IECON2015-Yokohama
November 9-12, 2015

A Machine Learning Technique in a Multi-Agent Framework for Online Outliers Detection in Wireless Sensor Networks

Hugo Martins[1], Fábio Januário[1,3], Luís Palma[1,2], Alberto Cardoso[3], Paulo Gil[1,2,3]
[1]Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal
[2]CTS-UNINOVA, Campus of Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal
[3]CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
Email: hmo.martins@campus.fct.unl.pt, f.januario@campus.fct.unl.pt, lbp@fct.unl.pt
alberto@dei.uc.pt, pgil@dei.uc.pt

***Abstract*—Wireless Sensor Networks enable flexibility, low operational and maintenance costs, as well as scalability in a variety of scenarios. However, in the context of industrial monitoring scenarios the use of Wireless Sensor Networks can compromise the system's performance due to several factors, being one of them the presence of outliers in raw data. In order to improve the overall system's resilience, this paper proposes a distributed hierarchical multi-agent architecture where each agent is responsible for a specific task. This paper deals with online detection and accommodation of outliers in non-stationary time-series by appealing to a machine learning technique. The methodology is based on a Least Squares Support Vector Machine along with a sliding window-based learning algorithm. A modification to this method is considered to improve its performance in transient raw data collected from transmitters over a Wireless Sensor Networks (WSNs). An empirical study based on laboratory test-bed show the feasibility and relevance of incorporating the proposed methodology in the context of monitoring systems over Wireless Sensor Networks.**

I. Introduction

WSNs are distributed networks of sensors nodes, which act together in order to monitor a diversity of physical environments or systems [1]. Each node is a small electronic device with wireless communication capabilities, including data storage and processing power. In addition, they can still be programmed to interact with the physical environment by means of built in sensors [2].

In industrial environments WSNs can be used in monitoring contexts, which contribute to reducing installation and operation costs. However, constraints on resources, in particular limited processing power, small data storage, narrow bandwidth and autonomy can lead to inaccuracies on data sent to the sink [3]. On the other hand, they are to some extent inherently vulnerable to cyber-intrusion, in which malignant actors can mask the system's degradation or provide "fake" data to higher management levels, concerning the current system's status [4]. Moreover, in the case of deployment of nodes in harsh environments, they may exhibit malfunction behaviours, which ultimately result in corrupted raw data that are subsequently sent to the sink. These errors in the readings are, as a whole, denoted as outliers, and should be cancelled out or accommodated. This issue can be addressed though the implementation of dedicated "coadjutants" that holistically lead to enhancing the overall systems' robustness and resilience. The implementation of such measures can be carried out through the implementation of algorithms and heuristics within a framework based on agents [4], and making use of agents' inherent features, such as, autonomy, reactivity, pro-activity, cooperation and intelligence.

Most of the techniques for outlier detection are computationally demanding, requiring large amounts of memory for data storage, high levels of energy consumption and communication overheads, and do not support distributed computation capabilities [2]. Besides, common methods are not tailor-made for heavy online data streaming, and are essentially based on data patterns, in order to minimize the redundancy of collected data. These considerations point out to the need for developing online outliers detection and accommodation schemes locally at nodes.

In order to cope with the computational constraints of sensor nodes this work proposes an approach based on a hierarchical multi-agents paradigm, where each agent is tailored for executing specific and coordinated tasks, namely to outliers detection and accommodation. The outliers detection and accommodation framework used in this paper relies on a Kernel-based technique, namely Least Square (LS)-Support Vector Machine (SVM), together with an online sliding window scheme [5]. Moreover, to improve its performance in terms of sensitivity and specificity in transient raw time-series, this paper considers a modification to the standard method, which is empirically proven out-perform the standard approach [6].

II. Multi-Agent Architecture for WSNs

The hierarchical multi-agent based architecture is composed of two layers, namely, a higher-level layer with coordination functionalities and a bottom layer comprising subordinate agents that are committed to specific tasks, namely, to monitoring analogue-to-digital converters (ADCs) readings, detection of outliers and their subsequent accommodation (see Fig. 1).

