Obey validity limits of data-driven models

Artur M. Schweidtmann[1,*] Jana M. Weber[2] Christian Wende[1]
Linus Netze[1] Alexander Mitsos[3,1,4]

October 7, 2020

[1] Process Systems Engineering (AVT.SVT), RWTH Aachen University, Aachen, Germany.
[2] Department of Chemical Engineering and Biotechnology, University of Cambridge, Cambridge, United Kingdom.
[3] JARA-CSD, 52056 Aachen, Germany.
[4] Institute of Energy and Climate Research, Energy Systems Engineering (IEK-10), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany.
* artur.schweidtmann@rwth-aachen.de

Abstract

Data-driven models are becoming increasingly popular in engineering, on their own or in combination with mechanistic models. Commonly, the trained models are subsequently used in model-based optimization of design and/or operation of processes. Thus, it is critical to ensure that data-driven models are not evaluated outside their validity domain during process optimization. We propose a method to learn this validity domain and encode it as constraints in process optimization. We first perform a topological data analysis using persistent homology identifying potential holes or separated clusters in the training data. In case clusters or holes are identified, we train a one-class classifier, i.e., a one-class support vector machine, on the training data domain and encode it as constraints in the subsequent process optimization. Otherwise, we construct the convex hull of the data and encode it as constraints. We finally perform deterministic global process optimization with the data-driven models subject to their respective validity constraints. To ensure computational tractability, we develop a reduced-space formulation for trained one-class support vector machines and show that our formulation outperforms common full-space formulations by a factor of over 3,000, making it a viable tool for engineering applications. The method is ready-to-use and available open-source as part of our MeLOn toolbox (`https://git.rwth-aachen.de/avt.svt/public/MeLOn`).

1 Introduction

Supervised machine-learning techniques have been re-emerging as a promising avenue for data-driven modeling in various engineering disciplines (Venkatasubramanian, 2019). In

most applications, the overall goal is the optimal decision-making based on available data and *a priori* knowledge. Thus, data-driven models and mechanistic models are often combined to form hybrid models (Mogk et al., 2002; Kahrs and Marquardt, 2008; Von Stosch et al., 2014; Glassey and Von Stosch, 2018). Subsequently, hybrid models are frequently used in model-based optimization of design and/or operation of processes (McBride and Sundmacher, 2019; Schweidtmann and Mitsos, 2019).

A critical issue of data-driven models is their limited extrapolability. Unless strong assumptions are posed on the learned function, data-driven models can only be valid in regions where they have sufficiently dense coverage of training data points. We refer to this as the *validity domain* (Courrieu, 1994). Consequently, there is a need to avoid the evaluation of data-driven models outside their validity domain during optimization. Note that we refer to the validity domain of individual data-driven models throughout this work, but the concept can also be applied to hybrid models (Kahrs and Marquardt, 2007).

The vast majority of previous publications use box constraints (i.e., hyperrectangles) to bound the inputs of data-driven models, i.e., each variable has independent bounds. This approach is practical when the training data is obtained from simulations based on regular grids or Latin hypercubes that are sufficiently dense. It is also advantageous for local and global optimization. However, it requires *a priori* known bounds and the possibility to obtain training data for any input combination. In practice, simulations can fail (Asprion, 2020) and industrial process data usually does not cover hyperrectangular spaces (Asprion et al., 2019). This leads to manual selection of wrong bounds, which may cut off optimal solutions or overestimate the validity domain.

As proposed by Courrieu (1994), a few previous works in process systems engineering (PSE) constructed the convex hull of the training data points to describe the validity domain and integrated it as a set of linear constraints in optimization problems (Kahrs and Marquardt, 2007; Zhang et al., 2016; Asprion et al., 2019). By definition, the convex hull is the smallest convex set that contains all data points. Commonly, evaluations of data-driven models inside the convex hull of the training data are called interpolation and outside extrapolation. However, roughly speaking, the convex hull cannot distinguish between for potential holes in the training data set, gaps between separated clusters of data, and nonconvex boundaries. Thus, staying within the convex hull seems only a necessary condition for the validity of data-driven models and not sufficient. Identifying if the convex hull is a suitable model for the data domain is very challenging in high dimensions. Notably, Zhang et al. (2016) extended the convex hull method to the union of multiple polytopes by introducing binary variables and additional constraints to the problem. However, this algorithm becomes impractical when the number of data points or their dimension is very high. Besides convex hull formulations, there exist also several publications that circumvent extrapolation problems en passant in different ways. For instance, Mistry et al. (2018) penalize deviations from a training data mean in a space that is parameterized using principal component analysis. Rall et al. (2019) constrain the maximal allowed distance from the nearest training data point resulting in a nonsmooth optimization problem. Kumar et al. (2019) train multiple data-driven models on a design problem and reject designs where the variation between the models is large. Similarly, Pinto et al. (2019) use bootstrap aggregation to estimate error bounds for hybrid

mechanistic/data-driven models. There exist further methods that quantify the variance or confidence interval of predictions such as Bayesian methods and maximum likelihood estimations (Papadopoulos et al., 2001). However, this leads to chance-constrained programming problems (Charnes and Cooper, 1959; Schweidtmann et al., 2020a). Likewise, there are a few studies on the adaptive exploration of the design space (Larson and Mattson, 2012; Chen et al., 2018; Knudde et al., 2019) and related works on constrained Bayesian optimization (Shahriari et al., 2016). However, we focus on fixed training data sets in this study while the extension of our method to adaptive sampling is a promising future research.

An alternative to box constrains and convex hull is to use a nonlinear classifier that can also model complicated validity domains. A few previous studies in mechanical engineering (Malak and Paredis, 2010; Roach et al., 2011) used Support Vector Domain Description (SVDD) (Tax and Duin, 1999) to model the validity domain of data-driven equipment models. Also, Quaglio et al. (2018) use binary support vector classification to include reliability constraints into model-based design of experiment. As only valid training data points are given in most engineering applications, we consider one-class classification in this work. There exists a broad variety of one-class classifiers that can be divided into density methods, boundary methods, and reconstruction methods (Tax, 2001). Also, one-class classification is closely related to novelty, outlier, or anomaly detection (Chandola et al., 2009; Pimentel et al., 2014; Khan and Madden, 2009, 2014; Ding et al., 2014). The previous literature indicates that one-class support vector machines (SVMs) (Schölkopf et al., 2000) are common and suitable for the problem at hand. Compared to density models, less training data is required to construct the boundary, since only the boundary is estimated and not a complete density distribution (Tax, 2001). In addition, the one-class SVM is tolerant to outliers in the training set (Pimentel et al., 2014).

Optimization problems with one-class SVMs embedded are nonconvex. Thus, deterministic global optimization is desirable to identify global solutions. However, these models lead to large-scale optimization problems that are difficult to solve. In our previous work, we showed that a reduced-space (RS) formulation and the use of McCormick relaxations are advantageous for the optimization of two other important classes of data-driven models, namely artificial neural networks (Schweidtmann and Mitsos, 2019) and Gaussian processes (Schweidtmann et al., 2020a). We propose a similar idea here for one-class SVM.

The global shape of data matters because it often provides important information about the underlying phenomena represented by the data. Especially in high-dimensional data, topological data analysis (TDA) can reveal and quantify objects and features not directly visible to the human eye. In the context of this work, it provides valuable information about typologies in the training data that can be colloquially thought of as holes or separated clusters. TDA was initiated relatively recently (Letscher et al., 2002; Zomorodian and Carlsson, 2005). Its roots lie in applied (algebraic) topology and computational geometry (Chazal and Michel, 2017) and it is commonly used to account for higher-order interactions in data, to comprehend mesoscale structures, or to compare different data spaces (Patania et al., 2017). The most common TDA method is persistent homology (Wasserman, 2018). So far, there are only a few applications of persistent ho-

mology in the fields of (bio-)chemical engineering and material science (Hiraoka et al., 2016; Saadatfar et al., 2017; Xia, 2018; Xia et al., 2019; Smith et al., 2020).

We propose a three-step approach to model the validity domain of data-driven models for optimization. We first perform TDA using persistent homology. In case clusters or holes are identified, we train a one-class SVM on the training data domain of the data-driven models and encode it as constraints in the subsequent process optimization. Otherwise, we construct the convex hull of the data and encode it as constraints. We finally perform deterministic global process optimization with the data-driven models and their respective validity constraints. To ensure computational tractability, we develop a RS formulation for trained one-class SVMs. Moreover, we employ convex and concave envelopes of kernel functions to accelerate optimization. We demonstrate the potential of our method on a set of illustrative mathematical case studies and an engineering case study, i.e., the open-loop control of a sulfur recovery unit.

