Novelty Detection Based on Genuine Normal and Artificially Generated Novelty Examples

George Gomes Cabral
Statistics and Informatics Department
Federal Rural University of Pernambuco
Dom Manoel de Medeiros St., 52171-900, Recife, Brazil
george.gcabral@ufrpe.br

Adriano Lorena Inácio de Oliveira
Center for Informatics
Federal University of Pernambuco
Prof. Moraes Rego Av., 50670-901, Recife, Brazil
alio@cin.ufrpe.br

***Abstract*—One-class classification (OCC) is an important problem with applications in several different areas such as outlier detection and machine monitoring. Since in OCC there are no examples of the novelty class, the description generated may be a tight or a bulky description. Both cases are undesirable. In order to create a proper description, the presence of examples of the novelty class is very important. However, such examples may be rare or absent during the modeling phase. In these cases, the artificial generation of novelty samples may overcome this limitation. In this work it is proposed a two steps approach for generating artificial novelty examples in order to guide the parameter optimization process. The results show that the adopted approach has shown to be competitive with the results achieved when using real (genuine) novelty samples.**

I. Introduction

In many pattern recognition problems, there are no explicit rules to distinguish objects belonging to different classes; however, samples of these objects may easily be gathered. In these cases, the problem is solved by creating a model of classifier from a limited set of training samples. The goal is to obtain models of classifiers able to correctly predict the classes of objects unknown during the modeling phase.

Regarding novelty detection problems, an object of the novelty class can be defined as one that does not resemble any object presented to the classifier during the training phase. As an example, suppose a model of a one-class classifier built aimed at recognizing dogs, cats, birds and bears. If any of these animals is presented to this model, the expected outcome is normal, nevertheless, if an unknown animal (e.g., a rabbit) is presented to this model, the expected outcome is novelty. One of the shortcomings of novelty detection methods based on Signature Detection [19] is the need samples of the novelty class during the training phase. A previous work has already advocated that the usage of samples of the novelty class for modeling the novelty distribution may not be adequate since these samples are rare and may not represent well the whole novelty distribution [4]. So, it is important to investigate methods that use only normal data during the training phase, this is the case of the One-class Classification paradigm [11] [12] [6] [3].

During the modeling phase, the parameters of most of the algorithms for pattern recognition need to be chosen such that the final model perform well for real cases. Most of the works where the parameters are automatically adjusted use some type of optimization method such as Genetic Algorithms (GA) or Particle Swarm Optimization (PSO) [15] [16] [17]. In [17], Yongqi proposes a hybrid complex particle swarm optimization algorithm to tune the parameters of a Least Squares Support Vector Machine (LSSVM). In [15], Chou *et. al.* integrated Genetic Algorithms with SVM classifiers to solve the problem of predicting the risk of Public Private Partnership (PPP) which is a financial strategy for stimulating private investments in public works. Both works try to optimize the classifier parameters (in this case, a SVM) by using optimization methods based on local search. In these cases, the search for the optimal parameters is performed considering an objective function that indicates whether or not the method is performing well for all the classes.

A major issue for OCC methods is to obtain valid samples of the novelty class in order to efficiently tuning the classifiers parameters. These samples are rare and may not be significant, as aforementioned. So, in a scenario where there are no samples of the novelty class it is not easy to check whether or not the model is able to fit the normal data rejecting unknown objects as well. In this case, the adjustment of the classifier parameters may lead to three cases: (i) an overfitting of the training set (i.e., the model is highly adjusted to the normal samples) - (Figure 1.a); (ii) a model properly adjusted to the training set - (Figure 1.b); and (iii) a large model (i.e., the model underfits the normal data) - (Figure 1.c).