Each message comprises a header and a payload. The message payload (see Table I) consists of *Message Type*: the message can be originated from the system's application or from a local agent; *Node ID*: denoting the node address; *Control ID*: the command flag for local agents; *Data ID*: data collected in the node ID; *Agent ID:* agent's identifier that will

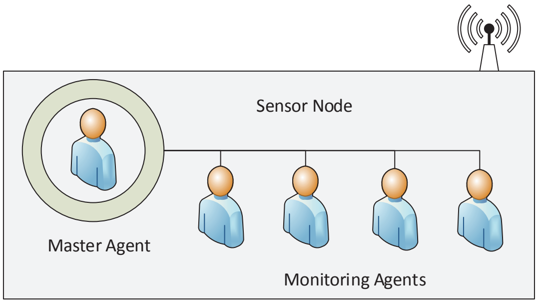

Fig. 1. Illustration of a multi-agent system.

be launched, stopped or resumed; *Agent MSG:* data provided by an agent.

TABLE I. MESSAGE PAYLOAD.

Message Type	Node ID	Control ID	Data ID	Agent ID	Agent MSG

Concerning the agents' commands, the platform is provided with: *i*) *Start Agent*: starts a monitoring agent at a sensor node by sending the command flag, the agent's ID and the node's destination address; *ii*) *Stop Agent*: stops a particular monitoring agent; *iii*) *Start All Agents*: starts all monitoring agents stacked at the sensor node memory; *iv*) *Stop All Agents*: stops all monitoring agents that are running at the sensor node.

A. Master-Agent

The main goal for this agent is to carry out extensive management routines related to other subordinate local agents, and to coordinate the communications between the node and the sink. When this agent is activated, it automatically launches dependent lower-level agents. This agent is also responsible for monitoring the status of all local dependent agents and, in case of agent's crash, the master will relaunch the corresponding "thread".

B. Monitoring Agent

This agent is responsible for collecting data from the environment and for accommodating possible outliers in raw data. The detection and accommodation of outliers is based on the approach proposed in Section III. If the sample taken at a given discrete time is tagged as an outlier, the corresponding value is replaced by accommodated value and an alarm is triggered and sent to the sink node.

III. OUTLIERS DETECTION AND ACCOMMODATION

In general, outliers detection techniques for WSNs should identify errors, due to nodes limited resources and harsh deployment environment conditions, from events associated with a given change in the state of the environment, with respect to a nominal regime or normal systems' behaviour [7]. Additionally, these methodologies, which are implemented at sensors' level, must cope with their operative constraints, notably, computing power and memory resources.

The Support Vector Novelty Detection (SVND) approach deals with the problem of given a set of vectors $X = \{x_1, \ldots, x_m\} \in \mathscr{X}^m$, such that the sequence $x_i, i = 1, \ldots, m \sim p_0$ (p_0 unknown) and two hypotheses H_0 and H_1, of categorising a new reading $x \in \mathscr{X}$ with identical probability density function p_0 under these two hypotheses. This problem is here addresses by defining a given decision function $f(x) \in \mathscr{S} \subset \mathscr{X}$ and a real number b, such that $f(x) - b \geq 0$ if $x \in \mathscr{S}$ (x is "normal"), and $f(x) - b < 0$ if x is an outlier. This work considers a Reproducing Kernel Hilbert Space (RKHS) (see e.g. [8], [9]) with kernel function $k(\cdot,\cdot)$. The RKHS can be selected by first considering a positive definite kernel function $k(\cdot,\cdot) : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$. A common choice for the kernel function is the Gaussian Radial Basis Function (RBF), given as:

$$k(x_1, x_2) = \exp\left[-\frac{1}{2\sigma^2}\|x_1 - x_2\|^2\right] \tag{1}$$

where $\|\cdot\|$ represents the canonical norm, or just norm.

For a positive definite kernel and the corresponding RKHS $\mathscr{F}$, the SVND methodology provides the function $f(x)$ as the solution to the following convex optimization problem, with $0 < \upsilon < 1$ [10]:

$$\begin{aligned} &\max_{f(\cdot)\in\mathscr{F}, e_i, b} -\frac{1}{2}\|f(.)\|^2 - \frac{1}{\upsilon m}\sum_{i=1}^{m} e_i^2 + b \\ &\text{subject to } f(x_i) - b = -e_i,\ e_i \geq 0 \end{aligned} \tag{2}$$

The slack variables e_i along with the constraints guarantees that the underlying decision function $f_x(\cdot)$ fits the training data, which implies that almost all the training data are located inside the region $\mathscr{S}$. Those readings x_i lying outside this region are assumed to be outliers.

