YASA: Yet Another Time Series Segmentation Algorithm for Anomaly Detection in Big Data Problems

Luis Martí[1], Nayat Sanchez-Pi[2], José Manuel Molina[3], and Ana Cristina Bicharra Garcia[4]

[1] Dept. of Electrical Engineering, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro (RJ) Brazil.
lmarti@ele.puc-rio.br
[2] Instituto de Lógica, Filosofia e Teoria da Ciéncia (ILTC), Niterói (RJ) Brazil.
nayat@iltc.br
[3] Dept. of Informatics, Universidad Carlos III de Madrid, Colmenarejo, Madrid, Spain.
molina@ia.uc3m.es
[4] ADDLabs, Fluminense Federal University. Niterói (RJ) Brazil.
cristina@addlabs.uff.br

Abstract. Time series patterns analysis had recently attracted the attention of the research community for real-world applications. Petroleum industry is one of the application contexts where these problems are present, for instance for anomaly detection. Offshore petroleum platforms rely on heavy turbomachines for its extraction, pumping and generation operations. Frequently, these machines are intensively monitored by hundreds of sensors each, which send measurements with a high frequency to a concentration hub. Handling these data calls for a holistic approach, as sensor data is frequently noisy, unreliable, inconsistent with *a priori* problem axioms, and of a massive amount. For the anomalies detection problems in turbomachinery, it is essential to segment the dataset available in order to automatically discover the operational regime of the machine in the recent past. In this paper we propose a novel time series segmentation algorithm adaptable to big data problems and that is capable of handling the high volume of data involved in problem contexts. As part of the paper we describe our proposal, analyzing its computational complexity. We also perform empirical studies comparing our algorithm with similar approaches when applied to benchmark problems and a real-life application related to oil platform turbomachinery anomaly detection.

Keywords: Time series segmentation, anomaly detection, big data, oil industry application

1 Introduction

The problem of finding patterns in data that do not conform to an expected behavior, is known as the anomaly detection problem[1]. Hence, unexpected patterns or instances are often referred as anomalies [2], outliers [3], faults [4] –just to mention a few– depending on the application domain.

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.

One of such cases is the detection of anomalies in turbomachinery installed in off-shore petroleum extraction platforms from a centralized company control hub. Recent history shows us how important a correct handling of these equipment is as failures in this industry has a dramatic economical, social and environmental impact.

Dealing with this problem calls for a comprehensive approach, as sensor data is frequently noisy, unreliable, inconsistent with a priori problem axioms. Furthermore, the amount of data to process is frequently vast upon as one platform has several turbomachines, that, on average, are monitored by more than 250 sensors, which are sampled at a relatively high-frequency.

Therefore, in this case, we are also facing a big data problem as the idea is to run a detection analysis over these data in an online fashion. In terms of social goods, big data uses concepts from non-linear system identification to reveal interesting patterns about anomaly events, energy usage and mechanical performance which can potentially help performing predictions of outcomes and behaviors to reduce fuel costs, maintenance costs, and improve safety.

One additional characteristic of this problem is these machines have different operational profiles. For example, they are used at different intensities or throttle depending on the platform exploitation profile. Therefore, in order to correctly detect future anomalies it is essential to segment the dataset available in order to automatically discover the operational regime of the machine in the recent past.

Time series segmentation [5] methods can be classified as explicit, implicit, or hybrid. Implicit methods produce high quality segmentation, but are slow. This type of segmentation method is one in which the application phase calculates the error of a given segmentation. The error is passed back to the segmentation phase and is then used to improve the segmentation. On the other hand, the explicit methods are fast but they produce lower quality segmentation results. The need of a fast and quality method for real-time applications became the motivation of this work.

In this work, we propose a fast and high quality segmentation algorithm to improve results in the anomaly detection problem that is currently used in the oil extraction platform supervision problem described above. The remainder of this paper is organized as following. In the next section, we discuss some related work. Subsequently, we describe our segmentation algorithm proposal in detail.

After that, we present a case study for offshore oil platform turbomachinery sensor data segmentation. This case study is used to empirically compare our approach with current state-of-the-art alternatives in terms of segmentation accuracy and computational cost. Finally on Section six, some conclusive remarks and directions for future work are presented.

