Incorporating User Feedback Into One-Class Support Vector Machines for Anomaly Detection

Julien Lesouple
TéSA Laboratory
Toulouse, France
julien.lesouple@tesa.prd.fr

Jean-Yves Tourneret
INP-ENSEEIHT/IRIT/TéSA
University of Toulouse
Toulouse, France

***Abstract*—Machine learning and data-driven algorithms have gained a growth of interest during the past decades due to the computation capability of the computers which has increased and the quantity of data available in various domains. One possible application of machine learning is to perform unsupervised anomaly detection. Indeed, among all available data, the anomalies are supposed to be very sparse and the expert might not have the time to label all the data as nominal or not. Many solutions exist to this unsupervised problem, but are known to provide many false alarms, because some scarce nominal modes might not be included in the training dataset and thus will be detected as anomalies. To tackle this issue, we propose to present an existing iterative algorithm, which presents potential anomaly to the expert at each iteration, and compute a new boundary according to this feedback using One Class Support Vector Machine.**

***Index Terms*—Machine learning, Semi-supervised Learning, Active Learning, User Feedback, Anomaly Detection, One-Class Support Vector Machines**

I. INTRODUCTION

Machine learning (ML) has gained a lot of attention in the past few years, due to the evolution of computer calculation and stocking capability. ML algorithms can be divided into two classes devoted to regression (estimation of parameters) or classification (labeling data), which have received an increasing interest in various applications [1]. Anomaly detection (AD) is generally considered as a particular classification problem consisting in classifying data that are not generated by some nominal process [2], [3]. Most AD techniques are unsupervised in the sense that the data are fed to the algorithms without any label, which then selects a subset of the data that are classified as anomalies. Most AD rules compute a score for each data vector and declare that this vector is an anomaly if the score is below a given threshold (otherwise the vector is declared as nominal). In other words, anomalies are considered as isolated data vectors compared to the whole dataset and can thus be isolated. Among all the existing techniques, one can mention the Local Outlier Factor (LOF) [4] and the Local Outlier Probability (LoOP) [5], which compute a local density score based on the nearest neighbors and their relative distances to a given data vector. Another famous algorithm is Isolation Forest (IF) [6], which computes random trees [7] until the tested point is isolated. The mean number of nodes required to isolate an anomaly should be much smaller than for nominal points, which should be more dense. Finally, other popular AD methods include the Support Vector Data Description (SVDD) [8] and the One Class Support Vector Machine (OCSVM) [9], finding an appropriate hypersphere or hyperplane (using some possible kernel mapping) isolating a given maximal proportion of data from the others.

A recurrent problem with AD algorithms is that they often raise false alarms, i.e., a lot of nominal data are often estimated as anomalies. This is due to the fact that real systems have multi-modal densities, and some of the modes correspond to nominal data that are rarely observed. This is for instance the case when a certain request is made by the user. As these events are rare, they might not be even included in the training database or with very few instances. To overcome this problem, an expert is generally obliged to check which of the detected anomalies are actual and those that have been misclassified. This operation will be referred to as "user feedback" in this paper, and is also referred to as Active Learning [10].

Anomaly Active Discovery (AAD) [11] is one way to include user feedback into linear scoring problems. This paper shows how this feedback can be included into the OCSVM using AAD. Note that active learning has already been applied to SVM [12]. Section II briefly recalls the principle of OCSVM and provides some implementation details. Section III shows how expert feedback can be included in the standard OCSVM algorithm. More precisely, Section III-A shows how labeled anomalies and nominal data can be included in the boundary estimation of OCSVM. Section III-B recalls the principle of AAD and shows how it can be combined with OCSVM. Some experiments are conducted In Section IV, assessing the performance of the proposed algorithms on various datasets, with comparison to some benchmark methods. Finally, Section V summarizes the paper contributions and give some perspectives about this work.