The dual minimization problem for (2) is obtained by appealing to a set of Lagrange multipliers $\alpha = \{\alpha_1, ..., \alpha_m\}$, with the underlying Lagrangian given as:

$$L = \frac{1}{2}\|f(.)\|^2 + \frac{1}{\upsilon m}\sum_{i=1}^{m} e_i^2 - b - \sum_{i=1}^{m} \alpha_i [f(x_i) - b + e_i] \tag{3}$$

By computing the Lagrangian's partial derivatives with respect to $f(x)$, b, e_i and α_i and set them equal to zero, it follows:

$$\begin{cases} \sum_{j=1}^{m} \alpha_j k(x_j, x_i) - b + \frac{\upsilon m}{2}\alpha_i = 0 \\ \sum_{j=1}^{m} \alpha_j = 1 \end{cases} \tag{4}$$

In the compact form, Eq. (4) can be represented by the following matrix equation:

$$\begin{bmatrix} 0 & I \\ -I^T & H \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5}$$

where I and α are vectors with length m and H is a square matrix of size $m \times m$, as follows:

$$I = [1 \cdots 1] \tag{6}$$

$$\alpha = [\alpha_1 \cdots \alpha_m]^T \tag{7}$$

$$H = \begin{bmatrix} k(x_1, x_1) + \frac{\upsilon m}{2} & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) + \frac{\upsilon m}{2} \end{bmatrix} \tag{8}$$

The optimal decision function $f_x(x)$ is given as the solution to Eq. (5), namely

$$f_x(x) = \sum_{i=1}^{m} \alpha_i k(x, x_i) - b \tag{9}$$

with $f_x(x) \geq 0$ when x is a "normal" reading and $f_x(x) < 0$ when x is an outlier.
Since readings collected from a given system are in most cases not clean, the above discriminant tends to be inefficient, with respect to the sensitivity and specificity of the underlying decision (H_0 or H_1). To overcome this issue, an outlier index I_t was proposed in [10]. At a given time t, the detection algorithm is trained using the m most recent observations, yielding the vector α_t and b_t, with the outlier index I_t computed according to:

$$I_t = -\log\left[\sum_{i=1}^{m} \alpha_{i,t} k\left(x_{t-(m+1)+i}, x_t\right)\right] + \log[b_t] \tag{10}$$

where b_t can be regarded as a scaling factor for α_t, while the subscript $(t-(m+1)+i)$ corresponds to the online sliding window used in the training algorithm. By making use of I_t, a measurement is consider an outlier if $I_t > 0$. In practice, however, in order to make I_t less sensitive to noise in raw data it is instead compared to a threshold $\eta > 0$, with $\eta \approx -\log(\eta')$, $\eta' < 1$, $\eta' \approx 1$, typically chosen as 0.99.

$$I_t \geq \eta \Leftrightarrow \frac{\|x_t - \mu\|}{\varsigma^2} \geq \psi\left(\frac{\sigma}{\varsigma}, \eta, \upsilon\right) \tag{11}$$

with $\psi(\cdot)$ a given threshold.

A. *Online implementation*

The training set is updated at each sampling time with a new sample collected from the system, while the oldest sample in the vector X is discarded. At time t the training data set consists of m samples, namely:

$$X = \begin{bmatrix} x_{t-m} & x_{t-m+1} & \cdots & x_{t-1} \end{bmatrix}^{\mathrm{T}} \tag{12}$$

By solving Eq. (5), it follows:

$$b_t = \frac{1}{I \cdot H_t^{-1} \cdot I^{\mathrm{T}}} \tag{13}$$

$$a_t = H_t^{-1} \cdot I^{\mathrm{T}} \cdot b_t \tag{14}$$

In order to compute b_t and a_t it is required first to find the inverse of matrix H_t.

$$H_t = \begin{bmatrix} f_t & F_t^{\mathrm{T}} \\ F_t & W_t \end{bmatrix} \tag{15}$$

with:

$$f_t = k(x_{t-m}, x_{t-m}) + \frac{\upsilon m}{2} \tag{16}$$

$$F_t = \left[k(x_{t-m+1}, x_{t-m}) \cdots k(x_{t-1}, x_{t-m})\right]^{\mathrm{T}} \tag{17}$$

$$W_t = \begin{bmatrix} k(x_{t-m+1}, x_{t-m+1}) + \frac{\upsilon m}{2} & \cdots & k(x_{t-m+1}, x_{t-1}) \\ \vdots & \ddots & \vdots \\ k(x_{t-1}, x_{t-m+1}) & \cdots & k(x_{t-1}, x_{t-1}) + \frac{\upsilon m}{2} \end{bmatrix} \tag{18}$$

