Technical Report – Université de technologie de Troyes

On L^p-norms in one-class classification for intrusion detection in SCADA systems

Patric Nader, Paul Honeine, and Pierre Beauseroy

June 11, 2013

Abstract

The massive use of Information and Communication Technologies in Supervisory Control and Data Acquisition (SCADA) systems has opened new ways for carrying out cyberattacks against critical infrastructures relying on SCADA networks. The various vulnerabilities in these systems and the heterogeneity of cyberattacks has made the task more difficult for traditional Intrusion Detection Systems. Modeling cyberattacks has become nearly impossible, and their potential consequences may be very severe. This paper investigates the use of L^p-norms in Radial Basis Function kernels for intrusion detection in SCADA systems using one-class classification algorithms. Two approaches of one-class classification are investigated, the Support Vector Data Description and the Kernel Principle Component Analysis. A heuristic is proposed to find the optimal choice of the bandwidth parameter in RBF kernels. Tests are conducted on simulated data, and on real data containing several types of cyberattacks.

1 Introduction

Supervisory Control and Data Acquisition (SCADA) systems provide remote access an control to critical infrastructures such as electrical power grids, oil and natural gas pipelines, chemical processing plants, water distribution, wastewater collection systems and nuclear power plants [1]. The principal components of SCADA systems [2] are: a) The *Human Machine Interface (HMI)* allows operators to monitor the state of the process under control and modify its control settings, b) the *Master Terminal Unit (MTU)* stores and processes the information from the field and transmits control signals and c) *Remote Terminal Units (RTU)* receive commands from the MTU to control the local process, acquire data from the field and transmit it to the MTU. The common protocols (ModBus, Profibus, DNP3) used in the communication between these components present some vulnerabilities [3]. These protocols don't perform any authentication mechanism between Master and Slave, don't check for the integrity of the command packets and don't apply any anti-repudiation or anti-replay mechanisms.

In addition to the vulnerabilities in the communication between their components, SCADA systems are facing today significant threats of cyberattacks due to the increasing dependence of their communications to the internet [4]. Security threats can be grouped into three categories: Hackers, insiders and malwares [5]. The *hackers* can access SCADA networks, collect data flows and inject false commands with the intention to disrupt the physical system under control. *Insiders* are the personnel of the facility having a legitimate access to the network and may cause damages to the infrastructure. The *malwares* are viruses, worms, trojans and spywares that can affect the operating systems and the softwares of the facility.

The past decade has witnessed several intentional cyberattacks against critical infrastructures relying on SCADA networks. In 2000, an ex-employee of Maroochy Water Services in Australia took control of 150 sewage pumping stations and released one million liters of untreated sewage into local parks and

rivers [6]. In 2003, the Slammer worm penetrated a private computer network at Ohios Davis-Besse nuclear power plant and disabled a safety monitoring system for nearly five hours [7]. In 2006, a hacker penetrated a water filtering plant in Pennsylvania (USA) and planted malicious software capable of affecting the plants water treatment operations [8]. In 2009, cyberspies have penetrated the U.S. electrical grid and left behind software programs that could be used to disrupt the system [9]. The most sophisticated malware Stuxnet installs a malicious program replacing the PLCs original file in a manner undetectable by the PC operator [10]. Stuxnet was discovered in Iran in June 2010 targeting the PLCs connected to a nuclear centrifuge used for enriching uranium. The speed fluctuations could cause the centrifuge to fly apart and to be destroyed[11].

The diversity of cyberattacks and the complexity of the studied systems make the role of traditional Intrusion Detection Systems (IDS) more difficult. Traditional IDS try to match signatures of known cyberattacks with network traffic, but they cannot detect new types of cyberattacks not existing in their databases [12]. Gross *et al.* proposed in [13] a mechanism for collaborative intrusion detection using a centralized server to dispatch activities coming from suspicious IP addresses. However, this approach do not provide any kind of specific technique for identifying high level and complex cyberattacks. Carano *et al.* presented in [14] the concept of critical state analysis for the detection of a particular type of cyberattacks against a given industrial installation. They emulated in their laboratory the *Boiling Water Reactor Scenario* and used the concept of "critical state proximity" to predict whether the system is heading to a dangerous state. This approach focuses on the restrictive assumption that the attacker interferes with the state of the installation forcing a transition from a safe state to a critical one. Morris *et al.* investigate in [15] [16] the vulnerabilities of functional control systems. They elaborated in the Mississippi State University Laboratory a SCADA testbed including commercial hardware and software that control physical processes such as a gas pipeline, an industrial blower, a smart grid transmission control system, a raised water tower and a factory conveyor belt. False commands and responses were

injected into the SCADA network in order to investigate cybersecurity vulnerabilities on functional control systems. Cyberattacks studied in their testbed include command injection attack, response injection attack and denial of service (DOS) attack. The diversity of these types of cyberattacks restricts the use of parametric model-based approach, and highlights the potential role of non-parametric model-based methods in detecting intrusions.

Statistical machine learning, kernel methods and classification techniques have been widely used in the past few years in the data mining field to discover hidden regularities in large volumes of data [17]. Machine learning and classification techniques adapt quickly to different types of data, and they provide an elegant way to learn a nonlinear system without the need of an exact physical model. In industrial systems, the majority of the data designates the normal functional mode, and it is nearly impossible to acquire data related to the malfunctioning or critical states [18]. For this reason, the role of one-class classification has been growing in detecting machine faults and intrusions, especially in critical infrastructures and industrials systems. To the best of our knowledge, machine learning has not been investigated for SCADA systems.

The main problem involved with using radial basis functions in machine learning is the choice of the bandwidth parameter σ of the kernel. Haykin proposed in [19] a heuristic for computing σ according to the spread of the centers in neural networks. However, this approach is not applicable in the classification methods since the number of support vectors is not known in advance. Shi *et al.* set in [20] the value of σ between 10 to 20 percent of the total range of the maximum distance between training samples, but this range works on some cases in image segmentation only. Soares *et al.* proposed in [21] a grid-search of 11 values for σ following a geometric series of factor 4. However, this cross-validation technique remains the most expensive in time consumption and does not always lead to the optimum choice of σ in classification problems. Cherkassky *et al.* suggested in [22] a more restricted range depending on the input data and faster than the grid-search, but with poorer results. Evangelista *et al.* introduced in [23] the notion of the coefficient of variance in order to find

the optimal σ, but it did not give the optimal performance on several simulated data. Recently, Gurram *et al.* proposed in [24] a two-step iterative heuristic using a gradient descent algorithm to optimize the bandwidth parameter of Gaussian RBF kernels. However, the convergence of the algorithm proposed is not guaranteed and the computational requirements are still important.

