Noname manuscript No.
(will be inserted by the editor)

Efficient SVDD Sampling with Approximation Guarantees for the Decision Boundary

Adrian Englhardt[1] · Holger Trittenbach[1] · Daniel Kottke[2] · Bernhard Sick[2] · Klemens Böhm[1]

Received: date / Accepted: date

Abstract Support Vector Data Description (SVDD) is a popular one-class classifier for anomaly and novelty detection. But despite its effectiveness, SVDD does not scale well with data size. To avoid prohibitive training times, sampling methods select small subsets of the training data on which SVDD trains a decision boundary hopefully equivalent to the one obtained on the full data set. According to the literature, a good sample should therefore contain so-called boundary observations that SVDD would select as support vectors on the full data set. However, non-boundary observations also are essential to not fragment contiguous inlier regions and avoid poor classification accuracy. Other aspects, such as selecting a sufficiently representative sample, are important as well. But existing sampling methods largely overlook them, resulting in poor classification accuracy.

In this article, we study how to select a sample considering these points. Our approach is to frame SVDD sampling as an optimization problem, where constraints guarantee that sampling indeed approximates the original decision boundary. We then propose RAPID, an efficient algorithm to solve this optimization problem. RAPID does not require any tuning of parameters, is easy to implement and scales well to large data sets. We evaluate our approach on real-world and synthetic data. Our evaluation is the most comprehensive one for SVDD sampling so far. Our results show that RAPID outperforms its competitors in classification accuracy, in sample size, and in runtime.

Keywords One-class Classification, Data Reduction, Outlier Detection, Anomaly Detection

1 Introduction

Support Vector Data Description (SVDD) is one of the most popular and actively researched one-class classifiers for anomaly and novelty detection Liu et al. (2010);

Corresponding Author: Adrian Englhardt
E-mail: {adrian.englhardt, holger.trittenbach, klemens.boehm}@kit.edu,
{daniel.kottke, bsick}@uni-kassel.de
[1]Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]University of Kassel, Kassel, Germany

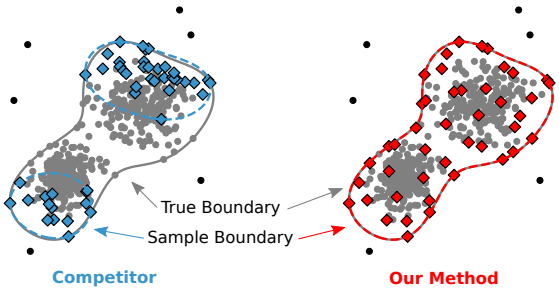

Fig. 1 Sample and decision boundary of a state-of-the-art boundary-point method Alam et al. (2020) and of our method RAPID.

Tax and Duin (2004); Trittenbach et al. (2018). The basic variant of SVDD is an unsupervised classifier that fits a tight hypersphere around the majority of observations, the inliers, to distinguish them from irregular observations, the outliers. Despite its resounding success, a downside is that SVDD and its progeny do not scale well with data size Trittenbach et al. (2019b). Even efficient solvers like decomposition methods Chaudhuri et al. (2018); Chu et al. (2004); Kim et al. (2007); Platt (1998) result in training times prohibitive for many applications. In these cases, sampling for data reduction is essential Alam et al. (2020); Hu et al. (2014); Krawczyk et al. (2019); Li et al. (2018); Li (2011); Li et al. (2019); Qu et al. (2019); Sun et al. (2016); Xiao et al. (2014); Zhu et al. (2014).

One of the defining characteristics of SVDD is that only a few observations, the support vectors, define a decision boundary. Thus, a good sample is one for which SVDD selects support vectors similar to the original ones, i.e., the ones obtained on the full data set. This has spurred the design of sampling methods that try to identify support-vector candidates in the original data, to retain them in the sample Alam et al. (2020); Hu et al. (2014); Li et al. (2018); Li (2011); Li et al. (2019); Qu et al. (2019); Xiao et al. (2014); Zhu et al. (2014). A common approach is to select so-called "boundary points" as support-vector candidates, e.g., observations that are dissimilar to each other Li (2011); Zhu et al. (2014).

But calibrating existing methods such that they indeed identify boundary points is difficult. A reason is that the sample they return depends significantly on the choice of exogenous parameters, and selecting suitable parameter values is not intuitive (see Section 5). A further shortcoming is that including all boundary points in a sample does not guarantee SVDD training to indeed yield the original support vectors. The issue is that selection of support vectors hinges on other aspects, such as the ratio between inliers and outliers in the sample and a sufficient number of non-boundary observations in the sample. Disregarding them may, for instance, fragment contiguous inlier regions and yield wrong outlier classifications after sampling, see Figure 1. The influence of these aspects on SVDD is known, but their effects on sample selection are not well studied. It is an open question how to select a sample where SVDD indeed approximates the original decision boundary. Finally, a point largely orthogonal to these issues is that there also is very limited experimental comparison among competitors. This makes an empirical selection of suitable SVDD sampling methods difficult as well.

Contributions. In this article, we propose a novel way to SVDD sampling. We make three contributions. First, we reduce SVDD sampling to a decision-theoretic problem of separating data using empirical density values. Based on this reduction, we formulate SVDD sampling as a constrained optimization problem. Its objective is to find a minimal sample where the density of all observations of the data set is close-to-uniform. We provide theoretical justification that a sample obtained in this way i) prevent a fragmentation of the inlier regions, and ii) retain the observations necessary to identify the original support vectors.

Second, we propose Reducing sAmples by Pruning of Inlier Densities (RAPID), an efficient algorithm to solve the optimization. RAPID is the first SVDD sampling algorithm with theoretical guarantees on retaining the original decision boundaries. RAPID does not require any parameters in addition to the ones already required by SVDD. This lets RAPID stand out from existing methods, which all hinge on mostly unintuitive, exogenous parameters. RAPID further is easy to implement, and scales well to very large data sets.

Third, we conduct the – by far – most comprehensive comparison of SVDD sampling methods. We compare RAPID against 8 methods on 23 real-world and 85 synthetic data sets. In all experiments, RAPID consistently produces a small sample with high classification quality. Overall, RAPID outperforms all of its competitors in the trade-off between algorithm runtime, sample size, and classification accuracy, often by an order of magnitude.

2 Fundamentals

The data mining community differentiates between *lazy* and *eager* learners Aggarwal (2015a). This differentiation is available for outlier detection as well. There, *lazy* learners perform instance-based learning by defining measures of "outlierness" of an observation Aggarwal (2015b). *Lazy* learners delay the learning until predicting the class of an observation. For an overview and experimental comparison of *lazy* learners we refer to Campos et al. (2016). For *eager* learners, the computational effort takes place before the predictions, since they do construct a classification model. *Eager* learners perform explicit generalization, and the classification of new observations tends to be much faster than for *lazy* learners Aggarwal (2015a). In our article, we focus on the most popular eager learners for outlier detection, Support Vector Data Description (SVDD) Tax and Duin (2004).

The objective of SVDD is to learn a description of a set of observations, the *target*. A good description allows to distinguish the target from other, non-target observations. In our article, we focus on unsupervised outlier detection. So the targets, i.e., the class that SVDD explicitly learns, are inliers, and the non-targets are outliers. However, one does not have any labels available when learning an SVDD classifier, i.e., the learning scenario is unsupervised. First, we introduce preliminaries and then the SVDD optimization problem.

Preliminaries Let $\mathbf{X} = \langle x_1, x_2, \ldots, x_N \rangle$ be a data set of N *observations* from the domain $\mathbb{X} = \mathbb{R}^M$ where M is the number of dimensions. A *sample* is a subset $\mathbf{S} \subseteq \mathbf{X}$ of the data set with sampling ratio $|\mathbf{S}|/N$. Further, we denote $x \in \mathbf{S}$ as *selected*, and $x \notin \mathbf{S}$ as *not-selected* observations. The probability density of $\mathbf{X}$ is $p(x)$. Further, let $\mathbf{Y} = \langle y_1, y_2, \ldots, y_N \rangle$ be a ground truth, i.e., each entry is the

realization of a dichotomous variable $\mathbb{Y} = \{\text{in}, \text{out}\}$. The ground truth densities are the conditional probability densities $p_{\text{inlier}}(x) = P(\mathbf{X} = x \mid \mathbf{Y} = \text{in})$, and $p_{\text{outlier}}(x) = P(\mathbf{X} = x \mid \mathbf{Y} = \text{out})$ respectively. One can estimate the empirical density of $\mathbf{X}$ by kernel density estimation.

$$d_{\mathbf{X}}(x) = \sum_{x' \in \mathbf{X}} k(x, x') \tag{1}$$

where k is a kernel function with $k(x, x) = 1$. A popular choice is the Gaussian kernel $k_\gamma(x, x') = e^{-\gamma \|x - x'\|}$, where $\gamma \geq 0$ is the parameter to control the kernel bandwidth. We use the shorthand $d_x = d_{\mathbf{X}}(x)$ when the reference to $\mathbf{X}$ is unambiguous. Note that $d_{\mathbf{X}}$ requires normalization further to represent a probability density. Densities can be used to characterize observations in different ways.

Definition 1 (Level Set) A level set is a set of observations with equal density $L_\theta \coloneqq \{x \in \mathbf{X} \colon d_x = \theta\}$. A super-level set is a set of observations with $L_\theta^+ \coloneqq \{x \in \mathbf{X} \colon d_x \geq \theta\}$.

One way to use level sets to categorize observations is to define a *level-set classifier* as a function of type $g \colon \mathbb{X} \to \mathbb{Y}$ with

$$g_\theta^{\mathbf{X}}(x) = \begin{cases} \text{in} & \textit{if } x \in L_\theta^+ \\ \text{out} & \textit{else.} \end{cases} \tag{2}$$

Another useful categorization is to separate observations into boundary points and inner points. There are different ways to define a boundary of $\mathbf{X}$ Alam et al. (2020); Hu et al. (2014); Li et al. (2018); Li (2011); Li et al. (2019); Qu et al. (2019); Xiao et al. (2014); Zhu et al. (2014). For this article, we define boundary points as observations with density values close to the minimum empirical density.

