Instance reduction for one-class classification

Bartosz Krawczyk1 · Isaac Triguero2 · Salvador García3 · Michał Woźniak4 · Francisco Herrera3,5

Received: 18 January 2017 / Revised: 13 February 2018 / Accepted: 7 May 2018 / Published online: 21 May 2018
© Springer-Verlag London Ltd., part of Springer Nature 2018

Abstract Instance reduction techniques are data preprocessing methods originally developed to enhance the nearest neighbor rule for standard classification. They reduce the training data by selecting or generating representative examples of a given problem. These algorithms have been designed and widely analyzed in multi-class problems providing very competitive results. However, this issue was rarely addressed in the context of one-class classification. In this specific domain a reduction of the training set may not only decrease the classification time and classifier’s complexity, but also allows us to handle internal noisy data and simplify the data description boundary. We propose two methods for achieving this goal. The first
one is a flexible framework that adjusts any instance reduction method to one-class scenario by introduction of meaningful artificial outliers. The second one is a novel modification of evolutionary instance reduction technique that is based on differential evolution and uses consistency measure for model evaluation in filter or wrapper modes. It is a powerful native one-class solution that does not require an access to counterexamples. Both of the proposed algorithms can be applied to any type of one-class classifier. On the basis of extensive computational experiments, we show that the proposed methods are highly efficient techniques to reduce the complexity and improve the classification performance in one-class scenarios.

**Keywords** Machine learning · One-class classification · Instance reduction · Training set selection · Evolutionary computing

### 1 Introduction

Data preprocessing is an essential step within the machine learning process [21,41,42]. This kind of techniques aims to simplify the training data by removing noisy and redundant data, so that machine learning algorithms can be later applied faster and more accurately. In the literature, we can find techniques that focus on the attribute space and others that take into consideration the instance space. From the perspective of attributes, the most well-known data reduction processes are feature selection, feature weighting and feature extraction [4,7,30,44]. Taking into consideration the instance space, we may highlight instance reduction (InR) methods [17,51].

Instance reduction models search for a reduced set of instances that represents the original training data [14]. These techniques could be grouped into instance selection [17] and instance generation [51] models. The former only selects a subset of instances from the training dataset [35]. The latter may select or generate new artificial instances. Thus, InR can be seen as a combinatorial and optimization problem. Most of the existing InR models have been designed to improve the classification capabilities of the nearest neighbor rule [10], and they are denoted as prototype reduction methods. Among the existing InR methods, evolutionary algorithms have been stressed as the most promising ones [53].

The issue of data preprocessing has so far been rarely addressed in the context of one-class classification (OCC) [28]. This branch of machine learning focuses on scenarios in which we do not have access to counterexamples during the training phase. Therefore, a classifier must be trained using objects coming only from a single class, thus creating a data description. Here the quality of training objects has even greater influence on the classification stage than in binary and multi-class problems, as we need to capture such properties of the target class that will allow us to discriminate against unknown outliers. Noisy or rare objects can undermine the performance of one-class classifier, leading to overestimated class boundary [36]. At the same time, training OCC models on large-scale and big data may be of high computational complexity, which will hinder the learning process. This is especially vivid in cases of massive, streaming or complex datasets [33]. Therefore, there is a need for dedicated InR methods.

Our preliminary works on applying InR in OCC [32] showed the potential of this approach. However, this study was limited only to the nearest neighbor data description model and two simple instance reduction techniques. Additionally, we confronted issues with some datasets in which some methods did not provide enough instances to perform the classification step. Therefore, we concluded that there is a need to develop instance reduction methods that can
work with any one-class model and are natively tailored to the specific nature of learning in the absence of counterexamples.

In this work, we propose to tackle the issue of InR for OCC from two different perspectives and propose flexible methods that can be used by any one-class classifier.

Our first proposal stems from the intuitive extension of existing InR methods to learning in the absence of counterexamples. We introduce an universal framework for adapting any InR technique to OCC by artificial counterexamples generation and overlapping data cleaning. This way one may transform a given one-class problem into a binary one by creating outliers and then apply any selected InR technique. Generating artificial counterexamples has been used so far in the process of training one-class classifiers [26], but not during the one-class preprocessing phase. This approach could be viewed as a data-level solution, as we modify our training data to allow unaltered usage of any InR algorithm from the literature.

The second proposal is a specific approach tailored to the nature of one-class classification. We introduce a novel modification of evolutionary InR technique that is based on differential evolution [15] and more concretely on a memetic algorithm named SFLSDE (Scale factor local search differential evolution) [39]. By using a fully unsupervised consistency measure for evaluating set of instances during each iteration we lift the requirement for counterexamples during the reduction phase. Thus, our second proposal is a pure one-class algorithm. We present filter and wrapper modes for our method. This allows user to choose between reduced complexity and a more general solution or a more costly search procedure in order to find a set of instances better fitting the specific one-class learner. This approach could be viewed as an algorithm-level solution, as we modify a specific InR approach to use a criterion suitable for OCC, while leaving our data unaltered.

We present a family of data-level and algorithm-level InR methods for OCC and validate their usefulness and impact on training set reduction, classification accuracy and recognition time on the basis of thorough computational experiments. Such a comparison allows us to gain an insight into how we can reduce the size of the training set in the absence of counterexamples, while maintaining or even improving the obtained predictive performance.

The rest of the manuscript is organized as follows: Next section gives necessary background on OCC and InR. Section 3 discusses the importance of InR in OCC, while Sect. 4 describes in detail the proposed methodologies. Section 5 presents the evaluation of examined algorithms in a carefully designed experimental study, while final section concludes the paper.

2 Related works

This section provides the necessary background for the remainder of the paper.

2.1 One-class classification

OCC works under the assumption that during the classifier training stage objects originating only from a single class are available [28,38]. We name this specific class as the target class (concept) and denote it as $\omega_T$. The aim of OCC is to derive a decision boundary enclosing all available (or relevant) training objects from $\omega_T$. Thus, we achieve a data (concept) description. As the training procedure is carried out with the usage of only objects from a given class, we may refer to OCC as learning in the absence of counterexamples.

In OCC scenario, during the classification phase, new objects may appear. They may be new instances from the target class or come from previously unknown distribution(s) that
are outside of $\omega_T$. Such objects must be rejected by an one-class classifier, as they may be potentially undesirable, harmful or dangerous. We name them as outliers and denote them by $\omega_O$.

A proper OCC method must display good generalization properties in order not to be overfitted on $\omega_T$, and good discrimination abilities to achieve a high rejection rate on $\omega_O$. While this task may seem quite similar to binary classification (having positive and negative classes), the primary difference lies in the training procedure of a classifier. In the standard binary problems, we may expect objects from the other classes to predominantly come from one direction (the distribution of the given class). In OCC the target class should be separated from all the possible outliers, without any knowledge where outliers may appear. This leads to a need for the decision boundary to be estimated in all directions in the feature space around the target class.

One may distinguish four main families of OCC classifiers present in the relevant literature:

- **Density-based** methods aim at capturing a distribution of the target class. During prediction phase, a new object is then compared with the estimated distribution and a decision is made on the basis of its resemblance. This approach suffers from a significant limitation, as it requires a high number of available objects from the target class and assumes a flexible density model [45].

- **Reconstruction-based** methods are rooted in clustering and data modeling. They aim at capturing the structure of the target class and during prediction phase check if new instances will fit into these structures. If the similarity level is below a given threshold, the new instance is labeled as outlier [46].

- **Boundary-based** methods concentrate on estimating only the enclosing boundary for the target class, assuming that such a boundary will be a sufficient descriptor [3,48]. They try to find the optimal size of the volume enclosing the given training objects, as one that is too small can lead to an overtrained model, while one that is too big may lead to an extensive acceptance of outliers into the target class.

- **Ensemble-based** methods [43,58] propose a more flexible data description by utilizing several base classifiers. By combining mutually complementary learners one may achieve a better coverage of the target class, especially when dealing with complex distributions [11]. This requires a proper exploitation of competence areas of each base classifier, like unique model properties (for heterogeneous ensembles [40]), or diversified inputs (for homogeneous ensembles [55]). Additionally, often a classifier selection/ensemble pruning step must be conducted to choose the most complementary and effective learners from the available pool [31].

### 2.2 Instance reduction in standard classification

This section provides a formal definition about InR techniques and its current trends. A formal notation of the InR problem is as follows: Let $\mathcal{T}R$ be a training dataset and $\mathcal{T}S$ a test set; they are formed by a determined number $n$ and $t$ of samples, respectively. Each sample $x_m$ is a tuple $(x_1^m, x_2^m, \ldots, x_d^m, \omega_i)$, where $x_p^m$ is the value of the $p$th feature of the $i$th sample. This sample belongs to a class $\omega_i$. For the $\mathcal{T}R$ set the class $\omega_j$ is known, while it is unknown for $\mathcal{T}S$.

