Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

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

A hybrid intelligent classifier for anomaly detection

Esteban Jove [a,*], Roberto Casado-Vara [b], José-Luis Casteleiro-Roca [a], Juan Albino Méndez Pérez [c], Zita Vale [d], José Luis Calvo-Rolle [a]

[a] University of A Coruña, Department of Industrial Engineering, Avda. 19 de febrero s/n, 15405 Ferrol, A Coruña, Spain
[b] BISITE Research Group, IoT Digital Innovation Hub, Edificio Multiusos I+D+i, 37007 Salamanca, Spain
[c] Universidad de La Laguna, Department of Computer Science and System Engineering, Avda. Astrofísico Francisco Sánchez, s/n, 38200 La Laguna, Spain
[d] Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development Institute of Engineering. Polytechnic of Porto (ISEP/IPP), Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto, Portugal

ARTICLE INFO

Article history:
Received 20 September 2019
Revised 22 November 2019
Accepted 10 December 2019
Available online xxxx

Keywords:
One-class
Outlier detection
SVDD
Autoencoder
PCA
APE

ABSTRACT

The present research is focused on the use of intelligent techniques to perform anomaly detection. This task represents a special concern in complex systems that operate in different regimes. Then, this work proposes a hybrid intelligent classifier based on one-class techniques, capable of detecting anomalies of the different operating ranges. The proposal is implemented over an industrial plant designed to control the water level in a tank, taking into consideration three different operating points. The hybrid classifier is validated by using real anomalies, obtaining successful results.

1. Introduction

Since the beginning of the industrial activity, the objectives and the production in general terms have changed significantly. For instance, initially the most important concern was to increase the production to the fullest, specially, trying to industrialize processes that were manual up to this time. But nevertheless, over time, many different goals have been included, obviously maintaining the production. Examples of these factors are the optimization, sustainability, all types of efficiency, and so on. At the same time, they are motivated by the need to take care of the environment, new regulations, trends and, of course, the need to be competitive in a globalized world [1].

Being more specifically, for achieving these new factors, it is very important to minimize the energy consumption, minimize the impact over the environment, reduce non-conforming products, increase the final quality of the product or the service, among others. In summary, with the minimum environment impact and the minimum expenditure, the same production is sought [2]. All the contributions on this way are welcomed.

A specific reason that causes a lot and huge problems during the operation of processes are the anomalies and faults. This fact triggers the consequent failures over the process or the service quality (total or partial). In spite of the fact that in an early stage, the consequences do not seem important, after a certain time, dire effects may appear. Frequently, one of the possible reasons of the above described problem is the sensor failures under a global point of view: reading failures, signal perturbations, physical damages, etc. At first, it could lead to smaller problems, but the carried consequences could be fatal or at least momentous [3].

In [4], a classification of sensor errors is presented: total failure and measure deviation. The detection and the possible solution of the first case could be easy to solve. However, the detection process in the second case could entail difficulties. This case, that needs special attention, could be a very problematic concern for the process under a control point of view, and of course, for accomplishing the fault detection (FD) and diagnosis, supervising and optimizing. Both problems mentioned at the classification are very important.

Due to the above mentioned, this fact plays a key role in a very huge amount of cases. Then, it was defined Sensor Fault Detection (SFD) as a malfunction of a sensor that involves a deviation of right

* Corresponding author.
E-mail addresses: esteban.jove@udc.es (E. Jove), rober@usal.es (R. Casado-Vara), jose.luis.casteleiro@udc.es (J.-L. Casteleiro-Roca), jamendez@ull.edu.es (J.A. Méndez Pérez), zav@isep.ipp.pt (Z. Vale), jlcalvo@udc.es (J.L. Calvo-Rolle).

https://doi.org/10.1016/j.neucom.2019.12.138
0925-2312/

Please cite this article as: E. Jove, R. Casado-Vara, José-Luis Casteleiro-Roca et al., A hybrid intelligent classifier for anomaly detection, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.12.138

reading. Some are the possible reasons that can give rise to this problem: reading device problems, perturbations, wrong location of the sensor, etc. Although the possible solution for the problem will be feasible and even easy, first, it is important to know that the sensor or transducer reading is not right. There are some proposals to accomplish the sensor fault detection [5]. One of them is physical redundancy, for both accomplishing diagnose and faults discovering. This kind of method for achieving SFD is implemented over applications that are very critical. But the main problem is that it does not have an easy implementation for reasons like difficult application, high cost and viability [4]. Then, lately it is in disuse due to these reasons.

Nowadays people live in the era of digitalization with trends like Industry 4.0, Internet of Things, Cyber-Physical systems, and so on. It implies that a lot of variables must be read, and monitored continually. These readings are carried out by sensors, and it is required to ensure the right sensors performance. This task is clear, but it is not easy, especially in industrial environments [6]. Some are the problems at these places: process perturbations, actuators faults, sensors faults and noise at sensors and transducers [7]. With the main purpose of avoiding the possible impact of the mentioned problems, it is desirable to make use of methods for accomplishing fault detection [8].

It could be a very interesting choice to make a SFD based on intelligent systems, when systems are not crucial, or even when it is not the case, use these techniques as a complementary tool to another one. With that kind of methods it could be possible to detect sensor failures, and even to predict possible deviations [9]. Also, there are methods for achieving wrong data recovering when the bad reading is detected [7,10].

The present research work deals with the anomaly detection of an industrial plant used to control the water level in a tank. This kind of installations represents a challenge, since the appearance of any anomaly may be produced by many different sources [6,11]. Then, the anomaly detection must take into consideration different plant parameters and variables. In this case, the data used to perform the anomaly detection is collected from the correct operation of the plant.

