S.I. : CYBERSECURITY APPLICATIONS OF COMPUTATIONAL INTELLIGENCE

A distributed topology for identifying anomalies in an industrial environment

Francisco Zayas-Gato[1] · Álvaro Michelena[1,2] · Esteban Jove[1,2] · José-Luis Casteleiro-Roca[1] ·
Héctor Quintián[1] · Paulo Novais[3] · Juan Albino Méndez-Pérez[4] · José Luis Calvo-Rolle[1,2]

Abstract

The devastating consequences of climate change have resulted in the promotion of clean energies, being the wind energy the one with greater potential. This technology has been developed in recent years following different strategic plans, playing special attention to wind generation. In this sense, the use of bicomponent materials in wind generator blades and housings is a widely spread procedure. However, the great complexity of the process followed to obtain this kind of materials hinders the problem of detecting anomalous situations in the plant, due to sensors or actuators malfunctions. This has a direct impact on the features of the final product, with the corresponding influence in the durability and wind generator performance. In this context, the present work proposes the use of a distributed anomaly detection system to identify the source of the wrong operation. With this aim, five different one-class techniques are considered to detect deviations in three plant components located in a bicomponent mixing machine installation: the flow meter, the pressure sensor and the pump speed.

1 Introduction

European Union (EU) is committed to generate a new ecological transition to become in the world's first climate-neutral continent by 2050. This effort requires a significant investment from public and private entities that EU tries to impulse through its European Green Deal's Investment Plan [1]. In this sense, renewable energies play a key role in this plan to fight against the fossil fuel markets. Among

✉ Esteban Jove
esteban.jove@udc.es

1 Department of Industrial Engineering, University of A Coruña. CTC, Avda. 19 de febrero s/n, 15405 Ferrol, A Coruña, Spain

2 University of A Coruña, CITIC, Campus de Elviña, s/n, 15008 A Coruña, Spain

3 Department of Informatics, University of Minho, Algoritmi Center, 4710-057 Braga, Portugal

4 Department of Computer Science and System Engineering, Universidad de La Laguna, Avda. Astrof. Francisco Sánchez s/n, 38200 S/C de Tenerife, Spain

the different options, wind power is the most extended in EU [2]. According to [3], in 2018, the global electricity from renewable sources represented only a 25.6 % of the total share. This value reveals a significant development in comparison with previous years [4, 5]. Although, nowadays, the hydroelectric energy is the most common, covering 16 % of the generation, the wind technologies have experienced a remarkable increase during the last two decades. In [6], it is stated that the wind power installed grew from 17 GW in 2000 to 591 GW in 2018. According to this sequence, there are some forecasts that estimate a global share of 23 % in 2030 [7]. This represents four to five times the current share [8]. In this context, there are several researches oriented to optimise and improve different renewable energies systems [9]. Specially, in case of wind turbines, it takes significant relevance the wind blades in terms of shape [10] and materials [11, 12].

Most of the commercial wind blades are made of polyester or epoxy reinforced with glass fibre or carbon fibre. During the manufacturing process, a key step occurs when the epoxy is mixed with catalytic to obtain the resin for the infusion process [13]. Due to the non-Newtonian

nature of both fluids, this is not a trivial step, since the mix of both fluids must kept in the same proportion for any flow demanded, and the presence of noise during monitoring can hinder the control of the system.

The facility under study obtains carbon fibre as final product, whose features in terms of stiffness, tensile and compression strengths are suitable to the use in wind generator blades. It is an interesting alternative to glass fibre, since it shows greater stiffness with lower density than the glass fibres, resulting in lighter blades. However, an accurate proportion of both bicomponents should be ensured to avoid possible wrinkles that could lead to undesired material features [14]. Then, flow rate deviations could have an important impact on material, such as product waste, defective material during curing process, or shortening of blade life. Both scenarios result in economic losses.

Therefore, an early diagnosis of any deviation from the normal operation of the system is crucial to take actions in several directions. On the one hand, the anomaly detection can be used to carry out corrective maintenance, and on the other hand, it can be used to develop fault tolerant systems. This means that the control system can continue the process trying to avoid the propagation of the anomaly effect. In both cases, besides the detection of an unexpected event, it is interesting to determine the part of the system that has suffered the fault. Then, it is desirable to develop a distributed topology, where an anomaly detection system is placed over each sensor or actuator. The left side of Fig. 1 represents the implementation of an anomaly detection system applied to the entire mixing facility, and right side depicts a distributed topology. This second topology has the advantage of locating the source of the anomaly.

In this sense, anomaly detection techniques [15] provide an important advantage during monitoring process to identify the abnormal behaviour of the system. From a generic approach, an anomaly can be considered as an unusual pattern that does not an expect behaviour [16]. The used techniques for anomaly detection can be categorised

in three main types, based on type of learning methods used:

- Type 1: Supervised Learning: in this case, all samples are labelled as normal and anomalous. Then, modelling techniques are applied over each class.
- Type2: Unsupervised Learning: here, there is not a previous knowledge about the dataset and therefore samples are not labelled.
- Type 3: Semi-supervised Learning: in this case, only samples belonging to normal operation are used to train the models, so just the normal operation is modelled.

Given the fact that in most cases, only information from correct operation is available, the definition of this circumstance is mandatory to determine an anomalous situation [16, 17]. Then, it is defined the concept of one-class classifiers that cover all the instances that belong to the target set, which is the set representing the normal operation. The use of one-class classifiers has been widely used for many different applications, such as anomaly detection, fraud detection or medicine diagnosis. However, this semi-supervised approach offers only the possibility of labelling data as target or non-target data. In [18], a one-class classifier is implemented to determine the occurrence of anomalies in a bearing system. Although it is an interesting tool, it is not capable of locating or, at least, narrowing the source of the anomaly. Focusing on a different approach, in [19], a virtual sensor using intelligent techniques is developed to determine the value measured by a sensor located in an industrial plant. Then, the deviation between real and predicted value is the criteria to determine anomalies. This approach can be considered to identify the source of the anomaly, and thus favour the diagnostic and supervision of the plant.

This work faces the anomaly detection in an industrial plant used to obtain bicomponent material used for wind generator blades. The proposal takes advantage of one-class techniques to implement a distributed system capable of locating anomalies in specific parts of the facility. To achieve this goal, a real dataset gathered during the correct