At time $t+1$, H_{t+1} is given by:

$$H_{t+1} = \begin{bmatrix} W_t & V_{t+1} \\ V_{t+1}^{\mathrm{T}} & v_{t+1} \end{bmatrix} \tag{19}$$

with,

$$v_{t+1} = k(x_t, x_t) + \frac{\upsilon m}{2} \tag{20}$$

$$V_{t+1} = \left[k(x_{t-m+1}, x_t) \cdots k(x_{t-1}, x_t)\right]^{\mathrm{T}} \tag{21}$$

In order to cope with the complexity of inverting block matrices, in this work, following [5], the block matrices H_t^{-1} and H_{t+1}^{-1} are computed based on the Sherman-Woodbury theorem (see e.g. [11]).

Theorem 1 (Sherman-Woodbury Theorem): Let Z be a symmetrical matrix with n rows and n columns, taking the following form:

$$Z = \begin{bmatrix} A & u \\ u^{\mathrm{T}} & a \end{bmatrix} \; or \; Z = \begin{bmatrix} a & u^T \\ u & A \end{bmatrix} \tag{22}$$

where A is a square matrix and a is a scalar quantity. Then the inverse matrix of Z can be computed as:

$$Z^{-1} = \begin{bmatrix} B & q \\ q^{\mathrm{T}} & \tau \end{bmatrix} \tag{23}$$

with:

$$B = A^{-1} + \tau A^{-1} u u^{\mathrm{T}} A^{-1} \tag{24}$$

$$q = -\tau A^{-1} u \tag{25}$$

$$\tau = \frac{1}{a - u^{\mathrm{T}} A^{-1} u} \tag{26}$$

Taking into account (23), matrices H_t^{-1} and H_{t+1}^{-1} can be calculated as follows:

$$H_t^{-1} = \begin{bmatrix} \tau & h_t \\ h_t^{\mathrm{T}} & G_t \end{bmatrix} \tag{27}$$

with,

$$\tau = \frac{1}{f_t - F_t^{\mathrm{T}} W_t^{-1} F_t} \tag{28}$$

$$h_t = -\tau F_t^{\mathrm{T}} W_t^{-1} \tag{29}$$

$$G_t = W_t^{-1} + \tau W_t^{-1} F_t F_t^{\mathrm{T}} W_t^{-1} \tag{30}$$

and,

$$H_{t+1}^{-1} = \begin{bmatrix} G_{t+1} & h_{t+1} \\ h_{t+1}^{\mathrm{T}} & \tau \end{bmatrix} \tag{31}$$

where,

$$\tau = \frac{1}{v_{t+1} - V_{t+1}^{\mathrm{T}} W_t^{-1} V_{t+1}} \tag{32}$$

$$h_{t+1} = -\tau W_t^{-1} V_{t+1} \tag{33}$$

$$G_{t+1} = W_t^{-1} + \tau W_t^{-1} V_{t+1} V_{t+1}^{\mathrm{T}} W_t^{-1} \tag{34}$$

By comparing (27) and (31), one observes that W_t^{-1} is common to both equations. From (27) follows:

$$G_t = W_t^{-1} + \frac{1}{\tau} h_t^{\mathrm{T}} h_t \Leftrightarrow W_t^{-1} = G_t - \frac{1}{\tau} h_t^{\mathrm{T}} h_t \tag{35}$$

Now, taking into account (35), the block matrix W_t^{-1} can be calculated from H_t^{-1}, and by replacing in (31), the block matrix H_{t+1}^{-1} computation can be recursively formulated.