InR techniques aim to reduce the available training set $\mathcal{T}R = \{ (x_1, \omega_1), \ldots, (x_n, \omega_n) \}$ of labeled instances to a smaller set of instances $\mathcal{R}S = \{ (x_1^*, \omega_1^*), \ldots, (x_r^*, \omega_r^*) \}$, with $r < n$ and each $x_i^*$ either drawn from $\mathcal{T}R$ or artificially constructed. The set $\mathcal{R}S$ is later used to train the classifier, rather than the entire set $\mathcal{T}R$, to eventually classify the test set $\mathcal{T}S$. Thus, the
instances of RS should be efficiently computed to represent the distributions of the classes and to discern well when they are used to classify the training objects.

Most of the existing InR models have been designed and combined with the nearest neighbor classifier [10]. As a lazy learning algorithm [1] it classifies unseen instances to the class to which the majority of their $k$ nearest neighbors in the training set belongs based on a certain distance or similarity measure [9]. Thus, the reduction of the training set allows us to alleviate the low computational efficiency and high storage requirement of this classifier. Many works extended the application of instance reduction techniques to other classifiers [8] and different domains, such as imbalanced classification [20]. High computational complexity of InR, especially in case of processing large-scale datasets, inspired the development of efficient distributed architectures for this task [13,54].

As we stated before, InR is usually divided into those approaches that are limited to select instances from TR, known as instance selection (InS) [17], and those that may generate artificial examples if needed, named as instance generation (InG) [51]. Both strategies have been deeply studied in the literature. Most of the recent proposals are based on evolutionary algorithms to select [16] or generate [27,53] an appropriate RS.

3 The role of instance reduction in one-class classification

In OCC the quality of training data has a direct influence on the estimated data description. If we deal with well-sampled and representative set of examples, then we can assume that it will reflect the true target class distribution. This will allow us to train a classifier displaying at the same time good generalization over the target concept and high discriminative power against outliers, regardless of their nature. This is, however, an idealized scenario.

In practice, we often deal with uncertain or contaminated training sets in OCC [12]. This means that some objects can be influenced by feature or class label noise, thus offering misleading information regarding the nature of the target concept. When a one-class classifier is being trained on such objects it will output an overestimated decision boundary that will lead to increased acceptance of outliers during the prediction phase. Therefore, to improve the one-class classifier performance such objects should be removed beforehand [37].

Some of training objects may be redundant, not carrying any additional useful information for creating an effective data description. This is especially common in large-scale or big data, where abundance of training examples does not directly translate onto their usefulness. The presence of such objects significantly increases the training and testing times of one-class classifiers. A good example of this can be seen during training one-class methods based on support vector machines [48]. Here one is interested only in objects that have high potential of becoming future support vectors [59]. Thus, reducing the number of potential candidates will speed up the training procedure.

One must also be aware of the possible negative results of InR in OCC. Firstly, InR methods are unadvised for density-based one-class classifiers. This family of algorithms requires a large sample of the target class in order to properly estimate its density in the feature space. Hence, reducing the size of the training set will negatively impact the quality of the estimation procedure.

From the point of view of any one-class classifier, preserving the original structure of the target class is of uttermost importance. When the training set is subject to a significant level of reduction, the probability of one-class classifier overfitting to a small target concept increases. This will in turn lead to deterioration of generalization capability. One cannot assume that
only objects located on the borders of the target concept are important, as outliers may also overlap with the target concept. Therefore, it is highly important to maintain the original object structure, concentrating on removing only redundant or noisy cases. Such a situation is depicted in Fig. 1.

Finally, as in standard classification tasks, InR is a trade-off between spending additional computational effort for training set reduction versus achieving a speedup during the classification phase. As we assume that the classifier, once trained, will be extensively used for continuous decision making; thus, improving the classification speed is the priority for us.

Instance reduction for OCC has so far been rarely addressed in the literature. Angiulli [2] introduced a prototype-based domain description rule that is similar to standard nearest neighbor-based one-class classifier, but exploits only a selected subset of the training set. Cabral and de Oliveira [6] proposed to analyze every limit of all the feature dimensions to find the true border which describes the normal class. Their method simulates the novelty class

Fig. 1  Examples of instance reduction in OCC.  
(a) A one-class classifier trained on complete set of target class objects.  
(b) A one-class classifier trained on incorrectly reduced set of objects. Here vital objects both within and on the borders of target concept were removed, thus leading to overfitted data description.  
(c) A one-class classifier trained on a reduced set of instances that preserves the characteristics of the original training set.  
(a) Full training set.  
(b) Incorrectly reduced instances.  
(c) Correctly reduced instances
by creating artificial prototypes outside the normal description and then uses minimal distance classification. Hadjadji and Chibani [25] described a special model of auto-associative neural network that can select samples for its own training in each iteration. This approach is not a strict instance reduction method as it aims only at removing noisy examples. One must notice that these literature proposals are specific only to a given classifier (neighbor-based or neural network). Therefore, there is a lack of universal InR methods for OCC that could be used with any type of one-class classifier.

4 Applying instance reduction to one-class classification

In this paper we propose two approaches for applying InR for OCC problems:
– A scheme for adapting existing InR solutions to one-class scenarios.
– A new method based on evolutionary InR with one-class evaluation criterion.

Both of these solutions will be described in detail in the following sections.

4.1 Adapting existing instance reduction methods to one-class classification

The first proposed approach is a general framework for adapting any existing InR method to OCC problems. As most of InR methods were designed for binary problems, we need to have access to examples from both classes in order to use them. As in real-life OCC scenarios outliers are not available during the training phase, one needs to find another way to have access to them. We propose to generate artificial counterexamples and use them to transform the input one-class problem into a binary one for the sake of InR procedure. This is a data-level approach that modifies the supplied training set in order to make it applicable to any InR methods, without modifying them in any way.

We will show now how to generate meaningful outliers when only objects from the target class are available. We assume that outliers are distributed uniformly around the target concept. For this, we can use a \( d \)-dimensional Gaussian distribution, following these steps [47]:

1. Generate a set of new outlier objects \( OS \) from a specified Gaussian distribution with zero mean and unit variance (making outliers appear all over the decision space similar to white noise):
   \[
   OS \sim \mathcal{N}(0, 1).
   \] (1)

2. For each artificial object calculate its squared Euclidean distance from the target concept origin. Please note that this squared distance \( r^2 \) is distributed as \( \chi^2 \) possessing \( d \) degrees of freedom. This allows to determine the overlap ratio between outliers and target concept.
   \[
   r^2 = \|x\|^2.
   \] (2)

3. Apply the cumulative distribution of \( \chi^2_d \) (denoted here as \( \chi^2_d \)) in order to transform the \( r^2 \) distribution into an uniform distribution \( \rho^2 \in [0; 1] \), which would lead to determining the new outliers distribution:
   \[
   \rho^2 = \chi^2_d (r^2) = \chi^2_d (\|x\|^2).
   \] (3)

4. Rescale the obtained distribution \( \rho^2 \) using \( r' = (\rho^2)^{\frac{1}{2}} \) that \( r' \) is distributed as \( r' \sim r^d \), where \( r \in [0; 1] \):
   \[
   r' = (\rho^2)^{\frac{1}{2}} = \left( \chi^2_d (\|x\|^2) \right)^{\frac{1}{2}}.
   \] (4)
5. Apply rescaling to all of artificially generated objects using obtained factor $r'$ in order to adapt them to the given decision space:

$$x' = r' \frac{x}{\|x\|}.$$  

(5)

These steps allow us to generate a set of artificial outliers uniformly in a $d$-dimensional hypersphere. They will be located in all directions surrounding the target class. This way we are able to transform the original one-class problem into a binary setting and apply any InR algorithm on it.

However, this outlier generation method has a significant drawback. There is a high probability that generated artificial outliers will highly overlap with the real target class objects. This can lead to improper selection of prototypes. We are interested in retaining examples that will maintain the best data description. Having outliers within the target class during InR procedure will lead to improper estimation of such potential boundary-building objects. Thus, we extend this artificial object generation procedure by a data cleaning step.

For this task we propose to apply Tomek links method [50]. Tomek link is a pair of closest neighbors from opposite classes. Let us assume that we have two objects $(x_i, x_j)$, where $x_i \in \omega_T$ and $x_j \in \omega_O$ and $d(x_i, x_j)$ is the distance between these two examples. Such a pair is called a Tomek link if there are no examples $x_k$ in the dataset that satisfy $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$. Therefore, if two instances form a Tomek link then one of them is a noisy/overlapping sample or both are located on class borders. One may use this as data cleaning method. Original attempts at this task removed objects from both classes. In the proposed approach we clean only artificial outliers, without affecting the original target class. This leads to removing generated overlapping outliers and obtaining a much better problem representation for InR method.

This data generation approach allows us to run InR methods on a binary set, consisting of target class examples and artificial outliers. As an output, we receive a reduced set of instances. We discard the artificial outliers and use only the ones selected for the target class.

We train a one-class classifier with the usage of reduced set of target class instances.

