Short-Long Term Anomaly Detection in Wireless Sensor Networks based on Machine Learning and Multi-Parameterized Edit Distance

Francesco Cauteruccio, Giancarlo Fortino, Antonio Guerrieri

University of Calabria, Italy

Antonio Liotta

University of Derby, UK

Decebal Mocanu

Technical Univesity of Eindhoven, The Netherlands

Cristian Perra

University of Cagliari, Italy

Giorgio Terracina

University of Calabria, Italy

Maria Torres Vega

Ghent University, Belgium

Abstract

Heterogeneous wireless sensor networks are a source of large amount of different information representing environmental aspects such as, for example, light, temperature, and humidity. A very important research problem related to the analysis of the sensor data is the detection of relevant anomalies. This paper proposes a novel approach for automatic anomaly detection in heterogeneous sensor networks based on coupling edge data analysis and cloud data analysis methods. The edge data analysis exploits a fully unsupervised artificial neural network algorithm. The cloud data analysis exploits the multi-parameterized edit distance algorithm. The experimental evaluation of the proposed method is performed applying the edge and cloud analysis on real data acquired in an indoor building environment distorted with artificial impairments. The obtained results show that the proposed method can self-adapt to the environment variation and correctly identify the anomalies. Moreover, the paper shows how the combination of the two approaches can mitigate their drawbacks while enhancing their positive properties.

Keywords: Intelligent sensing, Sensor fusion, Anomaly detection, Cloud-assisted sensing, Internet of Things

1. Introduction

A wireless sensor network (WSN) is a distributed network architecture composed of a set of autonomous electronic devices (network nodes) collecting data from the surrounding environment. Examples of data sources are temperature, humidity, light, noise, electric current, tension, and power.

The market of wireless sensor networks is continuously growing thanks to the technological and computational improvements [1]. At the same time, efficient management techniques are needed for dealing with the network complexity and the huge amount of sensor data [2, 3].

Wireless sensor network devices are typically connected to cloud services through the Internet. Cloud platforms provide the storage and computing infrastructures necessary for archiving and processing the large amount of data generated by sensors [4].

Graphical visualization, statistical analysis, tabular reporting of data sensor are very common applications in the wireless sensor newtwork and in the Internet of Things (IoT) domains.

A challenging research area is the problem of sensor data analysis for automatic anomaly detection [5]. The causes of anomalies are related to several factors. Device running out of power, device deviating from the expected behaviour, device breaking are common causes of anomalies. Nevertheless, even natural deviation of the environmental condition being sensed can be detected as anomaly in the sensed information.

In the last years, anomaly detection has been deeply focused by the academic community due to the extensibility of the problem to various context such as fraud detection and security classifications of users and systems. Most of the approaches regarding anomaly detection are dedicated to the analysis of data streams produced by a *single* device. In this case, a single device is analyzed, by means of different techniques, to understand whether in its behavior an anomaly has occurred or neither.

These techniques are usually based on complex mathematical analysis or statistical techniques applied on data streams [6]. Note, however, that these approaches can be applied on numerical data only. Indeed, the representation of data streams play an important role and several approaches have been proposed in the presence of different data representations.

For instance, in [7] a survey on a graph-based anomaly detection and description is presented: it focuses on providing a general and structured overview of methods for anomaly detection in data represented as graphs and categorized under various settings. Being able to differentiate data representation allows to apply anomaly detection in different domains such as financial auction and social network. In particular, anomaly detection on (or based on) social network has gained an increasing importance [8].

Other approaches apply mathematical or machine learning based analysis on different data levels. This kind of techniques have been applied in intrusion detection in security systems [9] and fraud detection for credit cards [10]. In [11], incoming data packets are compared to fixed patterns in order to identify known behavioral instances. Spatial anomaly detection is analyzed in [12] using neural networks. But anomaly detection algorithms have been developed also in other contexts, such as multivariate time series analysis [13] or biomedical contexts [14]

However, all of these approaches always assume that the data to monitor is homogeneous, thus they are not well suited in the context of heterogeneous sensor networks, due to each node can (possibly) produce different type of data.

An approach for monitoring heterogeneous wireless sensor networks and to identify hidden correlations between heterogeneous sensors has been proposed in [15]. An experimental environment has been designed for testing the proposed technique. The obtained results have shown a strong capability in identifying hidden correlations between sensors together with a good robustness to environment variations. In this paper, we significantly extend the approach proposed in [15] in order to identify anomalies spanning over long periods of time. The capability to identify hidden correlations is exploited also to reduce data and computational requirements of the approach.

Performing anomaly detection offline allows to get accurate results and, being able to work over long term data, allows also to get long term anomaly detection. However, performing anomaly detection just completely offline presents some drawbacks given by the wireless communication scheme, e.g. packet losses and delays, communication bottlenecks yeldied by too much data transmitted from nodes to the central servers, and so on [16]. To avoid these issues, anomaly detection has to be performed also online, directly on the nodes. Most of these approaches require samples of historical data to be kept in the nodes limited memory. Besides that, most of the state-of-the-art online learning algorithms target network organization, usually routing protocols [16]. Few works exist which target directly the measurements data from the sensors and, to the best of our knowledge, they still need a sliding window of historical measurements stored in the node memory to perform anomaly detection. For instance, a sliding window is used together with an ellipsoidal support vector machine (SVM) in [17], with various linear and non-linear machine learning models in [18, 19], and with ensemble methods [20].

To decrease the memory requirements on the wireless nodes for the online detection task, in this work we propose a novel method to perform online anomaly detection, named *Anomaly detection with Generative replay* (AnGe). AnGe can detect anomalies online, by making use of the generative and density estimation capabilities of a deep learning method, i.e. restricted Boltzmann machines, while it does not need to store in the node memory any historical measurements. The online method allows to overcome some of the drawbacks of offline methods outlined above; however, since it does not exploit historical measurements, it is very good at identifying short term anomalies but it may miss some long term ones. Coupling the two methods allows us to mitigate the drawbacks of both approaches, while taking advantage of their best qualities. In fact, the edge

computing based approach can continuously monitor single nodes; as soon as a short term anomaly is detected on the node, the cloud computing based approach is activated in order to carry out a more refined, sensor based, analysis and to check for long term anomalies.

The main contribution of this paper is the proposal of a novel method for automatic anomaly detection based on combining an edge computing based approach exploiting an artificial neural network algorithm and a cloud computing based approach exploiting a multi-parameterized edit distance algorithm.

The paper is organized as follows. Section 2 presents the proposed framework composed by a cloud based and an edge based computing approaches. The experimental analysis and related discussion are reported in Section 3. Finally, Section 4 draws the conclusions.

2. Proposed framework

2.1. Cloud based method for long term anomaly detection

In order to identify long term anomalies, we exploit our recently introduced string similarity metric, called Multi-Parameterized Edit Distance (hereafter, MPED) [15], to measure long term correlations between apparently unrelated data. In fact, given a pair of sensors, MPED is able to identify *hidden* correlations between them even if they measure different kind of events; this allows us to define a method to detect expected correlations first and verify actual correlations during the normal operation of sensors. Moreover, our approach allows us, after a training phase, to concentrate the analysis of correlations just between pairs of data streams thus significantly reducing the amount of computational needs.

In order to show our approach, we first recall basic notions about MPED, then we introduce a formal representation of data streams in terms of chunks of sequences, generated at given time intervals. Finally, we present the formalization of the training phase and of the test phase, used to identify potential anomalies.

2.1.1. Preliminaries: Multi-Parameterized Edit Distance

MPED allows the computation of the minimum edit distance between two strings, provided that finding the optimal matching schema, under a set of constraints, is part of the problem. In order to understand how MPED works, in the following, we briefly recall the theoretical components of MPED.

First of all, the notion of matching schema must be introduced, which is the core ingredient of MPED.

Let Π_1 and Π_2 be two (possibly disjoint) *alphabets* of symbols and let s_1 and s_2 be two strings defined over Π_1 and Π_2, respectively. A matching schema M over Π_1 and Π_2 is a schema representing how different combinations of the alphabets Π_1 and Π_2 can be combined via matching. Intuitively, given two strings s_1 and s_2 defined over Π_1 and Π_2, M states which symbols of s_1 can be considered matching with symbols of s_2. Many-to-many matchings are expressed

with π-partitions, and partitions disallow ambigous matchings. The following definitions introduce M formally.

