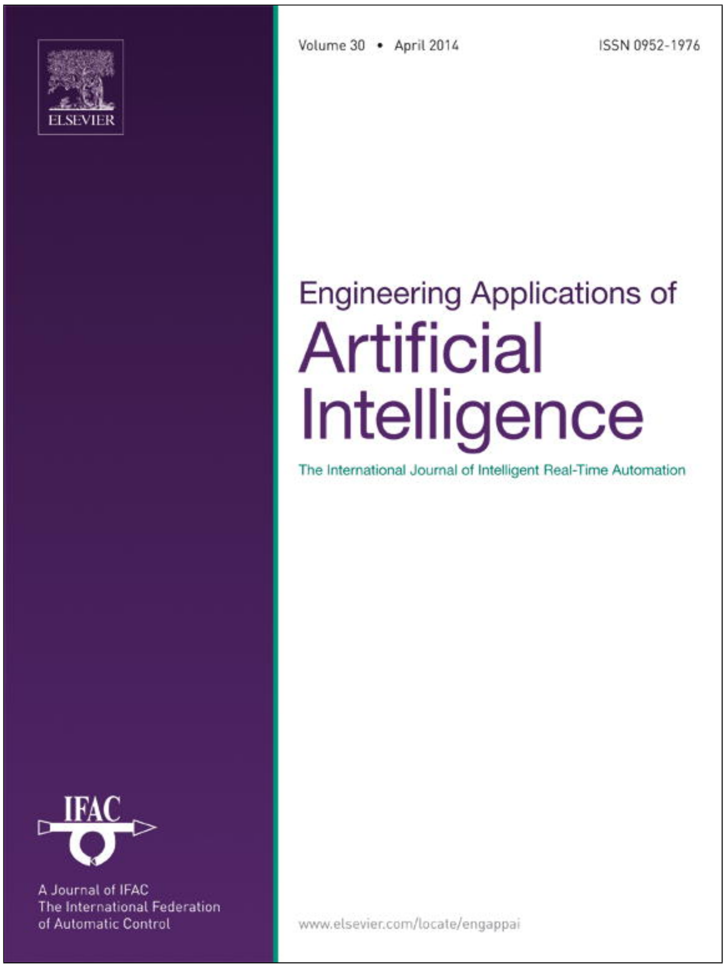


Volume 30 • April 2014 ISSN 0952-1976

ELSEVIER

Engineering Applications of
Artificial
Intelligence

The International Journal of Intelligent Real-Time Automation

IFAC

A Journal of IFAC
The International Federation
of Automatic Control

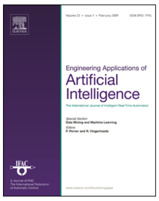
www.elsevier.com/locate/engappai

Engineering Applications of Artificial Intelligence 30 (2014) 18–29

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Combining learning in model space fault diagnosis with data validation/reconstruction: Application to the Barcelona water network

J. Quevedo [a],*, H. Chen [d], M.À. Cugueró [a], P. Tino [c], V. Puig [b], D. García [a], R. Sarrate [a], X. Yao [c]

[a] Advanced Control Systems (SAC), Automatic Control Department (ESAII), UPC, Rambla Sant Nebridi, 1008222 Terrassa, Spain
[b] Advanced Control Systems (SAC), Automatic Control Department (ESAII), UPC, Institut de Robòtica i Informàtica Industrial (IRI), CSIC-UPC, Pau Gargallo, 508028 Barcelona, Spain
[c] School of Computer Science, The University of Birmingham Edgbaston Birmingham B15 2TT, United Kingdom
[d] UBRI, University of Science and Technology of China Hefei, Anhui, China, 230027

ARTICLE INFO

Article history:
Received 20 July 2013
Received in revised form
3 November 2013
Accepted 13 January 2014
Available online 14 February 2014

Keywords:
Learning in model space
Sensor data validation/reconstruction
Time series
Fault diagnosis
Reservoir computing

ABSTRACT

In this paper, an integrated data validation/reconstruction and fault diagnosis approach is proposed for critical infrastructure systems. The proposed methodology is implemented in a two-stage approach. In the first stage, sensor communication faults are detected and corrected, in order to facilitate a reliable dataset to perform system fault diagnosis in the second stage. On the one hand, sensor validation and reconstruction are based on the combined use of spatial and time series models. Spatial models take advantage of the (mass-balance) relation between different variables in the system, whilst time series models take advantage of the temporal redundancy of the measured variables by means of Holt-Winters time series models. On the other hand, fault diagnosis is based on the learning-in-model-space approach that is implemented by fitting a series of models using a series of signal segments selected with a sliding window. In this way, each signal segment can be represented by one model. To rigorously measure the 'distance' between models, the distance in the model space is defined. The deterministic reservoir computing approach is used to approximate a model with the input–output dynamics that exploits spatial–temporal correlations existing in the original data. Finally, the proposed approach is successfully applied to the Barcelona water network.

1. Introduction

Critical infrastructure systems (CIS), including water, gas or electricity networks, are complex large-scale systems, geographically distributed and decentralized with a hierarchical structure, requiring highly sophisticated supervisory and real-time control (RTC) schemes to ensure high performance achievement and maintenance when conditions are non-favourable (Schütze et al., 2004; Marinaki and Papageorgiou, 2005) due to e.g. sensor and actuator malfunctions (faults). Each sub-system composing the CIS is constituted of a large number of elements with time-varying behaviour, having many different operating modes and subject to changes due to operational constraints. To deal with this problem, the use of an on-line fault diagnosis system able to detect such faults and correct them by activating different kinds of techniques e.g. data validation/reconstruction of sensor faults is desirable. This will prevent the RTC from being stopped every time that a fault appears, which is one of the main reasons why global RTC is not widely applied in the world (Schütze et al., 2004). Furthermore, the fault diagnosis process intends to identify which fault is causing the monitored events, including e.g. hardware and software faults.

Generally, two main strategies are available in the literature when addressing the fault diagnosis problem, which are hardware redundancy (preferred in critical systems) based on the use of extra sensors and actuators, and analytical redundancy, based on the use of software sensors or models combining information gathered by the sensor measurements or using other actuators to compensate the faulty ones. Nevertheless, the use of hardware redundancy in large-scale systems is very expensive and increases the number of maintenance and calibration operations, which calls for the use of combined hardware and analytical redundancy approaches in CIS (Carrozza et al., 2008). The capability to detect and isolate faults in these systems is important to keep their integrity safe. This problem has been targeted by numerous researchers from many different points of view, as overviewed in the compilation of techniques included in Venkatasubramanian et al. (2003a–c), and more recently in Ding (2008).

In this paper, an innovative framework that investigates the fault diagnosis problems in the model space instead of the data/signal space is developed. This fault diagnosis framework is integrated with a data validation/reconstruction methodology introduced in Quevedo et al. (2010a). 'Learning in model space'

* Corresponding author.
E-mail address: joseba.quevedo@upc.edu (J. Quevedo).

0952-1976/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2014.01.008

(Chen et al., 2014) is implemented by fitting a series of 'approximated' models using a series of signal segments selected with a sliding window, and then apply the learning techniques to discriminate and isolate fault models from healthy models. Reservoir computing is chosen as an example to approximate the signal segments. Dynamic reservoirs of reservoir models have been shown to be 'generic' in the sense that they are able to represent a wide variety of dynamical features of the input driven signals, so that given a task at hand only the linear readout on top of reservoir needs to be retrained (Lukoševičius and Jaeger, 2009). Hence, in the formulation the underlying dynamic reservoir will be the *same* throughout the signal – the differences in the signal characteristics at different times will be captured solely by the linear readout models and will be quantified in the function space of readout models.

Here it is assumed that, for some sufficiently long initial period, the system is in a 'normal/healthy' regime so that when a fault occurs the readout models characterizing the fault will be sufficiently 'distinct' from the normal ones. A variety of novelty/anomaly detection techniques can be used for the purposes of detection of deviations from the 'normal' regime. In this contribution, support vector machines (SVMs) in the readout model space are used, so new faults occurring will be captured by the algorithm proposed operating in the readout model space.

The contributions of this paper are listed as follows:

- First, data validation and reconstruction techniques are integrated with learning in the model space for effective fault diagnosis.
- Second, SVMs are used in the model space for fault detection/isolation.

Finally, the proposed methodologies are applied to the Barcelona water network as a case study in this paper.