In many different works [12], the use of global one-class classifiers is proposed to achieve the anomaly detection over a wide variety of systems. However, the plant under study presents an additional complexity, that is the possibility of working in different operating points. Then, a hybrid intelligent classifier based on one-class techniques is proposed by training different local classifiers for each point. The best configuration is selected after testing seven one-class techniques: Gaussian Model [13], Parzen Density Estimator [13], Principal Component Analysis [14], k-Nearest Neighbor [15], Approximate Polytope Ensemble [12], Autoencoder Artificial Neural Network [16] and Support Vector Data Description [13]. The proposed approach was tested and validated using real created anomalies, giving very successful results.

This paper is structured according to the next sections. After the present introduction, the case study describes the plant operation and the initial dataset. Then, the one-class classification techniques used to implement the hybrid classifier are described. The next section proposes the hybrid intelligent classifier approach to work with different operating points. Finally, the results section and the conclusions and future works section are presented.

2. Case study

This section describes the industrial plant, where the hybrid intelligent classifier is implemented, and the features of the dataset used on this work.

2.1. Industrial system description

The hybrid intelligent classifier based on one-class techniques proposed in this work is trained, tested and validated in the industrial plant (see Fig. 1), whose scheme is shown in Fig. 2. The main aim of this plant is to control the water level in a tank (1), that is pumped from a storage tank (2) using a three phase pump (3). The objective tank has an electric built-in valve (4), to send the water back to the initial tank. This element plays a significant role to validate this work, since the anomalies are created by opening the valve.

The control process is implemented using *Matlab* software, installed in a computer with a *Intel Core i7-8550U 1.80 GHz* processor and 8 GB of RAM memory. The communication between the computer and the plant is carried out by a National Instruments data acquisition card (model USB-6008 12-bit 10 KS/s Multifunction I/O). This element is in charge of two main tasks:

- Sending to the computer the current value measured the ultrasonic *Banner* sensor, model S18UUA.
- Sending to the variable frequency drive (VFD) the control signal from the computer.

Given the strong nonlinearity of the industrial plant, a virtual adaptive PID controller is implemented to manage the water level. The first stage for tunning the PID parameters consists on identifying the plant coefficients using the Recursive Least Squares (RLS) algorithm, following the Eq. (1) [17]:

$$H_{plant}(z^{-1}) = \frac{q_0^{-k}}{1 - p_1 z^{-1} - p_2^{-2}} \tag{1}$$

where:

- q_0 – Open loop gain
- k – System delay
- p_1 – First order coefficient
- p_2 – Second order coefficient

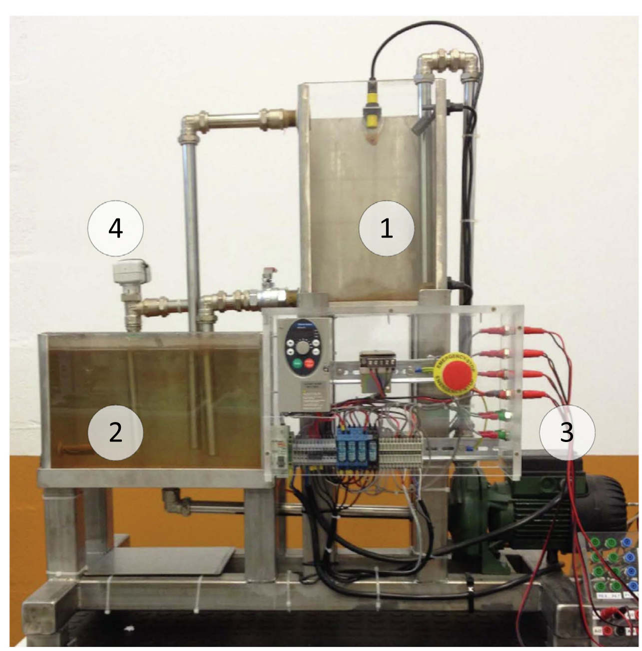


Fig. 1. Picture of the real plant.

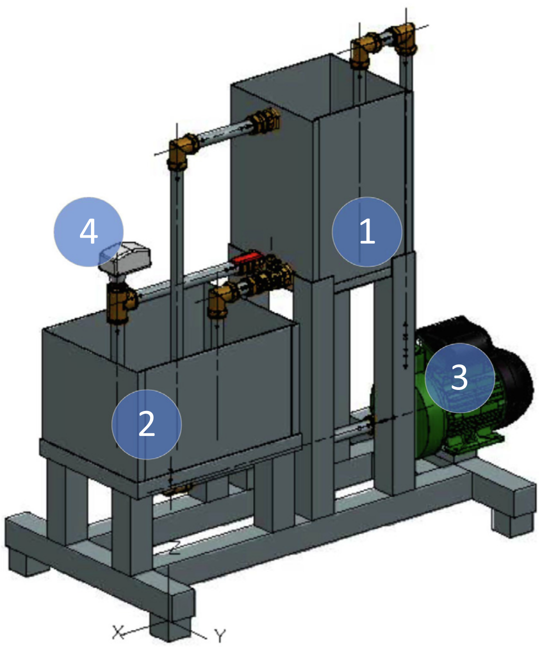


Fig. 2. Control level system.

Then, from the transfer function identified during process operation, an adaptive PID is self-tuned following the Eq. 2 [18].

$$PID_{controller}(z^{-1}) = \frac{a_0 + a_1^{-1} + a_2^{-2}}{1 - z^{-1}} \tag{2}$$

where:

- $a_0 = \frac{1}{q_0 T_c^2 (2K_c + 1)}$
- $a_1 = -p_1 a_0$
- $a_2 = -p_2 a_0$
- K_c – Critical gain
- T_c – Critical period

The scheme shown in Fig. 3 represents all the elements involved in the control loop described above.

2.2. Dataset

The initial dataset is comprised by five different variables registered during the system operation, whose control loop is designed to ensure a constant sampled rate of 2 Hz. These monitored variables are the following:

- Control signal calculated by the control loop that drives the pump. This signal can variate from 0 to 100 to modify the pump speed from 0 to 50 Hz.
- Error signal from the difference between the set point and the process value.
- The PID parameters $a_0,\ a_1,\ a_2$, described according to Eq. (2).

