Journal of Applied Logic 24 (2017) 71–84

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

On the combination of support vector machines and segmentation algorithms for anomaly detection: A petroleum industry comparative study

Luis Martí [a,*], Nayat Sanchez-Pi [b], José Manuel Molina López [c], Ana Cristina Bicharra Garcia [a]

[a] *Institute of Computing, Universidade Federal Fluminense, Niterói (RJ), Brazil*
[b] *Institute of Mathematics and Statistics, Universidade do Estado do Rio de Janeiro, Rio de Janeiro (RJ), Brazil*
[c] *Department of Informatics, Universidad Carlos III de Madrid, Colmenarejo (Madrid), Spain*

ARTICLE INFO

Article history:
Available online 11 November 2016

Keywords:
Anomaly detection
Support vector machines
Time series segmentation
Kalman filters
Oil industry application

ABSTRACT

Anomaly detection has to do with finding patterns in data that do not conform to an expected behavior. It has recently attracted the attention of the research community because of its real-world application. The correct detection unusual events empower the decision maker with the capacity to act on the system in order to correctly avoid, correct, or react to the situations associated with them. Petroleum industry is one of such real-world application scenarios. In particular, heavy extraction machines for pumping and generation operations like turbomachines are intensively monitored by hundreds of sensors each that send measurements with a high frequency for damage prevention. For dealing with this and with the lack of labeled data, in this paper we describe a combination of a fast and high quality segmentation algorithm with a one-class support vector machine approach for efficient anomaly detection in turbomachines. As a result we perform empirical studies comparing our approach to another using Kalman filters in a real-life application related to oil platform turbomachinery anomaly detection.

1. Introduction

The importance of anomaly detection is a consequence of the fact that anomalies in data translate to significant actionable information in a wide variety of application domains. The correct detection of such types of unusual information empowers the decision maker with the capacity to act on the system in order to correctly avoid, correct, or react to the situations associated with them.

* Corresponding author.
E-mail address: lmarti@ic.uff.br (L. Martí).

http://dx.doi.org/10.1016/j.jal.2016.11.015
1570-8683/

Anomaly detection has extensive use in a wide variety of applications such as fraud and intrusion detection [10], fault detection in safety critical systems [16], finance [4] or industrial systems (see [35,6] for surveys on this topic).

One example in industry applications is the detection of anomalies in turbomachinery installed in offshore petroleum extraction platforms. Recent history shows us how important a correct handling of this equipment is as failures in this industry have a dramatic economical, social and environmental impact. In the particular case of industrial anomaly detection, units suffer damage due to continuous usage and the normal wear and tear. Such damages need to be detected early to prevent further escalation and losses. Data in this domain is referred to as industrial sensor data because it is recorded using different sensors and collected for analysis and it has a temporal aspect and time series analysis is also used in some works like: [15].

Due to the lack of labeled data for training/validation of models, in [20,30], we provided a solution for the detection of anomalies in turbomachinery, using a one-class SVM. This technique uses one class learning techniques for SVM [26] and learns a region that contains the training data instances (a boundary). Kernels, such as radial basis functions (RBF), are used to learn complex regions. For each test instance, the basic technique determines if the test instance falls within the learnt region. If a test instance falls within the learnt region, it is declared as normal; else it is declared as anomalous. We combined this technique with a time series segmentation to prune noisy, unreliable and inconsistent data.

In this paper we apply this approach in a real-life context oil industry-related problem and, in order to assert its validity, we compare its performance with currently used approaches, like Kalman filters and confidence intervals.

The remainder of this paper is organized in the following manner. In the next section, we discuss some theoretical foundations of the work. Subsequently, we describe our proposal in detail. After that, we present a case study for offshore oil platform turbomachinery. This case study is used to compare our approach with the application of Kalman filters. Finally in Section 5, some conclusive remarks and directions for future work are presented.

2. Foundations

The preset work addresses the problem of anomaly detection by comparing one-class SVM classifiers and Kalman filters with a novel and fast segmentation algorithm specially devised for this problem. In this section we present the theoretical pillars supporting the proposal.

2.1. Anomaly detection

Fault and damage prevention is known as the problem of finding patterns in data that do not conform to an expected behavior [8]. Unexpected patterns are often referred as anomalies, outliers or faults, depending on the application domain. In broad terms, anomalies are patterns in data that do not conform to a well-defined 'normal' behavior and, hence, can be labeled as 'anomalies' [8].

Anomaly detection techniques are also classified in three categories based on the nature of the undesired anomaly: point anomaly, contextual anomaly, collective anomaly. Point anomaly is when an individual data instance can be considered as anomalous with respect to the rest of the data. If this data instance is anomalous in a specific context then it is considered to be contextual anomaly. Furthermore, if a collection of related data instances is anomalous with respect to the rest of entire dataset, where these individual instances are not anomalies by themselves but their occurrence together is anomalous, then it is termed as collective anomaly.

Applications of anomaly detection techniques vary depending on the user, the problem domains, and even the data. Anomaly detection techniques have been proposed in literature, based on distribution, distance, density, clustering and classification.

In many cases the anomaly detection is related to outlier detection. In statistics, outliers are data instances that are deviate from given sample in which they occur. Grubbs in [12] defined them as "an outlying observation, or 'outlier', is one that appears to deviate markedly from other members of the sample in which it occurs".