2. Data validation/reconstruction approach

In systems like CIS, a telecontrol system is acquiring, storing and validating data gathered from different kinds of sensors every given sampling time to accurately real-time monitor the whole system. In this process, problems in the communication system, e.g. between sensors and data loggers or in the telecontrol system itself, are frequent and produce data loss which may be of great concern in order to have valid historic records. When this is occurring, lost data should be replaced by a set of forecasted data which should be a representative of the data lost. Another common problem in CIS is caused by the unreliable sensors, which may be affected by e.g. offset, drift, freezing in the measurements (Kanakoudis and Tolikas, 2001; Kanakoudis and Tsitsifli, 2011; Tsitsifli et al., 2011). These unreliable data should also be detected and replaced by forecasted data, since it may be used for system management tasks e.g. maintenance, planning, investment plans, billing, security and operational control (Quevedo et al., 2010a) and system fault detection and isolation (Fig. 1).

Different types of data validation methods with distinct degrees of complexity may be considered according to the available system knowledge. Generally, two types of methods can be considered, one for elementary 'low-level' signal based methods and another for 'high-level' model-based methods. The first class use simple heuristics and limited statistical information from the sensors (Burnell, 2003; Jörgensen et al., 1998) and is typically based on checking either signal values or variations, whilst the second class uses models for consistency-checking of the sensor data (Tsang, 2003). Here, the first class of data validation methods has been used to deal with sensor communication faults.

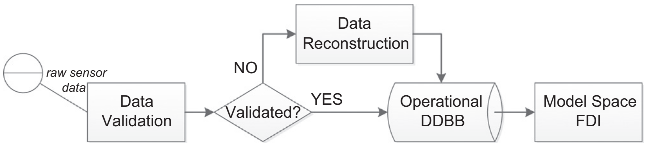


Fig. 1. Raw data validation/reconstruction and system FDI approach.

2.1. Data validation process

The data validation process is inspirited in the Spanish AENOR-UNE norm 500540 (Quevedo et al., 2010a). The methodology applies a set of consecutive validation tests to a given dataset (Fig. 2), to finally assign a certain quality level depending on the tests passed.

In a system like the one considered here, and in telecontrolled systems in general, one of the most common faults occurring is sensor communication fault. This type of fault is related with level zero of the sensor validation methodology in Quevedo et al. (2010a). This level checks whether the data is properly recorded, assuming that data acquisition systems sample data at a certain fixed rate. Hence, this level allows detecting problems in the data acquisition or communication system.

Here, communication faults are considered as the faults affecting the sensor of the telecontrol system, and data validation and reconstruction procedures are used as a prefilter to estimate the missing data when this type of faults is occurring.

2.2. Data reconstruction process

The output of the data validation process (Fig. 2) is used to identify the invalidated data that should be reconstructed. Spatial and time series (TS) models (Levels 4 and 5 in Fig. 2) are used for this purpose, depending on the performance of each model.

On the one hand, spatial models (SM) take advantage of the relation between different variables emplaced in the system. For example, in hydraulic systems, this relation is generally obtained from the mass balance model of the element relating the different measured variables involved, which states that the incoming and outcoming flows in a tank subsystem must be equal

$$\hat{x}_{SM}(k) = x(k-1) + \Delta t(q_{in}(k-1) - q_{out}(k-1)) \qquad (1)$$

where $\hat{x}_{SM}$ is the spatial model tank volume estimation, x is the measured tank volume, q_{in} is the incoming tank flow, q_{out} is the outcoming tank flow and Δt is the sampling time. From this equation, the volume estimation for a particular tank subsystem may be stated. Estimation of other variables (e.g. $\hat{q}_{in}$, $\hat{q}_{out}$) may be obtained from algebraic manipulation of the latter.

However, real elements include uncertainty (due to e.g. unexpected behaviour of the plant, inaccuracy of the model) which may lead to the non-satisfaction of the mass balance in the element considered. Hence, consistency of the data collected by a certain sensor with its spatial model (Quevedo et al., 2010b), i.e. the correlation between data coming from spatially related sensors, may be maintained. For example, the data of the flow meters located in different points of the same pipe in a transport water network allows for checking the reliability of the sensor set and performing the corresponding correction, e.g. by using a linear regression model of input–output measured data in the pipe (which is ideally the identity function). In the case of the tank level estimation (1), this correction is introduced as

$$\hat{x}_c(k) = a\hat{x}_{SM}(k) + b \qquad (2)$$

where $\hat{x}_c$ is the corrected estimation of the volume using regression model parameters $[a, b]$ obtained with the training dataset

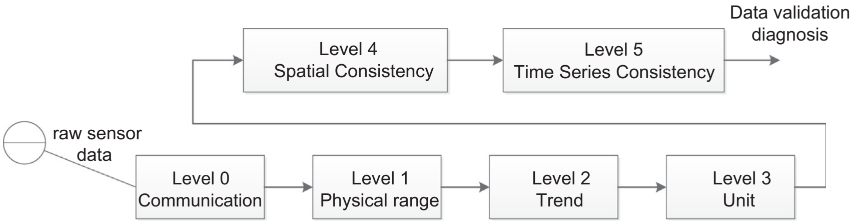


Fig. 2. Data validation tests.

and $\hat{x}_{SM}$ is the spatial model volume estimation using measured variables q_{in} and q_{out}. Parameters $[a, b]$ equal $a = 1$ and $b = 0$ in the ideal situation, i.e. when the estimated value $\hat{x}_{SM}(k+1)$ equals the measured value $x(k+1)$.

On the other hand, TS models take advantage of the temporal redundancy of the measured variables. A widely used method for signal forecasting is the Holt-Winters (HW) approach (Makridakis et al., 1998; Winters, 1960). This method is widely used because of its simplicity. There are various versions of this method e.g. additive or damped trend, additive or multiplicative seasonality, single or multiple seasonality. Here, good performance has been attained with the additive single seasonality version. In this case, the forescasted value of which is obtained in Eq. (3) for a forecasting horizon ℓ

$$\hat{x}_{TS}(k) = \overline{R}(k-\ell) + \ell\overline{G}(k-\ell) + \overline{S}(k-L) \tag{3}$$

where $\overline{R}$ is the level estimation removing seasonality

$$\overline{R}(k-\ell) = \alpha(x(k-\ell) - \overline{S}(k-L-\ell)) + (1-\alpha)(\overline{R}(k-\ell-1) + \overline{G}(k-\ell-1)), \quad 0<\alpha<1 \tag{4}$$

$\overline{G}$ is the trend estimation

$$\overline{G}(k-\ell) = \beta(\overline{R}(k-\ell) - \overline{R}(k-\ell-1)) + (1-\beta)\overline{G}(k-\ell-1), \quad 0<\beta<1 \tag{5}$$

$\overline{S}$ is the seasonal component estimation

$$\overline{S}(k-\ell) = \gamma(x(k-\ell) - \overline{R}(k-\ell)) + (1-\gamma)\overline{S}(k-\ell-L), \quad 0<\gamma<1 \tag{6}$$

and L is the season (daily) periodicity, α, β and γ are the HW parameters (level, trend and season smoothing factors, respectively), x is the measured value and $\hat{x}_{TS}(k+T)$ is the TS model forecasted value.

Hence, analysing the historic records of a certain sensor, a HW TS model is derived and used to forecast the data of this element when a fault is affecting its readings.

The models' accuracy is measured by the mean squared error (MSE) index, evaluated in the $n=48$ previous values to k. The model having the best MSE index when the communication fault is produced (i.e. when the data validation process is not satisfactory) is used to produce the reconstructed sensor signal.

3. Fault diagnosis approach

To address the challenges in learning based fault diagnosis, this section applies deterministic reservoir computing as a model to approximate complex multi-input/multi-output systems, and introduces the learning in the model space approach in the Barcelona water network.

3.1. Deterministic reservoir model

Reservoir computing (RC) (Lukoševičius and Jaeger, 2009) is a class of state space models based on a 'fixed' randomly constructed state transition mapping, realized through the so-called reservoir and a trainable (usually linear) readout mapping from the reservoir. Popular RC methods include echo state networks (ESNs) (Jaeger, 2001), liquid state machines (LSMs) (Maass et al., 2002) and the back-propagation decorrelation neural network (Steil, 2004).