2 Methodology

As illustrated in Figure 1, we propose a three step approach to obey validity limits of data-driven models during optimization. In the first step, we conduct a TDA of the training data. In the second step, we either construct the convex hull of the data or we train a one-class classifier, i.e., a SVM. In the third step, we embed the trained classifier or convex hull in the optimization problem and solve it to global optimality. The described methods are available open-source. We use the *Ripser.py* toolbox that is available open-source under MIT license in Python for performing the TDA (Tralie et al., 2018). The training of the one-class SVM is performed by Scikit-learn (Pedregosa et al., 2011) and the convex hulls are identified using SciPy (Virtanen et al., 2020). We provide the one-class SVM within the "MeLOn - **M**achin**e** **L**earning Models for **O**ptimizatio**n**" toolbox under the Eclipse public license (Schweidtmann et al., 2020b). The resulting optimization problems are solved using our open-source global solver MAiNGO (Bongartz et al., 2018).

2.1 Topological data analysis using persistent homology

In persistent homology, we are interested in so-called topological invariants, i.e., properties that are invariant under homeomorphisms. The topological invariants of interest are homology groups, i.e., H_k of dimension k, with $\beta_k = \dim(H_k)$ being the Betti numbers (Binchi et al., 2014; Chung et al., 2015). "Informally, β_0 is the number of connected components, β_1 is the number of two-dimensional holes or "handles" and β_2 is the number of three-dimensional holes or "voids" etc." (Binchi et al., 2014).

The topological invariants are computed by representing the original dataset, i.e., a point cloud, as a simplicial complex through a simplicial filtration. We utilize the common Vietoris-Rips filtration, where a n-simplex in the simplicial complex is formed if and only if the pairwise distance between all points in the n-simplex is at most ϵ. At the bottom of Figure 2, we show a series of simplicial complexes for an illustrative point cloud.

Persistent homology studies topological invariants that persist over multiple length scales (ϵ) in the data (Chambers and Letscher, 2018; Otter et al., 2017; Xia, 2018; Xia

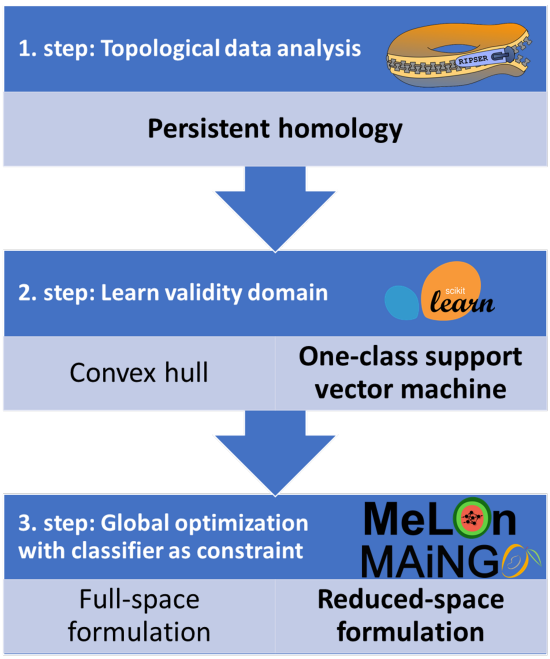


Figure 1: Overview of the proposed three step methodology to obey validity limits of data-driven models in optimization

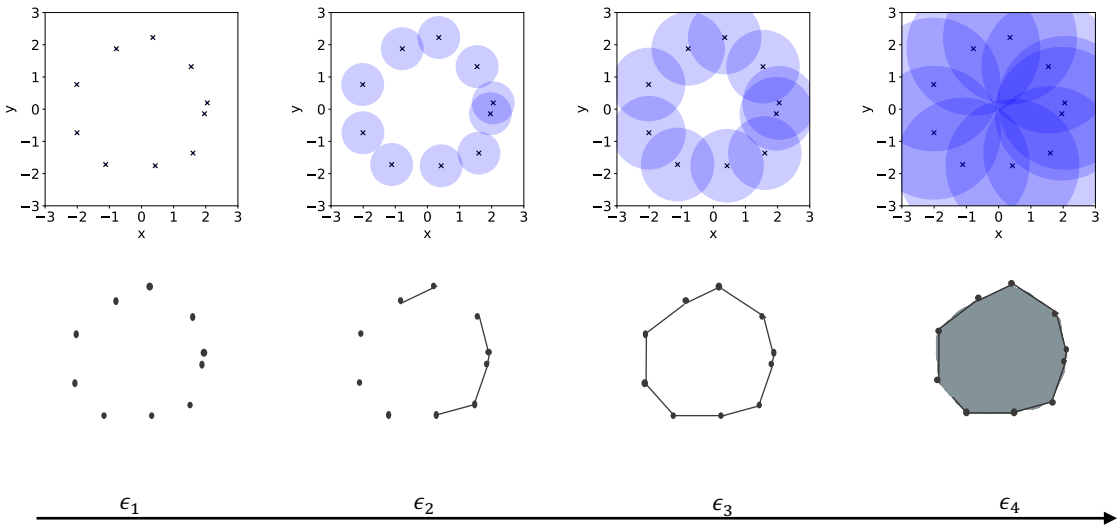


Figure 2: Illustration of a Vietoris-Rips filtration utilized for persistent homology. The upper part shows the data set and circles around the data points with increasing diameter ϵ. The bottom image illustrates the simplicial complexes formed during the filtration. The figure is based on Kimura and Imai (2017)

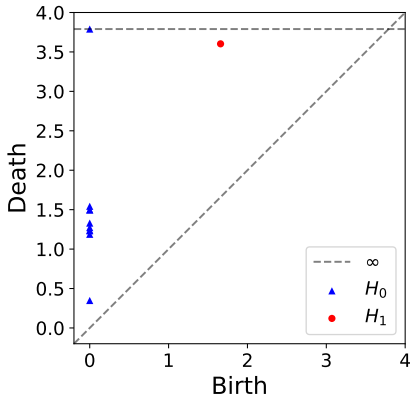


Figure 3: Persistent homology plot of the illustrative point cloud. The x-axis shows the birth and the y-axis the death of the homology groups

et al., 2019). In other words, we examine the lifespan of topological invariants by increasing ϵ incrementally and constructing simplicial complexes. At the bottom of Figure 2, we can observe that $\beta_0 = 10$ connected components (H_0) exist at ϵ_1, $\beta_0 = 5$ connected components exist at ϵ_2, $\beta_0 = 1$ connected component and $\beta_1 = 1$ two-dimensional hole (H_1) exist at ϵ_3, and $\beta_0 = 1$ connected component exist at ϵ_4.

The results of the persistent homology can be depicted in barcode diagrams or persistent diagrams. We use the more common persistent diagrams in this work. The coordinates of birth and death of the homology groups in the example are shown in the persistent diagram in Figure 3. The x-axis represents the ϵ_{birth} while the y-axis the ϵ_{death} distance of H_0 and H_1 homology groups. Features with long lifespan correspond to points far from the diagonal (Wasserman, 2018). The blue triangle at the bottom left corner of the plot corresponds to the merge of two very close data points at small ϵ. The blue triangles with ϵ_{death} between 1 and 1.5 in Figure 3 resemble the merge of connected components between ϵ_2 and ϵ_3 in Figure 2, describing the decrease in Betti number β_0 from 5 to 1. The highest blue triangle illustrates that one connected component exists until infinite. The red circle represents homology group H_1 and demonstrates the birth and death of the two-dimensional hole which is formed around ϵ_3 and dies before ϵ_4 in Figure 2.

In this example, the persistent diagram shows that a hole exist providing useful insight to guide the decision process for model selection. For example, the ϵ_{death} of the H_1 components provide information about the data density. In the example, the maximal distance in the last H_1 component is at most 1.5. This is significantly smaller than the ϵ_{death} of the H_1 hole. In other words, the life span of the hole is characteristic for the dataset.

2.2 Learn validity domain using one-class support vector machines

We model the validity domain using the convex hull or one-class SVM approach. The one-class SVMs are trained using the open-source python implementation in Scikit-learn (Pe-

dregosa et al., 2011) and the convex hulls are identified using the open-source implementation in SciPy (Virtanen et al., 2020). The details of the one-class SVM are described in the following.