As the main idea of this work is to propose a hybrid intelligent classifier, the following three different operating points are considered as correct:

- Set point 40% and electric valve closed. 5400 samples.
- Set point 50% and electric valve closed. 5400 samples.
- Set point 60% and electric valve closed. 5400 samples.

Then, to check the performance of each local classifier, the data from real anomalous situation is registered. This anomalies represent three cases:

- Set point 40% and electric valve open. 5400 samples.
- Set point 50% and electric valve open. 5400 samples.
- Set point 60% and electric valve open. 5400 samples.

3. Methods

This section describes the different methods considered on the present research work to perform the hybrid classifier. They are contemplated due to their very positive performance in a wide variety of one-class classification tasks.

3.1. Gaussian model

One of the most simple ways to achieve a one-class classifier is based on the density estimation of the data using a Gaussian or normal distribution [13]. Though it is quite simple, its use on one-class classification problems has offered interesting results [13].

The training data, which is collected from correct plant operation, is used to implement a Gaussian model of the target set, following Eq. (3) [19].

$$p_G(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{(d/2)} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})} \tag{3}$$

where:

- $\boldsymbol{x}$ is the test point.
- $\boldsymbol{\mu}$ is the mean vector of the target set.
- $\boldsymbol{\Sigma}$ is the covariance of the target set.
- d is the dimension of the dataset.

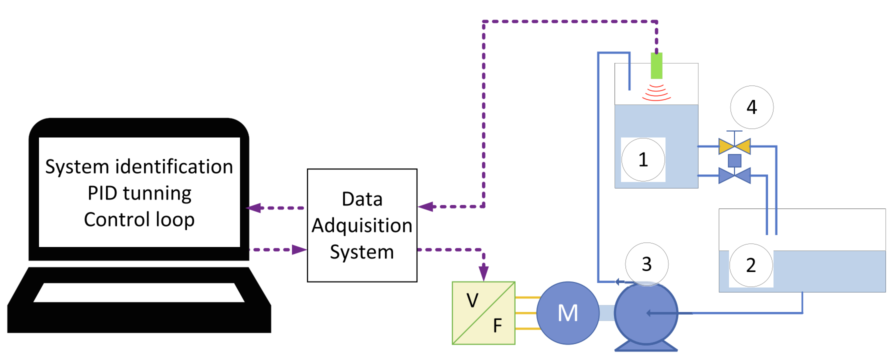


Fig. 3. Scheme of the industrial plant setup.

Then, a test point is classified depending on the value of the density function. When it exceeds a set threshold, the anomaly is detected. This method does not present a high computational cost, being the covariance matrix inversion the most significant process [13] and gives successful results when the data follows a normal distribution.

3.2. Parzen Density Estimator

A similar approach to obtain a one-class classifier can be considered using the non-parametric Parzen Density Estimator (PDE) [20], whose performance has proven to be successful [21].

Unlike the Gaussian, this method does not make any specific assumptions about the data shape. In this case, the density distribution is computed as shown in Eq. 4 [19].

$$p_{PDE}(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{V}\phi\left(\frac{\boldsymbol{x}-\boldsymbol{x_i}}{W_p}\right) \tag{4}$$

Where:

- $\boldsymbol{x}$ is the test point.
- N is the size of the training set.
- V is the volume of the region in a form of the hypercube with edge W_p.
- ϕ is the Parzen window.
- $\boldsymbol{x_i}$ is the i^{th} point of the dataset.

As with Gaussian Model, it performs properly when the training set is representative enough of the target set. After the training process, this method can estimate the novelty of each test sample, by establishing a threshold [22].

3.3. Principal Component Analysis

The Principal Component Analysis (PCA) has shown satisfactory results in both dimensional reduction and one-class tasks [23,14,12]. The use of PCA for one-class classification aims to map the training data into a linear subspace defined by the eigenvectors of the covariance matrix.

The principal components $\boldsymbol{x}'_i$ are the projection of the original variable over the eigenvectors $\boldsymbol{e_i} = (e_{i1}, \ldots, e_{id})$ of the covariance matrix $\boldsymbol{\Sigma}$, ordered according to the decreasing eigenvalues λ_i. Then, the first component of a test point $\boldsymbol{x}$ is computed according to Eq. (5) [19,24].

$$\boldsymbol{x}'_1 = \boldsymbol{e}_1^T(\boldsymbol{\mu} - \mathbf{x}) \tag{5}$$

Hence, the number of eigenvectors, known as components, can be at most the same as the number of variables. The criteria followed to decide if a test data belongs to the target class is based on the reconstruction error. This is computed as the difference between the original point and the point projected in the subspace. This technique offers good results when the subspace is clearly linear [13].

3.4. k-Nearest Neighbor

The k-Nearest Neighbor (kNN) method uses the distances between objects instead of probability density functions, such as Gaussian or Parzen. In particular, the outlier nature of a specific point x, is determined by the local density of the hypersphere containing its k^{th} nearest neighbors [25].

Then, a test point $\boldsymbol{x}$ is considered anomalous when its distance to the k^{th} nearest training data neighbor $kNN^{tr}(\boldsymbol{x})$ is higher than the local distance from the k^{th} neighbor to its k^{th} neighbor [13].

$$d(\boldsymbol{x}) = \frac{\left\|\boldsymbol{x} - kNN^{tr}(\boldsymbol{x})\right\|}{\left\|kNN^{tr}(\boldsymbol{x}) - kNN^{tr}(kNN^{tr}(\boldsymbol{x}))\right\|} \tag{6}$$

Hence, the classifier identifies an outlier when its local density d is lower than the local density of it first nearest neighbor belonging to the training set [13].

The value of k plays a significant role in the classifier performance, and it depends on the dataset structure shape [25].