This paper investigates the use of L^p-norms in Radial Basis Function kernels for intrusion detection in SCADA systems using one-class classification algorithms. Two distinct approaches are investigated, the Support Vector Data Description (SVDD) [25] and Kernel Principal Component Analysis (KPCA) [26]. In each approach, the description boundary of the normal behavior of the system is found, and the one-class classifier discriminates the data between normal or abnormal, and accordingly outliers are detected. The tests are conducted on simulated data, and on the *Gas Pipeline testbed* real data from the Mississippi State University SCADA Laboratory [16]. We also propose a heuristic for the optimization of the bandwidth parameter in RBF kernels. The remainder of this paper is organized as follows. Section 2 provides an overview on kernels, delineates the RBF kernels used in the simulations, and proposes a heuristic for the parameter optimization problem. Section 3 outlines kernel methods for one-class classification, namely the SVDD and the KPCA. Section 4 describes the gas pipeline testbed, the results on simulated data as well as on the gas pipeline data. Section 5 provides conclusion and future works.

2 Kernel methods

Machine learning techniques have been well developed for linear case problems, while real world data analysis problems require, most of the time, nonlinear methods for detecting patterns and interdependencies between the data [17] [27]. Kernel methods have become very popular in the past few years since they provide a powerful way for detecting nonlinear relations using linear algorithms in the feature space [28] [29]. An example of the feature mapping using a Gaussian kernel is illustrated in figure 1, where the input data embedded in the

feature space lay on a sphere with radius equals to 1. The mapping is applied in such a way that only the pairwise inner product between the embedded data is needed. This inner product is computed directly from the input data using a kernel function.

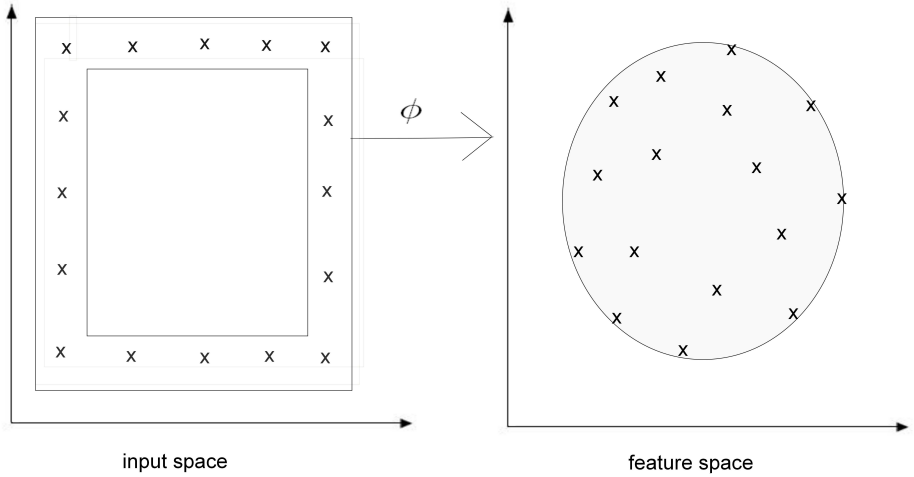

Figure 1: The feature mapping into the inner product feature space using a Gaussian kernel. The embedded data in the feature space lay on a sphere with radius equals to 1.

A *kernel* function k is a function that for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$ satisfies

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle,$$

where ϕ is the mapping from a nonempty input domain $\mathcal{X}$ into a feature space $\mathcal{H}$:

$$\phi \colon \boldsymbol{x}_i \in \mathcal{X} \rightarrow \phi(\boldsymbol{x}_i) \in \mathcal{H}.$$

Kernel methods use positive definite kernel functions for the mapping into a high dimensional feature space. A function k is called positive definite kernel if and only if it is *symmetric*, that is $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i)$ for any two samples $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$, and *positive definite* , that is :

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0,$$

for any choice of n objects $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathcal{X}$, and any choice of real numbers $c_1, \cdots, c_n \in \mathcal{R}$. The kernel matrix constructed on $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n \in \mathcal{X}$ is a n x n matrix $\boldsymbol{K}$ whose entries are computed as $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The kernel matrix contains all the information needed to compute the pairwise distances within the dataset, and plays an important role in the learning algorithms.

The feature space is an inner product *Hilbert Space* that is complete and separable, and satisfies the symmetry, the bilinearity and the positive definiteness conditions. The feature space $\mathcal{H}$ is a set of points that are in fact functions taking the following form:

$$\mathcal{H} = \Big\{ \sum_{i=1}^{l} \alpha_i k(\boldsymbol{x}_i, \cdot), \quad \boldsymbol{x}_i \in \mathcal{X}, \alpha_i \in \mathcal{R}, i = 1, \cdots, l \Big\},$$

where the $\cdot$ indicates the position of the argument of the function. Let $f, g \in \mathcal{H}$ be given by

$$f(x) = \sum_{i=1}^{l} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) \quad \text{and} \quad g(x) = \sum_{j=1}^{n} \beta_j k(\boldsymbol{x}_j, \boldsymbol{x}),$$

then the inner product on $\mathcal{H}$ is constructed as follows:

$$\langle f, g \rangle = \sum_{i=1}^{l} \sum_{j=1}^{n} \alpha_i \beta_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{i=1}^{l} \alpha_i g(\boldsymbol{x}_i) = \sum_{j=1}^{n} \beta_j f(\boldsymbol{x}_j), \tag{1}$$

where $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$, $\alpha_i, \beta_j \in \mathcal{R}$, $l, n \in \mathcal{N}$, and the second and the third equalities come from the definition of f and g. Taking $g = k(\boldsymbol{x}, \cdot)$ and computing the the inner product on $\mathcal{H}$ between f and g using equation (1) gives us the following property:

$$\langle f, k(\boldsymbol{x}, \cdot) \rangle = \sum_{i=1}^{l} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) = f(\boldsymbol{x}).$$

This property is know as the *reproducing property* of the kernel. Therefore, the feature space $\mathcal{H}$ corresponding to the kernel function k satisfying the positive definite property will be referred to as its *Reproducing Kernel Hilbert Space*

(RKHS). In fact, k satisfies the positive definite property since:

$$\begin{aligned}\sum_{i,j=1}^{l} \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \sum_{i,j=1}^{l} \alpha_i \alpha_j \langle k(\boldsymbol{x}_i, \cdot), k(\boldsymbol{x}_j, \cdot) \rangle_{\mathcal{H}} \\ &= \left\langle \sum_{i=1}^{l} \alpha_i k(\boldsymbol{x}_i, \cdot), \sum_{j=1}^{l} \alpha_j k(\boldsymbol{x}_j, \cdot) \right\rangle_{\mathcal{H}} \\ &= \left\| \sum_{i=1}^{l} \alpha_i k(\boldsymbol{x}_i, \cdot) \right\|_{\mathcal{H}}^2 \geq 0,\end{aligned}$$

where $\| \cdot \|_{\mathcal{H}}$ represents the corresponding distance in the feature space $\mathcal{H}$. The advantage of using such a kernel is that it allows to construct classification algorithms in inner product spaces without computing the coordinates of the data in that space, and therefore without any explicit knowledge of the mapping function ϕ. This key idea is known as the kernel trick, for it can be used to transform linear algorithms expressed only in terms of inner products into nonlinear ones.

