One-Class Classification Framework Based on Shrinkage Methods

Patric Nader, Paul Honeine, Pierre Beauseroy

One-Class Classification Framework Based on Shrinkage Methods

Patric Nader[1], Paul Honeine, and Pierre Beauseroy

Université de Technologie de Troyes,
12, rue Marie Curie BP 2060 - 10010 Troyes cedex, France
{*patric.nader, paul.honeine, pierre.beauseroy*}*@utt.fr*

Abstract

Statistical machine learning, such as kernel methods, have been widely used in the past decades to discover hidden regularities in data. In particular, one-class classification algorithms gained a lot of interest in a large number of applications where the only available data designate a unique class. In this paper, we propose a framework for one-class classification problems, by investigating the hypersphere enclosing the samples in the Reproducing Kernel Hilbert Space (RKHS). The center of this hypersphere is approximated using a sparse solution, by selecting an appropriate set of relevant samples. To this end, we investigate well-known shrinkage and selection methods for linear regression, namely LARS, LASSO, and Elastic Net. We revisit these selection methods and adapt their algorithms for estimating the sparse center of the one-class problem in the RKHS. The proposed framework is also extended to include the Mahalanobis distance in the RKHS. We compare our algorithms to well-known one-class methods, and the tests are conducted on simulated and real datasets.

Keywords: Elastic net, kernel methods, LARS, LASSO, machine learning, one-class classification, shrinkage methods, sparse approximation.

[1]Corresponding author, Tel : (+33)(0)3 25 71 76 47

1. Introduction

. Over the last decades, kernel methods have become very popular in the machine learning and data mining fields for estimation and learning problems [1]. Machine learning techniques with kernel methods provide a powerful way for detecting hidden regularities and patterns in large volumes of data [2]. They have been applied in different fields for classification and regression problems, such as autonomous robotics [3], biomedical signal processing [4], and wireless sensor networks [5][6]. Machine learning techniques use positive definite kernels to map the data into a reproducing kernel Hilbert space, where linear algorithms are applied on the mapped data in that space in order to detect nonlinear relations existing in the input space [7]. In practice, only the pairwise inner product between the mapped data is needed [8]. This inner product is computed directly from the input data using a kernel function, without any explicit knowledge on the mapping function.

. In several applications as in industrial systems, the only available data designate the normal functioning modes of the studied system, while the data related to the malfunctioning modes and to critical states are difficult to obtain. When it comes to industrial processes with detecting machine faults or intrusions, the number of the failure modes or the increasing number of new generated attacks may not be bounded in general [9]. This is the reason why researchers have been developing in the last few years one-class classification algorithms for such one-class problems, where the available dataset refers only to a single class. One-class classifiers learn the normal behavior modes of the studied system, and develop decision functions in a way to accept as many normal samples as possible, and to reject the outliers (any sample not belonging to the same distribution of the data) [10]. New samples are then classified as normal ones or outliers according to the decision function of the classifiers. One-class classification algorithms have been applied in many fields, including face recognition applications [11], mobile masquerades detection [12], seizure analysis from EEG signals [13], and recently for intrusion detection in industrial systems [14].

. Several formulations were proposed in the literature for one-class classification problems. Researchers have been facing many challenges to elaborate relevant one-class algorithms, namely in reducing the computational cost, and in improving the detection accuracy while maintaining a good decision rule that avoids both overfitting and underfitting the data. Schölkopf et al. proposed in [15] the one-class Support Vector Machines (one-class SVM), in which the mapped data are separated from the origin with maximum margin using a hyperplane. This approach requires to solve a constrained quadratic programming problem, thus it is greedy in terms of computational cost. Tax et al. introduced in [16] the Support Vector Data Description (SVDD) which estimates the hypersphere with minimum radius enclosing most of the training data. The resulting optimization problem is essentially similar to the one-class SVM, while they are equivalent when unit-norm kernels are used, such as the Gaussian kernel. Neither one-class SVM nor SVDD take into consideration the heterogenous nature of the mapped data, namely the scale variation in each direction. An attempt to overcome this drawback is proposed in [17] with a kernel whitening normalization by rescaling the data to have equal variance in each direction of the RKHS. The resulting optimization problem incorporated an eigen decomposition problem as well as the conventional constrained quadratic programming problem. Azami et al. proposed in [18] the use of the ℓ_0 pseudo-norm in a SVDD formulation, and provided an iterative procedure by solving a constrained quadratic programming problem at each iteration, which is very expensive in terms of computational cost. A fast one-class approach was introduced in [19] to overcome the drawbacks of existing algorithms. The slab Support Vector Machine (slab SVM) is described in [20][21][22], and unlike the original SVM, its optimization problem aims at finding a slab (two parallel hyperplanes) that encloses the samples which are maximally separated from the origin in the feature space. Similarly to the SVM, this approach requires to solve a constrained quadratic programming problem. All these methods use the Euclidean distance in the decision function of the classifier, which leads to high sensitivity towards outliers. Other approaches that use the covariance information to learn the kernel in one-class

SVM were proposed in [23][24]. These approaches require the optimization of a second order cone programming (SOCP) problem, and their complexity is cubic with the size of the training dataset. The "Robust SVM" algorithm was introduced in [25] for binary and multiclass classification problems, and it was modified in [26] for anomaly detection in one-class classification problems. This algorithm aims at reducing the influence of the existing outliers on the decision rule of the standard one-class SVM classifier, by introducing a new slack variable related to the distance between each sample and the center of the data in the feature space. This approach is less sensitive than the standard SVM towards outliers, yet it still requires to solve a constrained quadratic programming problem. Kernel Principal Component Analysis (KPCA) was introduced in [27] for several applications, and Hoffman used in [28] the KPCA for one-class classification problems by projecting the data into the subspace spanned by the most relevant eigenvectors. The reconstruction error used as a novelty measure has a relatively low computational cost, yet this approach loses the sparsity of SVM and SVDD.

. In this paper, we propose a sparse framework for one-class classification problems. The proposed classifier is defined by a hypersphere enclosing the samples in the Reproducing Kernel Hilbert Space (RKHS), determined by its center and its radius which discriminates new samples as normals or outliers. We approximate the center of the hypersphere by the empirical center of the data in the RKHS, where this sparse center depends only on a small fraction of the data. Since a wise selection of these samples is crucial in such sparse approaches, we propose to investigate well-known shrinkage methods [29], namely Least Angle Regression [30], Least Absolute Shrinkage and Selection Operator [31][32], and Elastic Net [33][34]. These shrinkage methods have been usually used in regression problems in the input space. We revisit these methods and adapt their algorithms to the estimation of the one-class center in the RKHS. Moreover, we propose a modified version that takes advantage of the KPCA approach, by replacing the Euclidean distance in the decision function with the Maha-

lanobis distance [35]. This allows to scale the data in the feature space, thus overcoming the heteroscedastic nature of the data. We also provide some theoretical results related to the proposed algorithms, namely on the error of the first kind, which represents the samples that are misclassified and considered as outliers. The tests are conducted on simulated datasets, and on three real datasets from the Mississippi State University SCADA Laboratory [36][37] and from the University of California machine learning repository [38].

The remainder of this paper is organized as follows. Section 2 provides an overview on the common one-class classification methods in the literature. Section 3 describes the proposed one-class framework and the adapted shrinkage methods. An extension including the Mahalanobis distance is detailed in Section 4. Section 5 provides some theoretical results, namely the error of the first kind. Section 6 discusses the results on simulated and real datasets, and Section 7 provides conclusion and future works.

2. One-class classification

. Consider a training dataset $\boldsymbol{x}_i$, $i = 1, \ldots, n$, in a d-dimensional space $\mathcal{X}$. Kernel methods map the data from the input space $\mathcal{X}$ into a higher dimensional feature space $\mathcal{H}$ via a mapping function $\phi(\boldsymbol{x}_i) = k(\boldsymbol{x}_i, \cdot)$. This allows to describe nonlinear relations in the input space, by converting them into linear ones in the feature space. In practice, only the pairwise inner product between the mapped data is needed, thus without any explicit knowledge of the mapping function ϕ. This inner product is computed directly from the input data using a kernel function. Let $\boldsymbol{K}$ be the $n \times n$ kernel matrix with entries $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathcal{X}$. The kernel matrix plays an important role in the learning algorithms. Next, we detail the common one-class classification methods in the literature.

2.1. Support Vector Data Description

. Support Vector Data Description (SVDD) estimates the hypersphere with minimum radius that encompasses all the data $\phi(\boldsymbol{x}_i)$ in the feature space $\mathcal{H}$.

