Unsupervised Machine Learning for Anomaly Detection in Multivariate Time Series Data of a Rotating Machine from an Oil and Gas Platform

[1]Ilan Sousa Figueirêdo[1], Tássio Farias Carvalho[1], Wenisten Dandas da Silva[1], Lílian Lefol Nani Guarieiro[1], Alex Alisson Bandeira Santos[1], Leonildes Soares de Melo Filho[2], Ricardo Emmanuel Vaz Vargas[3], and Erick Giovani Sperandio Nascimento[*1]

[1]*Manufacturing and Technology Integrated Campus – SENAI CIMATEC Salvador, Bahia, Brazil.*
[2]*Repsol Sinopec Brazil – Rio de Janeiro, Rio de Janeiro, Brazil*
[3]*Petróleo Brasileiro S.A. – Vitória, Espiríto Santo, Brazil*

Abstract[2]

Deep Learning (DP) models have been successfully applied to detect and predict failures in rotating machines. However, these models are often based on the supervised learning paradigm and require annotated data with operational status labels (e.g. normal or failure). Furthermore, machine measurement data is not commonly labeled by industry because of the manual and specialized effort that they require. In situations where labels are nonexistent or cannot be developed, unsupervised machine learning has been successfully applied for pattern recognition in large and multivariate datasets. Thus, instead of experts labeling a large amount of structured and/or non-structured data instances (also referred to as Big Data), unsupervised machine learning allows the annotation of the dataset from the few underlying interesting patterns detected. Therefore, we evaluate the performance of six unsupervised learning algorithms for the identification of anomalous patterns from a turbogenerator installed and operating in an oil and gas platform. The algorithms were C-AMDATS, Luminol Bitmap, SAX-REPEAT, k-NN, Bootstrap, and Robust Random Cut Forest. The evaluation performance was quantitatively calculated with seven classification metrics. The C-AMDATS algorithm was able to effectively and better detect the anomalous patterns, and it presented an accuracy of 99%, which leverages the further development of supervised DL models.

Keywords: *Multivariate Time Series, Anomaly Detection, Pattern Recognition, Unsupervised Machine Learning, Rotating Machinery.*

[1] *Corresponding Author. E-mail address: ericksperandio@gmail.com.
[2] We would like to express our deeply felt gratefulness to Professor Dr. Davidson Moreira and Dr. Leandro Andrade for the comprehensive and detailed peer-editing of this document, as well as for the gentle alerts with regards to important issues.

1. Introduction

Over the past decade, advances in technologies and engineering expertise have extended the lifespan of machines and devices of the oil and gas industry. However, machine malfunction and unexpected breakdown are still a reality for many oil companies. For example, in September 2019, Petrobras reported that the P-50 floating production storage and offloading (FPSO) had its production preventively interrupted because of a rupture of the mooring system. Fortunately, there was no record of labor accident or environmental disaster, but it caused an average loss of 20,000 bbl/d (barrels of oil per day), which means U$1,200,000 bbl/d considering an average value of U$60/bbl at the time (PETROBRAS, 2019). Furthermore, Petrobras has estimated a production loss from failures events in the Operational Unit located in the Brazilian state of Espírito Santo during 2016 of 1,514,000 bbl, which corresponds to US$75.7 million considering an average value of US$50/bbl in this period. Despite production losses and the company's financial damage, failures in the oil industry can lead to serious catastrophes, such as the Macondo Incident in 2010 that unfortunately caused the deaths of 11 workers, the sinking of the Deepwater Horizon rig, and massive marine and coastal damage, which marked this incident as one of the largest environmental disasters in the US history (Vargas et al., 2019).

Nevertheless, oil and gas production systems are designed to operate uninterruptedly, as the operating cost of an FPSO is high (approximately U$ 250,000/month) (Perera et al., 2019). Thus, an intelligent maintenance management is crucial for the maintenance of operational activities under required levels of structural integrity and system security, as well for the prevention production losses, environmental accidents, and human casualties and for reduction maintenance costs. However, the maintenance of FPSO machinery can be challenging, since the entire system is in a highly hostile environment.

Predictive Maintenance (PdM) is a specific method of maintainability that provides the integrity of the machinery and establishes the main necessary intervention activities. It uses historical measurement data for early detection of machine deterioration, *i.e.*, long before the malfunction exceeds

the acceptable operating limits designed by the project. The method has gained prominence for providing great efficiency in the durability of the system and reducing maintenance costs and logistical footprints (Gombé et al., 2019).

The measurement data of the machine are acquired in real time by multi-sensor or periodically by inspection; both ways may indicate the health status of the machinery, but normally critical assets are monitored in real time. Thus, computation models designed to implement PdM capabilities can diagnose or predict failures based on a long record of historical data of machine health status. Therefore, with the diagnostic or prognostic information of failures inferred by these models, maintenance managers would have a greater basis in decision making to improve operational performance for scheduling maintenance, reduce unplanned repairs, and minimize downtime.

In theory, the shutdown of machinery only occurs when there is an evidence of deterioration; however, the volume of data generated is large, and conventional diagnostic methods rely mainly on specialized knowledge, which hindes artificial characteristics and other factors that seriously restrict the intelligent and automated development of predictive models (Zhu et al., 2019).

Artificial Intelligence (AI) is an area of research that is currently on the rise and has been used to develop computational models for the detection, diagnosis, and prognosis of failures in dynamic machinery. The main tasks are: (i) to detect the normal or abnormal condition of the machine operation, (ii) to detect incipient failure and identify its cause, and (iii) to predict the development trend of the failures. The goal is to increase the reliability and safety of maintenance in dynamic and complex systems (Jardine et al., 2006; Liu et al., 2018).

Machine learning (within the AI field) has promising tools for pattern recognition of historical data from machine measurement (Caggiano et al., 2019). The tools are conceptualized in two types of learning: (i) supervised and (ii) unsupervised, both being a viable paradigm for nonlinear data analysis (Wachowiak et al., 2019).

In supervised classification learning, the learner is given a stimulus, classifies it, and then it is provided with corrective feedback (Love, 2002). Nevertheless, supervised models are limited by the need to have an annotated dataset for the corrective feedback during the training process, *i.e.*, each data sample must have its corresponding target or label, which can be discrete or continuous and with single or multiple values. However, all of this generated data from machine monitoring sensors turns the annotation into an increasingly complex, harder and thus unfeasible task . Because it usually demands highly specialized human workforce, which is normally overloaded with demand and does not have time for this relevant activity. Thereby, despite all promising publications with supervised DP models for failure classification in rotating machinery (Li et al., 2020; Souza et al., 2021), there is a gap in supervised machine learning models,which points to a desire and opportunity to employ unlabeled data in unsupervised machine learning algorithms (Jati & Georgiou, 2019).