Definition 2.1 (π-partition)**.** *Given an alphabet* Π *and an integer* π *such that* $0 < \pi \leq |\Pi|$*, a* π-partition *is a partition* Φ^π *of* Π *such that* $0 < |\phi_v| \leq \pi$*, for each* $\phi_v \in \Phi^\pi$*.*

Definition 2.2 ($\langle \pi_1, \pi_2 \rangle$-matching schema)**.** *Given two alphabets* Π_1 *and* Π_2 *and two integers* π_1 *and* π_2*,* a $\langle \pi_1, \pi_2 \rangle$-matching schema *is a function* $M_{\langle \pi_1, \pi_2 \rangle} : \Phi_1^{\pi_1} \times \Phi_2^{\pi_2} \rightarrow \{\text{true}, \text{false}\}$*, where* $\Phi_i^{\pi_i}$ *(*$i \in \{1, 2\}$*) is a* π_i*-partition of* Π_i *and, for each* $\phi_v \in \Phi_1^{\pi_1}$ *(resp.,* $\phi_w \in \Phi_2^{\pi_2}$*), there is at most one* $\phi_w \in \Phi_2^{\pi_2}$ *(resp.,* $\phi_v \in \Phi_1^{\pi_1}$*) such that* $M(\phi_v, \phi_w) = \text{true}$*. This means that all the symbols in* ϕ_v *match with all the ones in* ϕ_w*.* $M(\phi_v, \phi_w) = \text{false}$ *indicates that all the symbols in* ϕ_v *mismatch with all the ones in* ϕ_w*.*

Once the notion of matching schema is available, the definition of MPED is formally introduced by the following definitions.

Definition 2.3 (Transposition)**.** *Let* s_1 *and* s_2 *be two strings defined over the alphabets* Π_1 *and* Π_2*. Let* $-$ *be a symbol not included in* $\Pi_1 \cup \Pi_2$*. Then, a string* $\bar{s}_i$ *over* $\Pi_i \cup \{-\}$ *(*$i \in 1, 2$*) is a* transposition *of* s_i *if* $\bar{s}_i$ *can be obtained from* s_i *by deleting all the occurrences of* $-$*. The set of all the possible transpositions of* s_i *is denoted by* $\mathcal{TR}(s_i)$*.*

Definition 2.4 (Alignment)**.** *An* alignment *for the strings* s_1 *and* s_2 *is a pair* $\langle \bar{s}_1, \bar{s}_2 \rangle$*, where* $\bar{s}_1 \in \mathcal{TR}(s_1)$*,* $\bar{s}_2 \in \mathcal{TR}(s_2)$ *and* $\text{len}(\bar{s}_1) = \text{len}(\bar{s}_2)$*. Here,* $-$ *is meant to denote an insertion/deletion operation performed on* s_1 *or* s_2*.*

Definition 2.5 (Match and distance)**.** *Let* $\langle \bar{s}_1, \bar{s}_2 \rangle$ *be an alignment for* s_1 *and* s_2*, let* $M_{\langle \pi_1, \pi_2, \chi \rangle}$ *be a* $\langle \pi_1, \pi_2, \chi \rangle$*-constrained matching schema over* π*-partitions* $\Phi_1^{\pi_1}$ *and* $\Phi_2^{\pi_2}$ *and the set of constraints* χ*, and let* j *be a position with* $1 \leq j \leq len(\bar{s}_1) = len(\bar{s}_2)$*. We say that* $\langle \bar{s}_1, \bar{s}_2 \rangle$ *has a match at* j *if:*

- $s_1[j] \in \phi_v, s_2[j] \in \phi_w$*,* $\phi_v \in \Phi_1^{\pi_1}$*,* $\phi_w \in \Phi_2^{\pi_2}$ *and* $M_{\langle \pi_1, \pi_2, \chi \rangle}(\phi_v, \phi_w) = \text{true}$*.*

The distance *between* $\bar{s}_1$ *and* $\bar{s}_2$ *under* $M_{\langle \pi_1, \pi_2, \chi \rangle}$ *is the number of positions at which the pair* $\langle \bar{s}_1, \bar{s}_2 \rangle$ *does not have a match.*

Given the previous definitions, we can introduce the notion of *Multi-Parameterized Edit Distance* between two strings s_1 and s_2 as follows:

Definition 2.6 (Multi-Parameterized Edit Distance - MPED)**.** *Let* π_1 *and* π_2 *be two integers such that* $0 < \pi_1 \leq |\Pi_2|$ *and* $0 < \pi_2 \leq |\Pi_1|$*; the Multi-Parameterized Edit Distance between* s_1 *and* s_2 *(*$\mathcal{L}_{\langle \pi_1, \pi_2, \chi \rangle}(s_1, s_2)$*, for short) is the minimum distance that can be obtained with any* $\langle \pi_1, \pi_2, \chi \rangle$*-constrained matching schema and any alignment* $\langle \bar{s}_1, \bar{s}_2 \rangle$*.*

In order to simplify the notation, we will denote by $\mathcal{L}(s_1, s_2)$ the MPED obtained between s_1 and s_2.

2.1.2. Basic definitions for the approach

Let $\mathcal{N}$ be a set of nodes and $\mathcal{S}$ a set of sensors. Each sensor $s \in \mathcal{S}$ is equipped on a node $n \in \mathcal{N}$ which might accommodate several sensors. In order to simplify the notation, we assume that each sensor $s \in \mathcal{S}$ is uniquely identified in the set and, if necessary, function $\gamma : \mathcal{S} \rightarrow \mathcal{N}$ returns the node n the sensor s is equipped on.

A generic sensor s periodically collects data; we define an *observation* as the value v collected by a sensor s in a specific time instant t, and we denote it as $\alpha_s(t)$. We assume t stores the complete timestamp of the collection (date/time).

A certain set of sensors is run for an arbitrary amount of time T; an arbitrary sequence of time instances $t_i, t_{i+1}, \ldots, t_{i+k-1}$ defines an *interval* over which a chunk of data (an ordered sequence of observations) is collected; this must be transformed into a string in order to apply MPED. Moreover, in order to analyze the behaviour of sensors, it is important to organize observations in specific time intervals.

In the application context of the present paper, we analyze data by hours and days. In particular, assume that observations span over a set of days $d \in [1..D]$, and that each day d is subdivided in hours $h \in [1..24]$, given function $\rho(t_i)$ which provides the hour h the time instant t_i belongs to and the function $\delta(t_i)$ which provides the day t_i belongs to, the sequence of time instants belonging to a certain hour h of a certain day d is formalized by:

$$\Upsilon(d, h) = \{t_i \mid t_i \in T, \rho(t_i) = h \; and \; \delta(t_i) = d\}$$

which forms the basis for the construction of strings to be provided to MPED, which is formalized next.

Given a sensor s_i, a day of interest d and an hour of interest h, the corresponding sequence of observations is the ordered sequence:

$$q(s_i, d, h) = \{\Psi(\alpha_{s_i}(t_k)) \mid t_k \in \Upsilon(d, h)\}$$

where function Ψ transforms each single observation in the corresponding symbolic representation.

Clearly observations can be composed over several hours, or several days, if needed.

2.1.3. The workflow of the approach

The workflow of the proposed approach is two-phase: the first phase is devoted to train the system under "normal" operational conditions, in order to understand expected information for each sensor and for each time slot. The second phase applies information learned in the first phase to identify potential anomalies.

Intuitively, one of the novelties of the approach for long term anomaly detection introduced in this paper, relies on the fact that the training phase does not compute expected *values* for the various sensors, but expected *correlations* between sensors. In particular, in the training phase, for each sensor, and for each hour, we identify the so called *mate*-sensor, i.e. the most correlated sensor

among the set for that time slot. This mate is used as a reference during the test phase. In fact, whenever the correlation between the two significantly changes, a potential anomaly can be detected.

It is important to point out that MPED plays a crucial role in this approach, since compared sensors might be heterogeneous and correlations found between mate-sensors might be completely unexpected (for example, light and temperature of sensors positioned in different points in space).

We next formalize the two phases of the approach.

Training phase. The training phase starts by computing the average correlation between each pair of sensors for each hour, within a fixed period of training days D^T. In particular, for each pair of sensors s_i, s_j and each hour $h \in [1..24]$, we define $C(s_i, s_j, h)$ as the average correlation over days in D^T. Formally:

$$\forall s_i, s_j, h \;\; C(s_i, s_j, h) = \operatorname*{avg}_{d \in D^T} \{\mathcal{L}(q(s_i, d, h), q(s_j, d, h))\}.$$

Based on C, for each sensor s_i and each hour h, we can formally define the *mate sensor* $s^*_{i,h}$ of s_i as:

$$s^*_{i,h} = \tau(s_i, h) = \operatorname*{argmax}_{s_j} \{C(s_i, s_j, h)\}.$$

Finally, for each sensor s_i and each hour h, we define the *expected correlation* of sensor s_i at hour h with its *mate* as $\eta(s_i, h) = C(s_i, \tau(s_i, h), h)$.

As an example, Table 1 shows the correlation computed during N days for eight heterogeneous sensors for $h = 1$, i.e. from time 12 : 00 to time 12 : 59. Since in the one hour time interval the sensor A and the sensor B are the most correlated ones, it is reasonable to expect similar levels of correlation for corresponding time intervals in days different from the N considered ones. The mating between sensors that is extracted from the analysis of Table 1 for $h = 1$ is ($A \longleftrightarrow B$, $C \longleftrightarrow D$, $E \longleftrightarrow G$, $F \longleftrightarrow H$). A similar computation is carried out for the other values of h.

Table 1: Example of average correlation between values from eight sensors A, B, C, D, E, F, G, and H during N days for time interval number 1. In boldface the highest correlation between each couple of sensors.

