we observe that the definition of the six coefficients forming our dashboard suggests that a decrease of the connection level of a network or a subnetwork leads to: *(i)* an increase of $\mathcal{L}_{\mathcal{N}}$ and $\mathcal{T}_{\mathcal{N}}$; *(ii)* a decrease of $\mathcal{C}_{\mathcal{N}}$, $\mathcal{P}_{\mathcal{N}}$, $\mathcal{S}_{\mathcal{N}}$ and $\mathcal{Q}_{\mathcal{N}}$. The purpose of the training phase was to find the optimal values of some thresholds underlying our approach (for instance, the value of th_{max} in the definition of Connection Coefficient - see Section 2.2) and to have a first idea of its behavior. In Table 1, we report all the results regarding the training phase after the optimal values of thresholds were set. In particular, this table consists of four sub-tables, each corresponding to one of the four situations mentioned in Section 2.3. For each situation, we report the values of the six parameters of the dashboard for the overall network and the subnetworks $\mathcal{N}_t$, $\mathcal{N}_l$, $\mathcal{N}_h$, $\mathcal{N}_A$, $\mathcal{N}_B$ and $\mathcal{N}_C$ (see Section 2.1). In this table, Situation 1 represents the correct one. In Situation 2, we observe: *(i)* a very high increase of $\mathcal{L}_{\mathcal{N}}$ and $\mathcal{T}_{\mathcal{N}}$, along with a very high decrease of $\mathcal{C}_{\mathcal{N}}$, $\mathcal{P}_{\mathcal{N}}$, $\mathcal{S}_{\mathcal{N}}$ and $\mathcal{Q}_{\mathcal{N}}$ for the network $\mathcal{N}_h$; *(ii)* a high increase of $\mathcal{L}_{\mathcal{N}}$ and $\mathcal{T}_{\mathcal{N}}$, along with a high decrease of $\mathcal{C}_{\mathcal{N}}$, $\mathcal{P}_{\mathcal{N}}$, $\mathcal{S}_{\mathcal{N}}$ and $\mathcal{Q}_{\mathcal{N}}$ for the networks $\mathcal{N}_A$ and $\mathcal{N}_B$; *(iii)* a moderate increase of $\mathcal{L}_{\mathcal{N}}$ and $\mathcal{T}_{\mathcal{N}}$, along with a moderate decrease of $\mathcal{C}_{\mathcal{N}}$, $\mathcal{P}_{\mathcal{N}}$, $\mathcal{S}_{\mathcal{N}}$ and $\mathcal{Q}_{\mathcal{N}}$ for the overall network. In Situation 3 (resp., 4), we observe: *(i)* a very high increase of $\mathcal{L}_{\mathcal{N}}$ and $\mathcal{T}_{\mathcal{N}}$, along with a very high decrease of $\mathcal{C}_{\mathcal{N}}$, $\mathcal{P}_{\mathcal{N}}$, $\mathcal{S}_{\mathcal{N}}$ and $\mathcal{Q}_{\mathcal{N}}$ for the network $\mathcal{N}_l$; *(ii)* a high increase of $\mathcal{L}_{\mathcal{N}}$ and $\mathcal{T}_{\mathcal{N}}$, along with a high decrease of $\mathcal{C}_{\mathcal{N}}$, $\mathcal{P}_{\mathcal{N}}$, $\mathcal{S}_{\mathcal{N}}$ and $\mathcal{Q}_{\mathcal{N}}$ for the networks $\mathcal{N}_B$ and $\mathcal{N}_C$ (resp., $\mathcal{N}_A$ and $\mathcal{N}_C$); *(iii)* a moderate increase of $\mathcal{L}_{\mathcal{N}}$ and $\mathcal{T}_{\mathcal{N}}$, along with a moderate decrease of $\mathcal{C}_{\mathcal{N}}$, $\mathcal{P}_{\mathcal{N}}$, $\mathcal{S}_{\mathcal{N}}$ and $\mathcal{Q}_{\mathcal{N}}$ for the overall network. These results confirm that our approach is really capable of capturing the perturbations in wireless sensor networks or subnetworks caused by sensor anomalies (and, indirectly, it is able to evaluate the network and subnetwork resilience to sensor anomalies). The only weakness revealed by this first test is that, in its current version, our approach is not able to tell us if these perturbations are caused by an increase or a decrease of the corresponding physical quantity.