SVMs are a popular method for binary classification (Cortes and Vapnik, 1995) and regression (Smola and Schölkopf, 2004). One-class SVMs are a modification of these classical SVMs (Schölkopf et al., 2000) (c.f. Tax (2001) on similarity to SVDD). The goal is to learn a boundary of a given set of training points $X = \{\hat{\boldsymbol{x}}^{(1)}, ..., \hat{\boldsymbol{x}}^{(i)}, ..., \hat{\boldsymbol{x}}^{(N)}\}$ with $\hat{\boldsymbol{x}}^{(i)} \in \mathbb{R}^D$. Similar to classical SVMs, the data is mapped to high-dimensional feature space by $\phi : \mathbb{R}^D \mapsto \mathbb{R}^d$ with $d >> D$ and later solved in the dual formulation using the *kernel trick* (Schölkopf, 2001). In the feature space, a maximum-margin hyperplane is found that separates the data from the origin by solving:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, \xi_i\in\mathbb{R}, \rho\in\mathbb{R}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \rho + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i, \tag{1}$$

$$\text{s.t} \quad \boldsymbol{w}^T\phi(\hat{\boldsymbol{x}}^{(i)}) \geqslant \rho - \xi_i \qquad \forall i = \{1, .., i, ..N\}, \tag{2}$$

$$\xi_i \geqslant 0 \qquad \forall i = \{1, .., i, ..N\}, \tag{3}$$

where $\nu \in (0, 1)$ is a regularization hyperparameter, $\xi_i \in \mathbb{R}$ are slack variables, and $\boldsymbol{w}$ and ρ are the parameters of the hyperplane in high-dimensional feature space. Schölkopf et al. (2000) show that ν is an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors in the training set, which is known as the ν-property. The decision function $f_{\text{DF-P}}(\boldsymbol{x}) = \boldsymbol{w}^T\phi(\boldsymbol{x}) - \rho$ is positive if a candidate point $\boldsymbol{x}$ is classified to be within the training data domain and negative if not. The dual of (1)-(3) is the quadratic program:

$$\min_{\alpha_i\in[0,\frac{1}{\nu N}]} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j K(\hat{\boldsymbol{x}}^{(i)}, \hat{\boldsymbol{x}}^{(j)}), \tag{4}$$

$$\text{s.t} \quad \sum_{i=1}^{N}\alpha_i = 1, \tag{5}$$

where α_i are dual variables and $K(\cdot, \cdot)$ is a kernel function. Often, the radial basis kernel function $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{y}\|^2)$ with hyperparameter γ is used since it has been shown that it is best able to model the most complex boundaries (Tax, 2001). It holds that $\alpha_i = 0$ for training samples inside the learned boundary and $\alpha_i > 0$ for samples on or outside the boundaries. Samples for which $\alpha_i > 0$ are called support vectors. The decision function in the dual variables is given by $f_{\text{DF-D}}(\boldsymbol{x}) = \sum_{i\in I_{\text{sv}}}\alpha_i K(\hat{\boldsymbol{x}}^{(i)}, \boldsymbol{x}) - \rho$, where I_{sv} denotes the indexes of the support vectors in the training data (i.e., the data points with corresponding $\alpha_i > 0$). To obey validity limits of data-driven models in an optimization problem, the following inequality has to hold:

$$f_{\text{DF-D}}(\boldsymbol{x}) \geqslant 0. \tag{6}$$

The parameter ν can be estimated from an outlier fraction by using the aforementioned ν-property. This makes this method more tolerant to outliers in the training data (Pimentel et al., 2014). The hyperparameter γ controls the model complexity when using the radial

basis kernel. If a large γ is used, all samples are mapped to a small region in the feature space and the one-class SVM cannot distinguish between the samples well. In other words, the model lacks complexity. If γ is small, pairs of samples become orthogonal in the feature space. This leads to overfitting and a high number of support vectors. A common approach to identify an appropriate γ is to gradually decrease γ until the number of support vectors does not decrease much (e.g., Dreiseitl et al. (2010)). However, automatically selecting an appropriate γ is challenging (e.g.,(Evangelista et al., 2007; Xiao et al., 2014a,b)).

2.3 Optimization with classifier as constraint

We consider a global optimization problem where a classifier is used to obey validity limits of a data-driven model. In most cases, the inputs of the classifier model correspond to the degrees of freedom $\boldsymbol{x}$ of the optimization problem. The classifier can determine if a given $\boldsymbol{x}$ is feasible or infeasible by evaluating its decision function $f_{\text{DF-D}}(\cdot)$. To obey validity limits, Inequality (6) has to hold. Although the decision function is an explicit function, there exist different ways to formulate it in optimization problems. These problem formulations are equivalent as they have the same solution, but they can have a large impact on the computational performance of global optimization.

In the FS formulation, a set of nonlinear equations is provided as equality constraints while the dependent (or intermediate) variables are optimization variables. Note that there exist multiple valid FS formulations depending on the equality constraints and optimization variables provided to the solver. One representative FS formulation for optimization with one-class SVMs embedded is shown in the following:

$$\min_{\boldsymbol{x}\in\mathbb{R}^D, z_{\text{obj}}\in\mathbb{R}, d_i\in\mathbb{R}, k_i\in\mathbb{R}, \boldsymbol{z}_{\text{dd}}\in\mathbb{R}^Z} \quad z_{\text{obj}}, \tag{7}$$

$$\text{s.t} \quad z_{\text{obj}} = f_{\text{obj}}(\boldsymbol{x}, \boldsymbol{z}_{\text{dd}}), \tag{8}$$

$$\boldsymbol{h}_{\text{dd}}(\boldsymbol{x}, \boldsymbol{z}_{\text{dd}}) = \boldsymbol{0}, \tag{9}$$

$$\sum_{i\in I_{\text{sv}}} \alpha_i k_i \geqslant \rho, \tag{10}$$

$$k_i = \exp\left(-\gamma \cdot d_i\right) \qquad \forall i \in I_{\text{sv}}, \tag{11}$$

$$d_i = \|\hat{\boldsymbol{x}}^{(i)} - \boldsymbol{x}\|^2 \qquad \forall i \in I_{\text{sv}}. \tag{12}$$

Herein, Equation (7) minimizes the objective function value z_{obj} that is given by Equation (8). Note that the objective depends on the variables of the data driven model $\boldsymbol{z}_{\text{dd}}$ that are given by the solution of Equations (9). The decision function of the one-class SVM is given by the inequality constraint (10) while its intermediate variables are given by the solution of Equations (11),(12). This FS formulation has in total $D+2\cdot|I_{\text{sv}}|+\dim(\boldsymbol{z}_{\text{dd}})+1$ optimization variables, $2\cdot|I_{\text{sv}}|+\dim(\boldsymbol{z}_{\text{dd}})+1$ equality constraints, and one inequality constraint.

The equality constraints of the one-class SVM can be solved explicitly for the intermediate variables. Thus, we can directly formulate a RS formulation of the optimization

problem (c.f. (Bongartz and Mitsos, 2017)):

$$\min_{\boldsymbol{x}\in\mathbb{R}^D} \quad f_{\mathrm{RS}}(\boldsymbol{x}), \tag{13}$$

$$\text{s.t} \quad f_{\text{DF-D}}(\boldsymbol{x}) \geqslant 0. \tag{14}$$

Herein, $f_{\mathrm{RS}}(\cdot)$ is the RS formulation of the data-driven model and objective function. Thus, Equation (13) results from sequential substitutions of Equations (7)-(9). This is possible as most data-driven models such as ANNs or GPs are explicit functions (c.f. Schweidtmann and Mitsos (2019); Schweidtmann et al. (2020a)) and as the objective function is a function of the the degrees of freedom and the predictions of the data-driven model. Similarly, Equation (14) results from the substitution of Equations (10)-(12). The RS formulation has only D optimization variables and one inequality constraint.

The convex hull of a point cloud with a finite number of points can be formulated as a set of linear inequality constraints $X_{\text{convHull}} = \{\boldsymbol{x} \in \mathbb{R}^D \mid \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \leqslant \boldsymbol{0}\}$. Assuming that the convex hull has f facets, the matrix $\boldsymbol{A} \in \mathbb{R}^{f\times D}$ and the vector $\boldsymbol{b} \in \mathbb{R}^f$ (Kahrs and Marquardt, 2007). Thus, the FS and RS formulation of the convex hull are identical. Note that the data-driven model can still be formulated in the RS and FS formulation when using the linear convex hull constraints.

The RS formulation has three major advantages for global optimization: First, the problem formulation has a direct influence on the variables to be branched on. In the RS, the B&B solver branches only on the degrees of freedom $\boldsymbol{x}$. In the FS, the B&B solver branches on the degrees of freedom and also on the intermediate variables. This is undesirable given the exponential worst-case runtime of global optimization methods. Note that this issue can also be mitigated by selective branching (Epperly and Pistikopoulos, 1997). Second, the size of the subproblems that are solved during optimization is affected by the problem formulation and the method for constructing relaxations. Our previous work shows that a combination of McCormick relaxations and RS formulation can reduce the time to solve an iteration of the B&B solver significantly (Schweidtmann et al., 2020a; Bongartz, 2020). Third, global optimization solvers usually require bounds on all optimization variables. Often, meaningful bounds are known for degrees of freedom but bounds on intermediate variables can be difficult to determine. Note that this problem is mitigated by some state-of-the-art solvers through automatic bound tightening techniques.

The vast majority of previous literature approaches formulate global optimization in the FS because they frequently use modeling environments such as GAMS that essentially require an equation-oriented modeling approach. Recently, Hart et al. (2017) developed the Python-based optimization tool Pyomo which allows modeling, implementation of own solvers, and provides access to multiple solvers. Pyomo allows both FS and RS and recently Hüllen et al. (2019) demonstrated RS optimization of ANNs in BARON through Pyomo. However, BARON relies on the auxiliary variable method for relaxations which results in larger subproblems (Schweidtmann et al., 2020a). Thus, a RS formulation in BARON does not take full advantage of the RS formulation. In contrast, our open-source solver MAiNGO (Bongartz et al., 2018) relies on McCormick relaxations in the space of the original variables utilizing the library MC++ (Mitsos et al., 2009; Chachuat et al., 2015). Another open-source solver that allows for McCormick relaxations is called EAGO has been released by Wilhelm and Stuber (2020).