The first kernel investigated in this paper is the Gaussian kernel, since it is the most common and suitable kernel for one-class classification problems [30][31]. The Gaussian kernel is given by the following expression:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(- \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2s^2} \right),$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two input samples, $\| \cdot \|$ represents the Euclidean distance between the samples in the input space, and the free parameter to be optimized s is the bandwidth of the kernel. The second RBF kernel adopted in this paper is the exponential kernel that follows a Laplace distribution:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(- \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{b} \right),$$

where b is a scale parameter depending on the standard deviation σ:

$$\sigma = \sqrt{2b^2} \quad \longrightarrow \quad b = \frac{\sigma}{\sqrt{2}}.$$

The bandwidth parameter should be chosen wisely to obtain the best description that fits correctly the data and avoids overfitting. The proposed heuristic

for the optimization of the bandwidth parameter is detailed in the next subsection. Since we are working on industrial processes, the value of each variable is important to evaluate the criticality of the system, and it is very essential to predict whether the process is leading to a critical state. Therefore, we need more adapted kernels that treat simultaneous small changes in several features as much important as large variations in a single one. We propose to replace the Euclidean norm in these RBF kernels with other norms in order to study the effects of these norms on the decision function of the classifiers. For instance, the *City-block distance* or $l1$-norm that measures how close are the samples in each direction of the input space is given by:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_m |x_{im} - y_{im}|.$$

Figure 2 illustrates the variation in the behavior of different norms, where p takes one the values $0.75, 1, 1.25, 1.5, 1.75, 2$ and $infinite$. Each color represents equidistant contours with reference to the origin O. It is obvious from this figure that each norm operates differently on simultaneous variation of multiple feature values. If we consider the $l2$-norm (euclidean distance) for example, a large variation in the value of feature 2 has a much greater effect than simultaneous variation of feature 1 and feature 2; The samples B and C are equidistant from the origin O, whereas A is much further. However, for the $l1$-norm (City-block distance), C and D are equidistant and much closer than B, and a simultaneous small change in several features is as important as large variation in a single one. Therefore, the norms with a small value of p are particularly sensitive on simultaneous variation of multiple feature, while the ones with higher p are more sensitive to large variations in any single feature.

3 One-class classification

In multi-class classification problems, the decision boundary of the classifier is supported by the presence of samples from each class, and the multi-class algorithms are designed to classify unknown samples into one of several pre-defined

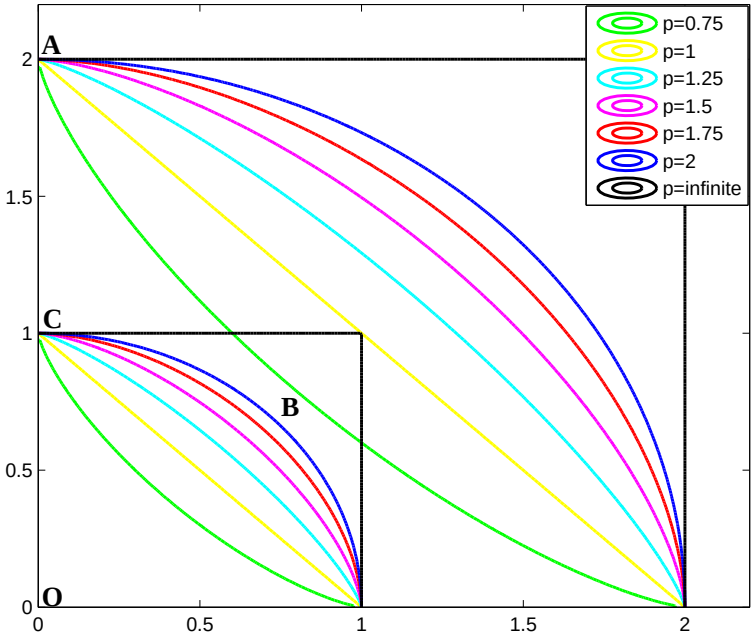

Figure 2: the City-block distance kernel (bold lines) is particularly sensitive with simultaneous small changes in several features (sample B), whereas the Gaussian kernel (dashed lines) is more sensitive to large variations in any single feature (sample A).

classes [18]. However, when it comes to industrial processes and detecting machine faults and intrusions, the data related to the malfunctioning modes are nearly impossible to acquire. This is the reason why researchers have been developing in the last few years one-class classification algorithms for novelty detection. One-class classification methods define a description boundary around the positive data (normal data) in a way to accept as many samples as possible from the positive class, and to minimize the chance of accepting negative samples (outliers). One-class classification algorithms are applied on training data in the feature space, and a decision function tests new samples to classify them

as normal data or outliers.

3.1 Support Vector Data Description

Support Vector Data Description (SVDD) was introduced by Tax *et al.* [25] in order to get a good description around a training dataset. Tax made this method more robust against outliers when he included negative examples(data that should be rejected) in the training set [32]. SVDD computes a spherically shaped decision boundary with minimum radius enclosing most of the training data. Samples that lay outside this description are considered outliers, and they should be rejected.

Given a training dataset $\boldsymbol{x}_i$, $i \in \{1, \ldots, N\}$ in a p-dimensional space, the SVDD estimates the hypersphere with minimum radius that encompasses all data in the feature space $\mathcal{H}$. The hypersphere is characterized by its center $\boldsymbol{a}$ and its radius $R > 0$, and we minimize its volume by minimizing R^2. To avoid a large description that does not represent the data very well, the presence of outliers in the training set is allowed, and the slack variables $\xi_i \geq 0$ are introduced to penalize the excluded samples. This boils down to the following constrained minimization problem:

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

subject to

$$\|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|_{\mathcal{H}}^2 \leq R^2 + \xi_i \quad \text{and} \quad \xi_i \geq 0 \;\; \forall i = 1, ..., N \tag{3}$$

The predefined parameter ν regulates the trade-off between the volume of the hypersphere and the number of outliers. $\nu \in (0, 1)$ represents an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors (the support vectors refer to the data *on* and *outside* the boundary).