Some of the anomaly detection techniques are:

- Distribution-based approaches are when a given statistical distribution is used to model the data points. Then, points that deviate from the model are flagged as anomalies or outliers. These approaches are unsuitable for moderately high-dimensional datasets and require prior knowledge of the data distribution. They are also named as parametric and non-parametric statistical modelings [15].
- Depth-based approaches compute the different layers of convex hulls and flags objects in the outer layer as anomalies or outliers. These approaches avoid the requirement of fitting a distribution to the data, but have a high computational complexity.
- Clustering approaches are when many clustering algorithms can detect anomalies or outliers as elements that do not belong—or are near—to any cluster.
- Distance-based anomalies or outliers detection marks, on the other hand, how distant is an element from a subset of the elements closest to it. It has been pointed out [5] that these methods cannot cope with datasets having both dense and sparse regions, an issue denominated *multi-density problem*. Density-based anomalies or outlier detection has been proposed to overcome the multi-density problem by means of the local outlier factor (LOF). LOF measures the degree of outlierness for each dataset element and depends on the local density of its neighborhood. This approach fails to deal correctly with another important issue: the *multi-granularity problem*. The local correlation integral (LOCI) method, and its outlier metric, the multi-granularity deviation factor (MDEF), were proposed with the purpose of correctly dealing with multi-density and multi-granularity [25]. Spectral decomposition is used to embed the data in lower dimensional subspace in which the data instances can be discriminated easily. Many techniques based on principal component analysis (PCA) have emerged [27]. Some of them decompose space to normal, anomaly and noise subspaces. The anomalies can be then detected in anomaly subspace [11].
- Classification approaches are those where the problem is posed as the identification of which categories an observation belongs to. It operates in two phases: first it learns a model based on subset observations (training set) and second it infers a class for new observations (testing set) based on learnt model. These methods operate under the assumption that a classifier distinguishes between normal and anomalous classes that can be learnt in the given feature space. Based on the labels available for training phase, classification based anomaly detection techniques can be grouped into two broad categories: binary-class or multi-class [1] and one-class anomaly detection techniques [28] (more about this on Section 2.4).

In any case, in order to deal with problem in an effective way a multi-component approach is required. This is because it is necessary to first identify instances that represent 'normal' or 'anomalous' data, and then apply one of the detection techniques relying on the previous information.

The situation is particularly like that in the case of multi-modal time series-based anomaly detection where the system being supervised has different 'normal' operating regimes which should be identified and not confused with anomalies. Therefore, in order to identify homogeneous series segments it is necessary to apply a time-series segmentation algorithm.

Furthermore, frequently there is little or none information or data instances of anomalies, leading to heavily unbalanced data. In those cases it is necessary to apply unsupervised or semi-supervised approaches, like the Kalman filters or the one-class support vector machines.

Because of these reasons, in subsequent subsections we deal with those concepts in detail.

2.2. Time series segmentation

In the problem of finding frequent patterns, the primary purpose of time series segmentation is dimensionality reduction. For the anomalous detection problems in turbomachines, it is essential to segment the dataset available in order to automatically discover the operational regime of the machine in the recent past. There is a vast work done in time series segmentation. We now provide a formal definition on the problem and describe the main segmentation methods available.

A time series can be expressed as a set of time-ordered possible infinite measurements, $\mathcal{S}$, that consists of pairs $\langle s_i, t_i \rangle$ of sensor measurements, s_i, and time instants, t_i, such that

$$\mathcal{S} = \{\langle s_0, t_0 \rangle, \langle s_1, t_1 \rangle, \ldots \langle s_i, t_i \rangle, \ldots\}, i \in \mathbb{N}^+; \forall t_i, t_j : t_i < t_j \text{ if } i < j. \tag{1}$$

Sensor measurements s_i take values on a set that depends on the particular characteristics of the sensor.

In practice, time series frequently have a simpler definition as: measurements that are usually obtained at equal time intervals between them. This type of time series is known as regular time series. In this case, the explicit reference to time can be dropped and exchanged a order reference index, leading to a simpler expression

$$\mathcal{S} = \{s_0, s_1, \ldots s_i, \ldots\}, i \in \mathbb{N}^+. \tag{2}$$

The use of regular time series is so pervasive that the remainder of this paper will deal only with them. Henceforth, the term time series will be used to refer to a regular time series.

Depending on the application, the goal of the segmentation is to locate stable periods of time, to identify change points, or to simply compress the original time series into a more compact representation. Although in many real-life applications a lot of variables must be simultaneously tracked and monitored, most of the segmentation algorithms are used for the analysis of only one time-variant variable.

A segmentation algorithm can be represented as a function $\Theta(\cdot)$ that creates K segments of time series such that

$$\Theta : \mathcal{S} \to \langle \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K \rangle, \tag{3}$$

where $\langle \mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K \rangle$ exhibit the properties (i) $\mathcal{S} = \cup_{i=1}^{K} \mathcal{S}_i$, or, in other words, that $\mathcal{S}$ can be reconstructed from the segmentation without data loss and (ii) $\mathcal{S}_i \cap \mathcal{S}_j = \varnothing$, $\forall i, j = 1, \ldots, K$ and $i \neq j$, that implies that each segment is disjoint with regard to the rest.

Depending on the application, the goal of the segmentation is used to locate stable periods of time, to identify change points, or to simply compress the original time series into a more compact representation. Although in many real-life applications a lot of variables must be simultaneously tracked and monitored, most of the segmentation algorithms are used for the analysis of only one time-variant variable.