Unsupervised learning is a branch of machine learning that consists of finding interesting information from the raw data without the help of any targets or labels. For instance, dimensionality reduction and clustering are well-known categories of unsupervised learning (Chollet, 2018). These approaches can extract useful features to handle unlabeled data. Dimensionality reduction consists of reducing the high dimensional data, as it captures the essence of the data by projecting the data to a lower dimensional subspace. Clustering algorithms can be used to find natural groupings or patterns. Clustering approaches can also serve as an advantageous data-preprocessing step to identify homogeneous groups on which to build supervised models (Kakarla et al., 2021).

There is an expectation that unsupervised machine learning will prove to be similarly effective in situations where labels are expensive, impractical to collect, or where the prediction target is unknown during the training process (Zhu et al., 2019). Thus, unsupervised algorithms can be implemented directly on raw data with no need of previous training. They only depend on internal clustering of the data without prior knowledge (Torabi Jahromi et al., 2016). This feature is one of the main advantages of

unsupervised methods, especially in industrial environments (Yiakopoulos et al., 2011).

Unsupervised learning models have already shown good results for fialure recognition in rotating machinery. However, most researches studies in the literature have been limited to the analysis of only vibration data, *i.e.*, an univariate approach (Abid et al., 2020; Ben Ali et al., 2018; Durbhaka & Barani, 2016; Nguyen et al., 2019; Soualhi et al., 2014; Wang et al., 2019; Yiakopoulos et al., 2011), but there are other numerous useful data for the failure diagnosis of rotating machinery, including for example oil analysis, acoustic emission, pressure and temperature measurement, and microwave energy.

Critical rotating machinery is generally monitored by multi-sensors because of the probability of a more robust representation of the phenomena implicated. However, multivariate data presents a bigger challenge for the use of machine learning algorithms, as they must be able to process and correlate attributes in a greater amount of data (Figueirêdo et al., 2020).

In this paper, we explored a turbogenerator, which is a critical rotating machine for the power generation system of an FPSO and other systems of oil and gas industry, for instance: Liquefied Natural Gas (LNG), pipeline, refinery, and petrochemical (Parrella et al., 2019). It operates on a wide variety of gaseous and liquid fuels, being extremely useful for offshore electricity generation, since oil and gas platforms are usually located hundreds of miles from the coast (Goldmeer et al., 2018).

In this context, this work proposes a comparative performance assessment between six unsupervised machine learning algorithms for pattern recognition and anomaly detection in multivariate time-series data from the oil and gas industry. The goal is to reveal the capacity of unsupervised algorithms to assist experts in the task of data annotation and to leverage the supervised models. We use the labels of the data only for evaluation purposes. The data was acquired from a turbogenerator of an FPSO power generation system.

2. Unsupervised Learning Algorithms

As our goal is to compare the performance of six unsupervised machine learning algorithms, understanding the logic of each one of them can contribute to a correct parameterization of the algorithm and consequently its performance. Thus, we review the concepts and applications of well-developed unsupervised algorithms.

2.1. C-AMDATS

The Cluster-based Algorithm for Anomaly Detection in Time Series Using Mahalanobis Distance (C-AMDATS) is an unsupervised clustering algorithm designed for pattern recognition or anomaly detection in multivariable time series.

The algorithm starts by dividing the time-series T into a set of groups C of the same size τ, where each subset $C_k = \{t_a, t_{a+1}, \dots, t_b\}$ is sized $|C_k| = \tau$, *i.e.*, $b - a = \tau$ (except the last set, in cases where $|T|$ is not divisible by τ). Afterwards, the algorithm reconstructs C iteratively using its copy C'. For this, the algorithm uses $f(C')$ to determine which set in C is closest to t_i by Mahalanobis distance.

The function $f(C')$ computes the average of the time-series segments within each C'_k group or its centroid, $m_k = (t_a + t_{a+1}, \dots, t_b)/(b - a)$, and the distance , $d(t_i, m_k)$, from t_i for each m_k centroid, for $1 \leq k \leq |C'|$. Using these distances, the algorithm moves t_i from its current group in C to the group C_k, where k is the index of the group C'_k whose centroid m_k is the closest to t_i according to $d(t_i, m_k)$.

After the iterative reconstruction of C, the set of groups C is definitely formed, which consists of groups of samples C_k of different sizes and better groups according to their values and distributed along the time axis, which is due to the Mahalanobis function.

In general, clustering algorithms use the Euclidean distance, which tends to form groups with a circular shape (since it does not consider the variance of each dimension of the dataset). Thus, if the shape of the group is more

elongated on the x-axis or the y-axis, the general shape of the group will always be circular. However, this circular shape may not be adequate to represent the real shape of the group. Therefore, the C-AMDATS apply another distance to solve this problem. Eq. (1) presents the Mahalanobis distance formula.

$$d_m(x, \mu) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \quad (1)$$

Where: $d_m(x, \mu)$ is the Mahalanobis distance between a specific point in the time series and its respective centroid; $x = (x_1, x_2, \dots, x_n)^T$ is a specific variable in the time-series data, where n is the number of variables; $\mu = (\mu_1, \mu_2, \dots, \mu_n)^T$ is a certain cluster centroid; and S is the covariance matrix relative to that cluster.

The next step implements the logic of finding the hidden patterns P in the time series T. After all the groups have been discovered, the algorithm compares which groups are similar to each other. This similarity is computed applying the standard deviation σy of the actual values of the T samples, the y coordinate of each centroid, and the clustering factor φ. Thus, if the modulus of the difference between the y coordinate of the centroids of two groups is less than or equal to the product of φ with σy, then these groups can be merged, which means that they will represent the same pattern in P. This task is carried out until every group has been analyzed.

In conclusion, the algorithm computes the probability of each pattern P being interesting in terms of being an anomaly in the time-series T. This is done by computing the anomaly index for each pattern P, which is calculated as the ratio between the total size of the time-series and the size of the pattern present in P. The higher the anomaly score is, the greater the probability of an anomalous behavior in T (Nascimento et al., 2015). Eq. (2) presents the Anomaly Score formula.

$$Anomaly\ Score_{P_i} = \frac{|T|}{|P_i|} \quad (2)$$

Where $Anomaly\ Score_{P_i}$ is the anomaly score of the pattern P_i, $|P_i|$ is the size of the pattern P_i, and $|T|$ is the size of the time series T.

2.2. Luminol Bitmap

Bitmap is an unsupervised learning algorithm in the Luminol library for anomaly detection or time-series correlation. The background of the Bitmap algorithm is based on the approaches of dimensionality reduction by symbolic aggregate approximation (SAX) and time-series bitmaps.

Initially, the algorithm normalizes each time series for a mean of zero and a standard deviation of one, since it is well understood that it is meaningless to compare time series with different offsets and amplitudes. Eq. (3) presents the Z-Score formula:

$$Z_{\mathrm{i}} = \frac{C_i - \bar{C}}{\sigma} \tag{3}$$

Where: Z_i is the Z-Score value of C_i, C_i is a specific sample in the time series data, $\bar{C}$ is the mean of the time-series, and σ is the standard deviation of the time series.

