Mahalanobis-Based One-Class Classification

Patric Nader, Paul Honeine, Pierre Beauseroy

MAHALANOBIS-BASED ONE-CLASS CLASSIFICATION

Patric Nader, Paul Honeine, and Pierre Beauseroy.

Institut Charles Delaunay (CNRS), Université de Technologie de Troyes, France
{patric.nader, paul.honeine, pierre.beauseroy}@utt.fr

ABSTRACT

Machine learning techniques have become very popular in the past decade for detecting nonlinear relations in large volumes of data. In particular, one-class classification algorithms have gained the interest of the researchers when the available samples in the training set refer to a unique/single class. In this paper, we propose a simple one-class classification approach based on the Mahalanobis distance. We make use of the advantages of kernel whitening and KPCA in order to compute the Mahalanobis distance in the feature space, by projecting the data into the subspace spanned by the most relevant eigenvectors of the covariance matrix. We also propose a sparse formulation of this approach. The tests are conducted on simulated data as well as on real data.

Index Terms— Kernel methods, one-class classification, Mahalanobis distance.

1. INTRODUCTION

Machine learning techniques have gained a lot of attention in the past few years since they provide a powerful tool for detecting nonlinear relations and hidden regularities in large volumes of data [1][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]. Machine learning techniques use positive definite kernel functions to map the data into a reproducing kernel Hilbert space, and provide an elegant way to learn a nonlinear system without the need of the exact physical model [2][6]. In several applications as in industrial systems, the training set refer to a unique/single class while the data from the other classes are difficult to obtain. This one-class classification problem has gained the interest of the researchers in the past decade, where the classifier must accept as many positive samples (target class) and reject as many outliers as possible [7].

This work is supported by the French "Agence Nationale de la Recherche"(ANR), grant SCALA.

The authors would like to thank Thomas Morris and the Mississippi state university SCADA Laboratory for providing the real datasets.

One-class classification algorithms have been applied in many fields, namely for face recognition applications [8], mobile masquerades detection [9], seizure analysis from EEG signals [10], and recently for intrusion detection in industrial systems [11]. Several approaches exist in the literature for one-class classification problems. Tax *et al.* introduced in [12][13] the Support Vector Data Description (SVDD) which estimates the hypersphere with minimum radius enclosing most of the training data. The drawbacks of the SVDD are that it requires to solve a quadratic problem, and it does not take into consideration the variance of the training data in each feature direction. Tax proposed a kernel whitening normalization in [14] to overcome the variance drawback, while the quadratic problem remains unchanged. Schölkopf *et al.* defines in [15] the hyperplane separating the mapped data from the origin with maximum margin. This approach, called the one-class Support Vector Machines (SVM), is equivalent to the SVDD when the Gaussian kernel is used and has the same drawbacks of variance inequality in each feature direction. Kernel Principal Component Analysis (KPCA) is introduced in [16] for several applications, and Hoffman used in [17] the KPCA for novelty detection. The implementation of the KPCA is faster than the SVDD, but its computational complexity is cubic with the size of the training dataset. Tsang *et al.* used in [18] the covariance information to learn the kernel in one-class SVM, but it requires the optimization of a second order cone programming (SOCP) problem. Wang *et al.* proposed in [19] another approach that uses the Mahalanobis distance in the feature space for SVDD. This approach is also a SOCP problem, and its complexity is cubic with the size of the training dataset.

A first attempt for a fast and a simple one-class approach is introduced in [20], and it computes the Euclidean distance in the feature space between the samples and the center of the data. This approach is faster than the aforementioned methods, but it is very sensitive to outliers. In this paper, we propose a one-class classification approach based on the use of the Mahalanobis distance in the feature space. The Mahalanobis distance takes into account the covariance in each feature direction and the different scaling of the coordinate axes [21]. We take advantage of the properties of KPCA [17] and kernel whitening [14] to project the data into the subspace

spanned by the most relevant eigenvectors having the same variance in each feature direction. The Mahalanobis distance is computed in this subspace, and it is used for novelty detection. Then, we propose a sparse formulation of this approach, where only a very small fraction of the training data are taken into account in order to obtain the decision boundary of the classifier. The remainder of this paper is organized as follows. Section 2 describes the proposed one-class approach. Section 3 discusses the results on simulated and real datasets. Section 4 provides conclusion and future works.

