A Support Vector Machine Based Technique for Online Detection of Outliers in Transient Time Series

Hugo Martins[1], 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, lbp@fct.unl.pt, alberto@dei.uc.pt, pgil@dei.uc.pt

Abstract— **This paper deals with online detection and accommodation of outliers in transient time series by appealing to a machine learning technique. The methodology is based on a Least Squares Support Vector Machine technique together with a sliding window-based learning algorithm. A modification to this method is proposed so as to extend its application to transient raw data collected from transmitters attached to a Wireless Sensor Network. The performance of two approaches are compared on a particular controlled data set.**

I. Introduction

A Wireless Sensor Network (WSN) is a network comprising tiny, low cost and low energy sensor nodes that are connected to one or more sink nodes, namely a gateway. This kind of infrastructure is becoming increasingly used in many fields, such as in environmental contexts, habitat monitoring, health monitoring or military surveillance, just to name a few [1]. Because of their inherent constraints, namely energy (battery autonomy), memory, computational power and communication bandwidth, raw data collected from a WSN are generally unreliable and inaccurate.

In order to account for possible artefacts in readings from WSNs, raw data need to be pre-processed, which includes commonly two steps, namely detection and accommodation. These anomalies are generically referred to as outliers, and in the context of WSNs they can be regarded as measurements that significantly deviate from the normal pattern of sensed data [2].

A number of detection methods have been proposed in the last few decades. They can be classified according to the underlying technique they use, the network structure or even the type of outliers they can detect. Several different taxonomies are found literature. For example in [2], the authors categorise those techniques as statistical-based, nearest neighbour-based, clustering-based, classification-based and spectral decomposition-based, with some of these being further categorized, while [3] classifies the methods as statistical, data mining/computational intelligence, rule based, hybrid, game theory and graph based approaches. The authors in [4], not only classify the methods according to the underlying techniques, but also categorise them in terms of data dimension, model structure, operation mode, adaptability to changes and types of correlations exploited. Regarding data dimension, outlier detection techniques can deal with univariate or multivariate data, while the operation mode refers to the possibility of using these techniques for online or offline detection. Regarding the model structure, outliers detection can be implemented using a local, a centralised approach or even a distributed methodology. In the first technique each node is provided with a local online detection agent, without taking into account the whole network, while in a centralised scheme all readings collected from the network are sent to a sole station (e.g. sink node or gateway), where the detection routine globally takes place. A distributed approach considers a spatially deployment of nodes, and makes use of correlations between neighbouring nodes to increase the detection performance. The adaptability to changes refers to whether the implemented method is adaptive, or not, to changes in the system behaviour, while the types of correlations exploited can be spatial, temporal, both or none.

It should be stressed, at this point, that outliers detection techniques designed for running on WSNs nodes need to have a high detection rate along with a low false rate, while maintaining low resources consumption, so as to cope with nodes constraints.

In this paper the problem of outliers detection and accommodation is tackled based on a Kernel-based technique, namely Least Square (LS)-Support Vector Machine (SVM), together with an online sliding window scheme [5]. The rationale for choosing this kind of techniques are to some extent related to the fact that they do not demand the definition of a given probability density function (p_0) for a given hypothesis, they provide computationally efficient decision functions, and they can be applied in high dimensional data sets [6]. Moreover, in order to extend its application to transient time series, this work proposes a modification to the standard method, which is empirically proven to improve the underlying sensitivity and specificity.

The remainder of this paper is organized as follows. Section II presents a brief introduction to the outliers detection technique considered in this study, describes the training algorithm used in online detection, and presents

978-1-4799-7862-5/15/$31.00 © 2015 IEEE

the proposed modification in order to improve the detection performance in non-stationary data sets. Section III presents some results, while concluding remarks are drawn in Section IV.

II. Outliers Detection and Accommodation

This section provides a brief introduction to the machine learning method used in the present work, and describes the proposed modification to improve its performance in transient data sets. The reader is referred to [6] and references therein for a comprehensive description of the standard method.

A. LS-SVND Algorithm

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. The decision function is designed under the following two constraints:

- Most of the training vectors are assumed as normal ($X \in \mathscr{S}$), except for a small subset of outliers;
- The bound that surrounds the uncorrupted data should be as small as possible, that is $\mathscr{S} \subset \mathscr{X}$ should have a minimum volume.