This paper is based on echo state networks, which are one of the simplest yet effective forms of RC. Generally speaking, ESNs are recurrent neural networks with a non-trainable sparse recurrent part (reservoir) and a simple linear readout. Typically, the reservoir connection weights and the input weights are randomly generated.

In order to satisfy the sufficient condition for the 'echo state property', i.e. the reservoir is an 'echo' of the entire input history, the weights in the reservoir are scaled to ensure that the spectral radius of these weight matrix W is less than one. ESNs have been successfully applied to many applications, such as short-term stock price prediction (Lin et al., 2009), speech recognition (Skowronski and Harris, 2007), nonlinear adaptive filtering (Xia et al., 2010), and analysing grammatical structure (Tong et al., 2007).

Given the input signal $\mathbf{u}$ and output signal $\mathbf{y}$, the reservoir model with N reservoir (state) units can be formulated as follows:

$$\mathbf{x}(t) = \tanh(R\ \mathbf{x}(t-1) + V\ \mathbf{u}(t)), \tag{7}$$

$$f(\mathbf{x}(t)) = W\mathbf{x}(t) + \mathbf{a}, \tag{8}$$

where $\mathbf{x}(t) = [x_1, \dots, x_N]^T \in \Re^N$ is the state vector of reservoir activations, $\mathbf{u}(t)$ is the input signal at time t, $\tanh(\cdot)$ is the state-transition function of the reservoir, and $f(\mathbf{x}(t))$ is the output[1] of the linear readout from the reservoir; R is a $(N \times N)$ dynamic coupling weight matrix; V and W are the input and output weight matrices, respectively.

In order to construct a reservoir model, a number of parameters should be specified, such as the reservoir size, sparsity of reservoir, input connections, the reservoir topology and its weights. The input connections, reservoir topology and the connection weights are randomly generated and fixed during the readout training. This specification often requires numerous trails and even luck (Xue et al., 2007), and the weight structure and the random connectivity in the reservoir are unlikely to be optimal (Ozturk et al., 2007). These problems are preventing ESNs from becoming a widely accepted tool.

There has been research that proposes fixed and simple reservoir topologies as an alternative to the randomized reservoir, including feedforward ESNs model with tapped delay connections (Čerňanský and Tiňo, 2008), simple cycle topology (Rodan and Tiňo, 2011) and simple cycle topology with regular jumps (Rodan and Tiňo, 2012). The simple cycle reservoir with regular jumps (CRJ) could identify the minimum complexity in constructing reservoirs, which is competitive with other ESNs algorithms. The

[1] The output signal y is the desired output and $f(x(t))$ is the actual MODEL output. In the framework of "learning in the model space", a good representation of multiple-input and multiple-output system by readout mapping is achieved by minimizing the difference between $y(t)$ and $f(x(t))$.

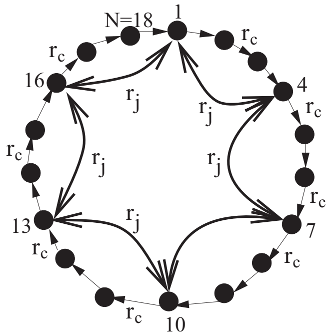


Fig. 3. Illustration of the CRJ model. The model has a fixed topology, i.e. a uni-directional cycle with bi-directional jumps. There are only three parameters in this model: the cyclic connection weights $r_c > 0$; the jumps weight $r_j > 0$; and the input weight $r_i > 0$.

deterministic CRJ can approach with arbitrary accuracy the memory capacity of any non-linear recurrent networks.

The CRJ model is deterministically constructed and extremely simple. In CRJ, the structure of R is particularly simple: reservoir units are connected in a uni-directional cycle with bi-directional shortcuts (jumps) (see Fig. 3). All cyclic connections have the same weight $r_c > 0$ (where the subscript c is the first letter of cycle and indicates that the related r is the cycle weight, the subscripts i and j in notation r_i and r_j appeared later have the same meaning). Likewise, all jumps share the same weight $r_j > 0$. The input weight matrix is also highly constrained: the input connections have the same absolute value $r_i > 0$. The state space model used in this paper is illustrated in Fig. 3.

In that case, the CRJ model has a small degree of local clustering and a small average path length. Jaeger and Haas Jaeger and Haas, 2004 suggested that reservoirs should ideally have small clustering degree (sparse reservoirs) to spread the dynamic information in the reservoir and a small average path length can be beneficial for a variety of dynamical time scales. The structure of the CRJ model is illustrated in Fig. 3.

The traditional randomize RC is largely driven by a series of randomized model building stages, which could be unstable and hard to understand, especially for fault diagnosis. In this paper, we propose to use the deterministic reservoir algorithm, i.e. CRJ, to fit the signals in fault diagnosis, because CRJ can approximate the nonlinear function with inner memory and due to the linear training, the CRJ model can be trained fast and run in real-time.

3.2. Reservoir model distance

Learning in the model space is proposed by Chen et al. (2014) based on the idea of using a model to approximate multi-input/multi-output systems that generate a model-induced feature space, and then construct some machine learning algorithms in the model-induced feature space.

There are several ways to generate the model-induced feature space from the original signal space. One possible way is to use the parameter vector of each model as an individual point in the model space. Generally speaking, the parametric space is not equivalent to the model space, especially for non-linear models. The following will define the distance in the model space.

In the model-induced feature space, the 2-norm distance between model $f_1(\mathbf{x})$ and model $f_2(\mathbf{x})$ can be defined as follows:

$$L_2(f_1,f_2) = \left(\int_C (f_1(\mathbf{x}) - f_2(\mathbf{x}))^2 \, dp(\mathbf{x})\right)^{1/2}, \tag{9}$$

where $(f_1(\mathbf{x}) - f_2(\mathbf{x}))^2$ is to measure the difference between $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$, $p(x)$ is the probability density function of $\mathbf{x}$ and C is the integral range. Here, x is assumed to be uniformly distributed.[2]

In the following, the application of the distance definition in the model space for linear models has been demonstrated. The linear model, e.g. auto regressive model, reservoir model,[3] can be represented by the following equation:

$$f_1(\mathbf{x}) = W_1\mathbf{x} + \mathbf{a},$$

where $\mathbf{x} = [x_1, \ldots, x_N]^T$ is a state vector or basis function, N is the number of variables in the model, W_1 is the parameters ($O \times N$ matrix) in the model, O is the number of output vectors and $\mathbf{a} = [a_1, \ldots, a_o]$ is the bias vector for output nodes.

The distance between two readouts from the *same* reservoir can be calculated based on the following equation (Chen et al., 2014):

$$L_2(f_1,f_2) = \left(\int_C \|W\mathbf{x}\|^2 + \|\mathbf{a}\|^2 \, d\mathbf{x}\right)^{1/2} = \left(\frac{2^N}{3}\sum_{j=1}^{N}\sum_{i=1}^{O} w_{i,j}^2 + 2^N \|\mathbf{a}\|^2\right)^{1/2}$$

where $\mathbf{w}_i^T$ is the i-th row of W, $w_{i,j}$ is the (i,j)-th element of W, $f_1(\mathbf{x}) = W_1\mathbf{x} + \mathbf{a}_1$, $f_2(\mathbf{x}) = W_2\mathbf{x} + \mathbf{a}_2$, $W = W_1 - W_2$ and $\mathbf{a} = \mathbf{a}_1 - \mathbf{a}_2$.

Scaling the squared model distance ($L_2^2(f_1,f_2)$) by 2^{-N} leads to

$$\frac{1}{3}\sum_{j=1}^{N}\sum_{i=1}^{O} w_{i,j}^2 + \|\mathbf{a}\|^2,$$

which differs from the squared Euclidean distance of the readout parameters

$$\sum_{j=1}^{N}\sum_{i=1}^{O} w_{i,j}^2 + \|\mathbf{a}\|^2,$$

by the factor 1/3 applied to the differences in the linear part W of the affine readouts. Hence, more importance is given to the ‘offset’ than to the ‘orientation’ of the readout mapping.

The above analysis assumed that the distribution of $\mathbf{x}$ is uniform in the integral range C. When the distribution of $\mathbf{x}$ is non-uniform, sampling techniques and analytical techniques using e.g. a Gaussian mixture model can be employed to calculate the distance.