B. Modified Kernel

A drawback of using the standard RBF kernel (1), is when readings are non-stationary, the outliers detection performance is seriously impacted. This is due to the argument used in computation of the norm, namely $\left\|x_j - x_{j+1}\right\|$, which is influenced by the transient response of the system, tending to increase the false positive rate. This issue is here addressed by replacing the argument of the norm by the difference to a trend line that is computed taking into account the most recent m samples. The new kernel function is defined as:

$$k(\tilde{x}_1, \tilde{x}_2) = \exp\left[-\frac{1}{2\sigma^2}\|\tilde{x}_1 - \tilde{x}_2\|^2\right] \tag{36}$$

with $\tilde{x}_t = \|x_t - \hat{x}_t\|$ the error between the current sample x_t and the corresponding estimate $\hat{x}_t$, obtained by Least Squares regression.
This change in the kernel function affects the computation of H, namely (8) and (15)-(21). In this new formulation they are found according to:

$$H = \begin{bmatrix} k(\tilde{x}_1, \tilde{x}_1) + \frac{\upsilon m}{2} & \cdots & k(\tilde{x}_1, \tilde{x}_m) \\ \vdots & \ddots & \vdots \\ k(\tilde{x}_m, \tilde{x}_1) & \cdots & k(\tilde{x}_m, \tilde{x}_m) + \frac{\upsilon m}{2} \end{bmatrix} \tag{37}$$

At time t, H_t is given by:

$$H_t = \begin{bmatrix} f_t & F_t^{\mathrm{T}} \\ F_t & W_t \end{bmatrix} \tag{38}$$

with,

$$f_t = k(\tilde{x}_{t-m}, \tilde{x}_{t-m}) + \frac{\upsilon m}{2} \tag{39}$$

$$F_t = \left[k(\tilde{x}_{t-m+1}, \tilde{x}_{t-m}) \cdots k(\tilde{x}_{t-1}, \tilde{x}_{t-m})\right]^{\mathrm{T}} \tag{40}$$

$$W_t = \begin{bmatrix} k(\tilde{x}_{t-m+1}, \tilde{x}_{t-m+1}) + \frac{\upsilon m}{2} & \cdots & k(\tilde{x}_{t-m+1}, \tilde{x}_{t-1}) \\ \vdots & \ddots & \vdots \\ k(\tilde{x}_{t-1}, \tilde{x}_{t-m+1}) & \cdots & k(\tilde{x}_{t-1}, \tilde{x}_{t-1}) + \frac{\upsilon m}{2} \end{bmatrix} \tag{41}$$

while at time $t+1$, H_{t+1} is computed as follows:

$$H_{t+1} = \begin{bmatrix} W_t & V_{t+1} \\ V_{t+1}^{\mathrm{T}} & v_{t+1} \end{bmatrix} \tag{42}$$

with,

$$v_{t+1} = k(\tilde{x}_t, \tilde{x}_t) + \frac{\upsilon m}{2} \tag{43}$$

$$V_{t+1} = \left[k(\tilde{x}_{t-m+1}, \tilde{x}_t) \cdots k(\tilde{x}_{t-1}, \tilde{x}_t)\right]^{\mathrm{T}} \tag{44}$$

Taking into account the proposed Gaussian kernel the accommodation of a detected outlier is carried out by replacing the reading by the trend provided by the the Least Squares predictor, namely when $I_t > \eta$ (outlier detected) then $x_t = \hat{x}_t$. The overall approach is presented in Algorithm 1.

Algorithm 1 Proposed Outlier Detection and Accomodation

Require: υ, m
  Initialise $X \leftarrow [x_1 \quad \cdots \quad x_m]$
  Obtain $\hat{X}$ by fitting a curve to X
  $\tilde{X} \leftarrow \|X - \hat{X}\|$
  Compute H as in (37)
  Calculate H^{-1}
  repeat
    $x_t \leftarrow$ read_sample
    Obtain predictor $\hat{x}$ by fitting a curve to X
    $\tilde{x}_t \leftarrow \|x_t - \hat{x}_t\|$
    Compute b_t and a_t as in (13) and (14)
    Obtain I_t from (10)
    if $I_t > \eta$ **then**
      x_t is an outlier
      $x_t \leftarrow \hat{x}_t$ % sample accommodated
    end if
    Obtain v_{t+1} and V_{t+1} from (43) and (44)
    Compute W_t^{-1} as in (35)
    Calculate H_{t+1}^{-1} using (31)
    Update X by adding x_t and removing the oldest sample
    Update $\tilde{X}$ by adding $\tilde{x}_t$ and removing the oldest sample
  until End_Detection