The optimization problems in this work are implemented in MeLOn (Schweidtmann et al., 2020b) and solved by MAiNGO (Bongartz et al., 2018). For comparison, the problems are additionally exported to GAMS and solved by the commercial solver BARON (Tawarmalani and Sahinidis, 2005). We provide the implementation of the one-class SVM in the open-source modeling toolbox MeLOn (Schweidtmann et al., 2020b).

Tight convex and concave relaxations are highly desirable in global optimization. Therefore, we use the tightest possible relaxations, i.e., the envelopes, of the radial basis function kernel in our solver MAiNGO. Note that we derived these envelopes in our previous work (Schweidtmann et al., 2020a) as the squared exponential covariance function in Gaussian processes is equivalent to the radial basis function kernel in SVMs.

3 Illustrative case studies

As high dimensional problems are difficult to visualize, we consider eight two-dimensional data sets for illustration of the proposed method in a first step. Afterwards, we consider an engineering case study in Section 4. As shown in Figure 5, the illustrative examples cover a variety of relevant scenarios. All data points are randomly generated within pre-specified bounds and perturbed by noise. Thus, the data sets do not exhibit sharp boundaries, rather they also include noisy outlier data points.

In the next step, we evaluate an adapted peaks function, $f_{\text{Peaks}} : \mathbb{R}^2 \mapsto \mathbb{R}$, on all data sets with $f_{\text{Peaks}}(x_1, x_2) = 3(1-x_1)^2 \cdot \exp\left(-x_1^2 - (x_2+1)^2\right) - 10(\frac{x_1}{5} - x_1^3 - x_2^5) \cdot \exp\left(-x_1^2 - x_2^2\right) - \frac{1}{3} \cdot \exp\left(-(x_1+1)^2 - x_2^2\right) - 1.3x_2$. Then, we train individual ANNs on the eight data sets using Keras (Chollet et al., 2015). All ANNs exhibit one input layer with 2 neurons, two hidden layers with six and eight neurons, respectively, and an output layer with one neuron. The hidden layers use tanh activation and the output layers use linear activation. For training, the inputs are scaled onto $[-1, 1]$ and the outputs are scaled to zero mean and unit variance. We further use a batch size of 128 and an epoch limit of 4,000. Note that we omit a hyperparameter study for the ANNs because ANN training is not the focus of this work.

All optimization problems are solved one core of an Intel Xeon CPU E5-2630 v3 (2.40GHz) with 192 GB RAM and Windows Server 2016 operating system. We use a 0.001 relative and absolute optimality tolerance, a CPU time limit of 1,000 seconds, and default settings in BARON and MAiNGO.

3.1 Topological data analysis

The persistent diagrams of the eight input data sets are shown in Figure 4. The Subfigures 4d and 4e show a H_1 component with a large life span each. These correspond to the holes in the respective data sets "box w/ hole" and "circle w/ hole". Moreover, the corresponding ϵ_{death} provide information about the diameter of the holes. Recall that H_1 components that are close to the diagonal have a short life span and are therefore not relevant for this analysis.

The persistent diagrams also show the existence of disjunct clusters in the data sets "two circles" and "two ovals". In Subfigures 4f and 4g, the H_0 components with a high ϵ_{death} indicate disjunct clusters and are a measure for the distance between them. Note

Oval	Two circles	Box	Two ovals	Banana	Box2	Box w/ hole	Circle w/ hole
0.31	0.28	0.25	0.35	0.25	0.25	0.23	0.25

Table 1: The values of the hyperparameter γ for the eight case studies. The hyperparameters are selected based on the incremental approach described in Section 2.2.

that the H_0 components at the infinity line persist when ϵ goes to infinity and do not die. They correspond to the H_0 components that include all data points.

The persistent diagrams show no distinct differences between the "box", "oval", "box2", and "banana" case studies. This illustrates that the method cannot distinguish between convex and nonconvex data sets in general. Therefore, the persistent diagrams cannot ensure that the convex hull is sufficient to describe the validity domain. Rather, it can only identify some cases where the convex hull is not sufficient.

3.2 Validity domain modeling

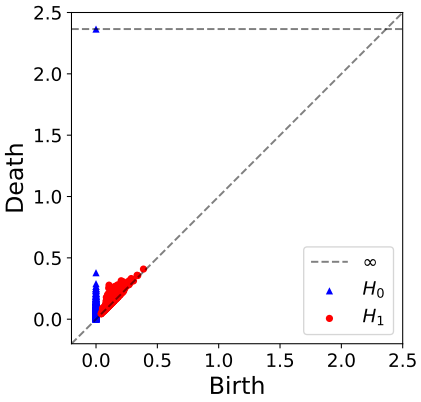
In order to compare the one-class SVM and the convex hull, we model the input data of the eight case studies with both methods. We employ the common radial basis function kernel for the one-class SVM. We set the hyperparameter ν to a low value 0.03 because there are only a few outliers through noise in the data. The hyperparameter γ is identified using the incremental approach described in Section 2.2. The selected γ values are summarized in Table 1.

The learned boundaries for the case studies are depicted in Figure 5. As expected, the convex hull does not model holes and disjunct data clusters. Instead, the convex hull overestimates the validity domain of the case studies. In contrast, the one-class SVM is able to model holes and disjunct data clusters. Furthermore, the convex hull includes all data points whereas the one-class SVM also allows for outliers in the data and excludes regions with only small data density from the validity domain (e.g., see the "box" case study).

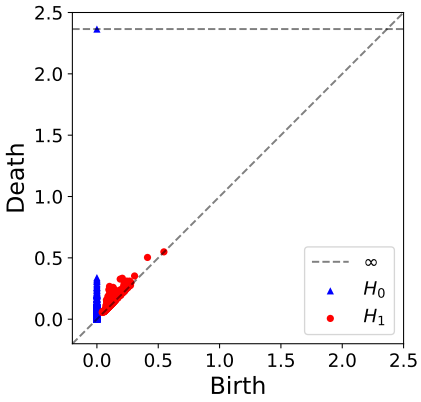
3.3 Optimization results

We minimize the prediction of the eight trained ANNs subject to the the convex hull or the one-class SVM as constraints. Table 2 shows the optimal solution points, $\mathbf{x}^*$, and objective function values, $f_{\text{ANN}}(\mathbf{x}^*)$, for the problem with convex hull and one-class SVM as constraints.

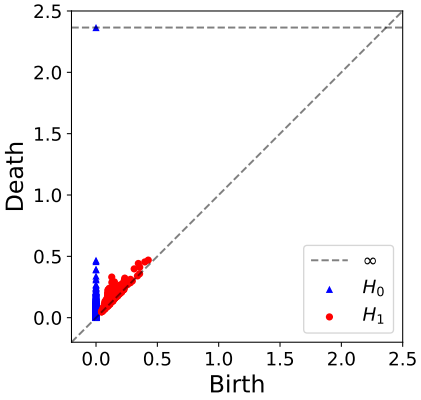
The optimal objective function values of the convex hull approach are lower than the ones with the one-class SVM for all case studies because the convex hull overestimates the validity domain. This overestimation can lead to large errors at the optimal solution points. For the "banana" case study, the optimal solution found by the convex hull approach is outside the validity domain but at the boundary of the convex hull (see Table 2). This leads to a wrongly estimated objective values by the ANN of -5.2 with an absolute error of 8.52. In contrast, the one-class SVM models the validity domain accurately and



