Information Fusion 65 (2021) 50–57

Contents lists available at ScienceDirect

Information Fusion

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

Full length article

A new method for anomaly detection based on non-convex boundaries with random two-dimensional projections

Esteban Jove [a,*], José-Luis Casteleiro-Roca [a], Héctor Quintián [a], Juan-Albino Méndez-Pérez [b], José Luis Calvo-Rolle [a]

[a] University of A Coruña, CTC, Department of Industrial Engineering, CITIC Avda. 19 de febrero s/n, 15405, Ferrol, A Coruña, Spain
[b] University of La Laguna, Department of Computer Science and System Engineering, Avda. Astrof. Francisco Sánchez s/n, 38200, S/C de Tenerife, Spain

ARTICLE INFO

Keywords:
One-class
Anomaly detection
Projection methods
Convex hull
Boundary
Limits

ABSTRACT

The implementation of anomaly detection systems represents a key problem that has been focusing the efforts of scientific community. In this context, the use one-class techniques to model a training set of non-anomalous objects can play a significant role. One common approach to face the one-class problem is based on determining the geometric boundaries of the target set. More specifically, the use of convex hull combined with random projections offers good results but presents low performance when it is applied to non-convex sets. Then, this work proposes a new method that face this issue by implementing non-convex boundaries over each projection. The proposal was assessed and compared with the most common one-class techniques, over different sets, obtaining successful results.

1. Introduction

Over the past decades, the use of classifiers has been commonly applied to solve a wide range of problems in many different fields, such as medicine [1] or industrial systems [2], among others [3]. The classification process consists of assigning an object to its class or category, where the object is defined by a set of feature values [4]. Typically, a classification problem cannot be solved using simple known rules [5]. Thus, the classifier implementation must face a learning process from a set of training objects. Once it is obtained, it is able to label unseen future objects.

Depending on the number of classes to be assigned, the classification can be binary or multi-class. In both cases, significant amount of instances belonging to each category must be ensured to achieve a good classifier [4]. However, in many applications, it is possible to obtain the training objects only from one class, because obtaining data from other classes is expensive, difficult, or even impossible [6]. All these cases, that may represent critical unknown events, or system failures belong to the non-target class or negative class. This kind of problems, where the objects can be assigned to a known class (target or positive class) or to the rest of possible classes (non-target or negative class), is defined as one-class classification [7,8], novelty detection [9] or outlier detection [10,11]. The main concern in one-class classification tasks is to obtain a proper description of the target class from the training set, due to the lack of information about the outliers behavior [7].

From a given set of objects, corresponding to the target class, different approaches can be considered to face the issue of one-class classification: density methods, reconstruction methods and boundary methods [4,12]. The most direct method to achieve a one-class classification is based on establishing a threshold in the density estimation of the training data. The use of different density distributions, such as Gaussian, Poisson or Parzen Density, have been proven to be successful [7]. However, a significantly high amount of training data is needed to achieve good results. Fig. 1 shows the threshold level over a one-dimensional Gaussian distribution. If a future object does not exceed the threshold distribution, it is classified as outlier.

Another common approach to achieve the novelty detection is based on reconstruction methods. With this method, a model is implemented from the training data with the aim of minimizing the reconstruction error. Once the model is obtained, objects from non-target class would lead to high reconstruction error, and the outlier should be detected. This approach has been validated with different techniques, such as k-means, Self-Organization Maps (SOM), Learning Vector Quantization (LVQ), Principal Component Analysis or Autoencoder Networks. Fig. 2 shows the main basis of this approach, where the real input u is compared with the reconstructed input u_R.

The last one-class approach consists of determining the spatial limits of the training instances [4]. Hence, once the boundaries are set, the

* Corresponding author.
E-mail addresses: esteban.jove@udc.es (E. Jove), jose.luis.casteleiro@udc.es (J.-L. Casteleiro-Roca), hector.quintian@udc.es (H. Quintián), jamendez@ull.es (J.-A. Méndez-Pérez), jlcalvo@udc.es (J.L. Calvo-Rolle).

https://doi.org/10.1016/j.inffus.2020.08.011
Received 31 March 2020; Received in revised form 20 July 2020; Accepted 4 August 2020
Available online 11 August 2020
1566-2535/

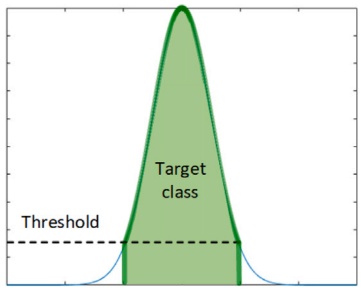

Fig. 1. Outlier detection using Gaussian distribution.

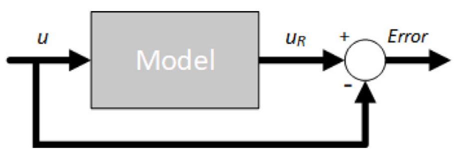

Fig. 2. General approach for reconstruction method.