The purpose of the testing phase was to verify both the setting of the threshold values and the corresponding results detected during the training phase. In Table 2, we report all the results regarding this phase. Observe that the situations considered during this phase are the same as the ones examined during the training phase; however, we modified the subnetworks (among A, B and C) involved in each perturbation in such a way as to prevent overfitting. Obtained results confirm that the selection of the threshold values performed during the training phase was correct. They also confirm all the observations about the features of our approach, which we drew at the end of the training phase.

Network	$\mathcal{L}_\mathcal{N}$	$\mathcal{C}_\mathcal{N}$	$\mathcal{P}_\mathcal{N}$	$\mathcal{T}_\mathcal{N}$	$\mathcal{Q}_\mathcal{N}$	$\mathcal{S}_\mathcal{N}$
Overall	1.1054	22.4387	6508290	64.2548	1163264	0.8944
$\mathcal{N}_t$	1.0322	7.1056	14232	15.1429	592	0.8413
$\mathcal{N}_l$	1.0451	7.1111	13200	16.6667	592	0.8595
$\mathcal{N}_h$	1.0278	7.5833	16758	16.9143	512	0.9722
$\mathcal{N}_A$	1.1944	5.6944	8012	23.7241	224	0.8339
$\mathcal{N}_B$	1.1667	5.9444	9274	22.4000	256	0.8582
$\mathcal{N}_C$	1.1944	6.0556	7896	23.7241	288	0.7794
Overall	1.1795	20.0684	4652472	74.7500	227328	0.8239
$\mathcal{N}_t$	1.1189	6.4444	10376	21.1613	384	0.8212
$\mathcal{N}_l$	1.1011	6.5833	11816	20.0000	320	0.7905
$\mathcal{N}_h$	1.4167	3.9444	2268	38.0952	96	0.5270
$\mathcal{N}_A$	1.3611	4.5000	3208	34.0870	120	0.5582
$\mathcal{N}_B$	1.3456	4.7778	4572	32.0800	144	0.5858
$\mathcal{N}_C$	1.1833	6.0444	7828	26.9091	248	0.7832
Overall	1.2194	19.1937	3790486	81.2263	99840	0.7796
$\mathcal{N}_t$	1.2556	5.8778	9924	20.8824	412	0.7392
$\mathcal{N}_l$	1.5000	4.1111	6102	26.3704	192	0.6000
$\mathcal{N}_h$	1.0556	7.2778	14924	17.8824	512	0.9392
$\mathcal{N}_A$	1.2111	5.4000	7990	23.0000	200	0.8571
$\mathcal{N}_B$	1.3222	4.5278	5990	29.1429	108	0.5630
$\mathcal{N}_C$	1.3333	4.7778	3824	32.0000	120	0.5407
Overall	1.2394	18.1937	3480632	80.2263	97650	0.7823
$\mathcal{N}_t$	1.2356	5.6648	9633	21.2435	408	0.7491
$\mathcal{N}_l$	1.5200	3.9345	6260	27.3221	192	0.5800
$\mathcal{N}_h$	1.0776	6.9318	13924	17.7623	512	0.9154
$\mathcal{N}_A$	1.3782	4.4987	5843	28.2322	108	0.661
$\mathcal{N}_B$	1.1911	5.1000	7232	23.0000	206	0.8200
$\mathcal{N}_C$	1.3433	4.6578	3126	31.6850	120	0.5207