Based on these constraints, the space of possible functions $f(x)$ is reduced to a Reproducing Kernel Hilbert Space (RKHS) (see e.g. [7], [8]) with kernel function $k(\cdot, \cdot)$. This 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 $\|.\|$ represents the canonical norm, or just norm.
It should be mentioned that a positive definite kernel $k(\cdot, \cdot)$ induces a RKHS, that is a linear space of functions $\mathscr{F}$ represented by a dot product and denoted as $\langle \cdot, \cdot \rangle_{\mathscr{F}}$, with the corresponding norm denoted as $\|\cdot\|_{\mathscr{F}}$. In addition, $\mathscr{F}$ is complete in this norm, and for any $f(\cdot) \in \mathscr{F}$ the reproducing property is held, namely $\langle k(x, \cdot), f(\cdot) \rangle_{\mathscr{F}} = f(\cdot)$.
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 $0 < \upsilon < 1$ (see [6]):

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

In (2) the slack variables e_i along with the constraints guarantee 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 tagged as outliers. The number of outliers is kept low by minimizing the term $\sum_{i=1}^{m} e_i^2$, while the term $\|f(.)\|^2$ ensures that the second constraint holds, which results in a minimum volume for $\mathscr{S}$.
The dual minimisation 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:

$$\frac{\partial L}{\partial f(.)} = 0 \Rightarrow f(.) = \sum_{i=1}^{m} \alpha_i k(x_i, .) \tag{4}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i = 1 \tag{5}$$

$$\frac{\partial L}{\partial e_i} = 0 \Rightarrow e_i = \frac{\upsilon m}{2}\alpha_i \tag{6}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \Rightarrow f(x_i) - b + e_i = 0 \tag{7}$$

The above four equations can be rewritten as:

$$\begin{cases} \sum_{j=1}^{m} \alpha_j k(x_j, x_i) - b + \frac{\upsilon m}{2}\alpha_i = 0 \\ \sum_{j=1}^{m} \alpha_j = 1 \end{cases} \tag{8}$$

In the compact form (8) can be described by the following matrix equation:

$$\begin{bmatrix} 0 & I \\ -I^T & H \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{9}$$

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

$$I = [1 \cdots 1] \tag{10}$$

$$\alpha = [\alpha_1 \cdots \alpha_m]^T \tag{11}$$

$$H = \begin{bmatrix} k(x_1, x_1) + \frac{\upsilon m}{2} & \cdots & k(x_1, x_m) \\ \vdots & \ddots & \vdots \\ k(x_m, x_1) & \cdots & k(x_m, x_m) + \frac{\upsilon m}{2} \end{bmatrix} \tag{12}$$

The optimal decision function $f_x(x)$ is given as the solution of (9), namely

$$f_x(x) = \sum_{i=1}^{m} \alpha_i k(x, x_i) - b \tag{13}$$

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, i.e. noisy raw data, the above discriminant is rather inefficient in what the sensitivity and specificity of the underlying decision (H_0 or H_1) is concerned. In order to get around this issue it was suggested in [6] using an outlier index I_t. At a given time t, the detection algorithm is trained using the m most recent observations, yielding the vector α_t and b_t. The outlier index I_t is computed according to:

$$I_t = -\log\left[\sum_{i=1}^{m} \alpha_{i,t} k\left(x_{t-(m+1)+i}, x_t\right)\right] + \log\left[b_t\right] \tag{14}$$

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 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.

Interestingly, when $x \in R : x \sim \mathcal{N}\left(\mu, \varsigma^2\right)$ and $k(\cdot,\cdot)$ is a Gaussian kernel (1), it can be shown for $m \to \infty$ that [9],

$$I_t \geq \eta \Leftrightarrow \frac{\|x_t - \mu\|}{\varsigma^2} \geq \psi\left(\frac{\sigma}{\varsigma}, \eta, \upsilon\right) \tag{15}$$

with $\psi(\cdot)$ a given threshold. In such conditions (see [6]), the proposed modified test is equivalent to comparing the distance to the distribution mean to the distribution spread.