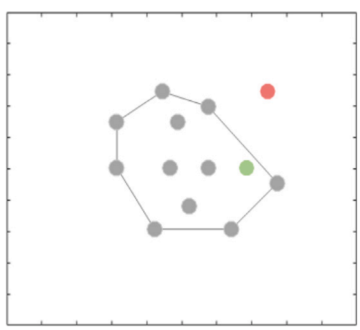

Fig. 3. Novelty detection in $\mathbb{R}^2$ using convex hull.

criteria to identify a non-target class object, is based on the distance to the decision boundary. In comparison with density methods, this approach can give better results when the training data size is low. The use of One-Class Support Vector Machine (OCSVM), maps the data into a high-dimensional space and then, a hyper-plane that maximizes the distance between the data and the origin is obtained. A similar process is followed with Support Vector Data Description (SVDD), but in this case, a hyper-sphere is implemented instead of a hyper-plane [7].

An effective method to obtain the approximated boundaries of a target class is based on the convex hull of the training set [13,14]. In this case, the novelty detection is solved from a geometrical point of view, using the dataset convex hull [4]. An outlier detection process in $\mathbb{R}^2$ is shown in Fig. 3. The gray points represent the training instances, the green point represents a test data belonging to the target class and the red point is an outlier.

This method leads to good performance in one-class classification tasks [15]. However, the convex hull definition in high-dimensional spaces is computationally expensive [16]. In [13], a convex hull approximation of a given dataset is obtained from p random 2D projections, reducing significantly the computational cost. The most critical weakness of this method appears when the dataset has non-convex nature, especially when the outliers lie inside the convex surface. This paper proposes a new method that solves the problem of determining the limits of non-convex datasets. The proposal was validated by testing different convex and non-convex sets, obtaining successful results in general terms.

The paper is structured as follows: after the present introduction, the motivation of this work is described. Section 3 provides a detailed

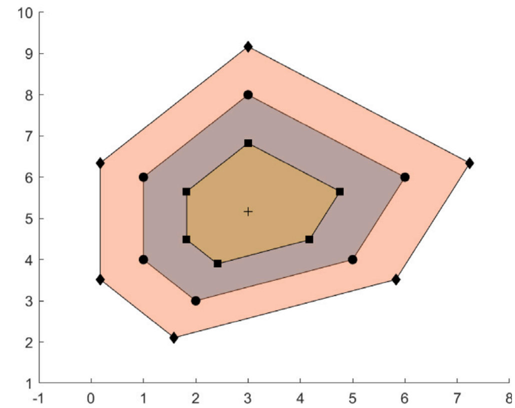

Fig. 4. Enlargement and contraction of a convex hull.

explanation about the proposed method. Then, the experiments and results are shown in Section 4. Finally, the conclusions and future works are listed.

2. Motivation

This section provides a general overview of the one-class technique based on convex hull calculation, whose limitations represent the main motivation of this work. As explained in Section 1, it is possible to define the limits of a target class from a set of training objects using its convex hull. The convex hull CH of a dataset $D \in \mathbb{R}^n$ is known as the minimum convex set that contains all points, according to Eq. (1) [14].

$$CH(D) = \left\{ \sum_{i=1}^{|D|} \beta_i x_i \mid (\forall i : \beta_i \geq 0) \wedge \sum_{i=1}^{|D|} \beta_i = 1, x_i \in D \right\} \tag{1}$$

Once the convex hull $CH(D)$ is calculated, the outlier is detected when a new object does not belong to the hull. This method can provide good performance if the dataset does not have anomalous objects, since the appearance of outliers in the training set may lead to an inaccurate decision model [17]. Thus, the size of the convex hull can be modified using a parameter $\lambda \in [0, +\infty)$, according to Eq. (2) [18].

$$v_\lambda : \{\lambda v + (1-\lambda)c \mid v \in CH(D)\} \tag{2}$$

where v contains the vertexes of the original convex hull with respect to the center $c = (1/D)\sum_i x_i, \forall i = 1, \ldots, |D|$, and v_λ contains the modified vertexes of the convex hull. From this equation, it is concluded that values of λ greater than 1 expand the convex hull and lower than 1, contract it. An example of this feature is shown in Fig. 4, where the dots represent the original convex hull, the diamonds delimit the enlarged convex hull and the squares represent the area contracted. This vertexes modification is performed from the center, identified with a cross.

However, this approach presents two main weaknesses: the computational cost and the wrong performance with non-convex sets. The calculation of the convex hull of a high-dimension dataset requires a significant computational cost [13]. If a dataset is composed of N samples in $\mathbb{R}^n$, the cost estimation of the convex hull calculation is $O(N^{(n/2)+1})$ [13]. This problem is solved by using the Approximate Polytope Ensemble (APE) technique. This technique consists of making p random 2D projections of the original dataset. Then, for each 2D projection, the convex hull is calculated. Once the convex hull is modeled, the criteria used to determine the nature of a test data is the following: if the point is out of at least one of these projections, it is labeled as outlier. The main idea of this approach can be seen in Fig. 5, where a dataset in $\mathbb{R}^3$ is projected in two 2D planes, where the red dot represents an outlier. In this case, the novelty detection is correctly achieved because the red dot is out of the convex hull of projection #2.