3.5. Approximate Polytope Ensemble

The Approximate Polytope Ensemble (APE) has been used for classification tasks in many different fields [26] and it was tested successfully over many UCI repositories [27].

The main idea of this technique is to obtain the boundaries of the training set $\boldsymbol{X}$ by using its convex hull $CH(\boldsymbol{X})$, according to Eq. (7) [26].

$$CH(X) = \left\{\sum_{i=1}^{N}\alpha_i x_i \| \sum_{i=1}^{N}\alpha_i = 1, 0 \leqslant \alpha_i \leqslant 1\right\} \tag{7}$$

However, this calculation has the problem of a significant computational cost when dealing high dimensional datasets. Then, it is possible to model the convex hull of the training data by using p random $2D$ projections.

After the training process, a test data is classified as outlier if it is out of at least one of the projections. It is also possible to expand or shrink the projections of the convex hull to evaluate the best classifier configuration.

3.6. Autoencoder Artificial Neural Network

The Autoencoder Artificial Neural Network has shown interesting results in different works [16,28,29]. This technique uses the well known Multilayer Perceptron (MLP) Artificial Neural Network (ANN), whose structure is divided into three parts or layers: input layer, hidden layer and output layer. This ANN is configured to reconstruct at the output the input pattern by means of a nonlinear intermediate reduction in the hidden layer [30].

Therefore, the hidden layer output $\boldsymbol{v}$ is computed as shown in Eq. (8), and the output $\boldsymbol{x}'$ is calculated following the expression in (9).

$$\boldsymbol{v} = f_1(\boldsymbol{W_1 x} + \boldsymbol{b_1}) \tag{8}$$

Where:

- $\boldsymbol{W_1}$ – Weight matrix between input and hidden layer.
- $\boldsymbol{b_1}$ – Bias vector.

$$\boldsymbol{x}' = f_2(\boldsymbol{W_{2}}_{v} + \boldsymbol{b_2}) \tag{9}$$

where:

- $\boldsymbol{W_2}$ – Weight matrix between hidden and output layer.
- $\boldsymbol{b_2}$ – Bias vector.

The criteria to detect anomalies is the following: since the MLP is trained with target objects to reconstruct the input, the test instances that are dissimilar to the target class may lead to high reconstruction error.

3.7. Support Vector Data Description

The Support Vector Machine (SVM) is a supervised learning used for classification and regression tasks [25]. Its main goal is to map the training set into a hyperspace and then, implement a hyperplane to maximize the distance between classes [31].

From this basic principles, the Support Vector Data Description (SVDD) method is designed to look for a closed boundary (hypersphere) instead of a hyperplane [32]. During the training phase, all the training points are closed by the high dimensional sphere. This minimum hypersphere with a center $\boldsymbol{a}$ and radius $\boldsymbol{R}$ is achieved minimizing the Eq. (10), subjected to the constraints in Eq. (11) [33].

$$F(\boldsymbol{R}, \boldsymbol{a}, \xi_i) = \boldsymbol{R}^2 + C\sum_i \xi_i \tag{10}$$

$$\|\boldsymbol{x_i} - \boldsymbol{a}\|^2 \leqslant \boldsymbol{R}^2 + \xi_i \qquad \xi_i \leqslant \boldsymbol{R}^2 \tag{11}$$

Where:

- ξ_i is the slack variable.
- C is a tradeoff parameter between volume and errors in the training set.

Then, when a new test data is assessed, it is labeled as anomaly if it is out of the hypersphere.

4. Hybrid intelligent classifier proposal

The present section is divided into two parts. First, the approach to achieve the one-class classifier is described. Then, the experiments setup proposed to obtain the hybrid topology is explained.

4.1. One-class classifier approach

The main goal of this approach is to achieve a classifier capable of detecting anomalies in the three different operating points of an industrial plant.

The proposal is illustrated in Fig. 4, where the process followed to detect an outlier is represented. First, the classifier selector block routes the system variables to the appropriate classifier depending on the current plant operating point. For instance, when the system is working in the first operating point, the system variables are routed to Classifier 1. Then, the selected classifier determines if the data belongs to the target class and the anomaly detection block reports it. Each local classifier is implemented with the technique that shown the best performance during the test phase. In the scheme of Fig. 4 is presented an example where the system is working in the third operating point. Hence, the classifier three is in charge to detect the anomalous situation.

To evaluate the proposed approach and obtain the three local one-class classifiers, the techniques described in Section 3 are applied to the dataset. The operating points considered as target class for each classifier are the following:

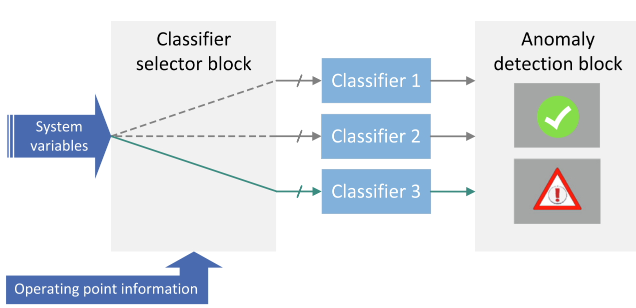


Fig. 4. Hybrid classifier approach.

- Level 40% and valve closed.
- Level 50% and valve closed.
- Level 60% and valve closed.

On the other hand, as mentioned in Section 2, the data registered with the same tank levels but with the valve open are labeled as anomalies.

Considering three different tank levels x, y and z, an example of the data distribution used to train and test each local classifier is presented in Fig. 5. The data is distributed and processed as follows:

1. The data corresponding to a $x\,\%$ level with the valve closed represents the target set. The 90% of this set is randomly selected to train the classifier.
2. Once the classifier is trained, it is tested with different data groups:
   - The 10% left from the target class.
   - The data from $x\,\%$ level and the valve open.
   - The data from $y\,\%$ and $z\,\%$ levels with valve open and closed.
3. This process is repeated 10 times following a 10 $k-fold$ cross-validation.

This process is the same for each one-class technique.