Table 1. Results obtained by our approach during the training phase

After the testing phase confirmed the suitability of our approach, we applied it to new situations not considered during the previous phases. These situations are described in detail in Section 2.3. In Table 3, we report the corresponding results. From their analysis we can draw very interesting observations. In particular, in Situation 1, we obtain the same trend as the one seen in Situation 2 of the training phase. However, the perturbation degree is more reduced. This is correct because, for locations A and B, we perturbed one sensor, instead of two. In Situation 2, we observe: *(i)* a very high increase of $\mathcal{L}_\mathcal{N}$ and $\mathcal{T}_\mathcal{N}$, along with a very high decrease of $\mathcal{C}_\mathcal{N}$, $\mathcal{P}_\mathcal{N}$, $\mathcal{S}_\mathcal{N}$ and $\mathcal{Q}_\mathcal{N}$ for the network $\mathcal{N}_l$; these increases and decreases are comparable with the ones observed in Situation 3 of the training phase; *(ii)* a moderate (resp., high, very high) increase of $\mathcal{L}_\mathcal{N}$ and $\mathcal{T}_\mathcal{N}$, along with a moderate (resp., high) decrease of $\mathcal{C}_\mathcal{N}$, $\mathcal{P}_\mathcal{N}$, $\mathcal{S}_\mathcal{N}$ and $\mathcal{Q}_\mathcal{N}$, for the networks $\mathcal{N}_A$ and $\mathcal{N}_B$ (resp., $\mathcal{N}_C$, $\mathcal{N}_l$); *(iii)* a moderate increase of $\mathcal{L}_\mathcal{N}$ and $\mathcal{T}_\mathcal{N}$, along with a moderate decrease of $\mathcal{C}_\mathcal{N}$, $\mathcal{P}_\mathcal{N}$, $\mathcal{S}_\mathcal{N}$ and $\mathcal{Q}_\mathcal{N}$, for the overall network. Observe that, since our approach considers perturbations, but it currently does not distinguish between increases and decreases, even if, in the network $\mathcal{N}_l$, there are opposite perturbations in two lightness sensors, their consequences are not nullified by our approach, but, on the contrary, are "combined" by it. In our opinion, this is a correct behavior of our approach. In Situation 3, we observe: *(i)* an increase (resp., decrease) of $\mathcal{L}_\mathcal{N}$ and $\mathcal{T}_\mathcal{N}$ (resp., $\mathcal{C}_\mathcal{N}$, $\mathcal{P}_\mathcal{N}$, $\mathcal{S}_\mathcal{N}$ and $\mathcal{Q}_\mathcal{N}$), comparable with the one of Situation 2 of the training phase for both the overall network and the network $\mathcal{N}_h$; *(ii)* a significant (resp., moderate) increase of $\mathcal{L}_\mathcal{N}$ and $\mathcal{T}_\mathcal{N}$, along with a significant (resp., moderate) decrease of $\mathcal{C}_\mathcal{N}$, $\mathcal{P}_\mathcal{N}$, $\mathcal{S}_\mathcal{N}$ and $\mathcal{Q}_\mathcal{N}$ for the network $\mathcal{N}_A$ (resp., $\mathcal{N}_B$). In Situation 4 (resp., 5), we ob-