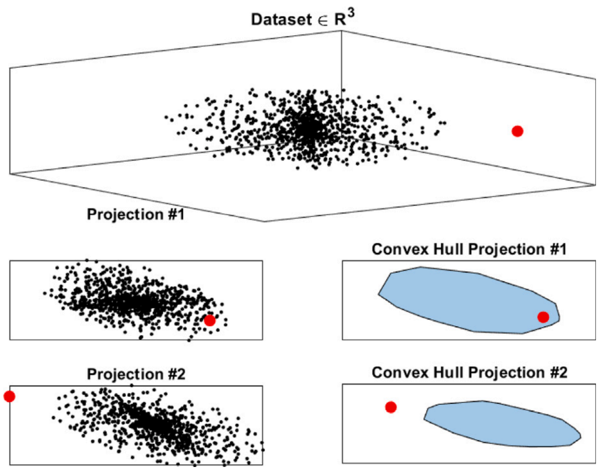

Fig. 5. Novelty detection using the approximate convex hull.

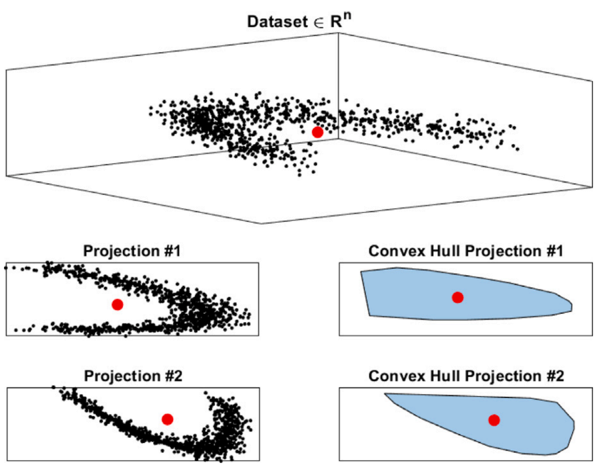

Fig. 6. Novelty detection using the approximate convex hull.

Despite the good performance shown by this method [12–14], its use in non-convex sets, may lead to unsuccessful classification. When the dataset is not convex, there are significant cases where the approximate polytope does not detect the anomalous points. This inaccurate classification would happen when the outliers are well separate from initial dataset but they lie inside the convex hull. An example of this situation with a c-shaped set is shown in Fig. 6. In this case, the anomaly, represented with a red dot, cannot be detected.

3. Non-convex boundary over projections

This work proposes the Non-Convex Boundary over Projections (NCBoP) method that aims to avoid the main weaknesses of APE described in previous section. To achieve this objective, once p random projections are made, the convex hull calculation is replaced by a non-convex polygon that will be the border of the points projected over each of those auxiliary planes $\pi_1, \dots, \pi_p$. The main idea of this novel method when applied over a non-convex set in $\mathbb{R}^3$ can be seen in Fig. 7. Then, this section describes the proposed technique to build the non-convex polygon as well as its mathematical aspects.

3.1. Mathematical background

The first problem that needs to be solved is to check if a point is completely within a non-convex polygon. Before moving on to the solution of this problem, let us first check whether a point is to the left or to the right of a line segment.

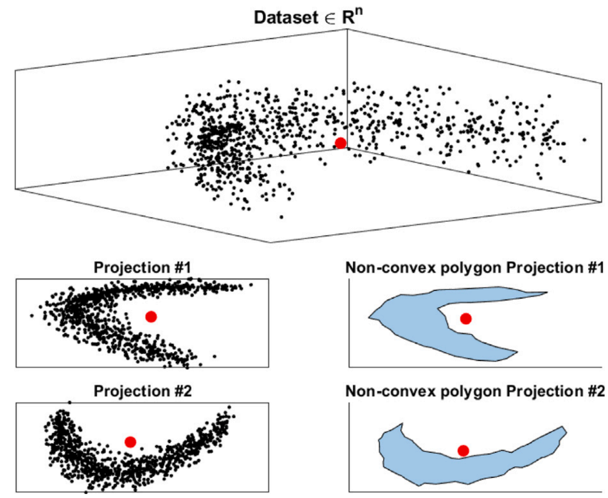

Fig. 7. Novelty detection using Non-convex polygon.

Lemma 3.1. *Let (a, b) be a line segment with coordinates of the end points of the segment (x_1, y_1) and (x_2, y_2) respectively. Let $p = (x, y)$ be a point somewhere in the plane XY, and let*

$$Ax + By + C = 0 \tag{3}$$

be the equation of the segment where $A = -(y_2 - y_1), B = (x_2 - x_1), C = -(Ax_1 - By_1)$.

Then, a point $p = (x, y)$ lies on the left of the line segment given by A,B and C if,

$$Ax + By + C > 0 \tag{4}$$

a point lies on the right of the line segment given by A,B and C if,

$$Ax + By + C < 0 \tag{5}$$

Finally, the point lies exactly on the line if,

$$Ax + By + C = 0 \tag{6}$$

Since a polygon is a combination of more than two line segments, the aim is to check if the point lies inside the polygon

Lemma 3.2. *Let $a_1, \dots, a_n$ be a convex polygon. The point $p = (x_y)$ is inside the polygon if it lies on the left of edges $a_1a_2, \dots, a_{n-1}a_n, a_na_1$.*

Proof. Follows from Lemma 3.1. □

Remark. It is drawn a horizontal ray originating from the point p and extend it towards infinity in the right direction, and then, it is counted the number of intersection the ray makes with the edges of the polygon. A point $p = (x, y)$ lies inside a non-convex polygon if the number of intersection is even, the point is outside the polygon, otherwise it is inside the polygon.