4.2. Experiments setup

As the main idea of this work is to implement the best one-class classifier for each plant operation, the performance of each technique is assessed. The applicability of these techniques is evaluated through a detailed comparative analysis of the hyperparameter influence over the results. A brief explanation of the tested hyperparameters is presented below:

- Gaussian.
  - O_{fr}: represents the percentage of outliers in the training set. This parameter sets the threshold to determine the criteria to detect anomalies. This parameter is also used in Parzen, kNN, PCA, Autoencoder and SVDD classifiers.
  - R_p: represents a factor for the covariance matrix.
- Parzen Density Estimator.
  - W_p: is the kernel width used to obtain the density function.
- Principal Component Analysis.

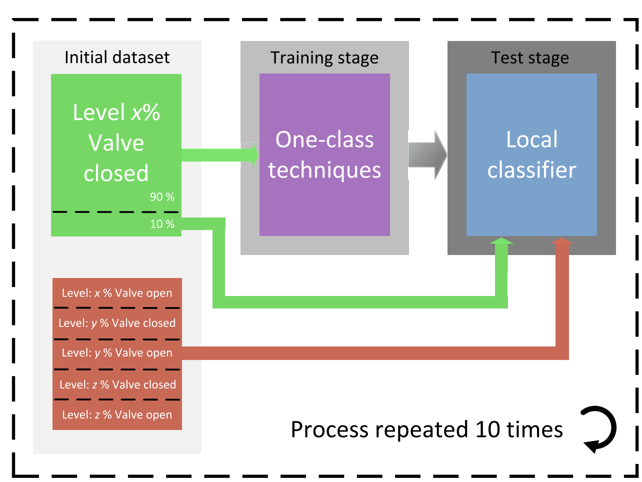


Fig. 5. Train and test process to obtain the classifier for the operating point $x\,\%$ and valve close.

- N_{comp}: represents the number of principal components used to project the new point into a lower dimensional space.

• k-Nearest Neighbor.
  - k: is the number of neighbors taken into consideration to compute the distance of a test object to the target set.

• Approximate Polytope Ensemble.
  - N_{pr}: number of 2D projections to determine the approximate convex hull of the target set.
  - θ: expansion parameter to modify the convex hull extension.

• Autoencoder Artificial Neural Network.
  - N_{hl}: number of neurons in the hidden layer.

• Support Vector Data Description.
  - σ: width parameter of the radio basis function.

Hence, all hyperparameters described are checked to achieve the best classifier. The values tested are shown in Table 1.

In addition to the different techniques setups, regardless of the technique applied, all classifiers were trained with three prior dataset configurations: raw data, normalization 0 to 1 and normalization using z-score [34]. The z-score z_i of a certain point x_i represents how many standard deviation is deviated from the mean, according to Eq. (12).

$$z_i = \frac{x_i - \mu}{\sigma} \tag{12}$$

All algorithms were implemented using Matlab software. The Gaussian Model, PDE, PCA, k-NN and SVDD classifiers were obtained using the dd_{tools} toolbox [15]. In the case of Autoencoder classifiers, they were achieved with the '*trainAutoencoder*' function [35].

5. Results

The Area Under the Receiving Operating Curve (AUC) parameter, in%, is used to evaluate the performance of each technique. This parameter, that represents a relationship between true positives and false positives, is a good indicator of the performance in one-class classification problems [36].

Table 1
Techniques setup.

Technique	Parameters	Values tested
Gaussian Model		
	Outliers % in the target class (O_{fr})	0:2,5:15
	Regularization parameter (R_p)	0:0,001:0,01
PDE		
	Outliers % in the target class (O_{fr})	0:2,5:15
	Width parameter (W_p)	0:0,001:0,01
PCA		
	Outliers % in the target class (O_{fr})	0:2,5:15
	Number of components (N_{comp})	1:1:4
k-NN		
	Outliers % in the target class (O_{fr})	0:2,5:15
	Number of neighbors (k)	1:1:10
APE		
	Number of projections (N_{pr})	50, 100, 500, 1000
	Expansion parameter (θ)	0,5:0,1:2
Autoencoder ANN		
	Outliers % in the target class (O_{fr})	0:2,5:15
	Neurons in the hidden layer (N_{hl})	1:1:4
SVDD		
	Outliers % in the target class (O_{fr})	0:2,5:15
	Width parameter of RBF kernel (σ)	1:1:10

As mentioned in Section 4, a 10-fold cross-validation was performed. The mean AUC and the standard deviation (STDD) calculated from the 10 iterations are used as a measure of the performance and repeatability of each classifier. In addition to the classifier performance, the time needed to train each one (t_{tr}) is also a key parameter to take into consideration.

Tables 2–4 represent the best classifiers results obtained by each technique, in terms of AUC, considering the three operating points (40%, 50% and 60% levels, respectively) with the valve closed. Furthermore, the training configuration needed to achieve these results is also presented.

From all the tested classifiers, the best configuration for each operating point is chosen following an AUC criteria. Then, the one-class hybrid topology is implemented according to Table 5.

5.1. Comparative results

This section describes the main findings of each technique. First, the average performance of all methods, in terms of AUC and training times, is show in Figs. 6 and 7. It can be noted that APE has the best general performance regardless the chosen configuration. The differences between training times are significant, being Gaussian the fastest and SVDD the slowest.

In addition to this general performance overview, a more detailed analysis of the hyperparameters influence for each technique was carried out.

Gaussian Model. In general terms, it can be noticed that the Gaussian Model presents a very low training time (t_{tr}) in the three local classifiers. This fact is reasonable, because it is the simplest technique. However, this technique does not lead to specially high values of AUC. The best regularization parameter R_p is always 0, and the outlier fraction and data conditioning is not the same for each operating point.

PDE. The PDE presents a very interesting performance in the three classifiers, with the best AUC values in two of them. The training time is always below 0,5 s, which is the system sample time. In the three cases, the best results were achieved with the data normalized.

