2014 22nd International Conference on Pattern Recognition

Weighted One-Class Classifier Ensemble Based on Fuzzy Feature Space Partitioning

Bartosz Krawczyk, Michał Woźniak
Department of Systems and Computer Networks
Wroclaw University of Technology
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
Email: {bartosz.krawczyk,michal.wozniak}@pwr.wroc.pl

Bogusław Cyganek
AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Krakow, Poland
Email: cyganek@agh.edu.pl

***Abstract*—This paper introduces a novel method for forming efficient one-class classifier ensembles. A common problem in one-class classification is a complex structure of the target class, which often leads to creation of a too expanded decision boundary. We propose to employ a clustering step in order to partition the target class into atomic subsets and using these as input for one-class classifiers. By this, we are able to detect sub-structures in the target concept. Additionally, to increase the diversity and robustness of our method weighted one-class classifiers are used. We introduce a novel scheme for calculating weights for training objects. Membership functions, obtained from the fuzzy clustering, are used to initialize the weighted classifiers. Based on the results of a number of computational experiments we show that the proposed method outperforms both the single one-class methods, as well as popular one-class ensembles. Other advantages are the highly parallel structure of the proposed solution, which facilitates parallel training and execution stages, and the relatively small number of control parameters.**

I. Introduction

Nowadays we are faced with a plethora of new and challenging problems such as an atypical class distribution, non-stationary environments, or massive data. Such cases very often cannot be handled efficiently by canonical and well-known classifiers. Therefore, there is a need for introducing novel methods with such arising difficulties and improve the quality of real-life decision support systems.

One of these newly introduced methodologies is known as one-class classification (OCC) [1]. It is based on the principle, that during the training stage only objects coming from a single class are at our disposal. These are called the target concept or target class, and are denoted by ω_T. OCC aims at calculating a decision boundary that encloses all available data objects, thereby describing the given concept [2]. During the exploitation step, new objects, unknown during training, may appear. These may originate from one or more distributions and represent data that do not belong to the target concept. Such objects, denoted by ω_O, are called outliers.

For a single OCC classifier it may be fairly difficult or even impossible to find a good decision boundary owing to limited training data, high feature space dimensionality, and/or the properties of the particular classifier. To avoid a too complex model and overfitting of the training target data, a simpler model with a lower number of features or one that has been trained with smaller chunks of data, can be created. Although the complexity of such a model is reduced, the quality thereof also declines significantly. However, it has been shown that a group of individual OCC models can help alleviate the aforementioned problems.

Here one may use an approach known as multiple classifier systems (MCSs). MCSs are based on the idea of combining a number of learners into one compound system, which takes advantages of the competence areas of all its members. Each classifier may display individual learning properties, and so have different competence areas [3]. When carefully combined, the quality of the combined decision can outperform any of the individual predictors. However, several important issues, such as selecting the individual classifiers, as well as choosing a fusion method to establish a group decision, must be considered when designing an MCS. Classifiers used to create the ensemble in an ideal situation should be highly accurate and complement each other (i.e., the ensemble should display high diversity). Adding classifiers that are not diverse with respect to those already in the pool will not improve the accuracy of the compound classifier, but will only increase the overall computational cost [4]. On the other hand, building an MCS with highly diverse but poor quality classifiers will result in a weak committee [5].

MCSs are a promising yet still unexplored, direction for tackling OCC problems. Most of the works done in this field deals with practical applications of OCC ensembles. Much still needs to be done to gain insight into the theoretical background to this problem, as well as to draw conclusions on how to build efficient OCC ensembles regardless of the intended application [6].

We propose a novel approach for forming one-class classifier committees, based on data clustering in the feature space. Partitioning approach had performed very good in our previous work, where it was used for decomposing multi-class datasets [7]. OCC models are built based on each of the clusters. In this way we ensure that the pool of predictors is highly diverse and mutually complementary (owing to training on different inputs, i.e., clusters of training objects). This can be seen as an extension of the popular family of ensembles derived from the idea of *clustering and selection* proposed by Kuncheva [8]. So far, two other research teams have worked on this

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.489

topic, proposing very simple hybrid methods for combining clustering and OCC [9]. In our approach we further extend this concept by utilizing weighted one-class classifiers and proposing a new scheme for calculating their weights.