	A	B	C	D	E	F	G	H
A	–	**0.9**	0.3	0.8	0.5	0.3	0.5	0.8
B	**0.9**	–	0.5	0.9	0.5	0.4	0.8	0.2
C	0.3	0.5	–	**0.9**	0.4	0.4	0.6	0.8
D	0.8	0.8	**0.9**	–	0.7	0.5	0.7	0.1
E	0.5	0.5	0.4	0.7	–	0.3	**0.8**	0.4
F	0.3	0.4	0.4	0.5	0.3	–	0.3	**0.6**
G	0.5	0.8	0.6	0.7	**0.8**	0.3	–	0.5
H	0.8	0.2	0.8	0.1	0.4	**0.6**	0.5	–

Test phase. The test phase starts after the training phase is completed. Here, each sensor has been already associated with its mate. Thus, the test phase works as follows.

Given a threshold $\theta \in [0, 1]$, for each sensor s_i, for each day d, and for each hour h, we compute the *actual correlation*, denoted by $\chi(s_i, h, d)$, as the correlation between sensors s_i and its mate $\tau(s_i, h)$. Formally:

$$\forall s_i, d, h \;\; \chi(s_i, h, d) = \mathcal{L}(q(s_i, d, h), q(\tau(s_i, h), d, h)).$$

Now, a potentially anomalous behavior is detected when the actual correlation of s_i with its mate, significantly differs from the expected one:

$$|\chi(s_i, h, d) - \eta(s_i, h)| > \theta.$$

In order to reduce false positives, an alert is issued if this condition is verified for an average difference greater than the threshold for a fixed number of hours H^*. Formally:

$$alert(s_i, h, d) \leftarrow \underset{h' \in [h-H^*, h]}{\text{avg}} \{|\chi(s_i, h', d) - \eta(s_i, h')|\} > \theta.$$

2.2. Edge based method for short term anomaly detection

In order to perform edge based anomaly detection we contribute by exploiting the possibility of performing online unsupervised learning in each node with Artificial Neural Networks (ANN). This ensures a fully decentralized method to detect anomalies, each node being completely independent of the others, and acts as a prefilter on the data that are transmitted further to the cloud leading to better network management. At the same time, it ensures data fusion for one node, i.e. the measurements of all sensors belonging to one node are treated together at each time step t to detect anomalies.

However, online learning with artificial neural networks is in many cases difficult due to the need of storing and relearning large amount of previous experiences to avoid catastrophic forgetting. For a standard computer this is a solvable issue. At the same time, in the world of low-resources devices, these excessive memory requirements, necessary to explicitly store previous observations, represent a big challenge. To overpass it, in this paper, we make use of a novel concept proposed by us in [21] and developed further in [22, 23], namely *generative replay*. Generative replay uses the generative capabilities of generative artificial neural network models to generate approximations of past experiences, instead of recording them, as experience replay does. Thus, the generative model can be trained online, and does not require the system to store any of the observed data points, this being a perfect scenario for anomaly detection in wireless sensor nodes. More exactly, in this paper, we use a generative model called Restricted Boltzmann Machine (RBM) [24] trained with *Online Contrastive Divergence with Generative Replay* (OCD_{GR}), and named RBM_{OCD} [21].

As an auxiliary contribution of this paper, due to the fact that RBM$_{OCD}$ as proposed in [21] is capable just to learn data distributions in an online manner, herein we propose an extension of it to perform online anomaly detection. Further on, in Section 2.2.1, RBM$_{OCD}$ and similarity metrics with RBMs are briefly summarized for the benefit on the non-specialist reader, while in Section 2.2.2, the new proposed method for online anomaly detection is introduced.

2.2.1. RBM$_{OCD}$ background

RBMs have been introduced in [24] as a powerful model to learn a probability distribution over its inputs. Formally, RBMs are generative stochastic neural networks with two binary layers: the hidden layer $\mathbf{h} = [h_1, h_2, .., h_{n_h}]$, and the visible layer $\mathbf{v} = [v_1, v_2, .., v_{n_v}]$, where n_h and n_v are the numbers of hidden neurons and visible neurons, respectively. In comparison with the original Boltzmann machine [25], the RBM architecture is restricted to be a complete bipartite graph between the hidden and visible layers, disallowing intra-layer connections between the units. The energy function of an RBM for any state $\{\mathbf{v}, \mathbf{h}\}$ is computed by summing over all possible interactions between neurons, weights, and biases as follows:

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{a}^T\mathbf{v} - \mathbf{b}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v}, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{n_h \times n_v}$ is the weighted adjacency matrix for the bipartite connections between the visible and hidden layers, and $\mathbf{a} \in \mathbb{R}^{n_v}$ and $\mathbf{b} \in \mathbb{R}^{n_h}$ are vectors containing the biases for the visible and hidden neurons, respectively. Functionally, the visible layer encodes the data, while the hidden layer increases the learning capacity of the RBM model by enlarging the class of distributions that can be represented to an arbitrary complexity [26]. The activations of the hidden or visible layers are generated by sampling from a sigmoid $\mathcal{S}(\cdot)$ according to: $P(\mathbf{h}) = \mathcal{S}(\mathbf{b} + \mathbf{W}\mathbf{v})$ and $P(\mathbf{v}) = \mathcal{S}(\mathbf{a} + \mathbf{W}^T h)$.

Motivated by the facts that: (1) hippocampal replay [27] in the human brain does not recall previous observations explicitly, but instead it generates approximate reconstructions of the past experiences for recall, and (2) RBMs can generate good samples of the incorporated data distribution via Gibbs sampling [28], in [21] we proposed RBM$_{OCD}$. Intuitively, RBM$_{OCD}$ uses generated samples by itself (instead of recalling previous observations from stored memory) during the online training process. Thus, the RBM model can retain knowledge of past observations while learning new ones. The interested reader is referred to [21] for a detailed discussion on RBM$_{OCD}$.

Same as any other RBM variant trained offline [29], during learning, RBM$_{OCD}$ minimizes the error between the reconstructed version of the input data, denoted further $\hat{\mathbf{v}}$, and the input data itself, $\mathbf{v}$. The reconstructed version ($\hat{\mathbf{x}}$) of a given input data point ($\mathbf{x}$), is computed by performing a one step Gibbs sampling starting from the original data point clamped to the visible neurons ($\mathbf{v}$), then by inferring the hidden neurons activations ($\mathbf{h}$), and then by inferring the visible neurons activations $\hat{\mathbf{v}}$ given the hidden neurons activations. The values of the latter one activations give $\hat{\mathbf{x}}$. Moreover, in our previous work [30], we showed

that the error computed between a testing data point and its reconstructed version given by an already trained offline RBM, can be used as a similarity metric. More exactly, it can say how far is the testing data point from the training data distribution. The interested reader is referred to [31, 30, 32, 33] for more thorough discussions.

2.2.2. Online anomaly detection with RBM_{OCD}

In this subsection, by making use of the above discussed concepts, we propose a novel method to perform online anomaly detection based on RBM_{OCD}, dubbed *Anomaly detection with Generative replay* (AnGe). AnGe works as follows.

The sensor measurements occur at specific time intervals. At any specific time t, new measurements are given by all sensors of a node and they are collected in a vector $\mathbf{x}^t$. Starting with $t = 0$ in a continuous loop, an RBM_{OCD}^t is trained online to model all measurements made until t. At the same time, we know [31, 30, 32, 33] that the reconstruction error of unseen data points with an offline trained RBM gives a similarity metric with respect to the training data points. Thus, our assumption is that if at the specific time t an anomaly happens then the reconstruction error of the measurements $\mathbf{x}^t$ will be very different from the reconstruction error of $\mathbf{x}^{t-1}$, both being reconstructed with RBM_{OCD}^{t-1}. As large is this difference, as higher is the chance of an anomaly. To quantify, let's note this metric m_{AnGe}. Using Root Mean Square Error (RMSE) for the reconstruction error, it can be computed as follows:

$$m_{AnGe} = \sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}(\widehat{\mathbf{x}_i^t} - \mathbf{x}_i^t)^2} - \sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}(\widehat{\mathbf{x}_i^{t-1}} - \mathbf{x}_i^{t-1})^2} \tag{2}$$

Further on, if more similar measurements with $\mathbf{x}^t$ will occur, $\text{RBM}_{OCD}^{>t}$ will gradually enlarge its encoded data distribution to incorporate also these types of measurements and to not consider them an anomaly anymore. It worth to be highlighted that AnGe needs to store in the device memory just the RBM_{OCD} weighted connections. This makes it a very suitable method to perform online anomaly detection in wireless nodes.

3. Experimental analysis

3.1. Sensor network deployment setup

For the experimentation proposed, eight WSN nodes have been deployed in a floor at DIMES, cubo 41C, University of Calabria, Italy. The used WSN nodes consist of TelosB motes [34] running TinyOS 2.1.2 [35]. Such nodes have been organized in a multi-hop wireless sensor network by using the Building Management Framework (BMF) [36].