PCA. Regarding the PCA, it is interesting to remark that the optimum number of components was always 4. However, the AUC obtained is not among the bests in any case. This method presents the advantage of a very low training time.

k-NN. The k-NN shown the best AUC in the second local classifier and the second best AUC in the third local classifier. In the three cases, the optimum number of cluster was 1, and the data was not conditioned. A slight disadvantage of this technique is the training time, that is about a minute.

APE. This technique gives very irregular AUC results. The first and second classifiers presents values of at least $99,50\%$, but the third classifier has the lowest value. It is important to remark the influence of the number of projections in the training time. The first classifier ($N_{pr} = 1000$) has a training time 25 times higher than the third one ($N_{pr} = 50$). It is also interesting to emphasize that the third classifier expands its projections ($\theta = 1,8$) and the rest reduce them ($\theta = 0,8$).

Autoencoder ANN. The Autoencoder technique presents relatively low values of AUC in the three local classifiers. Autoencoder ANN. The best results are obtained with raw data and 2, 4 and 4 neurons in the hidden layer, respectively. From all the experiments, it is concluded that number of neurons plays a significant role in the training times. In this case, an increase of the hidden layer neurons leads to an increase in the training time. For this reason, the first autoencoder classifier has lower training time (2 neurons) than the second and third classifiers (4 neurons).

Table 2
Best classifier obtained for each technique. Tank at 40% and valve closed.

Technique	AUC (%)	STDD (%)	$t_{tr}(s)$	Configuration
Gaussian Model	93,74	0,81	0,01	$O_{fr} = 0,05$ $R_p = 0$ Norm
PDE	99,69	0,03	0,41	$O_{fr} = 0$ $W_p = 0,005$ Norm
PCA	93,66	0,27	0,02	$O_{fr} = 0,025$ $N_{comp} = 4$ Norm
k-NN	98,84	0,03	1,03	$O_{fr} = 0$ $k = 1$ Raw
APE	99,50	0,03	1,01	$\theta = 0,8$ $N_{pr} = 1000$ Raw
Autoencoder ANN	96,61	2,77	0,90	$O_{fr} = 0,025$ $N_{hl} = 2$ Raw
SVDD	99,39	0,04	193,67	$O_{fr} = 0$ $\sigma = 1$ Zscore

Table 3
Best classifier obtained for each technique. Tank at 50% and valve closed.

Technique	AUC (%)	STDD (%)	$t_{tr}(s)$	Configuration
Gaussian Model	87,51	0,31	0,01	$O_{fr} = 0,075$ $R_p = 0$ Zscore
PDE	99,96	0,02	0,28	$O_{fr} = 0,025$ $W_p = 0,005$ Norm
PCA	91,31	0,20	0,05	$O_{fr} = 0,025$ $N_{comp} = 4$ Norm
k-NN	99,99	0,02	0,99	$O_{fr} = 0$ $k = 1$ Raw
APE	99,99	0,03	0,11	$\theta = 0,8$ $N_{pr} = 100$ Raw
Autoencoder ANN	97,84	1,66	1,26	$O_{fr} = 0,025$ $N_{hl} = 4$ Raw
SVDD	99,93	0,05	175,47	$O_{fr} = 0$ $\sigma = 3$ Raw

SVDD. This technique has the main drawback of its high computational cost, with a training time of at least 175 s. This value is significantly higher than the values of the rest of techniques. The AUC values are in the three cases really interesting, specially in the first and second classifiers.

6. Conclusions and future works

The present works deals with the anomaly detection problem in an industrial plant. The basic idea is to divide the different operating points of the system and then, train a one-class classifier for each one. Different one-class techniques were trained, tested and validated over the system, and the best one is chosen to implement the hybrid classifier. These results can lead to many competitive advantages in terms of cost reduction in predictive and corrective maintenance tasks, low rejection rates, high quality production or energy savings, among others.

From this work, we can conclude that it is possible to detect anomalies successfully in an industrial plant that operates in different points by using the hybrid intelligent methodology proposed. This approach can be presented as a very useful tool in a wide variety of applications, such as medicine, surveillance systems or fraud detection in credit card usage, to detect deviations from the correct operation.

As future works, due to the fact that the three chosen classifiers need a training time lower than 0,5 s, that is the sample time, it could be possible to retrain the system on line. This future proposal could be interesting to be applied with the aim of updating the system behavior during its use.

Table 4
Best classifier obtained for each technique. Tank at 60% and valve closed.

Technique	AUC (%)	STDD (%)	$t_{tr}(s)$	Configuration
Gaussian Model	92,78	0,39	0,01	$O_{fr} = 0,1$ $R_p = 0$ Raw
PDE	97,87	0,27	0,40	$O_{fr} = 0$ $W_p = 0,004$ Norm
PCA	94,54	0,44	0,03	$O_{fr} = 0,075$ $N_{comp} = 4$ Zscore
k-NN	97,71	0,27	1,11	$O_{fr} = 0,025$ $k = 1$ Raw
APE	85,63	0,06	0,04	$\theta = 1,8$ $N_{pr} = 50$ Raw
Autoencoder ANN	91,46	0,60	1,22	$O_{fr} = 0,125$ $N_{hl} = 4$ Raw
SVDD	91,49	1,83	209,98	$O_{fr} = 0,05$ $\sigma = 1$ Raw

Table 5
Final hybrid one-class classifier configuration.