2 Foundations

In the problem of finding frequent patterns, the primary purpose of time series segmentation is dimensionality reduction. For the anomalies detection problems in turbomachineries, 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. But before start citing them, we state a segmentation definition and describe the available segmentation method classification.

In general terms, a time series can be expressed as a set of time-ordered possibly infinite measurements [6], $\mathcal{S}$, such that,

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

In practice, time series frequently have a simpler definition as measurements 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

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

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

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 [7, 8] top-down [9], and bottom-up [10] strategies. The sliding windows method is a purely implicit segmentation technique. It consists of a segment is grown until it exceeds some error bound. This process is repeated with the next data point not included in the newly approximated segment.

However, like all implicit methods, it is extremely slow and not useful for real-time applications, its complexity is $O(Ln)$. Top-down methods are those where the time series is recursively partitioned until some stopping criteria is met. This method is faster than the sliding window method above, but it is

still slow, the complexity is $O(n^2K)$. And the bottom-up starts from the finest possible approximation and segments are merged until some stopping criteria is met. It produces similar results to top-down algorithms but are faster, $O(Ln)$.

Later, during the process of approximating a time series with straight lines, there are at least two ways of finding the approximating line: linear interpolation and linear regression [11]. Linear interpolation tends to closely align the endpoint of consecutive segments, giving the piecewise approximation a "smooth" look. In contrast, piecewise linear regression can produce a very disjointed look on some datasets. However, the quality of the approximating line, in terms of Euclidean distance, is generally better in the regression approach [5].

There also, more novel methods for instance those using clustering for segmentation. The clustered segmentation problem is clearly related with the time series clustering problem [12] and there are also several definitions for time series [13, 14]. One natural view of segmentation is the attempt to determine which components of a data set naturally "belong together".

There exist two classes of algorithms for solving the clustered segmentation problem: distance-based clustering of segmentations that measure distance between sequence segmentations and we employ a standard clustering algorithm (e.g.,k-means) on the pair-wise distance matrix. The second class consists of two randomized algorithms that cluster sequences using segmentations as "centroids". In particular, we use the notion of a distance between a segmentation and a sequence, which is the error induced to the sequence when the segmentation is applied to it. The algorithms of the second class treat the clustered-segmentation problem as a model selection problem and they try to find the best model that describes the data.

There also methods considering multiple regression models.In [15] it is considered a segmented regression model with one independent variable under the continuity constraints and studied the asymptotic distributions of the estimated regression coefficients and change-points. In [16–18] is considered some special cases of the model studied cited before, and provided more details on distributional properties of the estimators.

Bai [19–21] considered a multiple regression model with structural changes, the model without the continuity constraints at the change-points, and studied the asymptotic properties of the estimators.

3 YASA: Yet Another Segmentation Algorithm

In this section we introduce a novel and fast algorithm for time series segmentation. Besides the obvious purposed of obtaining a segmentation method that produces low approximation errors another set of guidelines were observed while devising it. They can be summarized as:

– *Low computational cost*: The application context calls for algorithms capable of handling large amounts of data and that scale properly as the those amounts are increased. Most current segmentation algorithms have such a

computational complexity that impairs them to correctly tackle the problems of interest.
- *Easy parameterization*: one important drawback of current approaches is that their parameters may be hard so set by end users. In our case we have as main parameter is the significance test threshold, which is a very good understood and easy to gasp feature.

The YASA algorithm is presented in Figure 1 in schematic 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. A call to the 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 models 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. It also warranties that that time instant does not creates a too-short time series chunk. Once an adequate time instant is located located and used as split point to carry out the segmentation the parts of the time series located at both sides of it.

4 Case Study in Offshore Oil Process Plant

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.

Turbomachinery, in mechanical engineering, describes machines that transfer energy between a rotor and a fluid, including both turbines and compressors [22]. While a turbine transfers energy from a fluid to a rotor, a compressor transfers energy from a rotor to a fluid. The two types of machines are governed by the same basic relationships including Newton's second Law of Motion and Euler's energy equation for compressible fluids. Centrifugal pumps are also turbomachines that transfer energy from a rotor to a fluid, usually a liquid, while turbines and compressors usually work with a gas.