The BMF is a domain-specific framework specifically designed to efficiently manage heterogeneous WSNs that have been scattered in buildings. Through the BMF it is possible to quickly prototype WSN applications, realize smart

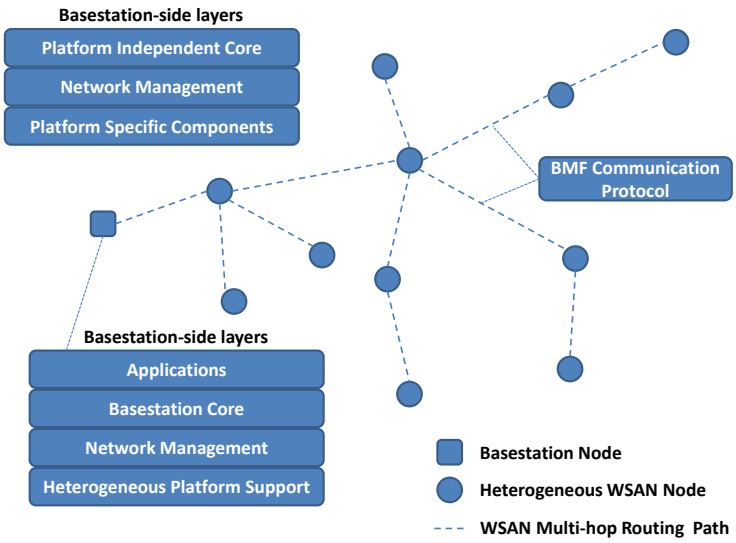


Figure 1: A BMF Network example.

sensing/actuation, and capture, by using specific abstractions, the floor plan of a building. BMF WSNs are controlled through a basestation that can be seen as both a data collector and a network configurator. BMF nodes communicate by using the BMF Communication Protocol, namely an application level protocol built on the Collection Tree and Dissemination Protocols [37][38].

In Figure 1, an example of a BMF network together with the BMF layers at both basestation and node sides is portrayed.

At basestation side, the BMF allows: (layer 1) the use of heterogeneous sensor platforms (i.e. TelosB, Tyndall, Shimmer, SunSPOT); (layer 2) a flexible network management through configurations packets sent according to the BMF Communication Protocol [36]; (layer 3) the efficient managing of a WSN by providing specific functionalities (e.g. group nodes organization, schedule of sensing or actuation requests), and (layer 4) to support the realization of applications on top of the Basestation Core. At node side, the BMF provides: (layer 1) platform-specific components on which the BMF platform independent components are built; (layer 2) functionalities to permit communication among nodes and with the basestation, and (layer 3) specific platform-independent core functionalities such as signal processing and multi-request scheduling on the nodes.

The BMF has been here used to collect every second data from light, temperature and humidity sensors, compute on the nodes the average on such data, and to send the results every minute to a BMF basestation. The BMF basestation has been enhanced with a specific filter to clean redundant packets received from the WSN and to mask packet losses.

Figure 2 shows all the nodes deployed and their location on the floor plan of the building involved. In particular, based on their location, the deployed nodes have been grouped in pairs:

- *nodes 1 and 124* are stuck on the window of an office. These nodes can be reached by direct sunlight;
- *nodes 17 and 27* are placed over a bookcase in an air conditioned office. Such nodes are less influenced, with respect to nodes 1 and 124, by direct sunlight;
- *nodes 25 and 31* over a desk in an air conditioned and artificially illuminated laboratory.

The experimental tests that have been carried out and that span over 27 days, are divided in three parts, of 9 days each:

- In the first part all the nodes worked in a normal situation (no induced interferences) and are mains powered.
- In the second part, some interferences are introduced at the nodes 1, 17, 31, and 5. In particular, node 1 has been covered with a thick sheet of paper and a bag full of silicon has been placed close to it; a lighted bulb has been located adjacently to nodes 17 and 31; a bag full of silicon has been posed close to the node 5.
- In the third part, no node has been subject to interferences. However, nodes 1, 17, 25, and 28 have been battery powered.

In the deployment of the experiments, we first carried out the training phase of the approach for long term anomaly detection defined in Section 2.1.3 by setting the fixed period of training days D^T to the first three days of nodes total acquisition time. In the fixed period, nodes worked in a normal situation with no external interferences and were mains powered. Raw data from sensors are shown in Figures 3-5. In this work, we set a threshold $\theta = 0.25$ for the test phase and the parameter H^* defined in Section 2.1.3 has been set to 6 hours.

3.2. Short-term anomaly experimental analysis

In this subsection, we analyze the behavior of the online anomaly detection algorithm, i.e. AnGe, described in Section 2.2. We run the algorithm for each node separately, considering the measurements of all sensors for a node.

The RBM_{OCD} model was set to have 3 visible neurons and 10 hidden neurons. This yields a total of 43 parameters which have to be stored in the memory. The model parameters have been updated after each measurement in an online and continuous manner. Before each update three samples were generated by the current model to avoid catastrophic forgetting during the learning of the new data measured.

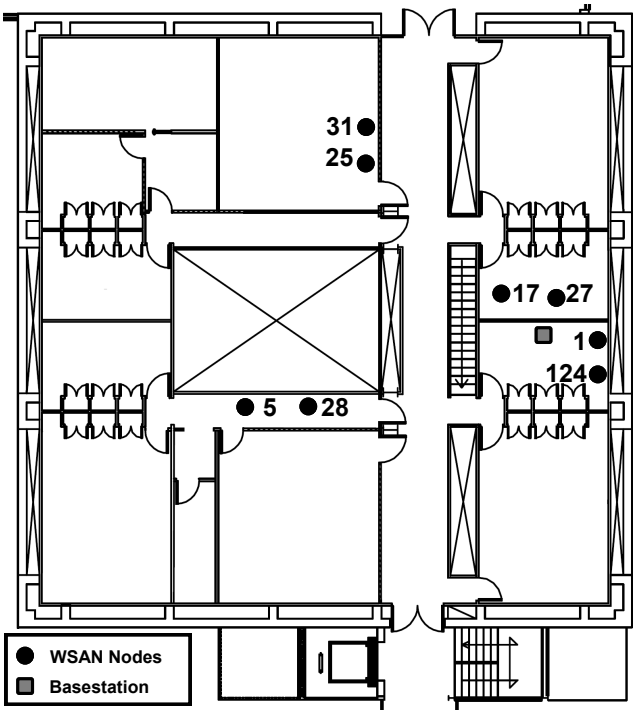
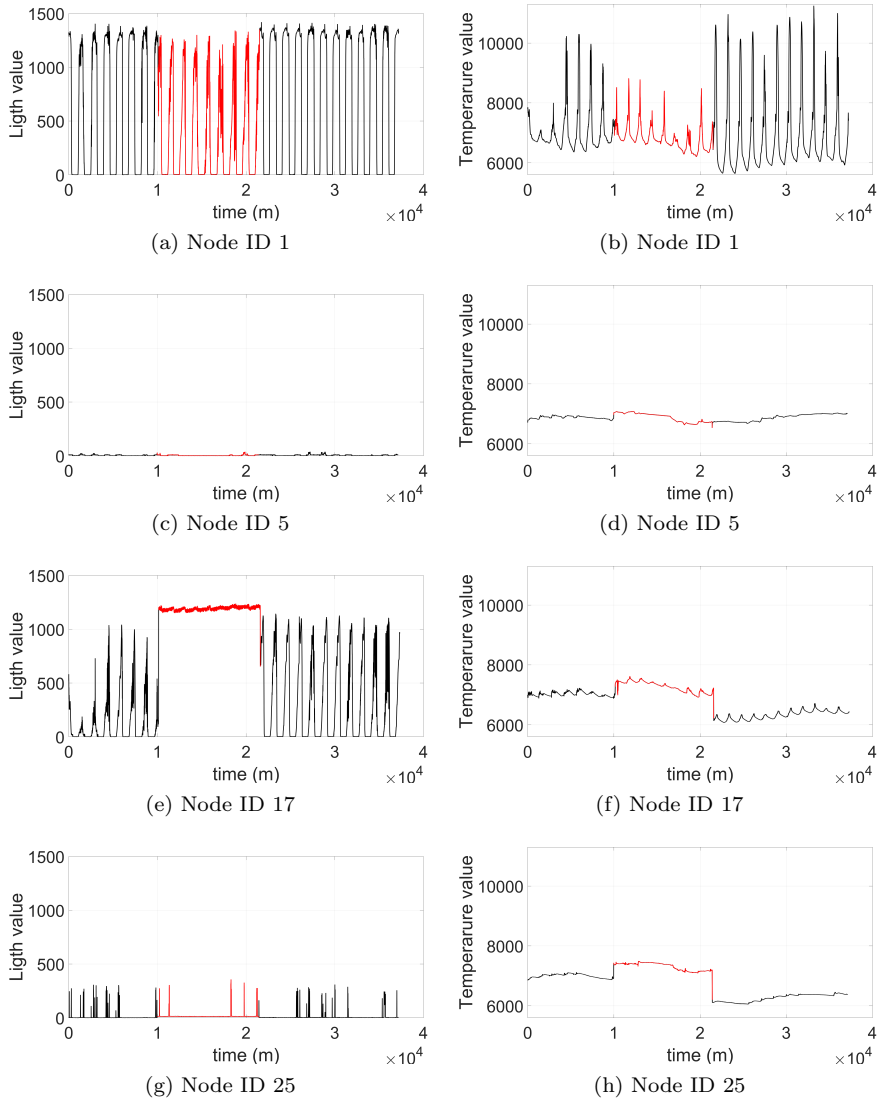


Figure 2: Floor plan and nodes with corresponding identifiers for the experimental analysis of the proposed framework.