2. THE PROPOSED APPROACH

In this paper, we propose a one-class classification approach based on the Mahalanobis distance in the feature space. The main objective for using the Mahalanobis distance is that it is a multivariate dissimilarity that takes into account the covariance in each feature direction and the different scaling of the coordinate axes. The proposed approach and its sparse formulation are discussed next.

2.1. Mahalanobis-based one-class

Let's consider a training dataset $\boldsymbol{x}_i$, $i = 1, \ldots, N$ in a d-dimensional input space $\mathcal{X}$. The data are mapped into a Reproducing Kernel Hilbert Space $\mathcal{H}$ via the mapping function $\phi(\boldsymbol{x}_i) = k(\boldsymbol{x}_i, \cdot)$, and the mapping is applied in a way to use only the pairwise inner product between the data. The inner product is computed directly from the input data using a kernel function, and the entries of the $N \times N$ kernel matrix $\boldsymbol{K}$ take this form: $k_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}}$ for $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathcal{X}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ represents the dot-product in the RKHS. This property is known as the kernel trick, and it allows to construct classification algorithms in inner product spaces without any explicit knowledge of the mapping function ϕ.

The simple one-class approach consists on computing the distance in the feature space between the training samples and the center of the data in that space. Based on this distance, a decision function classifies new samples as normal or outliers. In fact, the mean of the mapped data is given by the expectation of the data in the feature space, namely $\mathrm{E}[\phi(\boldsymbol{x})]$. One can estimate this expectation by the empirical computation of the center of the data in the feature space, with $c_n = \frac{1}{N} \sum_{i=1}^{N} \phi(\boldsymbol{x}_i)$.

The Mahalanobis distance between a sample $\phi(\boldsymbol{x})$ and the center c_n is given as follows:

$$\big\|\phi(\boldsymbol{x}) - c_n\big\|^2_{\boldsymbol{\Sigma}} = \big(\phi(\boldsymbol{x}) - c_n\big)^T \boldsymbol{\Sigma}^{-1} \big(\phi(\boldsymbol{x}) - c_n\big), \quad (1)$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the data in the feature space, namely $\boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\phi(\boldsymbol{x}_i) - c_n)(\phi(\boldsymbol{x}_i) - c_n)^T$. Without any explicit knowledge on the mapping function $\phi(\cdot)$, the covariance matrix cannot be expressed in terms of the data $\phi(\boldsymbol{x})$ in the feature space. To overcome this problem, we use the singular value decomposition of the covariance matrix $\boldsymbol{\Sigma}$, 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 eigenvalues λ_k for $k = 1, 2, \cdots, N$. Since $\boldsymbol{V}$ is an orthogonal matrix, $\boldsymbol{\Sigma}^{-1}$ can be expressed as follows:

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{V}^T \boldsymbol{D}^{-1} \boldsymbol{V} \quad (2)$$

Each eigenvalue λ_k satisfies $\lambda_k \boldsymbol{v}^k = \boldsymbol{\Sigma} \boldsymbol{v}^k$. From the definition of the matrix $\boldsymbol{\Sigma}$, it is easy to see that each eigenvector is a linear combination of the samples $\phi(\boldsymbol{x}_i)$ in the feature space, namely $\boldsymbol{v}^k = \sum_{i=1}^{N} \alpha_i^k \big(\phi(\boldsymbol{x}_i) - c_n\big)$. 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 $N \lambda_k \boldsymbol{\alpha}^k = \widetilde{\boldsymbol{K}} \boldsymbol{\alpha}^k$, where the kernel matrix $\widetilde{\boldsymbol{K}}$ is the centered version of $\boldsymbol{K}$.