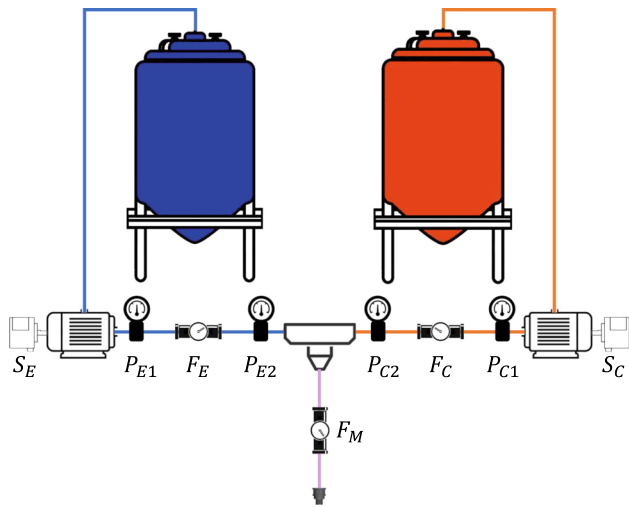**Fig. 1** Global and distributed topology of an anomaly detection system
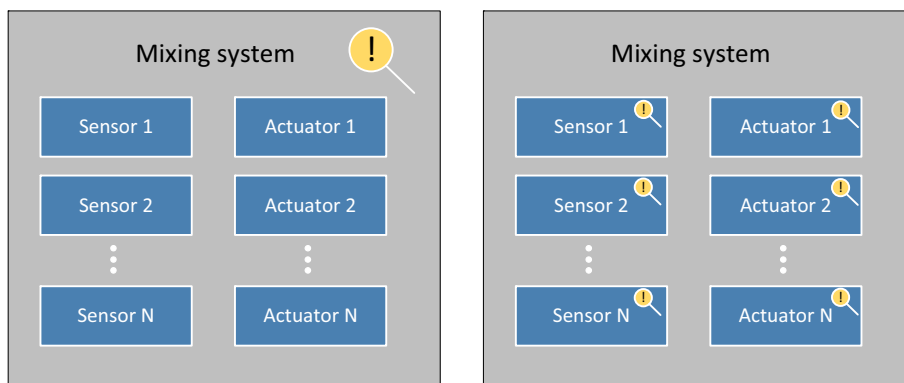
Fig. 2 Bicomponent mixing machine scheme

operation with different operating points is considered. Then, a deep analysis of different one-class techniques is carried out to evaluate the anomaly detection using three different plant anomalies. The anomalies are created by deviating the values measured with a flow meter, the pressure and the speed in a pump.

The present research is organised as follows. After this contextualisation section, the case of study section presents the bicomponent mixing machine. Later, Sect. 3 introduces the theoretical concepts of the four one-class anomaly detection techniques used in this work, to present in Sect. 4

the experiments and results and finally derive the conclusions and future works in Sect. 5.

2 Case study

In this section, the bicomponent mixing machines are described, detailing its main components and explaining its workflow. Also, the recorded dataset is described in terms of measured variables and sampling procedure, presenting an example of the dataset graphically.

2.1 Bicomponent mixing machine for turbine blade material

During fabrication of a turbine blade, the shape of the blade is created by combining a resin with reinforcement of fibre glass or carbon. In order to produce such resin, it is necessary an industrial machine to mix two primary fluids, the epoxy and the catalytic to obtain a final product which presents high tensile and compressive strengths and great chemical resistance [20]. This industrial mixing machine is schematised in Fig. 2.

As can be seen in Fig. 2, the system is monitored by means of a total of 9 sensors. Each fluid line has one pressure sensor at the output of the pumps (P_{E1}, P_{C1}) and another at the input of the mixing valve (P_{E2}, P_{C2}) and also one flow meter per line (F_{E2}, F_{C2}). Each pump has a speed sensor (S_E, S_C), and finally, the flow of mixed material at the output line is also measured (F_M).

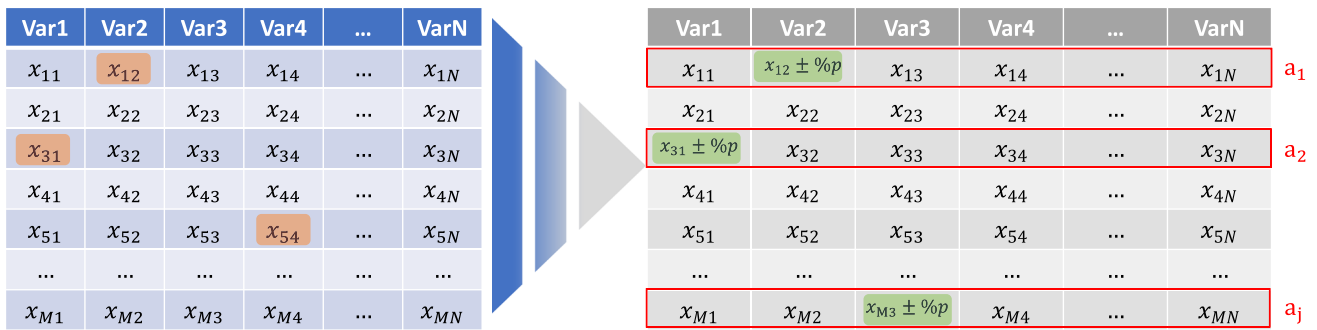Fig. 3 Examples of variables measured
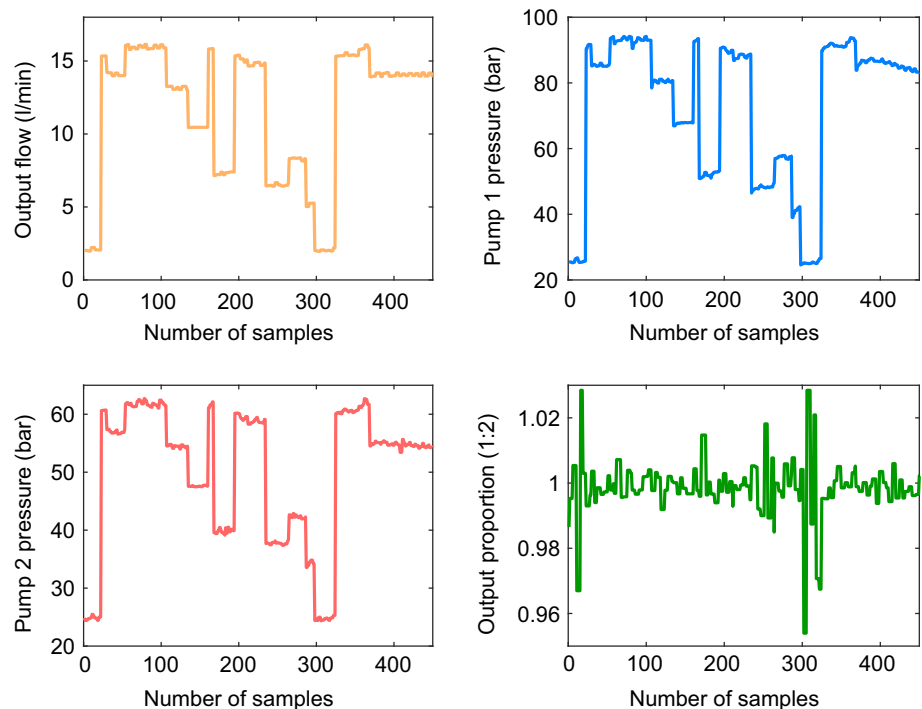
Fig. 4 Measurement deviation

The workflow is as follows:

1. Both fluids are stored in separated tanks and sent to the mixing valve by centrifugal pumps controlled through three phase variable frequency drives (VFD).
2. A control system adjusts the pumps speed, acting over each VFD, based on information sampled in the three flow meters to keep a constant output flow and homogeneous material.

2.2 Dataset

With the aim of generating an accurate anomaly detection model, the bicomponent mixing process is monitored by using a total of nine sensors, presented in previous Section. The monitoring process has a sample period 0.5 sec during an hour and 15 min. After removing wrong measures (null or negative values), the final dataset consists of 8549 samples and nine variables.