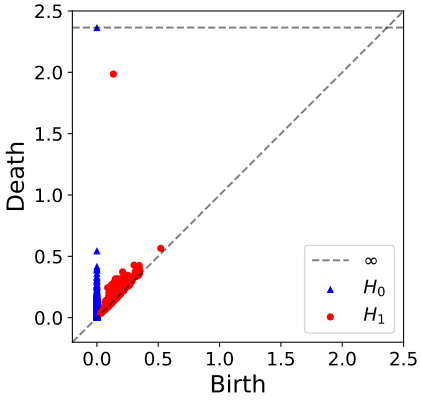
(a) “Box” case study



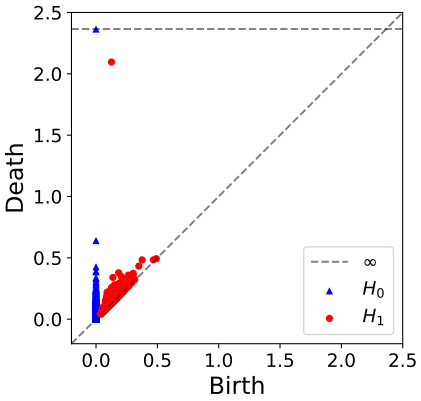
(b) “Oval” case study



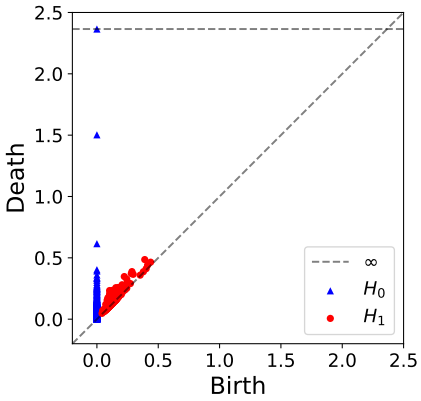
(c) “Box2” case study



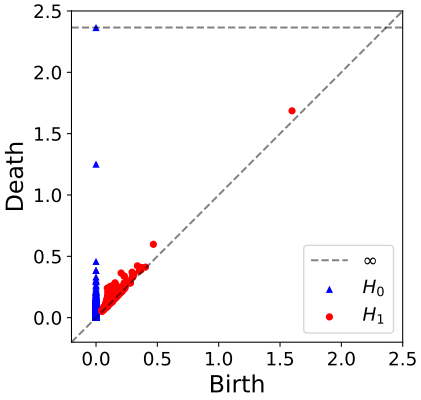
(d) “Box w/ hole” case study



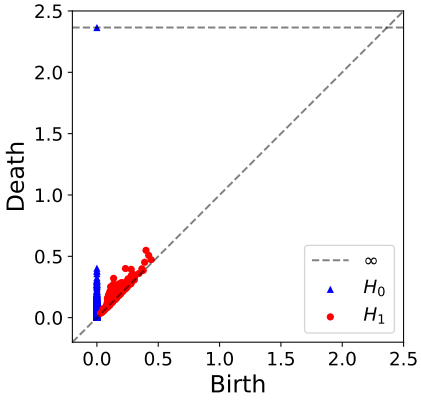
(e) “Circle w/ hole” case study



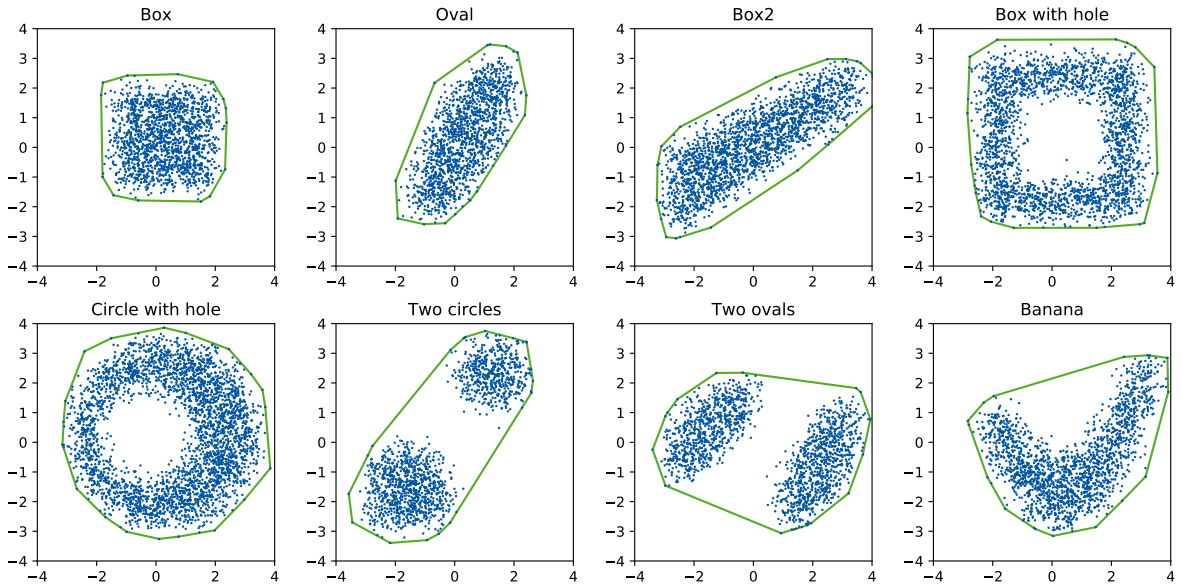
(f) “Two circles” case study



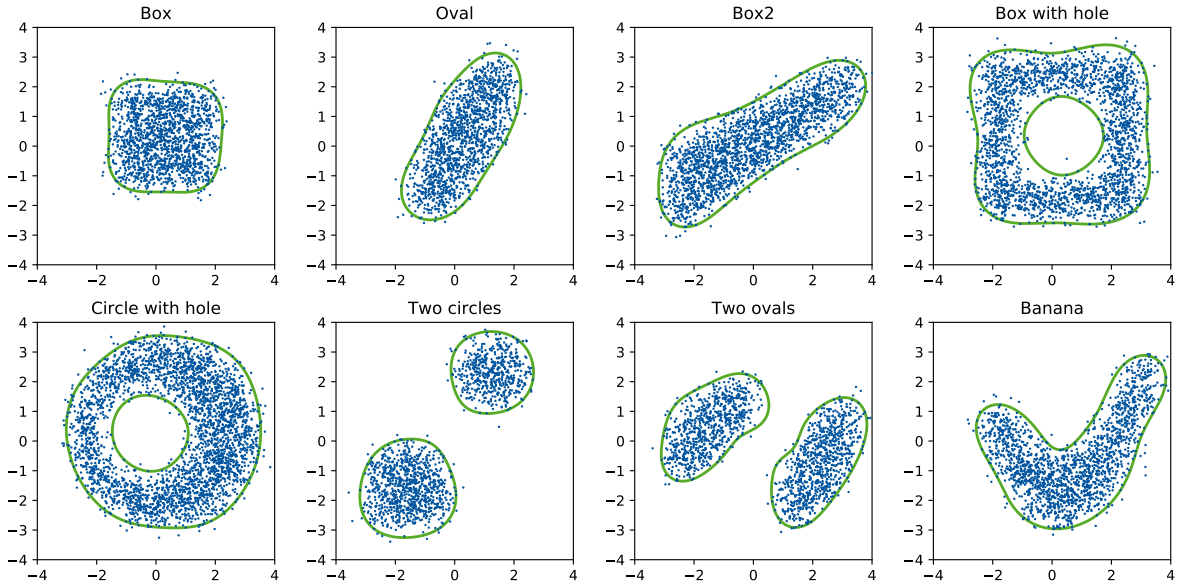
(g) “Two ovals” case study



(h) “Banana” case study



(a) Convex hulls of the eight illustrative data sets



(b) One-class SVM boundaries of the eight illustrative data sets

Figure 5: Comparison of the convex hull and the boundaries learned by the one-class SVMs

Case study	Convex hull x^*	Convex hull $f^*_{\mathbf{ANN}}$	Convex hull Δ	One-class SVM x^*	One-class SVM $f^*_{\mathbf{ANN}}$	One-class SVM Δ	Reference x^*
Banana	$(0.1, 2.0)$	-5.2	8.52	$(0.3, -1.6)$	-4.3	0.11	$(0.2, -1.6)$
Two circles	$(-1.5, 1.5)$	-5.1	3.65	$(1.0, 3.7)$	-4.8	0.01	$(1.3, 3.7)$
Box	$(0.2, -1.9)$	-5.4	2.14	$(0.3, -1.5)$	-4.5	0.07	$(0.2, -1.5)$
Box w/ hole	$(0.2, 3.6)$	-6.4	1.65	$(0.5, 3.2)$	-4.8	0.70	$(0.2, -1.6)$
Circle w/ hole	$(-1.5, 3.5)$	-5.0	0.41	$(0.1, 3.6)$	-4.6	0.02	$(0.2, 3.6)$
Two ovals	$(-1.3, 0.4)$	-3.5	0.13	$(-1.3, 0.4)$	-3.5	0.13	$(-1.3, 0.4)$
Oval	$(1.3, 3.5)$	-4.6	0.13	$(0.2, -1.6)$	-4.3	0.14	$(0.2, -1.6)$
Box2	$(0.1, -1.7)$	-4.2	0.05	$(2.8, 2.9)$	-3.8	0.05	$(2.9, 2.9)$

Table 2: The table compares the global optimal solutions with convex hull and one-class SVMs (SVMs) as constraints. The optimal solution point is given by x^* and the objective function value is given by f^*_{ANN}. Also, we provide the error of the data-driven model at the optimal solution $\Delta = |f^*_{\text{ANN}} - f_{\text{Peaks}}(x^*)|$. The reference solution point shows the optimal solution when considering the underlying function f_{Peaks} as the objective with the one-class SVM constraint.

yields an optimal solution of -4.3 with an absolute error of 0.11. Also, the solution point found by the ANN model with the one-class SVM constraint is close to the reference solution where the learned peaks function is optimized subject to the SVM constraint. Similarly, the one-class SVM leads to more reliable results in the data sets "two circles", "box w/ holes", and "circle w/ holes". Interestingly, the convex hull approach also leads to a substantial prediction error in the "box" case study while the one-class SVM models the validity domain accurately. This highlights the risk of using the convex hull approach in the presence of noise.

In Table 3, we provide the CPU times for optimization with one-class SVMs embedded. Using the FS formulation, BARON and MAiNGO perform similarly and solve most problems in the a few hundred CPU seconds. The RS formulation outperforms the FS formulation on all problem instances. In BARON, the speedup factor between the RS and the FS formulation ranges from 5 to over 14. In comparison, the the speedup factor between the RS and FS in MAiNGO ranges between 583 to over 3,226. This is in agreement with our previous studies where the McCormick relaxations in the RS lead to smaller subproblems compared to the auxiliary variable method (see Section 2.3).