II. ONE-CLASS SUPPORT VECTOR MACHINES

OCSVM has been introduced in [9] adapting the support vector machine classifier to AD. Consider N training data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \in \mathcal{X}$, and $\Phi : \mathcal{X} \to \mathcal{F}$ a feature map from the observation space $\mathcal{X}$ into an inner product space $\mathcal{F}$ such that its inner product can be computed using a kernel k [13]

$$\Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{1}$$

There are many ways of constructing kernels for classification or AD [14, Chap. 2.3]. This paper will focus on the well known Gaussian kernel defined as

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right), \tag{2}$$

where $\|.\|$ is a norm on $\mathcal{X}$ (this work considers the ℓ_2 norm $\|\boldsymbol{x}\|_2^2 = \boldsymbol{x}^T\boldsymbol{x}$) and $\gamma > 0$ is a unique hyperparameter (other kernels could be considered with minor modifications. An advantage of using the Gaussian kernel is that many heuristics have been proposed to estimate the hyperparameter γ, such as the median of the distances between the nominal data [15]. OCSVM consists in finding the hyperplane in the feature space $\mathcal{F}$ separating the training data located as far as possible from the origin, with a given tolerance ν, representing the maximal proportion of training data located outside the separating hyperplane. This problem can be formulated as follows

$$\begin{aligned} &\arg\min_{\boldsymbol{w},\rho} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \rho + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i \\ &\quad \text{s.t. } \boldsymbol{w}^T\Phi(\boldsymbol{x}_i) \geq \rho - \xi_i \quad \forall i = 1, \ldots, N \\ &\quad\quad \xi_i \geq 0 \quad \forall i = 1, \ldots, N. \end{aligned} \tag{3}$$

The dual problem of (3) can easily be solved using quadratic programming [9]. More details on this dual problem will be given in Section III-A.

III. Incorporating Expert Feedback

This section explains how expert feedback can be included in OCSVM. First, an extension of the original problem with some labeled data is introduced. The principle of AAD, introduced in [11] and applied to IF in [16], is then presented. AAD allows us to incorporate some expert knowledge about some data vectors, which can be nominal data or anomalies.

A. OCSVM with some labeled data

This section studies a OCSVM method with Expert Feedback (OCSVMEF). To take into account false positives and false negatives in OCSVM, we assume that we have some labels (corresponding to anomalies or nominal data) for a subset of the data, leading to a semi-supervised framework. The optimization problem (3) can be modified to take into account these labels, inspired by SVDD with negative examples [8]

$$\begin{aligned} \arg\min_{\boldsymbol{w},\rho,\boldsymbol{\xi},\boldsymbol{\zeta},\boldsymbol{\tau}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \rho + p(\boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\tau}) \\ \text{s.t. } & \boldsymbol{w}^T\Phi(\boldsymbol{x}_i) \geq \rho - \xi_i, \quad i = 1, \ldots, N_1 \\ & \boldsymbol{w}^T\Phi(\boldsymbol{y}_l) \geq \rho - \zeta_l, \quad l = 1, \ldots, N_2 \\ & \boldsymbol{w}^T\Phi(\boldsymbol{z}_p) \leq \rho + \tau_p, \quad p = 1, \ldots, N_3 \\ & \xi_i \geq 0, \quad i = 1, \ldots, N_1 \\ & \zeta_i \geq 0, \quad l = 1, \ldots, N_2 \\ & \tau_p \geq 0, \quad p = 1, \ldots, N_3 \end{aligned} \tag{4}$$

where $p(\boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\tau})$ is the following penalty

$$p(\boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{\tau}) = \frac{1}{\nu_1 N_1}\sum_{i=1}^{N_1}\xi_i + \frac{1}{\nu_2 N_2}\sum_{l=1}^{N_2}\zeta_l + \frac{1}{\nu_3 N_3}\sum_{p=1}^{N_3}\tau_p \tag{5}$$

and

- N_2 is the number of false positives labeled by the expert
- N_3 is the number of false negatives labeled by the expert
- $N_1 = N - N_2 - N_3$ is the number of non-labeled data
- $\boldsymbol{w}, \rho$ are the parameters of the hyperplane separating nominal data from anomalies in the feature space
- ν_1 is the maximal proportion of non-labeled training data outside the separating hyperplane
- ν_2 is the maximal proportion of mislabeled false positives (we should have $\nu_2 \ll \nu_1$ since the data labeled by the expert are more likely to have the correct labels
- ν_3 is the maximal proportion of mislabeled false negatives (with $\nu_3 \ll \nu_1$)
- $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1,\ldots,N_1}$ contains the non-labeled training data
- $\boldsymbol{Y} = \{\boldsymbol{y}_l\}_{l=1,\ldots,N_2}$ is the set of labeled false positives
- $\boldsymbol{Z} = \{\boldsymbol{z}_p\}_{p=1,\ldots,N_3}$ is the set of labeled false negatives
- $\boldsymbol{\xi} = \{\xi_i\}_{i=1,\ldots,N_1}$ are the slack variables for non-labeled training data
- $\boldsymbol{\zeta} = \{\zeta_l\}_{l=1,\ldots,N_2}$ are the slack variables for labeled false positives
- $\boldsymbol{\tau} = \{\tau_p\}_{p=1,\ldots,N_3}$ are the slack variables for labeled false negatives.