The contributions of this work are as follows:

- We propose building an ensemble of one-class classifiers based on clustering of the target class. This ensures initial diversity among the classifiers in the pool (as they are based on different inputs) and the correct handling of possible issues embedded in the nature of data, such as a rare distribution or chunks of objects.
- We apply weighted one-class classifiers, as they are more robust to complex data structures than their canonical counterparts. However, calculating the weights assigned to each object is a difficult and computationally costly task. Therefore, methods for improving the weighting step can significantly boost the quality of these methods.
- We introduce a novel weighting scheme, based on the output of the feature space partitioning phase. By utilizing fuzzy clustering, one obtain membership values for each object in the given cluster. Our proposed ensemble uses these values to initialize the weights in classifier trained on the given chunk of data. Therefore, weights reflect the importance of the object in the cluster and are calculated beforehand, reducing the complexity of the training phase. We show, that the proposed weighting scheme returns highly satisfactory performance.

Our ensemble is easy to use in many practical applications where it is difficult or even impossible to obtain counter-examples (e.g., machine fault diagnosis), or where, owing to a complex data distribution, the class decomposition approach can lead to a significant improvement in recognition quality over the well known multi-class approaches (e.g., imbalanced classification).

II. One-Class Classification

OCC uses training objects coming from a single class to create a classifier. It aims at dichotomizing between the given data and new, unseen examples that cannot be considered as target objects. Hence, OCC is also known as learning in the absence of counter-examples. Although OCC is quite similar to binary classification, the primary difference lies in how the one-class classifier is trained. In standard dichotomy problems it is expected that objects from the other classes tend to come from one direction. Here the available class must be separated from all the possible outliers, which leads to a situation in which a decision boundary must be estimated in all directions in the feature space around the target class.

OCC is a solution to many real-life problems where there are abundant data for a single class, but it is difficult or even impossible to obtain data for other objects. This is often the case in problems such as intrusion detection [10] or machine fault diagnosis.

Several methods dedicated to solving OCC problems have recently been introduced. Based on the literature, three main approaches can be distinguished:

- Density estimation of a target class [11].
- Reconstruction methods, based on clustering algorithms [12].
- Methods, that estimate only the close boundary for the given data, assuming that such a boundary will sufficiently describe the target class [13].

There are some works that discuss the problem of building OCC committees [14] and identifies areas of their practical applications [10]. Combined classifiers are a promising direction for OCC [6]. Their main advantage lies in training less complex individual classifiers, thereby reducing the risk of model overfitting, which is one of the major problems in OCC. Additionally, they can be easily implemented in a distributed environment; most of the OCC classifiers (especially the boundary-based ones) requires a high computational processing time, and therefore, relying on several weak models that run independently may significantly reduce the overall training time of the recognition system.

A. *Weighted One-Class Support Vector Machine*

One-Class Support Vector Machine (OCSVM) [15] is a boundary-based method for OCC. It estimates the decision surface in a form of a hypersphere enclosing all the objects from ω_T. During the exploitation phase a decision made about the new object is based upon checking whether it falls inside the hypersphere. If so, the new object is labeled as one belonging to ω_T. Otherwise it belongs to ω_O.

The center a and a radius R are the two parameters that are sufficient for describing such a decision hypersphere. To have a low acceptance of the possible outliers the volume of this d-dimensional hypersphere, which is proportional to R^d, should be minimized in such a way that tightly encompasses all available objects from ω_T. The minimization of R^d implies minimization with respect to R^2. Following this the minimization functional may be formulated as follows:

$$\Theta(a, R) = R^2, \tag{1}$$

with respect to the constraint:

$$\forall_{1 \leq i \leq N}: \quad \|x_i - a\|^2 \leq R^2, \tag{2}$$

where x_i are objects from ω_T, and, N stands for the quantity of training objects. Additionally to allow the fact that there may have been some outliers in the training set and to increase the robustness of the trained classifier some objects with distance to a greater than R are allowed in the training set, but associated with an additional penalty factor. This is done identically as in a standard SVM by the introduction of slack variables ξ_i.

This concept can be further extended to a Weighted One-Class Support Vector Machine (WOCSVM) [16] by the introduction of weights w_i that allows for an association of an importance measure to each of the training objects. This forces slack variables ξ_i, to be additionally controlled by w_i. If with object x_i there is associated a small weight w_i then the corresponding slack variable ξ_i indicates a small penalty. In effect, the corresponding slack variable will be larger, allowing x_i to lie further from the center a of the hypersphere. This reduces an impact of x_i on the shape of a decision boundary of WOCSVM.