Figure 6 shows for each node separately how AnGe is capable to detect anomalies at each time step. For instance, let us consider the Subplot 6(c) which corresponds to the node 17. Usually, m_{AnGe} is very close to zero suggesting that there are no anomalies at that time step, while sometimes it is very far from zero suggesting strong anomalies, e.g. around the 8000 minute m_{AnGe} shows high oscillations and values ranging between -100 and +100. This is exactly the moment when the node 17 starts to be exposed to artificial interferences, i.e. a light bulb in its physical neighborhood. This is reflected by the new pattern of the sensor measurements. Further on, a bit before the 20000 minute m_{AnGe} shows strongly again the possible apparition of an anomaly, reaching a value of -500. This is exactly the moment when the light bulb was removed from the neighborhood of the node 17. Similarly, it can be clearly observed for the other nodes which have been exposed to artificial interferences, i.e. 1, 31, and 5, how AnGe detects the artificially introduced anomalies. Moreover, it is interesting to see how AnGe corresponding to the nodes which were not exposed directly to the artificial interferences, but which were close enough to the nodes with artificial interferences, can also detect them. A more spiky behavior of AnGe can be observed for nodes 1 and 124. These can be explained by the fact that they were exposed to several unknown interferences as their environment was



(a) Node ID 1 (b) Node ID 1

(c) Node ID 5 (d) Node ID 5

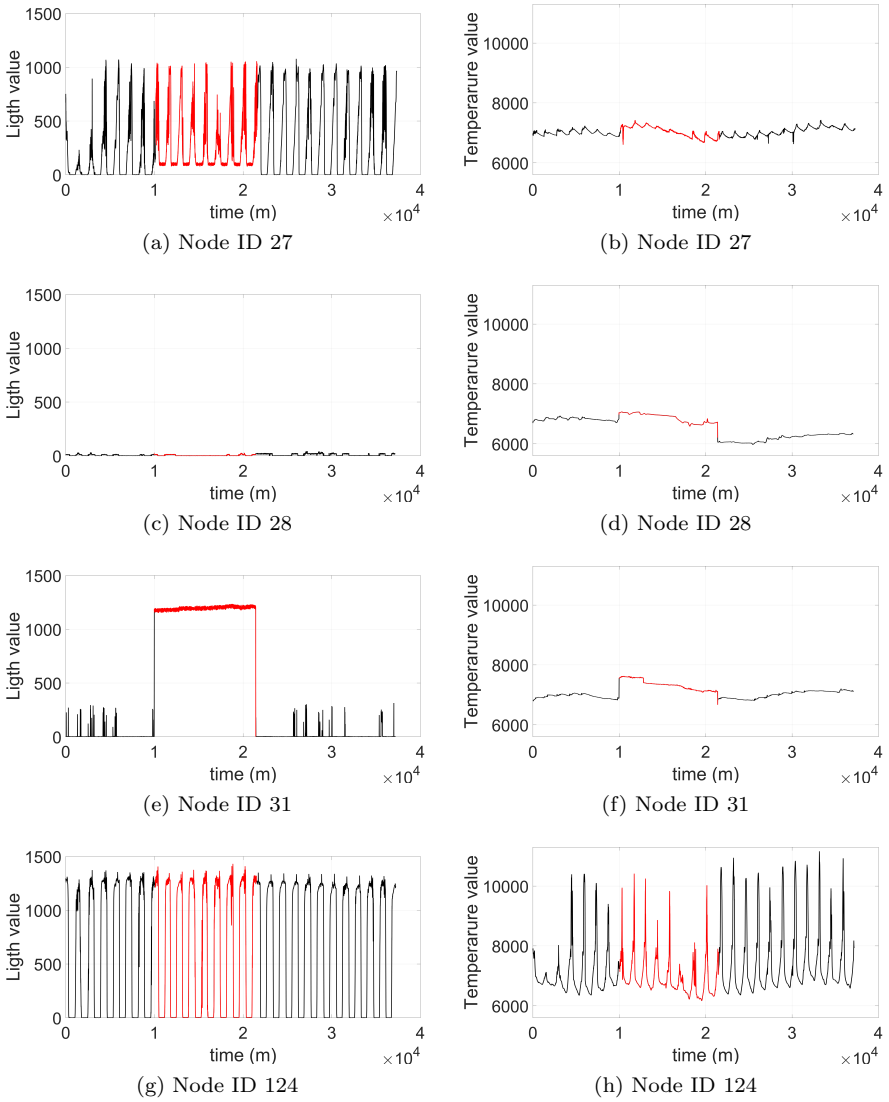
(e) Node ID 17 (f) Node ID 17

(g) Node ID 25 (h) Node ID 25

Figure 3: Light and temperature raw sensor data for node identifiers 1, 5, 17, and 25. The temporal window corresponding to the application of interferences to sensors 1, 5, 17 is highlighted in red for all the plots.

not perfectly controlled (i.e. direct exposure to sunlight).

It worths to be noted that in this paper we did not consider necessary to put thresholds on m_{AnGe} values as our goal was to show how AnGe is capable to detect big, but also small, changes in measurements patterns. If one would use thresholds then it could easily control the sensitivity of AnGe based on the



(a) Node ID 27 (b) Node ID 27

(c) Node ID 28 (d) Node ID 28

(e) Node ID 31 (f) Node ID 31

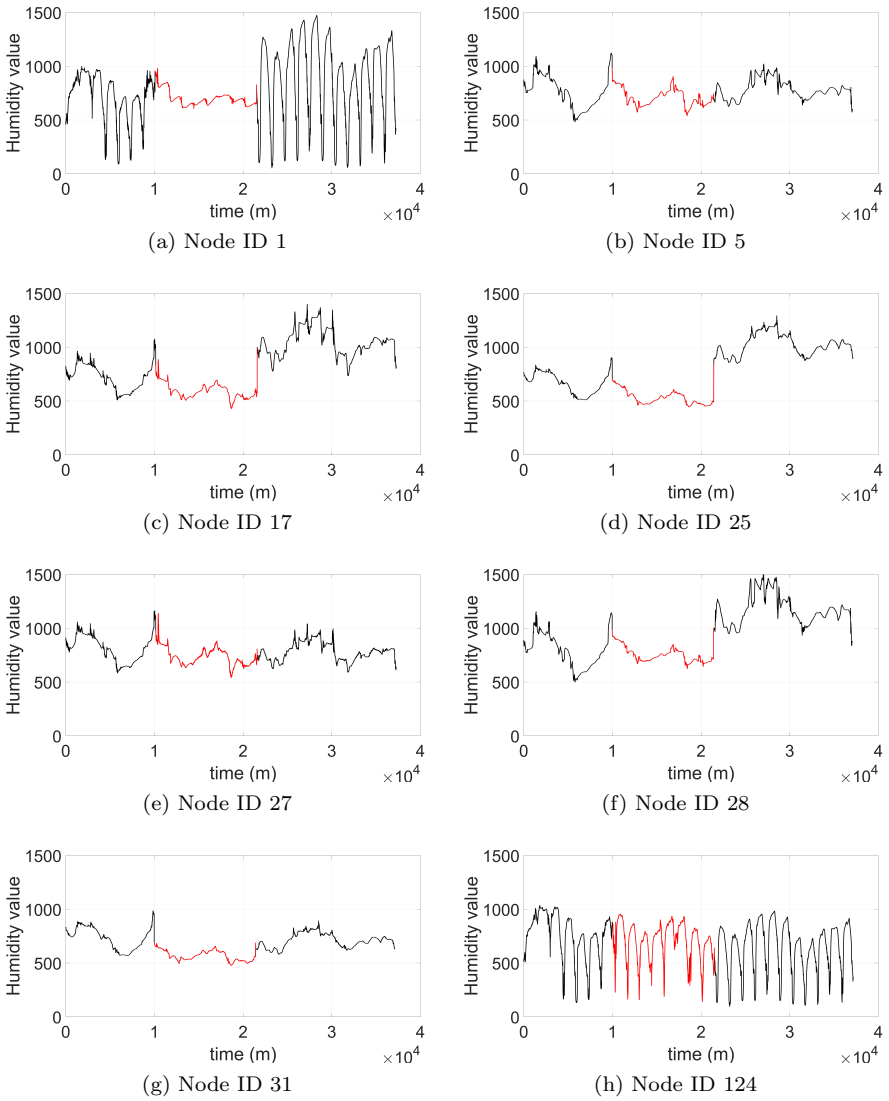
(g) Node ID 124 (h) Node ID 124

Figure 4: Light and temperature raw sensor data for node identifiers 27, 28, 31, and 124. The temporal window corresponding to the application of interferences to sensors 1, 5, 17 is highlighted in red for all the plots.

application requirements.