The dual problem of 4 can be classically expressed as [17]

$$\begin{aligned} &\arg\min_{\boldsymbol{\kappa}} \frac{1}{2}\boldsymbol{\kappa}^T\boldsymbol{K}\boldsymbol{\kappa} \\ &\quad \text{s.t. } \sum_{i=1}^{N_1}\alpha_i + \sum_{l=1}^{N_2}\beta_l - \sum_{p=1}^{N_3}\gamma_p = 1 \\ &\quad 0 \leq \alpha_i \leq \frac{1}{\nu_1 N_1}, \quad i = 1, \ldots, N_1 \\ &\quad 0 \leq \beta_l \leq \frac{1}{\nu_2 N_2}, \quad l = 1, \ldots, N_2 \\ &\quad 0 \leq \gamma_p \leq \frac{1}{\nu_3 N_3}, \quad p = 1, \ldots, N_3 \end{aligned} \tag{6}$$

which can be solved using quadratic programming. Moreover, the decision rule for a vector $\boldsymbol{x}$ is defined by

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{i\in \text{SV}}\kappa_i k(\boldsymbol{x}_i, \boldsymbol{x}) - \rho\right) \tag{7}$$

where $+1$ refers to a nominal data, -1 to an anomaly, SV is the set of support vectors learnt by the algorithm, i.e., the vectors for which the parameters κ_i are non zero. Finally, to compute ρ, we can notice that constraints in (4) become equalities if the Lagrange multipliers are nonzero, i.e., if $0 < \alpha_i < \frac{1}{\nu_1 N_1}$, $0 < \beta_l < \frac{1}{\nu_2 N_2}$ and $0 < \gamma_p < \frac{1}{\nu_3 N_3}$. Thus, ρ can be determined using the vectors satisfying one of the previous strict inequalities, i.e., such that $\sum_{i\in \text{SV}}\kappa_i k(\boldsymbol{x}_i, \boldsymbol{x}) = \rho$.

B. AAD Formulation

AAD has already been exploited to incorporate expert feedback in anomaly detection when the anomaly score is linear, i.e., when there is a vector $\boldsymbol{w}$ such that the score of any vector $\boldsymbol{x}$ is $\boldsymbol{w}^T\boldsymbol{x}$ [11], [16]. AAD is an iterative algorithm presenting data vectors to an expert who should declare if this vector is nominal or an anomaly, until some budget B of queries has been met. The principle of AAD is summarized in Algorithm 1.

Algorithm 1 AAD

```
1:  Initialization: $\boldsymbol{w} = \boldsymbol{w}_0$, $t = 0$, $N_2 = N_3 = 0$
2:  while $t \leq B$ do
3:      $t = t + 1$
4:      $\boldsymbol{x}_i = \arg\max_{\boldsymbol{x}} s(\boldsymbol{x})$ (data with maximum anomaly score)
5:      if $\boldsymbol{x}_i$ is an anomaly then
6:          $N_3 \leftarrow N_3 + 1$ one more anomaly is labeled
7:          $\boldsymbol{z}_{N_3} = \boldsymbol{x}_i$
8:      else
9:          $N_2 \leftarrow N_2 + 1$ one more anomaly is labeled
10:         $\boldsymbol{y}_{N_2} = \boldsymbol{x}_i$
11:     end if
12:     $N_1 =\leftarrow N_1 - 1$ remove $\boldsymbol{x}_i$ from unlabeled data
13:     $\boldsymbol{x}_i \leftarrow \{\}$
14:     Update $\boldsymbol{w}^{(t)}$ according to the user feedback
15: end while
```