The hypersphere is characterized by its center $\boldsymbol{a}$ and its radius $R > 0$, and the SVDD algorithm minimizes its volume by minimizing R^2. The presence of outliers in the training set is allowed by introducing the slack variables $\xi_i \geq 0$ for each training sample, in order to penalize the excluded ones. This boils down to the following constrained optimization problem:

$$\min_{\boldsymbol{a},R,\xi_i} R^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i \tag{1}$$

subject to $\|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|_{\mathcal{H}}^2 \leq R^2 + \xi_i$ and $\xi_i \geq 0 \quad \forall i = 1, ..., n$. The predefined parameter $\nu \in (0, 1)$ regulates the trade-off between the volume of the hypersphere and the number of outliers. Its value represents an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors. SVDD maximizes the following objective functional with respect to the Lagrangian multipliers α_i:

$$L = \sum_{i=1}^{n}\alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n}\alpha_i\alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2}$$

subject to $\sum_{i=1}^{n}\alpha_i = 1$ and $0 \leq \alpha_i \leq \frac{1}{\nu n}$. This a constrained quadratic programming problem, whose solution is found using any off-the-shelf optimization technique. For instance, one can use the MATLAB function quadprog.

The radius of the optimal hypersphere is obtained with the distance in the feature space from the center $\boldsymbol{a}$ to any sample $\phi(\boldsymbol{x}_k)$ on the boundary as follows::

$$R^2 = k(\boldsymbol{x}_k, \boldsymbol{x}_k) - 2\sum_{i=1}^{n}\alpha_i k(\boldsymbol{x}_k, \boldsymbol{x}_i) + \sum_{i,j=1}^{n}\alpha_i\alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

In order to evaluate a new sample $\boldsymbol{z}$, the decision rule is obtained by evaluating the distance between $\boldsymbol{a}$ and $\phi(\boldsymbol{z})$ in the feature space. The new sample $\boldsymbol{z}$ is considered as a normal sample if the calculated distance is smaller than the radius, namely: $\|\phi(\boldsymbol{z}) - \boldsymbol{a}\|_{\mathcal{H}}^2 \leq R^2$.

2.2. Slab Support Vector Machine

The slab Support Vector Machine (slab SVM) is a modified version of the standard SVM algorithm, and it aims at finding a region bounded by two parallel hyperplanes, called a slab, that encloses the samples in the feature space and

maximally separated from the origin. The constrained optimization problem of the slab SVM is given as follows:

$$\min_{\boldsymbol{w},\rho,\xi_i} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \tag{3}$$

subject to $0 \leq \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle - \rho + \xi_i \leq \delta$, where $\boldsymbol{w}$, ρ and δ are the parameters of the slab, ξ_i denote the slack variables and ν the upper bound on the fraction of outliers. The following objective functional needs to be minimized with respect to the Lagrangian multipliers α_i and β_i:

$$L = \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \beta_i)(\alpha_j - \beta_j)k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \delta\sum_{i=1}^{n}\beta_i \tag{4}$$

subject to $0 \leq \alpha_i, \beta_i \leq \frac{1}{\nu n}$ and $\sum_{i=1}^{n}(\alpha_i - \beta_i) = 1$ for $\alpha_i, \beta_i \geq 0$. A new sample $\boldsymbol{z}$ is considered as a normal one if it lies between the lower hyperplane and the upper hyperplane. Otherwise, it is considered as an outlier.

2.3. Robust Support Vector Machine

The "Robust SVM" algorithm is another modified version of the standard SVM, and it aims at reducing the influence of the outliers on the decision rule of the classifier. The slack variables are replaced with other ones related to the distance between the samples and the center of the data in the feature space, which will cause the hyperplane to be shifted towards the normal samples.

The new slack variable represents the ration between the distance of the mapped samples to the center and the maximal value of this distance d_{max}, and it is computed as follows:

$$d_i = \big(k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \frac{2}{n}\sum_{j=1}^{n}k(\boldsymbol{x}_i, \boldsymbol{x}_j)\big)/d_{max}.$$

The Robust SVM needs to solve the following constrained optimization problem:

$$\min_{\boldsymbol{w},\rho} \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho \tag{5}$$

subject to $\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle \geq \rho - \lambda d_i$, where d_i are the slack variables and λ a regularization parameter. The following objective functional needs to be minimized

with respect to the Lagrangian multipliers α_i:

$$L = \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \lambda \sum_{i=1}^{n} d_i \alpha_i \tag{6}$$

subject to $0 \leq \alpha_i \leq 1$ and $\sum_{i=1}^{n} \alpha_i = 1$. This type of slack variables makes the decision boundary of the classifier less affected by the outliers.

2.4. Simple one-class

The simple one-class was introduced to overcome the time consumption drawback of the existing algorithms. It is a fast and simple one-class approach, which is based on the computation of the Euclidean distance in the feature space, and it does not need to solve any quadratic programming problem.

Let $\boldsymbol{c}_n$ be the center of the data in the feature space, namely $\boldsymbol{c}_n = \frac{1}{n}\sum_{i=1}^{n} \phi(\boldsymbol{x}_i)$. The simple one-class approach computes the Euclidean distance between the training samples and the center $\boldsymbol{c}_n$, and the expression of this distance is given as follows:

$$\left\|\phi(\boldsymbol{x}) - \frac{1}{n}\sum_{i=1}^{n} \phi(\boldsymbol{x}_i)\right\|_{\mathcal{H}}^2 = k(\boldsymbol{x}, \boldsymbol{x}) - \frac{2}{n}\sum_{j=1}^{n} k(\boldsymbol{x}, \boldsymbol{x}_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

After evaluating the Euclidean distance between all the training samples and the center, the simple one-class algorithm defines a threshold based on the estimated fraction of outliers among the training dataset. The decision function for a new sample $\boldsymbol{x}$ is defined by its Euclidean distance to the center. If this distance is greater than the predefined threshold, this sample is considered as an outlier.

3. Proposed One-class Framework

. In this paper, we propose a framework for one-class classification problems, by investigating a sparse formulation. We define the one-class by the hypersphere enclosing the samples in the RKHS, and the decision function of the classifier uses the distance in the RKHS between the sample $\phi(\boldsymbol{x})$ under scrutiny and the center of this hypershpere. If this distance is greater than a fixed threshold,

the sample is considered as an outlier. Otherwise, it is considered as a normal sample, i.e., belonging to the same distribution as the training dataset.

The mean of the mapped data is given by the expectation of the data in the RKHS, namely $\mathrm{E}[\phi(\boldsymbol{x})]$. One can estimate this expectation by the empirical center of the data in that space, namely:

$$\boldsymbol{c}_n = \frac{1}{n}\sum_{i=1}^{n}\phi(\boldsymbol{x}_i).$$

The center of the hypersphere is a linear combination of the mapped samples, namely $\boldsymbol{c}_{\mathcal{A}} = \sum_{j=1}^{n}\beta_j\phi(\boldsymbol{x}_j)$, having β_j the corresponding coefficients. The center $\boldsymbol{c}_{\mathcal{A}}$ has to be chosen in a way to efficiently represent the first order moment of the distribution of the training dataset in the RKHS. Therefore, we define $\boldsymbol{c}_{\mathcal{A}}$ by the approximation of the empirical center $\boldsymbol{c}_n$ in the RKHS. The expression of the Euclidean distance between any sample $\boldsymbol{x}$ and the sparse center is given as follows:

$$\|\phi(\boldsymbol{x}) - \boldsymbol{c}_{\mathcal{A}}\|_2^2 = k(\boldsymbol{x},\boldsymbol{x}) - 2\sum_{i=1}^{n}\beta_i k(\boldsymbol{x}_i,\boldsymbol{x}) + \sum_{i,j=1}^{n}\beta_i\beta_j k(\boldsymbol{x}_i,\boldsymbol{x}_j).$$

. In order to obtain a sparse approach, by analogy with the SVM formulation, only a small fraction of the coefficients β_j in the center's expression has to be nonzero. Therefore, in order to estimate the sparse center, we propose to minimize the error of approximating the empirical center $\boldsymbol{c}_n$ with $\boldsymbol{c}_{\mathcal{A}}$ in a way to get a good representation of the training samples. The optimization problem takes the following form:

$$\arg\min_{\beta_j}\Big\|\frac{1}{n}\sum_{i=1}^{n}\phi(\boldsymbol{x}_i) - \sum_{j=1}^{n}\beta_j\phi(\boldsymbol{x}_j)\Big\|_2^2, \tag{7}$$

subject to some sparsity-inducing constraints. Such constraints include that the ℓ_0-norm of $\boldsymbol{\beta}$ shall not exceed some predefined threshold. For computational reasons, the ℓ_0-norm is often replaced by the ℓ_1-norm, i.e., $\sum|\boldsymbol{\beta}|$, which is the closest convex norm to the ℓ_0-norm.