### 4.2 Evolutionary filter and wrapper methods for one-class instance reduction

The second proposed approach is a modification of existing InR algorithm that allows to tailor it for OCC scenarios. Here we aim at creating a method that will require only access to target class examples in order to evaluate their usefulness for the classification step. This lifts the requirement for generating artificial counterexamples.

Such an approach may be justified by the fact that introduced artificial objects may not reflect the true nature of outliers. As we do not have any information about the real characteristic of negative objects during the training step we are forced to make certain assumptions. This uncertainty regarding the generated counterexamples highly affects both the preprocessing and training phases.

This is an algorithm-level approach that modifies the specific used InR method, without any alterations on the set of training instances. Let us now present the details of selected InR method (Sect. 4.2.1) and how to change it to a one-class method (Sect. 4.2.2).

#### 4.2.1 Scale factor local search in differential evolution for instance reduction

The scale factor local search in differential evolution (SFLSDE) was originally proposed for continuous optimization problems in [39] and then adapted to perform instance reduction in
[53], showing to be one of the top performing InG methods in the experimental study. The method uses differential evolution [15], which follows a standard evolutionary framework, evolving a population of candidate solutions over a number of generations.

Specifically, it starts off with a population of NP candidate solutions, the so-called individuals, each of which encodes a reduced set of instances randomly taken from the training set. The size of each individual is typically given by an initial reduction rate parameter that determines the percentage of initial elements selected.

Later, mutation and crossover operators will create new individuals that contain a new positioning for the instances. For each individual $c_1$, mutation is achieved by randomly selecting two other individuals $c_1$ and $c_2$ from the current population. A new individual is created by increasing $c_1$ by the difference of $c_1$ and $c_2$, weighted by a scale factor $F > 0$. A number of different mutation operators exist, but we have chosen to use the DE/RandToBest/1 strategy, which makes use of the current fittest $c_{\text{best}}$ individual in the population. It increases $c_1$ by both the difference of the two randomly selected individuals and the difference of $c_i$ and $c_{\text{best}}$, weighting both terms by $F$. After mutation, crossover is performed, randomly modifying the mutated individual in certain positions. The crossover is guided by another user-specified parameter $Cr$.

When new individuals have been generated, a selection operator decides which of those and the previous individuals should survive in the population of the next generation. For this, the one nearest neighbor algorithm is applied classifying the instances of the training set using the instances encoded in each individual as reference. Thus, we obtain a measure of performance of every reduced set that allows us to take the best individuals to the next generation.

The key distinctive point of the SFLSDE algorithm is that it uses adaptive values for the $F$ and $Cr$ values. Specifically, each individual $c_i$ has their own custom values for $F_i$ and $Cr_i$, which are encoded within the individual and thus updated in each iteration. The idea of using custom values for $F$ and $Cr$ comes from [5], and the reasoning behind this is that the better the values of the control parameters that lead to better individuals, the more likely they are therefore to survive and propagate those parameter values $F_i$ and $Cr_i$ to the next generations.

When updating the scale factors $F_i$, two local searches can be used: the golden section search and hill climbing. We refer to [39,53] for further details.

As stated above, SFLSDE was canonically used with measures such as accuracy to evaluate the selected pool of instances. However, as in OCC we do not have access to counterexamples during preprocessing/training stages, we cannot use such measures.

4.2.2 Adapting SFLSDE to OCC

To adapt SFLSDE algorithm to OCC nature we propose to augment it with optimization criterion using the consistency metric. It is a fully unsupervised measure, requiring only access to target class objects. It indicates how consistent a given classifier is in rejecting a preset fraction $t$ of the target concept instances.

Let us assume that we have a one-class classifier $\Psi$ trained to reject the fraction $t$ of objects and a validation set $VS$. The mentioned $VS$ can be either supplied externally or separated from $TR$ with the constraint that $TR \cap VS = \emptyset$.

As in OCC we have access only to objects from the target concept during the training phase, the error of such classifier may be expressed as false negatives (FN) in the form of:

$$\text{FN} = \sum_{i=1}^{\mid VS \mid} \left(1 - I(F_{\omega_T}(x_i) \geq \theta)\right),$$

(6)
where $F_{\omega T}(x_i)$ is the classifier’s support for object $x_i$ belonging to the target class, $I(\cdot)$ is the indicator function, and $\theta$ is the used classification threshold (estimated or inputted by the user).

We can model this as $|\mathcal{V} S|$ binomial experiments. This will allow us to compute the expected number of rejected objects and the variance:

$$E[\text{FN}] = |\mathcal{V} S| t,$$
$$V[\text{FN}] = |\mathcal{V} S| t(1 - t).$$

When the number of rejected objects from the target class exceeds some bounds around this average (usually $2\sigma$ is used [49]), the examined classifier $\Psi$ can be deemed as inconsistent. We may say that $\Psi$ is inconsistent at level $t$ when:

$$\frac{\text{FN}}{|\mathcal{V} S|} \geq t + 2\sqrt{|\mathcal{V} S| t(1 - t)}.$$  

(9)

One may compute the consistency for an examined one-class classifier by comparing the rejected fraction $t$ with an estimate of the error on the target class $\text{FN}$:

$$\text{CONS}(\Psi) = \frac{|\text{FN}|}{|\mathcal{V} S|} - t.$$  

(10)

To use this approach, we need to have a number of models to be selected. This is provided by consequent iterations of SFLSDE that supplies us with varying set of target class instances. We order them by their complexity. The model for which the boundary could be estimated with highest reliability will be selected. Consistency measure prefers the classifier with highest complexity (in order to offer the best possible data description) that can still be considered as consistent.

Therefore, SFLSDE objective is to select such set of instances that maximize the value of one-class classifier’s consistency, while satisfying Eq. (9). This allows us to conduct the instance reduction using only objects from the target class.

We propose two versions of our OC-SFLSDE:

- **Filter.** In this version we use an one-class nearest neighbor (OCNN) approach for evaluating the consistency of a reduced set of instances in each SFLSDE iteration. The benefits of such an approach are the flexibility of the method (it allows us to select a set of instances once that can be used by any OCC classifier) and reduced computational complexity within each step (OCNN is a lazy classifier; thus, no training is required). The main weakness is the fact that filter method will not take into consideration some specific mechanism embedded in training procedures of certain one-class classifiers.

- **Wrapper.** In this version we use an user-specified one-class classifier for evaluating the consistency of reduced set of instances in each SFLSDE iteration. The benefit of the method is the selection of instance set reflecting the data description properties of selected classifier. Drawbacks include increased computational complexity (the need to retrain model for each iteration can be time-consuming for more complex classifiers) and requirement for a priori knowledge of which one-class classifier model will be used for the problem at hand (if an user wants to test more than a single classifier, then OC-SFLSDE must be run for each model independently).
5 Experimental study

The aim of this experimental study was to compare the proposed InR algorithms for the OCC task with respect to their reduction rates, influence on classification accuracy and classification times.

5.1 Datasets

To assess the proposed methods we have selected a total of 21 datasets from the UCI repository. Most of them are binary ones, where the majority class was used as the target concept and the minority class as outliers. For the training purposes we extract only target class object from the given training cross-validation fold, while for testing both target concept examples and outliers from respective test cross-validation fold are used. In case of contraceptive method choice and KDD cup 1999 datasets, we merged all but one classes as target concept and used the remaining one (with smallest number of samples) as outliers. Details of the chosen datasets are given in Table 1.

We must comment that using binary datasets to transform them into one-class problems has been a popular strategy used so far. This way we can simulate a single target concept and a possible outlier distributions with varying overlap ratios, noise or difficult separation boundaries. However, when using binary datasets the outlier distribution will originate from a single class and thus will not cover all areas of the decision space. Such an observed limitation should be a future starting point for developing new one-class benchmarks. However, this is beyond the scope of this paper and we will use here standard approach for evaluating one-class learners.

5.2 Methods

In our experiments, we have selected a number of popular InR methods [22]: 2 InS models, 2 InG and one hybrid InR model:

- The ENN algorithm [56] is an edition-based InS method in which an instance is removed if it does not agree with the majority of its $k$ nearest neighbors. As such, the reduction power of this method is limited to remove potential noisy examples.
- The DROP3 [57] procedure combines an edition stage with a decremental approach where the algorithm checks all the instances in order to find those instances which should be deleted. The reduction rate achieved by this InS technique is much higher than that by the ENN method, eliminating both noisy and unrepresentative examples.
- The ICPL [34] technique is an InG technique that integrates instances by identifying borders and merging those instances that are not located in these borders.
- The IPADE [52] algorithm is an iterative InG model that searches for the smallest reduced set of instances that represent the training data by performing an evolutionary optimization process.
- The SSMA-SFLSDE [53] is a hybrid InS and InG model, in which first the InS method SSMA determines the best quantity of instances per class for a reduced set and then the evolutionary method SFLSDE adjusts the positioning of such instances.