The main differences between the approach investigated in this paper and that of [16] are 1) the initialization of the vector $\boldsymbol{w}_0$ (which result from OCSVM in this work and from IF in [16]), and 2) Line 14 of Algorithm 1 devoted to the update of vector $\boldsymbol{w}$. The Accuracy to The Top (AATP) approach was proposed in [18] to update $\boldsymbol{w}$. AATP requires to define a proportion τ of vectors corresponding to anomalies. The anomaly score $\boldsymbol{w}^T\boldsymbol{x}_i$ are calculated for each vector $\boldsymbol{x}_i$. The scores are then sorted in descending order and the top τ-th quantile of these scores is denoted as q_τ. A data vector is declared as an anomaly when its anomaly score is higher than q_τ. To include expert feedback, AAD selects the unlabeled vector with the highest anomaly score, send it to the user and re-estimates the vector $\boldsymbol{w}$. The re-estimation of $\boldsymbol{w}$ in [16] was conducted using a misclassification cost based on the so-called hinge loss function

$$\ell(q_\tau, \boldsymbol{w}; \boldsymbol{x}_i, y_i) =$$
$$\begin{cases} 0 & \text{if } \boldsymbol{w}^T\boldsymbol{x}_i \geq q_\tau \text{ and } y_i = \text{“anomaly”} \\ 0 & \text{if } \boldsymbol{w}^T\boldsymbol{x}_i < q_\tau \text{ and } y_i = \text{“nominal”} \\ (q_\tau - \boldsymbol{w}^T\boldsymbol{x}_i) & \text{if } \boldsymbol{w}^T\boldsymbol{x}_i < q_\tau \text{ and } y_i = \text{“anomaly”} \\ (\boldsymbol{w}^T\boldsymbol{x}_i - q_\tau) & \text{if } \boldsymbol{w}^T\boldsymbol{x}_i \geq q_\tau \text{ and } y_i = \text{‘nominal”.} \end{cases} \tag{8}$$

Note that this loss function is 0 if a data vector is correctly classified, and strictly positive when it is misclassified. The main contribution of this paper is to propose another update rule for $\boldsymbol{w}$, which is more adapted to OCSVM.

C. Reweighting OCSVM Using AAD

Looking carefully at the OCSVM decision function defined in (7), one can observe that $s(\boldsymbol{x}) = -\sum_{i\in\text{SV}} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x})$ can be considered as an anomaly score, since the data that are more likely to be anomalies are those associated with high values of $s(\boldsymbol{x})$. Since the solution of (4) is a linear combination of all support vectors, the score function can be written as a linear function of $\boldsymbol{w}$, i.e., $s(\boldsymbol{x}) = \boldsymbol{w}^T\Phi(\boldsymbol{x})$, which can be optimized using the results of [16]. Note that this formulation is given to show the link with AAD [16]. However, $\boldsymbol{w}$ will never be used explicitly due to the kernel trick.

Algorithm 1 iteratively updates the weight vector $\boldsymbol{w}$, based on the maximum number of requests to the expert (denoted as B). It ranks unlabeled vectors according to their anomaly score, and presents to the expert the vector with the highest score. According to its feedback, a new decision bound is calculated. To optimize $\boldsymbol{w}$ based on the user feedback, we propose to use the approach presented in Section III-A and to solve problem (4) instead of using the hinge loss function (8).

IV. Experiments

Experiments[1] were conducted on several datasets presented in [16], which are summarized in Table I, and available on the author webpage[2]. All datasets are from the UCI repository [19], except *Mammography* from [20]. The datasets were divided into two subsets defining the “nominal” and “anomaly” classes (the subset containing the maximum number of vectors was defined as the nominal class whereas the other subset contains the anomalies). As explained in [16], for the *Cardiotocography* dataset, the anomaly instances were downsampled to represent around 2% of the data.

In all our experiments, the parameters for the OCSVM algorithms were set to $\nu = \nu_1 = 0.03$ (same value in presence or absence of expert feedback) and $\nu_2 = 0, \nu_3 = 0$ (100% confidence in the expert feedback, which is equivalent to get rid of any upper bound for β_l and γ_p when solving (6)). The parameter γ in the RBF kernel was set at each iteration as the median of the ℓ_2 norms between all pairs of points in the dataset (as in [15]), except the anomalies that have been confirmed by the user feedback.