The Lagrangian of the above optimization problem is constructed by incor-

porating the constraints (3) into (2):

$$L = R^2 + C\sum_i^N \xi_i - \sum_i^N \gamma_i \xi_i \\ - \sum_i^N \alpha_i \Big\{ R^2 + \xi_i - (\parallel \phi(\boldsymbol{x}_i) \parallel^2 - 2\boldsymbol{a} \cdot \phi(\boldsymbol{x}_i) + \parallel \boldsymbol{a} \parallel^2) \Big\},$$

where $\alpha_i \geq 0$ and $\gamma_i \geq 0$ are the Lagrangian multipliers. The partial derivatives of the Lagrangian with respect to R, ξ_i and $\boldsymbol{a}$ give the following relations:

$$\begin{aligned} \frac{\partial L}{\partial R} = 0 \qquad &\longrightarrow \qquad \sum_i^N \alpha_i = 1 \\ \frac{\partial L}{\partial \xi_i} = 0 \qquad &\longrightarrow \qquad 0 \leq \alpha_i \leq \frac{1}{\nu N} \\ \frac{\partial L}{\partial \boldsymbol{a}} = 0 \qquad &\longrightarrow \qquad \boldsymbol{a} = \frac{\sum_i^N \alpha_i \phi(\boldsymbol{x}_i)}{\sum_i^N \alpha_i} = \sum_i^N \alpha_i \phi(\boldsymbol{x}_i) \end{aligned}$$

Incorporating these relations into the Lagrangian gives us the following objective functional to be maximized with respect to α_i:

$$L = \sum_{i=1}^N \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^N \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{4}$$

subject to $0 \leq \alpha_i \leq 1/\nu N$. The value of α_i depends on whether the constraint $\|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|_{\mathcal{H}}^2 \leq R^2 + \xi_i$ is satisfied by the corresponding sample $\boldsymbol{x}_i$. We can encounter one of the following three cases:

$$\begin{aligned} \|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|_{\mathcal{H}}^2 < R^2 \qquad &\Longleftrightarrow \qquad \alpha_i = 0, \quad \gamma_i = 0 \\ \|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|_{\mathcal{H}}^2 = R^2 \qquad &\Longleftrightarrow \qquad 0 < \alpha_i < \frac{1}{\nu N}, \gamma_i = 0 \\ \|\phi(\boldsymbol{x}_i) - \boldsymbol{a}\|_{\mathcal{H}}^2 > R^2 \qquad &\Longleftrightarrow \qquad \alpha_i = \frac{1}{\nu N}, \gamma_i > 0 \end{aligned}$$

Samples that lay inside the hypersphere have their corresponding Lagrangian multiplier α_i equal to zero. Furthermore, the partial derivative of the Lagrangian with respect to the center of the hypersphere $\boldsymbol{a}$ shows that $\boldsymbol{a}$ is a linear combination of the input data, with weight factors α_i obtained by optimizing equation (4). Therefore, the only objects needed for the description of

the boundary are those with their corresponding $\alpha_i > 0$ (samples *on* and *outside* the boundary). These vectors are called the *Support Vectors* of the description.

The radius of the optimal hypersphere is obtained by calculating the distance in the feature space $\mathcal{H}$ from the center $\boldsymbol{a}$ to any sample $\phi(\boldsymbol{x}_k)$ on the boundary (having $0 < \alpha_k < \frac{1}{\nu N}$):

$$\begin{aligned} R^2 &= \|\phi(\boldsymbol{x}_k) - \boldsymbol{a}\|_{\mathcal{H}}^2 \\ &= K(\boldsymbol{x}_k, \boldsymbol{x}_k) - 2\sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_k, \boldsymbol{x}_i) + \sum_{i,j=1}^{N} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j). \end{aligned}$$

In order to determine whether a new test sample $\boldsymbol{z}$ lays within the hypersphere, we evaluate the distance $\|\phi(\boldsymbol{z}) - \boldsymbol{a}\|_{\mathcal{H}}^2$ between the center $\boldsymbol{a}$ and the mapping $\phi(\boldsymbol{z})$ in the feature space. The new sample $\boldsymbol{z}$ is accepted and considered as a normal sample if the distance calculated is smaller than the radius:

$$\|\phi(\boldsymbol{z}) - \boldsymbol{a}\|_{\mathcal{H}}^2 \leq R^2.$$

Otherwise, an intrusion is detected, $\boldsymbol{z}$ is considered as an outlier and is rejected.

3.2 Kernel Principal Component Analysis

Kernel Principal Component Analysis (KPCA) is a nonlinear application of PCA in a kernel-defined feature space, where using the kernel $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y})$ is equivalent to performing the original PCA [26]. KPCA extracts the principal components of a dataset by projecting the mapped data onto the subspace spanned by the most relevant eigenvectors. Hoffmann investigates in [33] the use of KPCA for one-class classification when he introduces the *reconstruction error* as a measure of novelty. This error takes into account the heterogeneous variance of the distribution of the data in the feature space. The principal motivation of Hoffmann's algorithm is to enclose a smaller volume in $\mathcal{H}$ than other one-class algorithms for the same number of enclosed data samples.

Given a training dataset $\boldsymbol{x}_i$, $i \in \{1, \ldots, N\}$ in a p-dimensional input space, the first step in the KPCA algorithm is to find eigenvalues $\lambda > 0$ and eigenvec-

tors $\boldsymbol{v}$ of the covariance matrix $\widetilde{C}$ in the feature space $\mathcal{H}$, satisfying:

$$\lambda \boldsymbol{v} = \widetilde{C}\boldsymbol{v}.$$

Each eigenvector $\boldsymbol{v}$ is a linear combination of the mapped data, and takes the following form:

$$\boldsymbol{v} = \sum_{i=1}^{N} \alpha_i \widetilde{\phi}(\boldsymbol{x}_i),$$

where $\widetilde{\phi}(\boldsymbol{x}_i)$ is the centered version of $\phi(\boldsymbol{x}_i)$ in the feature space:

$$\widetilde{\phi}(\boldsymbol{x}_i) = \phi(\boldsymbol{x}_i) - \frac{1}{N}\sum_{i=1}^{N} \phi(\boldsymbol{x}_i).$$

The coefficients α_i are given by solving the following eigen decomposition problem:

$$N\lambda\alpha = \widetilde{K}\alpha,$$

where the kernel matrix $\widetilde{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \widetilde{\phi}(\boldsymbol{x}_i), \widetilde{\phi}(\boldsymbol{x}_j) \rangle$ corresponding to $\widetilde{\phi}(\boldsymbol{x}_i)$ is computed as follows:

$$\begin{aligned}\widetilde{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) &- \frac{1}{N}\sum_{r=1}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}_r) \\ &- \frac{1}{N}\sum_{r=1}^{N} K(\boldsymbol{x}_r, \boldsymbol{x}_j) + \frac{1}{N^2}\sum_{r,s=1}^{N} K(\boldsymbol{x}_r, \boldsymbol{x}_s).\end{aligned}$$

In fact, this kernel matrix will be used in the optimization problem without the need to compute directly the covariance matrix $\widetilde{C}$.