B. Online Algorithm

For online detection of outliers 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,

$$X = \begin{bmatrix} x_{t-m} & x_{t-m+1} & \cdots & x_{t-1} \end{bmatrix}^{\mathrm{T}} \tag{16}$$

By solving Eq.(9), it follows that,

$$b_t = \frac{1}{I \cdot H_t^{-1} \cdot I^{\mathrm{T}}} \tag{17}$$

$$a_t = H_t^{-1} \cdot I^{\mathrm{T}} \cdot b_t \tag{18}$$

In order to compute b_t and a_t, the inverse of matrix H_t has to be found.

$$H_t = \begin{bmatrix} f_t & F_t^{\mathrm{T}} \\ F_t & W_t \end{bmatrix} \tag{19}$$

with:

$$f_t = k\left(x_{t-m}, x_{t-m}\right) + \frac{\upsilon m}{2} \tag{20}$$

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

$$W_t = \begin{bmatrix} k\left(x_{t-m+1}, x_{t-m+1}\right) + \frac{\upsilon m}{2} & \cdots & k\left(x_{t-m+1}, x_{t-1}\right) \\ \vdots & \ddots & \vdots \\ k\left(x_{t-1}, x_{t-m+1}\right) & \cdots & k\left(x_{t-1}, x_{t-1}\right) + \frac{\upsilon m}{2} \end{bmatrix} \tag{22}$$

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

$$H_{t+1} = \begin{bmatrix} W_t & V_{t+1} \\ V_{t+1}^{\mathrm{T}} & v_{t+1} \end{bmatrix} \tag{23}$$

with,

$$v_{t+1} = k\left(x_t, x_t\right) + \frac{\upsilon m}{2} \tag{24}$$

$$V_{t+1} = \left[k\left(x_{t-m+1}, x_t\right) \cdots k\left(x_{t-1}, x_t\right)\right]^{\mathrm{T}} \tag{25}$$

To cope with the complexity of inverting block matrices, in this work, following [5], H_t^{-1} and H_{t+1}^{-1} are computed by appealing to the Sherman-Woodbury theorem (see e.g. [10]).

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

$$Z = \begin{bmatrix} A & u \\ u^{\mathrm{T}} & a \end{bmatrix} \; or \; Z = \begin{bmatrix} a & u^T \\ u & A \end{bmatrix} \tag{26}$$

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

$$Z^{-1} = \begin{bmatrix} B & q \\ q^{\mathrm{T}} & \tau \end{bmatrix} \tag{27}$$

with:

$$B = A^{-1} + \tau A^{-1} u u^{\mathrm{T}} A^{-1} \tag{28}$$

$$q = -\tau A^{-1} u \tag{29}$$

$$\tau = \frac{1}{a - u^{\mathrm{T}} A^{-1} u} \tag{30}$$

Taking into account (27), H_t^{-1} and H_{t+1}^{-1} can be computed as follows:

$$H_t^{-1} = \begin{bmatrix} \tau & h_t \\ h_t^{\mathrm{T}} & G_t \end{bmatrix} \tag{31}$$

with,

$$\tau = \frac{1}{f_t - F_t^{\mathrm{T}} W_t^{-1} F_t} \tag{32}$$

$$h_t = -\tau F_t^{\mathrm{T}} W_t^{-1} \tag{33}$$

$$G_t = W_t^{-1} + \tau W_t^{-1} F_t F_t^{\mathrm{T}} W_t^{-1} \tag{34}$$

and,

$$H_{t+1}^{-1} = \begin{bmatrix} G_{t+1} & h_{t+1} \\ h_{t+1}^{\mathrm{T}} & \tau \end{bmatrix} \tag{35}$$

where,

$$\tau = \frac{1}{v_{t+1} - V_{t+1}^{\mathrm{T}} W_t^{-1} V_{t+1}} \tag{36}$$

$$h_{t+1} = -\tau W_t^{-1} V_{t+1} \tag{37}$$

$$G_{t+1} = W_t^{-1} + \tau W_t^{-1} V_{t+1} V_{t+1}^{\mathrm{T}} W_t^{-1} \tag{38}$$

By comparing (31) and (35), one observes that W_t^{-1} is common to both equations. From (31),

$$G_t = W_t^{-1} + \frac{1}{\tau} h_t^{\mathrm{T}} h_t \Leftrightarrow W_t^{-1} = G_t - \frac{1}{\tau} h_t^{\mathrm{T}} h_t \tag{39}$$

Now, by taking into account (39), the block matrix W_t^{-1} can be evaluated from H_t^{-1}, and by replacing in (35), the matrix H_{t+1}^{-1} can be recursively computed. This approach is presented in Algorithm 1.