In Table 4, we compare the computational performance for optimization with the convex hull embedded. The CPU times with the convex hull are lower compared to the ones with one-class SVMs. MAiNGO is substantially faster than BARON when formulating the problem in the FS. On average, BARON requires about 83 seconds to solve the problem in the FS while MAiNGO requires only 3 seconds. The RS formulation again outperforms the FS formulation for all problems. However, in this case, the speedup factors are in the same order of magnitude for BARON and MAiNGO ranging between 13 and 54. It should be noted that the RS and the FS formulation of the convex hull constrains are identical. Therefore, the difference is only due to the formulation of the data-driven model in the objective function, i.e., the ANN (c.f. (Schweidtmann and Mitsos, 2019)).

Case study	# Sup. vec.	BARON FS	BARON RS	BARON sp-f	MAiNGO FS	MAiNGO RS	MAiNGO sp-f
Oval	48	158.2 s	35.0 s	**5**	197.2 s	0.20 s	**986**
Two circles	50	489.5 s	36.7 s	**13**	247.0 s	0.25 s	**988**
Box	52	346.6 s	25.8 s	**13**	139.9 s	0.24 s	**583**
Two ovals	58	341.3 s	38.6 s	**9**	433.2 s	0.22 s	**1,969**
Banana	62	1,000.0 s	70.6 s	**>14**	416.7 s	0.19 s	**2,193**
Box2	67	497.1 s	22.9 s	**22**	1,000.0 s	0.31 s	**>3,226**
Box w/ hole	81	1,000.0 s	73.1 s	**>14**	656.5 s	0.36 s	**1,823**
Circle w/ hole	103	1,000.0 s	75.8 s	**>13**	1,000.0 s	0.75 s	**>1,333**

Table 3: CPU times for optimization of the eight case studies with the one-class SVM as a constraint. The table compares the FS and RS formulations for the BARON and MAiNGO solvers. Here, the data-driven model (i.e., the ANN) and the one-class SVM are formulated in the RS and FS. The speed up factor (sp-f) is given as the ratio between the FS and the RS solution times. Also, the number of support vectors (# Sup. vec.) is shown as a measure for the problem complexity.

4 Engineering application

We consider a sulfur recovery unit as a relevant engineering case study for our work because a large data set of industry operating data is available online for this process. The efficient recovery of sulfur in petroleum refineries from tail gas is important for environmental reasons. The sulfur recovery unit process is illustrated in Figure 6. The process has two acid gases as inputs: the MAE gas stream is rich in hydrogen sulphide (H_2S) and comes from the gas washing plants. The SWS gas stream is rich in H_2S and ammonia and comes from a sour water stripping plant. In the sulfur recovery unit, the acid gases are burnt via partial reaction with air in a two-chamber reaction furnace. Then, the combustion product is further treated in two subsequent catalytic reactors resulting in a tail gas stream that contains residuals of H_2S and sulfur dioxide (SO_2). A detailed process description can be found in the literature (Fortuna et al., 2007).

A key issue of the sulfur recovery unit is the control of the secondary air flow to ensure optimal conditions for the total removal of the sulfur compounds in the catalytic converters. Previous works have investigated soft sensors for the tail gas concentrations of hydrogen sulphide (H_2S) and sulfur dioxide (SO_2) using ANNs and implemented those in industry for monitoring (Quek et al., 2000; Fortuna et al., 2003, 2007). In this case study, we solve an open-loop control problem to find the optimal secondary air flow rate. The objective is to minimize $|c_{H_2S} - 2 \cdot c_{SO_2}|$ such that the two reactants are in stoichiometric proportion. Similar to the previous literature by Fortuna et al. (2003, 2007), we also train

Case study	# Sup. vec.	BARON FS	BARON RS	BARON sp-f	MAiNGO FS	MAiNGO RS	MAiNGO sp-f
Oval	48	74.9 s	2.8 s	**27**	2.7 s	0.05 s	**54**
Two circles	50	46.4 s	3.5 s	**13**	2.4 s	0.09 s	**27**
Box	52	85.8 s	1.5 s	**57**	1.7 s	0.06 s	**28**
Two ovals	58	103.9 s	3.1 s	**34**	3.8 s	0.08 s	**48**
Banana	62	136.4 s	5.1 s	**27**	3.0 s	0.09 s	**33**
Box2	67	32.3 s	2.3 s	**14**	2.7 s	0.11 s	**25**
Box w/ hole	81	51.3 s	3.5 s	**15**	2.4 s	0.08 s	**30**
Circle w/ hole	103	130.2 s	4.1 s	**32**	3.1 s	0.11 s	**28**

Table 4: CPU times for optimization of the eight case studies with the convex hull as a constraint. The table compares the FS and RS formulations for the BARON and MAiNGO solvers. The speed up factor (sp-f) is given as the ratio between the FS and the RS solution times. Also, the number of support vectors (# Sup. vec.) is shown as a measure for the problem complexity.

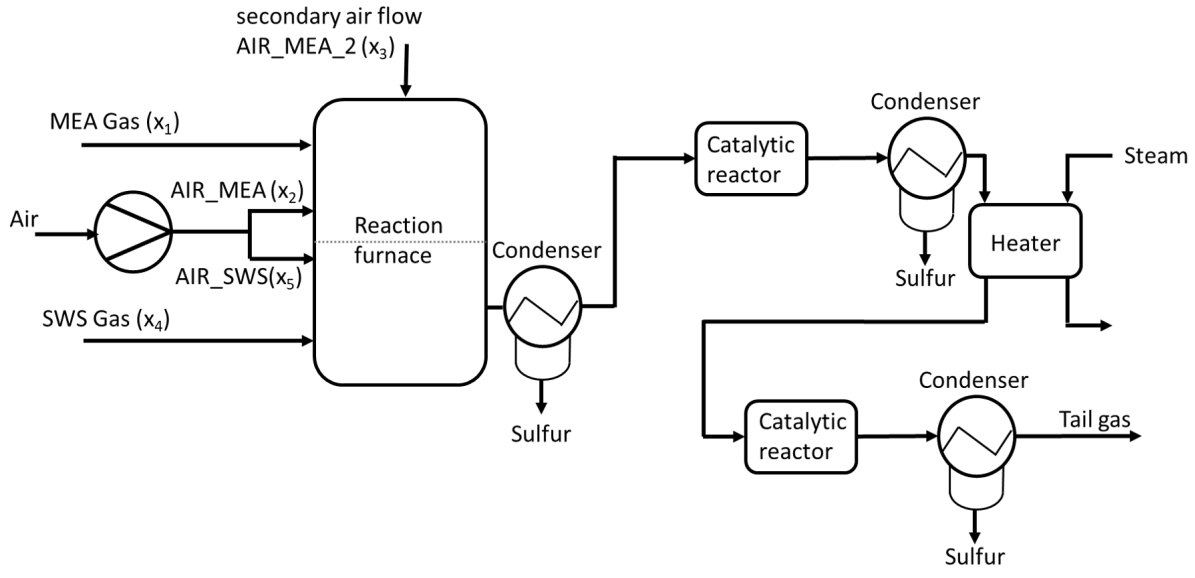


Figure 6: Flowchart of the sulfur recovery unit process

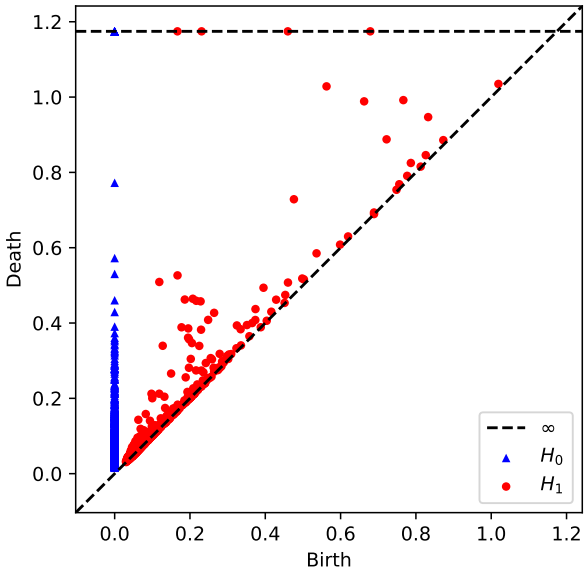


Figure 7: Persistent diagram for of the training data of the engineering case study presented in Section 4

two ANNs to predict the concentrations:

$$\begin{aligned} c_{H_2S,k} &= f_{ANN,H_2S}\left(x_{1,k}, x_{1,k-5}, x_{1,k-7}, x_{1,k-9}, ...x_{5,k}, x_{5,k-5}, x_{5,k-7}, x_{5,k-9}\right), \\ c_{SO_2,k} &= f_{ANN,SO_2}\left(x_{1,k}, x_{1,k-5}, x_{1,k-7}, x_{1,k-9}, ...x_{5,k}, x_{5,k-5}, x_{5,k-7}, x_{5,k-9}\right), \end{aligned}$$

where $x_{1,k}$ is the gas flow in the MEA zone, $x_{2,k}$ is the air flow in the MEA zone, $x_{3,k}$ is the secondary air flow in the MEA zone, $x_{4,k}$ is the air flow in the SWS zone, $x_{5,k}$ is the gas flow in the SWS zone at time step k. The SO_2 ANN has one hidden layer with eight neurons and the H_2S ANN has two hidden layers with eight neurons each. The data-driven models are trained on (scaled) industrial data collected at a plant located in Priolo, Italy available at `https://www.openml.org/d/23515`. The data set includes a time series with approximately 10,000 data samples and we use the first 90% of the data for training. The control variable of the NMPC is the secondary air flow $x_{3,k}$ while the other inputs are observable parameters. As the control is critical for process safety, the validity limits of the data-driven model should be considered.