Next, equation (1) takes this form $\big\|\phi(\boldsymbol{x}) - c_n\big\|^2_{\boldsymbol{\Sigma}} = \boldsymbol{a}^T \boldsymbol{a}$ having: $\boldsymbol{a} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{V} \big(\phi(\boldsymbol{x}) - \frac{1}{N} \sum_{i=1}^{N} \phi(\boldsymbol{x}_i)\big)$, where each entry a_k of $\boldsymbol{a}$ is associated to an eigenvector $\boldsymbol{v}^k$, with:

$$\begin{aligned} a_k &= \lambda_k^{-\frac{1}{2}} \Big(\sum_{i=1}^{N} \alpha_i^k k(\boldsymbol{x}_i, \boldsymbol{x}) - \frac{1}{N} \sum_{i,j=1}^{N} \alpha_i^k k(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\quad - \sum_{i=1}^{N} \alpha_i^k \frac{1}{N} \sum_{j=1}^{N} k(\boldsymbol{x}_j, \boldsymbol{x}) + \sum_{i=1}^{N} \alpha_i^k \frac{1}{N^2} \sum_{j,j'=1}^{N} k(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) \\ &= \lambda_k^{-\frac{1}{2}} \sum_{i=1}^{N} \alpha_i^k \widetilde{k}(\boldsymbol{x}_i, \boldsymbol{x}). \end{aligned}$$

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

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

Finally, the Mahalanobis distance in equation (1) is computed in the feature space as follows:

$$\big\|\phi(\boldsymbol{x}) - c_n\big\|^2_{\boldsymbol{\Sigma}} = \sum_{k=1}^{N} \lambda_k^{-1} \Big(\sum_{i=1}^{N} \alpha_i^k \widetilde{k}(\boldsymbol{x}_i, \boldsymbol{x}) \Big)^2. \quad (3)$$

After calculating the Mahalanobis distance in the feature space between each training sample $\phi(\boldsymbol{x}_i)$ and the center c_n, and knowing the number of outliers n_{out} in the training dataset, we fix a threshold R which represents the radius in that space. The decision function of our classifier considers a new sample as an outlier if its Mahalanobis distance in the feature space is greater than this threshold, namely $\big\|\phi(\boldsymbol{x}_i) - c_n\big\|^2_{\boldsymbol{\Sigma}} > R$. Otherwise, the sample is considered as a normal sample.

2.1.1. Advantages of kernel whitening and KPCA

As detailed in the previous section and in equation (3), the Mahalanobis distance in the feature space is computed via

the projection of the data into the subspace spanned by the eigenvectors of the covariance matrix $\mathbf{\Sigma}$. Instead of using all the eigenvectors $\boldsymbol{v}^k$ for the projection, we make use of the advantages in the KPCA approach, where only the most relevant eigenvectors whose correspondent eigenvalues satisfy $\lambda_k \neq 0$ are taken into consideration. The remaining eigenvectors are considered as noise.

We also adopt the kernel whitening normalization of the eigenvectors as proposed in [14], where the variance of the mapped data is constant for all feature directions. The normalization is achieved as follows:

$$(N\lambda_k)^2\|\boldsymbol{\alpha}^k\|^2 = 1 \Longrightarrow \|\boldsymbol{\alpha}^k\| = \frac{1}{N\lambda_k} \text{ for all components k.}$$

2.1.2. Theoretical results

Given a training dataset $\boldsymbol{x}_i$, $i = 1, \ldots, N$ in a d-dimensional input space with its covariance matrix $\mathbf{\Sigma}$, and let $\mathcal{P}$ be the projection operator onto the subspace spanned by the k most relevant eigenvectors $\boldsymbol{v}^k$.

Theorem 1 *The error of projecting the center of the data c_n onto this subspace 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 the center is expressed as follows:

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

where the first inequality follows from the generalized 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 5).