We note that this optimization problem has a similar form as the one in shrinkage approaches used for regression problems, namely $\arg\min_\beta \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$

subject to some sparsity-inducing constraints, such as $\sum |\beta|$ cannot exceed some predefined threshold. These shrinkage methods have been usually used for the selection of the most relevant features, where only the corresponding coefficients remain nonzero. We propose to revisit three of these well-known shrinkage approaches, namely Least Angle Regression, Least Absolute Shrinkage and Selection Operator, and Elastic Net. We modify the algorithms of these approaches in order to estimate the sparse center $\boldsymbol{c}_{\mathcal{A}}$, by adapting them to solve the optimization problem (7), which selects the most relevant samples among the training dataset. Next, we detail the modified shrinkage methods, by revisiting the corresponding optimization problems and the resulting solutions.

3.1. Least Angle Regression

. The first shrinkage method studied in this paper is the Least Angle Regression (LARS), which builds a model sequentially by augmenting the set of the most relevant samples, one sample at a time. The modification of the LARS algorithm for one-class problems allows to solve the following optimization problem:

$$\arg\min_{\beta_j} \|\frac{1}{n}\sum_{i=1}^{n}\phi(\boldsymbol{x}_i) - \sum_{j=1}^{n}\beta_j\phi(\boldsymbol{x}_j)\|_2^2 \tag{8}$$

subject to $\sum |\beta| < t$, for some parameter t. Let $\widehat{\boldsymbol{c}}_{\mathcal{A}_k}$ be the estimation of the sparse center in the subspace $\mathcal{A}_k$ of the most relevant samples at step k, and $(\boldsymbol{c}_n - \widehat{\boldsymbol{c}}_{\mathcal{A}_k})$ the current residual. LARS considers the sample having the largest absolute correlation with the current residual $(\boldsymbol{c}_n - \widehat{\boldsymbol{c}}_{\mathcal{A}_k})$, and projects the other samples on this first one. LARS repeats the selection process until a new sample has the same correlation level with the current residual, and continues in a direction that preserve equiangularity between the samples of $\mathcal{A}$, until a third one enters the set of the most correlated samples. LARS continues equiangularly between these three samples until a forth one enters this set, and so on. An example of the successive LARS estimates is illustrated in figure 1, where the algorithm starts at $\widehat{\boldsymbol{c}}_{\mathcal{A}_0}$, and the equiangular vectors are updated in a way to preserve equal angles with the original axes.

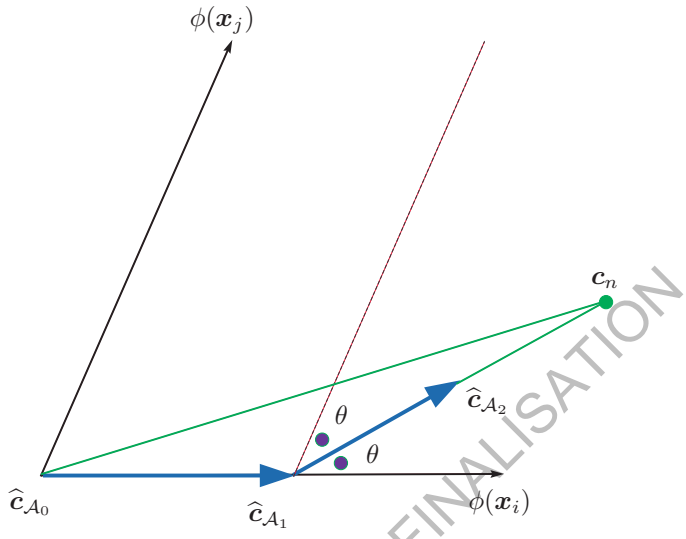

Figure 1: An illustration of the successive LARS estimates in a simple 2-dimensional space, where the algorithm starts at $\widehat{\boldsymbol{c}}_{\mathcal{A}_0} = \mathbf{0}$. In this example, the first residual $(\boldsymbol{c}_n - \widehat{\boldsymbol{c}}_{\mathcal{A}_0})$ makes a smaller angle with $\phi(\boldsymbol{x}_i)$ than with $\phi(\boldsymbol{x}_j)$, so we start moving in the direction of $\phi(\boldsymbol{x}_i)$ and $\widehat{\boldsymbol{c}}_{\mathcal{A}_1} = \beta_1 \phi(\boldsymbol{x}_i)$. At the next step, the current residual $(\boldsymbol{c}_n - \widehat{\boldsymbol{c}}_{\mathcal{A}_1})$ makes equal angles θ with $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$, so we have to move in a direction that preserves this equiangularity, as given with $\widehat{\boldsymbol{c}}_{\mathcal{A}_2}$.

. LARS algorithm begins at $\widehat{\boldsymbol{c}}_{\mathcal{A}} = \mathbf{0}$, and we update $\widehat{\boldsymbol{c}}_{\mathcal{A}}$ at each iteration. Let $\boldsymbol{X} = (\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \ldots, \phi(\boldsymbol{x}_n))$, $\boldsymbol{X}_{\mathcal{A}}$ denotes the matrix containing the retained samples of the set $\mathcal{A}$, based on the greatest absolute correlation criterion, and $\boldsymbol{K}_{\mathcal{A}}$ the $|\mathcal{A}| \times |\mathcal{A}|$ corresponding kernel matrix, where $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$. The expression of the current estimate of the sparse center takes the form: $\widehat{\boldsymbol{c}}_{\mathcal{A}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}$. LARS considers the sample having the largest absolute correlation with the current residual, where the vector of current correlations is defined as follows:

$$\begin{aligned}\widehat{\boldsymbol{corr}} &= \boldsymbol{X}^T(\boldsymbol{c}_n - \widehat{\boldsymbol{c}}_{\mathcal{A}}) \\ &= \frac{1}{n}\sum_{i,j=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i,j=1}^{n} \widehat{\beta}_j k(\boldsymbol{x}_i, \boldsymbol{x}_j),\end{aligned}$$

having $\widehat{\beta}_j$ the current estimate of the center's coefficients. The next step is to project all the samples on the subspace spanned by the samples of $\mathcal{A}$, in a way to preserve equal angles between these samples. The equiangular vector needed for the projection operation has the following form:

$$\boldsymbol{u}_{\mathcal{A}} = \boldsymbol{X}_{\mathcal{A}} \boldsymbol{w}_{\mathcal{A}},$$

where $\boldsymbol{w}_{\mathcal{A}} = A_{\mathcal{A}} \boldsymbol{G}_{\mathcal{A}}^{-1} \mathbf{1}_{\mathcal{A}}$ is the weight vector making equal angles with the columns of $\boldsymbol{X}_{\mathcal{A}}$, $\boldsymbol{G}_{\mathcal{A}} = \boldsymbol{s}^T \boldsymbol{K}_{\mathcal{A}} \boldsymbol{s}$ is a matrix related to the set $\mathcal{A}$, $A_{\mathcal{A}} = (\mathbf{1}_{\mathcal{A}}^T \boldsymbol{G}_{\mathcal{A}}^{-1} \mathbf{1}_{\mathcal{A}})^{-\frac{1}{2}}$, and $\boldsymbol{s}$ denotes the vector of the signs of the current correlations with entries:

$$s_j = \text{sign}\{\widehat{corr}_j\}, \quad \text{for} \quad j = 1, 2, \ldots, n.$$

After computing $\boldsymbol{X}_{\mathcal{A}}$, $A_{\mathcal{A}}$, and $\boldsymbol{u}_{\mathcal{A}}$, the previous estimate $\widehat{\boldsymbol{c}}_{\mathcal{A}}$ is updated to:

$$\widehat{\boldsymbol{c}}_{\mathcal{A}+} = \widehat{\boldsymbol{c}}_{\mathcal{A}} + \widehat{\gamma} \boldsymbol{u}_{\mathcal{A}}$$

using the equiangular vector, where

$$\widehat{\gamma} = \min_{j=1,\cdots,|\mathcal{A}^c|} \Big\{ \frac{\widehat{C} - \widehat{corr}_j}{A_{\mathcal{A}} - a_j}, \frac{\widehat{C} + \widehat{corr}_j}{A_{\mathcal{A}} + a_j} \Big\},$$

having min the minimum over the positive components, $\mathcal{A}^c$ the complementary set of $\mathcal{A}$, a_j an element of the inner product vector defined by

$$\boldsymbol{a} = \boldsymbol{X}^T \boldsymbol{u}_{\mathcal{A}} = \boldsymbol{X}^T \boldsymbol{X}_{\mathcal{A}} \boldsymbol{w}_{\mathcal{A}} = \sum_{i=1}^{n} \sum_{j=1}^{|\mathcal{A}|} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \boldsymbol{w}_{\mathcal{A}},$$

and $\widehat{C} = \max_j \{|\widehat{corr}_j|\}$. Finally, the coefficients $\boldsymbol{\beta}$ are updated as follows:

$$\boldsymbol{\beta}_{new} = \widehat{\boldsymbol{\beta}} + \widehat{\gamma} \boldsymbol{s}^T \boldsymbol{w}_{\mathcal{A}}. \tag{9}$$

This algorithm inherits the drawbacks of the conventional LARS algorithm. The main drawback is with highly correlated samples, which may limit its application to high dimensional data. Another drawback is its sensitivity to the effects of noise.