Definition 2 (Boundary Point) Let $d_{\min} = \min_{x \in \mathbf{X}} d_x$, and let δ be a small positive value. An observation $x \in \mathbf{X}$ is a boundary point of $\mathbf{X}$ if $x \in \mathbf{B}^{\mathbf{X}}$ with $\mathbf{B}^{\mathbf{X}} = L_{d_{\min}}^+ \setminus L_{(d_{\min}+\delta)}^+$.

SVDD Classifier SVDD Tax and Duin (2004) is a quadratic optimization problem that searches for a minimum enclosing hypersphere with center a and radius R around the data. The linear formulation of the optimization problem is

$$\begin{aligned} \text{SVDD}\colon \underset{a,\ R,\ \boldsymbol{\xi}}{\text{minimize}} \quad & R^2 + C \cdot \sum_{i=1}^{N} \xi_i \\ \text{subject to} \quad & \|x_i - a\|^2 \leq R^2 + \xi_i,\ i = 1, \ldots, N \\ & \xi_i \geq 0,\ i = 1, \ldots, N \end{aligned}$$

with cost parameter C and slack variables $\boldsymbol{\xi}$. Solving SVDD gives a fixed a and R and a decision function

$$f^{\mathbf{X}}(x) = \begin{cases} \text{in} & \textit{if } \|x - a\|^2 \leq R^2 \\ \text{out} & \textit{else.} \end{cases} \tag{3}$$

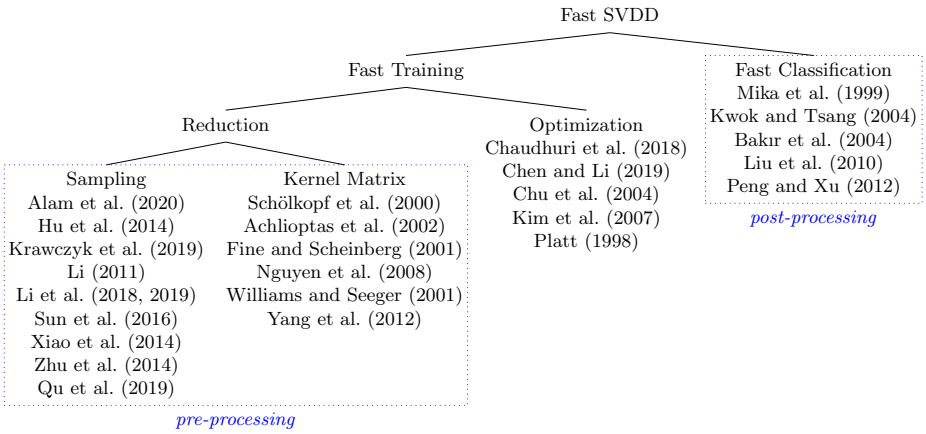

Fig. 2 Categorization of literature on SVDD speedup.

When solving SVDD in the dual space, $f^{\mathbf{x}}$ only relies on inner product calculations between x and some of the training observations, the support vectors. So classification with SVDD is efficient if the number of support vectors is low. Also note that under mild assumptions, SVDD is equivalent to ν-SVM Schölkopf et al. (2001).

SVDD has two hyperparameters, C and a kernel function k. $C \in \mathbb{R}_{[0,1]}$ is a trade-off parameter. It allows some observations in the training data to fall outside the hypersphere if this reduces the radius significantly. Formally, observations outside the hypersphere with positive slack $\xi > 0$ are weighted by a cost C. High values for C make excluding observations expensive; based on the dual of SVDD, one can see that if $C = 1$, SVDD degenerates to a hard-margin classifier Tax and Duin (2004).

To allow decision boundaries of arbitrary shape, one can use the well-known kernel trick to replace inner products in the dual of SVDD by a kernel function k. The most popular kernel with SVDD is the Gaussian kernel. Its bandwidth parameter γ controls the flexibility of the decision boundary. For $\gamma \to 0$, the decision boundary in the data space approximates a hypersphere. Choosing good values for the two hyperparameters γ and C is difficult Liao et al. (2018). There is no established way of setting the parameter values, and one must choose one of the many heuristics to tune SVDD in an unsupervised setting Liao et al. (2018); Scott (2015); Tax and Duin (2004); Trittenbach et al. (2019a).

3 Related Work

SVDD is a quadratic problem (QP). The time complexity of solving SVDD is in $\mathcal{O}(N^3)$ Chu et al. (2004). Thus, training does not scale well to large data sets. However, the time complexity for classification is only linear in the number of support vectors. So for large N, training time is much larger than classification time. Still, long classification times may be an issue, e.g., in time-critical applications. So curbing the runtimes has long become an important topic in the SVDD literature. In Section 3.1, we categorize existing approaches that focus on SVDD

speedup, see Figure 2 for an overview. In Section 3.2, we then turn to *Sampling*, the category our current article belongs to.

3.1 Categorization

We distinguish between *Fast Training* and *Fast Classification*.

Fast Training To speed up training of SVDD, one has two options: reduction of the problem size, and optimization of the solver. For *Reduction*, one can distinguish further: A first type reduces the number of observations by *Sampling*. This is the category of methods mentioned in our introduction Alam et al. (2020); Hu et al. (2014); Krawczyk et al. (2019); Li et al. (2018); Li (2011); Li et al. (2019); Qu et al. (2019); Sun et al. (2016); Xiao et al. (2014); Zhu et al. (2014). A second type reduces the size of the *Kernel matrix*, e.g., by approximation Achlioptas et al. (2002); Fine and Scheinberg (2001); Nguyen et al. (2008); Schölkopf et al. (2000). Examples are the Nyström-method Williams and Seeger (2001) and choosing random Fourier features Yang et al. (2012).

Optimization on the other hand decomposes QP into smaller chunks that can be solved efficiently. Literature features methods that decompose with clustering Kim et al. (2007) and with multiple random subsets Chaudhuri et al. (2018). The most widely used decomposition methods are sequential minimal optimization (SMO) Platt (1998) and its variants. These methods iteratively divide SVDD into small QP sub-problems and solve them analytically. Finally, there are core-set method that expands the decision boundary by iteratively updating an SVDD solution Chen and Li (2019); Chu et al. (2004). Core-set approaches are $(1 + \varepsilon)$ approximations, i.e., they may not find the exact decision boundary, given training data.

Reduction and Optimization are orthogonal to each other. Thus, one can use problem-size reduction in a *pre-processing* step before solving SVDD efficiently.

Fast Classification When SVDD uses a non-linear kernel, one cannot compute the pre-image of the center a. Instead, one must compute the distance of an observation to a by a linear combination of the support vectors in the kernel space. However, literature proposes several approaches to approximate the pre-image of a Bakır et al. (2004); Kwok and Tsang (2004); Liu et al. (2010); Mika et al. (1999); Peng and Xu (2012). With this, classification no longer depends on the support vectors, and is in $\mathcal{O}(1)$. Fast Classification is orthogonal to Fast Training, i.e., it can come as a *post-processing* step, after training.

3.2 Sampling Methods

Sampling methods take the original data $\mathbf{X}$ set as an input and produce a sample $\mathbf{S}$. All existing sampling methods assume the *target-only scenario*, i.e., all observations in $\mathbf{X}$ are from the target class. This is equivalent to a supervised setting where one has knowledge of the ground truth, and $\mathbf{Y} = \langle \text{in}, \text{in}, \ldots, \text{in} \rangle$. Thus, most of the competitors therefore require modifications to apply to the outlier scenario, see Section 4.1 for details. In the following, we discuss existing sampling methods

Table 1 Sampling methods proposed for SVDD.

Method	Publication	Year	Exogenous Parameters*
BPS	Li	2011	$k=\lfloor 10 \ln N \rfloor$, ε=0.05
DAEDS	Hu et al.	2014	k=30, ε=0.1, δ=0.3
DBSRSVDD	Li et al.	2019	*minPts*=7, ε=0.5
FBPE	Alam et al.	2020	n=360
HSR	Sun et al.	2016	k=20, $\varepsilon=0.01 \cdot M$
HSC†	Qu et al.	2019	k=20
IESRSVDD	Li et al.	2018	ε=0.5
KFNCBD	Xiao et al.	2014	k=100, ε=0.2
NDPSR	Zhu et al.	2014	k=20, ε=10
OCSFLSDE†	Krawczyk et al.	2019	8 different parameters

* The listed values for the exogenous parameters are the ones used in our experiments.
† Not included in our experiments, see Section 5.1 for details.

for the *target-only scenario*. We categorize them into different types: *Edge-point* detectors, *Pruning* methods and *Others*. Table 1 provides an overview.

Edge-point Most sampling approaches focus on selecting observations that demarcate p_{inlier} from p_{outlier}, and therefore are expected to be support vectors. Such observations are called "edge points" or "boundary points". Literature proposes different ways to identify edge points. One idea is to use the angle between an observation and its k nearest neighbors Li (2011); Zhu et al. (2014) as an indication. An observation is selected as edge point if most of its neighbors lie within a small, convex cone with the observation as the apex. One has to specify a threshold for the share of neighbors and the width of the cone Li (2011) as exogenous parameters. Others suggest to identify edge points through a farthest neighbor search. For instance, one suggestion is to first sort the observations by decreasing distance to its k-farthest neighbors (KFN) Xiao et al. (2014), and then select the top ε percent as edge points. The rationale presented in the paper is that inner points are expected to have a lower KFN distance than edge points. A more recent variant uses angle-based search Alam et al. (2020). The idea of the paper is to initialize the method by the mean over all observations as the apex and divide the space into a pre-specified number of cones. For each cone, one only keeps the farthest observation as edge points.

Next, there are methods that select edge points by density-based outlier rankings, e.g., DBSCAN Li et al. (2019) and LOF Hu et al. (2014). Here, the assumption is that edge points occur in sparse regions of the data space. A similar idea is to rank observations with a high distance to all other observations Li et al. (2018). Others have suggested to rank observation highly if they have low density and a large distance to high-density observations Qu et al. (2019). Naturally, ranking methods require to set a cutoff value to distinguish edge points from other observations.