In order to illustrate how sample values varies in time, four of the measures are presented in Fig. 3 during 8 min.

As all data samples represents the normal operation of the system, anomalies have been generated following the process shown in Fig. 4.

Here, j anomalies were generated by modifying in a $p\%$ one of the random variables in a total of j random samples from a MxN dimensional dataset.
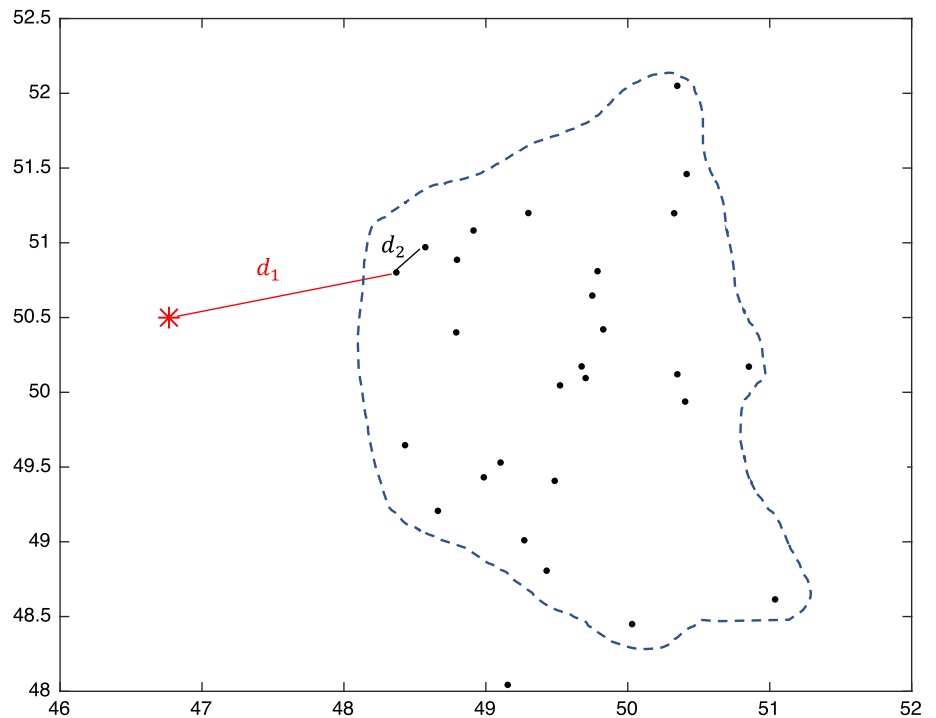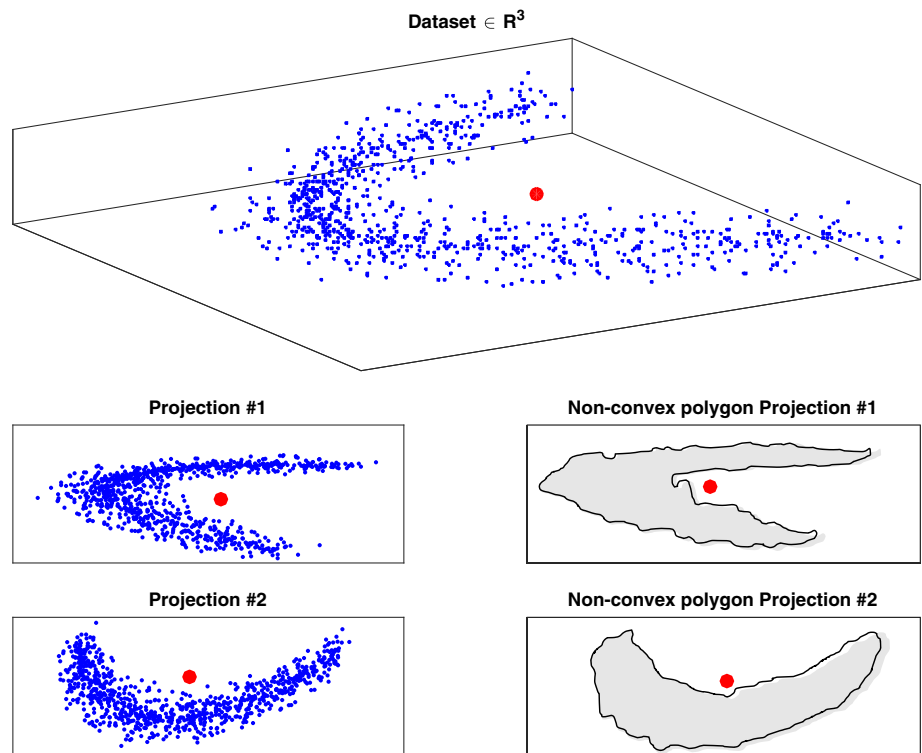
Fig. 5 kNN distances

Fig. 6 NCBoP projections

Dataset $\in \mathbf{R}^3$

Projection #1

Non-convex polygon Projection #1

Projection #2

Non-convex polygon Projection #2

3 Soft computing methods to validate the proposal

This section details the one-class techniques used to detect anomalies over the case of study previously presented. All algorithms work only with samples from normal behaviour of the system, and later anomalies are used to evaluate the performance of each one.

3.1 k-nearest neighbour

The k-Nearest Neighbour (kNN) for anomaly detection is based on relative distances between neighbour training samples and new test samples, therefore the local density of the hypersphere made of its kth nearest neighbours [21], determines whether a sample x is considered as an anomaly or not.

It can be expressed by Eq. (1) that calculates such density d, as the relation between the distance of a new testing sample x to the kth nearest training data neighbour $kNN^{tr}(x)$ and the local distance from the kth neighbour to its kth neighbour [17]

$$d(x) = \frac{d_1}{d_2} = \frac{||x - kNN^{tr}(x)||}{||kNN^{tr}(x) - kNN^{tr}(kNN^{tr}(x))||} \quad (1)$$

Figure 5 represents both distances graphically in a easier way.

The value of k plays a significant role in the classifier performance, and it depends on the dataset structure shape [21].

3.2 Minimum spanning tree

This one-class method is based on target set modelling by means of the structure obtained by a minimum spanning tree (MST). It relies on the assumption that two points $p_i, p_j \in \mathbb{R}^n$ that belong to target class should be neighbours in $\mathbb{R}^n$ representation [22]. Then, a linear transformation can be found for these points and all points considered as target class. As this set commonly contains more than two objects, then, more than one transformation can be applied. Hence, for a dataset D with n instances, $(n-1)$ linear transformations can be found [22]. Then, the MST consists of a set of edges e_{ij} that specify the linear transformations of each point. It is implemented a graph ensuring the absence of loops and minimising the total length of the edges. It can be defined as a problem of finding $(n-1)$ edges that implement a tree with a minimum total weight, as shown in Eq. (2) and subjected to Eq. (3).

$$\sigma_{e \in \phi} ||e|| \quad (2)$$

$$||\phi|| = (n-1), \phi \subset \epsilon \quad \exists! path(\{p_i, p_j\}), \quad \forall \{p_i, p_j\} \in D, \quad i > j \quad (3)$$

Once the MST is trained with data from the target set, the distance between a new test object p to training set D,

which is also the target set, is calculated as the shortest distance to the set of $(n-1)$ edges of the tree, as shown in Eq. (4). If this distance exceeds a threshold, β can be derived from a quantile function of the obtained MST.

$$d_{MST}(\boldsymbol{p}|D) = min_{e_{ij} \in MST} d(\boldsymbol{p}|e_{ij}) \qquad (4)$$

3.3 Non-convex boundary over projections

Non-Convex Boundary over Projections (NCBoP) is a novel one-class classification algorithm that overcome the weaknesses presented by well-known Convex Hull over random projections [23, 24].