In the next step, the algorithm makes a feature extraction of the raw time series data by converting them into a Piecewise Aggregate Approximation (PAA). The PAA takes the real valued signal, divided into equal sized sections, and calculates the mean value of each section. Thus, by replacing each section with its mean, a reduced dimensionality piecewise constant approximation of the data is already obtained.

A time-series C of length n can be represented in PAA number segments w by a vector $\bar{C} = \bar{C}_1, \ldots, \bar{C}_w$. The ith element of $\bar{C}$ is calculated in Eq. (4):

$$\bar{C}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} C_j \tag{4}$$

After transforming a time-series dataset into PAA, it is applied for a further transformation to obtain a discrete representation. Once the normalized time series has a Gaussian distribution, a discretization technique can be achieved which will produce symbols with equiprobability. The conversion of the PAA number into SAX words is made by the Breakpoints β. Breakpoints are a threshold that will produce a equal-sized areas under the Gaussian curve, i.e., the Gaussian distribution is divided into a bins with equal probability at every bin. A set of break points corresponding to every symbol in SAX alphabet and a correspond to the alphabet size. SAX can produce strings on any alphabet size, but Luminol Bitmaps are defined for sequences with an alphabet size of four.

For instance, Figure 1 shows an example of Gaussian distribution with two areas of equiprobability ($a = 2$), *i.e.*, the probability of a PAA number being in the left side is the same as the right side. We can clearly see β as the threshold between the areas, thus for example, if the PAA value is less than or greater than β, the value will be converted into A and B SAX symbol, respectively .That is how SAX transformation works (Lin et al., 2003).

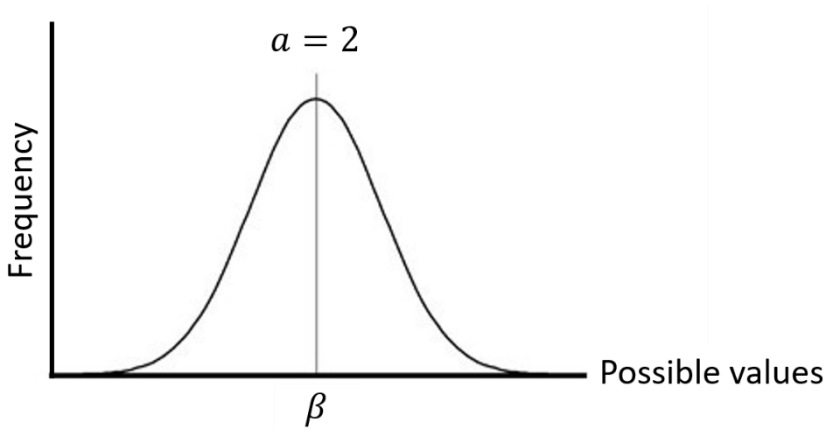

Figure 1. Gaussian curve with two areas of equiprobability.

The conversion into the SAX symbol is made by two slides of short windows. The lead window shows how far to look ahead for anomalous patterns and the lag window shows the size of the memory of the past to memorize.

After converting the original time series into SAX representation, the algorithm will count the frequencies of SAX subwords at the desired level of resolution to color the corresponding pixel of the bitmap grid in a principled way. The frequencies of SAX subwords are normalized [0,1] by the largest value and encode the pixels values to be in the RGB color space.

At this point, the distance between the two bitmap windows is measured and reported as an anomaly score at each time instance. The distance between them is defined as the summation of the square of the distance between each pair of pixels. More formally, for two nxn bitmaps BA and BB, the distance between them are represented in Eq. (5):

$$dist(BA, BB) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(BA_{ij} - BB_{ij}\right)^2 \quad (5)$$

The user must choose both lengths of the sliding window feature N, the number w of equal-sized sections in which to divide the time series N, and the detector score for the anomaly data computed. There is no choice to be made for alphabet size, because a simple alphabetical order of size 4 (*e.g.*: a, b, c, d) was fixedly defined in the Bitmap approach (Wei et al., 2005).

2.3 SAX-REPEAT

This approach consists of expanding the original SAX technique to handle multivariable data. The algorithm apply SAX to every dimension of the data separately and then combine the resulting string by assigning every possible combination of symbols in the resulting dimensionality D strings a unique identifier. The symbol combination leads to a string of length N but with an extended alphabet of length M^D. So, to remain the requirement that the final string must have an M-symbols alphabet, the algorithm use k-means in the multivariable SAX subwords to perform a clustering with M clusters and replace it with the centroid of its cluster (Mohammad & Nishida, 2014).

K-means is an unsupervised algorithm for pattern recognition based on the distance of a data point to its cluster centoride in a Euclidean space. It aims to partition n data points into k clusters in which each data point belongs to

the cluster with the nearest mean (also called cluster centers or cluster centroid) (Figueirêdo et al., 2020).

So, the SAX-REPEAT algorithm returned a set of groups or patterns from the multivariable dataset by applying SAX and k-means. Moreover, in order to extend this approach for an anomaly detection, we propose the Eq. (2) to compute the anomaly score for each cluster (the same approach used by the C-AMDATS algorithm). Therefore, SAX-REPEAT is now able to recognize interesting patterns in a multivariable dataset with their respective anomaly score, *i.e.*, their probability of being an anomalous pattern.

2.4 k-NN

The k-Nearest Neighbors (k-NN) algorithm is a popular method for classification and regression applications. The principle behind it is to predict the new data from the closest k training samples. The k is the number of neighbors that is predefined by the user as a constant and the distance is calculated in an Euclidean space. In general, the distance applied is an Euclidean function, which is the most common choice, but it can be calculated using other distance functions (Cover & Hart, 1967).

The Euclidean distance between the points $p = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$, in a n-dimensional Euclidean space is defined in Eq. (6):

$$d_e(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{6}$$

Where: $P = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$ are two points in Euclidean n-space. If we compare Eq. (6) to Eq. (1), a small difference can be noted, where Eq. (1) only includes the inverse of the covariance matrix in the calculation.

Ramaswamy et al. (2000) have extended k-NN to an unsupervised approach using the distance to the k^{th} nearest neighbor as an anomaly score. The anomaly score for each data point is the sum of the distance from its k nearest neighbors, called weight (Angiulli & Pizzuti, 2002). Each data point

is ranked by its anomaly score and the top n points of the rank are considered anomalous. The proportion of anomaly data is calculated based on contamination parameter times sample number.

The k-NN algorithm depends on three parameters, a metric used to compute the distance between two points (the Euclidean function applied in this study), a k value of the number of neighbors to consider, and contamination to define the threshold on the decision function (Cover & Hart, 1967; Gou et al., 2019).