3.3. Long-term anomaly experimental analysis

Figures 7, 8, and 9 show the values of $|\chi(s_i, h', d) - \eta(s_i, h')|$, defined in Section 2.1.3, for light, temperature and humidity sensors, respectively. In



(a) Node ID 1 (b) Node ID 5

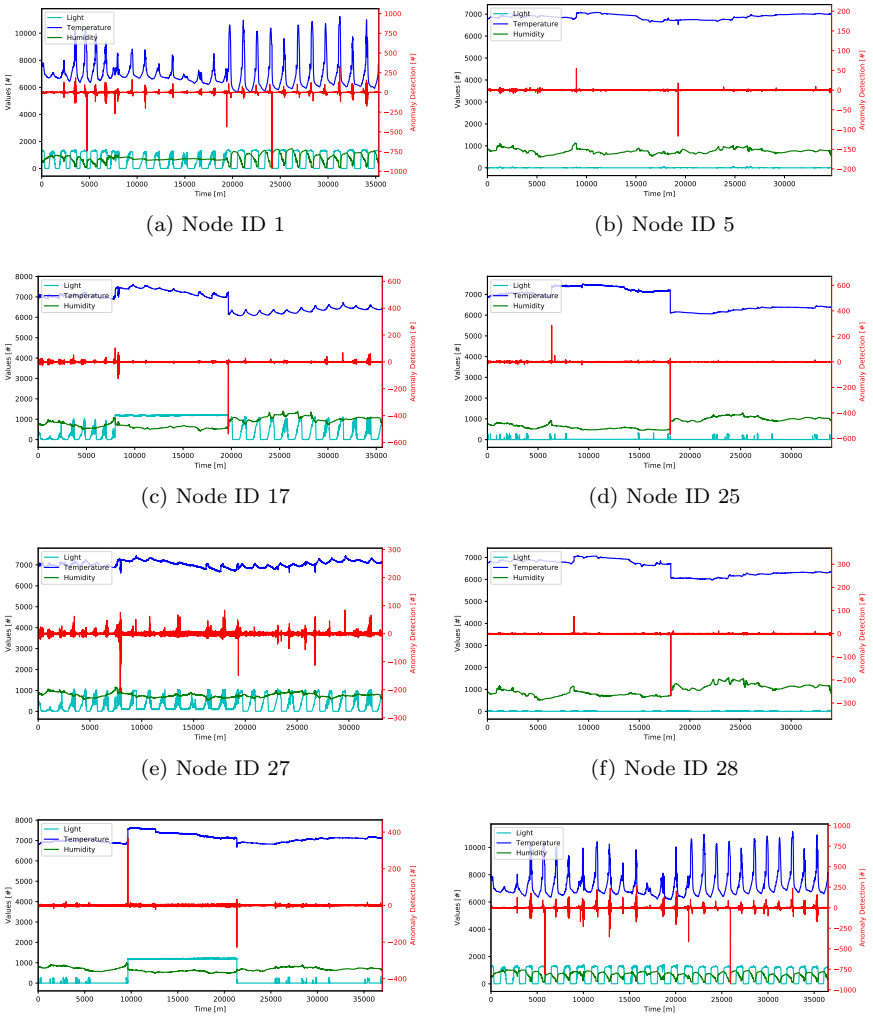
(c) Node ID 17 (d) Node ID 25

(e) Node ID 27 (f) Node ID 28

(g) Node ID 31 (h) Node ID 124

Figure 5: Humidity raw sensor data for node identifiers 1, 5, 17, 25, 27, 28, 31, and 124. The temporal window corresponding to the application of interferences to sensors 1, 5, 17, and 31 is highlighted in red for all the plots.

order to simplify the presentation, only values greater than the threshold are shown; these would correspond to an alert. In this case, setting a threshold was important since the formula measures even smooth differences between expected and computed values and, consequently, without a threshold it would have been difficult to read the graphs. Obviously, also in this case, an accurate tuning of



(a) Node ID 1

(b) Node ID 5

(c) Node ID 17

(d) Node ID 25

(e) Node ID 27

(f) Node ID 28

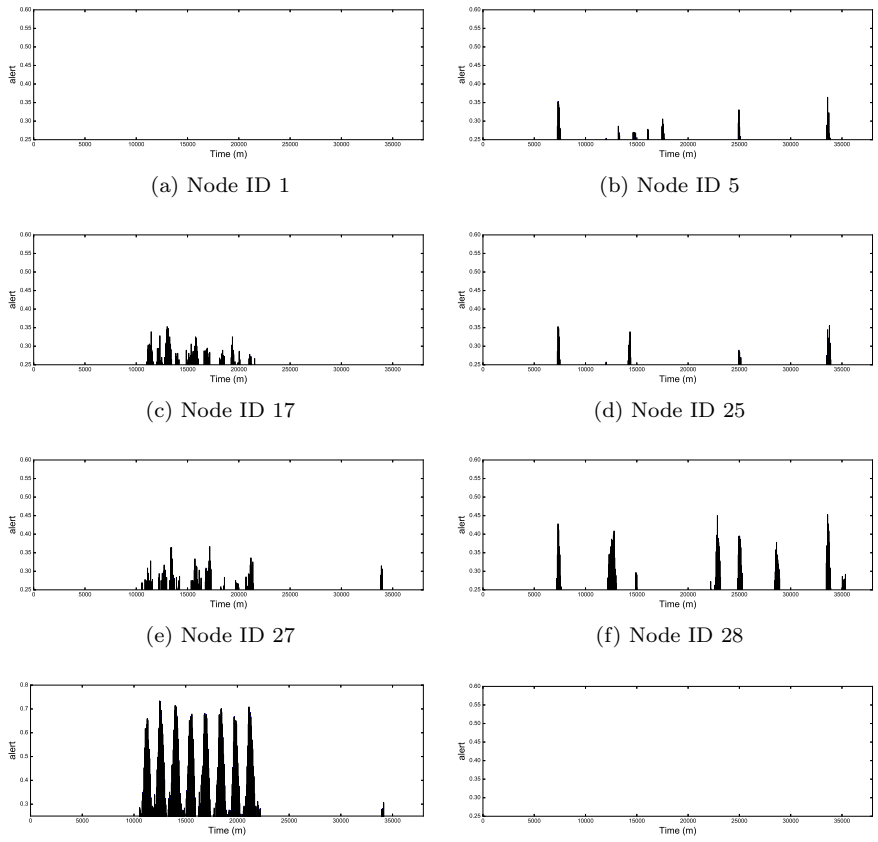
(g) Node ID 31

(h) Node ID 124

Figure 6: Short-term anomaly detection. Each subplot reflects how our proposed method, Anomaly detection with Generative Replay (AnGe), detects anomalies on a specific node. The x-axes represent the time, the left y-axes show the sensors measurements, while the right y-axes (red) show the values of m_{AnGe}.

the threshold could easily control the sensitivity of the approach.

Let us first consider light values. In particular, as far as node 1, it is interesting to observe that, even if an artificial interference was added, no long-term anomaly is alerted. This result is actually correct. In fact, the thick sheet of paper added in front of the sensor reduces the amount of light perceived by the sensor, but it does not prevent it to detect external light variations over



(a) Node ID 1 (b) Node ID 5

(c) Node ID 17 (d) Node ID 25

(e) Node ID 27 (f) Node ID 28

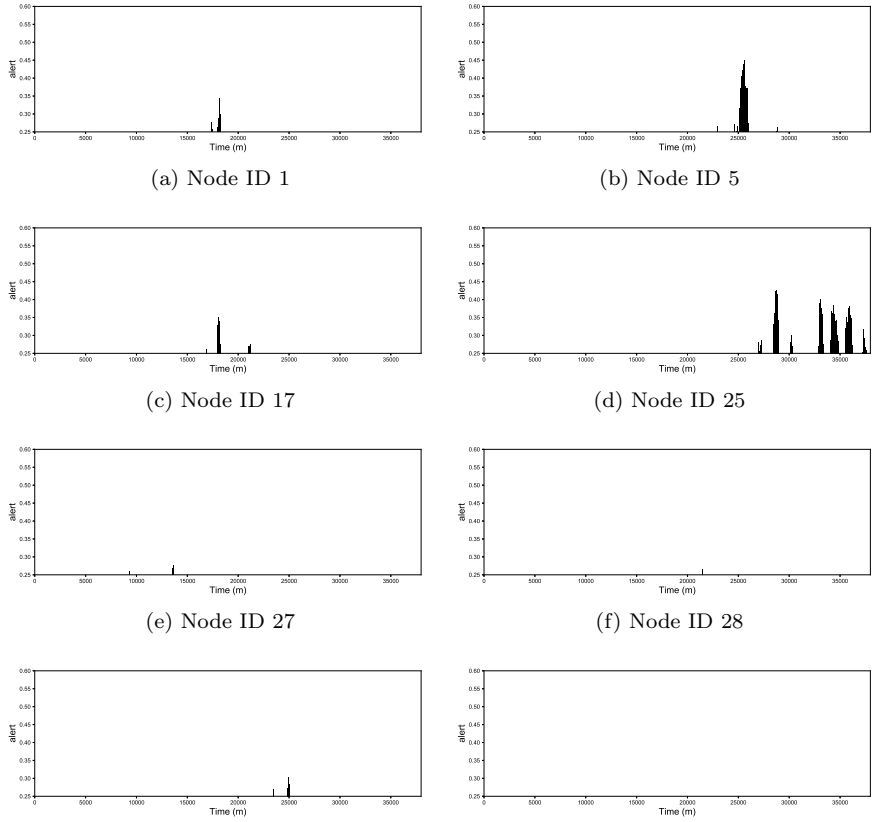
(g) Node ID 31 (h) Node ID 124

Figure 7: Alerts (with over-threshold values) for long-term anomaly detection of light sensors. The x-axis represents the time, the y-axis indicates the value of $|\chi(s_i, h', d) - \eta(s_i, h')|$.

long terms. This is also consistent with the result obtained by the short term approach, which identifies many small environment interferences. On the contrary, nodes 17 and 31, which were disturbed by a lighted bulb, became almost unable to detect light variations (see Figures 3 and 4); and in fact a long term alert during all the test period is fired.