There is a vast literature about segmentation methods for different applications. Basically, there are mainly three categories of time series segmentation algorithms using dynamic programming. Firstly, sliding windows [2] top-down [17], and bottom-up [13] strategies. The sliding windows method is a purely implicit segmentation technique. It consists of a segment that is grown until it exceeds some error bound. This process is repeated with the next data point not included in the last created segment.

There are other novel methods for instance those using clustering for segmentation. The clustered segmentation problem is clearly related with the time series clustering problem [34] and there are also several definitions for time series [3]. One natural view of segmentation is the attempt to determine which components of a data set naturally “belong together”.

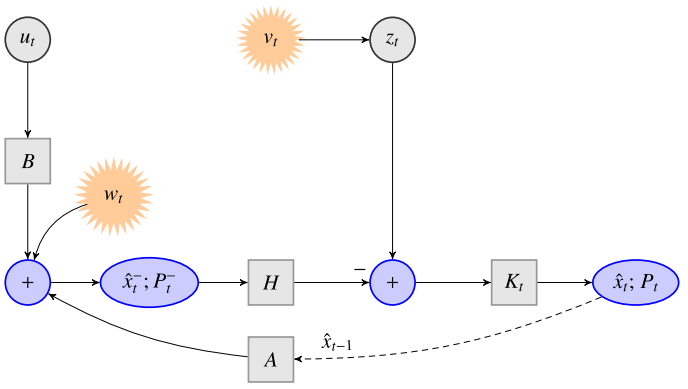


Fig. 1. Schematic representation of a Kalman filter iteration. $\hat{x}_{t-1}^-$, u_t and process noise are used to compute a priori estimation $\hat{x}_t^-$, and its error covariance, P_t^-. A prediction is made and subtracted from measurement z_t to calculate an error vector. This error is multiplied by the Kalman gain K_t, that generates a correction added to the prediction to yield the final estimate $\hat{x}_t$.

2.3. Kalman filters

The Kalman filter [14,21] provides an efficient computational means to estimate the state of a dynamic system from a series of incomplete and noisy measurements. This filter is the linear estimator with minimum squared error that can be applied to any dynamic system with errors following any distribution where the two first moments of the distribution are known. Furthermore, if we know that probability distributions are Gaussian and the system dynamics are linear, the Kalman filter is the globally optimal state estimator. It is very powerful since it supports estimations of past, current, and future states, even when some aspects of the modeled system are unknown.

The Kalman filter addresses the general problem of estimating the state of a discrete-time controlled process that is ruled by a linear stochastic difference equation.

The state of the filter is represented by two variables:

- $\hat{x}_t$, the estimate of the state at time t, and
- P_t, the error covariance matrix, which is a measure of the estimated accuracy of the current state estimate.

The Kalman filter estimates a process state by a recursive feedback control that can be separated in the *prediction* and *update* phases. The prediction phase is responsible for making an *a priori estimation* of the future state of the system relying on the current state and error covariance estimates. The update phase is responsible for feeding back the (noisy) measurement of the state of the system to output an improved *a posteriori estimate*. Fig. 1 summarizes these processes in a schematic form.

The Kalman filter assumes a dynamic model given by

$$x_t = Ax_{t-1} + Bu_t + w_t, \tag{4}$$

where u_t is an optional control input and the random variables $w_t \sim N(0, Q)$ represent the process noise.

Additionally, the measurement process is modeled by

$$z_t = Hx_t + v_t, \tag{5}$$

where H relates the real state of the process x_t to the measurement z_t and $v_t \sim N(0, R)$ is the measurement noise.

First, the a priori estimation, $\hat{x}_t^-$, and its error covariance, P_t^-, are calculated as

$$\hat{x}_t^- = A\hat{x}_{t-1} + Bu_t, \tag{6}$$

$$P_t^- = AP_{t-1}A^T + Q. \tag{7}$$

Then the update phase proceeds by computing the Kalman gain,

$$K_t = \frac{P_t^- H^T}{HP_t^- H^T + R}. \tag{8}$$

The a posteriori estimation is calculated as the feedback is entered in the filter as

$$\hat{x}_t = \hat{x}_t^- + K_t\left(z_t - H\hat{x}_t^-\right). \tag{9}$$

Finally, an a posteriori error covariance estimate is outputted by

$$P_t = (I - K_t H)P_t^-, \tag{10}$$

where I is the identity matrix.

2.4. One-class support vector machine

In order to understand one-class SVMs, it is convenient to first examine the traditional two-class support vector machine. Consider a (possibly infinite) data set,

$$\Psi = \{\langle \boldsymbol{x}_1, y_1\rangle, \langle \boldsymbol{x}_2, y_2\rangle, \ldots, \langle \boldsymbol{x}_i, y_i\rangle, \ldots\}, \tag{11}$$

where $\boldsymbol{x}_i \in \mathbb{R}^n$ is a given data point and $y_i \in \{-1, 1\}$ is the i-th output pattern, indicating the class membership.

SVMs can create a non-linear decision boundary by projecting the data through a non-linear function $\phi()$ to a space with a higher dimension. This implies that data points, which can't be separated by a linear threshold in their original (input) space, are converted to a feature space $\mathcal{F}$ where there is a hyperplane that separates the data points of one class from another. When that hyperplane would be projected back to the original space, it would have the shape of a non-linear curve. This hyperplane is represented by the equation,

$$\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b} = \boldsymbol{0}, \text{ with } \boldsymbol{w} \in \mathcal{F}, \ \boldsymbol{b} \in \mathbb{R}^n. \tag{12}$$

The hyperplane that is constructed determines the border between classes. All the data points for the class "-1" are on one side, and all the data points for class "1" on the other. The distance from the closest point from each class to the hyperplane is equal; thus, the constructed hyperplane searches for the maximal margin between the classes.