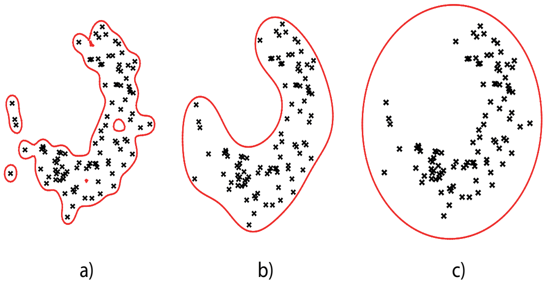

Fig. 1. Examples of one-class models with different degrees of adjustment to the normal data. a) - overffiting model, b) - proper model and c) - underfitting model.

One-class classifiers do not need novelty samples during the training phase, however, these samples are important to validate the model. Therefore, this work proposes a methodology for artificially generating data from the novelty class (positive class) such that this data can be added to the validation set for guiding the parameter optimization process. So, the validation set will contain genuine normal data and artificial novelty data (diferently from [3], where the validation set was formed only by genuine samples) such that the optimal parameters are the ones that produce the model with best accuracy for the validation set. Notice that in many problems the validation using only genuine novelty samples (e.g., from [3]) cannot be applied. This occurs in problems where there is no novelty samples available in the training phase, or these samples are very rare. For this work, the methods One-class SVM (OCSVM) [2] and the method Feature Boundaries Detector for OCC (FBDOCC) [3] were applied. Notice that the classifier One-class Random Forest [1] was not employed in this work because it didn't obtained good results in [3].

This paper is organized as follows: Section II briefly discusses the novelty detection problem and the computational methods employed in this work; Section III presents the proposed methodology for generating novelty data; Section IV discusses the experiments and results; and Section V presents the conclusion and future works.

II. Materials and Methods

In multi-class classification, data from two or more classes are available during the modeling phase and the decision boundary is supported by examples of each class. On the other hand, many problems such as machine monitoring and medical diagnosis may have a lot of normal data. Yet, usually it is very expensive to obtain data from an abnormal behavior of these monitored systems. For instance, machine fault examples may not exist and samples of the occurrence of some diseases may be rare. In such cases, the natural approach is to build a description of the normal data (i.e., a boundary surrounding the objects that represents the normal events and that are available during the modeling phase). Subsequently, abnormal events are detected when an event lies outside this description of the normal data. This is the basic concept of One-Class Classification.

It is important to point out that the novelty detection task is highly related to OCC, however, it can also be performed by methods that use examples of the novelty class during the modeling phase [19]. In [18], a survey of novelty detection is provided whereby several techniques are presented along with their advantages and disadvantages. These techniques comprise one-class and signature based classifiers[19] (where information regarding the novelty class is provided during the training phase), yet, they share the same purpose, namely to detect unusual objects (novelty detection).

This Section provides basic information about the classifiers used for novelty detection and the optimization method used for tuning the parameters of these classifiers.

A. Methods For Novelty Detection

With the aim of assessing the performance of the proposed approach for generating artificial samples, two one-class classification methods were employed: One-class SVM (OCSVM) [2] and Feature Boundaries Detector for OCC (FBDOCC) [3].

1) One-class SVM: Based on SVMs framework, Scholkopf *et. al.* [2] have proposed the one-class SVM (OCSVM). In OCSVM, the kernel function maps the training objects to a feature space. In the feature space, OCSVM then recognizes the origin as the only member of the second class (the novelty class). In contrast to the SVDD (Support Vector Data Description) [6] which tries to find the less bulky hyper sphere which contains almost all of the training objects, the OCSVM tries to find the hyper plane which separates the training data with maximal distance from the origin in the feature space. The goal is to maximize the margin of separation to the origin. As in the multi-class SVM, slack variables denoted by $\vec{\xi_i}$ enable some training objects to fall outside the side of the hyper plane which represents the normal class (i.e., misclassifies some training objects). A training sample is a support vector when it is misclassified or falls inside the hyper plane. When using a non-linear kernel such as a Gaussian function, both methods (One-class SVM and SVDD) are equivalent [5].