It is interesting to stress that, in a possible application of the short-long term combined approach, the short term method activates the long term one, which may confirm the persistence of an anomalous situation or it may categorize the alert just as occasional.

Interestingly, node 124 is completely unaffected by long-term anomalies; this is also right because it did not receive external interferences and the interference on the adjacent sensor is totally local (a sheet of paper). The same does not hold for nodes 25 and 27 which are near to sensors disturbed by a lighted bulb



(a) Node ID 1 (b) Node ID 5

(c) Node ID 17 (d) Node ID 25

(e) Node ID 27 (f) Node ID 28

(g) Node ID 31 (h) Node ID 124

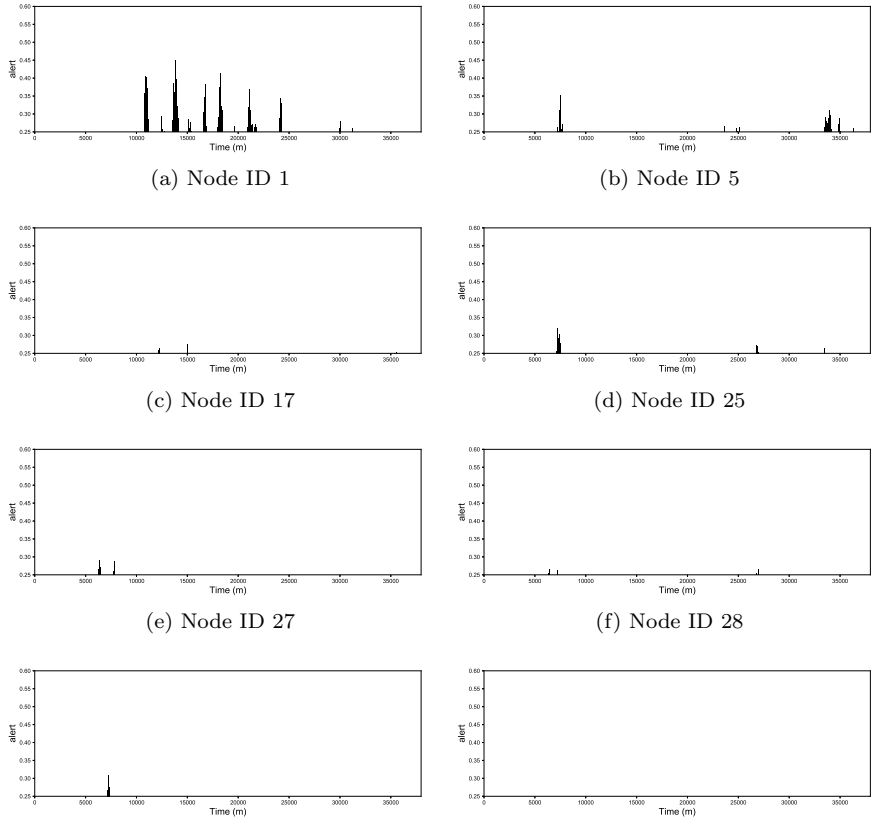
Figure 8: Alerts (with over-threshold values) for long-term anomaly detection of temperature sensors. The x-axis represents the time, the y-axis indicates the value of $|\chi(s_i, h', d) - \eta(s_i, h')|$.

(17 and 31); as a consequence, they are slightly affected too. This result is again consistent with the results of the short term approach.

Finally, as far as light results for nodes 5 and 28 are concerned, we can observe some spikes over all the period, but not particularly constant to motivate a long term anomaly, especially during the artificial interference. This can be motivated by both the fact that they are positioned in a corridor which generates highly irregular data and by observing that, in this case, interference is about humidity caused by the bag full of silicon.

Results for temperature show no particular alerts, except for node 25. This is again consistent, since no artificial interference on temperature was actually introduced, and the alerts on node 25 correspond to the last part of the experiment, when the node was battery powered.

Finally, as far as humidity is concerned, we observe consistent long term alerts only on node 1, where a silicon bag was placed close to it. As for node



(a) Node ID 1 (b) Node ID 5

(c) Node ID 17 (d) Node ID 25

(e) Node ID 27 (f) Node ID 28

(g) Node ID 31 (h) Node ID 124

Figure 9: Alerts (with over-threshold values) for long-term anomaly detection of humidity sensors. The x-axis represents the time, the y-axis indicates the value of $|\chi(s_i, h', d) - \eta(s_i, h')|$.

5, disturbed by the other silicon bag, we observe no long term alerts. If we observe the raw data for humidity shown in Figure 5 we may actually observe no particular variations in trend values. Again, this result is consistent also with short term analysis, which was able to point out the time instants when the bag was put/removed beside the sensor.

3.4. Discussion

The short-term approach clearly identifies potential anomalies signaled by the maxima values of $|m_{AnGe}|$ (Figure 6(a)) for node 1, the node exposed to a thick sheet of paper. The long-term approach for node 1 signals an anomaly for humidity only (Figure 9(a)). As explained in the previous section, this is consistent.

Node 124, the node close to node 1 but not exposed to any impairments, experiences several short term alerts (Figure 6(h)) but no long term ones (Figures

7(h), 8(h), 9(h)). As a matter of facts, by analyzing the patterns of short term alerts of nodes 1 and 124 we observe that they are almost identical, similarly to the overall trends of temperature, humidity and light measured by both sensors. It can be then concluded that short term alerts were issued by the environment.

As far as node 5 is concerned, which was exposed to a bag full of silicon, the short term approach issues two spikes for $|m_{AnGe}|$ (Figure 6(b)), but no consistent long term anomaly is issued (Figures 7(b), 8(b), 9(b)). As a matter of facts, the short term approach identifies the moments when the bag was placed and removed, but this did not alter the measurements for humidity, as shown in Figure 5.

Node 5 is close to node 28, which experiences similar behavior on short term analysis (cfr Figures 6(d) and 6(f)). However, only small alerts on light for long term anomalies are issued for this node (Figure 7(f)), and these are mostly outside the artificial interference period.

For node 17, disturbed with a lighted bulb, the short term approach properly identifies the beginning and the end of the interference period (see Figure 6(c)) and the long term approach confirms the anomaly for light sensor (see Figure 7(c)) while issuing no alerts for temperature and humidity (Figures 8(c) and 9(c)). A similar behavior is observed on node 27 (Figures 6(e), 7(e), 8(e), 9(e)) which was close to node 17 and, consequently, also influenced by the light of the bulb.

Also for node 31, the other node influenced by a lighted bulb, the approach properly identifies the interference, with start and end points identified by the short term approach (Figure 6(g)) and interference period identified on light sensor by the long term approach (Figure 7(g)). In this case, for node 25, the one close to node 31, the short term approach issues alerts (Figure 6(d)) which are not confirmed by the long term approach (Figures 7(d), 8(d), 9(d)), probably because the area where the nodes were positioned was much bigger than the area where nodes 17 and 27 were placed (see Figure 2) and consequently, node 25 was less influenced by the nearby light on node 31.

Almost no alerts are issued in the period when the nodes were battery powered. As a matter of facts, looking at raw data shown in Figures 3, 4, and 5, no real variations in sensed data can be observed in this case.

Summarizing the overall results, we can observe that experiments confirm the intuition about the different nature of anomalies detected by the two approaches. These can be seen as complementary tools for anomaly detection. Both correctly detect real anomalies at different stages. However, both are affected by false positive anomaly detection phenomena. This problem can be significantly reduced by using the short term approach to "trigger" long term observations, which can also drill down the analysis from nodes to single sensors.