To illustrate the effects of user feedback, we use the 2D synthetic dataset *toy2* from [16]. The influence of user feedback on the decision boundary is illustrated in Fig.1. As one can see, during the first iteration, the point located in the bottom left of the figure, which is the furthest from the OCSVM boundary, is presented to the expert. The expert gives its feedback about this point and mentions that it is a nominal data. Thus after the first iteration, this point is now included within the new boundary, and so on and so forth, until the maximal number of expert queries has been reached.

Another evaluation suggested in [16] was to count the number of actual anomalies that are proposed to the user for its feedback. Indeed, the algorithm should be able to identify as many anomalies as possible (and the only confirmations come from the data sent to the expert). To evaluate this criterion, the authors of [16] plotted the number of anomalies returned by the expert versus the number of queries. In the ideal case, this plot should be a straight line with slope 1, meaning that each query made to the expert corresponds to

[1] the Matlab code is available at http://perso.tesa.prd.fr/jlesouple/codes.html

[2] https://github.com/shubhomoydas/ad_examples/tree/master/ad_examples/datasets/anomaly

TABLE I
Datasets used in our experiments with their characteristics.

Name	**Nominal classes**	**Anomaly classes**	**Number of data**	**Number of features**	**Number of anomalies**
Abalone	8,9,10	3,21	1920	9	29 (1.5%)
ANN-Thyroid-1v3	3	1	3251	21	73 (2.25%)
Cardiotocography	1	3	1700	2	45 (2.65%)
Mammography	-1	1	11183	6	260 (2.32%)
Shuttle	1	2, 3, 5, 6, 7	12345	9	867 (7.02%)
Yeast	*CYT*, *NUC*, *MIT*	*ERL*, *POX*, *VAC*	1191	8	55 (4.6%)

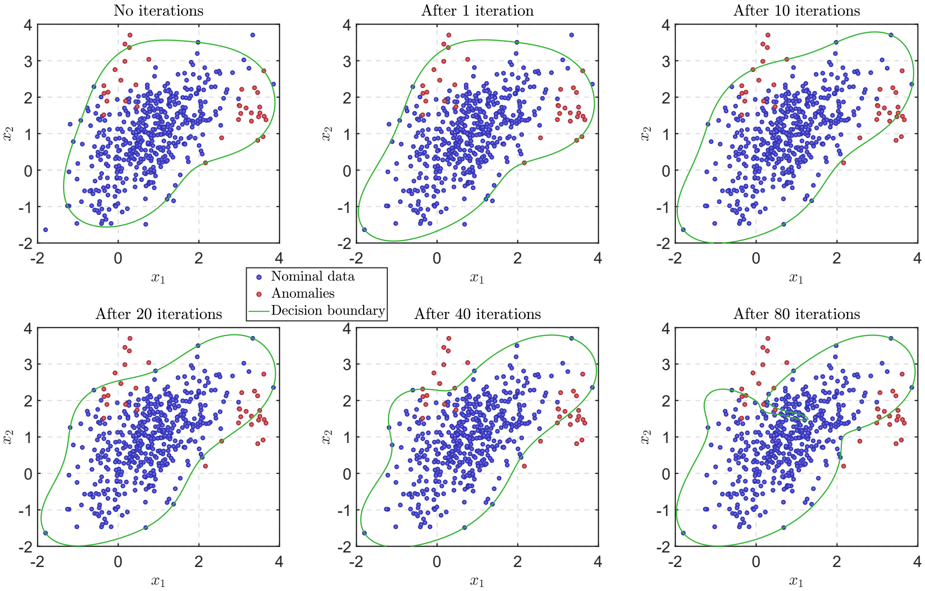

Fig. 1. Illustration of the expert feedback influence on the boundary decision after a given number of queries. At each iteration, the furthest data from the boundary is presented to the expert who gives feedback about its nature (anomaly or nominal). The algorithm then computes a new boundary according to this new feedback and the previous ones.

an anomaly. The proposed method (referred to as OCSVM-AAD) is compared to OCSVM, IF and IF-AAD in Fig. 2. For algorithms without AAD, this is equivalent to the cumulative sum of true negatives when ranking examples in decreasing order of anomaly score. A global remark is that AAD always improves the results of its counterpart, which does not use user feedback, as the OCSVM-AAD and IF-AAD curves are above those of OCSVM and IF. Moreover, OCSVM-AAD and IF-AAD seem to give equivalent results, except for *ANN-Thyroid-1V3* and *Shuttle*, where IF-AAD outperforms the proposed method.