2) Feature Boundaries Detector for OCC: In [3], Cabral and Oliveira introduced the novelty detection method Feature Boundaries Detector for One-class Classification (FBDOCC). The underlying intuition is to explore all feature dimensions of the problem for each instance in the training set (which contains only instances of the normal class) in order to find the best fitting boundaries for encompassing the normal data distribution. Adopting the Euclidean space, the FBDOCC generates $2l$ new artificial prototypes for each training instance t_i in a relatively small distance (defined by the parameter r) to the respective training instance. Each artificial prototype p_j is aimed at representing a piece of the limit between the normal and the abnormal classes. The idea is to generate one hyper sphere with radius th for each prototype p_j and check whether or not there is any training instance (except the instance which generated the prototype) inside this hyper sphere. If all the training instances t_i are located outside the hyper sphere defined by p_j, then: (i) information about p_j is stored in order to reproduce it as a positive prototype in the test phase and (ii) the training instance which generated this prototype is added to the set of negative instances. Once information of one of the $2l$ artificial prototypes is stored, the remaining prototypes for the current training instance are discarded. The positive prototypes define the novelty class whereas the set of negative prototypes represents the normal class.

For further information, please refer to [3].

B. Local Search For Finding the Optimal Parameters

In this work, the search method Particle Swarm Optimization (PSO) [20] was employed to conduct the search for the optimum parameter set of each OCC classifier (OCSVM and FBDOCC). This technique is usually aimed at finding