Network	$\mathcal{L}_\mathcal{N}$	$\mathcal{C}_\mathcal{N}$	$\mathcal{P}_\mathcal{N}$	$\mathcal{T}_\mathcal{N}$	$\mathcal{Q}_\mathcal{N}$	$\mathcal{S}_\mathcal{N}$
Overall	1.1135	20.4387	7120293	65.3746	1163264	0.9144
$\mathcal{N}_t$	1.0411	6.5306	13939	15.1529	592	0.8712
$\mathcal{N}_l$	1.0361	6.2480	13737	17.1227	592	0.8891
$\mathcal{N}_h$	1.0235	7.3311	16123	16.8242	512	0.8920
$\mathcal{N}_A$	1.1826	5.4129	7910	22.7241	228	0.8451
$\mathcal{N}_B$	1.1700	5.8331	8992	21.4000	256	0.8112
$\mathcal{N}_C$	1.1929	6.2410	7786	23.7241	288	0.8042
Overall	1.1896	20.1224	4993459	72.63	294629	0.8484
$\mathcal{N}_t$	1.1289	6.2468	11001	22.1982	320	0.8391
$\mathcal{N}_l$	1.2133	6.6631	10829	21.0782	384	0.8081
$\mathcal{N}_h$	1.5177	3.8104	3124	37.1719	112	0.5328
$\mathcal{N}_A$	1.1922	6.2324	7128	27.8801	208	0.7312
$\mathcal{N}_B$	1.3232	4.9188	4492	31.9500	128	0.5558
$\mathcal{N}_C$	1.3511	4.4780	3198	33.0870	118	0.5182
Overall	1.2766	20.2308	4290486	81.3094	97744	0.7824
$\mathcal{N}_t$	1.3111	5.5833	9850	20.0000	258	0.7825
$\mathcal{N}_l$	1.4389	4.0833	3438	25.9750	96	0.6412
$\mathcal{N}_h$	1.0242	7.3611	13978	18.4421	384	0.9825
$\mathcal{N}_A$	1.3056	4.5278	4762	30.1515	108	0.5713
$\mathcal{N}_B$	1.1896	5.5278	7288	22.1429	216	0.8462
$\mathcal{N}_C$	1.2825	4.9444	3594	32.9143	96	0.5356
Overall	1.2251	17.9876	3990563	82.2263	97650	0.7769
$\mathcal{N}_t$	1.2944	5.8326	9112	22.7241	408	0.7839
$\mathcal{N}_l$	1.4678	4.6161	6383	26.3352	112	0.5455
$\mathcal{N}_h$	1.1111	6.5833	13816	17.6686	384	0.9005
$\mathcal{N}_A$	1.4001	4.7144	6152	27.8652	96	0.6148
$\mathcal{N}_B$	1.3675	4.3056	3886	30.9850	88	0.5198
$\mathcal{N}_C$	1.1887	6.2421	7341	22.7692	256	0.8825

Table 2. Results obtained by our approach during the testing phase

serve: *(i)* a very moderate increase of $\mathcal{L}_\mathcal{N}$ and $\mathcal{T}_\mathcal{N}$, along with a very moderate decrease of $\mathcal{C}_\mathcal{N}$, $\mathcal{P}_\mathcal{N}$, $\mathcal{S}_\mathcal{N}$ and $\mathcal{Q}_\mathcal{N}$ for the overall network and for the networks $\mathcal{N}_h$ and $\mathcal{N}_A$ (resp., $\mathcal{N}_l$ and $\mathcal{N}_B$). This reveals a second weakness of our approach, which shows a difficulty to find a single anomaly. Indeed, in this case, it found a slight change in the dashboard parameters for both the whole network and the involved subnetworks. This is mainly due to the purpose of our approach, which does not aim at performing anomaly detection in one sensor (actually, a long list of approaches carrying out this task - e.g., [8, 12, 13, 15] - already exists) but, instead, it aims at detecting the consequences, on the whole network and its subnetworks, of anomalies involving more (heterogeneous) sensors installed in different locations. In fact, in this case, the interaction of these anomalies in the network could be extremely variegate and could depend on the number, the kind and the location of perturbed sensors, so that their detection, along with the detection of their effects, becomes extremely difficult and justifies the employment of quite time-expensive approaches like ours. As for this issue, the results described in this section allow us to conclude that our approach reaches the objectives for which it was designed.