Pruning The idea of pruning is to iteratively remove observations from high-density regions as long as the sample remains "density-connected". One way to achieve this is by pruning all neighbors of an observation closer than a minimum distance, starting from the observation closest to the cluster mean Sun et al. (2016). Yet this approach requires to set the minimum distance threshold, and a good choice is data dependent.

Others There is one method that differs significantly from the other ones Krawczyk et al. (2019). The basic idea is to generate artificial outliers to transform the problem into a binary classification problem. Based on the augmented data, one can apply conventional sampling methods such as binary instance reduction. The sampling method then relies on an evolutionary algorithm where the fitness function is the prediction quality on the augmented data. Finally, the method only retains the remaining inliers and discards all artificial observations. However, this requires to solve many SVDD instances in each iteration.

To summarize, there are many methods to select a sample for SVDD. However, they are based upon some intuition regarding the SVDD and do not come with any formal guarantee. Edge point detectors in particular return a poor sample in some cases, since they do not guarantee coherence of a selected sample, see Figure 1. Further, all existing approaches require to set some exogenous parameter. But the influence of the parameter values on the sample is difficult to grasp. Finally, existing sampling methods are designed for the *target-only scenario*. It is unclear whether they can be modified to work well with the *outlier scenario*.

4 Density-based Sampling for SVDD

In this section, we present an efficient and effective sampling method for scaling SVDD to very large data sets. In a nutshell, we exploit that an SVDD decision boundary is in fact a level-set estimate Vert and Vert (2006), and that inliers are a super-level set. The idea behind our sampling method is to remove observations from a data set such that the inlier super-level set does not change. To this end, we show that for the Gaussian kernel *the super-level set of inliers does not change as long as not-selected observations have higher density than the minimum density of selected observations*. If this *density rule* is violated, sampling may produce "gaps", i.e., regions of inliers that become regions of outliers. Such gaps curb the SVDD quality. Thus, we strive for a sample of minimal size that satisfies the density rule.

Figure 3 illustrates our approach. In a first step, we separate the unlabeled data into outlier and inlier regions based on their empirical density, see Section 4.1. We then frame sample selection as a optimization problem where the constraints enforce the density rule in Section 4.2. In Section 4.3 we propose RAPID, an efficient and easy-to-implement algorithm to solve the optimization problem. RAPID returns a small sample which has a close-to-uniform density, i.e., a small sample that still obeys the density rule, and also contains the boundary points of the original data.

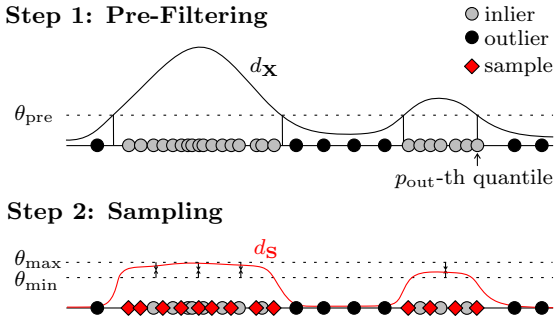

Fig. 3 The idea of density-based sampling for SVDD.

Algorithm 1: Pre-filtering

Input : Data set $\mathbf{X} \in \mathbb{R}^{N \times M}$ Kernel function $k(x_i, x_j)$, Outlier percentage $p_{\text{out}} \in [0, 1]$

Output: Indices for inliers $\mathcal{I}$ and outliers $\mathcal{O}$, density d

1 $d = \langle \sum_{j=1}^{N} k(x_1, x_j), \ldots, \sum_{j=1}^{N} k(x_N, x_j) \rangle$ ▷ $\mathcal{O}(N^2)$

2 $\theta_{\text{pre}} = \text{sort-ascending}(d)_{\lfloor p_{\text{out}} \cdot N \rfloor}$ ▷ $\mathcal{O}(N \log N)$

3 $\mathcal{I} = \{i \mid i \in \{1, \ldots, N\},\ d_i \geq \theta_{\text{pre}}\}$ ▷ $\mathcal{O}(N)$

4 $\mathcal{O} = \{i \mid i \in \{1, \ldots, N\}\} \setminus \mathcal{I}$ ▷ $\mathcal{O}(1)$

5 $d = d - \langle \sum_{j \in \mathcal{O}} k(x_1, x_j), \ldots, \sum_{j \in \mathcal{O}} k(x_N, x_j) \rangle$ ▷ $\mathcal{O}(N^2)$

6 **return** $\mathcal{I}$, $\mathcal{O}$, d

4.1 Density-based Pre-Filtering

Any sampling method faces an inherent trade-off: reducing the size of the data as much as possible while maintaining a good classification accuracy on the sample. One can frame this as an optimization problem

$$\begin{aligned} \underset{\mathbf{S} \subseteq \mathbf{X}}{\text{minimize}} \quad & |\mathbf{S}| \\ \text{subject to} \quad & \mathit{diff}(f^{\mathbf{S}}, f^{\mathbf{X}}) \leq \varepsilon, \end{aligned} \tag{4}$$

where *diff* is a similarity between two decision functions and ε a tolerable deterioration in accuracy. Solving Optimization Problem 4 requires knowledge of $f^{\mathbf{X}}$. But obtaining this knowledge is infeasible. The reason is that $|\mathbf{X}|$ is too large to solve — SVDD would not need any sampling in the first place otherwise. Thus, one cannot infer which observations $f^{\mathbf{X}}$ classify as inlier or outlier. However, we know that the SVDD hyperparameter C defines a lower bound on the share of observations predicted as outliers in the training data Tax and Duin (2004). A special case is if $C = 1$, since $f^{\mathbf{X}}(x;\ C = 1) = \text{in}, \forall x \in \mathbf{X}$. Recall that this is the upper bound of the cost parameter C where SVDD degenerates to a hard-margin classifier, cf. Section 2. In this case, *diff* is zero if SVDD trained on $\mathbf{S}$, i.e., $f^{\mathbf{S}}$, also includes all observations within the hypersphere. Further, we can make use of the following characteristic of SVDD.

Characteristic 1 (SVDD Level-Set Estimator) *SVDD is a consistent level set estimator for the Gaussian kernel Vert and Vert (2006).*

In consequence, inliers form a super-level set with respect to the decision boundary. Formally, this means that there exists a level set L_θ and a corresponding level-set classifier $g_\theta^{\mathbf{X}}$ such that $g_\theta^{\mathbf{X}} \equiv f^{\mathbf{X}}$. We can exploit this characteristic as follows. First, we *pre-filter* the data based on their empirical density, such that a share of p_{out} observations are outliers. Formally, p_{out} is equivalent to choosing a threshold θ_{pre} on the empirical density, where θ_{pre} is the p_{out}-th quantile of the empirical density distribution. Using this threshold in a level-set classifier separates observations into inliers $\mathbf{I}$ and outliers $\mathbf{O}$.

$$\mathbf{I} = \{x \in \mathbf{X} \colon g_{\theta_{\text{pre}}}^{\mathbf{X}} = \text{in}\} \qquad \mathbf{O} = \{x \in \mathbf{X} \colon g_{\theta_{\text{pre}}}^{\mathbf{X}} = \text{out}\}.$$

Second, we replace $f^{\mathbf{X}}$ with $f^{\mathbf{I}}$ and set $C = 1$. With this, we know that $f^{\mathbf{I}}(x) = \text{in}, \forall x \in \mathbf{I}$, without training $f^{\mathbf{I}}$. Put differently, pre-filtering the data with an explicit threshold allows to get rid of an implicit outlier threshold C. This in turn allows to estimate the level set estimated by SVDD without actually training the classifier. Algorithm 1 is the pseudo code for the pre-filtering.

Pre-filtering does not add any new exogenous parameter, but replaces the SVDD trade-off parameter C with p_{out}. Further, p_{out} is a parameter of SVDD, not of our sampling method. We also deem p_{out} slightly more intuitive than C, since it makes the lower bound defined by C tight, i.e., pre-filtering assumes an exact outlier ratio of $p_{\text{out}} = {}^{|\mathbf{O}|}/_{|\mathbf{X}|}$. This in turn makes the behavior of SVDD more predictable. We note further that in an unsupervised case the C parameter of the SVDD is commonly coupled with the "target error estimate" introduced in Tax and Duin (2004): The "target error estimate" is exactly the expected outlier percentage p_{out}, and one sets $C \leq 1/(N * p_{\text{out}})$. So our pre-filtering step uses exactly the p_{out} estimate that one would use for parametrization of SVDD in an unsupervised scenario. We close the discussion of pre-filtering with two remarks.

Remark 1 Technically, one may directly use the level-set classifier $g_{\theta_{\text{pre}}}^{\mathbf{X}}$ instead of SVDD. However, classification times are very high, since calculating the kernel density of an unseen observation is in $\mathcal{O}(N)$. So one would give up fast classification, one of the main benefits of SVDD. Next, one may be tempted to interpret this pre-filtering step as a way to transform an unsupervised problem into a supervised one to train a binary classifier (e.g., SVM) on $\mathbf{O}$ and $\mathbf{I}$. However, binary classification assumes the training data to be representative of the underlying distributions. This assumption is not met with outlier detection, since outliers may not come from a well-defined distribution. Thus, binary classification is not applicable.

Remark 2 Pre-filtering is a necessary step with all sampling methods discussed in related work. In Section 3, we have explained that existing sampling methods assume to only have inliers in the data set, i.e., $\mathbf{I} = \mathbf{X}$ and $\mathbf{O} = \emptyset$. However, if $\mathbf{X}$ contains outliers, this affects the sampling quality negatively and leads to poor SVDD results, see Section 5.3.