Let $\mathcal{P}$ be the projection operator onto the subspace spanned by the q most relevant eigenvectors $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \cdots, \boldsymbol{v}^{(q)}$. The reconstruction error measures the squared distance between the centered sample $\widetilde{\phi}(\boldsymbol{z})$ and its projection into the subspace spanned by the most relevant eigenvectors. The reconstruction error is computed as follows:

$$\begin{aligned}\|\widetilde{\phi}(\boldsymbol{z}) - \mathcal{P}\widetilde{\phi}(\boldsymbol{z})\|^2_{\mathcal{H}} = \langle \widetilde{\phi}(\boldsymbol{z}), \widetilde{\phi}(\boldsymbol{z}) \rangle &- 2\langle \widetilde{\phi}(\boldsymbol{z}), \mathcal{P}\widetilde{\phi}(\boldsymbol{z}) \rangle \\ &+ \langle \mathcal{P}\widetilde{\phi}(\boldsymbol{z}), \mathcal{P}\widetilde{\phi}(\boldsymbol{z}) \rangle,\end{aligned}$$

where $\langle\widetilde{\phi}(\boldsymbol{z}),\widetilde{\phi}(\boldsymbol{z})\rangle = \widetilde{K}(\boldsymbol{z},\boldsymbol{z})$. Since the projection operator $\mathcal{P}$ is idempotent ($i.e., \mathcal{P}^2 = \mathcal{P}$) and self-adjoint ($i.e., \langle\mathcal{P}\widetilde{\phi}(\boldsymbol{z}),\widetilde{\phi}(\boldsymbol{z}')\rangle = \langle\widetilde{\phi}(\boldsymbol{z}),\mathcal{P}\widetilde{\phi}(\boldsymbol{z}')\rangle$), then the reconstruction error's expression is simplified as follows:

$$\|\widetilde{\phi}(\boldsymbol{z}) - \mathcal{P}\widetilde{\phi}(\boldsymbol{z})\|^2_{\mathcal{H}} = \widetilde{K}(\boldsymbol{z},\boldsymbol{z}) - \langle\mathcal{P}\widetilde{\phi}(\boldsymbol{z}),\mathcal{P}\widetilde{\phi}(\boldsymbol{z})\rangle. \tag{5}$$

The projection of a mapped sample $\phi(\boldsymbol{z})$ onto the subspace spanned by the q most relevant eigenvectors is given by the following expression:

$$\mathcal{P}\widetilde{\phi}(\boldsymbol{z}) = \sum_{l=1}^{q}\langle\widetilde{\phi}(\boldsymbol{z}),\boldsymbol{v}^{(l)}\rangle\frac{\boldsymbol{v}^{(l)}}{\|\boldsymbol{v}^{(l)}\|}.$$

Therefore, since the eigenvectors are orthonormal, the right-hand-side of equation (5) becomes:

$$\langle\mathcal{P}\widetilde{\phi}(\boldsymbol{z}),\mathcal{P}\widetilde{\phi}(\boldsymbol{z})\rangle = \sum_{l=1}^{q}\langle\widetilde{\phi}(\boldsymbol{z}),\boldsymbol{v}^{(l)}\rangle^2,$$

where

$$\begin{aligned}
\langle\widetilde{\phi}(\boldsymbol{z}),\boldsymbol{v}^{(l)}\rangle &= \Bigg(\Bigg[\phi(\boldsymbol{z}) - \frac{1}{N}\sum_{r=1}^{N}\phi(\boldsymbol{x}_r)\Bigg] \\
&\qquad \cdot\Bigg[\sum_{i=1}^{N}\alpha_i^{(l)}\phi(\boldsymbol{x}_i) - \frac{1}{N}\sum_{i,r=1}^{N}\alpha_i^{(l)}\phi(\boldsymbol{x}_r)\Bigg]\Bigg) \\
&= \sum_{i=1}^{N}\alpha_i^{l}\Bigg[K(\boldsymbol{z},\boldsymbol{x}_i) - \frac{1}{N}\sum_{r=1}^{N}K(\boldsymbol{x}_i,\boldsymbol{x}_r) \\
&\qquad - \frac{1}{N}\sum_{r=1}^{N}K(\boldsymbol{z},\boldsymbol{x}_r) + \frac{1}{N^2}\sum_{r,s=1}^{N}K(\boldsymbol{x}_r,\boldsymbol{x}_s)\Bigg] \\
&= \sum_{i=1}^{N}\alpha_i^{(l)}\widetilde{K}(\boldsymbol{z},\boldsymbol{x}_i).
\end{aligned}$$

After evaluating the reconstruction error for the training dataset, an error threshold is fixed based on the predefined number of outliers and the description boundary of the normal data is computed. The threshold being fixed, to decide whether new test samples belong to the normal behavior of the system, the reconstruction error of each sample is computed. If this error is smaller than the threshold, the corresponding sample is accepted and treated as a normal sample. Otherwise, this sample is considered as an outlier and will be rejected. This is how the reconstruction error defines a novelty measure.

3.3 Bandwidth parameter optimization

The bandwidth parameter s of RBF kernels plays a crucial role in defining the description boundary around the training data. s must be chosen in a way to obtain a tight boundary that fits correctly the data and avoids overfitting. For instance, a small value of s causes the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to be almost equal to zero for any distinct input samples ($i \neq j$). In this case, the Lagrangian in equation (4) is optimized when all the samples become support objects with $\alpha_i = \frac{1}{N}$, and we have overfitting. On the other hand, a large value of s makes $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to be almost equal to 1, and all the samples become in the feature space on a hypersphere of radius equals to 1. Consequently, the classifier underfits the data and we obtain a loose description boundary.