The main basis of this method is to approximate the boundaries of a dataset $S \in \mathbb{R}^n$ using the non-convex hull over π random projections on $2D$ planes and then, determine the non-convex limits on that plane, reducing in this way the complexity of calculating the non-convex limits over $\mathbb{R}^n$.

The process to determine the non-convex polygon that contains a set of points $D = d_0, d_1, ..., d_i$ follows the next steps:

– Look for the point with the lowest y-coordinate (d_0). In case tie, the point with the lowest x-coordinate is selected.
– Find the k-nearest points to that current point. For this, the vectors $d_0 d_i$ with $i \in \{1, \cdots, n\}$ are implemented, looking for the smallest Euclidean distance.
– Sort the k-nearest points based on the angle resulting by the line with the x-axis. The further point from the starting point is selected.
– Once this first couple of points are calculated, a stack structure is created including the next third point.
– Then, it is checked if the next point in the list turns left (pushed into the stack) or right (point on the top is removed). This process is repeated until arrive again to the starting point.

Once the non-convex polygon is defined, it should have all training instances. Finally, a parameter λ is considered to increase or decrease the polygon vertexes v of the non-convex polygon $NC(X)$ from the centroid c, according to Eq. (5).

$$v^{\lambda} : \{\lambda v + (1-\lambda)c \mid v \in NC(X)\} \qquad (5)$$

Then, when a testing point is presented to the algorithm, if such point falls outside the non-convex polygon in at least one of the projections, it is considered as an anomaly, otherwise it represents the normal behaviour of the system. Figure 6 presents a 3D example where a sample that falls outside the limits of the non-convex polygon for one the calculated projections.

3.4 Principal component analysis

The principal component analysis [25] can be described as a linear operation involving unsupervised learning for data compression whose aim is to find that orthogonal basis that maximises the data variance for a given dimension of basis.

The final goal is to generate a new subspace using as new components the eigenvalues of the covariance matrix of the original input space. From a geometrical point of view, it consists on a rotation of the original axes, which can be expressed by Eq. (6).

$$y_i = W_i^T x^d \qquad (6)$$

where x^d in an N-dimensional space onto vectors in an M-dimensional space $(x_1...x_N)$, where $M \leq N$, W_i are the N eigenvectors of the covariance matrix, and y_i are the projected original data onto the new output M-dimensional subspace $(y_1...y_M)$.

Finally, to the criteria applied to identify an anomaly [26–28], consists of calculating the reconstruction error and determine if this reconstruction error is greater than the maximum threshold. The reconstruction error is expressed in Eq. (7) as the difference between the original dataset and its projection over the new subspace Eq. (8), computed in the original data. Such projection can be expressed as:

$$f(x) = ||x - x_{proj}||^2 \qquad (7)$$

$$x_{proj} = W(W^T W)^{-1} W^T x \qquad (8)$$

This technique offers good results when the subspace is clearly linear [17].

3.5 Support vector data description

The support vector machine (SVM) is a supervised learning used for classification and regression tasks [21]. Its main goal is to map the training set into a hyperspace and, then, implement a hyperplane to maximises the distance between classes [29].

Support vector data description (SVDD) aims to find a minimum volume space that contains all training data samples. Therefore, given a dataset X in $\mathbb{R}^n$, SVDD will try to find the hypersphere with centre a and radius R that includes most of the training data samples. It can be formulated as an optimisation problem with Eqs. (9) and (10).

$$F(\boldsymbol{R}, \boldsymbol{a}, \boldsymbol{\xi_i}) = \boldsymbol{R}^2 + C \sum_i \boldsymbol{\xi_i} \qquad (9)$$

$$||\boldsymbol{x_i} - \boldsymbol{a}||^2 \leq \boldsymbol{R}^2 + \boldsymbol{\xi_i} \qquad \boldsymbol{\xi_i} \leq \boldsymbol{R}^2 \qquad (10)$$

Due to the possible presence of noise in the dataset, the parameter ξ is used to control the slack in the limits of the

Fig. 7 Schematic description of $k - fold$ with $k = 5$
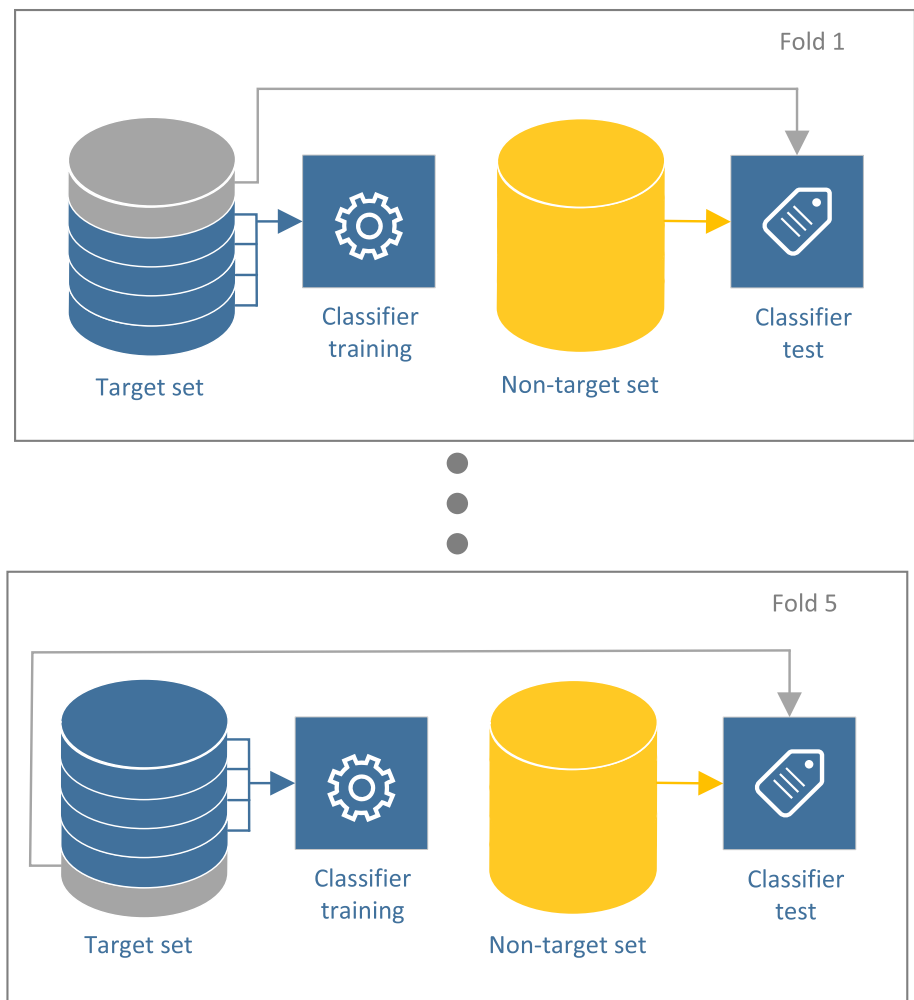
hypersphere. Parameter C is also considered to adjust the volume-errors trade-off.