By using the above mentioned ideas we can modify the minimization functional:

$$\Theta(a,R) = R^2 + O\sum_{i=1}^{N} w_i \xi_i, \tag{3}$$

with the modified constraints that almost all objects are within the hypersphere:

$$\forall_{1 \leq i \leq N}: \quad \|x_i - a\|^2 \leq R^2 + \xi_i, \tag{4}$$

where $\xi_i \geq 0$, $0 \leq w_i \leq 1$. Here O stands for a parameter that controls the optimization process - the larger O, the less outliers are allowed with the increase of the volume of the hypersphere.

The differences between OCSVM and WOCSVM are depicted in Figure 1.

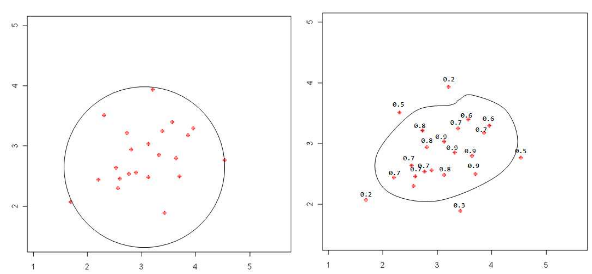

Fig. 1. Decision boundaries for a toy problem produced by OCSVM and WOCSVM (with weight values for each point from the target class).

III. Proposed One-Class Classifier Ensemble

In this paper we propose a new architecture for creating ensembles of one-class classifiers based on the clustering of a feature space into smaller partitions - a novel committee named one-class clustering-based ensemble (OCClustE). The idea behind OCClustE originates in our previous works where we showed that increasing the number of one-class classifiers assigned to the target class can often lead to a significant improvement in classification accuracy [17].

However, in such an approach we face the problem on how to create a pool of base one-class classifiers, that are at the same time individually accurate and mutually diverse.

Our method uses a clustering algorithm to partition the feature space into atomic subsets. In the next step each of these clusters is used to train a one-class classifier. This leads to the formation of a pool of K classifiers assigned to the target class, as follows:

$$\Pi = \{\Psi^{(1)}, \Psi^{(2)}, ..., \Psi^{(K)}\}. \tag{5}$$

This allows us to easily create a pool of several one-class learners, dedicated to the target class. It assures the initial diversity (as a result of using different inputs in their training) and complementarity (as classifiers together cover all the decision space), which leads to better performance of the ensemble.

For the clustering step, OCClustE uses kernel fuzzy c-means, which is a modification of the fuzzy c-means algorithm that operates in an artificial feature space created by a kernel function [18].

An overview of the proposed OCClustE method is illustrated in Fig. 2.

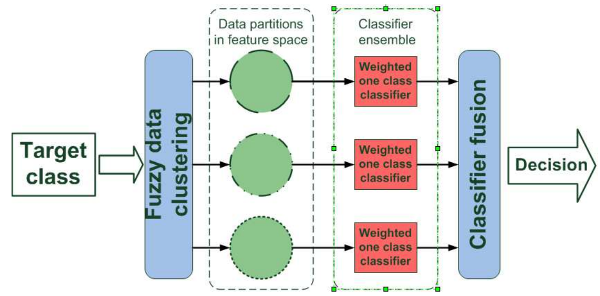

Fig. 2. Overview of the OCClustE.

Main differences between the single-model OCC and the proposed OCClustE are presented in Figure 3.

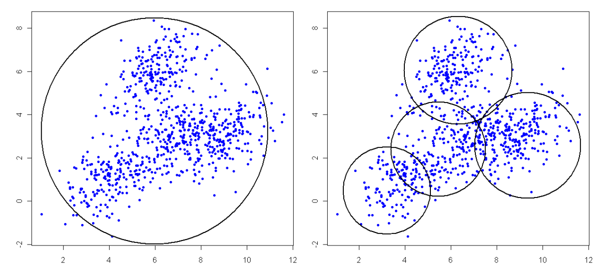

Fig. 3. Differences between the outputs of a standard approach and the proposed one for a one-class toy problem. (*Left*) Target concept enclosed by a single model approach. (*Right*) Target concept after OCClustE classification with four clusters.