In order to analyze the topology of the 20-dimensional input training data set of ANNs, we perform persistent homology. Due to the large number of data points, the exact computation of the persistent diagram is expensive. We apply approximate sparse filtration instead (Cavanna et al., 2015). The persistent diagram for this case study is shown in Figure 7. The diagram shows that there exist a number of holes in the data set that persist over a long time span. Also, a separate cluster can be observed in the data. This motivates the use of one-class SVM to obey validity limits of data-driven models.

As we have no physical model of the process available, the closed-loop performance of the controller is not studied in this example. For illustration, we perform one step of an open-loop controller for the secondary air flow $x_{3,k}$. We select a random operating

	k	$k-5$	$k-7$	$k-9$
x_1	0.627	0.6215	0.623	0.622
x_2	0.770	0.769	0.754	0.769
x_3	$x_{3,k}$	0.174	0.192	0.198
x_4	0.376	0.399	0.415	0.410
x_5	0.513	0.512	0.511	0.504

Table 5: Operating point of the sulfur recovery unit that is considered for the NMPC optimization step.

point from the historic plant data (Table 5) and let the solver determine the optimal control action. The problem is solved to global optimality within 0.33 CPU seconds and identifies a control action $x_{3,k} = 0.266$ that results in the desired stoichiometric composition, i.e., $|c_{H_2S} - 2 \cdot c_{SO_2}| = 1.3 \cdot 10^{-5}$. This engineering case study also demonstrates the potential of the proposed method for NMPC. Note that deterministic global NMPC can become computationally expensive for long control horizons and higher dimensional control vectors (Chachuat et al., 2006; Doncevic et al., 2020; Kappatou et al., 2020).

5 Conclusion

Safety concerns and extrapolation issues often impede industrial applications of machine learning models. We present a three-step approach to obey the validity limits of data-driven models. First, we perform a data topology analysis using persistent homology. Second, we model the validity domain of the data-driven model using either the convex hull or a one-class SVM. Third, we perform deterministic global optimization with the validity domain model as a constraint.

All used and developed methods are available open-source. Also, we currently develop a Python interface for our solver MAiNGO. Thus, all methods can be applied and further developed in academia and industry for free.

Our method has the potential to enhance safety, trust, and reliability of machine learning approaches. Moreover, we demonstrate that persistent homology is a valuable method for understanding the topology of data in high dimensional spaces. Besides industry applications, promising future work also includes the application to optimization problems occurring in molecular design where molecules are parameterized through graph neural networks (Schweidtmann et al., 2020c) or autoencoders (Jin et al., 2018). Also, time-dependent design space descriptions are desired in pharmaceutics (von Stosch et al., 2020). The proposed method can also be extended by considering and comparing other one-class classification methods.

Acknowledgements

We are grateful to Benoît Chachuat for providing MC++ under Eclipse Public License. We also thank Dominik Bongartz and Jaromił Najman for their work on MAiNGO.

Declarations

Code availability

The method is ready-to-use and available open-source as part of our "MeLOn - **M**achin**e** **L**earning Models for **O**ptimizatio**n**" toolbox under the Eclipse public license (`https://git.rwth-aachen.de/avt.svt/public/MeLOn`).

Availability of data and material

The (scaled) industrial data used in the engineering case study are available at `https://www.openml.org/d/23515`.

Authors' contributions

AMS and JMW designed the research concept. AMS wrote the manuscript. JMW run the persistent homology analyzed the persistent plots, and wrote the corresponding method and result sections. AMS, CW, and LN run the optimization. AMS, LN, and CW implemented the model in the MeLOn tool. AM is principal investigator who guided the effort and edited the manuscript.

Conflicts of interest

The authors declare that they have no conflict of interest.

Funding

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - Cluster of Excellence 2186 "The Fuel Science Center".

References

Asprion N (2020) Modeling, simulation, and optimization 4.0 for a distillation column. Chemie Ingenieur Technik 92(7):879–889

Asprion N, Böttcher R, Pack R, Stavrou ME, Höller J, Schwientek J, Bortz M (2019) Gray-box modeling for the optimization of chemical processes. Chemie Ingenieur Technik 91(3):305–313

Binchi J, Merelli E, Rucco M, Petri G, Vaccarino F (2014) jholes: A tool for understanding biological complex networks via clique weight rank persistent homology. Electron Notes Theor Comput Sci 306:5–18

Bongartz D (2020) Deterministic global flowsheet optimization for the design of energy conversion processes. PhD thesis, RWTH Aachen University

Bongartz D, Mitsos A (2017) Deterministic global optimization of process flowsheets in a reduced space using McCormick relaxations. Journal of Global Optimization 20(9):419

Bongartz D, Najman J, Sass S, Mitsos A (2018) MAiNGO: McCormick-based Algorithm for mixed integer Nonlinear Global Optimization. Technical report, Process Systems Engineering (AVTSVT), RWTH Aachen University, https://avtrwth-aachende/global/show_documentasp?id=aaaaaaaaabclahw

Cavanna NJ, Jahanseir M, Sheehy DR (2015) A geometric perspective on sparse filtrations. arXiv preprint arXiv:150603797

Chachuat B, Singer AB, Barton PI (2006) Global methods for dynamic optimization and mixed-integer dynamic optimization. Industrial & Engineering Chemistry Research 45(25):8373–8392

Chachuat B, Houska B, Paulen R, Peric N, Rajyaguru J, Villanueva ME (2015) Set-theoretic approaches in analysis, estimation and control of nonlinear systems. IFAC-PapersOnLine 48(8):981–995, DOI 10.1016/j.ifacol.2015.09.097

Chambers EW, Letscher D (2018) Persistent homology over directed acyclic graphs. In: Research in Computational Topology, Springer, pp 11–32

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: A survey. ACM computing surveys (CSUR) 41(3):1–58

Charnes A, Cooper WW (1959) Chance-constrained programming. Management science 6(1):73–79

Chazal F, Michel B (2017) An introduction to topological data analysis: fundamental and practical aspects for data scientists. arXiv preprint arXiv:171004019

Chen Q, Paulavičius R, Adjiman CS, García-Muñoz S (2018) An optimization framework to combine operable space maximization with design of experiments. AIChE Journal 64(11):3944–3957

Chollet F, et al. (2015) Keras. Accessed May 2020 https://keras.io

Chung MK, Hanson JL, Ye J, Davidson RJ, Pollak SD (2015) Persistent homology in sparse regression and its application to brain morphometry. IEEE Transactions on Medical Imaging 34(9):1928–1939

Cortes C, Vapnik V (1995) Support-vector networks. Machine learning 20(3):273–297

Courrieu P (1994) Three algorithms for estimating the domain of validity of feedforward neural networks. Neural Networks 7(1):169–174

Ding X, Li Y, Belatreche A, Maguire LP (2014) An experimental evaluation of novelty detection methods. Neurocomputing 135:313–327

Doncevic DT, Schweidtmann AM, Vaupel Y, Schäfer P, Caspari A, Mitsos A (2020) Deterministic global nonlinear model predictive control with recurrent neural networks embedded. In Press: IFAC Conference Proceedings

Dreiseitl S, Osl M, Scheibböck C, Binder M (2010) Outlier detection with one-class svms: an application to melanoma prognosis. In: AMIA Annual Symposium Proceedings, American Medical Informatics Association, vol 2010, p 172

Epperly TGW, Pistikopoulos EN (1997) A reduced space branch and bound algorithm for global optimization. Journal of Global Optimization 11(3):287–311

Evangelista PF, Embrechts MJ, Szymanski BK (2007) Some properties of the Gaussian kernel for one class learning. In: International Conference on Artificial Neural Networks, Springer, pp 269–278

Fortuna L, Rizzo A, Sinatra M, Xibilia M (2003) Soft analyzers for a sulfur recovery unit. Control Engineering Practice 11(12):1491–1500

Fortuna L, Graziani S, Rizzo A, Xibilia MG (2007) Soft sensors for monitoring and control of industrial processes. Springer Science & Business Media