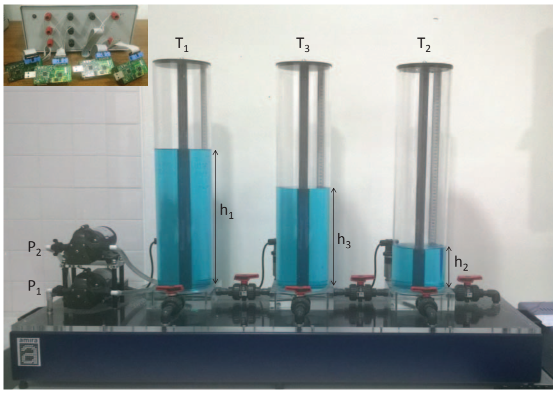

Fig. 2. Testbed schematics

IV. Case Study

A. Test-bed Description

The test-bed consists of a three-tank benchmark system (Fig 2), two remote desktop computers, one for running the applications (PC_2) and the other the middleware (Tunslip and dispatcher) (PC_1) and four wireless sensor nodes for implementing WSN, with the main goal of detecting and accommodating outliers in tanks readings over the WSN.

The AMIRA© DTS 200 benchmark three-tank system comprises three plexiglas cylindrical tanks with identical cross-section supplied with distilled water. The liquid levels, namely h_1, h_2 and h_3 are measured by piezoresistive transducers. The middle tank T_3 is connected to the other two tanks by means of circular cross-section pipes provided with manually adjustable ball valves. The main outlet of the system is located in the tank T_2, which is directly connected to the collecting reservoir by means of a circular cross-section pipe provided with an outflow ball valve. Additionally, this system includes two pumps for feeding tanks T_1 and T_2 with water.

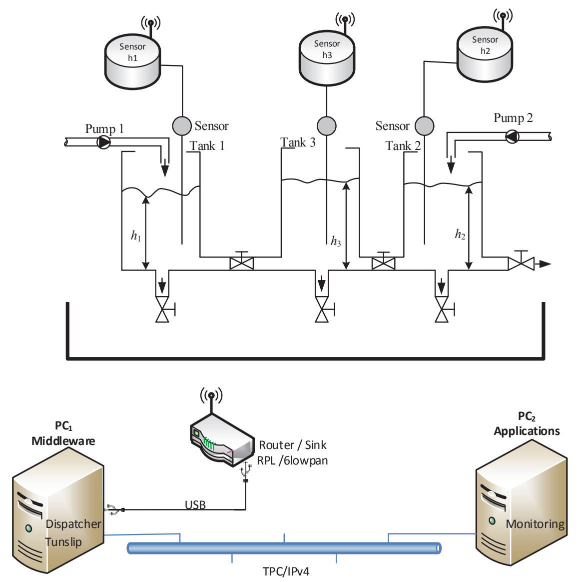

Fig. 3. Testbed schematics

Concerning the WSN infrastructure, it comprises Zolertia Z1 sensor nodes, which leverage several industry standards, such as, USB, IEEE 802.15.4 and Zigbee to interoperate seamlessly with other devices. The Z1 is a low power wireless device that comes with built in support for some of the currently most employed open source operating systems by the WSN community, namely TinyOS and Contiki. The supported network stacks include 6LoWPAN and Zigbee. Each node includes analogue and digital ports, to which sensors and actuators can be attached. The operating system used in WSN programming is based on the Contiki. This operating system has been written in C language with support for dynamic loading and replacement of individual programs and services. Additionally, it was built around an event-driven kernel, but provides optional preemptive multi-threading, which can be applied to individual processes [12].

Regarding the test-bed (Fig 3), three nodes are configured as sensors, in order to collect the tanks' levels, namely h_1, h_2 and h_3, while a fourth node is used as a router/sink. This node is attached via a USB port to the remote desktop computer (PC_1) where the Tunslip is running, thus allowing IPv6 communication with WSN nodes, creating a Serial Line Protocol (SLIP) tunnel between the physical serial port and the virtual network interface [13]. In addition a dispatcher is used to make the translation between IPv6 packets into the IPv4 format, in order to allow the communication with IPv4 based services and servers. Finally, the monitoring system are running in a different remote desktop (PC_2) were the readings from the WSN are recorded.

The proposed architecture is based on a multi-agent paradigm, where each agent is responsible for a specific task (see section II). In terms of modus operandi, each sensor node includes a master agent and a set of agents for monitoring purposes, including the detection and accommodation of outliers (see Algorithm 1). When an outlier event is detected the agent prompts an alarm that is transmitted by master agent to the sink node and subsequently to the monitoring system. At the same time, the outlying sample is locally accommodated and transmitted to the monitoring system. The message payload associated with the sensor readings that is transmitted to the sink (see Table II) includes: the sample number (k); reading (y_k); the accommodated sample (z_k), in case of an outlier alarm is raised; triggered alarm (0 for undetected outlier, 1 for an outlier); the outlier index (It_k).