To further boost the quality of OCClustE, we propose to use weighted one-class support vector machine (WOCSVM) as the base learner. It has been shown, that weighted one-class classifiers can outperform the canonical ones, due to manipulating the influence degree that each object has on the shape of the decision boundary. Additionally, weighted methods are insensitive to internal outliers, that may be present in the target class (as it may contain irrelevant, noisy objects). By assigning them a low weight, they have minimal impact on the process of shaping the decision boundary.

The crucial element in using WOCSVM is the process of establishing weights, which is heuristic and time-consuming [16]. We introduce a novel approach for establishing the degree of importance of objects, based on the output of clustering algorithm. We use fuzzy clustering algorithm, that returns the membership functions for each object in the given cluster. We use these membership values as weights for WOCSVM. This way, the new weights reflect the degree of importance of a given object in a cluster and are pre-calculated, therefore reducing the computational time needed for training WOCSVM.

Number of clusters on the target class reflects the number of classifiers delegated for handling the given one-class problem. One may see, that this has an crucial impact on the performance of the proposed method. For automatic assessment of the number of clusters we propose to use the entropy of the membership values, which depends on the data and the number of clusters C, as it is a good indicator of the quality of clustering [19]. This is computed as:

$$E(C) = -\sum_{c=1}^{C}\sum_{i=1}^{N} w_{ci} \log w_{ci}, \tag{6}$$

where N denotes the number of data points and w_{ci} is the weight assigned to a given cluster. If the number of clusters C is not known in advance, clustering can be performed for a varying numbers of clusters, and the number with minimal entropy can be chosen to build the ensemble [20]. Such a strategy usually provides a useful indication of the number of means, which is data dependent. We acknowledge that the entropy criterion is not a perfect solution for estimating the number of clusters. Yet, at the same time, it is less time-consuming than manually checking the correlation between the number of clusters and final accuracy. As our aim is to create an ensemble classifier that will require minimal intervention with regard to parameter settings from the end-user, the entropy criterion seems an attractive solution.

Finally, we need to combine the individual outputs of base classifiers at our disposal. One-class boundary methods (as used here WOCSVM) are based on computing the distance between the object x and the decision boundary that encloses the target class ω_T. To apply fusion methods we require the support function of object x for a given class.

We propose to use the following heuristic solution:

$$\widehat{F}(x, \omega_T) = \frac{1}{c_1} exp(-d(x|\omega_T)/c_2), \tag{7}$$

which models a Gaussian distribution around the classifier, where $d(x|\omega_T)$ is an Euclidean distance metric between the considered object and a decision boundary, c_1 is the normalization constant and c_2 is the scale parameter. Parameters c_1 and c_2 should be fitted to the target class distribution.

There are several propositions on how to fuse the outputs of individual OCC models after such a mapping [21]. Let us assume that there are K OCC classifiers in the pool. In this paper, we use the mean of the estimated support functions which is expressed by:

$$y_{mp}(x) = \frac{1}{K}\sum_{k}(F_k(x, \omega_T)). \tag{8}$$

This fusion method assumes that the outlier object distribution is independent of x and thus uniform in the area around the target concept.

In summary, the approach proposed in this paper leads to several improvements compared with the standard OCC models:

- Boundary-based approaches (such as WOCSVM) were shown to display better generalization abilities than clustering-based (reconstruction) OCC [22], but are highly prone to atypical and complex data distributions. Therefore, a hybrid method utilizing both approaches combines the advantages of each while reducing their drawbacks.
- As each classifier is trained only on a reduced chunk of the data, its computational complexity is reduced in comparison to a single model approach. This reduces the probability of overtraining the one-class learner. Additionally, a number of individual classifiers can easily be applied in a distributed environment, leading to a significant decrease in execution time.
- Using chunks of data as the classifier input reduces the influence of negative effect, known as the empty sphere; that is, the area covered by the boundary in which no objects from the training set are located.
- A boundary classifier trained on a more compact data partition usually has a lower number of support vectors.
- By combining the fuzzy clustering with weighting scheme, we are able to obtain good estimation of weights assigned to training objects in a reduced time.

IV. Experimental Investigations

The aims of this experiment was to evaluate the effectiveness of the proposed ensemble and compare it to popular single-model and committee approaches for one-class classification.

A. Datasets

As there are no benchmarks dedicated to one-class classification, we have chosen 10 binary datasets - 9 from the UCI Repository and an additional one, originating from chemoinformatics domain and describing the process of discovering pharmaceutically useful isoforms of CYP 2C19 molecule. The data set is available for download at [23].