4 Conclusion

In this paper, we have presented a new approach to analyzing WSNs, which considers network organization as a whole; this shifts the focus of the analysis from single sensors to the whole network and its subnetworks. Our approach

Network	$\mathcal{L}_{\mathcal{N}}$	$\mathcal{C}_{\mathcal{N}}$	$\mathcal{P}_{\mathcal{N}}$	$\mathcal{T}_{\mathcal{N}}$	$\mathcal{Q}_{\mathcal{N}}$	$\mathcal{S}_{\mathcal{N}}$
Overall	1.1435	21.5534	5580928	70.0000	114264	0.8534
$\mathcal{N}_t$	1.0712	6.3159	11432	18.2221	384	0.8613
$\mathcal{N}_l$	1.0572	6.4354	11202	18.6667	384	0.8564
$\mathcal{N}_h$	1.2578	4.5673	4564	22.2124	144	0.8123
$\mathcal{N}_A$	1.2235	5.1843	6006	28.3673	200	0.7034
$\mathcal{N}_B$	1.2351	5.4992	8842	27.4332	224	0.6982
$\mathcal{N}_C$	1.1833	5.3556	7828	24.7347	248	0.7792
Overall	1.2199	19.3747	3948573	80.3252	97650	0.7856
$\mathcal{N}_t$	1.2456	5.6658	8562	21.9383	388	0.7467
$\mathcal{N}_l$	1.6100	3.5039	5987	28.2392	192	0.5971
$\mathcal{N}_h$	1.0877	6.4837	12527	17.3877	512	0.8672
$\mathcal{N}_A$	1.2292	4.5948	7873	27.223	228	0.6823
$\mathcal{N}_B$	1.2334	5.1229	7367	26.2391	228	0.6891
$\mathcal{N}_C$	1.2921	4.6578	3834	32.2320	120	0.5012
Overall	1.1235	21.9987	3977283	74.5673	231872	0.8223
$\mathcal{N}_t$	1.1312	6.2989	12345	21.3939	512	0.8323
$\mathcal{N}_l$	1.1433	6.5643	12234	20.3332	512	0.8340
$\mathcal{N}_h$	1.4872	3.9440	3542	38.9412	120	0.7795
$\mathcal{N}_A$	1.8342	2.2338	1987	35.1843	96	0.4032
$\mathcal{N}_B$	1.2151	4.4738	6932	25.6230	224	0.5820
$\mathcal{N}_C$	1.1933	6.0872	8239	23.3235	284	0.7780
Overall	1.1228	21.3789	6184736	67.3233	131872	0.8534
$\mathcal{N}_t$	1.0613	6.4599	12341	17.3939	592	0.8613
$\mathcal{N}_l$	1.0732	6.8865	12854	16.3452	592	0.8564
$\mathcal{N}_h$	1.1640	5.6534	9532	20.9482	288	0.8123
$\mathcal{N}_A$	1.2132	5.1928	6987	26.1212	288	0.7034
$\mathcal{N}_B$	1.1951	5.4738	9928	24.7210	320	0.6982
$\mathcal{N}_C$	1.19445	5.5872	8239	23.3235	320	0.7792
Overall	1.1289	21.8729	6857326	67.3252	131662	0.8556
$\mathcal{N}_t$	1.0782	6.7654	12662	17.2352	592	0.8467
$\mathcal{N}_l$	1.1728	5.9987	5987	20.4568	288	0.8023
$\mathcal{N}_h$	1.0654	6.2356	12277	16.4555	592	0.8553
$\mathcal{N}_A$	1.1892	5.6457	9854	25.3356	320	0.7061
$\mathcal{N}_B$	1.2234	5.0101	5346	26.4564	288	0.7072
$\mathcal{N}_C$	1.1921	5.5482	8899	23.2845	284	0.7843

Table 3. Results obtained by our approach during the examination of some situations of interest

is based on network connectivity measures that, overall, contribute to a rich dashboard, which allows the effective detection of perturbations in WSNs. Our model also allows the network to be sliced in different subnetworks, supporting the investigation of this phenomenon under different perspectives, as well as a better characterization of perceived perturbations. Our experimental campaign confirms the effectiveness of our approach. In the future, we plan to remove the current weaknesses of our approach, as evidenced by our experiments. First, we aim at allowing our approach to distinguish perturbations caused by an increase or a decrease of a physical quantity. Then, we plan to integrate our approach with the ones detecting anomalies in single sensors. The ultimate goal is to construct an effective framework, which can detect anomalies on single sensors and can investigate their consequences on the whole network and its subnetworks, along with their resilience to sensor malfunctions.