Finally, it is interesting to analyze the effect of the proposed method on the number of support vectors when compared to OCSVM. This effect can be observed in Fig.3, showing t-SNE plots [16], [21] for the datasets *Abalone* and *ANN-Thyroid-1v3*, and indicating the numbers of support vectors in each figure. The number of support vectors obtained using OCSVM-AAD is smaller than that of OCSVM, which is an interesting property of the proposed algorithm.

V. Conclusion

This paper presented a new iterative anomaly detection algorithm using one-class support vector machines (OCSVM) with expert feedback. At each step of the algorithm, the data that is the most likely to be an anomaly is presented to the expert, who gives his feedback, in order to recompute the OCSVM decision function. This operation is conducted until some budget (maximal number of queries to the expert) has been reached. This iterative scheme resulting from active anomaly detection is the main contribution of this paper mixing one-class SVM and active anomaly detection ideas. Future work will include the study of other sequential anomaly detection rules with user feedback.

Acknowledgment

The authors would like to thank Thales Alenia Space for funding part of this work, especially Cédric Baudoin and Laura Lumale for fruitful discussions about potential applications to satellite data.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[2] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A Review of Novelty Detection," *Signal Processing*, vol. 99, pp. 215–249, June 2014.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, July 2009.

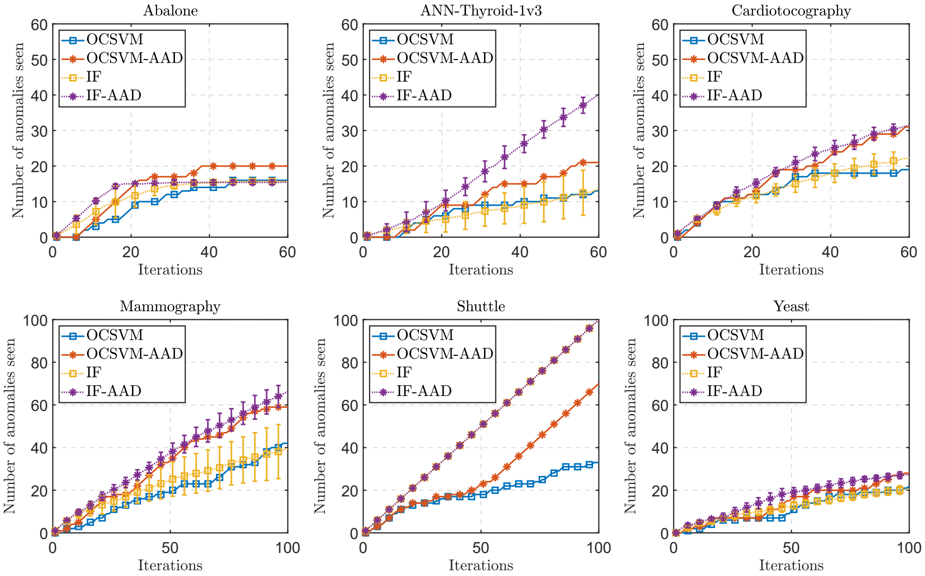

Fig. 2. Total number of true anomalies presented to the expert versus the number of queries for various datasets for both IF and OCSVM with and without AAD. The total number of queries is limited to 60 for small datasets (*Abalone*, *ANN-Thyroid-v3* and *Cardiotocography*), and 100 to big datasets (*Mammography*, *Shuttle* and *Yeast*). For IF, results were averaged over 10 Monte-Carlo runs and are displayed with their $\pm\sigma$ variations. Note that for OCSVM-based methods, the boundaries are deterministic, contrary to IF-based algorithms which generate trees randomly.