Algorithm 1 Outlier Detection

```
Require: $\upsilon, m$
  Initialise $X \leftarrow \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix}$
  Compute $H$ as in (12)
  Calculate $H^{-1}$
  repeat
    $x_t \leftarrow$ read_sample
    Compute $b_t$ and $a_t$ as in (17) and (18)
    Obtain $I_t$ from (14)
    if $I_t > \eta$ then
      $x_t$ is an outlier
    end if
    Obtain $v_{t+1}$ and $V_{t+1}$ from (24) and (25)
    Compute $W_t^{-1}$ as in (39)
    Calculate $H_{t+1}^{-1}$ using (35)
    Update X by adding $x_t$ and removing the oldest sample
  until End_Detection
```

C. Proposed Approach

One drawback of the standard approach based on the RBF kernel (1) is associated with the fact that when the system from which the readings are taken is not in steady state, the outliers detection performance is seriously compromised. This is related to the way the norm is computed, namely $\left\|x_j - x_{j+1}\right\|$, which is influenced by the transient response of the system, and it turns out to increase the false positive rate. This means that the kernel should be modified in order to cope with a deterministic transient behaviour.

This drawback is addressed in this work 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 rationale behind this approach is propped up on the fact that, by taking the deviation to the approximation to the deterministic behaviour, it makes the underlying discriminant less sensitive to the system dynamics. The new kernel function is then defined as:

$$k(\tilde{x}_1, \tilde{x}_2) = \exp\left[-\frac{1}{2\sigma^2}\|\tilde{x}_1 - \tilde{x}_2\|^2\right] \tag{40}$$

with $\tilde{x}_t = \|x_t - \hat{x}_t\|$ the error between the actual sample x_t and the estimated value $\hat{x}_t$, obtained by Least Squares regression. This change in the kernel function has an affect on the equations used for computing the matrix H, namely (12) and (19)-(25). In this new formulation they are found according to:

$$H = \begin{bmatrix} k(\tilde{x}_1, \tilde{x}_1) + \frac{\upsilon m}{2} & \cdots & k(\tilde{x}_1, \tilde{x}_m) \\ \vdots & \ddots & \vdots \\ k(\tilde{x}_m, \tilde{x}_1) & \cdots & k(\tilde{x}_m, \tilde{x}_m) + \frac{\upsilon m}{2} \end{bmatrix} \tag{41}$$

At time t, H_t is given by:

$$H_t = \begin{bmatrix} f_t & F_t^{\mathrm{T}} \\ F_t & W_t \end{bmatrix} \tag{42}$$

with,

$$f_t = k(\tilde{x}_{t-m}, \tilde{x}_{t-m}) + \frac{\upsilon m}{2} \tag{43}$$

Algorithm 2 Proposed Outlier Detection and Accomodation

```
Require: $\upsilon, m$
  Initialise $X \leftarrow \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix}$
  Obtain $\hat{X}$ by fitting a curve to X
  $\tilde{X} \leftarrow \|X - \hat{X}\|$
  Compute $H$ as in (41)
  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 (17) and (18)
    Obtain $I_t$ from (14)
    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 (47) and (48)
    Compute $W_t^{-1}$ as in (39)
    Calculate $H_{t+1}^{-1}$ using (35)
    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
```

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

$$W_t = \begin{bmatrix} k(\tilde{x}_{t-m+1}, \tilde{x}_{t-m+1}) + \frac{\upsilon m}{2} & \cdots & k(\tilde{x}_{t-m+1}, \tilde{x}_{t-1}) \\ \vdots & \ddots & \vdots \\ k(\tilde{x}_{t-1}, \tilde{x}_{t-m+1}) & \cdots & k(\tilde{x}_{t-1}, \tilde{x}_{t-1}) + \frac{\upsilon m}{2} \end{bmatrix} \tag{45}$$

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

$$H_{t+1} = \begin{bmatrix} W_t & V_{t+1} \\ V_{t+1}^{\mathrm{T}} & v_{t+1} \end{bmatrix} \tag{46}$$

with,

$$v_{t+1} = k(\tilde{x}_t, \tilde{x}_t) + \frac{\upsilon m}{2} \tag{47}$$

$$V_{t+1} = \left[k(\tilde{x}_{t-m+1}, \tilde{x}_t) \cdots k(\tilde{x}_{t-1}, \tilde{x}_t)\right]^{\mathrm{T}} \tag{48}$$

Taking into account the proposed Gaussian kernel the accommodation of a detected outlier is carried out by replacing the sample by the trend provided by the predictor based on the Least Squares regression, that is when $I_t > \eta$ (outlier detected) then $x_t = \hat{x}_t$. The overall approach is presented in Algorithm 2.

III. Case Study

In this section a data set generated with a virtual system is used to assess the performance of the proposed approach against the methodology based on the standard Gaussian kernel. The comparison is carried out using the outliers detection sensitivity and specificity, along with a computational complexity metric.