Proof. Follows from Lemma 3.2. □

3.2. Algorithm description

Let us describe the algorithm designed to build a non-convex polygon that is the border of the cloud of points projected in the auxiliary planes. This algorithm has been designed supporting us in the following works [19–21]. The step by step working of the algorithm on the point set P is given below.

Let $P = \{P_0, P_1, \dots, P_n\}$ be a set of points in the plane XY. The algorithm's first step is to find the starting point from which the non-convex pole is going to be built. Next, the polygon will be built so that this is the border of the non-convex hull. The detailed steps of the algorithm are enlisted below.

1. Find the point (p_0) with the lowest y-coordinate. If a tie occurs, select the point with the lowest x-coordinate.
2. Find the k-nearest points to the current point. For this, the vectors P_0P_i with $i \in \{1, \dots, \}$ are made and the shortest euclidean distance in the real plane is sought.
3. Sort the k-nearest points based on the polar angle, that is, the angle made by the line with the x-axis. This way, you will find the P_i with $i \in \{1, \dots, n\}$ point with the lowest polar angle. To determine if the segment P_0P_1 or the segment P_0P_3 makes the greater angle with the axis x, it is calculated the vector product of the vectors P_1P_0 and P_1P_3. If the cross product is positive, it means that the vector P_1P_0 is clockwise from the vector P_1P_3 with respect to the x axis. This indicates that the angle made by the P_1P_3 vector is greater.
4. After classification, the furthest point from P_0 is kept and all other points are removed.
5. The first two points of the list are always on the non-convex hull. It its maintained a stack data structure to keep track of the non-convex hull vertices. It is pushed these two points and the next point P_3 on the list into the stack.
6. Now let us see if the next point in the list turns left or right (Lemma 3.1) from the two points at the top of the stack. If it turns to the left, it pushed this object into the stack. If it turns right, the item from the top of the stack is removed and the process is repeated for the remaining items.
7. Loop to number 2 until come back to P_0, then, go next step.
8. The criteria to decide if the algorithm must be stopped, takes into consideration whether all the points are either in the non-convex polygon created by the algorithm (Lemma 3.1), or inside the non-convex polygon (Lemma 3.2). In the case that all the points are inside or in the polygon, the algorithm ends. Otherwise, it looks for the point closest to the point outside the polygon and then, a new iteration starts in step 2.

4. Experiments and results

In this section, the different experiments carried out and the achieved results are presented.

4.1. Performance assessment of the proposal

To validate the non-convex proposal, it was compared with the most typical one-class techniques, including the Approximate Polytope Ensemble, whose performance improvement is sought. The techniques were tested with different hyperparameter values with the aim of selecting the best possible configuration. These are summarized next:

- Approximate Polytope Ensemble (APE) [17].
  - Number of projections.
  - Expansion parameter.
- Autoencoder Artificial Neural Network (AANN) [22].
  - Hidden layer function.
  - Number of layers in the hidden layer.
  - Outlier fraction in the training set.
- Gaussian Model (GM) [7].
  - Model width.
  - Outlier fraction in the training set.

Table 1
Main features of each dataset.

Dataset	Instances	Target size	Outliers	Dimension
Normal	10 249	10 000	249	3
C-shaped	7649	7500	149	3
S-shaped	11 879	11 700	179	3
Y-shaped	6025	5850	175	3
Flower-Shaped	9143	9000	143	3
Breast Cancer	683	444	239	9
Cardio	1831	1655	176	21
Ionosphere	351	225	126	33
Letter Recognition	1600	1500	100	32
Vowels	1456	1406	50	12
Wine	129	119	10	13

- K-Centers (KC) [23].
  - Number of clusters.
  - Outlier fraction in the training set.
- K-Means (KM) [6].
  - Number of clusters.
  - Outlier fraction in the training set.
- K-Nearest Neighbor (KNN) [24,25].
  - Number of neighbors.
  - Outlier fraction in the training set.
- Minimum Spanning Trees (MST) [26].
  - Length of max paths.
  - Outlier fraction in the training set.
- Parzen Density Estimator (PDE) [27].
  - Width.
  - Outlier fraction in the training set.
- Principal Component Analysis (PCA) [28].
  - Components.
  - Outlier fraction in the training set.
- Support Vector Data Description (SVDD) [7].
  - Kernel Width.
  - Outlier fraction in the training set.

These algorithms were trained over two different groups of datasets, whose main features are detailed in Table 1:

- Convex and non-convex three-dimensional shapes comprise the first group: Normal distribution, C-shaped, S-shaped, Y-shaped, and Flower-shaped. These sets are generated artificially as a part of this work. In this case, the randomly generated outliers are placed nearby the target set ensuring by visual inspection that they are outside the positive class boundaries (Fig. 8).
- The second group of sets is collected from real applications, available in the ODDS benchmark [29]. In this case, the datasets used were: Breast Cancer Wisconsin, Cardio, Ionosphere, Letter Recognition, Vowels and Wine. They are chosen because they belong to a significant variety of fields, with also a wide range of dimensions, as shown in Table 1.