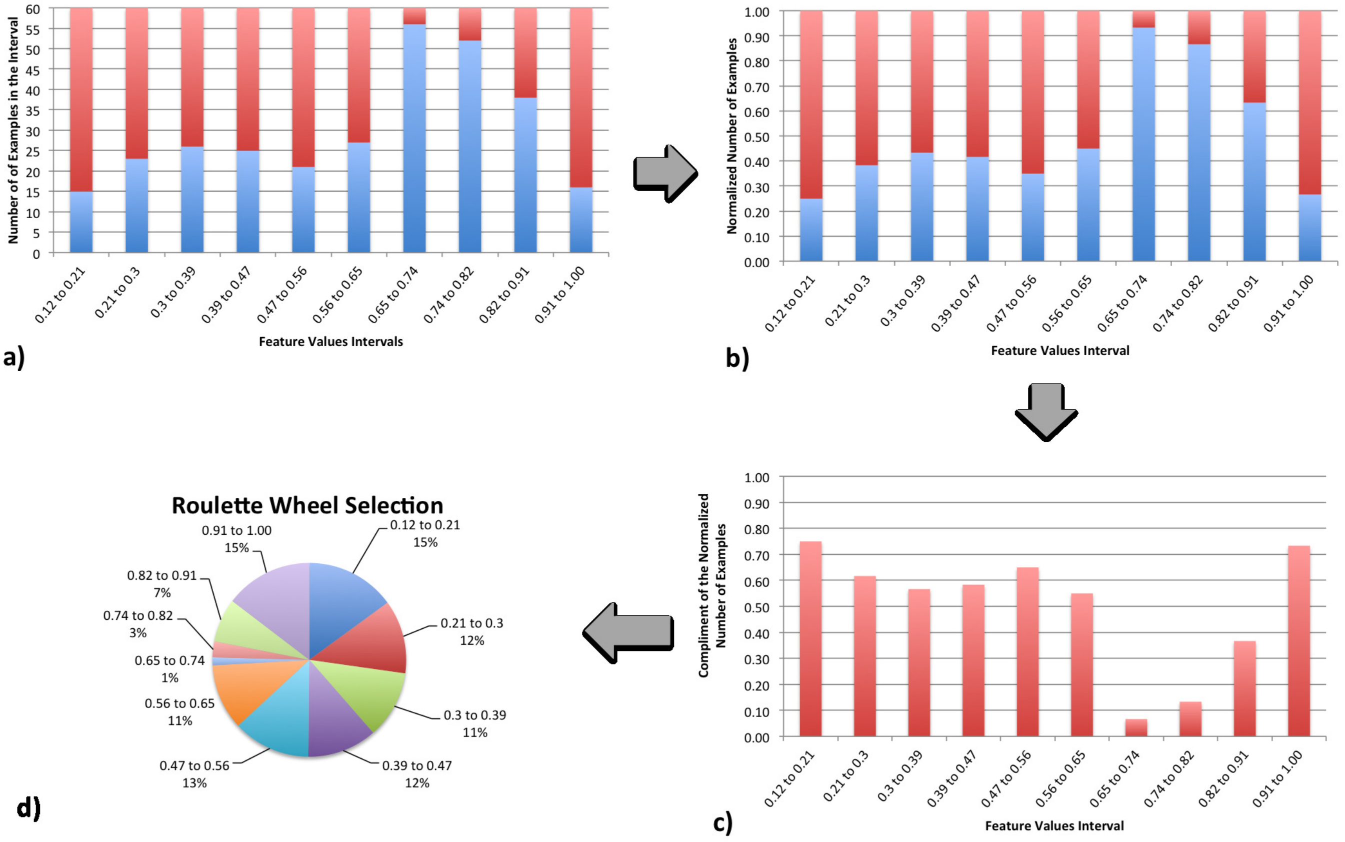

Fig. 2. Procedure to build artificial samples reducing the overlapping between the classes. a) the histogram of the frequencies of values for the considered feature; b) the same histogram of a), however with the frequencies scaled between 0 and 1; c) the complementary histogram of b); and c) the roulette used in the roulette wheel selection.

optimal solutions for non-linear problems. Inspired by the social behavior of a group of birds, the main intuition behind the PSO is to build a set (population) of particles to simulate the movements performed by the birds while searching for food in a specific region. PSO explores the social behavior of an intelligent set of individuals along with their ability to communicate to find a global solution. Each PSO particle represents a solution (model) in an n-dimensional space, where n stands for the number of parameters to be optimized.

In the standard implementation of the PSO, particles move inside the multi-dimensional search space using a combination of attraction to the best solution found by this individual particle and an attraction to the best particle belonging to its neighborhood [23] [21]. A neighborhood is defined as a subset of the swarm whose particle is able to establish communication. The swarm moves in the search space by updating the velocity and position of each particle [22].

III. Proposed Approach

This work proposes the use of a methodology to generate artificial samples of the novelty class so that these samples tightly encompass the normal data.

The generation of the artificial novelty samples is conducted in two phases:

1) Phase 1 - Generation of novelty samples based on the histogram of the values for each feature of the problem; and
2) Phase 2 - Removing the artificial samples nearer than a given threshold to the normal samples;

The first phase tries to generate artificial samples in regions where the normal samples are absent, however, this phase does not extinguish the occurrence of novelty samples in normal regions. The second phase removes these misplaced samples. Following, each phase of the approach is detailed.

A. Phase 1

This phase of the methodology was proposed by Désir et al. in [1]. In [1], the authors build artificial samples of the novelty class in four steps (depicted in Figure 2). Notice that these phases are executed for one feature of the problem per time.

Initially, a histogram of the values of all the normal data for one feature is created. The number of bins of the histogram can be chosen as in [1]. The blue bar represents the number of normal samples (frequency) that lie in the bin interval whereas the red bar represents its compliment (considering the upper bound as the highest frequency plus 10%). This is not a critical parameter. This step is depicted in Figure 2.a).

Figure 2.b) depicts the second step of this phase. This step consists in normalizing the frequencies (obtained in the previous step) between the range 0 and 1.

In the third step, Figure 2.c), the complementary histogram of the second step is obtained. This histogram is used to obtain the probability of an artificial value fall in a range of values.

The fourth step consists in building a pie chart where each slice represents a bin in the complementary histogram. The summation of the probabilities of all the bins shown in Figure 2.c) exceeds 1, so, all the probabilities are scaled such that their summation is equal to 1. In order to generate an artificial value, a random number belonging to the uniform distribution, between 0 and 1, is then generated and the roulette wheel selection method is used to pick the range in which the value must be in.

This roulette is used to generate as many values of one feature as needed. In order to build new artificial samples, and let m be the number of features, m roulettes must be generated and the values returned by these roulettes must be combined in order to generate a new sample.