2.5 Bootstrap

Bootstrap is a particularly useful algorithm to estimate any summary statistics, for example: variance, mean, standard error, and confidence intervals. The name bootstrap concerns the use of the original dataset to generate new datasets Z^*. The algorithm is statistical based and consists in generating samples of size B (called bootstrap samples) from an initial dataset of size n by randomly drawing with replacement B observations. There are two hypotheses that have to be verified to make this approach valid, which are: (i) the size N of the initial dataset should be large enough to capture most of the complexity of the underlying distribution (representativity), and (ii) the size N of the dataset should be large enough compared to the size B of the bootstrap samples so that samples are not too much correlated (independence).

The algorithm uses computational power to generate the bootstrap sample, which is a resampling of size n drawn to replace the original dataset $Z = (Z_1, Z_2, \dots, Z_n)$. The bootstrap sample is represented $Z^* = (Z_1^*, Z_2^*, \dots, Z_n^*)$. Each Z_i^* is one of the original Z values randomly chosen; the chosen probability for each Z value is equipollent, for instance: $Z_1^* = Z_7, Z_2^* = Z_5, Z_3^* = Z_9, Z_4^* = Z_7$, *etc*. Observe that the same original value can be chosen zero, one, or more times; in the example, Z_7 appeared twice, *i.e.*, the selection of Z value is not exclusive. The algorithm generates a larger number of bootstrap samples B of each size n using a random number device to perform the training. The number of bootstrap samples defines the variance of the estimate, *i.e.*, the higher the number is, the better the

variance, but, on the other hand, the computational cost increases withincreasing B value (Efron, 1992; Efron et al., 2015).

In this sense, we used Bootstrap to calculate a confidence interval of the time series to get the potential anomaly data. The algorithm initially requests the statistics stored during the training and selects values in the chosen percentile for the confidence interval. The chosen percentile is denoted as δ (Alpha or Significance Level). Eq. (7) and Eq. (8) define the formula to estimate the distribution of δ* for each Bootstrap sample.

$$\delta^* = \bar{x}^* - \bar{x} \tag{7}$$

Where: $\bar{x}^*$ is the mean of an empirical bootstrap sample and $\bar{x}$ is the mean of the original data.

After computing the distribution of the Bootstrap Sample, the confidence interval of the bootstrap distribution can be found. For instance, δ.05 and δ.95 are the 0.05 and 0.95 critical values of δ, which gives a confidence interval of 90%; that confidence interval is defined in Eq. (8)

$$\mathcal{C}\text{onfidence interval} = [\bar{x} - \delta^*_{.05}, \bar{x} - \delta^*_{.95}] \tag{8}$$

Where: $\bar{x}$ is the mean of the original data, $\delta^*_{.05}$ is significance level at the 5th percentile, and $\delta^*_{.95}$ is the significance level at the 95th percentile.

Thus, in order to obtain a very accurate estimate of $\delta^*_{.05}$ and $\delta^*_{.95}$, it is important to generate a large number of bootstrap samples.

2.6 Robust Random Cut Forest (RRCF)

RRCF is an ensemble algorithm for detecting anomaly data points. The algorithm takes a set of random data points (Random), cuts them to the same number of points, and creates trees (Cut). Then it looks at all trees together (Forest) to determine whether a particular data point is an anomaly.

The user can choose the number of trees and the size of each tree. The size corresponds to the number of data points that each tree has which is randomly sampled from the original dataset. Once the forest is developed,

the algorithm injects the new data point p inside the trees and start to cut to compute the average depth of the point across the collection of trees. Point p is labeled as an anomaly if the score overtakes the threshold, which corresponds to the average depth across the trees (Guha et al., 2016).

3. Multivariate Time-Series Data

The comparative performance evaluation was done using multivariate time series data from the oil and gas industry. The data comes from a rotating machine of the P-50 FPSO. The platform is located in the Albacoara Leste field (latitude 22º05'04 " S and longitude 039º49'45 " W) in the Campos Basin, 120 km from the coast of the state of Rio de Janeiro, Brazil. The P-50 is operated by Petrobras and Repsol Sinopec Brazil.

The platform has many critical machines that are constantly monitored for maintenance purposes. In this sense, multivariate time series data was collected from a turbogenerator of the power generation system. The datasets refer to the turbogenerator health status. The machine consists of a multi-stage gearbox, a generator, and a turbine with an axis rotation speed of 6,000 rpm.

The measurement data come from 28 different sensors installed in the gearbox, generator, and electrical parts of the turbogenerator, as given in Table 1. The variables collected are diverse, such as vibration, temperature, pressure, rotation, current, and reactive power. The high number of sensors installed on the machine demonstrates how important it is to the system.

Our goal was to evaluate the performance of unsupervised algorithms on multivariate time series data, as usually several parameters are collected from critical machines for a better representation of phenomena. However, the greater the amount of data, the more difficult human analysis is, especially in real time. However, it also presents a challenge for machine learning algorithms because of the increased volume of data and attributes to be correlated.

Table 1. List of variables collected, including sensor number (N), description, and measuring units.

N	Description	Unit
1	Lube oil supply pressure	kPa
2	Input shaft speed	RPM
3	Generator reactive power	VAr
4	Generator radial bearing temperature 1 DE	°C
5	Generator radial bearing temperature 2 DE	°C
6	Generator radial bearing NDE temperature 1	°C
7	Generator radial bearing NDE temperature 2	°C
8	Gearbox radial bearing temperature 1 DE	°C
9	Gearbox radial bearing temperature 2 DE	°C
10	Gearbox high shaft bearing DE X temperature	°C
11	Gearbox high shaft bearing DE Y temperature	°C
12	Gearbox high shaft bearing NDE X temperature	°C
13	Gearbox high shaft bearing NDE Y temperature	°C
14	Gearbox shaft bearing NDE X temperature	°C
15	Gearbox shaft Bearing NDE Y Temperature	°C
16	Gearbox thrust bearing temperature 1	°C
17	Gearbox thrust bearing temperature 1	°C
18	Gearbox thrust bearing temperature 2	°C
19	Gearbox thrust bearing temperature 2	°C
20	Lube oil supply temperature	°C
21	Gearbox frame accelerometer 1	g
22	Gearbox frame accelerometer 2	g
23	Generator DE frame accelerometer	g
24	Generator NDE frame accelerometer	g
25	Gearbox X DE radial vibration of high shaft	μm P-P
26	Gearbox Y DE radial vibration of high shaft	μm P-P
27	Gearbox X DE radial vibration of low shaft	μm P-P
28	Gearbox Y DE radial vibration of low shaft	μm P-P

Four cases of multivariable time series data was acquired from the turbogenerator. All of them were collected at a rate of 1 to 30 samples per second. In summary, four datasets of real cases were structured for this study:

Dataset I. Started on 07/17/2018 at 12:20:00 PM and ended on 07/19/2018 at 02:21:00 PM (2 days, 2 hours, and 1 minute) and comprising 82,174 data points;

Dataset II. Started on 06/25/2018 at 03:26:00 PM and ended on 27/06/2018 at 05:22:00 PM (2 days, 1 hour, and 56 minutes) and comprising 79,411 data points;

Dataset III. Started on 06/29/2018 at 07:50:00 AM and ended on 07/01/2018 at 04:05:00 PM (2 days, 8 hours, and 15 minutes) and comprising 91,116 data points;

Dataset IV. Started on 08/01/2018 at 04:00:00 AM and ended on 08/03/2018 at 07:39:00 AM (2 days, 3 hours, and 39 minutes) and comprising 83,297 data points.