3.3. Support vector machines for fault diagnosis

In fault diagnosis, it should be determined whether a running sub-system (or component) is under a healthy operation condition, or whether a faulty situation is occurring. Based on the ‘learning in the model space’ framework, support vector machines will be employed in the model space for fault diagnosis.

Support vector machines are supervised learning models that analyse data and recognize patterns used for classification. The traditional SVM is only applicable for binary classification. The one-against-one approach and the one-against-the-rest approach can be employed to extend it for multi-class classification. Here, one-the against-one approach is used for multi-class learning problems since it is more efficient than the one-against-the-rest approach.

In the learning in the model space approach, a sliding window is used to select signal segments to train reservoir models. The sliding window is moved forward by one step at a time, which can reduce fault detection delays. When the sliding window is moving forward, more reservoir models are obtained. Thus, the pairwise

[2] In practice, non-uniform distribution of $\mathbf{x}$ could be employed by estimating the density of $\mathbf{x}$ using Gaussian mixture models and sampling techniques for more accurate distance measures.

[3] Reservoir model can approximate non-linear mapping using a linear readout training by keeping the randomly constructed non-linear reservoir unchanged in the training.

model distance $L_2(f_i,f_j)$ can be obtained. The calculated model distance will be used in the Gaussian RBF kernel to replace the data distance with the *model distance* $L_2(f_i,f_j)$

$$\phi_\sigma(f_i,f_j)=\exp\{-\sigma\cdot L_2(f_i,f_j)\}.$$

where σ is the kernel parameter and will be tuned by cross validation.

4. Case study: Barcelona water supply network

4.1. Description

The Barcelona water network, used as the case study here, supplies water to about three million consumers, distributed in 23 different municipalities covering an area of 424 km^2. Water supplied may be extracted from both surface and underground sources, Ter (surface source) and Llobregat (both surface and underground source) being the most important ones in terms of capacity and use. Water coming from these sources is conducted through about 4645 km of pipes to 218 different demand sectors (district metered areas – DMAs) with about 400 control points installed. Currently, the Barcelona SCADA system receives real-time data gathered by 200 local monitoring stations, mainly flow meters and some pressure sensors. Most of these flow meter control points are installed in the inlet of the DMA, so that their measurement is highly representative of the whole DMA water demand. Summarizing, the complete Barcelona supply network is modelled by 63 storage tanks, 7 underground sources, 3 surface sources, 79 pumps, 50 valves, 18 nodes and 88 demands.

Regarding the actual SCADA system implemented in the network, a datalogger is recording raw data every 10 min from the ground sensors. These data are sent once a day using the ModBus communication protocol via the GSM radio/telephone XTC network to an ORACLE operational database of the telecontrol system (Fig. 4) which is implemented by a TopKapi SCADA (http://www.areal.fr/). These collected data are hourly averaged by the company and supplied for further use.

Before utilising the sensor data recorded by the SCADA system in the network management system, these data must be treated with two operations, which are the raw data insertion process and the data validation and replacement process.

4.2. Raw data insertion process

This is the procedure involving the data acquisition by the operational database in the SCADA system from the data-loggers, which may be affected by different kinds of malfunctions e.g. communication faults or missing sensor/data-logger data. This data loss, which will affect further management processes such as statistic and hydraulic balance studies, must be recovered by the use of replacement artificial data (Tsitsifli and Kanakoudis, 2009). The fault detection in this particular casuistic is trivial, since the data gap is easily detected and reported by the communication system itself by an attached error message. A more challenging problem is the replacement of these lost data by some virtual dataset which is a representative of the missed sensor readings if the malfunction would not occurred.

4.3. Data pre-filter

At present, there is an automatic data pre-filter, using data validation and reconstruction procedures implemented at each control point. This pre-filtering process, described in Section 2, is activated when a sensor communication fault is occurring, which is detected at Level 0 test in the data validation procedure (Fig. 2), in order to achieve a complete sensor dataset to fulfil further methods needing this information, despite this faulty situation.

5. Results

5.1. Fault scenarios

To illustrate the methodologies proposed here, a part of the Barcelona water network that includes two subsystems, known as Orioles and Cervelló (Figs. 5 and 6, respectively), is being used. This part of the network includes the following elements:

Tanks: *d*175*LOR*, *d*147*SCC*, *d*205*CES*, *d*263*CES*
Actuators with flow sensors: *iOrioles*, *iStaClmCervello*, *iCesalpina*1, *iCesalpina*2
Demands with flow sensors: *c*175*LOR*, *c*147*SCC*, *c*205*CES*, *c*263*CES*
Level sensors: *xd*175*LOR*, *xd*147*SCC*, *xd*205*CES*, *xd*263*CES*

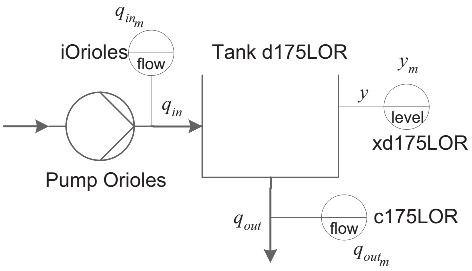


Fig. 5. Orioles subsystem.

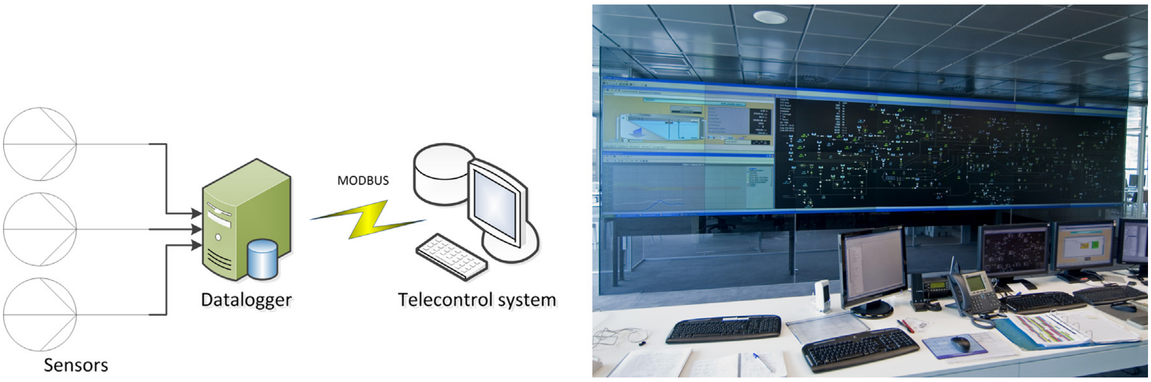


Fig. 4. Telecontrol of Barcelona water network system.

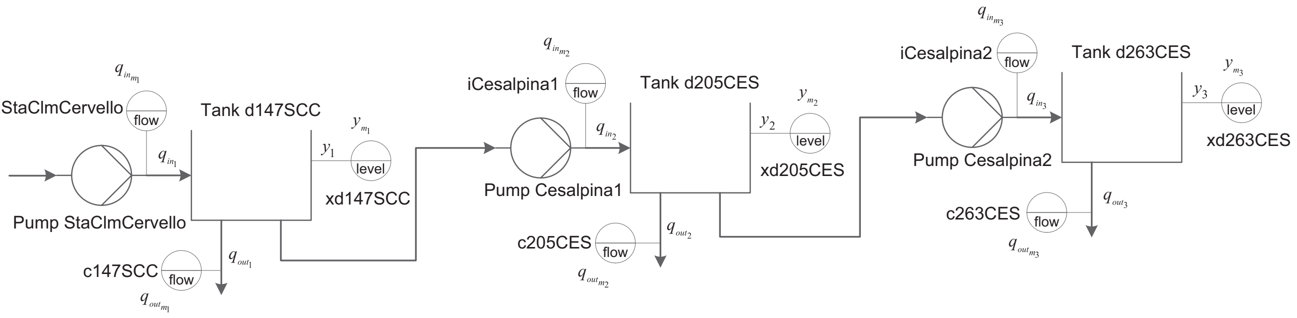


Fig. 6. Cervelló subsystem.

Table 1
Parameterizations of faults. MFD stands for maximum flow/demand.