To evaluate the performance of the classifier, the Area Under the Receiving Operating Characteristics Curve (AUC) parameter was taken into consideration [30]. This parameter, that establishes a relationship between true positive and false positive rates, presents two main advantages. First, it is able to offer a single measure of the classifier performance, representing the probability of classifying as positive a

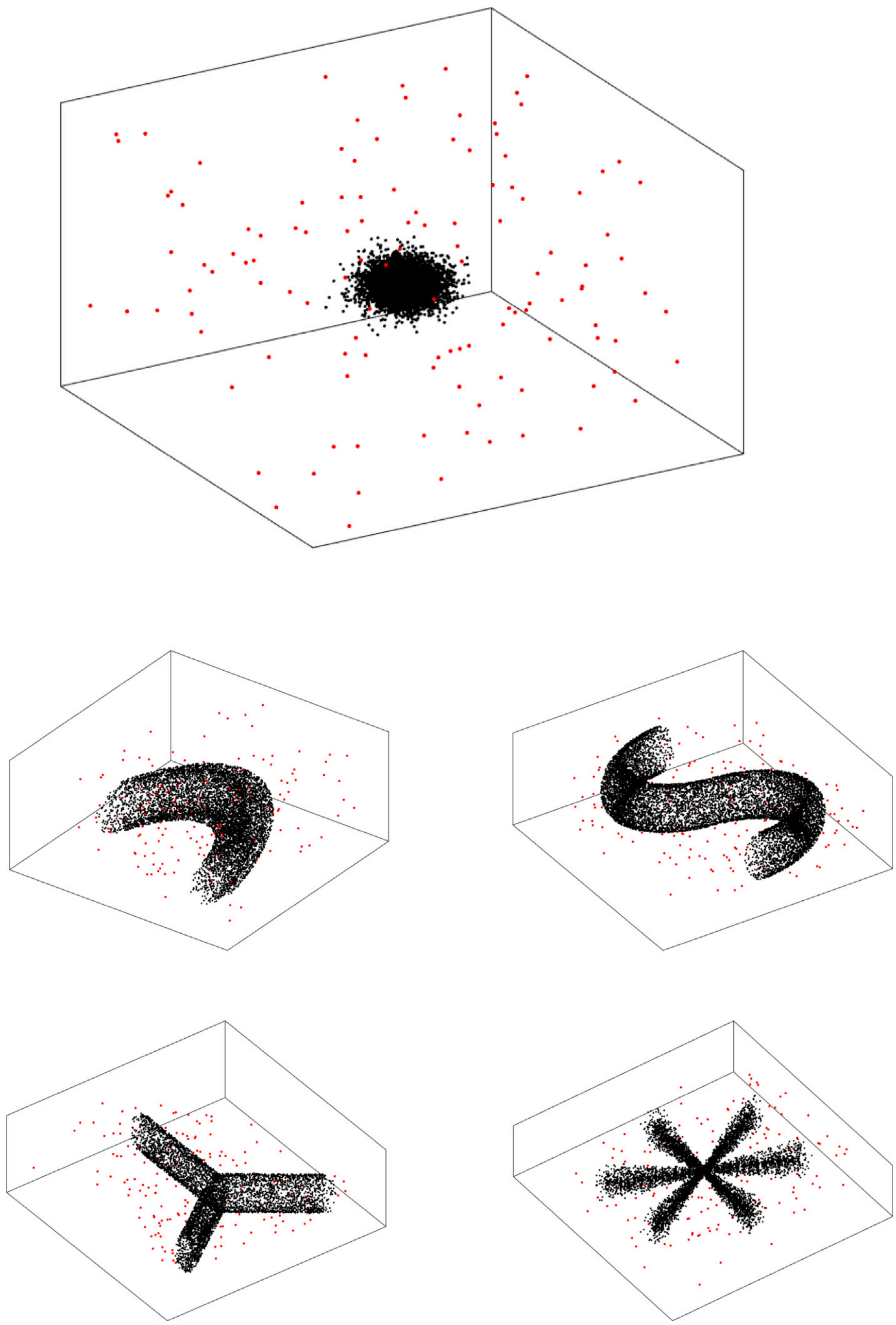

Fig. 8. Representation of the synthetic datasets used to validate the proposal.

random positive instance [30]. The second advantage of this parameter is the insensitivity to classes distribution changes [31], that is a especially relevant feature in one-class problems. A $k-fold$ cross-validation with $k = 10$ was implemented to ensure a reliable measure of each technique performance.

4.2. Results

The experiments configuration described above offered the results presented in this subsection. First, Table 2 shows the best AUC for each technique and dataset, which is the criteria to choose a configuration. Then, the time needed to implement the classifier and the time to calculate the nature of a test sample are shown in Tables 3 and Table 4, respectively.

4.3. Statistical analysis

As the main goal of the experiments' setup is to validate the proposal, a statistical analysis is mandatory [32,33]. In this work, two different statistical analysis were carried out. First, a Bonferroni–Dunn test was developed to compare the NCBoP with the rest of conventional one-class techniques [34]. After checking the results achieved over each dataset with a significance level at 5%, NCBoP only performs significantly better than PDE. The rest of one-class techniques remain inside the critical difference (CDD) around NCBoP ranking, as shown in Fig. 9.