The objects from the minor class were used as the target concept, while objects from the major class as outliers.

Details of the chosen data sets are given in Table I.

B. Set-up

For the experiment a Weighted One-Class Support Vector Machine with a RBF kernel is used as a base classifier. The pool of classifiers were homogeneous, i.e. consisted of classifiers of the same type.

In order to present a detailed comparison among a group of machine learning algorithms, one must use statistical tests to prove, that the reported differences among classifiers are significant. We use both pairwise and multiple comparison tests. Pairwise tests give as an outlook on the specific performance of methods for a given data set, while multiple comparison allows us to gain a global perspective on the performance

TABLE I
DETAILS OF DATASETS USED IN THE EXPERIMENTAL INVESTIGATION. NUMBERS IN PARENTHESES INDICATES THE NUMBER OF OBJECTS IN THE MINOR CLASS IN CASE OF BINARY PROBLEMS.

No.	Name	Objects	Features	Classes
1	Breast-cancer	286 (85)	9	2
2	Breast-Wisconsin	699 (241)	9	2
3	Colic	368 (191)	22	2
4	Diabetes	768 (268)	8	2
5	Heart-statlog	270 (120)	13	2
6	Hepatitis	155 (32)	19	2
7	Ionosphere	351(124)	34	2
8	Sonar	208 (97)	60	2
9	Voting records	435 (168)	16	2
10	CYP2C19 isoform	837 (181)	242	2

of the algorithms over all benchmarks. With this, we get a full statistical information about the quality of the examined classifiers.

- For a pairwise comparison, we use a 5x2 combined CV F-test. It repeats five-time two fold cross-validation so that in each of the folds the size of the training and testing sets is equal. This test is conducted by comparison of all versus all.
- For assessing the ranks of classifiers over all examined benchmarks, we use a Friedman ranking test. It checks, if the assigned ranks are significantly different from assigning to each classifier an average rank.
- We use the Shaffer post-hoc test to find out which of the tested methods are distinctive among an n x n comparison. The post-hoc procedure is based on a specific value of the significance level α. Additionally, the obtained p-values should be examined in order to check how different given two algorithms are.

We fix the significance level $\alpha = 0.05$ for all comparisons.

To put the obtained results into a context, we compare our method with a single WOCSVM and its bagged and boosted version (each consisting of 10 classifiers in the pool). Additionally, we present the results for a one-class clustering ensemble that uses standard OCSVM as a base learner, to show that using weighted classifiers boost the quality of the proposed OCClustE.

C. Results

The results are presented in Table II. $SINGLE$ stands for a single WOCSVM model, $BAGG$ stands for a bagged WOCSVM, $BOOST$ for a boosted WOCSVM, $CLUST$ for simple clustering-based ensemble without the weighting module and $OCCLUSTE$ for the proposed method. Small numbers under each method stands for the indexes of models from which the considered one is statistically better. The last row presents ranks according to the Friedman test.

Results of the Shaffer post-hoc test between the OCClustE and reference methods are depicted in Table III

D. Results Discussion

For 7 out of 10 cases the proposed OCClustE outperforms in a statistically significant way both single WOCSVM and remaining one-class ensemble methods. In the remaining three cases it returns statistically similar results to the ensemble reference methods. What is of high importance is the fact, that the proposed method, for tested cases, is never inferior to any of the remaining one-class classifiers. Shaffer test shows, that OCClustE is statistically superior to all other algorithms, when taking int account its performance over multiple data sets.

OCClustE always outperforms a single WOCSVM. This is due to the fact, that single one-class classifier often cannot find a good description boundary. Either the boundary volume is too big (due to the complex distribution), which leads to a high outlier acceptance rate; or it is too fitted to the data, which results in poor generalization. Using a larger number of less complex classifiers seems to reduce this problem and generates a robust classifier with good generalization abilities.

OCClustE often shows better performance than bagging and boosting, popular ensemble methods in one-class classification. This is due to the fact, that both bagging and boosting were originally introduced for multi-class problems and are not fitted to the specific nature of OCC. Bagging uses subgroups of objects, but do not assure that they are atomic. This leads to forming classifiers with too big volume of the

TABLE II
RESULTS OF THE EXPERIMENTAL RESULTS WITH THE RESPECT TO THE ACCURACY [%] AND STATISTICAL SIGNIFICANCE. SMALL NUMBERS UNDER EACH METHOD STANDS FOR THE INDEXES OF MODELS FROM WHICH THE CONSIDERED ONE IS STATISTICALLY BETTER.