3.2. Least Absolute Shrinkage and Selection Operator

. The second shrinkage algorithm modified in this paper is the Least Absolute Shrinkage and Selection Operator (LASSO). The objective function in the LASSO involves minimizing the residual sum of squares, the same entity as in ordinary least squares (OLS) regression and LARS, subject to a bound on the sum of the absolute value of the coefficients. In other words, LASSO minimizes the residual sum of squares under a constraint on the ℓ_1-norm of the coefficient vector. It is easy to see that the ℓ_1-norm constraint induces sparsity in the solution. The LASSO shrinks the estimated coefficients towards the origin and sets some of them to zero, in a way to retain the most relevant samples and to discard the other ones. The main advantage of LASSO is with large volume datasets, where the coefficients of irrelevant samples are shrunk to zero.

. The LASSO solves the following optimization problem:

$$\arg\min_{\beta_j} \Big\| \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x}_i) - \sum_{j=1}^{n} \beta_j \phi(\boldsymbol{x}_j) \Big\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \tag{10}$$

for a given tuning parameter $\lambda > 0$. The tuning parameter controls the sparsity level of the solution. The solution path is obtained for several values of λ, and all the LASSO solutions can be generated by some modifications of the LARS algorithm detailed previously. Indeed, the sign of any nonzero coefficient β_j must agree with the sign s_j of the corresponding current correlation $\widehat{corr}_j$, namely:

$$\text{sign}(\beta_j) = \text{sign}(\widehat{corr}_j) = s_j,$$

for any $\boldsymbol{x}_j \in \mathcal{A}$ [30]. Unlike in LARS, the coefficients in LASSO do not change signs during the update step since they are piecewise linear. Let $\widehat{\boldsymbol{d}}$ be the the vector with entries $s_j w_{\mathcal{A}j}$ for any $\boldsymbol{x}_j \in \mathcal{A}$, and zero elsewhere. To update the coefficients as in equation (9), we have:

$$\beta_j(\gamma) = \widehat{\beta}_j + \gamma d_j \quad \text{for} \quad \boldsymbol{x}_j \in \mathcal{A}.$$

Therefore, $\beta_j(\gamma)$ changes sign at:

$$\gamma_j = -\frac{\beta_j}{d_j},$$

having the first such change occurring at $\widetilde{\gamma} = \min_{\gamma_j > 0}\{\gamma_j\}$. The sign restriction is violated when $\widetilde{\gamma} < \widehat{\gamma}$, and $\beta_j(\gamma)$ cannot be a LASSO solution; $\beta_j(\gamma)$ has changed sign while $c_j(\gamma)$ has not. The corresponding sample $\boldsymbol{x}_j$ is removed from the set of the most relevant samples, namely $\mathcal{A} = \mathcal{A} \setminus \{\boldsymbol{x}_j\}$, and the algorithm moves to the next equiangular direction. Therefore, this modification allows the set of the most relevant samples to increase or decrease one at a time until the LARS algorithm leads to all LASSO solutions.

This modified version of the LASSO allows to overcome the LARS algorithm by adding/removing one sample at a time. On the other hand, this algorithm inherits the drawbacks of the conventional LASSO. The main drawback remains with high correlated variables, where LASSO tends to arbitrarily select only one variable from the group and ignores the others, thus it cannot do group selection.

3.3. Elastic Net

. The Elastic net is a LARS-derived regularization and variable selection method that overcomes the limitations of LARS and LASSO methods, specifically when it comes to hight correlated variables. The Elastic net optimization problem combines the ℓ_1 and ℓ_2 penalties of the LASSO and ridge methods, thus Elastic net produces a sparse model and it does both continuous shrinkage and variable selection. In addition, unlike LARS and LASSO, Elastic net has a grouping effect where strongly correlated samples are in or out of the model together. The Elastic net has the advantage of including automatically all the highly correlated variables in the group, and it was compared to a stretchable fishing net that retains "all the big fish" [33]. In addition, the entire Elastic net solution path can be directly computed from the LARS algorithm.

. The naïve Elastic net optimization problem is defined as follows:

$$\arg\min_{\beta_j} \Big\| \frac{1}{n} \sum_{i=1}^{n} \phi(\boldsymbol{x}_i) - \sum_{j=1}^{n} \beta_j \phi(\boldsymbol{x}_j) \Big\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2,$$

for some given tuning parameters $\lambda_1, \lambda_2 > 0$, and it becomes a pure LASSO optimization when $\lambda_2 = 0$, and a simple ridge regression when $\lambda_1 = 0$. This

optimization problem incurs a double amount of shrinkage from Ridge and LASSO procedures, which introduces unnecessary extra bias compared with pure LASSO or ridge shrinkage. In order to improve the prediction performance, the coefficients of the naïve version of Elastic net are rescaled to obtain the Elastic net coefficients as follows:

$$\boldsymbol{\beta}_{(\text{Elastic net})} = (1+\lambda_2)\boldsymbol{\beta}_{(\text{naïve Elastic net})}.$$

Therefore, rescaling the coefficients will undo the double amount of shrinkage.

. The naïve Elastic net problem can be transformed into an equivalent LASSO problem as in equation (10), and this is achieved by replacing the parameter λ with $\lambda_1/\sqrt{1+\lambda_2}$ [33]. And as detailed in the previous section, a simple modification in the LARS algorithm leads all the LASSO solution paths. Therefore, the LARS algorithm leads all the Elastic net solution paths. An example that highlights the differences in the solution paths of LARS, LASSO and Elastic net algorithms is illustrated in figure 2.

4. Extension to the Mahalanobis Distance

. The main drawback of using the Euclidian distance is its sensitivity to the scale variation in each direction, so we propose to use the Mahalanobis distance in the decision function of the classifier. In fact, the Mahalanobis distance takes into account the different scaling of the coordinate axes [35]. The Mahalanobis distance between a sample $\phi(\boldsymbol{x})$ and the center $\boldsymbol{c}_{\mathcal{A}}$ in the RKHS is given as follows:

$$\left\|\phi(\boldsymbol{x}) - \boldsymbol{c}_{\mathcal{A}}\right\|^2_{\boldsymbol{\Sigma}} = \left(\phi(\boldsymbol{x}) - \boldsymbol{c}_{\mathcal{A}}\right)^T \boldsymbol{\Sigma}^{-1} \left(\phi(\boldsymbol{x}) - \boldsymbol{c}_{\mathcal{A}}\right), \tag{11}$$

where $\boldsymbol{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\phi(\boldsymbol{x}_i) - \boldsymbol{c}_n)(\phi(\boldsymbol{x}_i) - \boldsymbol{c}_n)^T$ denotes the covariance matrix of the data. We cannot express $\boldsymbol{\Sigma}$ in terms of the data $\phi(\boldsymbol{x})$ without any explicit knowledge on the mapping function $\phi(\cdot)$. Therefore, we use the singular value decomposition of the covariance matrix, namely $\boldsymbol{\Sigma} = \boldsymbol{V}^T\boldsymbol{D}\boldsymbol{V}$, having $\boldsymbol{V}$ the matrix of eigenvectors $\boldsymbol{v}^k$ of $\boldsymbol{\Sigma}$, and $\boldsymbol{D}$ the diagonal matrix with the correspondent

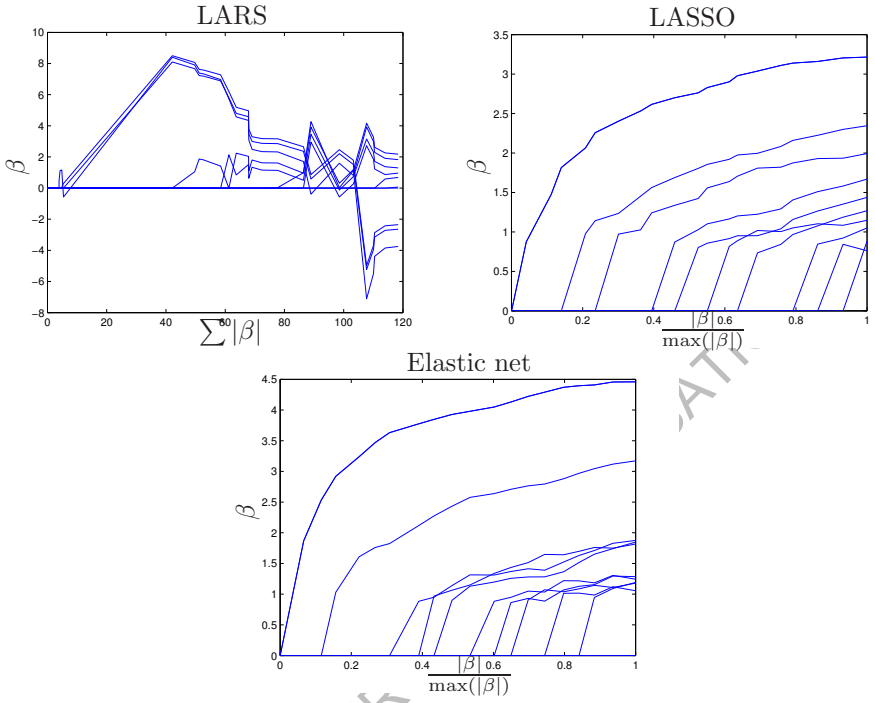

Figure 2: The solution paths of LARS, LASSO and Elastic net algorithms. The LARS solution paths are the most unstable, while Elastic net has smoother solution paths that clearly show the "grouping effect" advantage of correlated variables over the LASSO.