2.2. Sparse Mahalanobis-based one-class

We consider a sparse formulation of the aforementioned proposed method, especially for large training datasets, in order to reduce the computational complexity of the algorithm, and to maintain a good description boundary around the data. We propose to approximate the center of the dataset c_n using the furthest samples which are known as the support vectors. The sparse center is a linear combination of the support vectors, and only these samples are taken into account in the calculation of the Mahalanobis distance. Then we define a threshold based on the predefined number of outliers, in order to discriminate new samples as normals or outliers.

Many approaches have been proposed in the literature for the choice of the support vectors, based on the coherence criterion [22] or on the distance criterion [20]. The coherence is defined by the largest absolute values of the off-diagonal entries of the kernel matrix, namely $\max_{i,j,i\neq j}|k(\boldsymbol{x}_i,\boldsymbol{x}_j)|$, and the least coherence set is considered as the relevant set. The distance criterion lies on the computation of the Euclidean distance in the feature space between the samples and the center of the data, and the set $\mathcal{I}$ containing the furthest samples to the center: $\mathcal{I} = \{\boldsymbol{x}_i, \big\|\phi(\boldsymbol{x}_i) - c_n\big\|_2^2 > R^2\}$, where R is the radius/threshold based on the predefined number of outliers. We note that regardless of the approach used for selecting the support vectors, the following algorithm remains unchanged.

We adopt the distance criterion approach, after adapting it to the computation of the Mahalanobis distance in the feature space as detailed in the previous section. The sparse center takes this form: $\boldsymbol{c}_{\mathcal{I}} = \sum_{i\in\mathcal{I}}\beta_i\phi(\boldsymbol{x}_i)$, and the coefficients β_i are computed by minimizing the error of approximating the center $\boldsymbol{c}_n$ with the sparse center $\boldsymbol{c}_{\mathcal{I}}$:

$$\arg\min_{\beta_i}\Big\|\frac{1}{N}\sum_{i=1}^{N}\phi(\boldsymbol{x}_i) - \sum_{i\in\mathcal{I}}\beta_i\phi(\boldsymbol{x}_i)\Big\|_{\mathbf{\Sigma}}^2.$$

The partial derivative of this cost function with respect to each β_i is computed and set to zero, and for each eigenvector $\boldsymbol{v}^k$ we have:

$$\begin{aligned}&\lambda_k^{-\frac{1}{2}}\Big(\frac{1}{N}\sum_{i=1}^{N}\sum_{k\in\mathcal{I}}k(\boldsymbol{x}_i,\boldsymbol{x}_k) - \sum_{j,k\in\mathcal{I}}\beta_j k(\boldsymbol{x}_j,\boldsymbol{x}_k)\Big)\times \\ &\Big(\sum_{i=1}^{N}\alpha_i^k\Big(\phi(\boldsymbol{x}_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(\boldsymbol{x}_i)\Big)\Big) = 0,\end{aligned}$$

where the eigenvectors $\boldsymbol{v}^k$ and the eigenvalues λ_k already satisfy $\boldsymbol{v}^k \neq \mathbf{0}$ and $\lambda_k \neq 0$. This boils down to the following:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{k\in\mathcal{I}}k(\boldsymbol{x}_i,\boldsymbol{x}_k) = \sum_{j,k\in\mathcal{I}}\beta_j k(\boldsymbol{x}_j,\boldsymbol{x}_k).$$

The coefficients β_i are computed through the matrix notation:

$$\boldsymbol{\beta} = \boldsymbol{K}_{\mathcal{I}}^{-1}\boldsymbol{k},$$

where the entries of the kernel matrix $\boldsymbol{K}_{\mathcal{I}}$ are $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ for $i,j \in \mathcal{I}$, and $\boldsymbol{k}$ is the column vector with entries $\sum_{k\in\mathcal{I}}k(\boldsymbol{x}_i,\boldsymbol{x}_k)$. In order to avoid non-invertible singular matrix $\boldsymbol{K}_{\mathcal{I}}$, we include a regularization parameter ν, namely $\boldsymbol{\beta} = (\boldsymbol{K}_{\mathcal{I}} + \nu\boldsymbol{I})^{-1}\boldsymbol{k}$.