Slack variables, ξ_i, are introduced to allow some data points to lie within the margin in order to prevent the SVM classifier from over-fitting with noisy data (or to create a soft margin). Constant $C > 0$ determines the trade-off between maximizing the margin and the number of training data points within that margin (and thus training errors). Posed as an optimization problem, the adjustment of a SVM has as objective the minimization of the problem,

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{x}_i) = \frac{\|\boldsymbol{w}\|^2}{2} + C \sum_{i=1}^{n} \xi_i, \\ \text{subject to} \quad & y_i \left(\boldsymbol{w}^T \phi(x_i) + b \right) \geq 1 - \xi_i, \qquad \forall i = 1, \ldots, n; \\ & \xi_i \geq 0. \end{aligned} \tag{13}$$

Solving (13) using quadratic programming the decision (classification) function, $c(\boldsymbol{x})$, for a data point $\boldsymbol{x}$ becomes

$$c(\boldsymbol{x}) = \text{sign} \left(\sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \right). \tag{14}$$

Here the $\alpha_i > 0$ are the Lagrange multipliers that weight in the decision function and thus the "support" machine; hence the name Support Vector Machine. Since SVMs are generally considered to be sparse, there will be relatively few Lagrange multipliers with a non-zero value. Function $K(\boldsymbol{x}, \boldsymbol{x}_i)$ is known as the kernel function. Popular choices for the kernel function are linear, polynomial, sigmoidal. But the most popular choice by far, which provides that there is no enough a priori knowledge about the problem, is the Gaussian radial basis function

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp \left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2} \right), \tag{15}$$

where $\sigma \in \mathbb{R}$ is the kernel parameter and $\|\cdot\|$ is the dissimilarity measure. This is derived from the fact that this kernel function is able to model a non-linear decision boundary with a relatively simple mathematical tool. Furthermore, Gaussian kernels are universal kernels. This means that their use with appropriate regularization guarantees a globally optimal predictor which minimizes both the estimation and approximation errors of a classifier.

Kalman filters have been successfully applied to anomaly detection [33,32]. In this paper we compare our proposal with one of those approaches, the Kalman Filter based Outlier Detection (KFOD) [32]. This approach consists of two components, namely state transition module and measuring module. The state transition module predicates the state of next time based on that of current time. This module is essentially an auto-regression model, that makes use of the temporal dependency of successive readings on a sensor for prediction. The measuring module measures the local environment. Instead of using the local physical sensor device, we use the neighboring sensor readings as data produced by a virtual sensor device. Then KFOD combines the output of the two modules to generate an optimal estimate of the state.

3. Algorithm proposal

As already hinted earlier in the paper our proposal combines a fast segmentation algorithm with a support vector machine one-class classifier and a Kalman filter. The segmentation algorithms take care of identifying relatively homogeneous parts of the time series in order to focus the attention of the classifier to the most relevant portion of the time series. Therefore, parts of the time series that remain on the past can be safely disregarded. The Kalman filter plays a similar role to the one-class SVM by detecting deviations that indicate anomalies. Fig. 2 illustrates how the different components are connected to each other.

3.1. Segmentation algorithm

We devised a novel and fast algorithm for time series segmentation. In preliminary experiments [19] we assessed it with comparable state-of-the-art algorithms and it outperformed them in problems of similar nature to the one being discussed here.

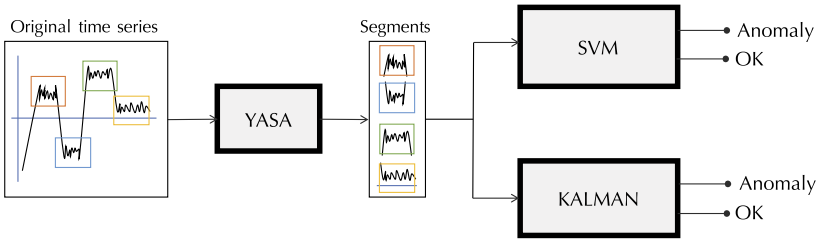


Fig. 2. Diagram representation of the interactions of the YASA segmentation algorithm with the Kalman filter and the one-class SVM.

function `segment_data`($\mathcal{S}^{(j)}_{t_{\max},t_0}$, $\rho_{\min}$, $l_{\max}$, $s_{\min}$, l)
Parameters:
▷ $\mathcal{S}^{(j)}_{t_{\max},t_0}$, time series data of sensor j corresponding to time interval $[t_0, t_{\max}]$.
▷ $\rho_{\min} \in [0, 1]$, minimum significance for statistical hypothesis test of linearity.
▷ $l_{\max} \in \mathbb{N}^+$, maximum levels of recursive calls.
▷ $s_{\min} \in \mathbb{N}^+$, minimum segment length.
Returns:
▷ $\Phi := \{\phi_1, \ldots, \phi_m\}$, data segments.
Algorithm:
if $l = l_{\max}$ **then**
return $\left\{\mathcal{S}^{(j)}_{t_{\max},t_0}\right\}$.
Perform linear regression,

$$\{m, b\} \leftarrow \texttt{linear\_regression}(\mathcal{S}^{(j)}_{t_{\max},t_0}).$$

if `linearity_test`($\mathcal{S}^{(j)}_{t_{\max},t_0}, m, b$) $> \rho_{\min}$ **then**
return $\left\{\mathcal{S}^{(j)}_{t_{\max},t_0}\right\}$.
Calculate residual errors,

$$\{e_0, \ldots, e_{\max}\} \leftarrow \texttt{residuals}(\mathcal{S}^{(j)}_{t_{\max},t_0}, m, b).$$