The nonlinear model (49) was originally suggested in [11], and later used in [12]. The main justification for including such a model in this work is that one can arbitrarily inject as many outliers in the clean generated data as needed, for statistical consistency.

$$y_k = \frac{y_{k-1}y_{k-2}y_{k-3}u_{k-2}\left(y_{k-3}-1\right)+u_{k-1}}{1+y_{k-2}^2+y_{k-3}^2} \tag{49}$$

where,

$$u_k = \begin{cases} \sin\left(\frac{2\pi k}{250}\right) & k \leq 500 \\ 0.8\sin\left(\frac{2\pi k}{250}\right)+0.2\sin\left(\frac{2\pi k}{25}\right) & k > 500 \end{cases} \tag{50}$$

The underlying time series is shown in Fig. 1.

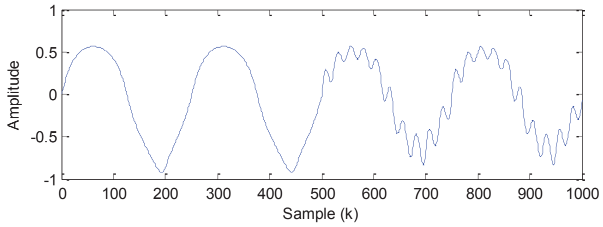
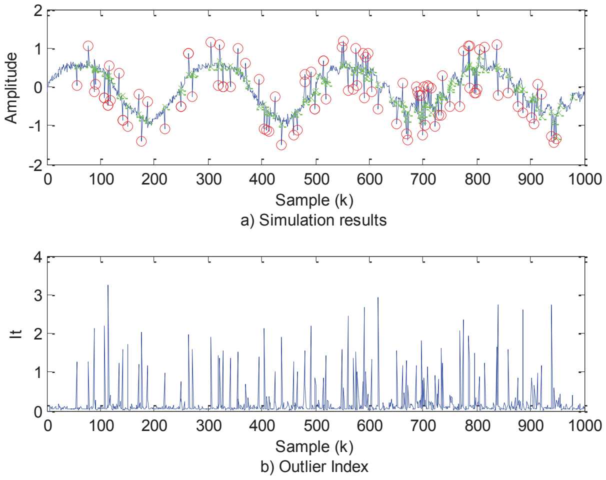


Fig. 1. Model of the first system

The output of Eq. (49), corresponding to clean data set, was subsequently corrupted with additive noise $\omega \sim \mathcal{N}(0, 0.06)$, to somehow emulate a real system environment. Moreover, the noisy data set was further manipulated with the inclusion of 100 outliers, randomly scattered throughout the data set, serving as targets for assessing the detection performance of competing methods. The outcomes for the original and the modified detection algorithm are presented in Fig. 2 and Fig. 3, respectively. In Fig. 2.a and Fig. 3.a the model



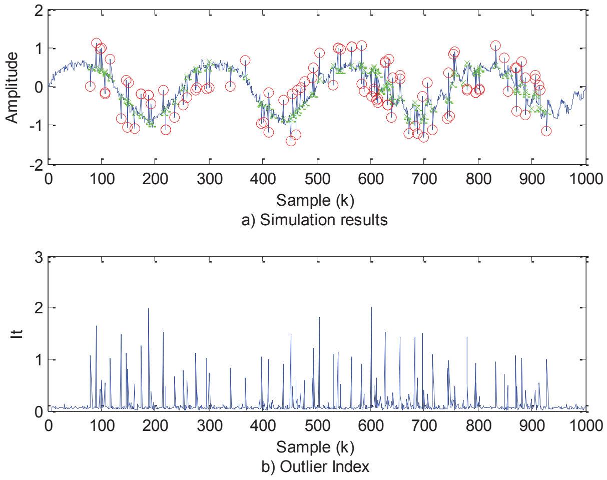
a) Simulation results

b) Outlier Index

Fig. 2. Simulation results for the original approach.

response is shown in blue, while the true identified outliers are represented by a red circle, and the accommodated samples shown as green crosses. The bottom figures, namely Fig. 2.b and Fig. 3.b, represent the outlier index I_t, which in both cases is approximately close to zero, except when an outlier is detected.