However, this method is generally conservative, so a Wilcoxon signed-ranks test is also applied to evaluate the NCBoP performance

Table 2
AUC results over the tested datasets.

	NCBoP	APE	AANN	GM	KC	KM	KNN	MST	PCA	PDE	SVDD
Normal	100.00	100.00	99.57	100.00	99.97	100.00	100.00	100.00	94.95	98.18	99.99
C-shaped	93.93	85.64	86.02	84.98	96.29	95.29	97.54	98.81	96.39	82.85	99.69
S-shaped	92.63	80.76	86.02	84.98	96.29	95.57	98.97	98.84	95.92	74.74	94.83
Y-shaped	98.83	66.42	83.44	61.92	92.96	93.34	97.28	98.17	94.84	62.46	98.61
Flower-Shaped	98.07	60.75	80.72	84.68	89.80	89.70	97.50	98.53	94.69	80.72	93.39
Breast Cancer	95.65	89.87	94.87	95.88	94.80	96.44	96.43	95.95	88.52	94.35	96.35
Cardio	93.96	92.36	88.35	88.88	88.62	90.82	92.18	92.40	53.88	89.00	93.06
Ionosphere	90.69	90.81	90.76	90.19	91.61	90.70	91.19	90.88	53.64	91.94	92.03
Letter Recognition	73.03	70.08	73.20	77.43	61.02	70.62	87.40	86.83	50.50	76.05	73.53
Vowels	91.32	84.61	88.01	89.39	83.45	89.19	96.71	98.00	50.32	89.71	89.11
Wine	98.18	95.46	93.64	97.27	90.96	97.73	96.68	97.68	55.46	95.00	92.73

Table 3
Training times over the tested datasets.

	NCBoP	APE	AANN	GM	KC	KM	KNN	MST	PCA	PDE	SVDD
Normal	0.55	0.07	1.45	0.03	62.70	0.07	12.96	30.77	20.82	0.30	1631.60
C-shaped	0.69	6.62	2.19	0.01	34.01	0.04	6.39	13.32	8.97	0.05	401.87
S-shaped	0.61	2.95	2.19	0.01	34.01	0.18	22.83	67.76	18.80	0.06	1263.21
Y-shaped	1.02	0.46	1.67	0.01	23.44	0.03	3.81	6.96	3.60	0.05	211.69
Flower-Shaped	11.56	1.00	0.48	0.01	50.78	0.02	8.78	17.39	10.73	0.06	601.70
Breast Cancer	1.20	0.11	0.21	0.01	1.20	0.00	0.01	0.04	0.10	0.02	0.07
Cardio	9.68	0.19	0.17	0.01	1.30	0.01	0.22	0.42	0.49	0.02	5.40
Ionosphere	1.54	0.10	4.05	0.01	0.11	0.01	0.01	0.01	0.04	0.02	0.02
Letter Recognition	12.98	0.29	787.64	0.01	0.92	0.03	0.28	0.61	0.46	0.05	7.98
Vowels	7.96	0.78	1.95	0.00	0.69	0.01	0.16	0.31	0.35	0.02	1.75
Wine	1.47	0.01	1.72	0.01	0.15	0.01	0.01	0.02	0.01	0.04	0.02

Table 4
Calculation times over the tested datasets.

	NCBoP	APE	AANN	GM	KC	KM	KNN	MST	PCA	PDE	SVDD
Normal	25.96	3.18	0.01	0.00	0.12	0.00	0.58	1.71	0.14	0.01	0.14
C-shaped	45.45	437.98	0.02	0.00	0.16	0.01	0.38	0.71	0.12	0.01	0.01
S-shaped	26.72	125.09	0.02	0.00	0.16	0.01	0.69	1.96	0.21	0.01	0.12
Y-shaped	79.67	34.74	0.02	0.01	0.11	0.01	0.25	0.76	0.10	0.01	0.01
Flower-Shaped	655.75	54.97	0.02	0.00	0.15	0.00	0.41	1.01	0.11	0.01	0.19
Breast Cancer	257.70	23.03	0.08	0.02	0.01	0.01	0.02	0.11	0.03	0.01	0.01
Cardio	1703.62	32.77	0.04	2.80	0.03	0.01	0.07	0.82	0.04	0.01	0.01
Ionosphere	637.97	38.61	0.10	0.04	0.04	0.02	0.03	0.22	0.05	0.02	0.02
Letter Recognition	3062.40	68.04	0.10	0.01	0.07	0.03	0.10	1.23	0.09	0.03	0.04
Vowels	2469.40	241.39	0.07	0.01	0.05	0.02	0.07	0.45	0.05	0.01	0.02
Wine	4117.04	25.23	0.88	0.15	0.27	0.35	0.16	0.46	0.26	0.26	0.13