ID	Faulty element	Type	Magnitude
1	iOrioles	1	−25%
2	iOrioles	2	−25%
3	iOrioles	2	−10%
4	iOrioles	3	0.001%
5	iOrioles	3	0.1%
6	iOrioles	4	0.1%
7	iOrioles	4	0.01%
8	iOrioles	5	–
9	iOrioles	6	2
10	c175LOR	1	−20%
11	c175LOR	2	−15%
12	c175LOR	3	0.01%
13	c175LOR	4	0.1%
14	c175LOR	5	–
15	iStaClmCervello	1	−15%
16	iStaClmCervello	2	−7.5%
17	iStaClmCervello	3	0.01%
18	iStaClmCervello	4	0.5%
19	iStaClmCervello	5	–
20	iStaClmCervello	6	4
21	iCesalpina1	1	10%
22	iCesalpina1	2	−15%
23	iCesalpina1	3	0.01%
24	iCesalpina1	4	0.25%
25	iCesalpina1	5	–
26	iCesalpina1	6	0.75%
27	c263CES	1	30%
28	c263CES	2	−15%
29	c263CES	3	0.025%
30	c263CES	4	0.25%
31	c263CES	5	–
32	iOrioles	7	–
33	c175LOR	7	–

Type	Details and parameter
1	Additive offset (%MFD)
2	Additive incipient offset (%MFD)
3	Noise (variance %MFD)
4	Additive drift (%MFD)
5	Abrupt freezing (–)
6	Multiplicative offset (divided by)
7	Communication fault (–)
8	

A simulator of the network has been developed in MATLAB/SIMULINK in order to test the proposed methodologies before being implemented in the real system. The model used by this simulator has been fully calibrated/validated using real-world data, providing a good degree of representativeness of the real network behaviour (in terms of MSE index) after this process (Pascual et al., 2013).

The simulator also includes the possibility to introduce faults in the different elements of the network. The faults that will be studied are located in actuator sensors and demand sensors. The

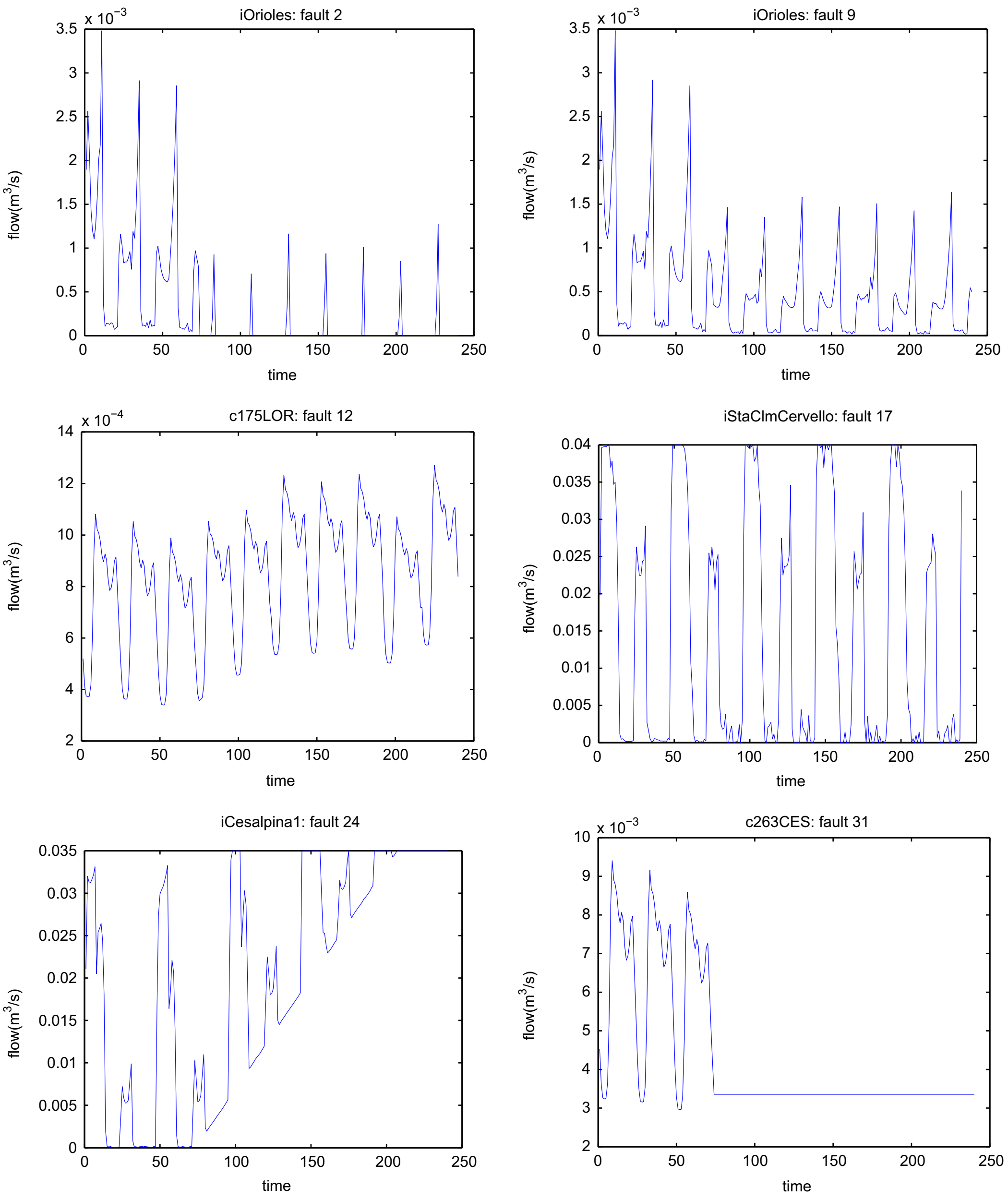


Fig. 7. Examples of faulty signals.

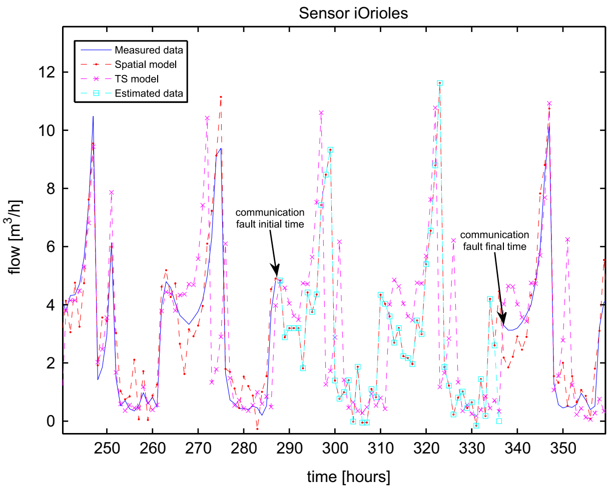


Fig. 8. Communication fault in iOrioles pump sensor.

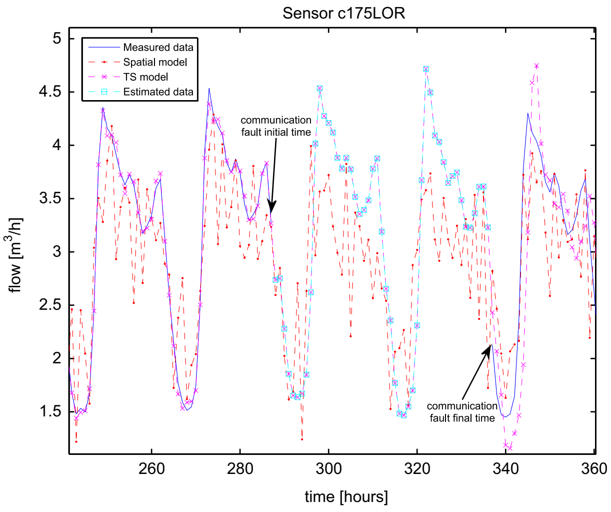


Fig. 9. Communication fault in c175LOR demand sensor.

simulator allows introducing different kinds of faults in these elements, which are detailed next:

Sensor communication: When this fault is occurring, a loss of new measured data gathered by the sensor affected is produced.

Freezing: When this fault is taking place, the signal remains constant after a given fault initial time.

Offset: When this fault is occurring, an unknown constant magnitude is added to the sensor measurements after a given fault initial time

Drift: When this fault appears, a ramp function is added to the sensor measurements after a given fault initial time

Sensor noise degradation: When this fault occurs, a zero-mean Gaussian noise with variance σ^2 is added to the sensor measurements after a given fault initial time.