As our focus is to evaluate the performance of unsupervised machine learning algorithms to assist experts on labeling the normal/anormal data, all datasets contain normal data and abnormal events that were indicated in operational reports. However, we have noted that the operator recorded the time afterthey observed the abnormal event and performed immediate follow-up actions, and thus some inaccurate records may be present. In addition, usually the control room clock does not display seconds (*i.e.*, one-minute resolution). Depending on the complexity of the abnormal event and the operator's expertise, the time difference between the exact abnormal event and the recorded time could be several seconds to several minutes.

As a data pre-processing step, a minute-to-minute resample was made to the interval of data points. We assume that all data points within the same one-minute interval would be equivalent to the average of the value. Eq. (9) displays the preprocessing formula.

$$\{t_0 \leq t(x_i) < t_0 + 1min\} \rightarrow \bar{x} = \sum_{i=1}^{n} \left(\frac{x_i}{n}\right) \tag{9}$$

Where t_0 is the initial time of 1 minute interval, $t(x_i)$ is the time of the data point i, n is the amount number of data points between t_0 and $t_0 + 1min$, and $\bar{x}$ is the average of the value of data points between t_0 and $t_0 + 1min$.

On July 18, 2018, the turbogenerator tripped because of the high vibration signal measured by sensor 28. Figure 2 shows the multi-sensor signals of the machine collected during this anomalous behavior (Dataset I). The vertical red lines indicate the beginning of the malfunction until the return

of the operation, which lasted at total of 2 hours and 1 minute. This event was recorded in the operational reports as two types of failures that occurred sequentially, which were: trip by vibration and trip by flame loss. Nevertheless, we can note that there are different sensors that show significantly distinct patterns and trends. Moreover, a comparison between this operation record and other records – as shown in Figure 3, Figure 4, and Figure 5 –from the same sensors show that they may behave quite differently under different operating circumstances. This complex data structure motivates us to explore the performance of unsupervised machine learning algorithms

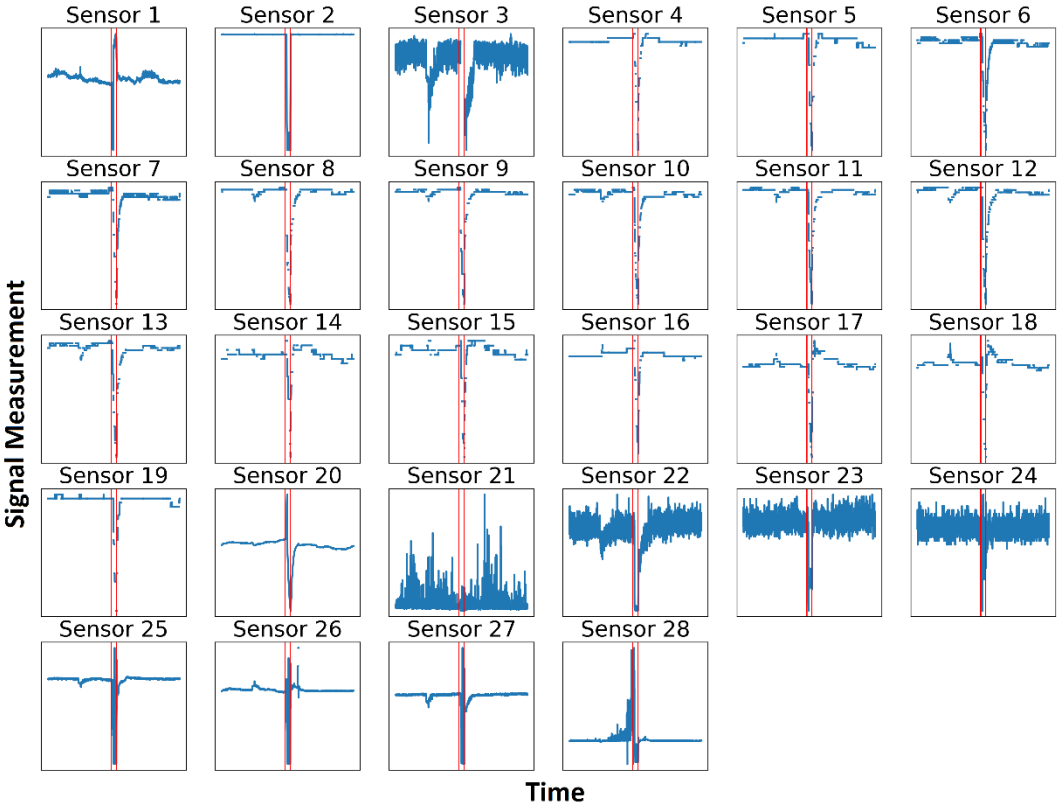

Figure 2. Multi-sensor signals of Dataset I with a recorded anomaly marked between the vertical red lines. Each sensor description can be found in Table 1 according to the sensor number .

The failure in Figure 2 is a typical case of machine degradation that occurred gradually, *i.e.*, in this case, predictive models can be used to detect the imminence of the failure to anticipate it. Thus, unsupervised algorithms can recognize interesting\anomalous patterns to assist specialists in data labeling for further training of DP models for maintenance purposes. All six

algorithms were applied in the same data to verify their ability to detect the same failure signal.

In Dataset II, on June 26, 2018, the turbogenerator had another trip because of the low lube supply pressure measured by Sensor 1. Figure 3 shows the multi-sensor signals collected from the machine operation with this anomaly data. The operational reports described three different types of failure that occurred sequentially: trip during alignment of the exchanger, trip by gearbox vibration, and trip by flame loss.

The beginning of the anomaly function until the return of operation of the machine is indicated between the vertical red lines in Figure 3, which lasted 2 hours.