No.	SINGLE[1]	BAGG[2]	BOOST[3]	CLUST[4]	OCCLUSTE[5]
1.	57.86	58.56	60.94	59.43	63.79
	–	–	1,2	1,2	1,2,3,4
2.	87.21	89.52	89.87	89.23	91.45
	–	1	1	1	1,2,3,4
3.	69.90	75.37	73.95	76.65	78.03
	–	1,3	1	1,3	1,2,3,4
4.	58.45	59.21	59.12	60.46	62.05
	–	–	–	1,2,3	1,2,3,4
5.	83.12	86.90	86.73	85.72	87.11
	–	1,4	1,4	1	1,4
6.	58.23	58.02	59.12	58.41	60.46
	−2	–	–	–	1,2,3,4
7.	73.52	79.41	81.04	77.21	80.63
	–	1,4	1,2,4	1	1,2,4
8.	85.23	90.01	89.34	87.53	92.12
	–	1,4	1,4	1	1,2,3,4
9.	87.45	89.32	89.71	88.30	89.64
	–	1,4	1,4	1	1,4
10.	73.90	76.04	77.56	75.28	80.09
	–	1,4	1,2,4	1	1,2,3,4
Rank	4.78	3.35	2.39	3.02	1.46

TABLE III
SHAFFER TEST FOR COMPARISON BETWEEN THE OCCLUSTE AND REFERENCE METHODS. SYMBOL '=' STANDS FOR CLASSIFIERS WITHOUT SIGNIFICANT DIFFERENCES, '+' FOR SITUATION IN WHICH THE METHOD ON THE LEFT IS SUPERIOR AND '-' VICE VERSA.

hypothesis	p-value
OCClustE vs SINGLE	+ (0.0032)
OCClustE vs BAGG	+ (0.0140)
OCClustE vs BOOST	+ (0.0298)
OCClustE vs CLUST	+ (0.0127)

decision boundary. Boosting reduces the error rate on a single class, which often may lead to overfitting to the training data, especially if the number of training objects is small. OCClustE can prevent both of these situations from happening by using effective kernel clustering and weighted classification.

In comparison with its simpler version (without using weighted classifiers) OCClustE always scores superior results. This shows the importance of weighting the influence of training objects in OCC. It also proves, that our proposed method for calculating weights based on the cluster membership functions, is an effective way for producing accurate and diverse classifiers.

Use of the entropy criterion allowed for the automatic determination of the most promising number of clusters for an ensemble. The experimental results confirm that this criterion coped well with the proposed classification architecture, eliminating the time-consuming manual tuning phase. Of course, the entropy criterion is merely a suggestion for the number of clusters and better results may be achieved after manual experimentation with the settings; this was, however, not our goal. We aimed to create an ensemble classifier that would be easy for the end-user to use. To this end, automatic cluster selection by means of the entropy criterion worked satisfactorily. In future work, we would like to explore the possibility of using clustering methods that do not require a priori knowledge about the number of groups in the data, e.g., DBSCAN.

V. Conclusion

This paper presented a novel method for forming a one-class classifier ensemble based on feature space partitioning. The main advantage of the proposed method is that the combined classifiers trained on the basis of clusters allow us to exploit individual classifier strengths and reduce their complexity. Additionally, we proposed a novel method for establishing weights for WOCSVMs, used as base classifiers. As a result, our committee usually outperform traditional methods for one-class classification and popular ensemble approaches. This observation was confirmed by the computational experiments carried out using a wide range of benchmark data sets. Our approach leads to a decrease in the overall training time by distributing the computations for each of the clusters, which could even be executed on different processors.

Acknowledgment

Bartosz Krawczyk was supported by the Polish National Science Centre under the grant PRELUDIUM number DEC-2013/09/N/ST6/03504.

Michał Woźniak was supported by EC under FP7, Coordination and Support Action, Grant Agreement Number 316097, ENGINE "European Research Centre of Network Intelligence for Innovation Enhancement" (http:// engine.pwr.wroc.pl/).