We have selected the three following boundary-based one-class classifiers to be used in our experiments:

- One-class nearest neighbor (OCNN) uses only distance to the first nearest neighbor. It works by comparing the distance from the new object to its nearest neighbor from the
### Table 1: Details of used datasets

| Lp. | Name                                      | Objects   | Features | Classes |
|-----|-------------------------------------------|-----------|----------|---------|
| 1.  | Climate model, simulation crashes         | 540 (494) | 18       | 2       |
| 2.  | Contraceptive method choice               | 1473 (629)| 9        | 3       |
| 3.  | Credit approval                           | 690 (307) | 15       | 2       |
| 4.  | Fertility                                 | 100 (88)  | 10       | 2       |
| 5.  | Haberman’s survival                       | 306 (225) | 3        | 2       |
| 6.  | Hepatitis                                 | 155 (123) | 19       | 2       |
| 7.  | Hill-Valley                               | 606 (305) | 101      | 2       |
| 8.  | Indian liver patient dataset              | 583 (416) | 10       | 2       |
| 9.  | Mammographic mass                         | 961 (516) | 9        | 2       |
| 10. | Musk (version 2)                          | 6598 (2521)| 168      | 2       |
| 11. | Ozone level detection (1 h)               | 2536 (2410)| 73       | 2       |
| 12. | Parkinsons                                | 197 (147) | 23       | 2       |
| 13. | Pima Indians diabetes                     | 768 (500) | 8        | 2       |
| 14. | Sonar, mines versus rocks                 | 208 (111) | 60       | 2       |
| 15. | Statlog (Heart)                           | 270 (150) | 13       | 2       |
| 16. | Wisconsin breast cancer (original)        | 699 (458) | 10       | 2       |
| 17. | MiniBooNE                                 | 130,065 (93,565) | 40   | 2       |
| 18. | Twitter buzz in social media              | 140,707 (112,932) | 77  | 2       |
| 19. | Skin segmentation                         | 245,057 (50,859) | 3   | 2       |
| 20. | Census-income                             | 299,285 (227,457) | 40  | 2       |
| 21. | KDD cup 1999 (DoS vs. rest)               | 494,020 (391,458) | 41  | 2       |

The number of objects in the target class is given in parentheses

training set (which consists of labeled target class examples) with the distance from this found nearest neighbor to its own nearest neighbor. This means that new object is accepted, if it satisfies the local density of objects in the target class.

- Minimum spanning tree data description (MST) [29] is a distance-based one-class classifier that uses minimum spanning tree structure constructed on the target class. New instances are classified on the basis of their distance (or similarity) to the closest edge of this tree.

- Support vector data description (SVDD) [48] is a technique that gives a closed boundary around the data in the form of a hypersphere. It is characterized by center $a$ and radius $R$. In its basic form it assumes that all objects from the training set must be enclosed by this hypersphere. Slack variables are used to remove inner outliers from the training set, while kernel function allows for searching a better representation in artificial spaces.

### 5.3 Setup

Detailed parameters of the used methods are given in Table 2. They are selected based on the setup returning the best averaged performance on examined datasets.

In order to carry out a thorough comparison, one needs to establish an experimental setup consisting of evaluation metrics, training/testing modes and statistical analysis [19]. We use the following tools on our framework:
Table 2 Details of algorithm parameters used in the experimental study

| Algorithm            | Parameters                                                                                           |
|----------------------|------------------------------------------------------------------------------------------------------|
| OCNN                 | distance = Euclidean metric                                                                          |
|                      | frac. rejected = 0.05                                                                                  |
| MST                  | max. path = 20                                                                                        |
|                      | frac. rejected = 0.05                                                                                  |
| SVDD [29,48]         | kernel type = RBF                                                                                     |
|                      | C = 5.0                                                                                               |
|                      | \( \gamma = 0.0045 \)                                                                                  |
|                      | parameter optimization = quadratic programming                                                          |
|                      | frac. rejected = 0.05                                                                                  |
| ENN [56]             | Number of neighbors = 3, Euclidean distance                                                           |
| DROP3 [57]           | Number of neighbors = 3, Euclidean distance                                                           |
| ICPL [34]            | Filtering method = RT2                                                                                |
| IPADE [52]           | Iterations of basic DE = 500, iterSFGSS = 8, iterSFHC = 20, Fl = 0.1, Fu = 0.9                       |
| SSMA-SFLSDE [53]     | PopulationSFLSDE = 50, IterationsSFLSDE = 500, iterSFGSS = 8, iterSFHC = 20, Fl = 0.1, Fu = 0.9        |
| OC-SFLSDE            | Population size = 50                                                                                  |
|                      | Initial reduction rate: 0.95                                                                          |
|                      | Iterations = 500                                                                                      |
|                      | iterSFGSS = 8                                                                                         |
|                      | iterSFHC = 20                                                                                         |
|                      | Fl = 0.1                                                                                              |
|                      | Fu = 0.9                                                                                              |
|                      | Mutation operator: RandToBest/1/Bin                                                                   |

- For evaluating classifiers we use the balanced accuracy metric [23] that is skew-insensitive:

\[
BAC (TP, FP, TN, FN) = \frac{1}{2} \left( \frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right).
\]

- Additionally, we check the reduction rate metric, which measures the reduction in storage requirements achieved by a InR algorithm:

\[
\text{ReductionRate} = 1 - \frac{|RS|}{TS}.
\]

- We use a tenfold cross-validation for training and testing.
- 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 Finner post hoc test for an \( 1 \times n \) comparison. It adjusts the value of \( \alpha \) in a step-down manner. We select it due to its high power [18]. Additionally, we examine obtained \( p \)-values in order to check how different the given two algorithms are.
- We use the Shaffer post hoc test to find out which of the tested classifiers are distinctive in an \( n \times n \) comparison. It is a modification of popular Holm’s procedure that provides
an increase in power at the cost of greater complexity of the testing procedure. We have selected it due to its effectiveness in multiple comparison tasks [19]. Additionally, we examine obtained $p$-values in order to check how different the given two algorithms are.

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

5.4 General comments on obtained results

Firstly, let us compare the two proposed approaches: based on adapting existing methods to OCC and using a consistency-based InR. Regardless of the classifier used, both BAC and statistical tests results clearly point out the superiority of the OC-SFLSDE solution. This can be explained by several factors. The adaptation scheme transforms an original one-class problem into a binary one, assuming a uniform distribution of outliers and generating artificial counterexamples. Despite using data cleaning procedures this approach may potentially lead to ill-defined decision boundaries that may mislead the instance reduction algorithm. One must remember that examined methods use nearest neighbor approach for evaluating the reduced set of instances. There, however, may not be a direct translation between decision boundaries estimated by binary nearest neighbor and actual shape of data description over the target class. Finally, by transforming a one-class task into a binary problem we overlook some difficulties embedded in the nature of OCC, such as overestimated decision boundaries, empty areas within the data description that lead to classifier’s incompetence, or internal noisy samples. These factors may result in a drop of the final accuracy observed for all types of one-class classifiers. The reduction rates displayed by each examined InR algorithm are given in Table 3.

To cope with these issues OC-SFLSDE (in both wrapper and filter versions) uses a consistency measure which does not require counterexamples. It allows to evaluate the complexity of classifier and how well it captures the target concept without actually being overfitted on training data. Experimental results show that this solution is closer to the nature of one-class problems and offers excellent performance both in filter and wrapper modes. Interestingly OC-SFLSDE is characterized by a slightly lower reduction rates than other methods. This shows that OCC problems cannot be too strongly reduced and a well-represented training set is crucial for proper data description. Additionally, OC-SFLSDE in some cases allows to improve the classification accuracy with respect to the original training set. This shows that by using the proposed hybrid consistency-based solution we are able to filter uncertain or noisy objects that may harm the one-class classifier being trained. While some of the binary methods adopted to OCC offer higher reduction rates and thus increased classification speedup, OC-SFLSDE returns the best trade-off in terms of classification efficacy and time.

5.5 Results for one-class nearest neighbor

The results obtained by examined InR algorithms with OCNN are presented in Table 4, while the outcomes of post hoc statistical test are presented in Table 5.