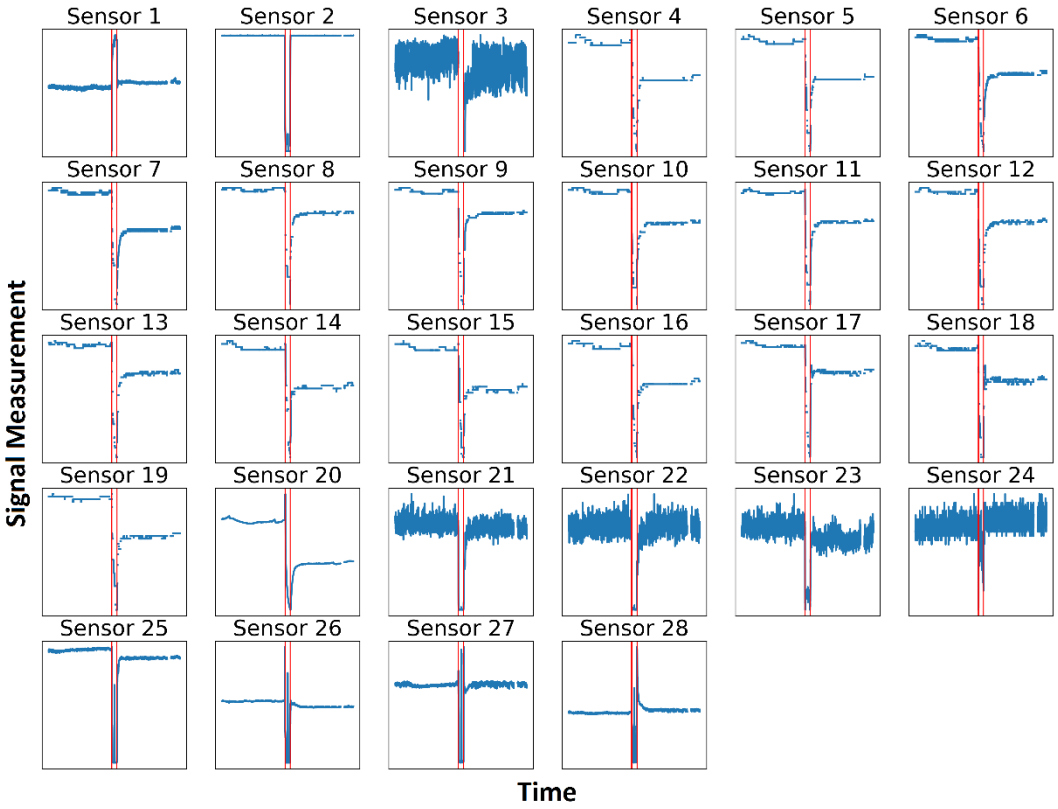

Figure 3. Multi-sensor signals of Dataset II with a recorded anomaly marked between the vertical red lines. Each sensor description can be found in Table 1 according to the sensor number .

In Dataset III, on June 30, 2018, the machine had another unscheduled stop. Figure 4 shows the multi-sensor signals collected before, during, and after

the machine operation at this specific unexpected stop. The operational reports described two different types of failure that occurred sequentially: trip by pressure differential in the pump filter and trip by vibration. In Figure 4, the red vertical lines represent the beginning of the anomalous behavior recorded in the operational reports until the return of the machine, which lasted 8 hours and 15 minutes.

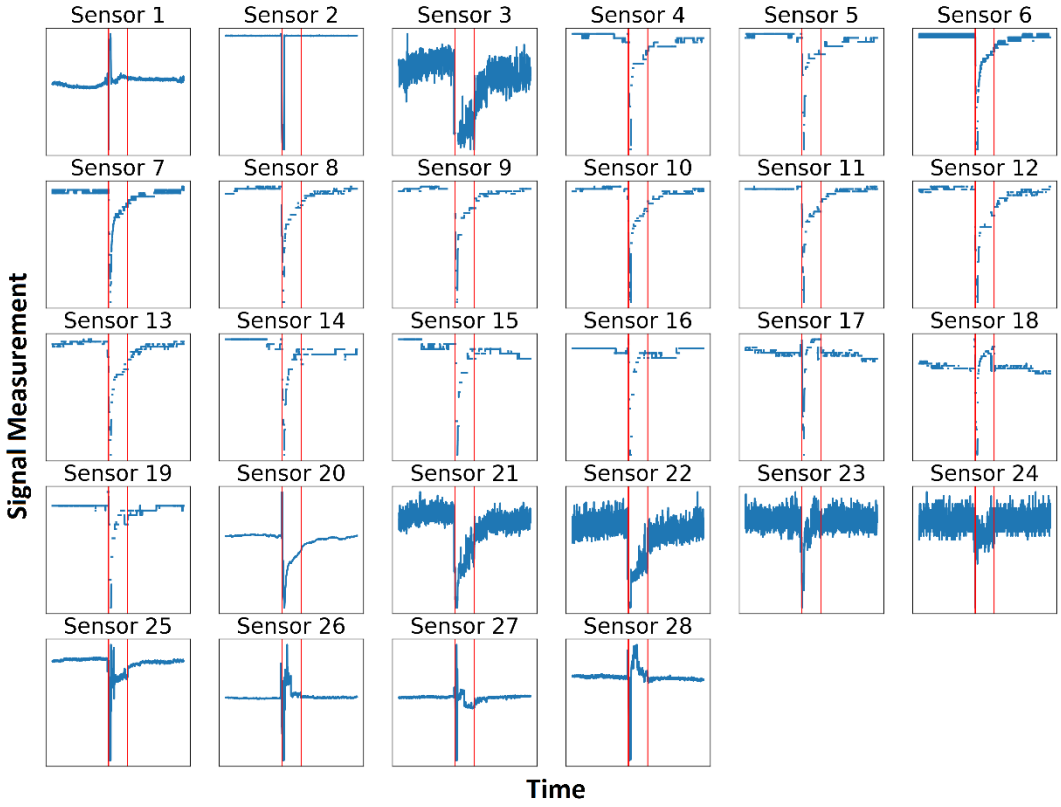

Figure 4. Multi-sensor signals of Dataset III with a recorded anomaly marked between the vertical red lines. Each sensor description can be found in Table 1 according to the sensor number.

Lastly, in Dataset IV the machine shuts down unexpectedly because of a trip by Vacuum Solenoid Valve (VSV) failure on August 2, 2018. Figure 5 shows the multi-sensor signals that contain the anomaly data that caused the machine shutdown, which lasted 3 hours and 35 minutes.