The classifier fixes a threshold R based on the predefined number of outliers n_{out}. The decision function for testing a

new sample is to measure the Mahalanobis distance between this sample and the sparse center $c_{\mathcal{I}}$ as follows:

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

If this distance is greater than the radius R^2, the sample is considered as an outlier. Otherwise, it is considered as a normal sample.

2.2.1. Theoretical results

Let c_∞ denote the expectation of the data in the feature space, namely $\mathrm{E}[\phi(\boldsymbol{x})]$, and ϵ_0 the projection error between c_∞ and c_n, namely $\epsilon_0 = \|\mathcal{P}c_n - \mathcal{P}c_\infty\|$. The samples of the training dataset are generated from the same distribution, the set $\mathcal{I}$ represents the set of support vectors, and n_{out} the number of outliers among this dataset.

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

$$\mathrm{P}(\|\mathcal{P}\phi(\boldsymbol{x})-\mathcal{P}c_{\mathcal{I}}\|>R+2\epsilon_0+2\|\mathcal{P}c_n-\mathcal{P}c_{\mathcal{I}}\|)\leq\frac{n_{out}}{N+1}.$$

Proof When all the training samples are inside the sphere centered on c_n, it has been shown in [2] that the probability of a new sample $\boldsymbol{x}_{N+1}$ that lies outside this description is bounded by

$$\mathrm{P}(\|\phi(\boldsymbol{x}_{N+1})-c_n\|>R_1+2\epsilon_1\|)\leq\frac{1}{N+1},$$

having ϵ_1 the error of approximating c_∞, and R_1 the radius of the sphere, namely $R_1 = \max_{i=1,\cdots,N}\|\phi(\boldsymbol{x}_i)-c_n\|$. If we consider the sphere centered on the projected sparse center $\mathcal{P}c_{\mathcal{I}}$, and the distance between the projected sample $\mathcal{P}\phi(\boldsymbol{x})$ and $\mathcal{P}c_{\mathcal{I}}$, we apply the triangular inequality twice and we get the following relations:

$$\begin{aligned}\|\mathcal{P}\phi(\boldsymbol{x})-\mathcal{P}c_{\mathcal{I}}\| &\leq \|\mathcal{P}\phi(\boldsymbol{x})-\mathcal{P}c_n\|+\|\mathcal{P}c_n-\mathcal{P}c_{\mathcal{I}}\|\\ &\leq \|\mathcal{P}\phi(\boldsymbol{x})-\mathcal{P}c_\infty\|+\epsilon_0+\|\mathcal{P}c_n-\mathcal{P}c_{\mathcal{I}}\|,\end{aligned}$$

and

$$\begin{aligned}\|\mathcal{P}\phi(\boldsymbol{x})-\mathcal{P}c_{\mathcal{I}}\| &\geq \|\mathcal{P}\phi(\boldsymbol{x})-\mathcal{P}c_n\|-\|\mathcal{P}c_n-\mathcal{P}c_{\mathcal{I}}\|\\ &\geq \|\mathcal{P}\phi(\boldsymbol{x})-\mathcal{P}c_\infty\|-\epsilon_0-\|\mathcal{P}c_n-\mathcal{P}c_{\mathcal{I}}\|.\end{aligned}$$

From these two inequalities, and by the symmetry of the i.i.d assumption, the probability of a new sample $\boldsymbol{x}_{N+1}$ lying outside this distribution is bounded by

$$\mathrm{P}(\|\mathcal{P}\phi(\boldsymbol{x}_{N+1})-\mathcal{P}c_{\mathcal{I}}\|>R+2\epsilon_0+2\|\mathcal{P}c_n-\mathcal{P}c_{\mathcal{I}}\|)\leq\frac{n_{out}+1}{N+1}.$$

3. EXPERIMENTAL RESULTS

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 [23], 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.