4.2 Optimal Sample Selection

After *pre-filtering*, we can reduce Optimization Problem 4 to a feasible optimization problem. We begin by replacing $f^{\mathbf{X}}$ with $f^{\mathbf{I}}$. With Characteristic 1, we further know that both classifiers have equivalent level-set classifiers. We set $g_{\theta_{\text{pre}}}^{\mathbf{I}}$ as the

equivalent level-set classifier for $f^{\mathbf{I}}$. For $f^{\mathbf{S}}$, there also exists a level-set classifier $g^{\mathbf{S}}_{\theta'}$, but the level set θ' depends on the choice of $\mathbf{S}$. Thus, we must additionally ensure that θ' indeed is the level set estimated by training SVDD on $\mathbf{S}$. The modified optimization problem is

$$\begin{aligned} \underset{\mathbf{S}\subseteq\mathbf{X}}{\text{minimize}} \quad & |\mathbf{S}| && (5)\\ \text{subject to} \quad & \mathit{diff}(g^{\mathbf{S}}_{\theta'}, g^{\mathbf{I}}_{\theta}) \leq \varepsilon && (5a)\\ & g^{\mathbf{S}}_{\theta'} \equiv f^{\mathbf{S}}, && (5b) \end{aligned}$$

where $\equiv$ denotes the equivalence in classifying $\mathbf{S}$. Constraint 5b is necessary, since one may select a sample that yields a level-set classifier similar to the one obtained from $\mathbf{I}$, but on which SVDD returns another decision boundary. This can, for instance, occur if $\mathbf{S}$ does not contain the boundary points of $\mathbf{I}$. Optimization Problem 5 still is very abstract. We will now elaborate on both of its constraints and show how to reduce them so that the problem becomes practically solvable.

Constraint 5a We now discuss how to obtain a sample that minimizes $\mathit{diff}(g^{\mathbf{S}}_{\theta'}, g^{\mathbf{I}}_{\theta})$. To this end, we use the following theorem.

Theorem 1 *$g^{\mathbf{S}}_{\theta'} \equiv g^{\mathbf{I}}_{\theta}$ if $d_{\mathbf{S}}$ is uniform on $\mathbf{I}$.*

Proof Think of a sample $\mathbf{S} \subseteq \mathbf{I}$ with uniform empirical density $d_{\mathbf{S}}$. Then $\mathbf{S}$ has exactly one level set $\theta' = \theta_{\min} = \min_{x\in\mathbf{S}} d_{\mathbf{S}}(x)$. Further, it also holds that $d_{\mathbf{S}}(x) = \theta_{\min}$, $\forall x \in \mathbf{I}$. It follows that $\min_{x\in\mathbf{I}\setminus\mathbf{S}} d_{\mathbf{S}}(x) = \min_{x\in\mathbf{S}} d_{\mathbf{S}}(x)$, and consequently $g^{\mathbf{S}}_{\theta_{\min}}(x) = g^{\mathbf{I}}_{\theta}(x), \forall x \in \mathbf{I}$. □

Theorem 1 implies that one can satisfy Constraint 5a with $\varepsilon = 0$ if one reduces the sample to one with a uniform empirical distribution $d_{\mathbf{S}}$. However, any empirical density estimate on a finite sample can only *approximate* a uniform distribution. So one should strive for solutions of Optimization Problem 5 where epsilon is small. Put differently, one can interpret the difference between a perfect uniform distribution and the empirical density to assess the quality of a sample. We propose to quantify the fit with a uniform distribution as the difference between the maximum density $\theta_{\max} = \max_{x\in\mathbf{S}} d_{\mathbf{S}}(x)$ and minimum density $\theta_{\min} = \min_{x\in\mathbf{S}} d_{\mathbf{S}}(x)$:

$$\Delta^{\mathbf{S}}_{\text{fit}} = \theta_{\max} - \theta_{\min} \tag{6}$$

There certainly are other ways to evaluate the goodness of fit between distributions. However, $\Delta^{\mathbf{S}}_{\text{fit}}$ has some desirable properties of the sample, which we discuss in Theorem 2.

One further consequence of only approximating a uniform density is that there may be some not-selected observations $x \in \mathbf{I}\setminus\mathbf{S}$ with a density value $d_{\mathbf{S}}(x)$ less than $\theta_{\min}$. Since the level set estimated by $f^{\mathbf{S}}$ is $L_{\theta_{\min}}$, these not-selected observations would be wrongly classified as outliers. Thus, we must also ensure that $\mathbf{S}$ is selected so that $d_{\mathbf{S}}(x) \geq \theta_{\min}, \forall x \in \mathbf{I} \setminus \mathbf{S}$. We can now re-formulate Constraint 5a as a

sample optimization problem SOP.

$$\text{SOP}: \underset{\mathbf{v},\mathbf{w},\theta_{\min},\theta_{\max}}{\text{minimize}} \quad \theta_{\max} - \theta_{\min} \tag{7}$$

$$\text{s.t.} \underbrace{\sum_{j\in\mathcal{I}} v_j \cdot k(x_i, x_j)}_{d_{\mathbf{S}}(x_i)} \geq \theta_{\min},\ \forall i \in \mathcal{I} \tag{7a}$$

$$\sum_{j\in\mathcal{I}} v_j \cdot k(x_i, x_j) \leq \theta_{\max},\ \forall i \in \mathcal{I} \tag{7b}$$

$$\sum_{j\in\mathcal{I}} w_i \cdot v_j \cdot k(x_i, x_j) \leq \theta_{\min},\ \forall i \in \mathcal{I} \tag{7c}$$

$$\sum_{j\in\mathcal{I}} v_j > 0; \sum_{j\in\mathcal{I}} w_j = 1;\ v_j \geq w_j, \forall j \in \mathcal{I} \cup \mathcal{O} \tag{7d}$$

$$v_j = 0, \forall j \in \mathcal{O}; v_j, w_j \in \{0,1\}, \forall j \in \mathcal{I} \cup \mathcal{O} \tag{7e}$$

where $\mathcal{I} = \{i \mid i \in \{1,\dots,N\}, x_i \in \mathbf{I}\}$, $\mathcal{O} = \{1,\dots,N\} \setminus \mathcal{I}$. The decision variable $v_j = 1$ indicates if an observation x_j is in $\mathbf{S}$, i.e., $\mathbf{S} = \{x_i \in \mathbf{X} \mid v_i = 1\}$. Constraint 7b is a technical necessity to obtain the maximum density of $d_{\mathbf{S}}$. The first constraint in 7d rules out the trivial solution $v = \vec{0}$. The first constraint in 7e results from the *pre-filtering*, cf. Section 4.1. If the solution set of SOP is not singular, we select the solution where $|\mathbf{S}|$ is minimal to minimize training time.

Constraints 7a, 7c, and 7d together guarantee that the density of not-selected observations is at least $\theta_{\min}$, as follows. Only for one observation j we have $w_j = 1$ and for all other observations $i \neq j$, $w_i = 0$. Then for Constraint 7c and 7d to hold, j must be the observation with the minimum density and $d_{\mathbf{S}}(x_j) = \theta_{\min}$. Additionally, with $v_j \geq w_j$ it follows that $v_j = 1$, thus observation j is in the sample $\mathbf{S}$. So, for any feasible solution of SOP all not-selected observations have a density of at least the minimum density of the selected observations. From 7a, it follows that $d_{\mathbf{S}}(x) \geq \theta_{\min}, \forall x \in \mathbf{I}$. So any solution of SOP satisfies Inequality 5a with a small ε.

Constraint 5b We now show that a solution of SOP also satisfies Constraint 5b. To this end, we make use of the following characteristic.

Characteristic 2 (Boundary Points) *The set of boundary points are a superset of the support vectors of SVDD Tax and Duin (2004).*

So for Constraint 5b to hold, an optimum of SOP must contain boundary points of $\mathbf{I}$. We show that a solution with boundary points is preferred over one without boundary points by the following theorem.

Theorem 2 *The set of boundary points does not change when solving SOP iteratively.*

Proof Suppose that there exists a sample $\mathbf{S}$ which is not a local optimum of SOP. Then there is a boundary point $x_{\min} = \arg\min_{x\in\mathbf{S}} d_{\mathbf{S}}(x)$, an observations $x_{\max} = \arg\max_{x\in\mathbf{S}} d_{\mathbf{S}}(x)$ and $x_p \in \mathbf{S}$. Let $\mathbf{S}_p = \mathbf{S}\backslash\{x_p\}$ and $\mathbf{S}_{\max} = \mathbf{S}\backslash\{x_{\max}\}$. If removing

x_p from $\mathbf{S}$ is an optimal choice, there must be no other observation that reduces the objective more than x_p. Thus, the following specific case must hold:

$$\begin{aligned} & \Delta_{\text{fit}}^{\mathbf{S}_p} \leq \Delta_{\text{fit}}^{\mathbf{S}_{\max}} \\ \Leftrightarrow\ & \theta_{\max} - k(x_p, x_{\max}) - (\theta_{\min} - k(x_p, x_{\min})) \\ & \leq \theta_{\max} - k(x_{\max}, x_{\max}) - (\theta_{\min} - k(x_{\max}, x_{\min})) \\ \Leftrightarrow\ & k(x_p, x_{\max}) - k(x_p, x_{\min}) \geq 1 - k(x_{\max}, x_{\min}). \end{aligned} \tag{8}$$

For one, we conclude that $x_p = x_{\min}$ is not feasible, because in this case the left hand side of Inequality 8 is strictly negative, and right hand side positive. Since boundary points have, per Definition 2, a density close to $\theta_{\min}$, they cannot be a candidate for removal.

Next, under two assumptions that (A1) the locations of the maximum and of the minimum density are distant from each other, and that (A2) the kernel bandwidth is sufficiently small, we have $k(x_{\max}, x_{\min}) \to 0$, and $k(x_p, x_{\max}) - k(x_p, x_{\min}) \geq 1 \Leftrightarrow x_p = x_{\max}$. So in this case, removing $x_{\max}$ is optimal. From this, it also follows that the minimum density does not change significantly when removing $x_{\max}$. With Definition 2, it follows that also the set of boundary points does not change after removing $x_{\max}$. □

Remark 3 Our proof hinges on two assumptions: *(A1) A sufficiently large distance between x_{max} and x_{min}.* This assumption is intuitive, since removing an observation with a density close to $\max_{x \in \mathbf{S}} d_{\mathbf{S}}(x)$ improves Δ_{fit} more than removing one close to $\min_{x \in \mathbf{S}} d_{\mathbf{S}}(x)$. Generally, the distance between $x_{\max}$ and $x_{\min}$ depends on the data distribution. However, we find that this is not a limitation in practice, see Section 5. *(A2) A sufficiently small kernel bandwidth.* This assumption is reasonable, because when selecting the kernel bandwidth, one strives to avoid underfitting, i.e., avoid kernels bandwidth that are too wide. This holds empirically as well, see Section 5.