To test and adjust the methodology proposed in this paper, different fault scenarios have been defined, all considering a dataset of one month period. On the one hand, sensor communication faults have been considered, appearing along the 12th day (before any other system fault is appearing) and causing the loss of new recorded data of the sensor affected whilst they are occurring. In many actual telecontrolled systems, these faults are notified by an external indicator provided by the communication system itself, with its detection being a trivial problem. On the other hand, further system faults have been considered, appearing at the beginning of the 16th day. The system faults considered are both abrupt or incipient nature. The parametrization of these faults is depicted in Table 1.

Several examples of faulty signals are illustrated in Fig. 7.

5.2. Data validation/reconstruction results

One month of data has been used to perform sensor data validation and reconstruction: the first four days of data have been used for model training (both spatial and TS, as described in Section 2) and the rest of the data have been used to test the method. The validation and reconstruction algorithms have been implemented in an on-line fashion, which is of paramount importance for a method to be applied in a real-time system. Sensor communication faults have been considered, particularly in demand and actuator sensors, as detailed in Section 5.1. These faults are occurring along day 12 (hour 288) and endure 2 days

Table 2
Fault detection ability of iOrioles.

Fault	SVM			CART			Bagging			Adaboost		
	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay
iOrioles-1	100	0	0	100	0	25	100	0	25	100	0	25
iOrioles-2	98.02	0	13	92.88	0.23	9	94.72	0	79	93.14	0.06	78
iOrioles-3	91.42	0	44	81.27	0.88	48	79.82	0	79	60.16	0.20	116
iOrioles-4	98.81	0	1	88.79	0.15	3	95.78	0	6	95.38	0.59	4
iOrioles-5	99.74	0	0	99.34	0.03	2	99.60	0	3	99.34	0	3
iOrioles-6	88.79	0	13	65.83	0.88	11	67.28	0	12	4.22	0.32	157
iOrioles-7	40.90	0.91	79	35.22	2.78	15	26.52	0	79	1.58	0.09	225
iOrioles-8	100	0	0	100	0	33	100	0	40	100	0	40
iOrioles-9	100	0	0	97.76	0.09	25	98.81	0	25	97.76	0.09	25

Table 3
Fault detection ability of c175LOR.

Fault	SVM			CART			Bagging			Adaboost		
	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay
c175LOR-1	50.79	0	0	20.32	2.99	3	15.04	0	1	3.82	1.38	237
c175LOR-2	41.69	0	24	21.64	3.16	13	11.48	0	79	1.98	1.38	212
c175LOR-3	100	0	0	100	0	4	100	0	4	0	0	113
c175LOR-4	82.45	0	30	44.20	1.64	2	39.45	0	59	4.49	1.02	150
c175LOR-5	100	0	0	100	0	0	100	0	0	100	0	0

Table 4
Fault detection ability of iStaClmCervello.

Fault	SVM			CART			Bagging			Adaboost		
	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay
iStaClm-1	95.12	0	2	81.79	0.53	2	82.32	0	16	44.06	2.25	101
iStaClm-2	76.39	0.44	74	63.06	1.20	6	63.85	0	79	25.20	2.05	110
iStaClm-3	95.12	0	3	85.09	0.56	1	93.54	0	1	59.89	3.37	2
iStaClm-4	96.31	0	17	89.45	0.12	47	91.69	0	47	79.68	0.38	47
iStaClm-5	100	0	0	100	0	0	100	0	0	100	0	0
iStaClm-6	99.47	0	0	96.31	0.12	9	96.83	0	18	95.38	0.18	19

Table 5
Fault detection ability of iCesalpina1.

Fault	SVM			CART			Bagging			Adaboost		
	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay
iCesalp-1	80.21	0	2	52.24	1.34	19	47.76	0	23	7.12	0.50	17
iCesalp-2	94.06	0	79	69.92	1.17	17	68.47	0	79	23.49	1.02	22
iCesalp-3	100	0	1	84.17	0.56	7	89.05	0	17	67.15	1.41	24
iCesalp-4	99.87	0	21	86.81	0.59	18	90.50	0	24	72.69	1.64	24
iCesalp-5	100	0	0	100	0	0	100	0	0	100	0	0
iCesalp-6	98.55	0	16	66.62	0.88	2	69.00	0	23	47.63	2.66	17

Table 6
Fault detection ability of c263CES.

Fault	SVM			CART			Bagging			Adaboost		
	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay	FDR	FAR	Delay
c263CES-1	29.55	0	0	14.64	3.02	1	0.53	0	4	0.92	0.35	42
c263CES-2	27.7	0	45	21.11	3.07	1	4.49	0	79	0.13	0.15	156
c263CES-3	100	0	0	99.74	0	4	100	0	4	100	0	4
c263CES-4	83.64	0	16	58.58	1.05	16	60.68	0	43	25.07	1.90	117
c263CES-5	100	0	0	100	0	0	100	0	0	100	0	0

Table 7
Confusion matrix of iOrioles.

Fault Name	iOrioles-1	iOrioles-2	iOrioles-3	iOrioles-4	iOrioles-5	iOrioles-6	iOrioles-7	iOrioles-8	iOrioles-9
iOrioles-1	758	0	0	0	0	0	0	0	0
iOrioles-2	8	728	3	0	0	9	1	0	9
iOrioles-3	2	19	643	1	0	37	38	0	18
iOrioles-4	0	7	0	738	3	0	1	0	0
iOrioles-5	0	0	0	0	758	0	0	0	0
iOrioles-6	0	10	19	2	1	633	81	0	12
iOrioles-7	0	6	38	2	0	75	633	0	4
iOrioles-8	0	0	0	0	0	0	0	758	0
iOrioles-9	2	2	6	0	0	1	1	0	746

(until hour 336), whilst there is no new data available from the sensor affected. When the fault occurs, the Level 0 data validation test (Fig. 2) notifies a communication fault and the data reconstruction process is activated. The reconstruction is performed via the SM or TS model, depending on their particular performance, which is measured by the MSE index as described in Section 2, with $n=48$ samples.

First, a communication fault in the Orioles subsystem pump sensor (ID.32 in Table 1) is produced, achieving the results depicted in Fig. 8. Model parameters obtained using the training dataset are $[a=0.627,\ b=1.04]$ for the spatial model and $\alpha=2.698\mathrm{e}-2$, $\beta=1\mathrm{e}-5$, $\gamma=1$ for the TS model with a prediction horizon of $\ell=20$ samples, assuming daily periodicity ($L=24$ samples). When the communication fault is produced, the spatial model obtains better MSE index, so it is used for sensor reconstruction, achieving good performance for this particular sensor.

Also, a communication fault in the Orioles subsystem demand sensor (ID.33 in Table 1) is introduced, reaching the results in Fig. 9. Model parameters obtained using the training dataset are $[a=0.627,\ b=1.04]$ for the spatial model and $\alpha=8.622\mathrm{e}-1$, $\beta=1\mathrm{e}-5$, $\gamma=3.526\mathrm{e}-1$ for the TS model with a prediction horizon of $\ell=20$ samples, assuming daily periodicity ($L=24$ samples). When the communication fault is produced, the TS model obtains better MSE index in this case, so it is used for sensor reconstruction, also attaining suitable performance for this particular sensor.

5.3. Fault diagnosis results

Since the default setting of MATLAB is to optimize the classification and regression trees (CART) algorithm,[4] the default setting in MATLAB for CART has been used. Bagging and Adaboosting are ensemble algorithms with decision trees (CARTs) as based learners (CARTs have been optimized by MATLAB). They have only one parameter[5] to specify, i.e. the number of trees in the ensembles. A popular choice (100 decision trees) is adopted in these comparisons.

The fault detection performance is measured by *fault detection rate* (FDR) and *false alarm rate* (FAR). In fault isolation, the performance is measured by *precision*, *recall* (or sensitivity), and *specificity*. The *precision*, *recall* and *specificity* are defined as follows:

$$precision=\frac{tp}{tp+fp},$$

$$recall=\frac{tp}{tp+fn},$$

$$specificity=\frac{tn}{tn+fp},$$

[4] In matlab function 'classregtree', the default is to compute the full tree and the optimal sequence of pruned subtrees.

[5] Different variants of Bagging and Adaboosting may require more parameters.

where *tp*, *tn*, *fp*, *fn* indicate true positive, true negative, false positive, false negative, respectively. *Precision* measures the proportion of positive test results that are true positives, also referred to as positive predictive value. *Recall* measures the proportion of actual faults which are correctly identified as such and *specificity* measures the proportion of normal data which are correctly identified.