Notice that, this method does not eliminate the occurrence of novelty samples wrongly inside the normal distribution, however, the number of such samples is considerably less in comparison to a random generation. In other words, the approach decreases the overlapping rate.

B. Phase 2

In this phase, the minimal distance (threshold) by which an artificial sample must be nearer to a nearest neighbor normal sample is found. To this aim, for each normal sample, the distance to its respective nearest neighbor is computed and stored in a vector. So, if the dataset contains 100 samples, 100 distances will form a distribution of distances that can be considered as belonging to a Gaussian distribution. So, in this distribution, the particular distance where the cumulative probability exceeds 95% is used as the threshold. The percentage 95% was used for all the experiments and was shown to be a non critical parameter.

Figure 3 shows an example where a CDF was built and the red line shows that around 95% of the distances are less than 0.04. So, this value can be used as a threshold to remove artificial novelty samples whose distances to the nearest neighbor in the normal class are less than this threshold.

Once the two phases were carried out, the final dataset is supposed to be free of class overlapping. The Figure 4.a) shows the data after phase 1. In this case, overlapping samples can be seen. Figure 4.b) shows the result of second phase. The dataset is now a two-class dataset with non-overlapping classes.

Since the methods considered for our experiments belong to the OCC paradigm, the artificial information cannot be added to the training set. However, it can be used in the validation set (assuming that the Hold-out validation is employed) to guide the optimization process to a desired goal.

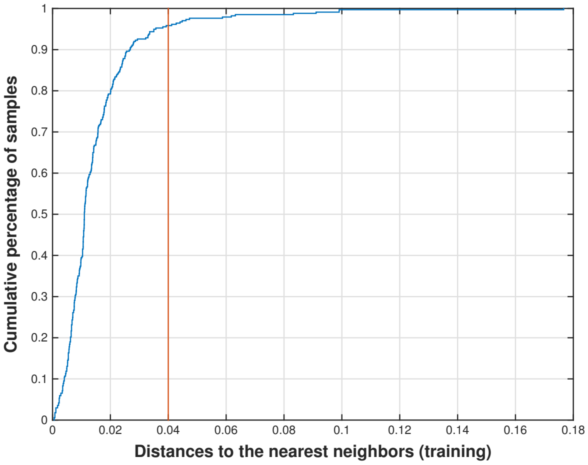

Fig. 3. Cumulative density function for one dataset.

IV. Experiments

This Section reports on experiments carried out to evaluate the performance of the proposed approach for generating novelty samples for the validation set. The datasets used for training and testing the models are rigorously the same used in Cabral and Oliveira [3]. The experiments were conducted using ten datasets, eight of them from the UCI repository [7] and two bi-dimensional datasets artificially generated (Gaussian Distributions and Banana). For the multiclasses data sets (Iris and Wine), one of the classes was picked as the novelty class and the others were merged to represent the normal class. This is similar to the procedure used by Oliveira *et al.* [8], Cao *et al.* [9] and Tax [6].

For the Artificial Dataset, instances belonging to the normal class were generated by a Gaussian distribution with 0 mean vector and covariance matrix with both entries 4; the samples belonging to the novelty class were generated by a Gaussian with mean vector with both entries 4 and covariance matrix with both entries 4. The Banana shaped dataset was generated by the DDTools toolbox [10]. These datasets are particularly important because they are bi-dimensional and thus it is possible to visualize the behavior of the algorithm and to validate it.

The experiments were performed by dividing each dataset into three sets: training, validation and test. The training set contains 50% of the available normal data while the validation and test sets contain 25% each. The training and test set are exactly the same ones used in [3]. The validation set differs only in the novelty samples. In [3], the validation set contained real samples of the novelty class while in the present work, the validation set contains only artificially generated novelty samples.