Bogusław Cyganek was supported in the year 2014 from the funds of the Polish National Science Centre NCN, contract no. DEC-2011/01/B/ST6/01994, as well as AGH Statutory Funds no. 11.11.230.017.

References

[1] M. W. Koch, M. M. Moya, L. D. Hostetler, and R. J. Fogler, "Cueing, feature discovery, and one-class learning for synthetic aperture radar automatic target recognition," *Neural Networks*, vol. 8, no. 7-8, pp. 1081–1102, 1995.

[2] D. Tax and R. P. W. Duin, "Characterizing one-class datasets," in *Proceedings of the Sixteenth Annual Symposium of the Pattern Recognition Association of South Africa*, 2005, pp. 21–26.

[3] P. P. K. Chan, D. S. Yeung, W. W. Y. Ng, C. M. Lin, and J. N. K. Liu, "Dynamic fusion method using localized generalization error model," *Information Sciences*, vol. 217, pp. 1–20, 2012.

[4] Y. Bi, "The impact of diversity on the accuracy of evidential classifier ensembles," *International Journal of Approximate Reasoning*, vol. 53, no. 4, pp. 584–607, 2012.

[5] Q. Dai, "A competitive ensemble pruning approach based on cross-validation technique," *Knowledge-Based Systems*, vol. 37, pp. 394–414, 2013.

[6] B. Krawczyk and M. Woźniak, "Diversity measures for one-class classifier ensembles," *Neurocomputing*, vol. 126, pp. 36–44, 2014.

[7] B. Krawczyk, M. Woźniak, and B. Cyganek, "Clustering-based ensembles for one-class classification," *Information Sciences*, vol. 264, pp. 182–195, 2014.

[8] L. I. Kuncheva, "Clustering-and-selection model for classifier combination," in *KES*, 2000, pp. 185–188.

[9] N. Lipka, B. Stein, and M. Anderka, "Cluster-based one-class ensemble for classification problems in information retrieval," in *SIGIR'12 - Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012, pp. 1041–1042.

[10] G. Giacinto, R. Perdisci, M. Del Rio, and F. Roli, "Intrusion detection in computer networks by a modular ensemble of one-class classifiers," *Inf. Fusion*, vol. 9, pp. 69–82, January 2008.

[11] S. Sonnenburg, G. Rtsch, C. Schfer, and B. Schlkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[12] K. F. Cheung, "Fuzzy one-mean algorithm: Formulation, convergence analysis, and applications," *Journal of Intelligent and Fuzzy Systems*, vol. 5, no. 4, pp. 323–332, 1997.

[13] P. Juszczak, D. M. J. Tax, E. Pekalska, and R. P. W. Duin, "Minimum spanning tree based one-class classifier," *Neurocomputing*, vol. 72, no. 7-9, pp. 1859–1869, 2009.

[14] T. Wilk and M. Woźniak, "Soft computing methods applied to combination of one-class classifiers," *Neurocomput.*, vol. 75, pp. 185–193, Jan. 2012.

[15] B. Schölkopf and A. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, ser. Adaptive computation and machine learning. MIT Press, 2002.

[16] M. Bicego and M. A. T. Figueiredo, "Soft clustering using weighted one-class support vector machines," *Pattern Recognition*, vol. 42, no. 1, pp. 27–32, 2009.

[17] B. Krawczyk and M. Woźniak, "Combining diverse one-class classifiers," in *Hybrid Artificial Intelligent Systems*, ser. Lecture Notes in Computer Science, E. Corchado, V. Snasel, A. Abraham, M. Wozniak, M. Grana, and S.-B. Cho, Eds. Springer Berlin / Heidelberg, 2012, vol. 7209, pp. 590–601.

[18] L. Zhang, W. . Zhou, and L. . Jiao, "Kernel clustering algorithm," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 25, no. 6, pp. 587–590, 2002.

[19] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[20] B. Cyganek, "One-class support vector ensembles for image segmentation and classification," *Journal of Mathematical Imaging and Vision*, vol. 42, no. 2-3, pp. 103–117, 2012.

[21] D. M. J. Tax and R. P. W. Duin, "Combining one-class classifiers," in *Proceedings of the Second International Workshop on Multiple Classifier Systems*, ser. MCS '01. London, UK: Springer-Verlag, 2001, pp. 299–308.

[22] ——, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[23] *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA*, SIAM. SIAM Omnipress, 2011, http://tunedit.org/challenge/QSAR.