eigenvalues λ_k for $k = 1, 2, \cdots, n$. Since $\boldsymbol{V}$ is an orthogonal matrix, $\boldsymbol{\Sigma}^{-1}$ takes this form: $\boldsymbol{\Sigma}^{-1} = \boldsymbol{V}^T \boldsymbol{D}^{-1} \boldsymbol{V}$, where each eigenvalue λ_k satisfies $\lambda_k \boldsymbol{v}^k = \boldsymbol{\Sigma} \boldsymbol{v}^k$, and each eigenvector is a linear combination of the samples $\phi(\boldsymbol{x}_i)$ in the RKHS, namely $\boldsymbol{v}^k = \sum_{i=1}^n \alpha_i^k (\phi(\boldsymbol{x}_i) - \boldsymbol{c}_n)$. By injecting the expression of $\boldsymbol{v}^k$ into the eigen decomposition of $\boldsymbol{\Sigma}$, namely $\lambda_k \boldsymbol{v}^k = \boldsymbol{\Sigma} \boldsymbol{v}^k$, the coefficients α_i are given by solving the eigen decomposition problem $n\lambda_k \boldsymbol{\alpha}^k = \widetilde{\boldsymbol{K}} \boldsymbol{\alpha}^k$, where the matrix $\widetilde{\boldsymbol{K}}$ of entries[2] $\widetilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the centered version of $\boldsymbol{K}$. Finally, the Mahalanobis

[2]The kernel function $\widetilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \widetilde{k}_{ij}$ is the centered version of $k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and it is computed as follows:

$$\widetilde{k}_{ij} = k_{ij} - \frac{1}{n} \sum_{r=1}^n k_{ir} - \frac{1}{n} \sum_{r=1}^n k_{rj} + \frac{1}{n^2} \sum_{r,s=1}^n k_{rs}.$$

distance in equation (11) is computed in the RKHS as follows:

$$\sum_{k=1}^{n}\frac{1}{\lambda_k}\Big(\sum_{i=1}^{n}\alpha_i^k k(\boldsymbol{x}_i,\boldsymbol{x})-\sum_{i=1}^{n}\alpha_i^k\frac{1}{n}\sum_{j=1}^{n}k(\boldsymbol{x}_j,\boldsymbol{x}) - \sum_{i,j=1}^{n}\alpha_i^k\beta_j k(\boldsymbol{x}_i,\boldsymbol{x}_j)+\sum_{i=1}^{n}\alpha_i^k\frac{1}{n}\sum_{j,l=1}^{n}\beta_l k(\boldsymbol{x}_j,\boldsymbol{x}_l)\Big)^2,$$

We make use of the advantages in KPCA, and only the most relevant eigenvectors are taken into consideration while the remaining ones are considered as noise. In other words, in order to compute the Mahalanobis distance in the RKHS, we project the data into the subspace spanned by the most relevant eigenvectors. We also adopt the kernel whitening normalization of the eigenvectors as proposed in [17], where the variance of the mapped data is constant for all the feature directions.

5. Theoretical Results

. In this section, we provide some theoretical results on the error of projecting the center of the data $\boldsymbol{c}_n$ and on the first kind error. To this end, we consider the projection into the subspace spanned by the most relevant eigenvectors as described in Section 4, thus replacing $\boldsymbol{\Sigma}$ by the corresponding approximation $\widehat{\boldsymbol{\Sigma}}$. Therefore, the Mahalanobis distance, *i.e.,* $\|\phi(\boldsymbol{x})-\boldsymbol{c}_n\|^2_{\boldsymbol{\Sigma}}$, is approximated by the distance between the projections in the corresponding subspace as follows: $\|\mathcal{P}\phi(\boldsymbol{x})-\mathcal{P}\boldsymbol{c}_n\|_{\widehat{\boldsymbol{\Sigma}}}$, having $\mathcal{P}$ the projection operator.

5.1. Projection error

Theorem 1. *Given a training dataset $\boldsymbol{x}_i$, $i=1,\ldots,n$ in a d-dimensional input space with its covariance matrix $\boldsymbol{\Sigma}$. The error of projecting the center of the data c_n onto the subspace spanned by the most relevant eigenvectors can be upper bounded by*

$$\frac{1}{n^2}\sum_{i=k+1}^{n}\lambda_i,$$

where $\lambda_{k+1},\ldots,\lambda_n$ represent the least relevant eigenvalues related to the remaining eigenvectors unused in the projection operation.

PROOF. The error of projecting $\boldsymbol{c}_n$ is expressed as follows:

$$\begin{aligned}\left\|(\boldsymbol{I}-\mathcal{P})c_n\right\|^2_{\widehat{\boldsymbol{\Sigma}}} &= \left\|\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{I}-\mathcal{P})\phi(\boldsymbol{x}_i)\right\|^2_{\widehat{\boldsymbol{\Sigma}}} \\ &\leq \frac{1}{n^2}\sum_{i=1}^{n}\left\|(\boldsymbol{I}-\mathcal{P})\phi(\boldsymbol{x}_i)\right\|^2_{\widehat{\boldsymbol{\Sigma}}} \\ &\leq \frac{1}{n^2}\sum_{i=k+1}^{n}\lambda_i,\end{aligned}$$

where the first inequality follows from the triangular inequality, and the error of projecting the samples $\phi(\boldsymbol{x}_i)$ can be bounded by $\sum_{i=k+1}^{n}\lambda_i$ as detailed in [2, Chapter 6].

255 *5.2. Error of the first kind*

. Let $\boldsymbol{c}_\infty$ denotes the expectation of the data in the feature space, namely $\mathrm{E}[\phi(\boldsymbol{x})]$, and ϵ_0 the projection error between $\boldsymbol{c}_\infty$ and $\boldsymbol{c}_n$, namely $\epsilon_0 = \|\mathcal{P}\boldsymbol{c}_n - \mathcal{P}\boldsymbol{c}_\infty\|_{\widehat{\boldsymbol{\Sigma}}}$. The samples of the training dataset are generated from the same distribution, and n_{out} is the number of outliers among this dataset.

Theorem 2. *If we consider the sphere centered on $\mathcal{P}\boldsymbol{c}_\mathcal{A}$ with radius R, and by the symmetry of the i.i.d assumption, we can bound the probability that a new random sample $\boldsymbol{x}$ lies outside this sphere excluding the outliers, with*

$$\mathrm{P}(\|\mathcal{P}\phi(\boldsymbol{x}) - \mathcal{P}\boldsymbol{c}_\mathcal{A}\|_{\widehat{\boldsymbol{\Sigma}}} > R + 2\epsilon_0 + 2\|\mathcal{P}\boldsymbol{c}_n - \mathcal{P}\boldsymbol{c}_\mathcal{A}\|_{\widehat{\boldsymbol{\Sigma}}}) \leq \frac{n_{out}+1}{n+1}.$$