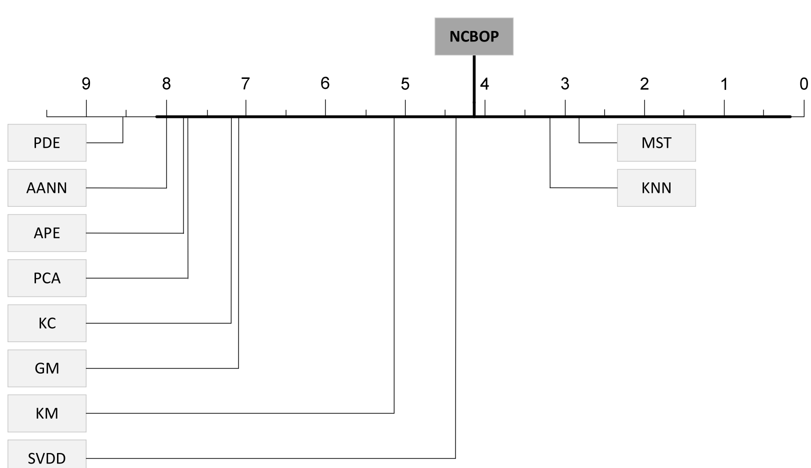

Fig. 9. Graphical representation of Bonferroni–Dunn test ($p = 0.05$, CD = 3.9697).

[34]. This non-parametric test establishes a comparison between each pair of classifiers (NCBoP against the rest), taking into consideration the differences over each dataset, ranking these differences. This test leads to two main conclusions ($p = 0.05$):

- NCBoP performs significantly better than APE, AANN, GM, KC, PCA and PDE.
- The null hypothesis of similar performance is accepted for KM, KNN, MST and SVDD.

4.4. Results overview

This subsection aims to detail an overview of the final results previously presented. In general terms, the proposed approach presented a successfully performance with all datasets. Furthermore, it is important to remark that this novel method overcomes the APE technique in all sets but one, with a remarkable difference in non-convex sets (C-Shaped, S-Shaped, Y-Shaped and Flower-Shaped). Then, the weaknesses exposed in the motivation section seems to be overtaken.

Besides the AUC performance, that has been analyzed through the statistical analysis, it is important to consider the computational cost of each technique, in terms of training time to achieve each classifier. In this field, NCBoP presents greater values when it is tested over the real dataset instead of the synthetic datasets, although the synthetic ones have a significantly more samples. Then, we can conclude that working with high-dimensional datasets results in greater training times.

The NCBoP presents a main point to be reinforced, which is the time needed to estimate the label of a new test sample. This situation is consequence of the number of projections configured to achieve the classifier. Increasing the number of random planes implies an increase in the number of projections to check if the data belongs to the non-convex polygon.

5. Conclusions and future works

The present research work proposes a novel method to implement one-class classifiers based on boundary methods. The main idea of this technique to improve the existing APE algorithm when it is applied over convex and non-convex sets. With this aim, instead of a convex hull, a non-convex polygon is constructed over each random projection of the training set. The proposal has been validated over eleven different datasets: five of them correspond to synthetic convex and non-convex three-dimensional sets, and the six left correspond to datasets from real applications. The proposal is compared with ten typical one-class techniques. After a statistical analysis, it is concluded that NCBoP presents performance rates that matches, at least all of them.

This contribution can present an interesting support to detect deviations in a wide variety of fields. The increasing competitiveness and the pursuit of energy efficiency, especially in developed nations, are focusing the attention in tools that help to detect anomaly situations. Hence, its implementation can complement predictive and corrective maintenance plans, and it can be a key part of systems optimization procedures in industries. In this sense, the low computational cost compared with other cutting edge one-class techniques can be a really interesting feature when the novelty detection system is implemented using the edge computing methodology.

As future works, there are many lines that can continue with the present research. First, it could be interesting to think about an online implementation that could offering the possibility of modifying the non-convex polygons as the system evolves. To implement this idea, the training time should be reduced. Then, it could be interesting to perform a preliminary study to determine the proper number of projections based on the number of instances and variables. Since many systems could present different operating points corresponding to the target class, which are clearly separate, the implementation of hybrid topologies could represent a good idea. These topologies could consist of dividing the target class into different groups using clustering algorithms.

CRediT authorship contribution statement

Esteban Jove: Conceptualization, Methodology. **José-Luis Casteleiro-Roca:** Software. **Héctor Quintián:** Writing - original draft, Formal analysis. **Juan-Albino Méndez-Pérez:** Visualization, Writing - review & editing. **José Luis Calvo-Rolle:** Writing - review & editing, Supervision, Project administration.

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] E.K. Alexander, M. Schorr, J. Klopper, C. Kim, J. Sipos, F. Nabhan, C. Parker, D.L. Steward, S.J. Mandel, B.R. Haugen, Multicenter clinical experience with the afirma gene expression classifier, J. Clin. Endocrinol. Metab. 99 (1) (2014) 119–125.

[2] G.A. Susto, A. Schirru, S. Pampuri, S. McLoone, A. Beghi, Machine learning for predictive maintenance: A multiple classifier approach, IEEE Trans. Ind. Inf. 11 (3) (2014) 812–820.

[3] C. Liang, F. Wen, Z. Wang, Trust-based distributed Kalman filtering for target tracking under malicious cyber attacks, Inf. Fusion 46 (2019) 44–50.