References

1. Cauteruccio, F., Stamile, C., Terracina, G., Ursino, D., Sappey-Marinier, D.: An automated string-based approach to extracting and characterizing White Matter

fiber-bundles. Computers in Biology and Medicine, Volume 77, 1 October 2016, Pages 64-75
2. Curia, V., Tropea, M., Fazio, P., Marano, S.: Complex networks: Study and performance evaluation with hybrid model for Wireless Sensor Networks. In Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference (pp. 1-5). IEEE.
3. Cuzzocrea, A., Papadimitriou, A., Katsaros, D., Manolopoulos, Y.: Edge betweenness centrality: A novel algorithm for QoS-based topology control over wireless sensor networks. Journal of Network and Computer Applications, Volume 35, Issue 4, July 2012, pp. 1210-1217.
4. Bosman, H., Liotta, A., Iacca, G., Wrtche, H.: Anomaly Detection in Sensor Systems Using Lightweight Machine Learning. 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, 2013, pp. 7-13.
5. Bosman, H., Liotta, A., Iacca, G., Wrtche, H.: Online extreme learning on fixed-point sensor networks. IEEE 13th International Conference on Data Mining Workshops (ICDMW), 2013, pp. 319-326.
6. Fortino, G., Guerrieri, A., OHare, G., Ruzzelli, A.: A flexible building management framework based on wireless sensor and actuator networks. Journal of Network and Computer Applications 35 (2012) 1934 1952.
7. Gnawali, O., Fonseca, R., Jamieson, K., Kazandjieva, M., Moss, D., Levis, P.: CTP: An Efficient, Robust, and Reliable Collection Tree Protocol for Wireless Sensor Networks. ACM Transactions on Sensors Networks 10(1) (December 2013) 16:116:49.
8. Ho, J.-W., et al.: Distributed detection of replica node attacks with group deployment knowledge in wireless sensor networks. Ad Hoc Networks 2009, 7 November, pp. 147688.
9. Jain, A.: Betweenness centrality based connectivity aware routing algorithm for prolonging network lifetime in wireless sensor networks. Wireless Networks, Volume 22, Issue 5, 2016, pp. 1605-1624.
10. Koschützki, D., Lehmann, K. A, Peeters, L., Richter, S., Tenfelde-Podehl, D., Zlotowski, O.: Centrality indices. In *Network analysis*, pages 16–61. 2005. Springer.
11. Levis, P., Patel, N., Culler, D., Shenker, S.: Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1. NSDI 04, Berkeley, CA, USA, USENIX Association (2004)
12. Li, G. et al.: Group-based intrusion detection system in wireless sensor networks. Computer Communications 2008a; 31 December, pp. 4324-32.
13. Ngai, E. C. H., Liu, J., Lyu, M. R.: An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks. Computer Communications, Volume 30, Issues 1112, 10 September 2007, pp. 2353-2364.
14. Papadimitriou, A., Katsaros, D., Manolopoulos, Y.: Social network analysis and its applications in wireless sensor and vehicular networks. International Conference on e-Democracy(pp. 411-420). Springer Berlin Heidelberg.
15. Yu, Z., Tsai, J. J. P.: A Framework of Machine Learning Based Intrusion Detection for Wireless Sensor Networks. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2008), Taichung, 2008, pp. 272-279.
16. Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. Annual Reviews in Control, 32(2), 2008, pp. 229252.
17. Xie, M., Han, S., Tian, B., Parvin, S.: Anomaly detection in wireless sensor networks: A survey. Journal of Network and Computer Applications, 34(4), pp. 2011, 1302-1325.