To allow comparing the performance of the two approaches



a) Simulation results

b) Outlier Index

Fig. 3. Simulation results for the modified Gaussian kernel.

in competition, the underlying Receiver Operator Characteristic (ROC) curve [13], which illustrates the performance of each binary classifier, is presented in Fig. 4. Taking into account Fig. 4, one can observe that both approaches provide a high true positive rate, with a fairly low false positive rate. Nevertheless, the ROC for the modified Gaussian kernel outperforms the standard scheme.

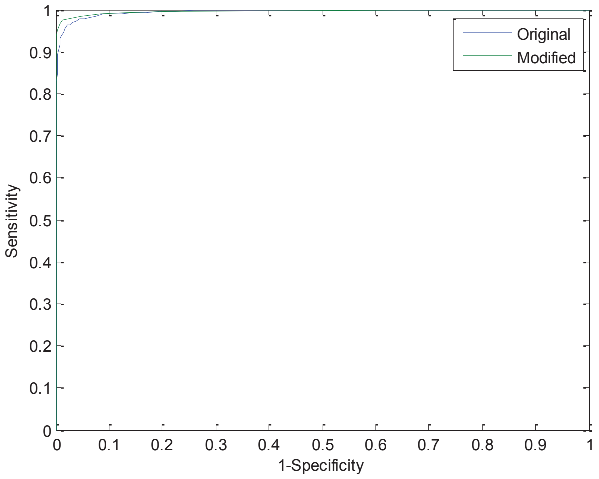


Fig. 4. Receiver Operator Characteristic

The results from Fig. 2 and Fig. 3 were taken using the best tradeoff between the true and false positive rates, which correspond to the elbow of the ROC curve. These values are summarized in Table I, together with the computational overhead.

TABLE I
RESULTS OF STUDIED APPROACHES

Detection Approach	True Positive Rate	False Positive Rate	Time Elapsed /Sample
Original	94.57%	2.88%	1.76 ms
Modified	95.70%	0.89%	3.58 ms

As can be inferred from Table I, the modified approach out-performs the original methodology, allowing the detection of a superior number of true positives, while maintaining the false positive rate significantly lower. In terms of computational complexity, expressed as the relative running time per cycle, the modified approach took approximately as twice the time spent by the original algorithm. The main reason for this overhead is associated with the computation, at each cycle, of a new curve fitting the m samples, comprising the training set, see Section II-C.

IV. Conclusions

This paper focussed on online detection of outliers based on a Least Squares-Support Vector Machine algorithm, under the form of a Reproducing Kernel Hilbert Space (RKHS) with Radial Basis Function (RBF) kernel, along with a sliding window-based learning technique. In order to improve the sensitivity and specificity of this method in transient time series, this work proposed a modification to the RBF kernel that is characterised by replacing the Euclidean norm between adjacent samples with the norm of the respective differences to a Least Squares estimates. The two algorithms were compared in terms of detection performance using a time series generated with a nonlinear virtual system, and corrupted with white noise and outliers. Simulation results have demonstrated the out-performance of the proposed approach.

Acknowledgment

This work has been partially supported by Project CENTRO-07-ST24-FEDER-002003 (iCIS-Intelligent Computing in the Internet of Services).

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.

[3] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302–1325, 2011.

[4] M. A. Rassam, A. Zainal, and M. A. Maarof, "Advancements of data anomaly detection research in wireless sensor networks: A survey and open issues," *Sensors*, vol. 13, no. 8, pp. 10087–10122, 2013.

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

[6] 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, 2006. Special Section: Advances in Signal Processing-assisted Cross-layer Designs.

[7] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*, vol. 3. Springer, 2004.

[8] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels," *Information Theory, IEEE Transactions on*, vol. 52, no. 10, pp. 4635–4643, 2006.

[9] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2961–2974, 2005.

[10] M. S. Bartlett, "An inverse matrix adjustment arising in discriminant analysis," *The Annals of Mathematical Statistics*, pp. 107–111, 1951.

[11] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[12] G. P. Liu, V. Kadirkamanathan, and S. A. Billings, "On-line identification of nonlinear systems using volterra polynomial basis function neural networks," *Neural Networks*, vol. 11, no. 9, pp. 1645–1657, 1998.

[13] C. D. Brown and H. T. Davis, "Receiver operating characteristics curves and related decision measures: A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 80, no. 1, pp. 24–38, 2006.

[14] P. Gil, C. Lucena, A. Cardoso, and L. Palma, "Gains tuning of fuzzy pid controllers for mimo systems: A performance-driven approach," *Fuzzy Systems, IEEE Transactions on*, vol. pp, p. 1, Jun 2014.