[4] M. Zeng, Y. Yang, S. Luo, J. Cheng, One-class classification based on the convex hull for bearing fault detection, Mech. Syst. Signal Process. 81 (2016) 274–293, http://dx.doi.org/10.1016/j.ymssp.2016.04.001, URL http://www.sciencedirect.com/science/article/pii/S0888327016300413.

[5] L.B. Booker, D.E. Goldberg, J.H. Holland, Classifier systems and genetic algorithms, Artif. Intell. 40 (1–3) (1989) 235–282.

[6] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 15.

[7] D.M.J. Tax, One-Class Classification: Concept Learning in the Absence of Counter-Examples (Ph.D. thesis), Technische Universiteit Delft, 2001.

[8] H.-J. Xing, W.-T. Liu, Robust adaboost based ensemble of one-class support vector machines, Inf. Fusion 55 (2020) 45–58.

[9] C.M. Bishop, Novelty detection and neural network validation, IEE Proc., Vis. Image Signal Process. 141 (4) (1994) 217–222.

[10] G. Ritter, M.T. Gallegos, Outliers in statistical pattern recognition and an application to automatic chromosome classification, Pattern Recognit. Lett. 18 (6) (1997) 525–539.

[11] B. Wang, Z. Mao, Outlier detection based on a dynamic ensemble model: Applied to process monitoring, Inf. Fusion 51 (2019) 244–258.

[12] E. Jove, J.M. Gonzalez-Cava, J.-L. Casteleiro-Roca, H. Quintián, J.A. Méndez-Pérez, J.L. Calvo-Rolle, Anomaly detection on patients undergoing general anesthesia, in: F. Martínez Álvarez, A. Troncoso Lora, J.A. Sáez Muñoz, H. Quintián, E. Corchado (Eds.), International Joint Conference: 12th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2019) and 10th International Conference on EUropean Transnational Education, ICEUTE 2019, Springer International Publishing, Cham, 2020, pp. 141–152.

[13] P. Casale, O. Pujol, P. Radeva, Approximate convex hulls family for one-class classification, in: C. Sansone, J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 106–115.

[14] D. Fernández-Francos, ó. Fontenla-Romero, A. Alonso-Betanzos, One-class convex hull-based algorithm for classification in distributed environments, IEEE Trans. Syst. Man Cybern. (2018) 1–11, http://dx.doi.org/10.1109/TSMC.2017.2771341.

[15] J. Wang, B. Yang, W. Zhang, B. Qin, Convex hull-based support vector machine rule extraction, in: 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, 2012, pp. 689–692.

[16] E. Jove, J.-L. Casteleiro-Roca, H. Quintián, J.A. Méndez-Pérez, J.L. Calvo-Rolle, Outlier generation and anomaly detection based on intelligent one-class techniques over a bicomponent mixing system, in: International Workshop on Soft Computing Models in Industrial and Environmental Applications, Springer, 2019, pp. 399–410.

[17] P. Casale, O. Pujol, P. Radeva, Approximate polytope ensemble for one-class classification, Pattern Recognit. 47 (2) (2014) 854–864, http://dx.doi.org/10.1016/j.patcog.2013.08.007, URL http://www.sciencedirect.com/science/article/pii/S0031320313003282.

[18] Z. Liu, J. Liu, C. Pan, G. Wang, A novel geometric approach to binary classification based on scaled convex hulls, IEEE Trans. Neural Netw. 20 (7) (2009) 1215–1220.

[19] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2009.

[20] C.D. Toth, J. O'Rourke, J.E. Goodman, HandBook of Discrete and Computational Geometry, Chapman and Hall/CRC, 2017.

[21] D.M. Mount, Cmsc 754 Computational Geometry, in: Lecture Notes, University of Maryland, 2002, pp. 1–122.

[22] 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, 2014, p. 4.

[23] T. Li, L. Du, Target discrimination for sar atr based on scattering center feature and k-center one-class classification, IEEE Sens. J. 18 (6) (2018) 2453–2461.

[24] S. Agarwal, A. Sureka, Using knn and svm based one-class classifier for detecting online radicalization on twitter, in: International Conference on Distributed Computing and Internet Technology, Springer, 2015, pp. 431–442.

[25] S. Garcia, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: Taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 417–435.
[26] P. Juszczak, D.M. Tax, E. Pe, R.P. Duin, et al., Minimum spanning tree based one-class classifier, Neurocomputing 72 (7–9) (2009) 1859–1869.
[27] 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, 2008, pp. 21–26.
[28] T. Ban, S. Abe, Implementing multi-class classifiers by one-class classification methods, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, 2006, pp. 327–332.
[29] S. Rayana, Odds library, 2016, http://odds.cs.stonybrook.edu.
[30] T. Fawcett, An introduction to roc analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.
[31] A.P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recognit. 30 (7) (1997) 1145–1159.
[32] S. Garcia, F. Herrera, An extension onstatistical comparisons of classifiers over multiple data setsfor all pairwise comparisons, J. Mach. Learn. Res. 9 (Dec) (2008) 2677–2694.
[33] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Dynamic classifier selection for one-vs-one strategy: avoiding non-competent classifiers, Pattern Recognit. 46 (12) (2013) 3412–3424.
[34] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.