$t_s \leftarrow t_0$.
while $\max(\{e_0, \ldots, e_{\max}\}) > 0$ **and** $t_s \notin (t_0 + s_{\min}, t_{\max} - s_{\min})$ **do**
Determine split point,

$$t_s \leftarrow \arg\max_t \{e_t\}.$$

if $t_s \in (t_0 + s_{\min}, t_{\max} - s_{\min})$ **then**
$\Phi_{\text{left}} \leftarrow$ `segment_data`($\mathcal{S}^{(j)}_{t_s,t_0}$, $\rho_{\min}$, $l_{\max}$, $s_{\min}$, $l+1$).
$\Phi_{\text{right}} \leftarrow$ `segment_data`($\mathcal{S}^{(j)}_{t_{\max},t_s}$, $\rho_{\min}$, $l_{\max}$, $s_{\min}$, $l+1$).
return $\Phi_{\text{left}} \cup \Phi_{\text{right}}$.
return $\left\{\mathcal{S}^{(j)}_{t_{\max},t_0}\right\}$.

Fig. 3. Pseudocode of the proposed algorithm.

Besides the obvious purpose of obtaining a segmentation method that produces low approximation errors, another set of guidelines was observed while devising it. In particular we were interested in low computational impact and easy parameterization.

The yet another segmentation algorithm (YASA) [20,19] is sketched in Fig. 3 in pseudocode form. It is best understood when presented in recursive form, as it goes by computing a linear regression with the time series passed as parameter. Segmentation procedure first checks if the current level of recursion is acceptable. After that it goes by fitting a linear regression to the time series data. If the regression passes the linearity statistical hypothesis test then the current time series is returned as a unique segment.

If the regression does not model correctly the data, it means that it is necessary to partition the time series in at least two parts that should be further segmented. The last part of YASA is dedicated to this task. It locates the time instant where the regression had the larger error residuals.

3.2. One-class SVM for anomaly detection

Classification is a class of machine learning problem that consists on identifying to which (set of) classes or categories a given observation belongs, relying on a training set of data containing observations whose class membership is known. Binary classification implies classifying the observations of a given set into two classes on the basis of a classification rule. It can be extended to multi-class classification, where instances are classified into one of the more than two classes.

One-class classification [23] tries to identify observations of a specific class amongst all possible observations, by learning from a training set containing only the observations of that class. This is different (and more complex) than the traditional classification problem, which tries to distinguish between two or more classes with the training set containing objects from all the classes.

One-class classification based anomaly detection techniques assume that all training instances have only the same class label. Then, a machine learning algorithm is used to construct a discriminative boundary around the normal instances using a one-class classification algorithm. Any test instance that does not fall within the learned boundary is declared as anomalies. Support Vector Machines (SVMs) have been applied to anomaly detection in the one-class setting. One-class SVMs find a hyperplane in feature space, which has maximal margin to the origin and a preset fraction of the training examples lays beyond it.

The support vector method for novelty detection [31] essentially separates all the data points from the origin (in feature space $\mathcal{F}$) and maximizes the distance from this hyperplane to the origin. This results in a binary function which captures regions in the input space where the probability density of the data lives. Thus the function returns "+1" in a reduced region (capturing the training data points) and "−1" elsewhere.

The quadratic programming minimization problem is slightly different from the previously stated, but the similarity is evident,

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{w}, \boldsymbol{x}_i, \rho) = \tfrac{1}{2}\|\boldsymbol{w}\|^2 + \tfrac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho, \\ \text{subject to} \quad & \boldsymbol{w}\cdot\phi(\boldsymbol{x}_i) \geq \rho - \xi_i, \\ & \forall i = 1, \ldots, n \text{ and } \xi_i \geq 0. \end{aligned} \tag{16}$$

Applying Lagrange techniques and using a kernel function for the dot-product calculations, the decision function becomes

$$c(\boldsymbol{x}) = \operatorname{sign}\left[(\boldsymbol{w}\cdot\phi(\boldsymbol{x}_i)) - \rho\right] = \operatorname{sign}\left[\sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i) - \rho\right]. \tag{17}$$

This method thus creates a classification hyperplane characterized by $\boldsymbol{w}$ and ρ which has maximal distance from the origin in feature space $\mathcal{F}$ and separates all the data points from the origin. Another method is to create a circumscribing hypersphere around the data in feature space.

In this paper we have applied this approach combined with an evolutionary algorithm [18] for optimizing the maximal margin, as well as other SVM parameters, with respect to outlier detection accuracy.

4. Applying anomaly detection for offshore oil extraction turbomachines

Equipment control automation that includes sensors for monitoring equipment behavior and remote controlled valves to act upon undesired events is nowadays a common scenario in the modern offshore oil platforms. Oil plant automation physically protects plant integrity. However, it acts reacting to anomalous conditions. Extracting information from the raw data generated by the sensors, is not a simple task when turbomachinery is involved.