Remark 4 Overfitting the kernel parameter of SVDD affects all sampling methods. When the kernel bandwidth is very small, removing any observations from a sample yields a decision boundary that is different from the one obtained with training on the full data set. For SOP an overfitted kernel bandwidth results in density values of approximately 1 for all observations with the Gaussian kernel, i.e., the density is already uniform. The objective function of SOP then is already minimal, with a value of 0. Thus, SOP does not remove any observation from the sample and retains the original decision boundary. In practice, one can rely on one of the many heuristics to choose a suitable kernel parameter to avoid overfitting, see for example our choice in Section 5.

SOP is appealing in theory. However, it is a mixed-integer problem with non-convex constraints, and it is hard to solve. Thus, solver runtimes quickly become prohibitive, even for relatively small problem instances. This contradicts the motivation for sampling. We therefore propose RAPID, a fast algorithm to search for a local optimum of SOP.

4.3 A RAPID Approximation

The idea of our approximation is to initialize $\mathbf{S} = \mathbf{I}$, which is a feasible solution to SOP, and remove observations from $\mathbf{S}$ iteratively as long as $\mathbf{S}$ remains feasible,

Algorithm 2: RAPID

Input : Data set $\mathbf{X} \in \mathbb{R}^{N \times M}$ Kernel function $k(x_i, x_j)$,
Outlier percentage $p_{\text{out}} \in [0, 1]$
Output: Sample indices $\mathcal{S}$

▷ `Pre-filtering, see Algorithm 1`
1 $\mathcal{I}, \mathcal{O}, d = \text{pre-filtering}(\mathbf{X}, k, p_{\text{out}})$ ▷ $\mathcal{O}(N^2)$
▷ `Sampling`
2 **for** $iter \leftarrow 1 \dots |\mathcal{I}| - 1$ **do** ▷ $\mathcal{O}(N^2)$
3 $r = \arg\max_{i \in \mathcal{S}} d_i$
4 $d = d - \langle k(x_1, x_r), \dots, k(x_N, x_r) \rangle$
5 $\theta_{\text{min}} = \min_{i \in \mathcal{S}} (d_i)$
6 **if** $\exists\, i \in \mathcal{I} : d_i < \theta_{min}$ **then**
7 **return** $\mathcal{S}$
8 **end**
9 $\mathcal{S} = \mathcal{S} \setminus \{r\}$
10 **end**
11 **return** $\mathcal{S}$

see Algorithm 2. RAPID is a fast greedy algorithm, i.e., it may not produce the smallest sample with uniformity, cf. objective function of SOP. However, the proofs for SOP that sampling retains the decision boundary also hold for RAPID.

As input parameters RAPID takes the data set $\mathbf{X}$, the expected outlier percentage p_{out} and a kernel function k. Line 1 is the *pre-filtering*. RAPID then iteratively selects the most dense observation x_{max} in the current sample $\mathbf{S}$ for removal (Line 3) and updates the densities (Line 4). If $\mathbf{S} \setminus \{x_{\text{max}}\}$ is infeasible, RAPID terminates (Line 5–7). Line 6 checks whether there is an observation $x_i \in \mathbf{I}$ that violates Constraint 7a. As required by SOP, RAPID does not remove boundary points. This is because x_{max} must not be a boundary point, as long as $\mathbf{S}$ is not uniform, i.e., $\Delta^{\mathbf{S}}_{\text{fit}} > 0$. Thus, a solution of RAPID satisfies both Constraint 5a and Constraint 5b. The return in Line 11 is the special case where a single observations remains in the sample. In this case uniformity is achieved with one observations, i.e., all observations are equal.

The overall time complexity of RAPID is in $\mathcal{O}(N^2)$, see Algorithm 1 and Algorithm 2 for the step-wise time complexities. Further, RAPID is simple to implement with only a few lines of code. It is efficient, since each iteration (Line 3–7) requires only one pass over the data set to update the densities, compute the new x_{max}, θ_{min} and minimum inlier density for the termination criterion. One may further pre-compute the Gram matrix $\mathbf{K}$ for $\mathbf{X}$ to avoid redundant kernel function evaluations.

Remark 5 RAPID does not require any hyperparameters in addition to the ones already required by SVDD. The two parameters are: a parametrized kernel function k and the outlier percentage p_{out}. The outlier percentage p_{out} is commonly estimated to calculate the C parameter of SVDD Tax and Duin (2004). Since we guarantee that RAPID retains the decision boundary one would learn on the full data set, the kernel parametrization affects the sampling. However, due to the density rule, the parametrization only affects how many observations RAPID removes from the sample Yet RAPID always retains the decision boundary. While

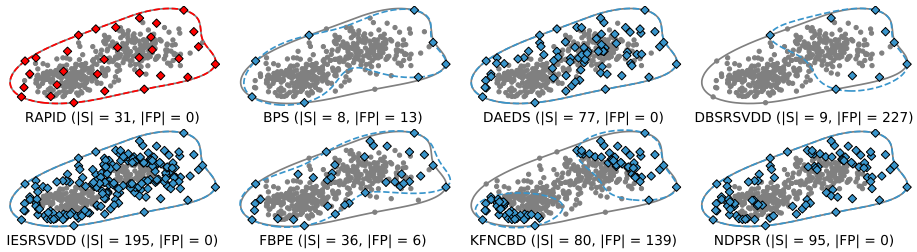

Fig. 4 Sampling strategies applied to a synthetic Gaussian mixture with two components and $N = 400$. The grey points are the original data set and the red/blue diamonds the selected observations. The original decision boundary is the grey line and the red/blue one is the boundary trained on the sample. $|\mathbf{S}|$ is the sample size and |FP| the number of misclassified inliers. We omit HSR since it returns $\mathbf{S} = \mathbf{X}$ with recommended parameter values.

the exact sampling always depends on the data set, the general intuition is that, with a higher kernel width, RAPID can remove more observations than with a more narrow one. In the extreme case of a very small kernel width, RAPID cannot remove any observations without violating the density rule, c.f. our discussion in Remark 4. Ultimately, given a novel data set, one must set the same parameters for SVDD with or without sampling with RAPID. One commonly relies on one of the many heuristics to parametrize SVDD, see our discussion at the end of Section 2.

5 Experiments

We now turn to an empirical evaluation of RAPID. Our evaluation consists of two parts. In the first part, we evaluate how well RAPID copes with different characteristics of the data, i.e., with the dimensionality, the number of observations, and the complexity of the data distribution, see Section 5.2. The second part is an evaluation on a large real-world benchmark for outlier detection. We have implemented RAPID as well as the competitors in an open-source framework written in Julia Bezanson et al. (2017). Our implementation, data sets, raw results, and evaluation notebooks are publicly available. [1]

5.1 Setup

We first introduce our experimental setup, including evaluation metrics, as well as the parametrization of SVDD and its competitors. Recall that RAPID does not have any exogenous parameter. One must only specify p_{out} instead of the SVDD hyperparameter C, cf. Section 4.1.

Metrics Sampling methods trade classification quality for sample size, and one must evaluate this trade-off explicitly. We report the sample size $|\mathbf{S}|$ and sample ratio $|\mathbf{S}|/|\mathbf{X}|$ for each result. To evaluate the classification quality, we use the Matthews Correlation Coefficient (MCC) on $\mathbf{X}$. MCC is well-suited for imbalanced data and returns values in $[-1, 1]$; higher values are better. SVDD returns a binary

[1] `https://www.ipd.kit.edu/ocs`

classification which is different from many other outlier-detection methods which produce score-based outputs Aggarwal (2015b). For such score-based outputs, one usually calculates ROC-AUC. ROC-AUC and MCC are statistically consistent with each other Halimu et al. (2019), we report the values for other evaluation metrics (ROC-AUC, F1-score and Cohen's kappa coefficient) in the appendix of this article. For a full analysis see our supplementary material. We report the averages over five runs on synthetic data and perform 5-fold cross-validation on real-world data. For non-deterministic methods, we report average values over five repetitions. Our experiments ran on an AMD Ryzen Threadripper 2990WX with 64 virtual cores and 128 GB RAM.

SVDD SVDD requires to set two hyperparameters: the Gaussian kernel parameter γ and the trade-off parameter C. We tune γ with *Scott's Rule* Scott (2015) for real-world data. For high-dimensional synthetic data, however, we found that the *Modified Mean Criterion* Liao et al. (2018) is a better choice. The *Modified Mean Criterion* in these cases yields a higher kernel bandwidth. This allows sampling to remove more observations, c.f. Remark 5. Because of *pre-filtering* we set $C = 1$, cf. Section 4.1.

Competitors We compare our method against 8 competitors, see Table 1. The approaches from Qu et al. (2019) and Krawczyk et al. (2019) require to solve several hundreds of SVDDs, resulting in prohibitive runtimes. We do not include them in our evaluation. We initialize the exogenous parameters according to the guidelines in the original publications. In some cases, the recommendations do not lead to a useful sample, e.g., $\mathbf{S} = \emptyset$. To ensure a fair comparison, we mitigate these issues by fine-tuning the parameter values through preliminary experiments.

Next, we compare two variants of each competitor: sampling on $\mathbf{X}$ as in their original version, and sampling on $\mathbf{I}$, i.e., after applying our *pre-filtering*. The *pre-filtering* requires to specify the expected outlier percentage p_{out}. In practice, one can rely on domain knowledge or estimate it Achtert et al. (2010). To avoid any bias when over- or under-estimating the outlier percentage, we set it to the true percentage. Nevertheless, we have run additional experiments where we deliberately deviate from the true percentage. We found that deviating affects the performance of all sampling methods similarly. So, our conclusions do not depend on this variation, and we report the respective results only in the supplementary materials.[1]

We also evaluate against random baselines. Each baseline $Rand_r$ returns a random subset with a specified sample ratio r. We report results for a range of sample ratios $r \in [0.01, 1.0]$ to put the quality of competitors into perspective. When choosing the C parameter of SVDD for the random baseline, one must observe that outliers may be part of the selected sample. However, in experiments of ours, we have observed that $C = 1$ generally yields the most competitive baseline even if some outliers are part of the training data. Training a $r = 1$ baseline on the full data set is prohibitive for large data sets. So we only report the values for the smaller data sets.