The proposed algorithms are applied in the first place on two simulated datasets, the sinusoidal and the square noise datasets [17]. We compared the results with three other one-class classification approaches, namely SVDD, KPCA and simple one-class as shown in figure 1. The proposed algorithms have the best results with a very tight description boundary around the data, the KPCA gives a good description, while the presence of outliers led to loose decision boundaries with the simple one-class and the SVDD. We note that the sparse approach used only 15% of the training data to define these tight boundaries.

The proposed algorithms are now applied on two real datasets, the gas pipeline and the water storage tank from the Mississipi State University SCADA Laboratory [24]. The gas pipeline is used to move natural gas or other petroleum products to market, it is connected to an air pump which pumps air into the pipeline, and contains valves to release the air pressure from it. A pressure sensor is attached to the pipeline which allows pressure visibility at the pipeline and remotely on a Human Machine Interface screen. The water storage tank testbed is similar to the oil storage tanks found in the petrochemical industry. It contains primary and secondary storage tanks, a pump to move water from the secondary to the primary tank, a relieve valve which allows water to flow from the primary to the secondary tank, and a sensor which provides the water level in the primary tank. Each input sample has 27 attributes for the gas pipeline and 24 attributes for the water storage tank. The attributes represent heterogenous variables, such as gas pressure, water level, pump state, target gas pressure/water level, valve state, PID's parameters, time interval between packets, device ID in command/response packets, and length of command/response packets. The diversity in the attributes requires a high performing classifier capable of correctly discriminating between normal samples and outliers. 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).

The error detection probabilities of the studied approaches for the gas pipeline and the water storage testbeds are given in tables 1 and 2, and the estimated time for each approach as well as the time to test each new sample are computed in tables and 4 and 3. These results show that the simple one-class approach is the fastest one, but with poorer results since it is very sensitive to the presence of outliers among the training dataset. The proposed approach gives better detection rates than the others for different types of attacks, expect for the MSCI and MFCI attacks on the gas pipeline. It is important to draw attention to the fact that only 10% of the training dataset are used in the sparse approach to define the description boundary, and this hardly affected the results that are still better than SVDD and KPCA. In addition, the proposed algorithms take almost the same computational time, and they are twice faster than KPCA and much more than SVDD. Furthermore, our approaches are the fastest regarding the time needed to test a new sample, and the spare approach is almost 10 times faster than SVDD. These results are very important if we want to apply our algorithms in real-world scenarios, where the sparse approach can process over 200 samples per second, compared to only 25 samples for SVDD and 50 samples for KPCA.

Table 1. Error detection probabilities of several approaches for the gas pipeline testbed.

	SVDD	KPCA	simple 1-class	**proposed 1-class**	**sparse 1-class**
NMRI	98.1	98.7	91.7	**99.6**	99.3
CMRI	99.5	**99.8**	95.4	**99.8**	**99.8**
MSCI	**89.1**	86.2	22.6	83.1	81.1
MPCI	98.2	98.6	94.1	**99.1**	**99.1**
MFCI	**89.9**	89.3	31.6	85.1	85.4
DOS	96.1	96.8	68.51	**97.7**	96.7
RA	**99.8**	**99.8**	98.1	**99.8**	**99.8**

Table 2. Error detection probabilities of several approaches for the water storage testbed.

	SVDD	KPCA	simple 1-class	**proposed 1-class**	**sparse 1-class**
NMRI	95.1	97.1	88.2	**98.8**	98.5
CMRI	61.2	75.3	46.2	**82.4**	80.1
MSCI	97.3	98.1	96.3	**98.7**	98.4
MPCI	98.6	99.5	97.6	**99.7**	99.6
MFCI	97.9	**99.9**	40.6	**99.9**	**99.9**
DOS	71.7	79.9	55.3	**83.3**	80.6
RA	97.8	99.5	95.9	**99.7**	**99.7**

Table 3. Estimated time (in seconds) of each approach on the gas pipeline and the water storage testbeds.