As introduced in Section 5.1, since there are four components with faults, four deterministic reservoir computing models, each with 25 nodes in the reservoir, have been employed in the proposed framework.

For each dataset, 50% of data have been employed as training data and the remaining 50% of data have been used as testing data. The experiments are run for 100 times.

In fault diagnosis, the first step to perform is to discriminate faulty situations from faultless situations. Tables 2–6 reported the fault detection results using the proposed algorithm. In this table, fault detection rate (FDR), false alarm rate (FAR) and detection delay are employed as three different metrics. According to these tables, it is observed that SVM outperformed all other algorithms.

Table 8
Confusion matrix of c175LOR.

Fault Name	c175LOR-1	c175LOR-2	c175LOR-3	c175LOR-4	c175LOR-5
c175LOR-1	422	198	0	138	0
c175LOR-2	217	417	0	124	0
c175LOR-3	0	0	754	4	0
c175LOR-4	112	64	0	582	0
c175LOR-5	0	0	0	0	758

Table 9
Confusion matrix of iStaClmCervello.

Fault Name	iStaClm-1	iStaClm-2	iStaClm-3	iStaClm-4	iStaClm-5	iStaClm-6
iStaClm-1	615	102	17	15	0	9
iStaClm-2	86	645	18	3	0	6
iStaClm-3	81	13	654	2	0	8
iStaClm-4	1	1	4	671	0	81
iStaClm-5	0	0	0	0	758	0
iStaClm-6	9	0	15	16	0	718

Table 10
Confusion matrix of iCesalpina1.

Fault name	iCesalp-1	iCesalp-2	iCesalp-3	iCesalp-4	iCesalp-5	iCesalp-6
iCesalp-1	651	26	20	19	0	42
iCesalp-2	51	604	18	27	0	58
iCesalp-3	15	9	715	17	0	2
iCesalp-4	16	17	11	699	0	15
iCesalp-5	0	0	0	0	758	0
iCesalp-6	34	56	10	39	0	619

Table 11
Confusion matrix of c263CES.

Fault name	c263CES-1	c263CES-2	c263CES-3	c263CES-4	c263CES-5
c263CES-1	457	217	0	84	0
c263CES-2	137	553	0	68	0
c263CES-3	0	0	758	0	0
c263CES-4	105	74	0	579	0
c263CES-5	0	0	0	0	758

The fault isolation ability of the proposed algorithm is reported in Tables 7–11. In these tables, the confusion matrices are illustrated to demonstrate the fault isolability capabilities of the method. The confusion matrix is a specific table layout that allows visualization of the performance of the fault diagnosis algorithm. Each column of the matrices represents the instances in a predicted class/fault, whilst each row represents the instances in an actual class/fault. The name stems from the fact that it makes easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another). A confusion matrix displays the number of correct and incorrect predictions made by the fault isolation model compared with the actual classification in the test data. The matrix is $N \times N$, where N is the number of faults in this study.

According to Tables 7–11, fault type 1 is often confused with fault type 2. This is because type 1 and type 2 are both additive offset faults with similar magnitudes, and the only difference is that fault type 1 is abrupt and fault type 2 is incipient (slowly developing). Therefore, fault type 2 will be similar to fault type 1 after some time when fault type 2 develops to the desired magnitude.

Table 12 reports the fault isolation performance in model spaces measured by *precision*, *recall* (or sensitivity) and *specificity*. Based on this table, Bagging and SVM outperform other classifiers in terms of three metrics. Adaboost seems to overfit the noise and leads to inferior results.

Some examples of faulty signals and fault diagnosis results obtained using the 'learning in model space' approach are presented in Fig. 10. Based on Tables 7–12 and this figure, it is easy to see that 'additive incipient offset' and 'additive drift' faults are difficult to detect and usually with relatively large delay, since these two types of faults are similar to normal signal in the beginning.

6. Conclusion

In this paper, an effective fault diagnosis framework has been proposed to tackle the challenge in complex engineering systems like CIS, with unstructured and time-varying environments. Combined two-stage successful method has been proposed to detect and correct sensor communication faults using a data pre-filter in the first stage, in order to guarantee a reliable dataset to perform system fault diagnosis of faults of different nature in the second stage. Regarding the sensor data pre-filter, sensor data validation and reconstruction using combined spatial and time series models have been implemented with successful results when communication faults occur in the telecontrol system, which is one of the most common faults affecting this kind of systems. After pre-filtering the sensor data, the 'learning in model space' method is implemented by fitting a series of 'generative' models using a series of signal segments selected with a sliding window. Each signal segment can be represented by one model. To rigorously measure 'distance' between models, the distance in model space has been defined and an example to use deterministic reservoir computing to act as a 'generative' model has been shown. The deterministic reservoir computing can approximate a model with input–output dynamics and it can exploit spatial–temporal correlations that exist in collected data, thus it fits the framework well.

'Learning in model space' is a generic learning framework for cognitive fault diagnosis, and it is applicable to any systems, including complex multiple-input and multiple-output (MIMO) systems. In order to maintain the generality of our approach and to

Table 12
Comparisons of fault isolation ability using supervised learning techniques.

Fault	SVM			CART			Bagging			Adaboost		
	prec.	recall	spec.	prec.	recall	spec.	prec.	recall	spec.	prec.	recall	spec.
iOrioles	93.21	93.26	99.16	87.38	87.35	98.42	93.43	93.42	99.18	62.83	60.10	95.01
c175LOR	72.73	72.92	90.97	65.47	65.67	88.56	70.23	70.52	90.17	60.24	52.67	84.22
iStaClmCervello	90.58	90.47	97.62	82.87	82.66	95.67	90.82	90.79	97.70	48.93	48.87	87.22
iCesalpina1	87.22	87.18	96.79	72.01	71.93	92.98	84.43	84.51	96.13	30.81	36.94	84.23
c263CES	78.00	78.03	92.67	67.28	67.55	89.18	72.31	72.53	90.85	61.71	53.89	84.63