When a new testing sample is presented to the trained SVDD, the distance from this new sample to the centre of the hypersphere is calculated and if this distance is greater than radius R, it is considered as an outlier.

4 Experiments and results

4.1 Experiments setup

The performance of each one-class technique was evaluated according to the different configurations described next:

– **kNN**:

  – Number of neighbours (NN) = 1:5:25.
  – Fraction of anomalies in the target set (v): 0 : 0, 05 : 0, 25.

– **MST**:

– Fraction of anomalies in the target set (v): 0 : 0, 05 : 0, 25.

– **NCBoP**:

  – Number of projections $\pi = 5, 10, 50, 100, 500, 1000$.
  – Expansion parameter $\lambda = 0, 5 : 0.1 : 1, 5$.

– **PCA**:

  – Fraction of anomalies in the target set (v): 0 : 0, 05 : 0, 25.
  – Components $\beta = 1 : 1 : 9$.

– **SVDD**:

  – Fraction of anomalies in the target set (v): 0 : 0, 05 : 0, 25.
  – Kernel width (%) = 0.1 : 0, 1 : 10.

The process followed to evaluate the performance of each classifier is based on the $k - fold$ cross-validation method, with $k = 5$, which is depicted in Figure 7. This procedure consists of dividing in a random way the target set in five

Table 1 Best results for each technique and flow percentage deviation

	Deviation (%)	Norm	NN	Nu	AUC (%)	STD (%)	t_{tr} (s)
kNN	5	Type C	1	0,05	94,27	0,31	4,02
	10	Type C	1	0,05	96,58	0,57	3,92
	15	Type C	1	0,05	97,38	0,54	3,99
	20	Type C	1	0,05	97,45	0,36	4,16
	25	Type C	1	0,05	97,61	0,46	4,05
	Deviation (%)	Norm	-	v	AUC (%)	STD (%)	t_{tr} (s)
MST	5	Type C	-	0,15	91,40	0,47	7,54
	10	Type C	-	0,10	95,73	0,36	7,68
	15	Type C	-	0,15	97,78	0,29	7,92
	20	Type C	-	0,10	98,90	0,37	8,29
	25	Type C	-	0,05	99,53	0,17	7,70
	Deviation (%)	Norm	λ	π	AUC (%)	STD (%)	t_{tr} (s)
NCBoP	5	Type C	1	1000	87,62	0,50	119,71
	10	Type C	1	500	92,05	0,77	51,94
	15	Type C	1	500	92,60	0,70	51,58
	20	Type C	1	100	93,50	0,67	10,37
	25	Type C	1	100	94,65	0,58	9,92
	Deviation (%)	Norm	β	v	AUC (%)	STD (%)	t_{tr} (s)
PCA	5	Type C	9	0,05	96,22	0,28	0,03
	10	Type C	8	0,05	97,49	0,42	0,02
	15	Type C	9	0,00	97,50	0,23	0,02
	20	Type C	9	0,00	99,62	0,03	0,02
	25	Type C	9	0,00	99,99	0,05	0,04
	Deviation (%)	Norm	Σ	v	AUC (%)	STD (%)	t_{tr} (s)
SVDD	5	Type C	1	0,15	61,97	0,96	160,84
	10	Type B	1	0,00	77,78	0,53	223,20
	15	Type B	1	0,00	85,59	2,17	214,44
	20	Type B	1	0,00	95,91	0,18	218,89
	25	Type B	1	0,00	98,23	0,13	219,21

different groups. Then, five classifiers are implemented using the 80 % of the data and leaving the 20 % to the test phase. The use 5 folds ensures that all instances are considered for the training and test phases. Then, the classifier is tested using the non-target set and 20 % of the target data.

Furthermore, for each of the configurations tested, the data followed three different conditioning prior the classifier training:

– Type A: normalisation between the interval 0 to 1.
– Type B: normalisation using the mean and standard deviation of each variable. This is also known as ZScore normalisation [30].
– Type C: the data remains unaltered without any pre-processing technique.

The performance of each classifier configuration is evaluated using the well-known area under the receiving operating characteristics curve (AUC) measure [31]. The AUC

Table 2 Best results for each technique and speed percentage deviation

	Deviation (%)	Norm	NN	v	AUC (%)	STD (%)	t_{tr} (s)
kNN	5	Type C	1	0,05	95,34	0,27	3,96
	10	Type C	1	0,05	97,50	0,72	3,44
	15	Type C	1	0,05	97,60	0,49	4,04
	20	Type C	1	0,05	97,51	0,43	3,50
	25	Type C	1	0,05	97,70	0,51	3,97
	Deviation (%)	Norm	-	v	AUC (%)	STD (%)	t_{tr} (s)
MST	5	Type C	-	0,15	94,85	0,29	8,09
	10	Type C	-	0,10	98,81	0,34	7,89
	15	Type C	-	0,05	99,06	0,16	7,82
	20	Type C	-	0,05	99,52	0,20	7,68
	25	Type C	-	0,05	99,65	0,19	7,85
	Deviation (%)	Norm	λ	π	AUC (%)	STD (%)	t_{tr} (s)
NCBoP	5	Type C	1	50	94,71	0,63	5,26
	10	Type C	1	50	98,73	0,27	5,50
	15	Type C	1	50	98,97	0,36	5,01
	20	Type C	1,2	1000	99,91	0,08	100,95
	25	Type C	1,2	50	100,00	0,00	5,45
	Deviation (%)	Norm	β	v	AUC (%)	STD (%)	t_{tr} (s)
PCA	5	Type C	4	0,05	90,81	0,48	0,02
	10	Type A	5	0,05	95,55	0,34	0,02
	15	Type C	4	0,05	97,49	0,41	0,02
	20	Type C	3	0,05	97,51	0,57	0,02
	25	Type C	4	0,00	97,56	0,00	0,04
	Deviation (%)	Norm	σ	v	AUC (%)	STD (%)	t_{tr} (s)
SVDD	5	Type C	4	0,00	86,26	5,34	211,56
	10	Type C	3	0,00	89,79	8,28	228,39
	15	Type B	1	0,00	90,28	0,16	206,23
	20	Type B	1	0,00	92,87	0,24	203,14
	25	Type B	1	0,00	98,98	0,20	200,63

combines the true positive and false positive rates, obtaining a unique measure of the classifier performance. From a statistical point of view, this value represents the probability of classifying a random positive instance as positive [31]. Furthermore, in contrast to other parameters such as sensitivity, precision or recall, AUC is not sensitive to class distribution, which is a significant advantage especially in one-class tasks [32]. This parameter is calculated five times, one for each fold, and its mean value and standard deviation between runs (STD, in %) are also

registered. As a comparative analysis of each technique is sought, the time needed to train (t_{tr}) each classifier is also registered as a measure of the computational cost of the classifiers.

To evaluate the proposal of a distributed one-class classifier for detecting anomalies in different plant components, three different variables are deviated from its correct operation:

– Flow rate of epoxy resin (FE).