Tank level	Technique	AUC (%)	STDD (%)	$t_{tr}(s)$	Configuration
40% level	PDE	99,69	0,03	0,42	$O_{fr} = 0$ $W_p = 0,005$ Norm
50% level	APE	99,99	0,03	0,11	$\theta = 0,8$ $N_{pr} = 100$ Raw
60% level	PDE	97,87	0,27	0,40	$O_{fr} = 0$ $W_p = 0,004$ Norm

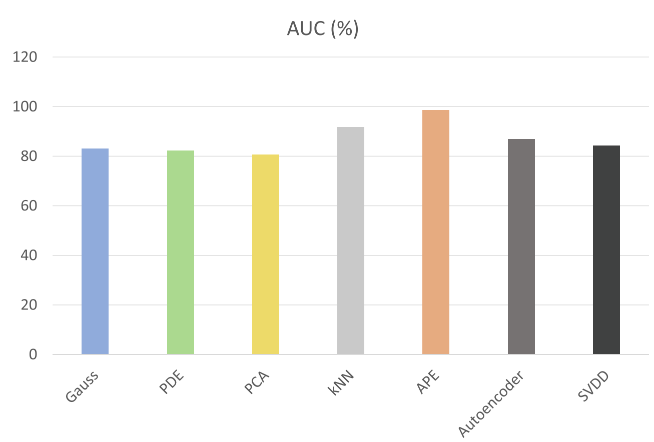


Fig. 6. Average AUC for each technique (%).

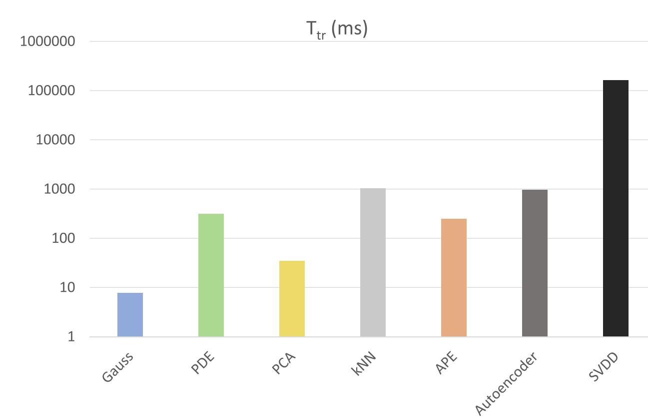


Fig. 7. Average training times for each technique (ms).

Another challenge to face in future works would consist of implementing one classifier for each sensor and actuator. Then, besides the anomaly detection, it could be possible to identify the source of the problem.

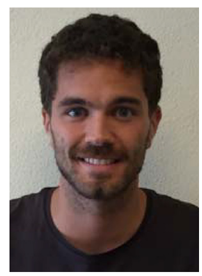
In addition, the use of other well known one-class techniques could help the proposed approach to improve its results.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I. Parmee, P. Hajela, Optimization in Industry, Springer Science & Business Media, 2012.
[2] T. Nicholson, Optimization in Industry: vol. 1, Optimization Techniques, Routledge, 2017.
[3] A.B. Sharma, L. Golubchik, R. Govindan, Sensor faults: Detection methods and prevalence in real-world datasets, ACM Transactions on Sensor Networks (TOSN) 6 (2010) 23.
[4] S. Wang, J. Cui, Sensor-fault detection, diagnosis and estimation for centrifugal chiller systems using principal-component analysis method, Applied Energy 82 (2005) 197–213.
[5] A. Oliva, Sensor fault detection and analytical redundancy satellite launcher flight control system, SBA Controle & Automação 9 (1998) 156–164.
[6] E. Jove, J.-L. Casteleiro-Roca, H. Quintián, J.A. Méndez-Pérez, J.L. Calvo-Rolle, A fault detection system based on unsupervised techniques for industrial control loops, Expert Systems (2019), e12395.
[7] J. Carvajal-Godinez, J. Guo, E. Gill, Agent-based algorithm for fault detection and recovery of gyroscope's drift in small satellite missions, Acta Astronautica 139 (2017) 181–188.
[8] E. Jove, J.M. Gonzalez-Cava, J.-L. Casteleiro-Roca, J.-A. Méndez-Pérez, J. Antonio Reboso-Morales, F. Javier Pérez-Castelo, F. Javier de Cos Juez, J. Luis Calvo-Rolle, Modelling the hypnotic patient response in general anaesthesia using intelligent models, Logic Journal of the IGPL 27 (2018) 189–201.
[9] G. González, C.D. Angelo, D. Forchetti, D. Aligia, Detection and isolation of faults on the rotor side converter of doubly fed induction generators, Revista Iberoamericana de Automática e Informática Industrial 15 (2018) 297–308.
[10] J.-L. Casteleiro-Roca, A. Javier Barragan, F. Segura, J. Luis Calvo-Rolle, J. Manuel Andujar, Intelligent hybrid system for the prediction of the voltage-current characteristic curve of a hydrogen-based fuel cell, Revista Iberoamericana de Automática e Informática Industrial 16 (2019) 492–501.
[11] J.L. Calvo Rolle, R. Ferreiro Garcia, A. Alonso Alvarez, H. Alaiz Moreton, Knowledge-based system for vibration analysis in predictive maintenance, DYNA 83 (2008) 484–492.
[12] E. Jove, J.-L. Casteleiro-Roca, H. Quintián, J.A. Méndez-Pérez, J.L. Calvo-Rolle, Anomaly detection based on intelligent techniques over a bicomponent production plant used on wind generator blades manufacturing, Revista Iberoamericana de Automática e Informática Industrial (2019).
[13] D.M.J. Tax, One-class classification: concept-learning in the absence of counter-examples [ph. d. thesis], Delft University of Technology, 2001.
[14] J. Wu, X. Zhang, A pca classifier and its application in vehicle detection, in: IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), vol. 1, IEEE, pp. 600–604.
[15] D. Tax, Ddtools, the data description toolbox for matlab, 2018, Version 2.1.3.
[16] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, ACM, p. 4.
[17] J.L. Calvo-Rolle, O. Fontenla-Romero, B. Pérez-Sánchez, B. Guijarro-Berdiñas, Adaptive inverse control using an online learning algorithm for neural networks, Informatica 25 (2014) 401–414.
[18] E. Jove, H. Alaiz-Moretón, I. García-Rodríguez, C. Benavides-Cuellar, J.L. Casteleiro-Roca, J.L. Calvo-Rolle, Pid-its: an intelligent tutoring system for pid tuning learning process, in: International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding, Springer, 2017, pp. 726–735.
[19] O. Mazhelis, One-class classifiers: a review and analysis of suitability in the context of mobile-masquerader detection, South African Computer Journal 2006 (2006) 29–48.
[20] E. Parzen, On estimation of a probability density function and mode, The Annals of Mathematical Statistics 33 (1962) 1065–1076.
[21] P. Casale, O. Pujol, P. Radeva, Approximate polytope ensemble for one-class classification, Pattern Recognition 47 (2014) 854–864.
[22] G. Cohen, H. Sax, A. Geissbuhler, et al., Novelty detection using one-class parzen density estimator. An application to surveillance of nosocomial infections., in: MIE, pp. 21–26..
[23] L.H. Chiang, E.L. Russell, R.D. Braatz, Fault Detection and Diagnosis in Industrial Systems, Springer Science & Business Media, 2000.
[24] R. Vega Vega, H. Quintián, J.L. Calvo-Rolle, Á. Herrero, E. Corchado, Gaining deep knowledge of android malware families through dimensionality reduction techniques, Logic Journal of the IGPL 27 (2018) 160–176.
[25] T. Sukchotrat, Data mining-driven approahces for process monitoring and diagnosis, 2009.
[26] D. Fernández-Francos, O. Fontenla-Romero, A. Alonso-Betanzos, One-class convex hull-based algorithm for classification in distributed environments, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2018) 1–11.
[27] P. Casale, O. Pujol, P. Radeva, Approximate polytope ensemble for one-class classification, Pattern Recognition 47 (2014) 854–864.
[28] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, vol. 1, MIT Press, Cambridge, 2016.
[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, Journal of Machine Learning Research 11 (2010) 3371–3408.
[30] E. Jove, H. Aláiz-Moretón, J.L. Casteleiro-Roca, E. Corchado, J.L. Calvo-Rolle, Modeling of bicomponent mixing system used in the manufacture of wind generator blades, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, pp. 275–285.
[31] P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big data classification, Physical Review Letters 113 (2014), 130503.
[32] C. Sanchez-Hernandez, D.S. Boyd, G.M. Foody, One-class classification for mapping a specific land-cover class: Svdd classification of fenland, IEEE Transactions on Geoscience and Remote Sensing 45 (2007) 1061–1073.
[33] J.H. Shin, B. Lee, K.S. Park, Detection of abnormal living patterns for elderly living alone using support vector data description, IEEE Transactions on Information Technology in Biomedicine 15 (2011) 438–448.
[34] L.A. Shalabi, Z. Shaaban, Normalization as a preprocessing engine for data mining and the approach of preference matrix, in: 2006 International Conference on Dependability of Computer Systems, pp. 207–214.
[35] MathWorks, Autoencoder, https://es.mathworks.com/help/deeplearning/ref/trainautoencoder.html, 01-29-2019.
[36] A.P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (1997) 1145–1159.