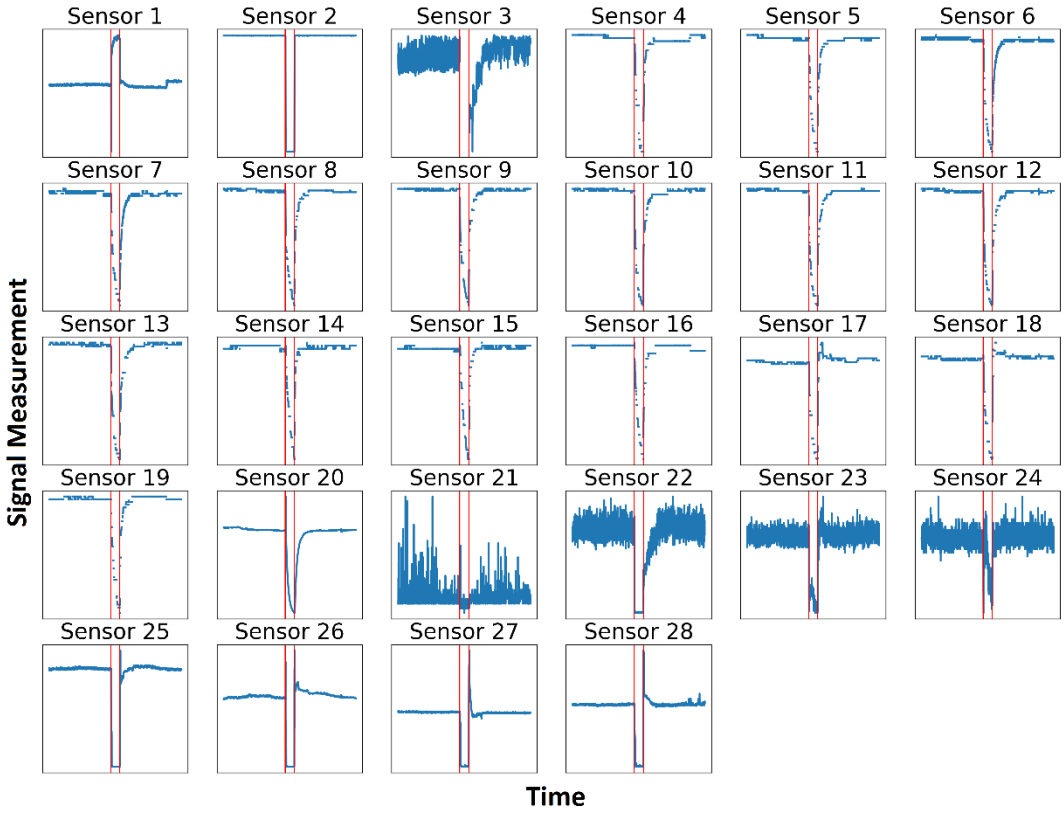

Figure 5. Multi-sensor signals of Dataset IV with a recorded anomaly marked between the vertical red lines. Each sensor description can be found in Table 1 according to the sensor number .

Table 2 shows the statistics of each dataset and their respective failure signals. We can note that the datasets are highly unbalanced with 3.86 to 14.65% of anomaly data. Unbalanced data is quite common in real machine measurement data, as the machine is expected to operate most of the time in good condition. Furthermore, the data has few Not a Number[3] (NaN) records; Dataset II is the highest one, with 5.81%. However, the amount of NaN in the failure zone is quite significant: Dataset III has the highest proportion with 15.83%, and Dataset II has the lowest one with 6.10%.

[3] Not a Number means missing data points.

Table 2. Summary of the dataset statistics.

Dataset Id	Data	Data Points (#)	NaN (#)	Variables (#)	Rows (#)
I	Total	82,174	1,882	28	3,002
	Failure	3,175	241	28	122
II	Total	79,411	4,617	28	3,002
	Failure	3,107	281	28	121
III	Total	91,116	3,412	28	3,376
	Failure	13,348	540	28	496
IV	Total	83,297	3,391	28	3,096
	Failure	5,615	433	28	216

NaN: Not-a-Number.

A linear interpolation was performed to fill the NaN because most of the algorithms are sensitive to missing data points. The reason for existing NaN in the dataset is because the 28 sensors are not synchronized to collect data at the same sample rate.

4. Experimental Setup

Experimental setup in the field of machine learning is an optimization of parameters or adjustment of the algorithm. In summary, it concerns in choosing a set of ideal parameters to optimize the performance of the algorithm.

Each algorithm has hyperparameters that need to be adjusted to each case study. Thus, expertise in the algorithm logic and some attempts at success and error are necessary to find good values among them.

Figueirêdo et al. (2020) have reported difficulty in fine tuning the algorithms manually . Thus, we have implemented a loop in the execution of algorithms that varies the parameters in each iteration, which generates different combinations of parameters. The following combinations are defined in Table 3.

Table 3. Parameter tuning.

Algorithm	Parameter combination
C-AMDATS	• Initial Cluster Size (minutes): [15, 20, 25, 30] • Clustering Factor: [0.5, 1, 1.4, 1.6, 1.8, 2.0]
Luminol Bitmap	• Detector Score: [0.1,0.5,1.0] • Lead / Lag Size: [1, 10, 100, 1000] • Precision: [1, 10, 100] • Chunk Size: [1, 10, 100]
SAX-REPEAT	• Window Size: [1, 10, 100, 500, 1000] • PAA Size: [1, 2, 3, 4] • Alphabet Size: [1, 2, 3, 4]
k-NN	• K: [3, 5, 10, 50, 100, 150, 300, 500] • Contamination: [0.1, 0.2, 0.3]
Bootstrap	• Confidence Interval: [0.90, 0.95, 0.99] • Iteractions: [100, 200, 400, 500, 600, 1000, 2000, 5000]
RRCF	• Number of Trees: [55, 110, 220] • Tree Size: [128, 256] • Shingle Size: [4]

The RRCF had the lowest number of iterations because of the small number of parameters and the little variation in results during the loop. The metric score to define the best model setting was Area Under the Precision-Recall Curve (AUC-PRC). Section 5 describes AUC-PRC in more detail. Table 4 summarizes the best parameter settings of the presented algorithms in the four case studies.

We cannot say that our proposed parameterization is the best solution to the problem because it would be necessary to test all existing possibilities of hyperparameters, which would make this study exhaustive and with a high level of computational cost.

As the different cases analyzed are data from the same machine and the same multiple sensors, but for different moments of time, the parameters values in Table 4 did not vary much between them. This suggests that for the same applications the parameters may vary little, which allows the skipping of the step of finding the best parameters and the reduction of the computational power once the application is already known.

Table 4. Parameter settings of the unsupervised algorithms.

Algorithm	Parameter	Cases			
		#1	#2	#3	#4
C-AMDATS	ICS (min)	30	30	30	30
	Cluster Factor	2.0	2.0	1.8	2.0
Luminol Bitmap	Detector Score	0.1	0.1	0.1	0.1
	Precision	10	10	10	10
	Window Size(1)	100	100	100	100
	Chunk Size	1	21	10	1
SAX-REPEAT	Window Size	1	1	1	1
	PAA Size	1	1	1	1
	Alphabet Size	4	4	5	4
k-NN(2)	k	300	300	500	500
	Contamination	0.1	0.1	0.2	0.1
Bootstrap	CI	0.99	0.99	0.99	0.99
	Iteractions	400	200	200	400
RRCF(3)	Number of Trees	110	110	110	220
	Tree Size	256	256	128	256

ICS: Initial Cluster Size, PAA: Piecewise Aggregate Approximations, k: Number of Neighbors, CI: Confidence Interval, (1) same value applied for Lag window and Lead window, (2) Euclidean distance applied for every case, (3) Shingle Size = 4 for every cases.