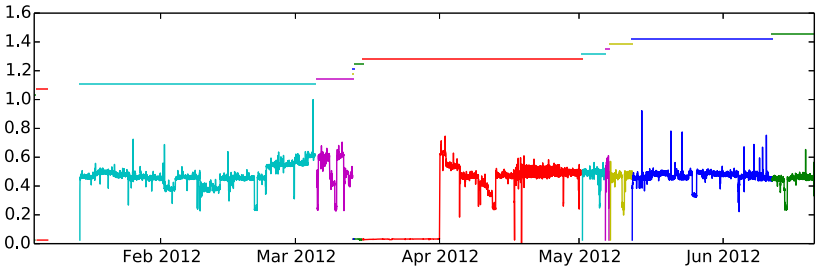


Fig. 4. Example of a training dataset where valid data chunks are marked in different colors. Data have been transformed for sensitivity reasons.

Any devices that extract energy from or import energy to a continuously moving stream of fluid (liquid or gas) can be called a turbomachine. Elaborating, a turbomachine is a power or head generating machine which employs the dynamic action of a rotating element, the rotor; the action of the rotor changes the energy level of the continuously flowing fluid through the machine. Turbines, compressors and fans are all members of this family of machines.

In contrast to Positive displacement machines especially of the reciprocating type which are low speed machines based on the mechanical and volumetric efficiency considerations, majority of turbomachines run at comparatively higher speeds without any mechanical problems and volumetric efficiency close to hundred percent.

The approach described in Section 3 was prompted by the complexity and requirements of the task of early detection of behaviors that could potentially lead to machine or platform failures in the application context of interest.

In order to experimentally study and validate our approach we carried out a study involving a real-world test case. In this case in particular we deal with a dataset of measurements taken with a five minutes frequency obtained during the first half of year 2012 from 64 sensors connected to an operational turbomachine. An initial analysis of the data yields that there are different profiles or patterns that are shared by different sensors. This is somewhat expected as sensors with similar purposes or supervising similar physical properties should have similar readings characteristics.

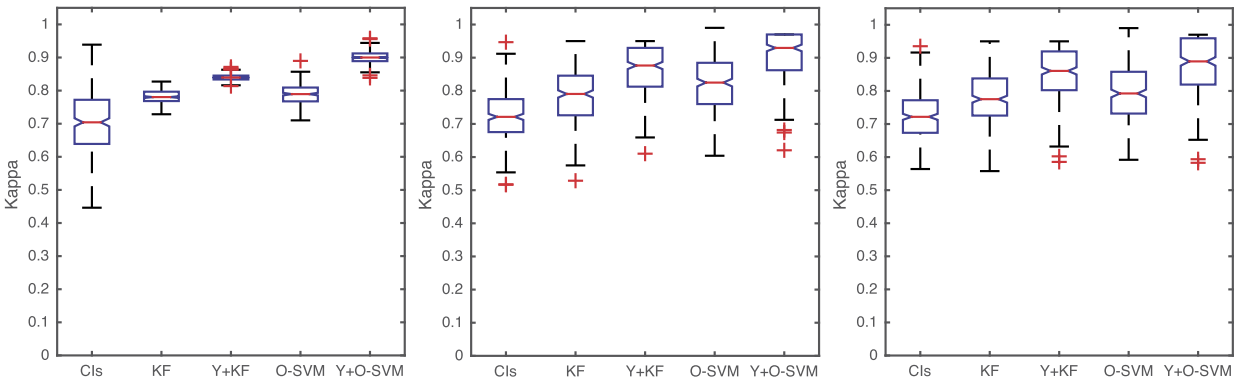
There are at least three time series profiles in the dataset. On one hand, we have smooth homogeneous time series that are generally associated with slow-changing physical properties. Secondly, we found fast changing/unstable sensor readings that could be a result of sensor noise or unstable physical quantity. There is a third class of time series which exhibit a clear change in operating profile attributable to different usage regimes of the machine or the overall extraction/processing process.

In order to provide a valid ground for comparison, we tested the method currently used by the platform operator, which is based on statistical confidence intervals [24], a one-class support vector machine-based classifier—as described earlier in this work—and our proposal. Problem data were transformed as to detect an anomaly based on consecutive sensor measurements in one hour.

The approach in current use was not (and can not be) fully disclosed, as it is business sensitive information. However, in broad terms, for each sensor, this method receives a sample data chunk, which has been selected by an expert as a valid one (see Fig. 4 for an example). It filters out outlier elements and computes the confidence intervals at a predefined percent of the resulting dataset. A possible failure is detected when a given set of sensor measurements is consistently outside such interval.

Similarly, in order to provide better grounds for comparison we also include in the comparative study the outlier detection method using Kalman filters as described in Section 2.3.

All of these approaches can be said to be of an unsupervised learning nature, as they do not require to have labeled data. However, in order to evaluate the quality of the methods in anomaly detection it was necessary to prepare a test dataset that contains regular and anomalous data. We carried out that task by



(a) Errors for homogeneous series. (b) Errors for multi-modal series. (c) Errors for noisy series.

Fig. 5. Box plots of the Kappa statistic yielded by confidence interval (CIs), Kalman filters (KF), Kalman filters coupled with YASA (Y + KF), one-class SVMs (O-SVM) and one-class SVMs coupled with YASA (Y + O-SVM) when faced with each subset of the dataset.

Table 1
Confusion matrices regarding the detection of anomalies (Anom) and 'normal' states (OK) yielded by each of the approaches involved in the experiments using all available data.

		Predicted									
		Confidence intervals		Kalman filter		YASA + Kalman		One-class SVM		YASA + o-c SVM	
		Anom.	OK	Anom.	OK	Anom.	OK	Anom.	OK	Anom.	OK
Actual	Anom.	746	534	731	549	1046	234	789	491	1099	181
	OK	486	794	289	991	180	1037	413	867	159	1121

creating a test data set, which contained 20 anomaly instances extracted from each of the 64 time series and 20 regular or non-anomalous situations.