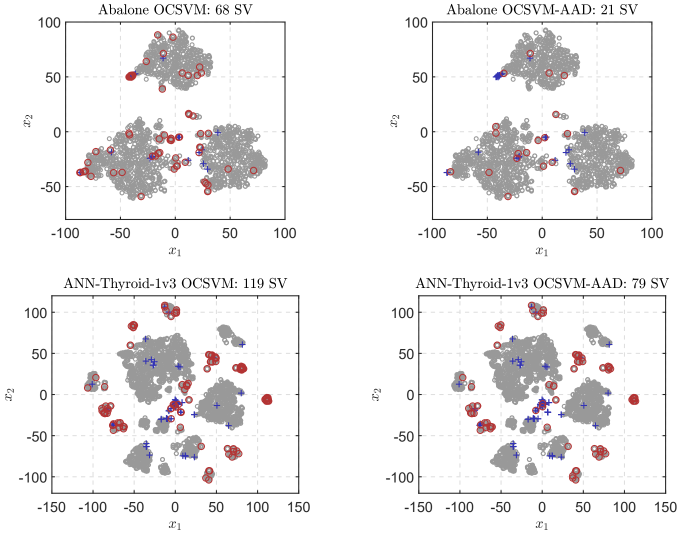

Fig. 3. 2D visualization of *Abalone* and *ANN-Thyroid-1v3* datasets using t-SNE. Blue plus markers represents anomalies whereas small grey circles represent nominal data. Big red circles represents the support vectors of the learnt decision boundary.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," in *Proc. Int. Conf. on Management of Data (SIGMOD)*, Dallas, Tx, May 2000, pp. 93–104.

[5] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: Local Outlier Probabilities," in *Proc. Int. Conf. on Information and Knowledge Management (CIKM)*, Hong-Kong, China, Nov. 2009, pp. 1649–1652.

[6] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. Int. Conf. on Data Mining (ICDM)*, Pisa, Italy, Dec. 2008, pp. 413–422.

[7] L. Breiman, "Random Forests," *Machine Learning*, vol. 05, no. 32, pp. 5–32, 2001.

[8] D. M. Tax and R. P. Duin, "Support Vector Data Description," *Machine Learning*, no. 54, pp. 45–66, Jan. 2004.

[9] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, July 2001.

[10] D. Cohn, L. Atlas, and R. Ladner, "Improving Generalization with Active Learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, May 1994.

[11] S. Das, W.-K. Wong, T. G. Dietterich, A. Fern, and A. Emmott, "Incorporating Expert Feedback into Active Anomaly Discovery," in *Proc. Int. Conf. on Data Mining (ICDM)*, Barcelona, Spain, Dec. 2016, pp. 853–858.

[12] P. Schlachter and B. Yang, "Active Learning for One-Class Classification Using Two One-Class Classifiers," in *Proc. Eur. Conf. on Signal Processing (EUSIPCO)*, Rome, Italy, Sept. 2018, pp. 1197–1201.

[13] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proc. Workshop on Computational Learning Theory*, Pittsburgh, PA, July 1992, pp. 144–152.

[14] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* Cambridge, MA: The MIT Press, 2002.

[15] T. Jaakkola, M. Diekhans, and D. Haussler, "Using the Fisher Kernel Method to Detect Remote Protein Homologies," in *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, Heidelberg, Germany, Aug. 1999, pp. 149–158.

[16] S. Das, W.-K. Wong, A. Fern, T. G. Dietterich, and M. A. Siddiqui, "Incorporating Feedback into Tree-based Anomaly Detection," in *Proc. Workshop on Interactive Data Exploration and Analytics (IDEA)*, Halifax, Canada, Aug. 2017, pp. 25–33.

[17] J. Lesouple and J.-Y. Tourneret, "Incoporating User Feedback Into One-Class Support Vector Machines Anomaly Detection - Supplementary Material," TéSA Laboratory, Toulouse, France, Tech. Rep., Aug. 2020. [Online]. Available: http://perso.tesa.prd.fr/jlesouple/documents/TReusipco2020.pdf

[18] S. Boyd, C. Cortes, M. Mohri, and A. Radovanovic, "Accuracy at the Top," in *Proc. Advances in Neural Information Processing System 25*, Lake Tahoe, NV, Dec. 2012, pp. 953–961.

[19] (2007) The UC Irvine Machine Learning Repository. [Online]. Available: http://archive.ics.uci.edu/ml/

[20] K. S. Woods, C. C. Doss, K. W. Bowyer, J. L. Solka, C. E. Priebe, and W. P. Kegelmeyer, "Comparative Evaluation of Pattern Recognition Techniques for Detection of Microcalcifications in Mammography," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 07, no. 06, pp. 1417–1436, 1993.

[21] L. V. D. Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, no. 32, pp. 2579–2605, 2008.