The heuristic proposed in this paper inspired by Haykin [19] avoids all the time consuming of the other methods exixting in the literature, and leads to the optimal choice of the parameter s. Since s is related to the spread of the training dataset, the number of input samples and the fraction of samples considered as outliers, hence the computation of s should take into consideration all these factors in order to obtain the optimal bandwidth. Therefore, we propose the following expression for computing s:

$$s = \frac{d_{max}}{\sqrt{2M}},$$

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 (equivalent to the fraction of rejected samples multiply by the total number of input data). We note that the type of the distance used in the kernel function is the same as the distance in this equation, i.e. in the case of the gaussian kernel we compute the euclidian distance, while in the case of the proposed City-block kernel we use the city-block distance. This expression of s ensures that the extreme cases (overfitting and underfitting the data) are avoided, and the optimization of this parameter is obtained without any time computational cost.

Figure 3: Gas pipeline testbed

4 Simulations and results

In this paper, one-class classification algorithms are applied on simulated data as well as on real data from the Gas pipeline testbed of the Mississippi State University SCADA Laboratory. This section provides in the first place a description of the Gas pipeline testbed. The results on simulated and real data using the proposed kernel and proposed optimization parameter heuristic are presented afterwards.

4.1 Gas Pipeline Testbed

The gas pipeline testbed illustrated in figure 3 is used to move natural gas or any other petroleum products to the market. This testbed represents a typical SCADA system embracing a *Master Terminal Unit (MTU)*, *Remote Terminal Units (RTU)* and a *Human Machine Interface (HMI)* that allows operators to monitor and control the physical process. The gas pipeline control system contains an air pump that pumps air into the pipeline, a pressure sensor which allows pressure visibility at the pipeline and remotely on the HMI, a release valve and a solenoid release valve to loose air pressure from the pipeline. The

control scheme includes an automatic and a manual mode. In the automatic mode, a PID is used to control the pressure in the pipeline, while in the manual mode the operator can supervise the system and take charge over the pump state and the two release valves.

Cyber intrusions on the gas pipeline monitoring system can cause a loss of control of the physical process, and this may lead to huge financial and physical losses. For this reason, several types of false commands and responses are injected into the network traffic of the system to make its behavior abnormal, in order to study the vulnerabilities of the system and their implications on the controlled process. For instance, the "negative pressure value injection" returns a negative response of the pressure from the RTU while the pressure can not be negative in the system, the "fast change response injection" sends measurements that change very fast opposed to the case of a normal behavior of the pipeline, the "burst response injection" sends only one value equals to the maximum pressure limit, the "wave pressure injection" and the "single packet injection". The training phase of the classification algorithms is made on the normal training dataset while the tests were conducted on data containing these types of cyberattacks.

4.2 Results on simulated data

The proposed City-block kernel and the optimization parameter heuristic are tested, in the first place, on the *ring-line-square* data as illustrated in figure 4 [33]. This simulated dataset is very interesting and challenging, since it combines three different distributions in one training example: a ring, a line and a square. The upper bound on the fraction of outliers is fixed to 10% in this case, and the optimal bandwidth parameter s is computed as detailed in the previous section. The trade-off parameter ν in the SVDD approach is set to $\nu = 0.1$, while preliminary experiments were conducted and the number of eigenvectors q in the feature space in KPCA is set to $q = 40$. The computation of the optimal s for the Gaussian kernel leads to $s = 0.4441$, and $s = 0.5478$ for the City-block

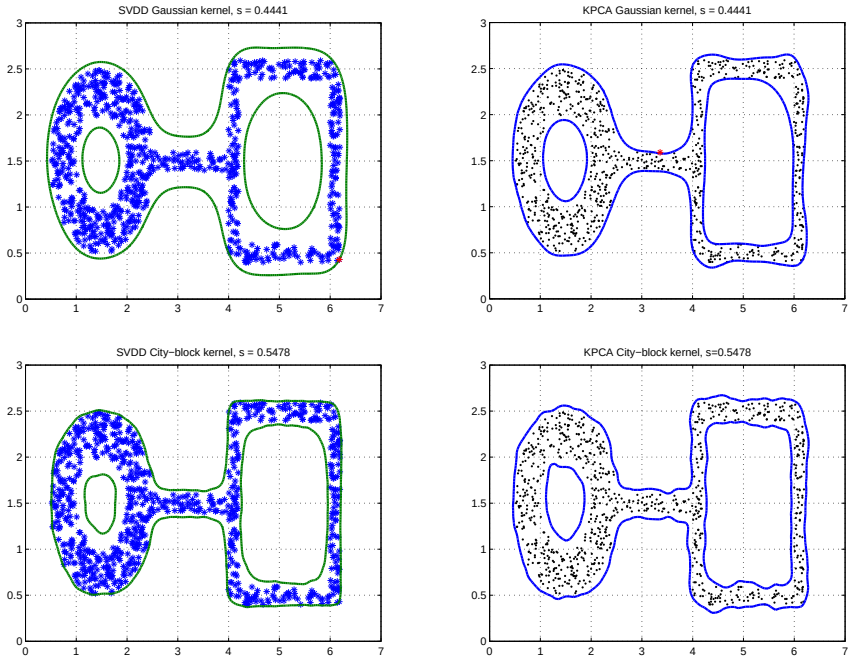

Figure 4: The Gaussian kernel is used on the ring-line-square data with the SVDD approach (the image on the top left) and the KPCA (the image on the top right), while the results with the proposed City-block kernel appear with the SVDD approach (the image on the bottom left) and the KPCA (the image on the bottom right).

kernel.

The results shown in figure 4 have two important meanings. First, the decision boundaries follow the shape of the distribution of the training data in all the studied cases, which indicates that the proposed heuristic for computing the bandwidth parameter gives indeed the optimal s. Secondly, the decision boundaries obtained when using the proposed City-block kernel are more tight than the boundaries obtained with the Gaussian kernel, and they describe the variation in the shape of the data with the most appropriate behavior. These results confirm that using the city-block distance in RBF kernels leads to the

optimal description of the data, since it is proven to be more sensitive than the Gaussian kernel on simultaneous variation of multiple features.