Any devices that extracts energy from or imparts 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

function SEGMENTDATA($\mathcal{S}^{(j)}_{t_{\max},t_0}$, $\rho_{\min}$, $l_{\max}$, $s_{\min}$, 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.
▷ $l_{\max} > 0$, maximum levels of recursive calls.
▷ $s_{\min} > 0$, minimum segment length.
Returns:
▷ $\Phi := \{\phi_1, \ldots, \phi_m\}$, data segments.
  if $l = l_{\max}$ **then**
    return $\Phi = \left\{\mathcal{S}^{(j)}_{t_{\max},t_0}\right\}$
  end if
  Perform linear regression,

$$\{m, b\} \leftarrow \text{LINEARREGRESSION}(\mathcal{S}^{(j)}_{t_{\max},t_0}).$$

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

$$\{e_0, \ldots, e_{\max}\} = \text{RESIDUALS}(\mathcal{S}^{(j)}_{t_{\max},t_0}, m, b)$$

  $t_s \leftarrow t_0$.
  while $\max(\{e_0, \ldots, e_{\max}\}) > 0$ **and** $t_s \notin (t_0 + s_{\min}, t_{\max} - s_{\min})$ **do**
    Determine split point, $t_s = \arg\max_t \{e_t\}$.
  end while
  if $t_s \in (t_0 + s_{\min}, t_{\max} - s_{\min})$ **then**
    $\Phi_{\text{left}} = \text{SEGMENTDATA}(\mathcal{S}^{(j)}_{t_s,t_0}, \rho_{\min}, l_{\max}, s_{\min}, l + 1)$.
    $\Phi_{\text{right}} = \text{SEGMENTDATA}(\mathcal{S}^{(j)}_{t_{\max},t_s}, \rho_{\min}, l_{\max}, s_{\min}, l + 1)$.
    return $\Phi = \Phi_{\text{left}} \cup \Phi_{\text{right}}$.
  end if
  return $\Phi = \left\{\mathcal{S}^{(j)}_{t_{\max},t_0}\right\}$.
end function

Fig. 1: Pseudocode of the proposed algorithm.

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 per cent.

Turbomachines can be categorized on the basis of the direction of energy conversion:

- Absorb power to increase the fluid pressure or head (ducted Fans, compressors and pumps).
- Produce power by expanding fluid to a lower pressure or head (hydraulic, steam and gas turbines).

4.1 Problem Formalization

Assuming independence between turbomachines we can deal with each one separately. Although, in practice, different machines do affect each other, as they are interconnected, for the sake of simplicity we will be dealing with one at a time.

Using that scheme we can construct an abstract model of the problem. A given turbomachine, $\mathcal{M}$, is monitored by a set of m sensors $s^{(j)} \in \mathcal{M}$, with $j = 1, \ldots, m$. Each of these sensors are sampled at regular time intervals in order to produce the time series

$$\mathcal{S}^{(j)}_{t_{\max},t_0} := \left\{ s^{(j)}_t \right\}, t_0 \leq t \leq t_{\max} \, . \tag{3}$$

Using this representation and assuming that sensors are independent, the problem of interest can be expressed as a two-part problem: (i) predict a future anomaly in a sensor, and; (ii) decision making from anomaly predictions. This can be expressed more formally as:

Definition 1 (Sensor Anomaly Prediction). *Find a set of* anomaly prediction functions*,* $A^{(j)}(\cdot)$*, such that*

$$A^{(j)}\left(\mathcal{S}^{(j)}_{t,t-\Delta t} \middle| \widehat{\mathcal{S}}^{(j)}_{t_{\max},t_0} \right) = \begin{cases} 1 \textit{ predicted anomaly} \\ \\ 0 \textit{ in other case} \end{cases} , \tag{4}$$

that is constructed using a given reference (training) set of sensor data, $\widehat{\mathcal{S}}^{(j)}_{t_{\max},t_0}$*, and determines if there will be a failure in the near future by processing a sample of current sensor data* $\mathcal{S}^{(j)}_{t,t-\Delta t}$*, with* $t_{\max} < t - \Delta t < t$ *and, generally,* $\Delta t \ll t_{\max} - t_0$*.*