Table 3 Best results for each technique and pressure percentage deviation

	Deviation (%)	Norm	NN	v	AUC (%)	STD (%)	t_{tr} (s)
kNN	5	Type C	1	0,05	97,35	0,38	3,45
	10	Type C	1	0,05	97,57	0,42	3,93
	15	Type C	1	0,05	97,69	0,62	3,92
	20	Type C	5	0,05	97,50	0,33	4,53
	25	Type C	1	0,00	99,75	0,60	4,06
	Deviation(%)	Norm	-	v	AUC (%)	STD (%)	t_{tr} (s)
MST	5	Type C	-	0,15	97,97	0,39	7,91
	10	Type C	-	0,05	99,41	0,31	8,12
	15	Type C	-	0,05	99,65	0,18	7,66
	20	Type C	-	0,05	99,63	0,23	9,45
	25	Type C	-	0,05	99,64	0,18	7,89
	Deviation (%)	Norm	λ	π	AUC (%)	STD (%)	t_{tr} (s)
NCBoP	5	Type C	1	1000	97,90	0,40	101,96
	10	Type C	1	100	98,79	0,16	9,95
	15	Type C	1,2	500	99,92	0,04	51,30
	20	Type C	1,4	500	100,00	0,00	49,04
	25	Type C	1,4	500	99,99	0,02	49,48
	Deviation (%)	Norm	β	v	AUC (%)	STD (%)	t_{tr} (s)
PCA	5	Type C	5	0,05	97,49	0,45	0,02
	10	Type B	9	0,00	98,77	0,39	0,04
	15	Type A	8	0,00	99,99	0,02	0,03
	20	Type C	5	0,00	100,00	0,00	0,04
	25	Type C	4	0,00	100,00	0,00	0,05
	Deviation (%)	Norm	σ	v	AUC (%)	STD (%)	t_{tr} (s)
SVDD	5	Type C	4	0,00	88,17	7,21	242,89
	10	Type C	3	0,00	86,25	11,03	237,67
	15	Type B	1	0,00	91,14	0,28	200,78
	20	Type B	1	0,00	94,40	0,44	219,02
	25	Type B	1	0,00	96,20	0,78	226,43

– Speed of catalyst pump (SE).
– Pressure measure in epoxy resin pump (PE1).

These anomalies are generated by modifying the original measurements a $\pm 5\%$, $\pm 10\%$, $\pm 15\%$, $\pm 20\%$ and $\pm 25\%$.

4.2 Results

This subsection describes the best results, in terms of AUC for each technique, component with anomalous behaviour,

and percentage deviation. This information is reflected in Tables 1, 2 and 3. Furthermore, to provide a deeper insight of the results, the behaviour of each technique depending on the deviated percentage is presented in Figs. 8, 9 and 10.

Focusing on the anomaly appearance of the flow meter sensor, it is important to remark that PCA is in almost all cases the technique with the best AUC value, with at least
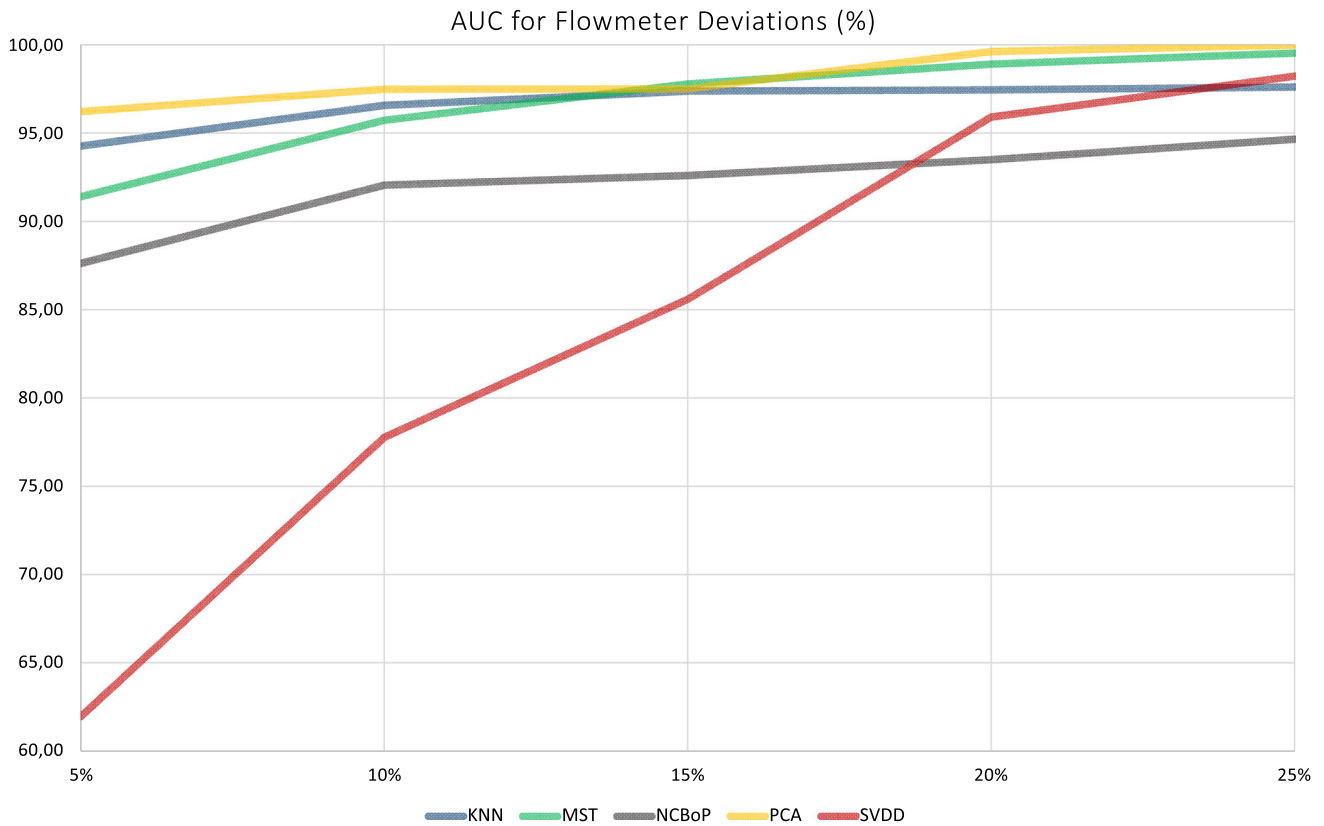
Fig. 8 Classifier performance depending on flow percentage deviation

96,22 %. These results are achieved with a significantly high number of components, varying between 8 and 9. In these experiments, MST and kNN compete with PCA in terms of AUC. However, they present a remarkable disadvantage comparing with PCA, which is the training time, whose values are more than one hundred times greater. NCBoP does not reach the level of PCA, MST and kNN, but in all cases exceeds a minimum of 87,62 % AUC. Finally, SVDD shows a gradually increase in the anomaly detection task as the percentage deviation grows, starting with a 61,97 % for a 5% deviation and finishing with a 98,23 % when the deviation is 25 %. Furthermore, SVDD is the unique technique that offer the best results with ZScore normalisation.