The need for comparing the performance of the algorithms when confronted with the different sensor data prompts the use of statistical tools in order to reach a valid judgment regarding the quality of the solutions, how different algorithms compare with each other and their computational resource requirements. Box plots [7] are one of such representations and have been repeatedly applied in our context. Although box plots allow a visual comparison of the results, in principle, some conclusions could be deduced out of them.

Fig. 5 shows the quality of the results in terms of the Kappa statistic [9] obtained from each algorithm in the form of box plots. We have grouped the results according to the class of sensor data for the sake of a more valuable presentation of results. These results are also detailed in Table 1.

The statistical validity of the judgment of the results calls for the application of statistical hypothesis tests [29]. The McNemar test [22] is particularly suited for the assessment of classification problem results, like ones addressed here. This test is a normal approximation used on paired nominal data. It is applied to 2×2 contingency tables with a dichotomous trait, with matched pairs of subjects, to determine whether the row and column marginal frequencies are equal. In our case, we applied the test using to the confusion matrices performing pair-wise tests on the significance of the difference of the indicator values yielded by the executions of the algorithms. A significance level, α, of 0.05 was used for all tests.

Table 2 contains the results of the statistical analysis which confirm the judgments put forward before. There is clearly confirmed that the proposed approach, that is, the combined used of YASA and one-class SVMs, produced statistically significant better results than the other approaches.

5. Final remarks

In this work we showed that the combination of a novel online segmentation method specially devised to deal with massive or big data problems with a one-class support vector machine is able to correctly identify

Table 2
Results of the McNemar statistical hypothesis tests. Green cells (+) denote cases where the algorithm in the row statistically was better than the one in the column. Cells marked in red (−) are cases where the method in the column yielded statistically better results when compared to the method in the row. Finally, cells in blue (~) denote cases where results from both methods were statistically indistinguishable.

	Y+OSVM	*OSVM*	*Y+KF*	*KF*	*CIs*
Homogeneous series					
YASA + One-class SVM (Y+OSVM)	·	~	~	+	+
One-class SVM (OSVM)		·	~	+	+
YASA + Kalman filter (Y+KF)			·	+	+
Kalman filter (KF)				·	~
Confidence intervals (CIs)					·
Multi-modal series					
YASA + One-class SVM (Y+OSVM)	·	+	+	+	+
One-class SVM (OSVM)		·	−	+	~
YASA + Kalman filter (Y+KF)			·	+	+
Kalman filter (KF)				·	~
Confidence intervals (CIs)					·
Noisy series					
YASA + One-class SVM (Y+OSVM)	·	+	+	+	+
One-class SVM (OSVM)		·	−	+	~
YASA + Kalman filter (Y+KF)			·	+	+
Kalman filter (KF)				·	~
Confidence intervals (CIs)					·
All data					
YASA + One-class SVM (Y+OSVM)	·	+	+	+	+
One-class SVM (O-SVM)		·	+	+	+
YASA + Kalman filter (Y+KF)			·	+	+
Kalman filter (KF)				·	~
Confidence intervals (CIs)					·

and detect anomalies yielding a substantially better results that current state of the art approaches. We have compared this algorithm applying it to the identification of anomalies in turbomachines used as part of offshore oil extraction and processing plants. In the problem under study, our approach was able to outperform the current approach used in the production system as well as the traditional formulation of a one-class SVM and Kalman filter-based approaches.

It is also relevant to underscore the role of the segmentation algorithm. Both our proposal and the Kalman filter yielded substantially better results when coupled with YASA. This could be used as a starting for further developments in this topic.

A computational system—whose essential formulation is the method described in this paper—is currently deployed by a major petroleum industry conglomerate of Brazil.

Further work in this direction is called for and is currently being carried out. An important direction is the formal understanding of the computational complexity of the proposal. We also intend to extend the context of application to other big data application contexts.

Acknowledgements

This work was partially funded by MINECO Project TEC2012-37832-C02-01, CNPq BJT Project 407851/2012-7, CNPq PVE Project 314017/2013-5 and FAPERJ APQ1 Projects 213969 and 214411.