PROOF. When all the training samples are inside the sphere centered on $\boldsymbol{c}_n$, it has been shown in [2] that the probability of a new sample $\boldsymbol{x}$ that lies outside this description is bounded by

$$\mathrm{P}(\|\phi(\boldsymbol{x}) - \boldsymbol{c}_n\| > R_1 + 2\epsilon_1) \leq \frac{1}{n+1},$$

having ϵ_1 the error of approximating $\boldsymbol{c}_\infty$, and R_1 the radius of the sphere, namely $R_1 = \max_{i=1,\cdots,n}\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}_n\|$. If we consider the sphere centered on the projected sparse center $\mathcal{P}c_\mathcal{A}$ with n_{out} outliers, and the distance between the projected sample $\mathcal{P}\phi(\boldsymbol{x})$ and $\mathcal{P}\boldsymbol{c}_\mathcal{A}$, we apply the triangular inequality twice

and we get the following relations:

$$\begin{aligned}\|\mathcal{P}\phi(\boldsymbol{x}) - \mathcal{P}\boldsymbol{c}_{\mathcal{A}}\|_{\widehat{\boldsymbol{\Sigma}}} &\leq \|\mathcal{P}\phi(\boldsymbol{x}) - \mathcal{P}\boldsymbol{c}_n\|_{\widehat{\boldsymbol{\Sigma}}} + \|\mathcal{P}\boldsymbol{c}_n - \mathcal{P}\boldsymbol{c}_{\mathcal{A}}\|_{\widehat{\boldsymbol{\Sigma}}} \\ &\leq \|\mathcal{P}\phi(\boldsymbol{x}) - \mathcal{P}\boldsymbol{c}_{\infty}\|_{\widehat{\boldsymbol{\Sigma}}} + \epsilon_0 + \|\mathcal{P}\boldsymbol{c}_n - \mathcal{P}\boldsymbol{c}_{\mathcal{A}}\|_{\widehat{\boldsymbol{\Sigma}}},\end{aligned}$$

and

$$\begin{aligned}\|\mathcal{P}\phi(\boldsymbol{x}) - \mathcal{P}\boldsymbol{c}_{\mathcal{A}}\|_{\widehat{\boldsymbol{\Sigma}}} &\geq \|\mathcal{P}\phi(\boldsymbol{x}) - \mathcal{P}\boldsymbol{c}_n\|_{\widehat{\boldsymbol{\Sigma}}} - \|\mathcal{P}\boldsymbol{c}_n - \mathcal{P}\boldsymbol{c}_{\mathcal{A}}\|_{\widehat{\boldsymbol{\Sigma}}} \\ &\geq \|\mathcal{P}\phi(\boldsymbol{x}) - \mathcal{P}\boldsymbol{c}_{\infty}\|_{\widehat{\boldsymbol{\Sigma}}} - \epsilon_0 - \|\mathcal{P}\boldsymbol{c}_n - \mathcal{P}\boldsymbol{c}_{\mathcal{A}}\|_{\widehat{\boldsymbol{\Sigma}}}.\end{aligned}$$

From these two inequalities, and by the symmetry of the i.i.d assumption, the probability of a new sample $\boldsymbol{x}$ lying outside this distribution is bounded by

$$\mathrm{P}(\|\mathcal{P}\phi(\boldsymbol{x}) - \mathcal{P}\boldsymbol{c}_{\mathcal{A}}\|_{\widehat{\boldsymbol{\Sigma}}} > R + 2\epsilon_0 + 2\|\mathcal{P}\boldsymbol{c}_n - \mathcal{P}\boldsymbol{c}_{\mathcal{A}}\|_{\widehat{\boldsymbol{\Sigma}}}) \leq \frac{n_{out}+1}{n+1}.$$

6. Experimental Results

. The proposed one-class algorithms are applied on two simulated datasets and on three real datasets from the Mississipi State University SCADA Laboratory and from the University of California machine learning repository. The selection of the most relevant samples in the proposed framework is performed via the aforementioned shrinkage algorithms, namely LARS, LASSO and Elastic net. In each case of these three subset selection approaches, the decision function of the classifier is defined using the Euclidean diastance and the Mahalanobis distance. The Gaussian kernel is used in this paper, for it is the most common and suitable kernel for one-class classification problems. It is given by $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\big(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2}\big)$, where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two input samples, and $\|\cdot\|_2$ represents the l_2-norm in the input space. The bandwidth parameter σ is computed as proposed in [14], namely $\sigma = \frac{d_{\max}}{\sqrt{2M}}$, where $d_{\max}$ refers to the maximal distance between any two samples in the input space, and M represents the upper bound on the number of outliers among the training dataset.

6.1. Results On Simulated Data

. The proposed algorithms are applied in the first place on two simulated datasets, the sinusoidal and the square noise datasets [28]. The main chal-

lenge when it comes to simulated data is to define a good description boundary around the training data, in a way to enclose the normal samples while avoiding the extremes cases, namely overfitting and underfitting the data. We compared the results of the proposed algorithms with four other one-class classification approaches, namely SVDD, simple one-class, slab SVM and robust SVM as shown in figures 3 and 4. We note that the sinusoidal dataset has 95 samples, the square noise has 450 samples, and the sparse center used only 15% of the training data to define these boundaries. When it comes to the results on the sinusoidal dataset, the SVDD, simple one-class, slab SVM and robust SVM have loose boundaries. The use of the Euclidean distance in the decision function of the classifier in the proposed algorithms led also to loose boundaries, which can be explained by the sensitivity of this distance to the scale in each feature direction. On the other hand, the use of the Mahalanobis distance instead of the Euclidean distance gives better boundaries, and combining Elastic net with the Mahalanobis distance outperforms both LASSO and LARS, and it leads to the best result with the tightest description boundary. When it comes to the square noise dataset, the SVDD, simple one-class, slab SVM and robust SVM have the same loose boundaries, LARS and LASSO have good results with the Mahalanobis distance, and the best result is achieved when the Mahalanobis distance is used with the modified Elastic net algorithm. Therefore, the latter combination, namely Elastic net with Mahalanobis distance, outperforms LASSO, LARS and the other approaches.

6.2. Results On Real Data

. The proposed one-class algorithms are now tested on two real datasets from the Mississipi State University SCADA Laboratory, the gas pipeline and the water storage tank testbeds [36]. The gas pipeline is used to move petroleum products to the market, and the water storage tank is similar to the oil storage tanks found in the petrochemical industry. These real datasets raise many challenges, where each input sample consists of 27 heterogeneous attributes for the gas pipeline and 24 attributes for the water storage tank, i.e., gas pressure,

water level, pump state, target gas pressure/water level, PID's parameters, time interval between packets, length of the packets, and command/response functions. Furthermore, 28 types of attacks are injected into the network traffic of the system in order to hide its real functioning state and to disrupt the communication. These attacks are arranged into 7 groups: Naive Malicious Response Injection (NMRI), Complex Malicious Response Injection (CMRI), Malicious State Command Injection (MSCI), Malicious Parameter Command Injection (MPCI), Malicious Function Command Injection (MFCI), Denial of Service (DOS) and Reconnaissance Attacks (RA). See [36] for more details. We also tested the one-class algorithm on another complex dataset, namely the water treatment plant dataset from the UCI machine learning repository [38]. This dataset comes from the daily measures of sensors in a urban waste water treatment plant. Each sample contains 38 attributes related to the measurements of several important components in the water like input zinc, input PH, input biological demand of oxygen, input suspended solids, input conductivity, input volatile suspended solids, input sediments to secondary settler, output chemical demand of oxygen, output volatile suspended solids, and other attributes. The train set contains samples related to four different normal situations while the test set encloses measurements of abnormal situations like after storms or when solids overload.

. In high-dimensional spaces, the first criterion for one-class classification algorithms is to have good detection rates. We tested these one-class algorithms on nearly 100 000 samples related to the aforementioned attacks, and the detection rates are given in Tables 1, 2 and 3. We note that the sparse center in the proposed algorithms depends only on 10% of the training samples. The worst detection rates are achieved with the simple one-class approach, which can be explained by its high sensitivity to the presence of outliers among the training dataset. The cases where the Mahalanobis distance is used in the decision function of the classifier have better results than the cases with the Euclidean distance, due to the strong properties of the first one and to the scale sensitivity

Table 1: Error detection probabilities for the gas pipeline testbed.

					In this paper					
					Euclidean distance			Mahalanobis distance		
	SVDD	Slab SVM	Robust SVM	simple one-class	LARS	LASSO	Elastic net	LARS	LASSO	Elastic net
NMRI	98.1	98.4	92.9	91.7	98.3	98.7	99.1	99.1	98.9	**99.2**
CMRI	**99.5**	**99.5**	98.5	95.4	98.1	98.3	99.2	98.7	98.8	**99.5**
MSCI	**89.1**	86.2	54.9	22.6	55.8	57.3	68.1	71.1	74.5	79.3
MPCI	98.2	96.9	98.1	94.1	97.1	96.7	97.8	98.2	97.6	**98.9**
MFCI	**89.9**	89.4	64.7	31.6	77.8	80.1	83.6	81.3	82.7	85.9
DOS	96.1	96.3	95.5	68.5	96.1	96.9	97.1	97.3	97.2	**97.5**
RA	**99.8**	**99.8**	99.7	98.1	99.1	99.5	**99.8**	99.6	99.7	**99.8**

of the latter one. LARS and LASSO algorithms have nearly the same good results, whereas Elastic net outperforms both shrinkage algorithms with both Euclidean and Mahalanobis distances. The best results are achieved when Elastic net is used to select the most relevant samples, and the norm in the decision function of the classifier is the Mahalanobis distance. The latter combination gives better detection rates than all of the other approaches for the different types of the studied attacks. The second criterion for one-class approaches is the time consumption of the algorithms. Table 4 shows the estimated time for each approach, and it indicates that the modified subset selection algorithms in the proposed framework are faster than the other approaches regardless of the shrinkage method used, with both Euclidean and Mahalanobis distances. The fastest algorithm occurs with LARS for the gas pipeline and the UCI testbeds, and the simple one-class for the water storage testbed, while the slowest approaches are the ones in which a constrained quadratic programming problem has to be resolved, namely SVDD, slab SVM and robust SVM, having the slab SVM the slowest one. Therefore, combining the Mahalanobis distance with Elastic net leads to the best detection rates, and it is up to 25 times faster than the other one-class algorithms.