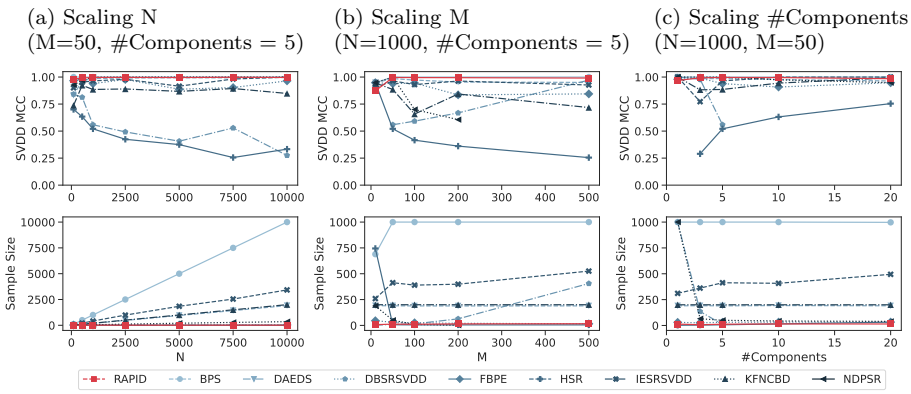

Fig. 5 Evaluation on synthetic data with varying data size (N), dimensionality (M), and complexity (#Components).

5.2 Evaluation of Sample Characteristics

The first part of our experiments validates different properties of RAPID and of its competitors. Our intention is to give an intuition of how a sample is selected, and to explore under which conditions the sampling methods work well. The basis for our experiments are synthetic data sets with controlled characteristics. Specifically, we generate data from Gaussian mixtures with varying number of mixture components, data dimensions, and number of observations, see Algorithm 3 for the data generation algorithm. We run these experiments to answer the following two questions.

Q1 How are observations in a sample distributed?

To get an intuition about the sample distribution, we run RAPID and the competitors on a bi-modal Gaussian mixture, see Figure 4. The tendencies of the methods to select boundary points and inner points are clearly visible. For instance, BPS only selects a sparse set of boundary points; IESRSVDD only prunes high-density areas. As expected, RAPID selects both the boundary points and a uniformly distributed set of inner points. The decision boundary of RAPID matches the one obtained from the full data set perfectly. Only three competitors (DAEDS, IESRSVDD, and NDPSR) also result in an accurate decision boundary. But all of them produce significantly larger sample sizes than RAPID.

Q2 To what extent do data characteristics influence a sample and the resulting classification quality?

To explore this question, we individually vary the number of observations, the dimensionality, and the number of the mixture components. In the following visualizations, an optimal sampling always yields a MCC of 1 in the upper row and very small sample sizes in the bottom row, i.e., altering any data characteristic does not influence the sampling. Some values for the competitors are missing since the sample has been empty.

Number of observations: Ceteris paribus, increasing the number of observations should not have a significant impact on the observations selected. This expectation is reasonable, since increasing the data size does not change the underlying distribution and the true decision boundary. Figure 5a graphs the sample quality and sample size for the different methods. Many competitors (BPS, IESRSVDD, KFNCB, and DAEDS) do not scale well with more observations, i.e., the sample sizes increase significantly. BPS scales worst and only removes a tiny fraction of observations. Further, the sample quality drops significantly with more than 500 observations for some competitors (DBSRSVDD and HSR). RAPID on the other hand is robust with increasing data size, for both sample quality and sample size. The sample sizes returned are small, even for large data sets, and the resulting quality is always close to MCC = 1.0.

Dimensionality: The expectation is that the sample quality does not deteriorate with increasing dimensionality. However, sample sizes may increase slightly. This is because determining a decision boundary of a high-dimensional manifold requires more observations than of a low-dimensional one. Figure 5b shows the sample quality and size. For some competitors (HSR, NDPSR, and KFNCBD), sample quality decreases with increasing dimensionality. This indicates that they do not select observations in all regions. This in turn leads to misclassification. Even tuning exogenous parameter values does not mitigate these effects. As desired, RAPID returns a small sample in all cases, with high classification accuracy.

Number of Mixture Components: Finally, we make the data set more difficult by increasing the number of Gaussian mixture components. Like before, we expect sample sizes to increase slightly, since the generated manifolds are more difficult to classify. Figure 5c shows the sample quality and size. For HSR and DBSRSVDD, sampling quality fluctuates significantly. NDPSR and DBSRSVDD do not prune any observation with only one component. We think that these effects are due to the sensitivity to the exogenous parameters of the various methods. This is, methods with fluctuating results would require different parameter values for data sets of different difficulties. However, the competitors do not come with a systematic way to choose parameter values to adapt to varying data set difficulty. RAPID in turn is very robust to changes in difficulty. As expected, the sample size increases only slightly with increasing difficulty. The classification accuracy is close to MCC = 1.0, even for high difficulties.

In summary, our experiments on synthetic data reveal that many competitors are sensitive to data size, dimensionality, and complexity. Different parameter values may mitigate the effects in a few cases, but selecting good values is difficult. RAPID on the other hand is very robust. It adapts well to different data characteristics and does not require any parameter tuning.

5.3 Benchmark on Real-World Data

Next, we turn to data sets with real distributions and more diverse data characteristics. The basis for our experiments are 23 standard benchmark data sets for outlier detection Campos et al. (2016). Campos et al. constructed this benchmark

[2] Because of limited space, we report median statistics, but results also hold for mean values and individual comparisons (ranks), see `https://www.ipd.kit.edu/ocs`.

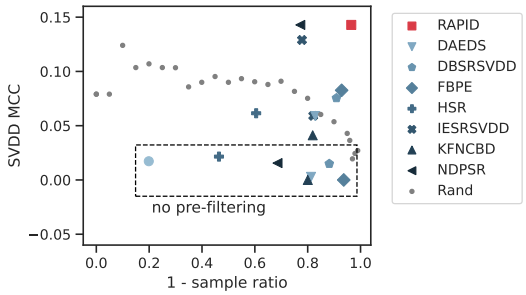

Fig. 6 Median MCC and ratio of observations removed by sampling (1 - sample ratio = $(N-|\mathbf{S}|)/|\mathbf{X}|$) over real-world data.[2]

Table 2 Median metrics over real-world data.[2]

	runtimes			sample		quality
	t_{samp}	t_{train}	t_{class}*	size	ratio	MCC
RAPID	0.01	0.02	0.00	18.0	0.04	0.14
BPS†	0.08	0.29	0.01	279.0	0.60	†
DAEDS	0.35	0.03	0.00	77.0	0.17	0.06
DBSRSVDD	0.01	0.02	0.00	35.0	0.09	0.08
FBPE	0.04	0.02	0.00	40.0	0.07	0.08
HSR	0.12	0.04	0.00	111.0	0.40	0.06
IESRSVDD	0.01	0.05	0.00	127.0	0.22	0.13
KFNCBD	0.27	0.03	0.00	80.0	0.18	0.04
NDPSR	0.04	0.04	0.00	87.0	0.23	0.14

* time for classification in seconds per 1000 observations.
† did not solve for large data sets.

from classification data where one of the classes is downsampled and labeled as outlier. The data sets have different sizes (80 to 49 534 observations), dimensionality (3 to 1555 dimensions) and outlier ratios (0.2 % to 75.38 %, median 9.12 %[3]). Again, we structure our experiments along two questions.

Q3 How well do methods adapt to real-world data sets?

First, we compare RAPID against competitors without any pre-processing. Figure 6 plots the median sample ratio against the SVDD quality over all data sets.[2] Good sampling methods return small sample ratios and yield high SVDD quality, i.e., they appear in the upper right corner of the plot. Rand is shown for different $r \in [0.01, 1.0]$. All of the competitors in their original version, i.e., without pre-filtering, result in poor SVDD quality, much lower than the Rand baselines. The reason is that they expect all observations to be inliers. BPS with pre-filtering did not yield any solution for large data sets.

With our *pre-filtering*, SVDD qualities of competitors improve considerably, see Figure 6 and Table 2. Still, RAPID outperforms its competitors; none of them produces a sample with higher SVDD quality or smaller sample size than RAPID.

[3] Only the data set "Parkinson" has an outlier percentage higher than 40%.

The methods closest to RAPID are IESRSVDD and NDPSR, with similar SVDD quality, but significantly larger sample sizes. On average, the sample selected by RAPID even yields the same quality as training a SVDD without sampling.[4] Figure 7 in the Appendix of this article features a more detailed evaluation per data set.

Q4 What are the runtime benefits of sampling?

Next, we look at the impact of sampling on algorithm runtimes, see Table 2. We measure the execution runtimes of the sampling method (t_{samp}), of SVDD training on the sample (t_{train}), and of the classification (t_{class}). Overall, all methods have reasonable runtimes for sampling, with DAEDS being the slowest with 0.35 s on average. However, RAPID is the fastest method overall Methods with runtimes similar to RAPID, such as DBSRSVDD, feature significantly lower SVDD quality. For the big data sets (ALOI and KDDCup99), RAPID, DBSRSVDD, FBPE, and HSR have a t_{samp} of around one minute or less, see Figure 8 and Table 3 in the Appendix of this article. RAPID achieves the highest classification quality nevertheless, even compared to the slower competitors. Compared to SVDD applied to large original data sets without sampling, RAPID reduces training times from over one hour to only a few seconds.[4]

Finally, we look at the statistical significance of our results. We perform a Friedman test with a pairwise comparison of the methods via a post-hoc Neményi test for three metrics: SVDD quality (MCC), sample ratio ($|\mathbf{S}|/|\mathbf{X}|$) and algorithm runtime t_{samp}. The test on SVDD quality confirms that no other method is significantly better than RAPID. Yet RAPID produces significantly smaller samples ($p < 0.01$ for all competitors except for FBPE where $p < 0.05$). RAPID also is significantly faster at sampling the data set than BPS, DAEDS, DBSRSVDD, KFNCBD, and NDPSR, the closest competitor in terms of quality ($p < 0.01$). For more details see Figure 9, Figure 10, and Figure 11 in the Appendix of the article.