References

[1] D. Barbara, N. Wu, S. Jajodia, Detecting novel network intrusions using Bayes estimators, in: First SIAM Conference on Data Mining, SIAM, 2001.
[2] E. Bingham, A. Gionis, N. Haiminen, H. Hiisilä, H. Mannila, E. Terzi, Segmentation and dimensionality reduction, in: Proceedings of the 2006 SIAM International Conference on Data Mining, SIAM, 2006, pp. 372–383.
[3] B. Bollobás, G. Das, D. Gunopulos, H. Mannila, Time-series similarity problems and well-separated geometric sets, in: Proceedings of the Thirteenth Annual Symposium on Computational Geometry, ACM, 1997, pp. 454–456.
[4] M.L. Borrajo, B. Baruque, E. Corchado, J. Bajo, J.M. Corchado, Hybrid neural intelligent system to predict business failure in small-to-medium-size enterprises, Int. J. Neural Syst. 21 (04) (2011) 277–296.
[5] M.M. Breunig, H.P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, ACM, New York, NY, USA, 2000, pp. 93–104.
[6] J.L. Calvo-Rolle, E. Corchado, A bio-inspired knowledge system for improving combined cycle plant control tuning, Neurocomputing 126 (2014) 95–105.
[7] J. Chambers, W. Cleveland, B. Kleiner, P. Tukey, Graphical Methods for Data Analysis, Wadsworth, Belmont, 1983.
[8] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. 41 (3) (2009) 15.
[9] B. Di Eugenio, M. Glass, The Kappa statistic: a second look, Comput. Linguist. 30 (1) (2004) 95–101, http://dx.doi.org/10.1162/089120104773633402.
[10] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, A geometric framework for unsupervised anomaly detection, in: Applications of Data Mining in Computer Security, Springer, 2002, pp. 77–101.
[11] R. Fujimaki, T. Yairi, K. Machida, An approach to spacecraft anomaly detection problem using kernel feature space, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, ACM, 2005, pp. 401–410.
[12] F.E. Grubbs, Procedures for detecting outlying observations in samples, Technometrics 11 (1) (1969) 1–21.
[13] J. Hunter, N. McIntosh, Knowledge-based event detection in complex time series data, in: Artificial Intelligence in Medicine, Springer, 1999, pp. 271–280.
[14] R.E. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1960) 35–45.
[15] E. Keogh, S. Lonardi, B.Y.-c. Chiu, Finding surprising patterns in a time series database in linear time and space, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2002, pp. 550–556.
[16] S. King, D. King, K. Astley, L. Tarassenko, P. Hayton, S. Utete, The use of novelty detection techniques for monitoring high-integrity plant, in: Proceedings of the 2002 International Conference on Control Applications, vol. 1, 2002, IEEE, 2002, pp. 221–226.
[17] D. Lemire, A better alternative to piecewise linear time series segmentation, in: Proceedings of the 2007 SIAM International Conference on Data Mining, SIAM, 2007.
[18] L. Martí, Scalable Multi-objective Optimization, PhD thesis, Departmento de Informática, Universidad Carlos III de Madrid, Colmenarejo, Spain, 2011.
[19] L. Martí, N. Sanchez-Pi, J.M. Molina, A.C. Bicharra Garcia, YASA: yet another time series segmentation algorithm for anomaly detection in big data problems, in: Hybrid Artificial Intelligence Systems: 9th International Conference (HAIS 2014), Springer International Publishing, 2014, pp. 697–708.
[20] L. Martí, N. Sanchez-Pi, J.M. Molina, A.C.B. Garcia, Anomaly detection based on sensor data in petroleum industry applications, Sensors 15 (2) (2015) 2774–2797.
[21] P.S. Maybeck, Stochastic Models, Estimation, and Control, Math. Sci. Eng., vol. 141, Academic Press, 1979.
[22] Q. McNemar, Note on the sampling error of the difference between correlated proportions or percentages, Psychometrika 12 (2) (1947) 153–157, http://dx.doi.org/10.1007/BF02295996.
[23] M.M. Moya, D.R. Hush, Network constraints and multi-objective optimization for one-class classification, Neural Netw. 9 (3) (1996) 463–474, http://dx.doi.org/10.1016/0893-6080(95)00120-4.
[24] J. Neyman, Outline of a theory of statistical estimation based on the classical theory of probability, Philos. Trans. R. Soc. A 236 (1937) 333–380.
[25] S. Papadimitriou, H. Kitagawa, P. Gibbons, C. Faloutsos, LOCI: fast outlier detection using the local correlation integral, in: Proceedings of the 19th International Conference on Data Engineering (ICDE'03), IEEE Press, 2003, pp. 315–326.
[26] G. Ratsch, S. Mika, B. Scholkopf, K. Muller, Constructing boosting algorithms from SVMs: an application to one-class classification, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1184–1199.
[27] H. Ringberg, A. Soule, J. Rexford, C. Diot, Sensitivity of PCA for traffic anomaly detection, in: ACM SIGMETRICS Perform. Eval. Rev., vol. 35, ACM, 2007, pp. 109–120.
[28] V. Roth, Outlier detection with one-class kernel Fisher discriminants, in: Adv. Neural Inf. Process. Syst., vol. 17, MIT Press, 2005, pp. 1169–1176.
[29] S.L. Salzberg, On comparing classifiers: pitfalls to avoid and a recommended approach, Data Min. Knowl. Discov. 1 (3) (1997) 317–328, http://dx.doi.org/10.1023/A:1009752403260.
[30] N. Sanchez-Pi, L. Martí, J.M. Molina, A.C. Bicharra Garcia, High-level information fusion for risk and accidents prevention in pervasive oil industry environments, in: Highlights of Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection: PAAMS 2014 International Workshops, Salamanca, Spain, June 4–6, 2014, Springer International Publishing, 2014, pp. 202–213.
[31] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, J.C. Platt, Support vector method for novelty detection, in: NIPS Conference, Denver, Colorado, USA, November 29–December 4, 1999, in: Adv. Neural Inf. Process. Syst., vol. 12, The MIT Press, 1999, pp. 582–588.

[32] M. Shuai, K. Xie, G. Chen, X. Ma, G. Song, A Kalman filter based approach for outlier detection in sensor networks, in: 2008 International Conference on Computer Science and Software Engineering, IEEE, 2008, pp. 154–157.
[33] J.-A. Ting, E. Theodorou, S. Schaal, A Kalman filter for robust outlier detection, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE Press, 2007, pp. 1514–1519.
[34] M. Vlachos, J. Lin, E. Keogh, D. Gunopulos, A wavelet-based anytime algorithm for k-means clustering of time series, in: Proc. Workshop on Clustering High Dimensionality Data and Its Applications, Citeseer, 2003.
[35] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17.