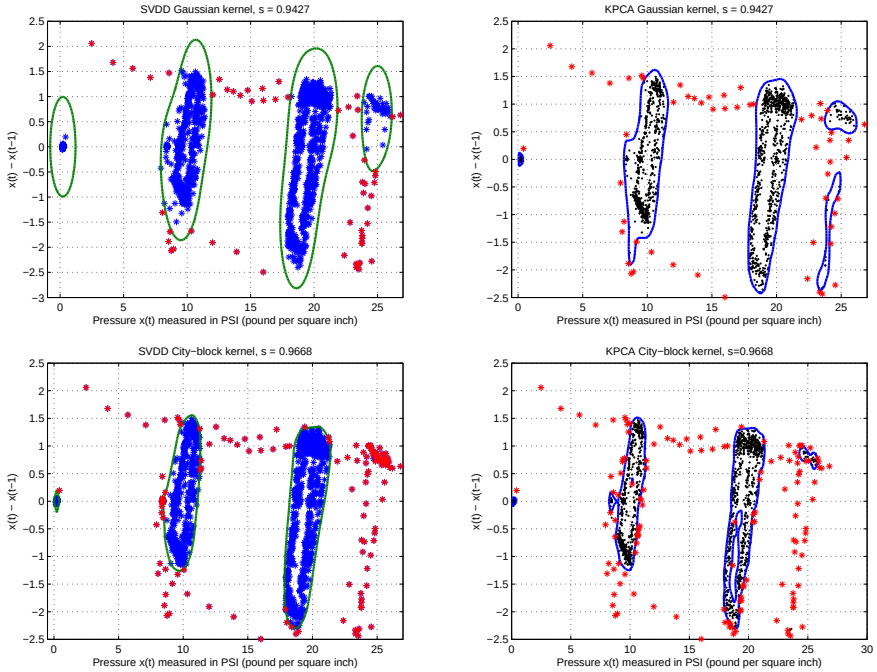

Figure 5: The Gaussian kernel is applied on the gas pipeline data with the SVDD approach (the image on the top left) and the KPCA (the image on the top right), while the results with the proposed City-block kernel appear with the SVDD approach (the image on the bottom left) and the KPCA (the image on the bottom right). The decision boundary is given by the lines, and the red samples represent the outliers. The City-block kernel describes the data of the Gas pipeline in a better way than the Gaussian kernel.

4.3 Results on the Gas pipeline testbed

The allowed pressure range measured by the sensor in the pipeline is from 0 to 20 PSI (pound per square inch), with a margin of 10% which fixes the maximum accepted pressure to 22 PSI. The pipeline operates in three principal modes; the

Table 1: The confusion matrix of several types of attacks with the SVDD approach.

		Gaussian SVDD		City-block SVDD	
		Normal	Outlier	Normal	Outlier
Training data	Normal	99.7	0.3	98.56	1.44
	Outlier	60.3	39.7	16.03	83.97
Slow injection	Normal	99.7	0.3	99.41	0.59
	Outlier	0.9	99.1	0.47	99.53
Fast injection	Normal	99.35	0.65	98.7	1.3
	Outlier	11.6	88.4	11.6	88.4
Burst injection	Normal	99.3	0.7	98.61	1.39
	Outlier	33.9	66.1	26.6	73.4
Single injection	Normal	99.2	0.8	99.2	0.8
	Outlier	0.78	99.22	0.78	99.22
Wave injection	Normal	99.76	0.24	98.81	1.19
	Outlier	35.08	64.92	33.3	66.7

first mode is characterized by a very low pressure maintained around 0.1 PSI, the second mode keeps it around 10 PSI (the accepted range lays between 9 and 11 PSI), while the third mode should maintain the pressure around 20 PSI (the accepted range is 18 to 22 PSI). The high pressure (greater than 22 PSI) and the transitional states between different modes are considered as outliers.

Let $x(t)$ be the pressure in the pipeline at instant t. The choice of the input vectors should be made to draw attention to the fact that the pressure measurements of two consecutive instants in the normal functioning modes of the system must be close to each other. Furthermore, the presence of gaps in the pressure between two consecutive instants may be a strong sign of a cyberattack. For these reasons, the time series is folded into 2-dimensional input vectors composed of the pressure at instant t and the difference in the

Table 2: The confusion matrix of several types of attacks with the KPCA approach.

		Gaussian KPCA		City-block KPCA	
		Normal	Outlier	Normal	Outlier
Training data	Normal	99.57	0.43	96.63	3.37
	Outlier	63.36	36.64	34.35	65.65
Slow injection	Normal	99.41	0.59	91.95	8.05
	Outlier	0.95	99.05	0.47	99.53
Fast injection	Normal	98.3	1.7	83.7	16.3
	Outlier	11.6	88.4	11.6	88.4
Burst injection	Normal	99.3	0.7	90.3	9.7
	Outlier	27.9	72.1	24.8	75.2
Single injection	Normal	98.37	1.63	88	12
	Outlier	0.78	99.22	0.78	99.22
Wave injection	Normal	98.8	1.2	88.6	11.4
	Outlier	35.1	64.9	3.51	96.49

pressure between instants t and $t-1$, namely $\boldsymbol{x}_t = [x(t) \quad x(t) - x(t-1)]$.

The results on real Gas pipeline data for the Gaussian and the proposed City-block kernels are shown in figure 5. The optimal bandwidth parameter is computed as detailed in the previous section, and the upper bound on the fraction of outliers is fixed at 0.2. As illustrated in figure 5, the decision boundary in each case encloses the samples accepted as normal data, while outliers are rejected outside this boundary. Compared to the Gaussian kernel that gives a description that might be considered a little bit loose, the City-block kernel leads to a more tight boundary that delineates the variation in the distribution of the training data with the most appropriate way. The error probabilities of the different types of cyberattacks studied in this paper are detailed in Tables 1 and 2. The use of the City-block kernel gives a better performance than the

Gaussian kernel especially when it comes to decreasing the error of the second type (The outliers that are accepted as normal data). The outliers detected in the SVDD approach using the City-block kernel for different types of cyberattacks are illustrated in figure 6.

5 Conclusion

The conclusion goes here.

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.