OCNN was the only classifier that displayed good performance with methods other than the proposed filter approach. For some of datasets ENN and SSMA-SFLSDE approaches augmented with artificial data were able to deliver satisfactory accuracy as well. This can be explained by the minimal distance classification approach, similar in both binary and one-class scenarios. Hence, the evaluation of standard instance selection methods based on two-class nearest neighbor was able to locate useful examples for being preserved in the reduced training set. However, our filter-based solution delivered the best performance in 20 datasets. In this scenario filter and wrapper modes were equivalent due to the base classifier used.
### Table 3 Reduction rates (%) obtained by different examined instance reduction approaches

| Dataset                  | ENN   | DROP3 | ICPL  | IPADE | SSMA-SFLSDE | OCC-filter | OCC-wrapper-NN | OCC-wrapper-MST | OCC-wrapper-SVDD |
|--------------------------|-------|-------|-------|-------|-------------|------------|----------------|-----------------|-----------------|
| Climate                  | 37.04 | 94.07 | 88.89 | 98.89 | 95.19       | 81.67      | 81.67          | 80.58           | 80.89           |
| Contraceptive            | 38.86 | 74.18 | 89.01 | 99.46 | 97.01       | 80.25      | 80.25          | 78.94           | 47.28           |
| Credit                   | 20.00 | 88.99 | 90.33 | 99.13 | 99.42       | 84.89      | 84.89          | 83.92           | 81.28           |
| Fertility                | 34.00 | 94.00 | 90.00 | 96.00 | 94.00       | 88.73      | 88.73          | 88.12           | 86.73           |
| Haberman                 | 41.18 | 86.93 | 87.59 | 98.04 | 98.69       | 85.28      | 85.28          | 82.71           | 83.72           |
| Hepatitis                | 71.43 | 71.43 | 79.23 | 93.51 | 92.21       | 73.08      | 73.08          | 70.86           | 71.24           |
| Hill-Valley              | 53.80 | 68.98 | 71.62 | 99.34 | 96.70       | 75.29      | 75.29          | 74.09           | 72.06           |
| Indian                   | 39.18 | 78.35 | 81.45 | 98.97 | 98.63       | 75.82      | 75.82          | 74.51           | 72.88           |
| Mammographic             | 25.21 | 82.92 | 90.00 | 99.38 | 98.54       | 81.03      | 81.03          | 78.54           | 77.90           |
| Musk                     | 74.24 | 97.73 | 98.31 | 99.79 | 99.36       | 92.61      | 92.61          | 91.30           | 91.82           |
| Ozone                    | 20.27 | 97.87 | 97.85 | 99.68 | 95.11       | 92.30      | 92.30          | 88.58           | 85.60           |
| Parkinsons               | 36.73 | 80.61 | 80.62 | 94.90 | 94.90       | 78.27      | 78.27          | 76.37           | 75.22           |
| Pima                     | 30.99 | 79.43 | 80.99 | 98.70 | 96.61       | 73.68      | 73.68          | 70.49           | 68.79           |
| Sonar                    | 31.73 | 73.08 | 75.97 | 97.12 | 91.35       | 69.42      | 69.42          | 68.76           | 68.03           |
| Statlog                  | 31.11 | 89.63 | 85.93 | 96.30 | 96.30       | 73.12      | 73.12          | 72.26           | 73.12           |
| Wisconsin                | 15.80 | 97.70 | 97.13 | 99.14 | 99.14       | 54.83      | 54.83          | 52.64           | 53.14           |
| MiniBooNE                | 19.37 | 93.54 | 94.19 | 97.54 | 97.36       | 85.49      | 85.49          | 82.71           | 81.09           |
| Twitter buzz in social media | 27.43 | 95.28 | 97.03 | 99.01 | 99.01       | 87.92      | 87.92          | 83.99           | 80.35           |
| Skin segmentation        | 31.54 | 91.59 | 91.59 | 94.12 | 97.35       | 81.46      | 81.46          | 81.20           | 80.93           |
| Census-income            | 41.54 | 97.42 | 95.89 | 99.11 | 98.89       | 79.82      | 79.82          | 73.10           | 71.69           |
| KDD cup 1999             | 63.20 | 89.19 | 90.86 | 93.02 | 93.02       | 83.19      | 83.19          | 82.28           | 81.93           |
### Table 4  BAC (%) results for different examined instance reduction methods with one-class nearest neighbor classifier

| Dataset          | No reduction | ENN | DROP3 | ICPL | IPADE | SSMA-SFLSDE | OCC-filter |
|------------------|--------------|-----|-------|------|-------|-------------|------------|
| Climate          | 75.54        | 75.24 | 74.68 | 72.98 | 70.74 | 75.24       | 78.23      |
| Contraceptive    | 89.76        | 89.03 | 88.63 | 87.11 | 85.93 | 89.12       | 90.76      |
| Credit           | 79.63        | **79.38** | 78.75 | 76.90 | 75.69 | 76.84       | **79.38**  |
| Fertility        | 86.89        | 86.03 | 85.64 | 86.03 | 85.9  | 86.04       | **87.89**  |
| Haberman         | 70.98        | 68.78 | 68.54 | 68.37 | 62.90 | 68.90       | **70.98**  |
| Hepatitis        | 67.43        | **67.22** | 67.01 | 64.37 | 61.35 | 66.31       | 67.00      |
| Hill-Valley      | 86.94        | 86.81 | 86.67 | 86.12 | 85.15 | 86.52       | **86.94**  |
| Indian           | 93.41        | **93.41** | 93.12 | 92.53 | 88.79 | 92.80       | **93.41**  |
| Mammographic     | 87.37        | 86.21 | 86.92 | 85.11 | 84.06 | 86.80       | **87.04**  |
| Musk (V2)        | 71.38        | 70.79 | 70.54 | 69.79 | 68.94 | 70.50       | **71.27**  |
| Ozone            | 78.49        | 76.75 | 77.32 | 75.93 | 75.93 | 77.83       | **77.96**  |
| Parkinsons       | 67.39        | 67.04 | 66.78 | 67.02 | 67.12 | 67.14       | **67.39**  |
| Pima             | 90.52        | **89.78** | 88.43 | 88.75 | 86.42 | **89.78**   | **89.78**  |
| Sonar            | 82.66        | 81.66 | 81.49 | 78.34 | 77.83 | 81.66       | **82.30**  |
| Statlog          | 68.06        | 66.18 | 66.82 | 64.32 | 62.11 | 67.06       | **70.02**  |
| Wisconsin        | 91.36        | 90.89 | 90.15 | 89.74 | 88.12 | 90.89       | **93.03**  |
| MiniBooNE        | 72.89        | 65.35 | 63.72 | 60.04 | 67.83 | 68.22       | **76.44**  |
| Twitter buzz in  | 87.43        | 82.21 | 80.10 | 76.39 | 82.78 | 83.19       | **88.72**  |
| social media     |              |      |      |      |      |             |            |
| Skin segmentation| 92.91        | 84.90 | 85.18 | 81.99 | 84.78 | 86.11       | **93.27**  |
| Census-income    | 71.90        | 59.85 | 60.14 | 57.72 | 63.21 | 63.99       | **70.01**  |
| KDD cup 1999     | 72.88        | 65.31 | 63.20 | 58.84 | 64.25 | 66.96       | **74.06**  |
| Avg. rank        | 2.69         | 3.69  | 4.81  | 5.72  | 2.84  | 1.25        |             |

Please note that for this base classifier both filter and wrapper methods are identical; therefore, we present results for only one of them.

Best results are bolded.

### Table 5  Finner test for comparison between the proposed filter instance reduction and reference methods for one-class nearest neighbor classifier

| Hypothesis                      | p value          |
|---------------------------------|------------------|
| OCC-filter versus no reduction  | = 0.197458       |
| OCC-filter versus ENN           | > 0.026588       |
| OCC-filter versus DROP3         | > 0.000174       |
| OCC-filter versus ICPL          | > 0.000000       |
| OCC-filter versus IPADE         | > 0.000000       |
| OCC-filter versus SSMA-SFLSDE   | > 0.013132       |

Symbol ‘=’ stands for classifiers without significant differences, ‘>’ for situation in which OCC_f method is superior, and ‘<’ for vice versa.

Statistical testing confirms the superiority of this approach to reference ones. Additionally, it shows that there are no statistically significant differences between this instance reduction method and classifier trained on the full dataset.
5.6 Results for minimum spanning tree data description

The results obtained by examined InR algorithms with MST are presented in Table 6, while the outcomes of post hoc statistical test are presented in Table 7.

MST works significantly better with OC-SFLSDE than with the remaining methods. Here we are able to analyze two operating modes: as a filter and as a wrapper. In the first case we can see that obtained results are statistically not worse than when using full training set. This shows that despite the lack of knowledge which classifier will be used the filter mode can still deliver meaningfully selected training samples. However, Shaffer test pointed out that there are no significant differences between this filter method and ENN approach using artificial counterexamples. For the wrapper mode situation changes in favor of the hybrid evolutionary approach. Here we are able not only to significantly outperform all other methods (including filter version), but also get improved results in comparison with using the entire training set. Using wrapper solution allows us to evaluate the usefulness of training samples for the minimum spanning tree construction and eliminate redundant, similar or noisy samples, which contributes to decreased complexity and improved accuracy. Additional cost connected to training MST at each iteration pays off in the improved final model. When comparing classification times we can see that both filter and wrapper solutions achieve similar reduction rates and response times of the trained classifier.

5.7 Results for support vector data description

The results obtained by examined InR algorithms with SVDD are presented in Table 8, while the outcomes of post hoc statistical test are presented in Table 9.

SVDD is characterized by the most atypical behavior from the three examined methods. Here, for some of the datasets, the standard InR techniques returned too small training set to build a one-class support vector classifier. This is because they use the nearest neighbor approach which has no lower bound on the size of the training set, while methods based on support vectors require a certain amount of samples for processing. SVDD cannot estimate an enclosing boundary with too few samples, as in cases when some InR methods return less than five training samples. Our proposed method once again returns the best performance, especially in the wrapper mode. This allows us to embed the SVDD training procedure with robustness to internal outliers that is of significant benefit to this classifier. SVDD displays the highest gain in classification efficacy from all of examined classifiers when wrapper selection is applied.