Table 2: Error detection probabilities for the water storage testbed.

					In this paper					
					Euclidean distance			Mahalanobis distance		
	SVDD	Slab SVM	Robust SVM	simple one-class	LARS	LASSO	Elastic net	LARS	LASSO	Elastic net
NMRI	95.1	92.2	94.1	88.2	93.4	91.7	94.7	97.4	94.1	**98.1**
CMRI	61.2	63.5	59.7	46.2	59.1	62.4	69.2	71.8	67.7	**74.1**
MSCI	97.3	96.9	98.1	96.3	97.1	97.4	97.9	98.1	98.1	**98.3**
MPCI	98.6	99.1	99.2	97.6	98.9	97.9	99.1	99.1	98.4	**99.7**
MFCI	97.9	98.7	98.1	40.6	97.1	98.4	99.1	99.1	99.3	**99.8**
DOS	71.7	73.4	59.8	55.3	72.3	71.2	74.7	81.1	79.1	**82.6**
RA	97.8	98.1	98.7	95.9	98.1	98.4	98.7	99.1	99.3	**99.5**

Table 3: Error detection probabilities for the UCI water treatment testbed.

					In this paper					
					Euclidean distance			Mahalanobis distance		
Approach	SVDD	Slab SVM	Robust SVM	simple one-class	LARS	LASSO	Elastic net	LARS	LASSO	Elastic net
P_{ed}	78.6	81.6	74.7	64.8	71.4	71.4	78.6	85.7	85.7	**92.1**

7. Conclusion

. In this paper, we proposed a sparse framework for one-class classification problems. We defined our one-class classifier by the hypersphere enclosing the samples in the Reproducing Kernel Hilbert Space. We defined the center of this hypersphere by the approximation of the empirical center of the data in the RKHS, and this sparse center depends only on a small fraction of the training samples. In order to select the most relevant samples, we modified the algorithms of three well-known shrinkage approaches, namely LARS, LASSO and Elastic net. We adapted these algorithms to the estimation of the center in the RKHS. We used the Euclidean and the Mahalanobis distances in the decision function of the classifier, and we tested the proposed algorithms on simulated data as well as on real data from the Mississippi State University SCADA Laboratory and from University of California UCI machine learning repository. The tests showed that the proposed algorithms have the best results specially when the modified Elastic net is used to select the most relevant samples, and the

Table 4: Estimated time (in seconds) of each approach.

					In this paper					
					Euclidean distance			Mahalanobis distance		
	SVDD	Slab SVM	Robust SVM	simple one-class	LARS	LASSO	Elastic net	LARS	LASSO	Elastic net
gas	70.23	302.74	61.32	9.23	**9.12**	12.31	13.81	9.85	13.93	14.22
water	123.72	557.23	102.28	**10.41**	10.83	13.62	14.27	11.79	14.10	15.72
UCI	12.91	78.95	14.78	1.79	**1.61**	2.27	2.51	2.11	2.83	2.94

norm in the decision function of the classifier is the Mahalanobis distance. The latter combination is faster than the common one-class algorithms existing in the literature, and it leads to the best detection rates for most of the attacks.

. For future works, a detailed study on the other existing subset selection algorithms could be investigated, since it is a very important step in sparse approaches for the selection of the most relevant samples. The advantages of using of the Mahalanobis distance should be widely investigated to obtain more adapted kernel functions. Furthermore, the implementation of the modified algorithms should be optimized in order to decrease their computational cost. Online versions of the proposed algorithms could also be investigated to improve live detection in real time applications. Finally, an extension of this work to become suitable for multiclass classification could be investigated in order to identify the type of the detected intrusion.

Acknowledgment

The authors would like to thank Thomas Morris and the SCADA Laboratory for providing the SCADA dataset, and the French "Agence Nationale de la Recherche"(ANR) grant SCALA for supporting this work.

References

[1] T. Hofmann, B. Schölkopf, A. J. Smola, Kernel methods in machine learning, Annals of Statistics 36 (2008) 1171–1220.

[2] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, NY, USA, 2004.

[3] N. Bredeche, Z. Shi, J.-D. Zucker, Perceptual learning and abstraction in machine learning: an application to autonomous robotics, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 36 (2) (2006) 172–181. doi:10.1109/TSMCC.2006.871139.

[4] D. Strauss, W. Delb, J. Jung, P. Plinkert, Adapted filter banks in machine learning: applications in biomedical signal processing, in: Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). IEEE International Conference on, Vol. 6, 2003, pp. VI–425–8 vol.6. doi:10.1109/ICASSP.2003.1201709.

[5] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, H. Snoussi, Target tracking using machine learning and Kalman filter in wireless sensor networks, Sensors Journal, IEEE 14 (10) (2014) 3715–3725. doi:10.1109/JSEN.2014.2332098.

[6] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, H. Snoussi, Kernel-based localization using fingerprinting in wireless sensor networks, in: Proc. 14th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Germany, 2013, pp. 744–748. doi:10.1109/SPAWC.2013.6612149.

[7] J. P. Vert, K. Tsuda, B. Scholkopf, A primer on kernel methods, Kernel Methods in Computational Biology (2004) 35–70.

[8] N. Aronszajn, Theory of reproducing kernels, Trans. Amer. Math. Soc. 68 (3) (1950) 337 – 404.

[9] C.-W. Ten, C.-C. Liu, G. Manimaran, Vulnerability assessment of cybersecurity for scada systems, Power Systems, IEEE Transactions on 23 (4) (2008) 1836–1846. doi:10.1109/TPWRS.2008.2002298.

[10] S. S. Khan, M. G. Madden, A survey of recent trends in one class classification, in: Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science, AICS'09, 2010, pp. 188–197.

[11] Z. Zeng, Y. Fu, G. Roisman, Z. Wen, Y. Hu, T. Huang, One-class classification for spontaneous facial expression analysis, in: Automatic Face and Gesture Recognition (FGR). 7th International Conference on, 2006, pp. 281–286. doi:10.1109/FGR.2006.83.

[12] O. Mazhelis, One-class classifiers : a review and analysis of suitability in the context of mobile-masquerader detection, South African Computer Journal 36 (2006) 29–48.

[13] A. B. Gardner, A. M. Krieger, G. Vachtsevanos, B. Litt, One-class novelty detection for seizure analysis from intracranial EEG, Journal of Machine Learning Research 7 (2006) 1025–1044.

[14] P. Nader, P. Honeine, P. Beauseroy, l_p-norms in one-class classification for intrusion detection in SCADA systems, Industrial Informatics, IEEE Transactions on 10 (4) (2014) 2308–2317. doi:10.1109/TII.2014.2330796.

[15] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471. doi:10.1162/089976601750264965.

[16] D. M. J. Tax, R. P. W. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.

[17] D. M. J. Tax, P. Juszczak, Kernel whitening for one-class classification, in: Pattern Recognition with Support Vector Machines, First International Workshop, Niagara Falls, Canada, August 10, 2002, pp. 40–52. doi:10.1007/3-540-45665-1_4.
URL http://dx.doi.org/10.1007/3-540-45665-1_4

[18] M. E. Azami, C. Lartizien, S. Canu, Robust outlier detection with L0-SVDD, in: 22th European Symposium on Artificial Neural Networks, ESANN 2014, Bruges, Belgium, April 23-25, 2014, 2014.

[19] Z. Noumir, P. Honeine, C. Richard, On simple one-class classification methods, in: Proc. IEEE International Symposium on Information Theory, MIT, Cambridge (MA), USA, 2012.

[20] B. Schölkopf, J. Giesen, S. Spalinger, Kernel methods for implicit surface modeling, in: Advances in Neural Information Processing Systems 17, MIT Press, 2005, pp. 1193–1200.

[21] Q. Tao, G.-w. Wu, J. Wang, A new maximum margin algorithm for one-class problems and its boosting implementation, Pattern Recogn. 38 (7) (2005) 1071–1077.

[22] M. Eigensatz, J. Giesen, M. Manjunath, The solution path of the slab support vector machine, in: The 20th Canadian Conference on Computational Geometry, Mcgill University, CCCG, 2008, pp. 211–214.