Regarding the detection of anomalous speed measurements, the obtained results are slightly different than the ones obtained with the flow meter deviations. In this case, NCBoP and MST present the best performance in terms of AUC. However, NCBoP has a computation time that is significantly greater than the presented by MST. PCA and kNN show similar performance, especially with 15 %, 20 % and 25 % deviation. It is interesting to emphasise the fact that, in this case, the number of components β is significantly lower than the obtained in first experiment. The trend shown by SVDD in previous experiment is repeated in this case, since the AUC increases gradually with the percentage variation.
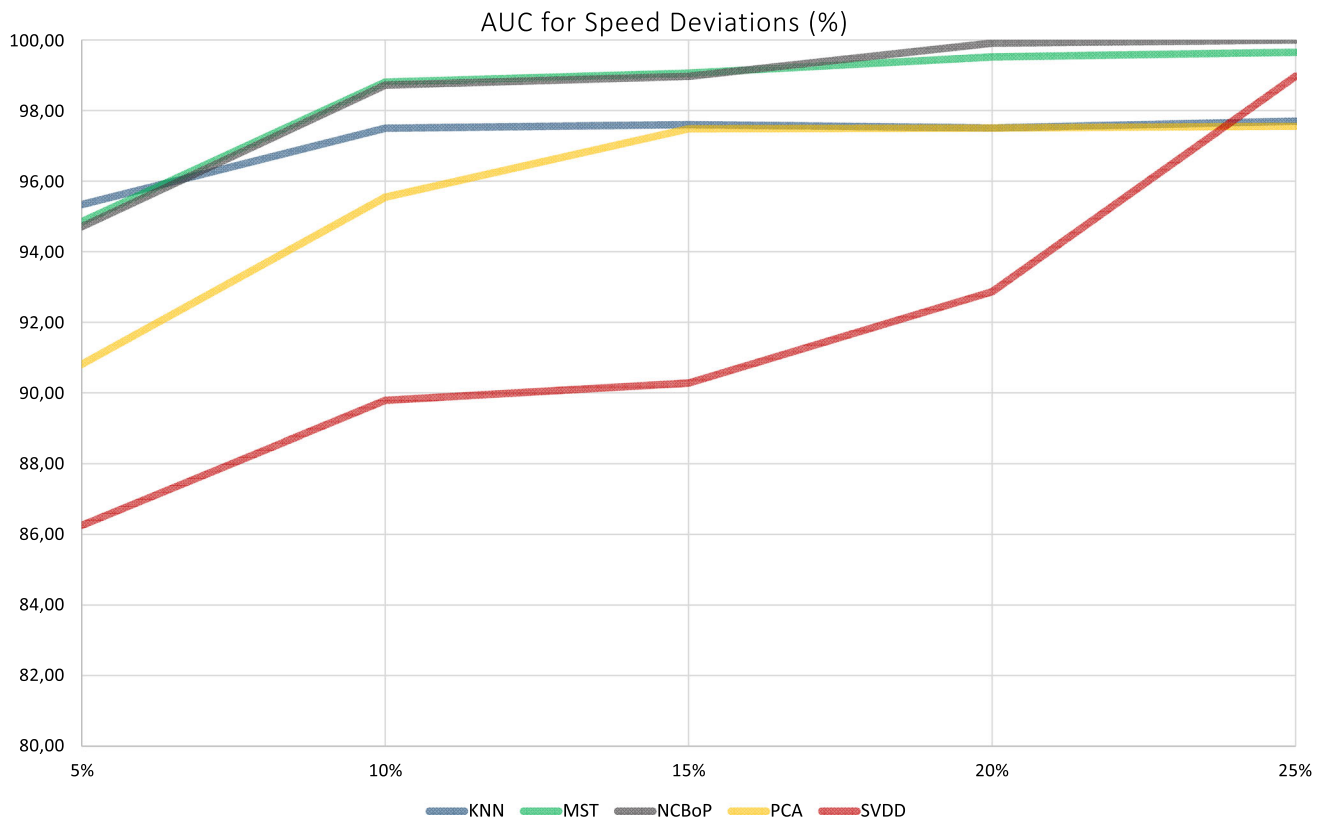
Fig. 9 Classifier performance depending on speed percentage deviation

In respect of the third experiment, in which the pressure measured is modified, PCA, MST and NCBoP compete again for the best results, being PCA the fastest technique. This is a key issue to select this technique in case of a tie. SVDD is once again the worst technique taking AUC and training time values. It is interesting to indicate that it is, as it happened in the two previous experiments, the unique technique whose best performance is achieved with ZScore pre-processing. Finally, it worth mentioning that the number of neighbours is one in all cases but one.

5 Conclusions and future works

The present work presents five different one-class classifier techniques to detect anomalies in three components of an industrial system used to obtain a bicomponent material. These anomalies are artificially generated by deviating the original measurement a variable percentage, modifying the flow, pressure and speed measured. From the experiments developed and the obtained results, in general terms, it can be assumed that, in the three experiments, the one-class techniques presented a successful performance. A valuable contribution of this work consists of the implementation of three one-class classifiers to consider the possibility locating the source of the anomaly. Hence, when the
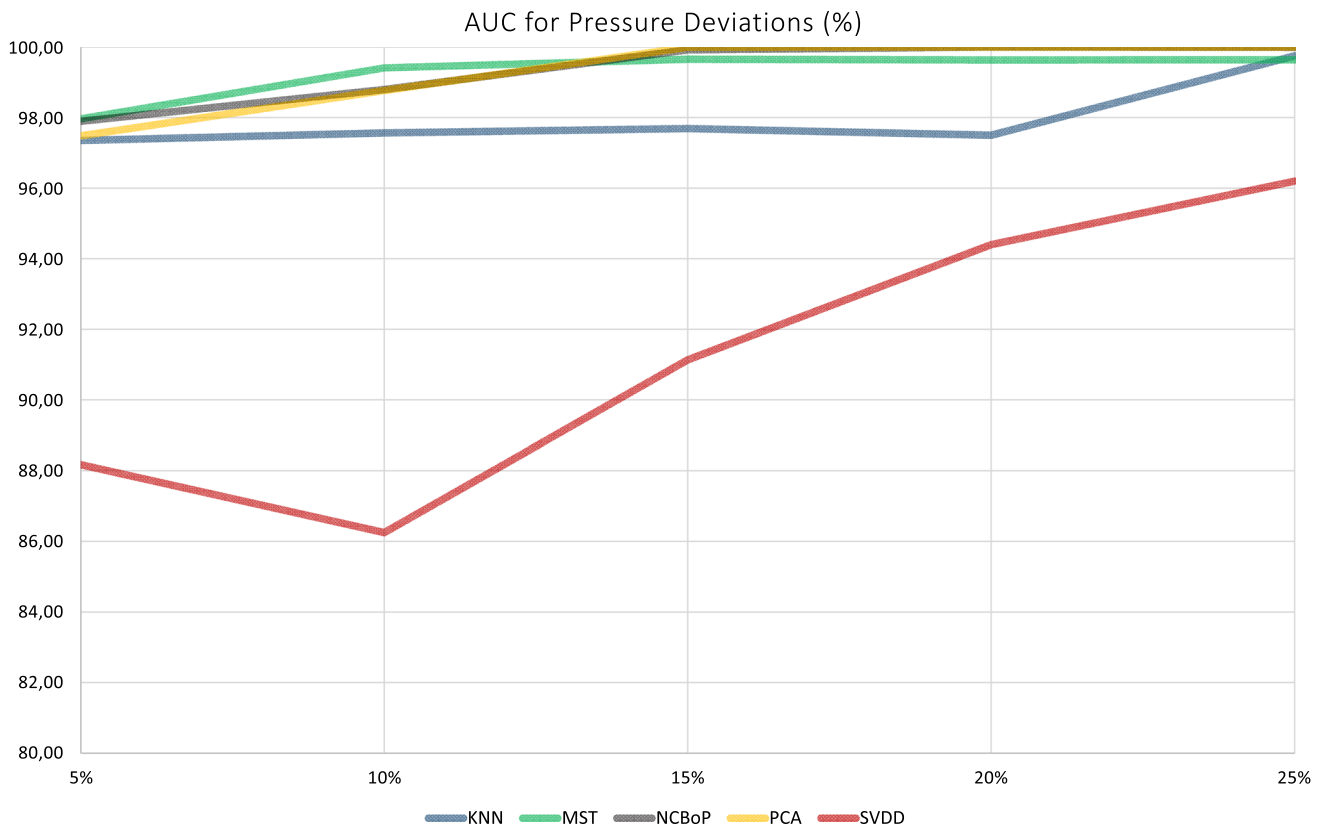
AUC for Pressure Deviations (%)