5.8 Impact on the computational complexity

The average runtime of examined InR methods is presented in Fig. 2. Average classification times before and after reduction for examined classifiers with respect to small datasets (< 10,000 instances) are depicted in Figs. 3, 4 and 5, while for large-scale datasets (≥ 100,000 instances) are depicted in Fig. 6. The red points stand for a baseline classification time, while remaining points stand for reduced classification times after applying examined InR algorithms. This allows us to examine the computational gains from using simplified one-class learners.

In case of OCNN, we can see that our filter is a highly suitable method for OCNN, increasing the classification speed while not decreasing its accuracy. In case of MST, when comparing classification times we can see that both filter and wrapper solutions achieve similar reduction rates and response times of the trained classifier. As for SVDD, we can
| Dataset                  | No reduction | ENN   | DROP3 | ICPL  | IPADE | SSMA-SFLSDE | OCC-filter | OCC-wrapper-MST |
|--------------------------|--------------|-------|-------|-------|-------|-------------|------------|----------------|
| Climate                  | 78.36        | 77.18 | 60.12 | 65.53 | 64.89 | 68.54       | 77.45      | 79.45          |
| Contraceptive            | 87.75        | 86.56 | 76.32 | 79.32 | 80.17 | 80.22       | 86.89      | 87.75          |
| Credit                   | 82.45        | 81.36 | 62.95 | 69.75 | 68.65 | 75.28       | 81.64      | 83.53          |
| Fertility                | 87.46        | 86.89 | 80.41 | 82.41 | 82.24 | 77.49       | 87.46      | 87.46          |
| Haberman                 | 70.13        | 68.53 | 53.58 | 57.74 | 58.05 | 61.93       | 69.15      | 70.02          |
| Hepatitis                | 71.38        | 70.36 | 59.37 | 66.12 | 64.74 | 64.50       | 69.36      | 71.16          |
| Hill-Valley              | 84.23        | 83.45 | 76.96 | 77.43 | 78.32 | 77.94       | 84.23      | 85.41          |
| Indian                   | 94.14        | 93.72 | 78.24 | 78.94 | 79.27 | 84.39       | 94.51      | 94.51          |
| Mammographic             | 85.89        | 85.02 | 70.53 | 77.23 | 75.51 | 81.05       | 85.11      | 85.11          |
| Musk (V2)                | 73.76        | 72.22 | 55.85 | 58.34 | 57.84 | 62.63       | 72.88      | 73.41          |
| Ozone                    | 77.97        | 77.13 | 69.93 | 71.52 | 72.22 | 76.86       | 78.48      | 80.39          |
| Parkinsons               | 71.84        | 71.08 | 58.47 | 59.32 | 60.78 | 60.94       | 72.46      | 72.46          |
| Pima                     | 92.65        | 91.94 | 71.04 | 77.34 | 75.88 | 81.56       | 92.07      | 92.43          |
| Sonar                    | 81.87        | 80.83 | 63.68 | 67.80 | 69.97 | 73.74       | 81.87      | 81.87          |
| Statlog                  | 67.27        | 64.89 | 50.02 | 54.53 | 51.68 | 61.89       | 68.37      | 70.16          |
| Wisconsin                | 93.53        | 93.21 | 84.36 | 87.27 | 87.09 | 90.11       | 93.21      | 93.53          |
| MiniBooNE                | 74.59        | 62.18 | 60.94 | 58.59 | 63.93 | 65.42       | 74.81      | 77.58          |
| Twitter buzz in social media | 88.92      | 80.05 | 77.92 | 74.11 | 80.22 | 80.44       | 85.72      | 90.01          |
| Skin segmentation        | 93.44        | 81.15 | 82.03 | 77.22 | 81.46 | 82.02       | 89.17      | 94.58          |
| Census-income            | 72.55        | 58.64 | 59.23 | 55.97 | 61.89 | 62.36       | 70.99      | 70.99          |
| KDD cup 1999             | 72.53        | 61.18 | 59.23 | 54.38 | 62.05 | 64.91       | 72.55      | 74.03          |
| Avg. rank                | 2.91         | 6.94  | 5.31  | 5.37  | 4.37  | 1.94        | 1.94       | 1.16           |

Best results are bolded
Table 7  Shaffer test for comparison between the proposed filter and wrapper instance reduction methods and reference approaches for minimum spanning tree classifier

| Hypothesis                      | p value       | Hypothesis                      | p value       |
|---------------------------------|---------------|---------------------------------|---------------|
| OCC-filter versus no reduction  | = 0.614851    | OCC-wraper versus no reduction  | > 0.041704    |
| OCC-filter versus ENN           | = 0.257327    | OCC-wraper versus ENN           | > 0.019860    |
| OCC-filter versus DROP3         | > 0.000522    | OCC-wraper versus DROP3         | > 0.000000    |
| OCC-filter versus ICPL          | > 0.000000    | OCC-wraper versus ICPL          | > 0.000000    |
| OCC-filter versus IPADE         | > 0.000000    | OCC-wraper versus IPADE         | > 0.000000    |
| OCC-filter versus SSMA-SFLSDE   | > 0.00709     | OCC-wraper versus SSMA-SFLSDE   | > 0.000016    |
| OCC-filter versus OCC-wraper    | < 0.044270    |                                 |               |

Symbol ‘=’ stands for classifiers without significant differences, ‘>’ for situation in which the first analyzed method is superior, and ‘<’ for vice versa.
Table 8  BAC (%) results for different examined instance reduction methods with support vector data description classifier

| Dataset               | No reduction | ENN  | DROP3 | ICPL  | IPADE | SSMA-SFLSDE | OCC-filter | OCC-wrapper-SVDD |
|-----------------------|--------------|------|-------|-------|-------|-------------|------------|------------------|
| Climate               | 81.73        | 81.73| –     | 72.84 | –     | –           | 82.38      | 84.51            |
| Contraceptive         | 85.38        | 84.79| 79.24 | 74.62 | –     | 65.27       | 84.90      | 85.38            |
| Credit                | 83.90        | 82.58| 78.46 | 63.56 | –     | –           | 83.72      | 84.43            |
| Fertility             | 82.61        | 79.71| 74.07 | –     | –     | –           | 81.12      | 81.94            |
| Haberman              | 72.98        | 71.66| 69.31 | 59.36 | –     | –           | 71.35      | 72.56            |
| Hepatitis             | 70.14        | 67.48| 62.98 | 63.51 | –     | –           | 69.54      | 70.00            |
| Hill-Valley           | 77.36        | 75.38| 70.93 | 71.68 | –     | –           | 76.30      | 77.04            |
| Indian                | 93.62        | 92.54| 86.08 | 82.90 | –     | –           | 92.98      | 93.18            |
| Mammographic          | 87.92        | 87.44| 83.11 | 59.98 | –     | –           | 87.44      | 89.03            |
| Musk (V2)             | 71.21        | 70.03| 64.82 | 67.82 | 48.29 | 58.58       | 70.52      | 71.06            |
| Ozone                 | 75.28        | 74.00| 69.05 | 54.29 | 44.89 | 54.48       | 76.38      | 77.81            |
| Parkinsons            | 74.02        | 71.33| 66.43 | 62.74 | –     | –           | 72.35      | 73.60            |
| Pima                  | 93.26        | 91.27| 80.27 | 82.73 | –     | 51.57       | 92.18      | 93.26            |
| Sonar                 | 79.53        | 78.18| 67.98 | 71.04 | –     | –           | 79.53      | 81.13            |
| Statlog               | 69.32        | **69.90**| – | 56.68 | –     | –           | **69.90** | 69.90            |
| Wisconsin             | 94.65        | **94.65**| – | –     | –     | –           | **94.65** | **94.65**        |
| MiniBooNE             | 74.23        | 55.77| 54.25 | 44.46 | 54.38 | 57.02       | 70.05      | 76.93            |
| Twitter buzz in       | 89.41        | 69.87| 68.42 | 59.93 | 62.17 | 63.03       | 83.98      | **91.64**        |
| social media          |              |      |       |       |       |             |            |                  |
| Skin segmentation     | 93.72        | 68.91| 70.01 | 54.81 | 64.27 | 67.72       | 90.02      | **95.03**        |
| Census-income         | 76.48        | 42.20| 41.81 | 39.97 | 38.55 | 39.06       | 69.94      | **79.11**        |
| KDD cup 1999          | 73.87        | 53.99| 51.95 | 46.63 | 48.63 | 51.89       | 72.16      | **75.86**        |
| Avg. rank             | 2.78         | 4.66 | 4.72  | 6.47  | 6.16  | 2.09        | 1.12       |                  |

– stands for situation in which the number of instances after reduction was too small to efficiently train SVDD classifier