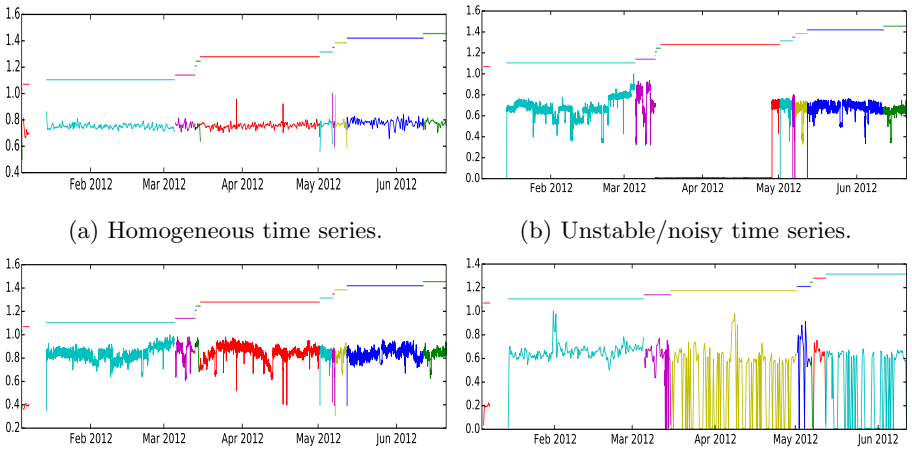
Using those functions the second problem can be stated as:

Definition 2 (Machine Anomaly Alarm). *For each turbomachine* $\mathcal{M}$*, obtain a* machine alarm function

$$F_{\mathcal{M}}\left(a^{(1)}_t, \ldots, a^{(m)}_t \middle| \boldsymbol{w}_{\mathcal{M}} \right) = \begin{cases} 1 \textit{ alarm signal} \\ \\ 0 \textit{ in other case} \end{cases} , \tag{5}$$

where $a^{(j)}_t = A^{(j)}\left(\mathcal{S}^{(j)}_{t,t-\Delta t} \right)$ *and the weights vector,* $\boldsymbol{w}_{\mathcal{M}} = \left\{ w^{(1)}, \ldots, w^{(m)} \right\}$ *represents the contribution –or relevance– of each sensor to an alarm firing decision.*

It must be noted that, although we have expressed these problems in a crisp (Boolean) form they can be expressed in a continuous [0, 1] form suitable for application of fuzzy logic or other forms of uncertainty reasoning methods.



(a) Homogeneous time series.

(b) Unstable/noisy time series.

(c) Multi-modal series

(d) Highly unstable time series, probably caused by faulty sensors.

Fig. 2: A sample of the four main types of time series contained in the dataset. We have marked with color changes the moments in which the machine was switched on/off.

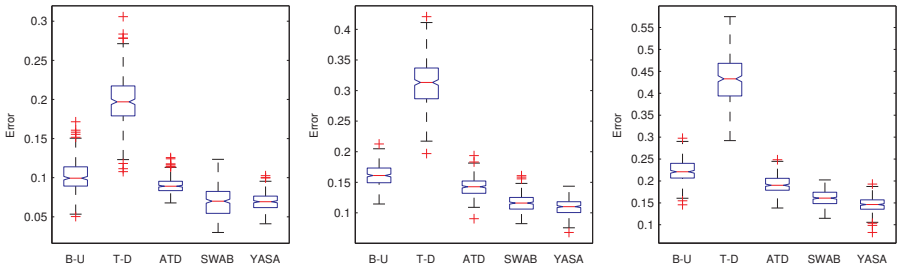
In order to synthesize adequate $A^{(j)}$ and $F_{\mathcal{M}}$ it is necessary to identify the different operational modes of the the machine. Knowing the operational modes of the machine enables the creation of $A^{(j)}$ and $F_{\mathcal{M}}$ functions —either explicitly or by means of a modelling or machine learning method— that correctly responds to each of modes.

4.2 Comparative Experiments

YASA is been currently applied with success to the problem of segmenting turbomachine sensor data of a major petroleum extraction and processing conglomerate of Brazil. In this section we present an part of the experimental comparison involving some of the current state-of-the-art methods and our proposal that was carried out in order to validate the suitability of our approach. Readers must be warned that the results presented here had to be transformed in order to preserve the sensitive details of the data.

In this case in particular we deal with a dataset of measurements taken with a five minute frequency obtained during the first half of 2012 from more than 250 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.

Figure 2 displays the four shared time series profiles found in the dataset. On hand hand, we have smooth homogeneous time series that are generally as-



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

Fig. 3: Box plots of the root mean squared errors yielded by the Bottom-Up (B-U), Top-Down (T-D), adaptive Top-Down (ATD), Sliding Window and Bottom-up (SWAB) and our proposal (YASA).Data has been transformed for sensitivity reasons.

sociated 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.