To assess the generalization performance of the methods, the average of the Mathews Correlation Coefficient (MCC) [24] of each model was used. Even being the Area Under (the ROC) Curve one of the most adopted metrics for assessing

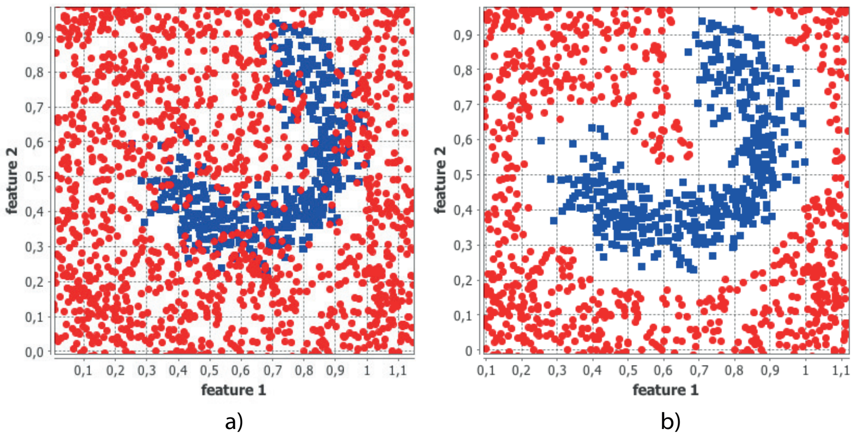

Fig. 4. Banana shaped dataset after the first phase of the artificial generation (a) and after the second phase (b).

the generalization power of One-class methods, in cases of unbalanced data, the classifier may yield a good AUC value by misclassifying a high rate of the minority class. In such cases, the MCC metric detects the misclassification of some of the classes and decreases the score of the model. The MCC value is obtained by picking up an operational point of the ROC curve and evaluating the error rate of the model for that point. For all the experiments, the chosen operational point is the nearest point to the coordinate (0,1) in the ROC space (i.e., the operational point which yields the lower error rate). The Equation 1 shows how to compute the MCC. The range of possible values for the MCC varies from -1 to 1.

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{1}$$

In Equation 1: **TP** represents the number of True Positive objects; **TN** represents the number of True Negative objects; **FP** represents the number of False Positive objects; and **FN** represents the number of False Negative objects. Notice that the class POSITIVE is the same as the novelty class and the NEGATIVE class is the same as the normal class.

Table I shows the averages and the standard deviations of the MCCs obtained by the experiments carried out with the employed OCC methods FBDOCC and OCSVM. The first two columns present the results obtained by the classifiers applied to the proposed approach, the next two columns present the results obtained in [3] and the last two columns contains the percentage difference between [3] and the present work. The results shown in Table I illustrates how similar are the results obtained by a modeling using real novelty data for validation and a 100% one-class modeling. In the case of the FBDOCC, the proposed approach achieved an overall result only 3% worse than [3] and the results achieved by the OCSVM performed only 5% worse.

In order to better illustrate the results from Table I, Figure 5 presents a bar chart containing the average MCCs of the experiments in this work and obtained in [3] for the FBDOCC method. In this Figure it is possible to verify that our approach yielded similar results to the ones in [3] for all problems, except Wine(3). In this case, our approach yielded a result 13% worse than [3].

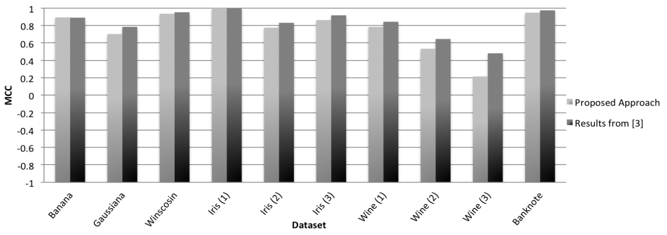

Fig. 5. Comparison of the results obtained by the FBDOCC using real novelty data[3] and artificial novelty data in the validation set.

Figure 6 shows another bar chart containing the average MCCs of the experiments executed in this work and in [3] for the OCSVM method. For this method, it is possible to verify that our approach worked similarly to [3] in seven out of 10 problems. For the problems Iris(2), Wine(2) and Wine(3) the results of the OCSVM of this work were considerably worse than [3].