Best results are bolded
Table 9  Shaffer test for comparison between the proposed filter and wrapper instance reduction methods and reference approaches for support vector data description classifier

| Hypothesis               | p value   | Hypothesis               | p value   |
|--------------------------|-----------|--------------------------|-----------|
| OCC-filter versus no reduction | = 0.162701 | OCC-wraper versus no reduction | > 0.042915 |
| OCC-filter versus ENN     | = 0.174658 | OCC-wraper versus ENN     | > 0.027993 |
| OCC-filter versus DROP3   | > 0.000000 | OCC-wraper versus DROP3   | > 0.000000 |
| OCC-filter versus ICPL    | > 0.000003 | OCC-wraper versus ICPL    | > 0.000001 |
| OCC-filter versus IPADE   | > 0.000003 | OCC-wraper versus IPADE   | > 0.000000 |
| OCC-filter versus SSMA-SFLSDE | > 0.000989 | OCC-wraper versus SSMA-SFLSDE | > 0.000000 |
| OCC-filter versus OCC-wraper | < 0.028299 |                           |           |

Symbol ‘=’ stands for classifiers without significant differences, ‘>’ for situation in which the first analyzed method is superior, and ‘<’ for vice versa.
Fig. 2 Average reduction times of examined InR methods over all datasets

Fig. 3 Average classification times in seconds for one-class nearest neighbor classifier before and after reduction by examined instance reduction methods over small datasets

see a slightly bigger difference in reduction rates and classification times between filter and wrapper approaches. Using wrapper method results in slightly bigger training set with potentially higher number of support vectors (which explains the classification time). Still SVDD is characterized by the lowest classification time from all classifiers examined. These additional vectors have a clear positive influence on the classification procedure, which allows us to conclude that wrapper offers the best trade-off between reduction rates and classification performance for one-class support vector classifiers.
**Fig. 4** Average classification times in seconds for minimum spanning tree classifier before and after reduction by examined instance reduction methods over small datasets.

**Fig. 5** Average classification times in seconds for support vector data description classifier before and after reduction by examined instance reduction methods over small datasets. Time equal to 0 stands for a situation in which the dataset after reduction was too small to train a classifier.
Fig. 6 Average classification times in seconds for (left) one-class nearest neighbor, (center) minimum spanning tree and (right) support vector data description classifiers before and after reduction by examined instance reduction methods over large-scale datasets

6 Conclusions and future works

In this paper we have discussed the usability of instance reduction methods for one-class classification. We showed that a carefully conducted InR can lead to a significant lowering of OCC complexity and often to improved classification performance. We have proposed two approaches to applying InR for OCC. Firstly, an universal framework was introduced that allowed to transform any existing InR method to one-class version. It used uniform generation of artificial objects combined with data cleaning procedure to remove overlap between classes. This way we obtained a binary dataset that can be supplied to any conventional InR algorithm. After reduction artificial outliers were discarded and one-class classifier was trained using a reduced set of target concept instances.

We have identified potential shortcoming of the methods relying on artificial data generation and proposed a second solution, native to OCC characteristics. It used SFLSDE, a method based on differential evolution, to select training samples. We have augmented it with a consistency measure in order to evaluate the set of instances without a need for counterexamples. Filter and wrapper versions were proposed, varying in complexity and adaptability to selected classification model. Thorough experimental study backed up by a statistical analysis proved the high usefulness of the proposed solutions regardless of the base classifier used. We showed that consistency-based InR significantly outperforms methods that require artificial counterexamples.

Obtained results encourage us to continue work on instance reduction for one-class classification. In future, we plan to develop ensemble learning techniques based on diverse subsets of selected examples [24].

References

1. Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. Mach Learn 6(1):37–66
2. Angiulli F (2012) Prototype-based domain description for one-class classification. IEEE Trans Pattern Anal Mach Intell 34(6):1131–1144
3. Bicego M, Figueiredo MAT (2009) Soft clustering using weighted one-class support vector machines. Pattern Recogn 42(1):27–32
4. Bolón-Canedo V, Sánchez-Marino N, Alonso-Betanzos A (2015) Feature selection for high-dimensional data. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer, pp 1–132. ISBN 978-3-319-21857-1.
5. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657. https://doi.org/10.1109/TEVC.2006.872133
6. Cabral GG, de Oliveira ALI (2011) A novel one-class classification method based on feature analysis and prototype reduction. In: Proceedings of the IEEE international conference on systems, man and cybernetics, Anchorage, October 9–12, 2011, pp 983–988
7. Cano A, Ventura S, Cios KJ (2017) Multi-objective genetic programming for feature extraction and data visualization. Soft Comput 21(8):2069–2089
8. Cano JR, Herrera F, Lozano M (2003) Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study. IEEE Trans Evol Comput 7(6):561–575
9. Chen Y, Garcia EK, Gupta MR, Rahimi A, Cazzanti L (2009) Similarity-based classification: concepts and algorithms. J Mach Learn Res 10:747–776
10. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory 13(1):21–27
11. Cyganek B (2012) One-class support vector ensembles for image segmentation and classification. J Math Imaging Vis 42(2–3):103–117
12. Cyganek B, Wiatr K (2011) Image contents annotations with the ensemble of one-class support vector machines. In: NCTA 2011—proceedings of the international conference on neural computation and applications [part of the international joint conference on computational intelligence IJCCI 2011], Paris, 24–26 October, 2011, pp 277–282
13. Czarnowski I (2010) Prototype selection algorithms for distributed learning. Pattern Recogn 43(6):2292–2300
14. Czarnowski I (2012) Cluster-based instance selection for machine classification. Knowl Inf Syst 30(1):113–133
15. Das S, Suganthan P (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31
16. García S, Cano JR, Herrera F (2008) A memetic algorithm for evolutionary prototype selection: a scaling up approach. Pattern Recogn 41(8):2693–2709
17. García S, Derrac J, Cano J, Herrera F (2012) Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans Pattern Anal Mach Intell 34(3):417–435
18. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180(10):2044–2064
19. García S, Herrera F (2008) An extension on “Statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. J Mach Learn Res 9:2677–2694
20. García S, Herrera F (2009) Evolutionary under-sampling for classification with imbalanced data sets: proposals and taxonomy. Evol Comput 17(3):275–306
21. García S, Luengo J, Herrera F (2014) Data preprocessing in data mining. Springer Publishing Company, Berlin
22. García S, Luengo J, Herrera F (2016) Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. Knowl Based Syst 98:1–29
23. García V, Mollineda RA, Sánchez JS (2009) Index of balanced accuracy: a performance measure for skewed class distributions. In: Pattern recognition and image analysis, 4th Iberian Conference, IbPRIA 2009, Póvoa de Varzim, Portugal, June 10–12, 2009, Proceedings, pp 441–448
24. García-Pedrajas N, de Haro-García A (2014) Boosting instance selection algorithms. Knowl Based Syst 67:342–360
25. Hadjadji B, Chibani Y (2014) Optimized selection of training samples for one-class neural network classifier. In: 2014 international joint conference on neural networks, IJCNN 2014, Beijing, July 6–11, 2014, pp 345–349
26. Hempstalk K, Frank E, Witten IH (2008) One-class classification by combining density and class probability estimation. In: Machine learning and knowledge discovery in databases, European conference, ECML/PKDD 2008, Antwerp, September 15–19, 2008, Proceedings, Part I, pp 505–519
27. Hu W, Tan Y (2016) Prototype generation using multiobjective particle swarm optimization for nearest neighbor classification. IEEE Trans Cybern 46(12):2719–2731
28. Japkowicz N, Myers C, Gluck M (1995) A novelty detection approach to classification. In: Proceedings of the 14th international joint conference on artificial intelligence, IJCAI 95, Montréal Québec, August 20–25 1995, vol 2, pp 518–523
29. Juszczak P, Tax DMJ, Pekalska E, Duin RPW (2009) Minimum spanning tree based one-class classifier. Neurocomputing 72(7–9):1859–1869
30. Kim K, Lin H, Choi JY, Choi K (2016) A design framework for hierarchical ensemble of multiple feature extractors and multiple classifiers. Pattern Recogn 52:1–16
31. Krawczyk B (2015) One-class classifier ensemble pruning and weighting with firefly algorithm. Neurocomputing 150:490–500
32. Krawczyk B, Triguero I, García S, Woźniak M, Herrera F (2014) A first attempt on evolutionary prototype reduction for nearest neighbor one-class classification. In: Proceedings of the IEEE congress on evolutionary computation, CEC 2014, Beijing, July 6–11, 2014, pp 747–753. https://doi.org/10.1109/CEC.2014.6900469
33. Krawczyk B, Woźniak M, Herrera F (2015) On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. Pattern Recogn 48(12):3969–3982
34. Lam W, Keung CK, Liu D (2002) Discovering useful concept prototypes for classification based on filtering and abstraction. IEEE Trans Pattern Anal Mach Intell 14(8):1075–1090
35. Leyva E, Muñoz AG, Pérez R (2015) Three new instance selection methods based on local sets: a comparative study with several approaches from a bi-objective perspective. Pattern Recogn 48(4):1523–1537
36. Li Y (2011) Selecting training points for one-class support vector machines. Pattern Recogn Lett 32(11):1517–1522
37. Liu W, Hua G, Smith JR (2014) Unsupervised one-class learning for automatic outlier removal. In: 2014 IEEE conference on computer vision and pattern recognition, CVPR 2014, Columbus, June 23–28, 2014, pp 3826–3833
38. Moya M, Hush D (1996) Network constraints and multi-objective optimization for one-class classification. Neural Netw 9(3):463–474
39. Neri F, Tirronen V (2009) Scale factor local search in differential evolution. Memet Comput 1(2):153–171
40. Parhizkar E, Abadi M (2015) Beeowa: a novel approach based on ABC algorithm and induced OWA operators for constructing one-class classifier ensembles. Neurocomputing 166:367–381
41. Pyle D (1999) Data preparation for data mining. The Morgan Kaufmann series in data management systems. Morgan Kaufmann, Burlington
42. Ramírez-Gallego S, Krawczyk B, García S, Woźniak M, Herrera F (2017) A survey on data preprocessing for data stream mining: current status and future directions. Neurocomputing 239:39–57
43. Rokach L (2016) Decision forest: twenty years of research. Inf Fus 27:111–125
44. Shu W, Shen H (2016) Multi-criteria feature selection on cost-sensitive data with missing values. Pattern Recogn 51:268–280
45. Sonnenburg S, Ratsch G, Schafer C, Scholkopf B (2006) Large scale multiple kernel learning. J Mach Learn Res 7:1531–1565
46. Spurek P, Wójcik M, Tabor J (2015) Cross-entropy clustering approach to one-class classification. In: Artificial intelligence and soft computing—14th international conference, ICAISC 2015, Zakopane, June 14–18, 2015, Proceedings, Part I, pp 481–490
47. Tax DJM, Duin RPW (2001) Uniform object generation for optimizing one-class classifiers. J Mach Learn Res 2:155–173
48. Tax DJM, Duin RPW (2004) Support vector data description. Mach Learn 54(1):45–66
49. Tax DJM, Müller K (2004) A consistency-based model selection for one-class classification. In: 17th international conference on pattern recognition, ICPR 2004, Cambridge, August 23–26, 2004, pp 363–366
50. Tomek I (1976) Two modifications of cnn. IEEE Trans Syst Man Cybern SMC–6(11):769–772
51. Triguero I, Derrac J, García S, Herrera F (2012) A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Trans Syst Man Cybern C Appl Rev 42(1):86–100
52. Triguero I, García S, Herrera F (2010) IPADE: iterative prototype adjustment for nearest neighbor classification. IEEE Trans Neural Netw 21(12):1984–1990
53. Triguero I, García S, Herrera F (2011) Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification. Pattern Recogn 44(4):901–916
54. Triguero I, Peralta D, Bacardit J, García S, Herrera F (2015) MRPR: a mapreduce solution for prototype reduction in big data classification. Neurocomputing 150:331–345
55. Wilk T, Woźniak M (2012) Soft computing methods applied to combination of one-class classifiers. Neurocomputing 75:185–193
56. Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern 2(3):408–421
57. Wilson DR, Martínez TR (2000) Reduction techniques for instance-based learning algorithms. Mach Learn 38(3):257–286
58. Woźniak M, Grana M, Corchado E (2014) A survey of multiple classifier systems as hybrid systems. Inf Fus 16(1):3–17
59. Zhu F, Ye N, Yu W, Xu S, Li G (2014) Boundary detection and sample reduction for one-class support vector machines. Neurocomputing 123:166–173
Bartosz Krawczyk is an assistant professor in the Department of Computer Science, Virginia Commonwealth University, Richmond, VA, USA, where he heads the Machine Learning and Stream Mining Lab. He obtained his M.Sc. and Ph.D. degrees from Wroclaw University of Science and Technology, Wroclaw, Poland, in 2012 and 2015, respectively. His research focuses on machine learning, data streams, ensemble learning, class imbalance, one-class classifiers, and interdisciplinary applications of these methods. He has authored 45+ international journal papers and 90+ contributions to conferences. Dr. Krawczyk received prestigious awards for his scientific achievements like IEEE Richard Merwin Scholarship, IEEE Outstanding Leadership Award, and Best Ph.D. Thesis Award from Polish Artificial Intelligence Society among others. He served as a Guest Editor in four journal special issues and as a chair of twelve special sessions and workshops. He is a member of Program Committee for over 50 international conferences and a reviewer for 30 journals.