Using this dataset we carried out an study comparing four of the main segmentation algorithms and our proposal. In particular we compare the Bottom-Up [10], Top-Down [23], adaptive Top-Down [9] and Sliding Window and Bottom-up algorithms [5].

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 judgement regarding the quality of the solutions, how different algorithms compare with each other and their computational resource requirements. Box plots [24] are one of such representations and have been repeatedly applied in our context. Although box plots allows a visual comparison of the results and, in principle, some conclusions could be deduced out of them.

Figure 3 shows the quality of the results in terms of the mean squared error obtained from the segmentation produced by 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. The main conclusion to be extracted from this initial set of results is that our proposal was able to achieve a similar performance –and in some cases a better performance– when compared with the other methods.

The statistical validity of the judgment of the results calls for the application of statistical hypothesis tests. It has been previously remarked by different authors that the Mann–Whitney–Wilcoxon U test [25] is particularly suited for experiments of this class. This test is commonly used as a non-parametric method for testing equality of population medians. In our case we performed pair-wise

Table 1: Results of the statistical hypothesis tests. Cells marked in red are cases where no statistically significant difference was observed. Green cells mark cases where results of both algorithms was statistically homogeneous.

(a) Tests on the segmentation errors.

	T-D	B-U	ATD	SWA	YAS
Homogeneous series					
Top-Down	·	−	+	+	+
Bottom-Up		·	−	−	−
Adaptive T-D			·	−	+
SWAB				·	+
YASA					·
Multi-modal series					
Top-Down	·	−	+	+	+
Bottom-Up		·	−	−	−
Adaptive T-D			·	−	+
SWAB				·	+
YASA					·
Noisy series					
Top-Down	·	+	−	+	+
Bottom-Up		·	−	−	−
Adaptive T-D			·	−	+
SWAB				·	+
YASA					·
All data					
Top-Down	·	+	−	+	+
Bottom-Up		·	−	−	−
Adaptive T-D			·	−	+
SWAB				·	+
YASA					·

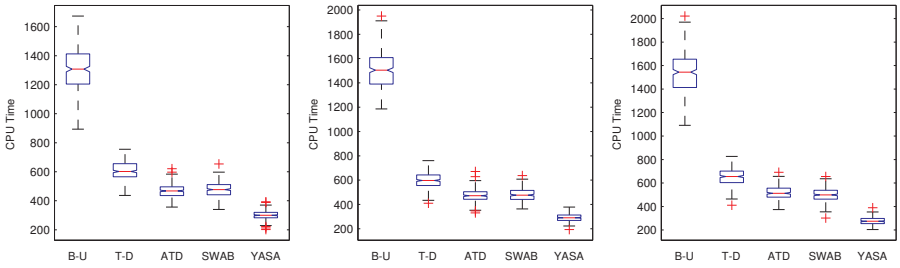
(b) Tests on the CPU time required.

	T-D	B-U	ATD	SWA	YAS
Homogeneous series					
Top-Down	·	−	−	−	−
Bottom-Up		·	−	−	−
Adaptive T-D			·	+	−
SWAB				·	−
YASA					·
Multi-modal series					
Top-Down	·	−	−	−	−
Bottom-Up		·	−	−	−
Adaptive T-D			·	+	−
SWAB				·	−
YASA					·
Noisy series					
Top-Down	·	−	−	−	−
Bottom-Up		·	−	−	−
Adaptive T-D			·	+	−
SWAB				·	+
YASA					·
All data					
Top-Down	·	−	−	−	−
Bottom-Up		·	−	−	−
Adaptive T-D			·	+	−
SWAB				·	−
YASA					·

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 1a contains the results of the statistical analysis which confirm the judgements put forward before.

Comparing performance is clearly not enough as one of the leit motifs of this work is to provide a good and fast segmentation algorithm. That is why we carry out a similar study to the previous one, this time focusing on the amount of CPU time required by each algorithm. Figure 4 summarizes this analysis. It is visible how our approach required less computation to carry out the task. Table 1b allows to assert this analysis with the help of statistical hypothesis tests, as explained in the previous analysis.



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

Fig. 4: Box plots of the CPU time needed by the Bottom-Up (B-U), Top-Down (T-D), adaptive Top-Down (ATD), Sliding Window and Bottom-up (SWAB) and our proposal (YASA). Data has been transformed for sensitivity reasons.