Fig. 10 Classifier performance depending on pressure percentage deviation

classifier assigned to the flow meter detects a deviation, it can be used to isolate its measurements and avoid its propagation. Besides this application, the idea of incorporating this distributed topology could reduce significantly the maintenance cost, since early detection of wrong performance can be addressed.

In future works, it could be considered the use of imputation techniques or intelligent models to recover the real data when it is lost due to wrong sensor performance. Then, a combination of system modelling and distributed one-class topology would impact in the system optimisation. Furthermore, an online training stage as the system is in operation would improve the classifier performance. Finally, the possibility of applying a prior clustering process to obtain hybrid intelligent classifiers can be considered. Hence, each cluster would correspond to a different operating point.

Declarations

Conflict of interest Authors do not declare any conflict of interest.

References

1. European Commission (2021). A European Green deal. https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en. Accessed 18 February 2021
2. Eurostat (2021). Renewable energy statistics. https://ec.europa.eu/eurostat/statistics-explained/index.php/Renewable_energy_statistics#Wind_and_water_provide_most_renewable_electricity.3B_solar_is_the_fastest-growing_energy_source. Accessed 18 February 2021

3. Repsol's Economic Research Department (2020) Annual energy-statistics 2020. https://www.repsol.com/content/dam/repsol-cor porate/en_gb/energia-e-innovacion/annual-energy-statistics-2020_tcm14-168076.pdf. Accessed 4 Nov 2021

4. Owusu PA, Asumadu-Sarkodie S (2016) A review of renewable energy sources, sustainability issues and climate change mitigation. Cogent Eng 30(1):1167990

5. Lund H (2007) Renewable energy strategies for sustainable development. Energy 32(6):912–919

6. Asociación Empresarial Eólica (2019) Anuario eólico. La voz del sector 2019. https://www.aeeolica.org/images/Publicaciones/Anuario-Elico-2019.pdf. Accessed 15 Oct 2021

7. Infield D, Freris L (2020) Renewable energy in power systems. Wiley, Chichester

8. BP plc (2019). Renewable energy - wind energy. https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy/renewable-energy.html.html#wind-energy. Accessed 3 Mar 2021

9. Zuo Y, Liu H (June 2012) Evaluation on comprehensive benefit of wind power generation and utilization of wind energy. In: Software engineering and service science (ICSESS), 2012 IEEE 3rd international conference on. pp 635–638

10. Xudong W et al (2009) Shape optimization of wind turbine blades. Wind Energy Int J Progr Appl Wind Power Conv Technol 12(8):781–803

11. Cognet V, du Pont SC, Thiria B (2020) Material optimization of flexible blades for wind turbines. Renew Energy 160:1373–1384

12. Brondsted P, Nijssen RP (Eds) (2013) Advances in wind turbine blade design and materials. Woodhead Publishing Limited

13. Bank LC, Arias FR, Yazdanbakhsh A, Gentry TR, Al-Haddad T, Chen JF, Morrow R (2018) Concepts for reusing composite materials from decommissioned wind turbine blades in affordable housing. Recycling 3(1):3

14. Mishnaevsky L, Branner K, Petersen HN, Beauson J, McGugan M, Sorensen BF (2017) Materials for wind turbine blades: an overview. Materials 10(11):1285

15. Miljković D (2011) Fault detection methods: a literature survey. In: MIPRO, 2011 proceedings of the 34th international convention. pp 750–755. IEEE

16. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):15

17. Tax DMJ (2001) One-class classification: concept-learning in the absence of counter-examples [ph. d. thesis]. Delft University of Technology

18. Zeng M, Yang Y, Luo S, Cheng J (2016) One-class classification based on the convex hull for bearing fault detection. Mech Syst Signal Process 81:274–293

19. Jove E, Casteleiro-Roca JL, Quintián H, Méndez-Pérez JA, Calvo-Rolle JL (2019) Virtual sensor for fault detection, isolation and data recovery for bicomponent mixing machine monitoring. Informatica 30(4):671–687

20. Jove E, Casteleiro-Roca J, Quintián H, Méndez-Pérez JA, Calvo-Rolle JL (2020) Anomaly detection based on intelligent techniques over a bicomponent production plant used on wind generator blades manufacturing. Revista Iberoamericana de Automática e Informática industrial 17(1):84–93

21. Sukchotrat T (2008) Data mining-driven approaches for process monitoring and diagnosis. Doctoral dissertation, The University of Texas at Arlington

22. Juszczak P, Tax DM, Pe E, Duin R (2009) Minimum spanning tree based one-class classifier. Neurocomputing 72(7–9):1859–1869

23. Casale P, Pujol O, Radeva P (2011) Approximate convex hulls family for one-class classification. In: International workshop on multiple classifier systems. pp 106–115. Springer

24. Fernandez-Francos D, Fontenla-Romero O, Alonso-Betanzos A (2017) One-class convex hull-based algorithm for classification in distributed environments. IEEE Trans Syst Man Cybern Syst 50(2):386–396

25. Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2(6):559–572

26. Mazhelis O (2006) One-class classifiers: a review and analysis of suitability in the context of mobile-masquerader detection. South African Comput J 2006:29–48

27. Chiang LH, Russell EL, Braatz RD (2000) Fault detection and diagnosis in industrial systems. Springer Science & Business Media, Heidelberg

28. Wu J, Zhang X A pca classifier and its application in vehicle detection. In: IJCNN'01. International joint conference on neural networks. Proceedings (Cat. No. 01CH37222), volume 1, IEEE, pp 600–604

29. Rebentrost P, Mohseni M, Lloyd S (2014) Quantum support vector machine for big data classification. Phys Rev Lett 113:130503. https://doi.org/10.1103/PhysRevLett.113.130503

30. Shalabi LA, Shaaban Z (May 2006) 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

31. Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27(8):861–874

32. Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159