Isaac Triguero received the M.Sc. and Ph.D. degree in Computer Science from the University of Granada, Spain, in 2009 and 2014, respectively. He is currently an assistant professor in data science at the University of Nottingham, UK. He has published more than 25 papers in international journals. His research interests include data mining, data reduction, evolutionary algorithms, semi-supervised learning, bioinformatics and big data learning.

Salvador García received the M.Sc. and Ph.D. degrees in Computer Science from the University of Granada, Granada, Spain, in 2004 and 2008, respectively. He is currently an Associate Professor in the Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain. Salvador García has published more than 70 papers in international journals (more than 50 in Q1), with more than 4500 citations, h-index 26, over 45 papers in international conference proceedings (data from Web of Science). He is a member of the editorial board of the “Information Fusion” (Elsevier), “Swarm and Evolutionary Computation” (Elsevier), and “AI Communications” (IOS Press) journals, and he is Co-Editor-in-Chief of the international journal “Progress in Artificial Intelligence” (Springer). He is a co-author of the book entitled “Data Preprocessing in Data Mining” published by Springer. His research interests include data science, data preprocessing, big data, evolutionary learning, deep learning, metaheuristics, and biometrics. Dr. García has been given some awards and honors for his personal work or for his publications and conferences, such as IFSA-EUSFLAT 2015 Best Application Paper Award and IDEAL 2015 Best Paper Award. He belongs to the list of the Highly Cited Researchers in the area of Computer Sciences (2014–2017): http://highlycited.com/ (Clarivate Analytics).
Michał Woźniak is a professor of computer science at the Department of Systems and Computer Networks, Wrocław University of Science and Technology, Poland. He received M.Sc. degree in biomedical engineering from the Wrocław University of Technology in 1992, and Ph.D. and D.Sc. (habilitation) degrees in computer science in 1996 and 2007, respectively, from the same university. In 2015 he was nominated as the professor by President of Poland. His research focuses on machine learning, compound classification methods, classifier ensembles, data stream mining, and imbalanced data processing. Prof. Woźniak has been involved in research projects related to the above-mentioned topics and has been a consultant of several commercial projects for well-known Polish companies and public administration. He has published over 260 papers and three books. His recent Hybrid classifiers: Method of Data, Knowledge, and Data Hybridization was published by Springer in 2014. Prof. Wozniak was awarded with numerous prestigious awards for his scientific achievements as IBM Smarter Planet Faculty Innovation Award (twice) or IEEE Outstanding Leadership Award, and several best paper awards of the prestigious conferences. He served as program committee chair and member for the numerous scientific events and prepared several special issues as the guest editor. He is the member of the editorial board of the high-ranked journals as Information Fusion (Elsevier), Applied Soft Computing (Elsevier), and Engineering Applications of Artificial Intelligence (Elsevier). Prof. Woniak is a senior member of the IEEE.

Francisco Herrera (SM’15) received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain. He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has been the supervisor of 41 Ph.D. students. He has published more than 350 journal papers, receiving more than 57,000 citations (Google Scholar, h-index 118). He is co-author of the books “Genetic Fuzzy Systems” (World Scientific, 2001), “Data Preprocessing in Data Mining” (Springer, 2015), “The 2-tuple Linguistic Model. Computing with Words in Decision Making” (Springer, 2015), “Multilabel Classification. Problem analysis, metrics and techniques” (Springer, 2016), among others. He currently acts as Editor-in-Chief of the international journals “Information Fusion” (Elsevier) and “Progress in Artificial Intelligence” (Springer). He acts as editorial member of a dozen of journals. He received the following honors and awards: ECCAI Fellow 2009, IFSA Fellow 2013, 2010 Spanish National Award on Computer Science ARITMEL to the “Spanish Engineer on Computer Science,” International Cajastur “Mamdani” Prize for Soft Computing (Fourth Edition, 2010), IEEE Transactions on Fuzzy System Outstanding 2008 and 2012 Paper Award (bestowed in 2011 and 2015, respectively), 2011 Lotti A. Zadeh Prize Best paper Award (IFSA Association), 2013 AEPIA Award to a scientific career in Artificial Intelligence, 2014 XV Andalucía Research Prize Maimónides, 2017 Security Forum I + D + I Prize, and 2017 Andalucía Medal (by the regional government of Andalucía). He has been selected as a Highly Cited Researcher http://highlycited.com/ (in the fields of Computer Science and Engineering, respectively, 2014 to present, Clarivate Analytics). His current research interests include, among others, soft computing (including fuzzy modeling, evolutionary algorithms, and deep learning), computing with words, information fusion and decision making, and data science (including data preprocessing, prediction, and big data).