TABLE II. MESSAGE PAYLOAD.

k	y_k	z_k	alarm	It_k

B. Experimental Results

This section is devoted to assess the support vector machine based technique within a multi-agent framework for local online detection and accommodation of outliers over a WSN. With this regard, readings are taken from sensor nodes' ADCs, at the frequency of 1 Hz, while the input to the system's pumps is implemented via a USB data acquisition board, namely a NI6008, from National Instruments. Fig. 4, Fig. 5 and Fig. 6 show the obtained results in the three-tank system.

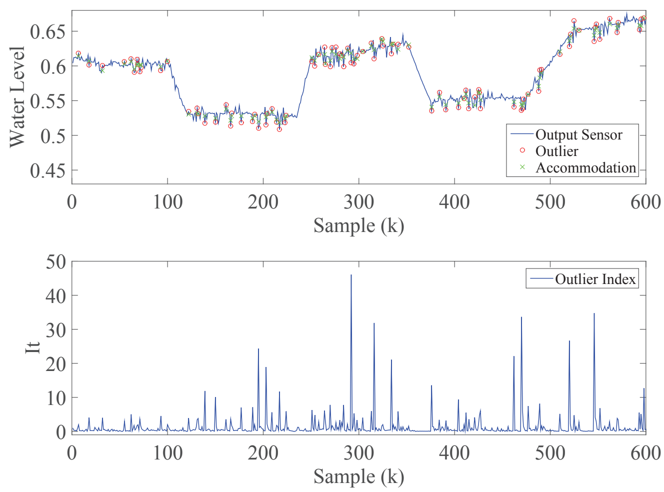

Fig. 4. MIMO System: Tank 1

In the upper figures the raw data is plotted in blue, and outliers represented by a red circle, while the accommodated samples are shown as green crosses. On the other hand, the bottom figures represent the outlier index I_t, which in both cases is approximately close to zero, except when an outlier is detected. As can be observed, the implementation of the real-time outliers detection and accommodation framework proves to be feasible and effective, contributing to improve the quality of data sent to the sink. To corroborate the above assertion some statistical measures regarding both tanks are presented in Table III. Specifically, in the case of tank 1, over 10 minutes of running experiment, represented by 600 samples, the algorithm detected 89 outliers, which accounts for approximately 14.8% of all the sensor readings.