4. Conclusions

Automatic anomaly detection in heterogeneous wireless sensor networks is a very challenging task. The signals captured by the sensors are affected by natu-

ral environmental variations that can mask signals variations caused by anomalies. A huge amount of information is captured by wireless sensor networks and there is a need to optimize the data analysis problem devising algorithms for local data preprocessing aiming at reducing the amount of data to be transferred for further processing.

The approach proposed in this paper explores how a combination of short-long term algorithms can correctly identify anomalies in a wireless sensor network. The proposed short term approach has shown good performances for a local identification of potential anomalies and identifying temporal windows of potential interest to be transferred to a cloud service for storage and further long term analysis. The long term approach has shown good performances for identifying the temporal windows affected by anomalies. Nodes close to each other can results in signaling a double alert nevertheless the impact of such false positive result is not so relevant considering that an eventual manual intervention will affect a single location.

Overall we demonstrated how a combined use of short-long term approaches may reduce the drawbacks of both, i.e. false positives and computational requirements, while taking advantage of the best qualities of both, i.e. timeliness and accuracy.

As far as future work is concerned, we plan to improve the coupling of the two approaches by automatizing the process and fine-tuning the thresholds. Moreover, the approach can also be improved to detect other kind of anomalies that currently can not be detected, like slowly changing values of sensed data.

Acknowledgement

This work was partially supported by the Italian Ministry for Economic Development (MISE) under the project "Smarter Solutions in the Big Data World", funded within the call "HORIZON2020" PON I&C 2014-2020.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer networks 38 (4) (2002) 393–422.

[2] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, R. Jafari, Enabling effective programming and flexible management of efficient body sensor network applications, IEEE Transactions on Human-Machine Systems 43 (1) (2013) 115–133. doi:10.1109/TSMCC.2012.2215852.

[3] G. Fortino, A. Guerrieri, G. M. O'Hare, A. Ruzzelli, A flexible building management framework based on wireless sensor and actuator networks, Journal of Network and Computer Applications 35 (6) (2012) 1934–1952.

[4] J. A. Stankovic, When sensor and actuator networks cover the world, ETRI journal 30 (5) (2008) 627–633.

[5] H. H. Bosman, A. Liotta, G. Iacca, H. Wortche, Anomaly detection in sensor systems using lightweight machine learning, in: Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, IEEE, 2013, pp. 7–13.

[6] Y. Zhang, J. Jiang, Bibliographical review on reconfigurable fault-tolerant control systems, Annual reviews in control 32 (2) (2008) 229–252.

[7] L. Akoglu, H. Tong, D. Koutra, Graph based anomaly detection and description: a survey, Data Mining and Knowledge Discovery 29 (3) (2015) 626–688. doi:10.1007/s10618-014-0365-y.
URL https://doi.org/10.1007/s10618-014-0365-y

[8] D. Savage, X. Zhang, X. Yu, P. Chou, Q. Wang, Anomaly detection in online social networks, Social Networks 39 (2014) 62 – 70. doi:https://doi.org/10.1016/j.socnet.2014.05.002.

[9] P. Garca-Teodoro, J. Daz-Verdejo, G. Maci-Fernndez, E. Vzquez, Anomaly-based network intrusion detection: Techniques, systems and challenges, Computers & Security 28 (1) (2009) 18 – 28. doi:https://doi.org/10.1016/j.cose.2008.08.003.

[10] C. Phua, V. Lee, K. Smith, R. Gayler, A comprehensive survey of data mining-based fraud detection research, arXiv preprint arXiv:1009.6119.

[11] T. Ahmed, M. Coates, A. Lakhina, Multivariate online anomaly detection using kernel recursive least squares, in: IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, 2007, pp. 625–633. doi:10.1109/INFCOM.2007.79.

[12] H. H. W. J. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, A. Liotta, Spatial anomaly detection in sensor networks using neighborhood information, Information Fusion 33 (2017) 41–56.

[13] F. Serdio, E. Lughofer, K. Pichler, T. Buchegger, M. Pichler, H. Efendic, Fault detection in multi-sensor networks based on multivariate time-series models and orthogonal transformations, Information Fusion 20 (2014) 272–291.

[14] J. Greensmith, W. Aickelin, G. Tedesco, Information fusion for anomaly detection with the dendritic cell algorithm, Information Fusion 11 (1) (2010) 21–34.

[15] F. Cauteruccio, G. Fortino, A. Guerrieri, G. Terracina, Discovery of hidden correlations between heterogeneous wireless sensor data streams, in: International Conference on Internet and Distributed Computing Systems, Springer, 2014, pp. 383–395.

[16] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wrtche, A. Liotta, Spatial anomaly detection in sensor networks using neighborhood information, Information Fusion 33 (2017) 41 – 56. doi:https://doi.org/10.1016/j.inffus.2016.04.007.

[17] Y. Zhang, N. Meratnia, P. J. Havinga, Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine, Ad Hoc Networks 11 (3) (2013) 1062 – 1074. doi:https://doi.org/10.1016/j.adhoc.2012.11.001.

[18] H. H. W. J. Bosman, A. Liotta, G. Iacca, H. J. Wrtche, Anomaly detection in sensor systems using lightweight machine learning, in: 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 7–13. doi:10.1109/SMC.2013.9.

[19] H. H. W. J. Bosman, A. Liotta, G. Iacca, H. J. Wrtche, Online extreme learning on fixed-point sensor networks, in: 2013 IEEE 13th International Conference on Data Mining Workshops, 2013, pp. 319–326. doi:10.1109/ICDMW.2013.74.

[20] H. H. W. J. Bosman, G. Iacca, H. J. Wrtche, A. Liotta, Online fusion of incremental learning for wireless sensor networks, in: 2014 IEEE International Conference on Data Mining Workshop, 2014, pp. 525–532. doi:10.1109/ICDMW.2014.79.

[21] D. C. Mocanu, M. T. Vega, E. Eaton, P. Stone, A. Liotta, Online contrastive divergence with generative replay: Experience replay without storing data, CoRR abs/1610.05555.

[22] H. Shin, J. K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 2994–3003.

[23] N. Kamra, U. Gupta, Y. Liu, Deep generative dual memory network for continual learning, CoRR abs/1710.10368. arXiv:1710.10368.

[24] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory, in: D. E. Rumelhart, J. L. McClelland, et al. (Eds.), Parallel Distributed Processing: Volume 1: Foundations, MIT Press, Cambridge, 1987, pp. 194–281.

[25] H. Ackley, E. Hinton, J. Sejnowski, A learning algorithm for boltzmann machines, Cognitive Science (1985) 147–169.

[26] G. W. Taylor, G. E. Hinton, S. T. Roweis, Two distributed-state models for generating high-dimensional time series, Journal of Machine Learning Research 12 (2011) 1025–1068.

[27] J. L. McClelland, B. L. McNaughton, R. C. O'Reilly, Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory, Psychological Review 102 (1995) 419–457.

[28] Y. Bengio, Learning deep architectures for ai, Found. Trends Mach. Learn. 2 (1) (2009) 1–127. doi:10.1561/2200000006.

[29] D. C. Mocanu, E. Mocanu, P. H. Nguyen, M. Gibescu, A. Liotta, A topological insight into restricted boltzmann machines, Machine Learning 104 (2) (2016) 243–270. doi:10.1007/s10994-016-5570-z.

[30] D. Mocanu, G. Exarchakos, H. Ammar, A. Liotta, Reduced reference image quality assessment via boltzmann machines, in: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, 2015, pp. 1278–1281. doi:10.1109/INM.2015.7140481.

[31] M. T. Vega, D. C. Mocanu, J. Famaey, S. Stavrou, A. Liotta, Deep learning for quality assessment in live video streaming, IEEE Signal Processing Letters 24 (6) (2017) 736–740. doi:10.1109/LSP.2017.2691160.

[32] D. Mocanu, G. Exarchakos, A. Liotta, Deep learning for objective quality assessment of 3d images, in: Image Processing (ICIP), 2014 IEEE International Conference on, 2014, pp. 758–762. doi:10.1109/ICIP.2014.7025152.

[33] H. B. Ammar, E. Eaton, M. E. Taylor, D. C. Mocanu, K. Driessens, G. Weiss, K. Tuyls, An automated measure of mdp similarity for transfer in reinforcement learning, in: Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

[34] J. Polastre, R. Szewczyk, D. Culler, Telos: enabling ultra-low power wireless research, in: IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005., 2005, pp. 364–369. doi:10.1109/IPSN.2005.1440950.

[35] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al., Tinyos: An operating system for sensor networks, Ambient intelligence 35 (2005) 115–148.

[36] G. Fortino, A. Guerrieri, G. O'Hare, A. Ruzzelli, A flexible building management framework based on wireless sensor and actuator networks, Journal of Network and Computer Applications 35 (6) (2012) 1934 – 1952. doi:https://doi.org/10.1016/j.jnca.2012.07.016.

[37] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection tree protocol, in: Proceedings of the 7th ACM conference on embedded networked sensor systems, ACM, 2009, pp. 1–14.

[38] O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, P. Levis, Ctp: An efficient, robust, and reliable collection tree protocol for wireless sensor networks, ACM Transactions on Sensor Networks (TOSN) 10 (1) (2013) 16.