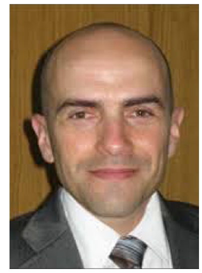
Esteban Jove received a M.S. degree in Industrial Engineering from the University of Leon in 2014. After two years working in the automotive industry, he joined the University of A Coruña, Spain, where he is Professor of Power Electronics in the Faculty of Engineering since 2016. He is a Ph.D student in the University of La Laguna and his research has been focused on the use of intelligent techniques for nonlinear systems modelling

Roberto Casado-Vara received the degree in mathematics from the University of Salamanca and the master's degree in big data and visual analytics from the International University of la Rioja. He is currently pursuing the Ph.D. degree in computer engineering with the BISITE Research Group. He is with Viewnext as a Data Scientist and is a Powercenter Consultant for important clients in the pharmaceutical and public administration sectors. He is a member with the BISITE Research Group. As a researcher, his interests include deep learning, advanced mathematical models for intelligent robust and non-linear control and monitoring, blockchain, and knowledge discovery data as well as other fields.

José-Luis Casteleiro-Roca received a B.S. degree from the University of A Coruña in 2003, a M.S. degree in Industrial Engineering from the University of Leon in 2012, and currently he is a Ph.D. student in the University of La Laguna. He is a Technical Engineer in the Spanish Navy since 2004 and, since 2014, he is also part of the teaching and research staff of the UDC as a part-time associate professor. His main lines of research are focused on applying expert system technologies to the diagnosis and control systems and to intelligent systems for control engineering and optimization.

Juan-Albino Méndez-Pérez is full professor at the University of La Laguna since 1993 in the Department of Computer Science and Systems Engineering. His teaching activity has been focused on system modelling and control. He works on lines of research related to control engineering: fuzzy control, predictive control and control applications. He earned a PhD degree in 1998 which had been recognized and awarded with the title of the best PhD Thesis in the field of engineering. He has participated in 25 research projects (7 were organized by the Spanish government, 9 by the government of the Canary Islands and 9 by private companies). He was the principal investigator in three of them.

Zita Vale received the diploma degree in electrical engineering and the Ph.D. degree, both from the University of Porto, Porto, Portugal, in 1986 and 1993, respectively. She is currently the Director of GECAD and a Professor with the Polytechnic of Porto, Porto, Portugal. Her main research interests include artificial Intelligence applications to power system operation and control, electricity markets, and distributed generation.



José-Luis Calvo Rolle received M.S. and Ph.D. degrees in Industrial Engineering from the University of Leon in 2004 and 2007, respectively. He is an Associate Professor of Automatic Control and the head of the Industrial Engineering Department, Faculty of Engineering, University of A Coruña, Spain. His main research areas are associated with the application of expert system technologies in diagnosis and control systems and in intelligent training systems for control engineering,