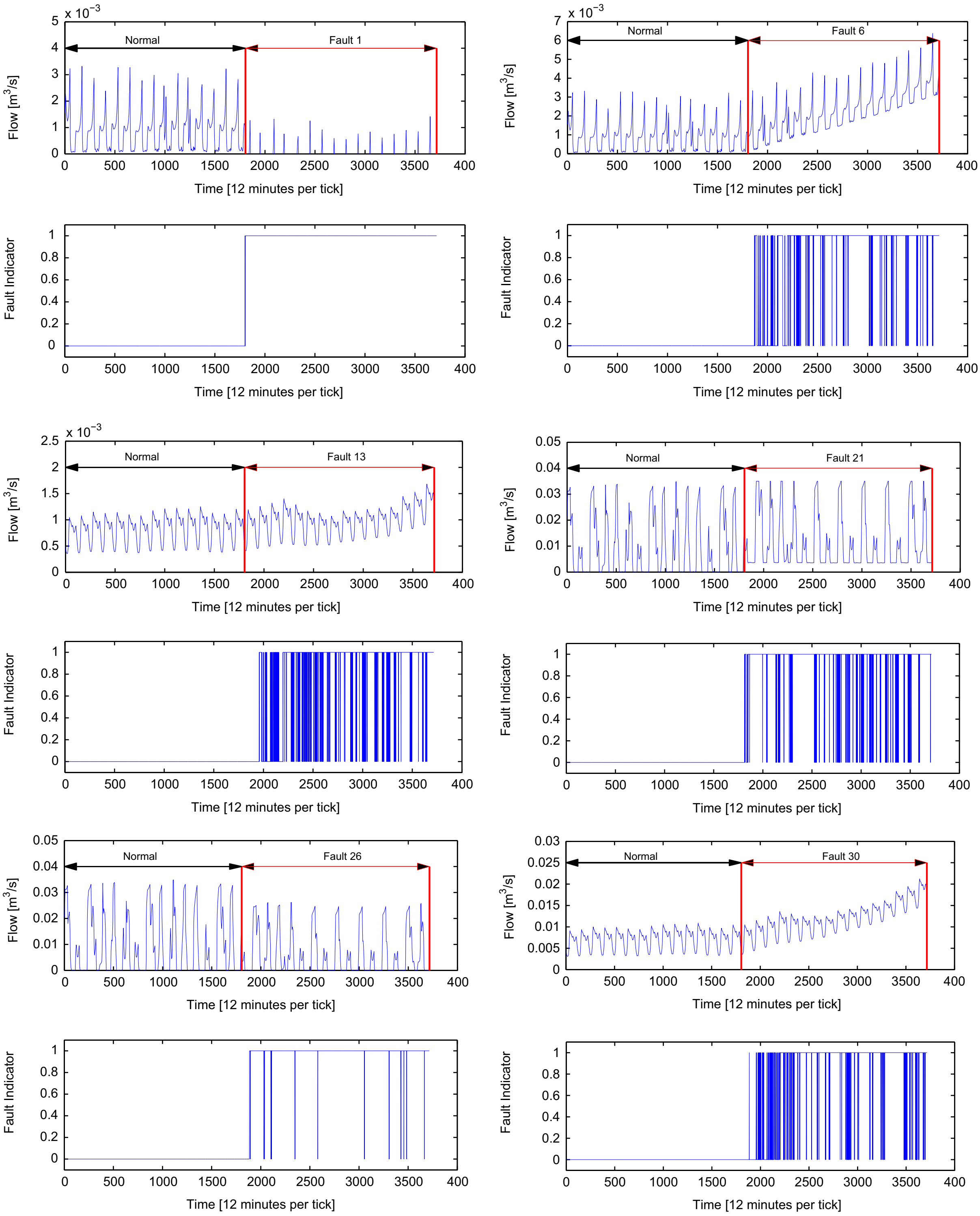


Fig. 10. Examples of faulty signals and fault diagnosis results using the 'learning in model space' approach.

illustrate how this generic approach works, the reservoir model is adopted here. In addition, other fitted models and learning algorithms in the model space can also be used in this framework. Future work includes more in-depth studies of different generative (fitted) models and learning algorithms that best suit the suggested framework.

Acknowledgements

This work is supported by the European Union Seventh Framework Programme, in the framework of the iSense project, under Grant agreement no. INSFO-ICT-270428.

References

Burnell, D., 2003. Auto-validation of district meter data. In: CCWI '03 Advances in Water Supply Management, London.

Carrozza, G., Cotroneo, D., Russo, S., 2008. Software faults diagnosis in complex OTS based safety critical systems. In: Seventh European, Dependable Computing Conference, EDCC 2008, pp. 25–34.

Chen, H., Tiňo, P., Rodan, A., Yao, X., 2014. Learning in the Model Space for Fault Diagnosis. IEEE Transactions on Neural Networks and Learning Systems 25 (1), 124–136.

Čerňanský, M., Tiňo, P., 2008. Predictive modeling with echo state networks. In: Proceedings of the 18th International Conference on Artificial Neural Networks, Part I, pp. 778–787.

Ding, S.X., 2008. Model-based Fault Diagnosis Techniques.

Jaeger, H., 2001. The Echo State Approach to Analysing and Training Recurrent Neural Networks. Technical Report, German National Research Center for Information Technology.

Jaeger, H., Haas, H., 2004. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304 (5667), 78–80.

Jörgensen, H., Rosenórn, S., Madsen, H., Mikkelsen, P., 1998. Quality control of rain data used for urban run-off systems. Water Sci. Technol. 37 (11), 113–120.

Kanakoudis, V.K., Tolikas, D.K., 2001. The role of leaks and breaks in water networks: technical and economical solutions. Water Supply: Res. Technol.—Aqua 50 (5), 301–311.

Kanakoudis, V.K., Tsitsifli, S., 2011. Water pipe network reliability assessment using the DAC method. Desalin. Water Treat. 33 (1–3), 97–106.

Lin, X., Yang, Z., Song, Y., 2009. Short-term stock price prediction based on echo state networks. Expert Syst. Appl. 36 (3), 7313–7317.

Lukoševičius, M., Jaeger, H., 2009. Reservoir computing approaches to recurrent neural network training. Comput. Sci. Rev. 3 (3), 127–149.

Maass, W., Natschläger, T., Markram, H., 2002. Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput. 14 (11), 2531–2560.

Makridakis, S., Wheelwright, S., Hyndman, R., 1998. Forecasting Methods and Applications. John Wiley & Sons

Marinaki, M., Papageorgiou, M., 2005. Optimal Real-time Control of Sewer Networks. Springer

Ozturk, M., Xu, D., Príncipe, J., 2007. Analysis and design of echo state networks. Neural Comput. 19 (1), 111–138.

Pascual, J., Romera, J., Puig, V., Cembrano, G., Creus, R., Minoves, M., 2013. Operational predictive optimal control of barcelona water transport network. Control Eng. Pract. 21 (8), 1020–1034, URL ⟨http://www.sciencedirect.com/science/article/pii/S0967066113000518⟩.

Quevedo, J., Puig, V., Cembrano, G., Blanch, J., Aguilar, J., Saporta, D., Benito, G., Hedo, M., Molina, A., 2010a. Validation and reconstruction of flow meter data in the Barcelona water distribution network. Control Eng. Pract. 18 (6), 640–651, http://dx.doi.org/10.1016/j.conengprac.2010.03.003.

Quevedo, J., Blanch, J., Puig, V., Saludes, J., Espin, S., Roquet, J., 2010b. Methodology of a data validation and reconstructions tool to improve the reliability of the water network supervision. In: International Conference of IWA Water Loss, Sao Paolo, Brazil.

Rodan, A., Tino, P., 2012. Simple Deterministically Constructed Cycle Reservoirs with Regular Jumps. Neural Computation 24 (7), 1822–1852.

Rodan, A., Tiňo, P., 2011. Minimum complexity echo state network. IEEE Trans. Neural Netw. (99), 1–14.

Schütze, M., Campisano, A., Colas, H., Schilling, W., Vanrolleghem, P.A., 2004. Real time control of urban wastewater systems—where do we stand today? J. Hydrol. 299, 335–348, http://dx.doi.org/10.1016/j.jhydrol.2004.08.010.

Skowronski, M., Harris, J., 2007. Automatic speech recognition using a predictive echo state network classifier. Neural Netw. 20 (3), 414–423.

Steil, J.J., 2004. Backpropagation-decorrelation: online recurrent learning with o (n) complexity. In: Proceedings of IEEE International Joint Conference on Neural Networks, vol. 2, pp. 843–848.

Tong, M., Bickett, A., Christiansen, E., Cottrell, G., 2007. Learning grammatical structure with echo state networks. Neural Netw. 20 (3), 424–432.

Topkapi Scada Home Page. URL ⟨http://www.areal.fr/⟩.

Tsang, K., 2003. Sensor data validation using gray models. ISA Trans. 42, 9–17.

Tsitsifli, S., Kanakoudis, V., 2009. Evaluating the performance of an urban water distribution network: tips and tricks for troubleshooting. In: The Seventh International Conference on EWRA Water Resources Conservation and Risk Reduction Under Climatic Instability, Limasol, Cyprus, pp. 519–526.

Tsitsifli, S., Kanakoudis, V.K., Bakouros, I., 2011. Pipe networks risk assessment based on survival analysis. Water Resour. Manag. 25 (14), 3729–3746.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003a. A review of process fault detection and diagnosis: part I: quantitative model-based methods. Comput. Chem. Eng. 27 (3), 293–311, http://dx.doi.org/10.1016/S0098-1354(02)00160-6.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., 2003b. A review of process fault detection and diagnosis: part II: qualitative models and search strategies. Comput. Chem. Eng. 27 (3), 313–326, http://dx.doi.org/10.1016/S0098-1354(02)00161-8.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003c. A review of process fault detection and diagnosis: part III: process history based methods. Comput. Chem. Eng. 27 (3), 327–346, http://dx.doi.org/10.1016/S0098-1354(02)00162-X.

Winters, P.R., 1960. Forecasting sales by exponentially weighted moving averages. Manag. Sci. 6 (52), 324–342.

Xia, Y., Jelfs, B., Van Hulle, M., Príncipe, J., Mandic, D., 2010. An augmented echo state network for nonlinear adaptive filtering of complex noncircular signals. IEEE Trans. Neural Netw. (99), 1–10.

Xue, Y., Yang, L., Haykin, S., 2007. Decoupled echo state networks with lateral inhibition. Neural Netw. 20 (3), 365–376.