	SVDD	KPCA	simple 1-class	**proposed 1-class**	**sparse 1-class**
gas	70.23	18.31	**9.23**	10.08	10.21
water	123.72	20.1	**10.41**	11.89	12.02

Table 4. Estimated time (in seconds) to test a new sample for each approach on the gas pipeline and the water storage testbeds.

	SVDD	KPCA	simple 1-class	**proposed 1-class**	**sparse 1-class**
gas	0.039	0.019	0.011	0.010	**0.0047**
water	0.043	0.032	0.015	0.019	**0.0051**

4. CONCLUSION

In this paper, we investigated the use of the Mahalanobis distance in the feature space for one-class classification problems. We proposed a one-class method based on this distance, and a sparse formulation of this approach. We tested the algorithms on simulated data and on real data containing several types of attacks. The proposed approach achieved the best description boundaries on the simulated data, and the highest error detection rates with minimum computational costs on the real datasets. These results proved that the use of the Mahalanobis distance as a novelty measure has increased the performance of our classifier, since it takes into account the covariance among the variables and the different scaling of the coordinate axes.

For future works, a further and more detailed study on the effect of using the Mahalanobis distance on the stability of the studied system is required. Furthermore, we should consider to integrate our approach in the traditional intrusion detection systems in industrial infrastructures, since these approaches could play an important and complementary role to the IDS in detecting malicious attacks on physical systems, and they have a high processing performance (over 200 samples per second).

5. REFERENCES

[1] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Annals of Statistics*, vol. 36, pp. 1171–1220, 2008.

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

[3] N. Bredeche, Z. Shi, and 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*, vol. 36, no. 2, pp. 172–181, March 2006.

[4] D. Strauss, W. Delb, J. Jung, and 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, April 2003, pp. VI–425–8 vol.6.

[5] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and 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, 16 - 19 June 2013, pp. 744–748.

[6] J. P. Vert, K. Tsuda, and B. Scholkopf, "A primer on kernel methods," *Kernel Methods in Computational Biology*, pp. 35–70, 2004.

[7] S. S. Khan and 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*, ser. AICS'09, 2010, pp. 188–197.

[8] Z. Zeng, Y. Fu, G. Roisman, Z. Wen, Y. Hu, and T. Huang, "One-class classification for spontaneous facial expression analysis," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, April 2006, pp. 281–286.

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

[10] A. B. Gardner, A. M. Krieger, G. Vachtsevanos, and B. Litt, "One-class novelty detection for seizure analysis from intracranial eeg," *Journal of Machine Learning Research*, vol. 7, pp. 1025–1044, 2006.

[11] P. Nader, P. Honeine, and P. Beauseroy, "Intrusion detection in scada systems using one-class classification," in *Proc. 21th European Conference on Signal Processing*, Marrakech, Morocco, 9–13 September 2013.

[12] D. M. J. Tax and R. P. W. Duin, "Data domain description using support vectors," in *Proceedings of the European Symposium on Artificial Neural Networks*, 1999, pp. 251–256.

[13] ——, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[14] D. M. J. Tax and P. Juszczak, "Kernel whitening for one-class classification," in *SVM*, 2002, pp. 40–52.

[15] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[16] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[17] H. Hoffmann, "Kernel pca for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863 – 874, 2007.

[18] I. Tsang, J. Kwok, and 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.

[19] D. Wang, D. Yeung, and E. C. C. Tsang, "Structured one-class classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 6, pp. 1283–1295, Dec 2006.

[20] Z. Noumir, P. Honeine, and C. Richard, "On simple one-class classification methods," in *Proc. IEEE International Symposium on Information Theory*, MIT, Cambridge (MA), USA, 1–6 July 2012.