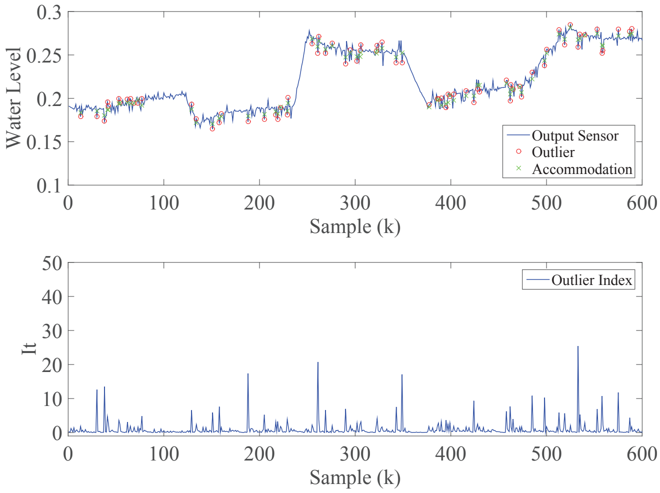

Fig. 5. MIMO System: Tank 2

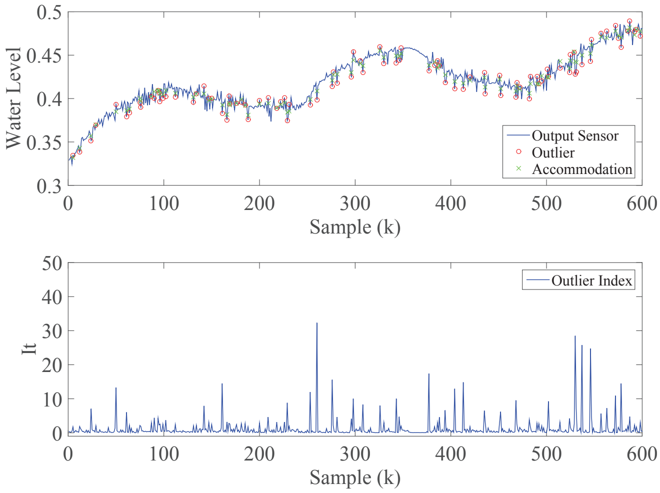

Fig. 6. MIMO System: Tank 3

V. Conclusion

This paper addressed the problem of outliers detection and accommodation in Wireless Sensor Networks. The proposed approach relies on a hierarchical multi-agent framework, with dedicated agents running on sensor nodes. Outliers are detected using a Least Squares-Support Vector Machine algorithm, under the form of a Reproducing Kernel Hilbert Space with Radial Basis Function kernel, along with a sliding window-based learning technique. In order to assess the performance of this methodology some experiments were conducted on a testbed comprising a benchmark three-tank system. Results collected from this testbed show the effectiveness and pertinence of the proposed methodology.

Acknowledgment

Januário, F. acknowledge Fundação para a Ciência e Tecnologia (FCT), Portugal for the Ph.D. Grant SFRH/BD/85586/2012. This work has been partially supported by iCIS-Intelligent Computing in the Internet of Services, Project CENTRO-07-ST24-FEDER-002003.

TABLE III. Statistical measures.

	Tank 1	Tank 2	Tank 3
Number of samples	600	600	600
Number of Outliers	89	72	92
Detection Rate	14.8%	12.0%	15.3%
Maximum accom. ($\times 10^{-2}$)	1.19	0.99	0.95
Minimum accom. ($\times 10^{-2}$)	-1.38	-1.19	-1.46
Mean ($\times 10^{-4}$)	-1.86	0.17	-2.39
Std. Deviation ($\times 10^{-2}$)	0.27	0.20	0.25

References

[1] M. J. G. C. Mendes, B. M. S. Santos, and J. S. da Costa, “Multi-agent Platform and Toolbox for Fault Tolerant Networked Control Systems,” *Journal of Computers*, vol. 4, no. 4, Apr. 2009.

[2] P. Gil, A. Santos, and A. Cardoso, “Dealing With Outliers in Wireless Sensor Networks: An Oil Refinery Application,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1589–1596, Jul. 2014.

[3] F. Tirkawi, “Generality Challenges and Approaches in WSNs,” *Int’l J. of Communications, Network and System Sciences*, vol. 02, no. 01, pp. 58–63, 2009.

[4] C. G. Rieger and K. Villez, “Resilient control system execution agent (ReCoSEA),” in *2012 5th International Symposium on Resilient Control Systems*. IEEE, Aug. 2012, pp. 143–148. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6309308

[5] L. Fang and M. Zhi-zhong, “An online outlier detection method for process control time series,” in *2011 Chinese Control and Decision Conference (CCDC)*. IEEE, May 2011, pp. 3263–3267.

[6] H. Martins, L. Palma, A. Cardoso, and P. Gil, “A Support Vector Machine Based Technique for Online Detection of Outliers in Transient Time Series,” in *10th Asian Control Conference*, 2015.

[7] Y. Zhang, N. Hamm, N. Meratnia, A. Stein, M. van de Voort, and P. Havinga, “Statistics-based outlier detection for wireless sensor networks,” pp. 1373–1392, Aug. 2012.

[8] A. Berlinet and C. Thomas-Agnan, “Reproducing kernel Hilbert spaces in probability and statistics,” 2004.

[9] I. Steinwart, D. Hush, and C. Scovel, “An explicit description of the reproducing Kernel Hilbert spaces of Gaussian RBF kernels,” *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4635–4643, Oct. 2006.

[10] M. Davy, F. Desobry, A. Gretton, and C. Doncarli, “An online support vector machine for abnormal events detection,” *Signal Processing*, vol. 86, no. 8, pp. 2009–2025, Aug. 2006.

[11] M. S. Bartlett, “An Inverse Matrix Adjustment Arising in Discriminant Analysis,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 107–111, Mar. 1951.

[12] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” in *29th Annual IEEE International Conference on Local Computer Networks*. IEEE (Comput. Soc.), 2004, pp. 455–462.

[13] F. Januário, A. Santos, L. Palma, A. Cardoso, and P. Gil, “A distributed multi-agent approach for resilient supervision over a IPv6 WSAN infrastructure,” in *Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT)*, 2015.