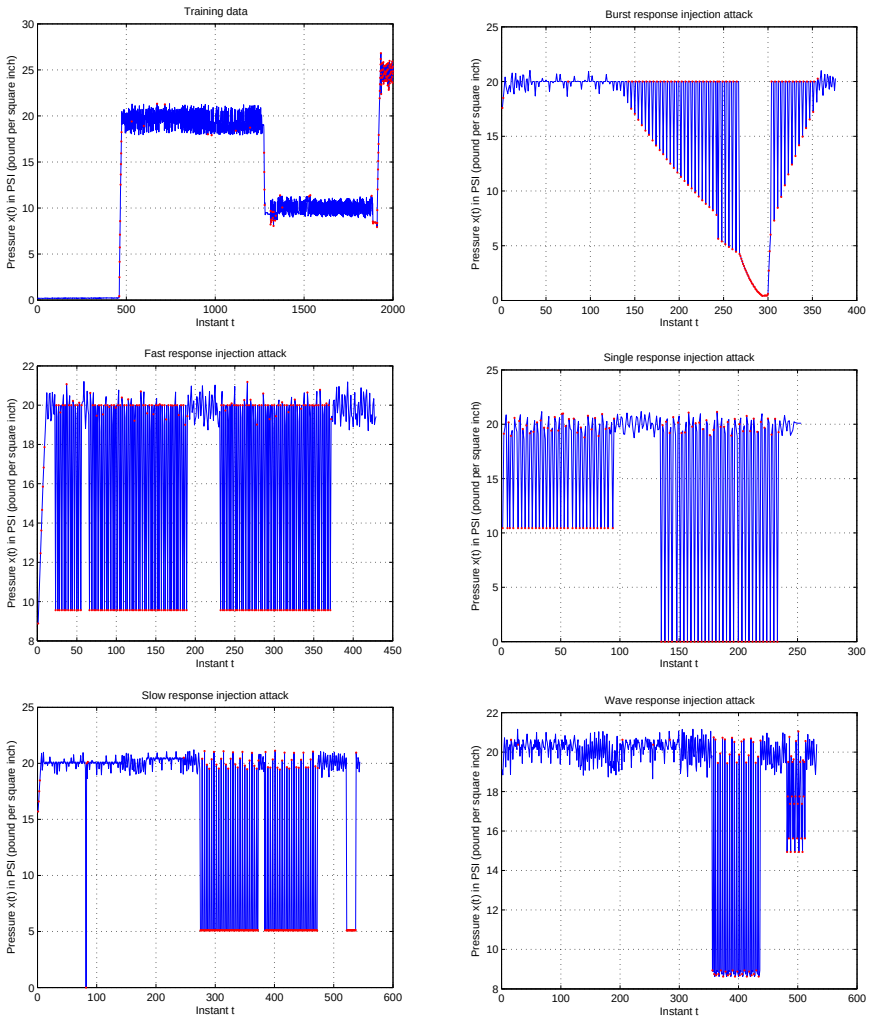

Figure 6: Detection of outliers for several types of attacks with the SVDD approach using the proposed City-block kernel. The blue samples refer to the data accepted as normal data while the red samples are considered as outliers.

Bibliography

[1] K. Stouffer, J. Falco, and K. Kent, "Guide to supervisory control and data acquisition (scada) and industrial control systems security," National Institute of Standards and Technology (NIST), Tech. Rep., September 2006.

[2] K. A. Stouffer, J. A. Falco, and K. A. Scarfone, "Sp 800-82. guide to industrial control systems (ics) security: Supervisory control and data acquisition (scada) systems, distributed control systems (dcs), and other control system configurations such as programmable logic controllers (plc)," Gaithersburg, MD, United States, Tech. Rep., 2011.

[3] I. Fovino, M. Masera, L. Guidi, and G. Carpi, "An experimental platform for assessing scada vulnerabilities and countermeasures in power plants," in *Human System Interactions (HSI), 2010 3rd Conference on*, may 2010, pp. 679 –686.

[4] C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for scada systems," *Power Systems, IEEE Transactions on*, vol. 23, no. 4, pp. 1836–1846, 2008.

[5] V. Urias, B. Van Leeuwen, and B. Richardson, "Supervisory command and data acquisition (scada) system cyber security analysis using a live, virtual, and constructive (lvc) testbed," in *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012*, 2012, pp. 1–8.

[6] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in *Critical Infrastructure Protection*, 2007, pp. 73–82.

[7] H. Christiansson and E. Luiijf, "Creating a european scada security testbed," in *Critical Infrastructure Protection*, ser. IFIP International Federation for Information Processing, E. Goetz and S. Shenoi, Eds. Springer US, 2007, vol. 253, pp. 237–247.

[8] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '11. New York, NY, USA: ACM, 2011, pp. 355–366. [Online]. Available: http://doi.acm.org/10.1145/1966913.1966959

[9] S. Gorman, "Electricity Grid in U.S. Penetrated By Spies," *The Wall Street Journal*, Apr. 2008.

[10] T. Chen and S. Abu-Nimeh, "Lessons from stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011.

[11] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security Privacy, IEEE*, vol. 9, no. 3, pp. 49–51, 2011.

[12] P. W. Oman and M. Phillips, "Intrusion detection and event monitoring in scada networks," in *Critical Infrastructure Protection*, 2007, pp. 161–173.

[13] P. Gross, J. Parekh, and G. Kaiser, "Secure selecticast for collaborative intrusion detection systems," in *3rd International Workshop on Distributed Event-Based Systems (DEBS'04)*, Edinburgh, Scotland, UK, May 2004. [Online]. Available: http://serl.cs.colorado.edu/ carzanig/debs04/debs04gross.pdf

[14] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. Fovino, and A. Trombetta, "A multidimensional critical state analysis for detecting intrusions

in scada systems," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 2, pp. 179 –186, May 2011.

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

[16] 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.

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

[18] 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.

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*, 2nd ed. Prentice Hall, Jul. 1998. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0132733501

[20] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[21] C. Soares, P. B. Brazdil, and P. Kuba, "A Meta-Learning Method to Select the Kernel Width in Support Vector Regression," *Machine Learning*, vol. 54, no. 3, pp. 195–209, 2004. [Online]. Available: http://dx.doi.org/10.1023/b:mach.0000015879.28004.9b

[22] V. Cherkassky and Y. Ma, "Practical selection of svm parameters and noise estimation for svm regression," *Neural Netw.*, vol. 17, no. 1, pp.

113–126, Jan. 2004. [Online]. Available: http://dx.doi.org/10.1016/S0893-6080(03)00169-2

[23] P. F. Evangelista, M. J. Embrechts, and B. K. Szymanski, "Some properties of the gaussian kernel for one class learning," in *Proceedings of the 17th international conference on Artificial neural networks*, ser. ICANN'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 269–278. [Online]. Available: http://dl.acm.org/citation.cfm?id=1776814.1776844

[24] P. Gurram and H. Kwon, "Support-vector-based hyperspectral anomaly detection using optimized kernel parameters," *Geoscience and Remote Sensing Letters, IEEE*, vol. 8, no. 6, pp. 1060–1064, 2011.

[25] 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.

[26] 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.

[27] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*. London, UK, UK: Springer-Verlag, 2001, pp. 416–426.

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

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

[30] 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. [Online]. Available: http://dx.doi.org/10.1162/089976601750264965

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

[32] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

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