[21] P. C. Mahalanobis, "On the generalised distance in statistics," in *Proceedings National Institute of Science, India*, vol. 2, no. 1, Apr. 1936, pp. 49–55.

[22] Z. Noumir, P. Honeine, and C. Richard, "One-class machines based on the coherence criterion," in *Proc. IEEE workshop on Statistical Signal Processing*, Ann Arbor, Michigan, USA, 5–8 August 2012, pp. 600–603.

[23] P. Nader, P. Honeine, and P. Beauseroy, "l_p-norms in one-class classification for intrusion detection in scada systems," *Industrial Informatics, IEEE Transactions on*, minor revision 2014.

[24] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu, and R. Reddi, "A control system testbed to validate critical infrastructure protection concepts," *International Journal of Critical Infrastructure Protection*, vol. 4, no. 2, pp. 88 – 103, 2011.

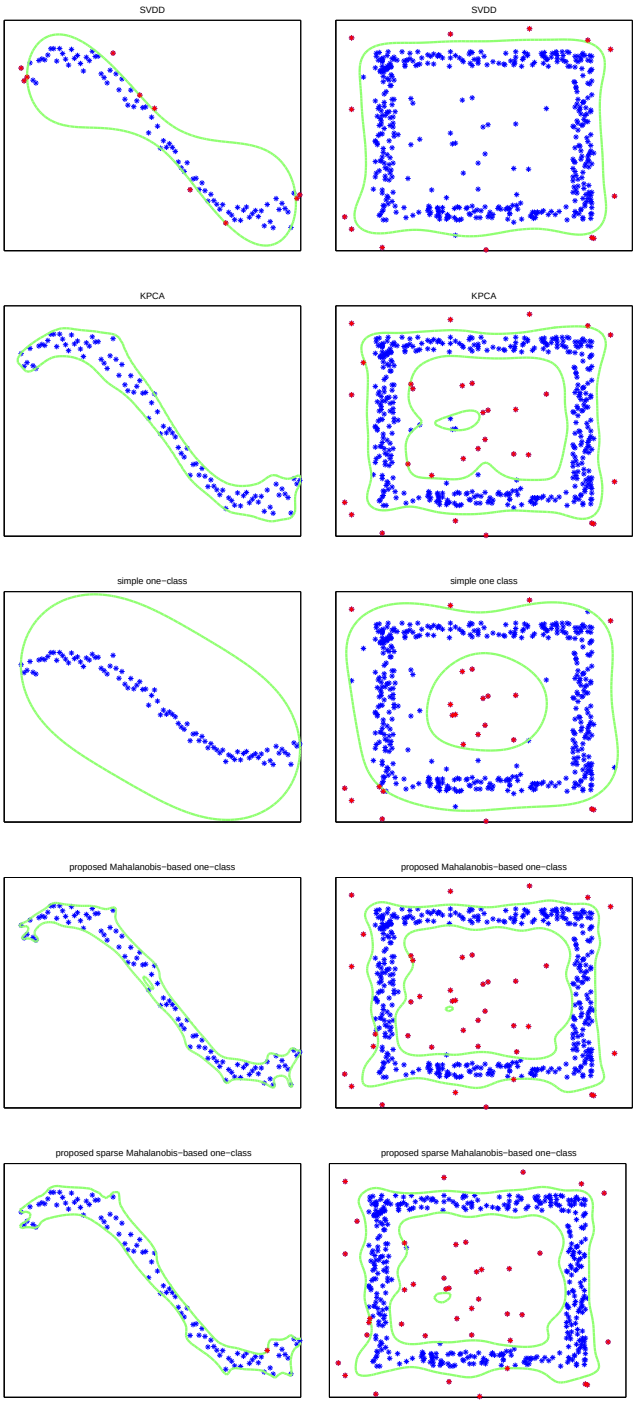

Fig. 1. The decision boundaries on the sinusoidal and the square-noise datasets for different approaches. The two figures at the bottom (in each column) represent the proposed approach.