5 Final Remarks

In this work we introduced a novel segmentation online segmentation method specially devised to deal with massive or big data problems. We have applied this algorithm to the segmentation sensor measurements of turbomachines used as part of offshore oil extraction and processing plants. In the problem under study, our approach was able to yield adequate results at a lower computational cost.

Although we have introduced and presented the YASA algorithm focusing of the segmentation problem itself, it must be pointed out that the algorithm is currently deployed as part of a larger system that rely of the segmentation to train a set of one-class support vector machine classifiers capable. These classifiers are used to detect anomalies in turbomachinery platform operation. The global system is currently in use by a major petroleum industry conglomerate of Brazil and is to be presented as a whole in a forthcoming paper.

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.

References

1. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys (CSUR) **41**(3) (2009) 15
2. DeCoste, D.: Mining multivariate time-series sensor data to discover behavior envelopes. In: KDD. (1997) 151–154
3. Hawkins, D.M.: Identification of outliers. Volume 11. Springer (1980)
4. Yairi, T., Kato, Y., Hori, K.: Fault detection by mining association rules from house-keeping data. In: Proc. of International Symposium on Artificial Intelligence, Robotics and Automation in Space. Volume 3., Citeseer (2001)

5. Keogh, E., Chu, S., Hart, D., Pazzani, M.: Segmenting time series: A survey and novel approach. Data mining in time series databases **57** (2004) 1–22
6. Bouchard, D.: Automated time series segmentation for human motion analysis. Center for Human Modeling and Simulation, University of Pennsylvania (2006)
7. Bingham, E., Gionis, A., Haiminen, N., Hiisilä, H., Mannila, H., Terzi, E.: Segmentation and dimensionality reduction. In: SDM, SIAM (2006)
8. Terzi, E., Tsaparas, P.: Efficient algorithms for sequence segmentation. In: SDM, SIAM (2006)
9. Lemire, D.: A better alternative to piecewise linear time series segmentation. In: SDM, SIAM (2007)
10. Hunter, J., McIntosh, N.: Knowledge-based event detection in complex time series data. In: Artificial Intelligence in Medicine. Springer (1999) 271–280
11. Shatkay, H., Zdonik, S.B.: Approximate queries and representations for large data sequences. In: Data Engineering, 1996. Proceedings of the Twelfth International Conference on, IEEE (1996) 536–545
12. Vlachos, M., Lin, J., Keogh, E., Gunopulos, D.: A wavelet-based anytime algorithm for k-means clustering of time series. In: In Proc. Workshop on Clustering High Dimensionality Data and Its Applications, Citeseer (2003)
13. Bollobás, B., Das, G., Gunopulos, D., Mannila, H.: Time-series similarity problems and well-separated geometric sets. In: Proceedings of the thirteenth annual symposium on Computational geometry, ACM (1997) 454–456
14. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. Volume 23. ACM (1994)
15. Feder, P.I.: On asymptotic distribution theory in segmented regression problems–identified case. The Annals of Statistics (1975) 49–83
16. Hinkley, D.V.: Inference about the intersection in two-phase regression. Biometrika **56**(3) (1969) 495–504
17. Hinkley, D.V.: Inference in two-phase regression. Journal of the American Statistical Association **66**(336) (1971) 736–743
18. Hušková, M.: Estimators in the location model with gradual changes. Comment. Math. Univ. Carolin **39**(1) (1998) 147–157
19. Bai, J.: Estimation of a change point in multiple regression models. Review of Economics and Statistics **79**(4) (1997) 551–563
20. Bai, J., Perron, P.: Estimating and testing linear models with multiple structural changes. Econometrica (1998) 47–78
21. Bai, J., Perron, P.: Computation and analysis of multiple structural change models. Journal of applied econometrics **18**(1) (2003) 1–22
22. Logan Jr, E.: Handbook of Turbomachinery. Second edition edn. CRC Press (2003)
23. Duda, R.O., Hart, P.E., et al.: Pattern classification and scene analysis. Volume 3. Wiley New York (1973)
24. Chambers, J., Cleveland, W., Kleiner, B., Tukey, P.: Graphical Methods for Data Analysis. Wadsworth, Belmont (1983)
25. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Annals of Mathematical Statistics **18** (1947) 50–60