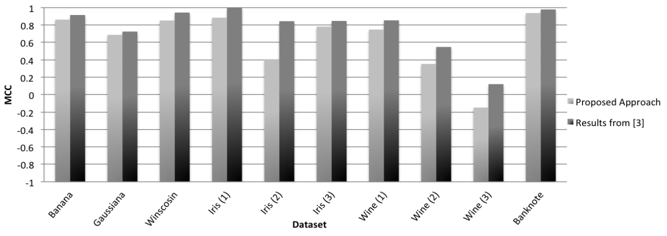

Fig. 6. Comparison of the results obtained by the OCSVM using real novelty data[3] and artificial novelty data in the validation set.

TABLE I
MCCS OBTAINED BY THE EXPERIMENTED METHODS USING THE VALIDATION SET COINTAINING GENUINE NOVELTY DATA AN ARTIFICIAL NOVELTY DATA.

Dataset	Proposed Approach		Results from [3]		Percentual difference	
	FBDOCC	OCSVM	FBDOCC	OCSVM	FBDOCC	OCSVM
Banana	0.894 (0.037)	0.862 (0.042)	0.891 (0.218)	0.915 (0.167)	0.15%	-2.65%
Gaussian Distributions	0.703 (0.062)	0.687 (0.094)	0.785 (0.232)	0.725 (0.200)	-4.10%	-1.90%
Winsconsin Breast Cancer	0.936 (0.018)	0.852 (0.054)	0.953 (0.137)	0.943 (0.125)	-0.85%	-4.55%
Iris (1)	1.000 (0.000)	0.884 (0.084)	1.000 (0.000)	1.000 (0.000)	0.00%	-5.80%
Iris (2)	0.776 (0.124)	0.407 (0.454)	0.832 (0.275)	0.844 (0.230)	-2.80%	-21.80%
Iris (3)	0.863 (0.054)	0.782 (0.065)	0.918 (0.213)	0.847 (0.259)	-2.75%	-3.25%
Wine (1)	0.785 (0.054)	0.748 (0.0786)	0.844 (0.232)	0.854 (0.195)	-2.95%	-5.30%
Wine (2)	0.534 (0.131)	0.353 (0.104)	0.646 (0.301)	0.548 (0.290)	-5.60%	-9.75%
Wine (3)	0.216 (0.200)	-0.148 (0.115)	0.482 (0.377)	0.121 (0.417)	-13.30%	-13.45%
Banknote	0.948 (0.026)	0.937 (0.035)	0.975 (0.297)	0.979 (0.168)	-1.35%	-2.10%

V. CONCLUSION

In this paper we have proposed the use of an approach for artificial generation of samples of the novelty class. The idea is to use these artificial samples to form a validation dataset to be used by the optimization method for finding the best model. Since examples of the novelty class are rare or they don't exist during the modeling phase, the use of artificial samples can support in building a tight closed description of the normal class. Furthermore, the adopted approach extinguishes the need of the manual search for the best parameter configuration.

Our simulations using real and synthetic data sets have shown that the proposed approach has achieved a good performance in terms of MCC in comparison with the experiments carried out in [3].The advantage of our proposal, in comparison to the method of ref. [3], is that it can be applied to model selection of OCC in datasets that do not have or have few novelty samples. Considering the method FBDOCC, in nine out of ten data sets our approach yielded similar results to [3] - where genuine examples of the novelty class were used in the validation dataset. Considering the OCSVM method, in seven out of ten data sets our approach yielded similar results to [3].

Our future works include the improvement and development of new methods for generating artificial examples of the novelty class for batch and online problems.

REFERENCES

[1] C. Désir, S. Bernard, C. Petitjean and L. Heutte. One class random forests. Pattern Recognition, Vol. 46(12), pp. 3490-3506, 2013.

[2] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola and R. C. Williamson. Estimating the support of a high-dimensional distribution. Neural Computation, Vol. 13(7), pp. 1443–1471, 2001.