In summary, RAPID outperforms its competitors on real-world data as well. There is no other method with higher SVDD quality and similarly small sample sizes. RAPID scales very well to very large data sets and reduces overall runtimes by up to an order of magnitude.

6 Conclusions

SVDD does not scale well to large data sets due to long training runtimes. Therefore, working with a sample instead of the original data has received much attention in the literature. Various existing sampling approaches guess the support vectors of the original SVDD solution from data characteristics. These methods are difficult to calibrate because of unintuitive exogenous parameters. They also tend to perform poorly regarding outlier detection. One reason is that including support vector candidates in the sample does not guarantee them to indeed become support vectors.

Our article addresses these issues. We formalize SVDD sample selection as an optimization problem, where constraints guarantee that SVDD indeed yields the

[4] Based on data sets with non-prohibitive runtime, i.e., $N < 25\,000$, see `https://www.ipd.kit.edu/ocs` for details.

correct decision boundaries. We achieve this by reducing SVDD to a density-based decision problem, which gives way to rigorous arguments why a sample indeed retains the decision boundary. To solve this problem effectively, we propose a novel iterative algorithm RAPID. RAPID does not rely on any parameter tuning beyond the one already required by SVDD. It is efficient and consistently produces a small high-quality sample. Experiments show that the way we have framed sampling as an optimization problem improves substantially on existing methods with respect to runtimes, sample sizes, and classification accuracy.

Declarations

Funding Not Applicable.

Conflicts of interest/Competing interests Due to previous joint work, please do not invite the following individuals to become a reviewer of our manuscript:

- Kristian Kersting (Editorial Board), Technische Universität Darmstadt
- Johannes Fürnkranz (Editorial Board), Johannes Kepler University Linz
- Eyke Hüllermeier (Editorial Board), University of Marburg
- Stefano Teso (Editorial Board), University of Helsinki

Ethics approval Not Applicable.

Consent to participate Not Applicable.

Consent for publication Not Applicable.

Availability of data and material All data and material used in this article is available on our website.[5] The website contains links to our raw results, figures and tables. The real-world data sets used in Section 5.3 were published in Campos et al. (2016) and are available on their website.[6]

Code availability All code used for reproducing our experiments, evaluation, and plotting for this article is open source, MIT licensed and available on Github.[7]

Author contributions All authors contributed to the algorithm conception and design. Experimental evaluation was performed by Adrian Englhardt. The first draft of the manuscript was written by Adrian Englhardt and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

[5] `https://www.ipd.kit.edu/ocs`

[6] `https://www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI/`

[7] `https://github.com/englhardt/ocs-evaluation/`

References

Achlioptas D, McSherry F, Schölkopf B (2002) Sampling techniques for kernel methods. In: NIPS

Achtert E, Kriegel HP, Reichert L, Schubert E, Wojdanowski R, Zimek A (2010) Visual evaluation of outlier detection models. In: DASFAA, Springer

Aggarwal CC (2015a) Data mining: the textbook. Springer

Aggarwal CC (2015b) Outlier analysis. Springer

Alam S, Sonbhadra SK, Agarwal S, Nagabhushan P, Tanveer M (2020) Sample reduction using farthest boundary point estimation (fbpe) for support vector data description (svdd). Pattern Recognition Letters

Bakır GH, Weston J, Schölkopf B (2004) Learning to find pre-images. NIPS

Bezanson J, Edelman A, Karpinski S, Shah VB (2017) Julia: A fresh approach to numerical computing. SIAM Review

Campos G, Zimek A, Sander J, Campello RJGB, Micenková B, Schubert E, Assent I, Houle ME (2016) On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Min Knowl Disc

Chaudhuri A, Kakde D, Jahja M, Xiao W, Kong S, Jiang H, Percdriy S (2018) Sampling method for fast training of support vector data description. In: RAMS, IEEE

Chen Y, Li S (2019) A lightweight anomaly detection method based on svdd for wireless sensor networks. Wireless Personal Communications 105(4):1235–1256

Chu CS, Tsang IW, Kwok JT (2004) Scaling up support vector data description by using core-sets. In: IJCNN, IEEE

Fine S, Scheinberg K (2001) Efficient svm training using low-rank kernel representations. JMLR

Halimu C, Kasem A, Newaz SS (2019) Empirical comparison of area under roc curve (auc) and mathew correlation coefficient (mcc) for evaluating machine learning algorithms on imbalanced datasets for binary classification. In: ICMLSC, pp 1–6

Hu C, Zhou B, Hu J (2014) Fast support vector data description training using edge detection on large datasets. In: IJCNN, IEEE

Kim PJ, Chang HJ, Song DS, Choi JY (2007) Fast support vector data description using k-means clustering. In: ISNN, Springer

Krawczyk B, Triguero I, García S, Woźniak M, Herrera F (2019) Instance reduction for one-class classification. KAIS

Kwok JY, Tsang IH (2004) The pre-image problem in kernel methods. Trans Neural Networks

Li D, Wang Z, Cao C, Liu Y (2018) Information entropy based sample reduction for support vector data description. Applied Soft Computing

Li Y (2011) Selecting training points for one-class support vector machines. Pattern Recognition Letters

Li Z, Wang L, Yang Y, Du X, Song H (2019) Health evaluation of mvb based on svdd and sample reduction. IEEE Access

Liao Y, Kakde D, Chaudhuri A, Jiang H, Sadek C, Kong S (2018) A new bandwidth selection criterion for using svdd to analyze hyperspectral data. In: Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery, SPIE

Liu YH, Liu YC, Chen YJ (2010) Fast support vector data descriptions for novelty detection. Trans Neural Networks

Mika S, Schölkopf B, Smola AJ, Müller KR, Scholz M, Rätsch G (1999) Kernel pca and de-noising in feature spaces. In: NIPS

Nguyen X, Huang L, Joseph AD (2008) Support vector machines, data reduction, and approximate kernel matrices. In: ECML, Springer

Peng X, Xu D (2012) Efficient support vector data descriptions for novelty detection. Neural Comput Appl

Platt J (1998) Sequential minimal optimization: A fast algorithm for training support vector machines. Tech. rep., Microsoft Research

Qu H, Zhao J, Zhao J, Jiang D (2019) Towards support vector data description based on heuristic sample condensed rule. In: CCDC, IEEE

Schölkopf B, Smola AJ, Williamson RC, Bartlett PL (2000) New support vector algorithms. Neural Computation

Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC (2001) Estimating the support of a high-dimensional distribution. Neural Comput DOI 10.1162/089976601750264965

Scott DW (2015) Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons

Sun W, Qu J, Chen Y, Di Y, Gao F (2016) Heuristic sample reduction method for support vector data description. Turkish Journal of Electrical Engineering & Computer Sciences

Tax D, Duin R (2004) Support vector data description. Machine Learning

Trittenbach H, Englhardt A, Böhm K (2018) An overview and a benchmark of active learning for outlier detection with One-Class classifiers. arXiv:180804759

Trittenbach H, Böhm K, Assent I (2019a) Active learning of svdd hyperparameter values. arXiv:191201927

Trittenbach H, Englhardt A, Böhm K (2019b) Validating one-class active learning with user studies – a prototype and open challenges. ECML PKDD Workshop

Vert R, Vert JP (2006) Consistency and convergence rates of One-Class SVMs and related algorithms. JMLR

Williams CK, Seeger M (2001) Using the nyström method to speed up kernel machines. In: NIPS

Xiao Y, Liu B, Hao Z, Cao L (2014) A k-farthest-neighbor-based approach for support vector data description. Applied intelligence

Yang T, Li YF, Mahdavi M, Jin R, Zhou ZH (2012) Nyström method vs random fourier features: A theoretical and empirical comparison. In: NIPS

Zhu F, Ye N, Yu W, Xu S, Li G (2014) Boundary detection and sample reduction for one-class support vector machines. Neurocomputing

Appendix

Algorithm 3: Synthetic data generation

Input : number inliers N_{in}, number outliers N_{out}, dimensionality M, number of clusters N_{c}, cluster standard deviation σ_{cluster}, Kernel function $k(x_i, x_j)$, threshold probability $\theta_{\text{pre}} \in [0, 1]$, data range $r_{\min}, r_{\max} = 0, 1$

Output: Synthetic inliers $\mathcal{I}$ and outliers $\mathcal{O}$

$N_1, \ldots, N_{N_{\text{c}}} = \lfloor {}^{N_{\text{in}}}/_{N_{\text{c}}} \rfloor$

for $i \leftarrow 1 \ldots (N_{in} \mod N_c)$ **do** ▷ distribute remaining inliers

  $N_i = N_i + 1$

end

▷ `Create Gaussian Mixture`

center-box $= (r_{\min} + (r_{\max} - r_{\min}) \cdot 0.2, r_{\min} + (r_{\max} - r_{\min}) \cdot 0.8)$

for $i \leftarrow 1 \ldots N_c$ **do**

  Draw M-dimensional vector μ_i from $\mathcal{U}_{\text{center-box}}$

  $\theta_i = (\mu_i, \sigma_{\text{cluster}})$

end

▷ `Generate Inliers`

$\mathcal{I}_i, \ldots \mathcal{I}_{N_{\text{c}}} = \emptyset$

for $i \leftarrow 1 \ldots N_c$ **do**

  while $|\mathcal{I}_i| \leq N_i$ **do**

    Draw M-dimensional vector x from $\mathcal{N}(\mu_i, \sigma_i)$ with $p(x \mid \theta_i) \geq \theta_{\text{pre}}$

    $\mathcal{I}_i = \mathcal{I}_i \cup \{x\}$

  end

end

$\mathcal{I} = \bigcup_{i \leftarrow 1, \ldots, N_{\text{c}}} \mathcal{I}_i$

▷ `Generate Outliers`

$\mathcal{O} = \emptyset$

while $|\mathcal{O}| \leq N_{out}$ **do**

  Draw M dimensional vector x from $\mathcal{U}_{[r_{\min}, r_{\max}]}$ where $\forall i \in \{1, \ldots, N_{\text{c}}\} : p(x \mid \theta_i) < \theta_{\text{pre}}$