[23] I. Tsang, J. Kwok, S. Li, Learning the kernel in Mahalanobis one-class support vector machines, in: Neural Networks, 2006. IJCNN '06. International Joint Conference on, 2006, pp. 1169–1175. doi:10.1109/IJCNN.2006.246823.

[24] D. Wang, D. Yeung, E. C. C. Tsang, Structured one-class classification, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 36 (6) (2006) 1283–1295. doi:10.1109/TSMCB.2006.876189.

[25] Q. Song, W. Hu, W. Xie, Robust support vector machine with bullet hole image classification, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 32 (4) (2002) 440–448. doi:10.1109/TSMCC.2002.807277.

[26] M. Amer, M. Goldstein, S. Abdennadher, Enhancing one-class support vector machines for unsupervised anomaly detection, in: Proceedings of the

ACM SIGKDD Workshop on Outlier Detection and Description (ODD), August 11-14,, New York, USA, 2013, pp. 8–15.

[27] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.

[28] H. Hoffmann, Kernel pca for novelty detection, Pattern Recognition 40 (3) (2007) 863 – 874. doi:10.1016/j.patcog.2006.07.009.

[29] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[30] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Annals of Statistics 32 (2004) 407–499.

[31] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, Series B 58 (1996) 267–288.

[32] M. R. Osborne, B. Presnell, B. A. Turlach, On the lasso and its dual, Journal of Computational and Graphical Statistics 9 (1999) 319–337.

[33] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society, Series B 67 (2005) 301–320.

[34] D.-X. Zhou, On grouping effect of elastic net, Statistics & Probability Letters 83 (2013) 2108–2112.

[35] P. C. Mahalanobis, On the generalised distance in statistics, in: Proceedings National Institute of Science, India, Vol. 2, 1936, pp. 49–55.

[36] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu, R. Reddi, A control system testbed to validate critical infrastructure protection concepts, International Journal of Critical Infrastructure Protection 4 (2) (2011) 88 – 103. doi:10.1016/j.ijcip.2011.06.005.

[37] T. Morris, R. B. Vaughn, Y. S. Dandass, A testbed for scada control system cybersecurity research and pedagogy, in: CSIIRW, Oak Ridge, Tennessee, 2011.

[38] K. Bache, M. Lichman, UCI machine learning repository (2013).

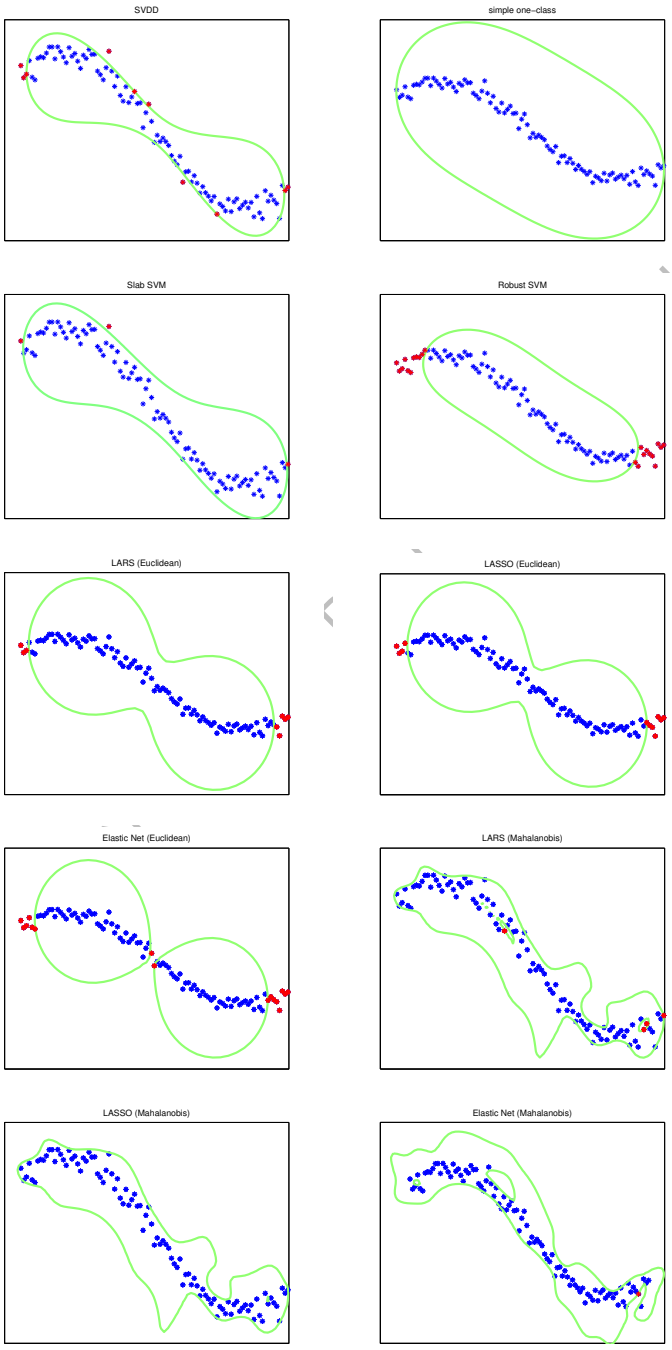

Figure 3: The decision boundaries on the sinusoidal dataset for several one-class algorithms. The description boundaries are given by the green lines, the red samples are the ones considered as outliers while the normal samples are in blue. Elastic net outperforms LARS and LASSO, and it gives the best decision boundary with the Mahalanobis distance.

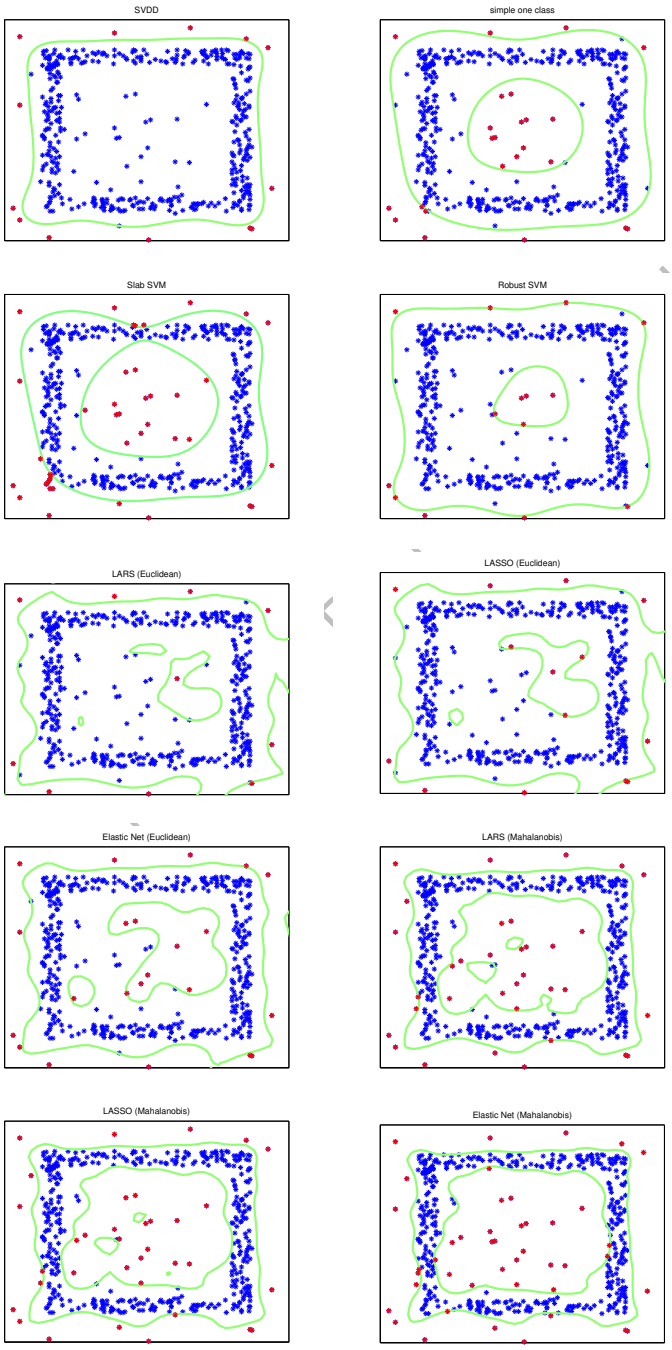

Figure 4: The decision boundaries on the square-noise dataset for the studied algorithms. The description boundaries are given by the green lines, the outliers correspond to the red samples while the normal samples are in blue. Elastic net outperforms LARS, LASSO, and the other approches, and it leads to the best description boundary when the metric in the decision function of the classifier is the Mahalanobis distance.