[3] G. G. Cabral and A. L. I. Oliveira. One-Class Classification based on searching for the problem features limits. Expert Systems with Applications. Vol. 41(16), pp. 7182-7199, 2014.

[4] A. L. I. Oliveira. Neural Networks Forecasting and Classification-Based Techniques for Neural Networks Forecasting and Classification-Based Techniques for Novelty Detection in Time Series. 2004. PhD Thesis Federal University of Pernambuco.

[5] H. Hoffmann. Kernel PCA for novelty detection. Pattern Recognition, Vol. 40, pp. 863-874, 2007.

[6] D. M. J. Tax, (2001). One-class classification concept-learning in the absence of counterexamples (Ph.D. thesis). Technische Universiteit Delft.

[7] A. Frank and A. Asuncion. UCI machine learning repository, 2010. http://archive.ics.uci.edu/ml.

[8] A. L. I. Oliveira, F. R. G. Costa and C. O. S. Filho. Novelty detection with constructive probabilistic neural networks. Neurocomputing, Vol. 71(4-6), pp. 1046–1053, 2008.

[9] L. Cao, H. P. Lee and W. K. Chong. Modified support vector novelty detector using training data with outliers. Pattern Recognition Letters. Vol. 24(14), pp. 2479–2487, 2003.

[10] D. Tax. Ddtools, the data description toolbox for matlab. Version 1.9.0, 2011.

[11] M. Kemmler, E. Rodner and J. Denzler. One-class classification with Gaussian processes. Pattern Recognition, Vol. 46, n.12, pp. 35073518, 2013.

[12] J. Oh, N. Kwak, M. Lee and C. H. Choi. Generalized mean for feature extraction in one-class classification problems. Pattern Recognition, Vol. 46(12), pp. 3328–3340, 2013.

[13] E. Carrizosa, B. M.-Barragán and D. R. Morales. A nested heuristic for parameter tuning in Support Vector Machines. Computers & Operations Research, Vol. 43, pp. 328–334, 2014.

[14] E. Keogh, S. Lonardi and C. A. Ratanamahatana. Towards parameter-free data mining. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04) pp. 206–215, 2004.

[15] J.-S. Chou, M.-Y. Cheng, Y.-W. Wu and A.-D. Pham. Optimizing parameters of support vector machine using fast messy genetic algorithm for dispute classification. Expert Systems with Applications, Vol. 41(8), pp. 3955–3964, 2014.

[16] C. Liu, D. Chen and F. Wan. Multiobjective learning algorithm based on membrane systems for optimizing the parameters of extreme learning machine. Optik - International Journal for Light and Electron Optics, Vol. 127(4), pp. 1909–1917, 2016.

[17] C. yongqi. LSSVM Parameters Selection Based on Hybrid Complex Particle Swarm Optimization. Energy Procedia, Vol. 17, pp. 706–710, 2012.

[18] V. Chandola, A. Banerjee and V. Kumar. Anomaly detection: A survey. ACM Computing Surveys, 41(3), pp. 1-72, 2009.

[19] S. W. Salvador and P. Chan. Learning states and rules for detecting anomalies in time series. Applied Intelligence, Vol. 23(3), pp. 241255, 2005.

[20] J. Kennedy and R. Eberhart. Particle swarm optimization, in: IEEE International Conference on Neural Networks (ICNN95), Vol. 4, pp. 1942-1947, 1995.

[21] D. Bratton and J. Kennedy. Defining a standard for particle swarm optimization, in: Swarm Intelligence Symposium, pp. 120-127, 2007.

[22] J. Kennedy. Why does it need velocity?. In Proceedings of IEEE Swarm Intelligence Symposium, 2005 (SIS 2005), pp. 38-44, 2005.

[23] A. P. Engelbrecht. Fundamentals of Computational Swarm Intelligence, Wiley, 2005.

[24] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: An overview. Bioinformatics, Vol. 16, pp. 412-424, 2000.