  $\mathcal{O} = \mathcal{O} \cup \{x\}$

end

return $\mathcal{I}$, $\mathcal{O}$

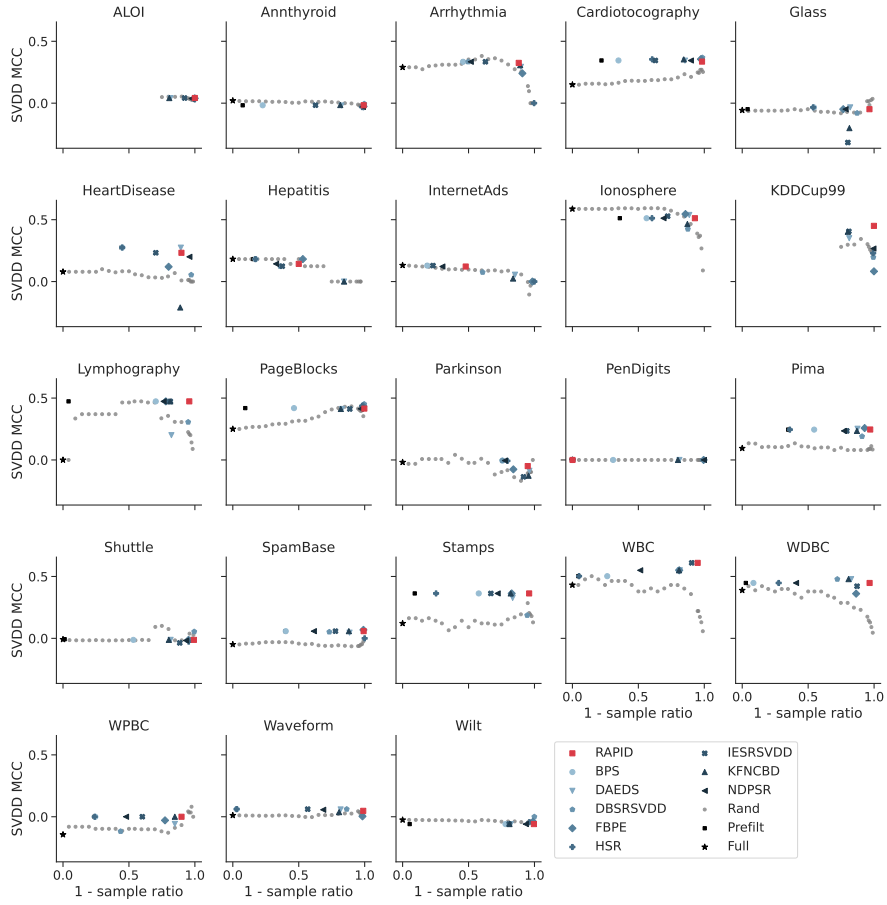

Fig. 7 MCC and ratio of observations removed by sampling (1 - sample ratio = $^{(N-|\mathbf{S}|)}/_{|\mathbf{X}|}$) for each real-world data set. *Prefilt* is the performance of an SVDD trained on the sample after pre-filtering, and the values are equivalent to the performance of the level-set classifier. *Full* is an SVDD trained on the full data set.

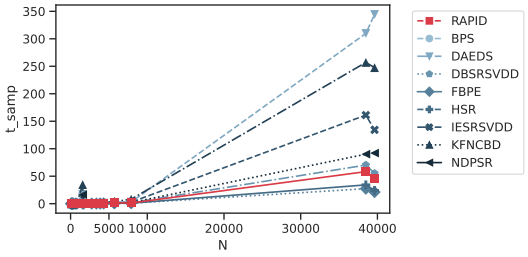

Fig. 8 Data set size and sampling time t_{samp} over real-world data.

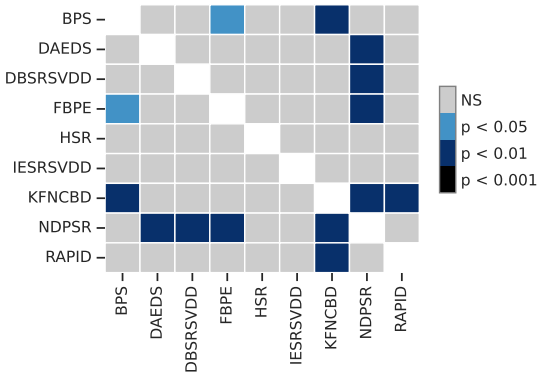

Fig. 9 Statistical significance p-values after a Friedman and post-hoc Neményi test for the resulting SVDD quality measured via MCC over real-world data.

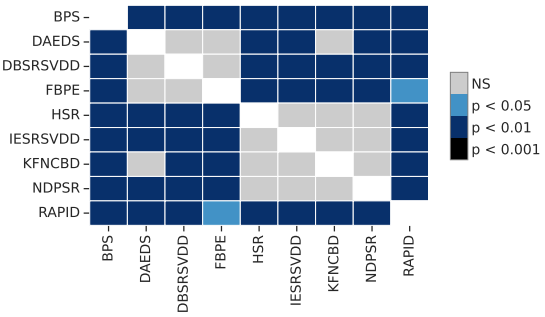

Fig. 10 Statistical significance p-values after a Friedman and post-hoc Neményi test for sampling ratio ($|\mathbf{S}|/|\mathbf{X}|$) over real-world data.

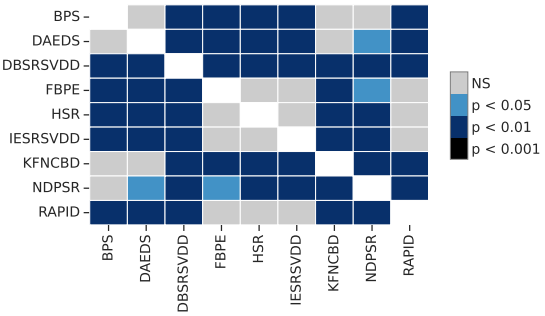

Fig. 11 Statistical significance p-values after a Friedman and post-hoc Neményi test for the sampling time t_{samp} over real-world data.

Table 3 Metrics over two large real-world data sets.

Data set	Method	t_{samp}	t_{train}	$t_{\text{class}}{}^{*}$	ratio	MCC
ALOI	DAEDS	344.2817	3847.4930	13.1636	0.1844	0.0410
	DBSRSVDD	55.0896	0.7726	0.9503	0.0106	0.0357
	FBPE	21.1945	0.0198	0.4990	0.0010	0.0392
	HSR	24.0747	0.0204	0.1129	0.0010	0.0306
	IESRSVDD	134.3110	296.5463	8.7749	0.0777	0.0416
	KFNCBD	247.4868	4607.4732	12.9629	0.1939	0.0412
	NDPSR	92.1190	8.3500	2.5350	0.0250	0.0375
	RAPID	45.8714	0.0184	0.0638	0.0005	0.0420
KDDCup99	DAEDS	309.7272	3865.6893	14.6692	0.1885	0.3501
	DBSRSVDD	70.2679	0.1531	0.5541	0.0057	0.1966
	FBPE	27.3335	0.0207	0.4622	0.0010	0.0826
	HSR	34.2100	0.2661	0.5596	0.0069	0.2296
	IESRSVDD	161.1876	3969.8312	17.6572	0.1913	0.4056
	KFNCBD	256.9911	4591.3423	16.2102	0.1992	0.4025
	NDPSR	90.1258	0.1127	0.4317	0.0050	0.2667
	RAPID	58.3369	0.0229	0.1312	0.0013	0.4501

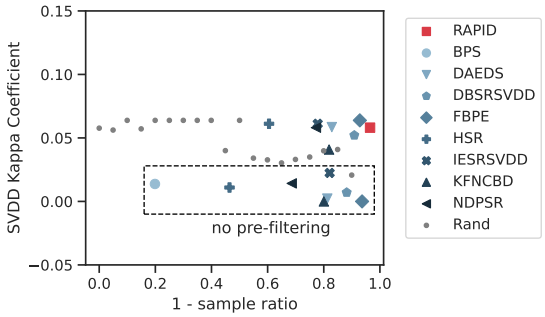

Fig. 12 Median Cohen's kappa coefficient and ratio of observations removed by sampling (1 - sample ratio = $(N-|\mathbf{S}|)/|\mathbf{X}|$) over real-world data.

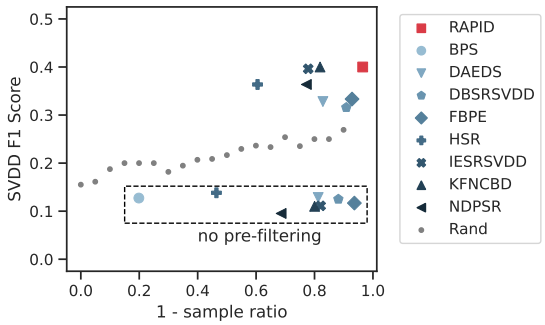

Fig. 13 Median F1-score and ratio of observations removed by sampling (1 - sample ratio = $(N-|\mathbf{S}|)/|\mathbf{X}|$) over real-world data.

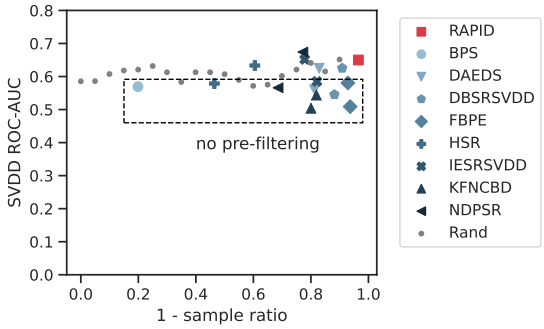

Fig. 14 Median ROC-AUC and ratio of observations removed by sampling (1 - sample ratio $= {}^{(N-|\mathbf{S}|)}/_{|\mathbf{X}|}$) over real-world data.