Glassey J, Von Stosch M (2018) Hybrid Modeling in Process Industries. CRC Press

Hart WE, Laird CD, Watson JP, Woodruff DL, Hackebeil GA, Nicholson BL, Siirola JD (2017) Pyomo-optimization modeling in python, vol 67. Springer

Hiraoka Y, Nakamura T, Hirata A, Escolar EG, Matsue K, Nishiura Y (2016) Hierarchical structures of amorphous solids characterized by persistent homology. Proceedings of the National Academy of Sciences 113(26):7035–7040

Hüllen G, Zhai J, Kim SH, Sinha A, Realff MJ, Boukouvala F (2019) Managing uncertainty in data-driven simulation-based optimization. Computers & Chemical Engineering p 106519, DOI 10.1016/j.compchemeng.2019.106519

Jin W, Barzilay R, Jaakkola T (2018) Junction tree variational autoencoder for molecular graph generation. arXiv preprint arXiv:180204364

Kahrs O, Marquardt W (2007) The validity domain of hybrid models and its application in process optimization. Chemical Engineering and Processing: Process Intensification 46(11):1054–1066

Kahrs O, Marquardt W (2008) Incremental identification of hybrid process models. Computers & Chemical Engineering 32(4-5):694–705

Kappatou CD, Bongartz D, Najman J, Sass S, Mitsos A (2020) Global dynamic optimization with hammerstein-wiener models embedded. Pre-Print `http://www.optimization-online.org/DB_HTML/2020/09/8018.html`

Khan SS, Madden MG (2009) A survey of recent trends in one class classification. In: Irish conference on artificial intelligence and cognitive science, Springer, pp 188–197

Khan SS, Madden MG (2014) One-class classification: taxonomy of study and review of techniques. The Knowledge Engineering Review 29(3):345–374

Kimura Y, Imai K (2017) Quantification of lss using the persistent homology in the sdss fields. Advances in Space Research 60(3):722–736

Knudde N, Couckuyt I, Shintani K, Dhaene T (2019) Active learning for feasible region discovery. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, pp 567–572

Kumar JN, Li Q, Tang KY, Buonassisi T, Gonzalez-Oyarce AL, Ye J (2019) Machine learning enables polymer cloud-point engineering via inverse design. npj Computational Materials 5(1):1–6

Larson BJ, Mattson CA (2012) Design space exploration for quantifying a system model's feasible domain. Journal of Mechanical Design 134(4)

Letscher H, Edelsbrunner D, Zomorodian A (2002) Topological persistence and simplification. Discrete & Computational Geometry 28:511–533

Malak RJ, Paredis CJ (2010) Using support vector machines to formalize the valid input domain of predictive models in systems design problems. Journal of Mechanical Design 132(10)

McBride K, Sundmacher K (2019) Overview of surrogate modeling in chemical process engineering. Chemie Ingenieur Technik 91(3):228–239, DOI 10.1002/cite.201800091

Mistry M, Letsios D, Krennrich G, Lee RM, Misener R (2018) Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded. arXiv preprint arXiv:180300952

Mitsos A, Chachuat B, Barton PI (2009) McCormick-based relaxations of algorithms. SIAM Journal on Optimization 20(2):573–601, DOI 10.1137/080717341

Mogk G, Mrziglod T, Schuppert A (2002) Application of hybrid models in chemical industry. In: Computer Aided Chemical Engineering, vol 10, Elsevier, pp 931–936

Otter N, Porter MA, Tillmann U, Grindrod P, Harrington HA (2017) A roadmap for the computation of persistent homology. EPJ Data Science 6(1):17

Papadopoulos G, Edwards PJ, Murray AF (2001) Confidence estimation methods for neural networks: A practical comparison. IEEE transactions on neural networks 12(6):1278–1287

Patania A, Vaccarino F, Petri G (2017) Topological analysis of data. EPJ Data Science 6:1–6

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. (2011) Scikit-learn: Machine learning in python. the Journal of machine Learning research 12:2825–2830

Pimentel MA, Clifton DA, Clifton L, Tarassenko L (2014) A review of novelty detection. Signal Processing 99:215–249

Pinto J, de Azevedo CR, Oliveira R, von Stosch M (2019) A bootstrap-aggregated hybrid semi-parametric modeling framework for bioprocess development. Bioprocess and biosystems engineering 42(11):1853–1865

Quaglio M, Fraga ES, Cao E, Gavriilidis A, Galvanin F (2018) A model-based data mining approach for determining the domain of validity of approximated models. Chemometrics and Intelligent Laboratory Systems 172:58–67

Quek C, Balasubramanian R, Rangaiah G (2000) Consider using soft analyzers to improve SRU control. Hydrocarbon processing 79(1):101–106

Rall D, Menne D, Schweidtmann AM, Kamp J, von Kolzenberg L, Mitsos A, Wessling M (2019) Rational design of ion separation membranes. Journal of Membrane Science 569:209–219

Roach E, Parker RR, Malak Jr RJ (2011) An improved support vector domain description method for modeling valid search domains in engineering design problems. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol 54822, pp 741–751

Saadatfar M, Takeuchi H, Robins V, Francois N, Hiraoka Y (2017) Pore configuration landscape of granular crystallization. Nature communications 8(1):1–11

Schölkopf B (2001) The kernel trick for distances. In: Advances in neural information processing systems, pp 301–307

Schölkopf B, Williamson RC, Smola AJ, Shawe-Taylor J, Platt JC (2000) Support vector method for novelty detection. In: Advances in neural information processing systems, pp 582–588

Schweidtmann AM, Mitsos A (2019) Deterministic global optimization with artificial neural networks embedded. Journal of Optimization Theory and Applications 180(3):925–948

Schweidtmann AM, Bongartz D, Grothe D, Kerkenhoff T, Lin X, Najman J, Mitsos A (2020a) Global optimization of Gaussian processes. arXiv preprint arXiv:200510902

Schweidtmann AM, Netze L, Mitsos A (2020b) Melon: Machine learning models for optimization. `https://git.rwth-aachen.de/avt.svt/public/MeLOn/`

Schweidtmann AM, Rittig JG, König A, Grohe M, Mitsos A, Dahmen M (2020c) Graph neural networks for prediction of fuel ignition quality. ChemRxiv preprint ChemRxiv:12280325

Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N (2016) Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE 104(1):148–175, DOI 10.1109/JPROC.2015.2494218

Smith AD, Dlotko P, Zavala VM (2020) Topological data analysis: Concepts, computation, and applications in chemical engineering. arXiv preprint arXiv:200603173

Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Statistics and computing 14(3):199–222

von Stosch M, Schenkendorf R, Geldhof G, Varsakelis C, Mariti M, Dessoy S, Vandercammen A, Pysik A, Sanders M (2020) Working within the design space: Do our static process characterization methods suffice? Pharmaceutics 12(6):562

Tawarmalani M, Sahinidis NV (2005) A polyhedral branch-and-cut approach to global optimization. Mathematical Programming 103(2):225–249, DOI 10.1007/s10107-005-0581-8

Tax DM, Duin RP (1999) Data domain description using support vectors. In: ESANN, vol 99, pp 251–256

Tax DMJ (2001) One-class classification: Concept learning in the absence of counter-examples. PhD thesis, Delft University of Technology

Tralie C, Saul N, Bar-On R (2018) Ripser.py: A lean persistent homology library for python. The Journal of Open Source Software 3(29):925

Venkatasubramanian V (2019) The promise of artificial intelligence in chemical engineering: Is it here, finally. AIChE J 65(2):466–78

Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, et al. (2020) Scipy 1.0: fundamental algorithms for scientific computing in python. Nature methods 17(3):261–272

Von Stosch M, Oliveira R, Peres J, de Azevedo SF (2014) Hybrid semi-parametric modeling in process systems engineering: Past, present and future. Computers & Chemical Engineering 60:86–101, DOI 10.1016/j.compchemeng.2013.08.008

Wasserman L (2018) Topological data analysis. Annual Review of Statistics and Its Application 5:501–532

Wilhelm M, Stuber M (2020) Eago. jl: easy advanced global optimization in julia. Optimization Methods and Software pp 1–26

Xia K (2018) Persistent homology analysis of ion aggregations and hydrogen-bonding networks. Physical Chemistry Chemical Physics 20(19):13448–13460

Xia K, Anand DV, Shikhar S, Mu Y (2019) Persistent homology analysis of osmolyte molecular aggregation and their hydrogen-bonding networks. Physical Chemistry Chemical Physics 21(37):21038–21048

Xiao Y, Wang H, Xu W (2014a) Parameter selection of Gaussian kernel for one-class svm. IEEE transactions on cybernetics 45(5):941–953

Xiao Y, Wang H, Zhang L, Xu W (2014b) Two methods of selecting Gaussian kernel parameters for one-class svm and their application to fault detection. Knowledge-Based Systems 59:75–84

Zhang Q, Grossmann IE, Sundaramoorthy A, Pinto JM (2016) Data-driven construction of convex region surrogate models. Optimization and Engineering 17(2):289–332, DOI 10.1007/s11081-015-9288-8

Zomorodian A, Carlsson G (2005) Computing persistent homology. Discrete & Computational Geometry 33(2):249–274