The algorithms were executed on the supercomputer named AIRIS (Artificial Intelligence Repsol Sinopec Brazil Integrated System) at the Supercomputing Center for Industrial Innovation (CS2I) at SENAI CIMATEC. The AIRIS processor model is the Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz and has 376 GB RAM memory. All algorithms have been implemented with Python 3.6.

5. Metrics Employed for Evaluation Performance

In this paper we evaluate the performance of the unsupervised algorithms for identifying the same anomaly patterns by seven metrics. It is prudent to measure the performance with a set of metrics to avoid any bias deviation. The evaluation compares the real data points (classified by experts) against the predicted data points (predicted by the unsupervised algorithms). The seven metric are:

- Accuracy (ACC): considering all normal and faulty samples, ACC can be computed as:

$$ACC = \frac{TP + TN}{n_{total}} \tag{10}$$

Where: TP and TN are respectively the number of true positives and true negatives in samples, and n_{total} is the overall number of samples.

- Precision (PR): indicates the true positive value compared to the false negative; PR can be calculate as:

$$PR = \frac{TP}{TP + FP} \tag{11}$$

Where: FP is the number of false positives.

- Recall (REC): or True Positive Ratio (TPR) indicates the proportion of anomalies that are correctly detected out of all anomalies. Normally, it is a high prioritized metric since the failure of anomaly detection (false negatives) leads to much more deleterious results than false alarms (false positives) in industrial applications. REC can be calculated as:

$$REC = \frac{TP}{TP + FN} \tag{12}$$

where FN is the number of false negatives.

- Specificity (SP): demonstrates the capacity of the model to predict the true negative over false positives, and it can be computed as:

$$SP = \frac{TN}{TN + FP} \tag{13}$$

- F1-Score (F1): is a harmonic average between REC and PR, and it can be calculated as:

$$F1 = \frac{2 \times (PR \times REC)}{PR + REC} \tag{14}$$

- Area Under the Receiver Operating Characteristics Curve (AUC-

ROC): provides a relative tradeoff between the False Positive Rate (FPR) and the TPR. FPR is the equivalent of $1 - SP$;

- AUC-PRC: provides a relative tradeoff between the PR and the REC. It is an important metric for assessing unbalanced datasets, being a great advantage over the others metrics, because the vast majority of real data has a higher volume of normal data in relation to abnormal data.

All metrics above take a value between 0 and 1. A perfectly inaccurate anomaly detection model has a value of 0; a model that makes random guesses has a value of 0.5; and a perfectly accurate model has a value of 1, *i.e.*, the higher the metric value, the better is the anomaly detection performance.

The performance evaluation would not be properly fair as the Luminol, k-NN, Bootstrap, and RCCF algorithms made an univariable analysis, unlike C-AMDATS and SAX-REPEAT that have a multivariate approach. Thus, to obtain a more appropriate analysis, the metrics were calculated for all proposed variables and then we extracted an average evaluation, except for C-AMDATS and SAX-REPEAT.

6. Results and Discussion

In this section, we present the results for all six unsupervised machine learning algorithms described in Section 2. The algorithms were applied in four different cases studies with multivariate time series data, described in Section 3. Therefore, the goal is to investigate how each unsupervised algorithm can work to detect anomalies.

The C-AMDATS and SAX-REPEAT algorithms are designed to find interesting/anomalous patterns in a multivariate time series data, and each pattern of interest is assigned an anomaly index. Thus, patterns with the highest indexes are selected as anomalous patterns, unlike the other algorithms that are designed to find anomaly and normal data.

The evaluation performance is summarized in Table 5. We can see that the C-AMDATS was the one that stood out among the other algorithms. The

algorithm achieved the best performance in all seven metrics. SAX-REPEAT had the second best performance. A rank in decreasing order of performance the algorithms would be: (i) C-AMDATS, (ii) SAX-REPEAT, (iii) Bootstrap, (iv) k-NN, (v) Luminol Bitmap, and (vi) RRCF. Both C-AMDATS and SAX-REPEAT algorithms with a multivariate analysis were intrinsically superior. However, more case studies must be carried out to affirm the superiority of the algorithms studied here.

Table 5. Unsupervised machine learning performance in four real cases (the best performance is highlighted in bold).

Algorithm	Metric	Cases (%)				Mean (%)	SD (%)
		#1	#2	#3	#4		
C-AMDATS	ACC	**99**	**100**	**99**	**99**	**99**	**1**
	PR	**80**	**100**	**94**	**90**	**91**	**8**
	REC	**98**	**99**	**97**	**100**	**99**	**1**
	F1	**88**	**100**	**95**	**95**	**94**	**5**
	SP	**98**	**99**	**97**	**100**	**99**	**1**
	AUC-ROC	**99**	**100**	**98**	**100**	**99**	**1**
	AUC-PRC	**79**	**99**	**92**	**90**	**90**	**8**
Luminol Bitmap	ACC	61	68	70	75	69	6
	PR	14	19	43	33	27	13
	REC	100	100	91	99	98	4
	F1	24	31	55	48	39	14
	SP	100	100	91	99	98	4
	AUC-ROC	80	84	79	86	82	3
	AUC-PRC	14	19	39	33	26	12
SAX-REPEAT	ACC	96	100	98	98	98	2
	PR	47	92	95	82	79	22
	REC	68	99	90	96	88	14
	F1	56	95	92	89	83	18
	SP	68	99	90	96	88	14
	AUC-ROC	82	99	95	97	93	8
	AUC-PRC	34	91	87	79	73	26
k-NN	ACC	93	95	92	96	94	2
	PR	34	46	72	65	54	17
	REC	65	91	85	90	83	12
	F1	44	60	77	75	64	15
	SP	65	91	85	90	83	12
	AUC-ROC	79	93	89	93	89	6
	AUC-PRC	25	43	65	60	48	18

Algorithm	Metric	Cases (%)				Mean (%)	SD (%)
		#1	#2	#3	#4		
Bootstrap	ACC	95	94	94	97	95	2
	PR	51	55	89	78	68	18
	REC	62	87	66	88	76	14
	SP	54	65	73	82	69	12
	F1	62	87	66	88	76	14
	AUC-ROC	79	91	82	93	86	7
	AUC-PRC	35	52	65	71	56	16
RRCF	ACC	91	91	82	89	88	4
	PR	17	16	27	21	20	5
	REC	27	27	13	20	22	7
	SP	20	20	17	20	19	1
	F1	27	27	13	20	22	7
	AUC-ROC	61	60	53	57	58	3
	AUC-PRC	8	8	16	10	11	4

SD: Standard Deviation

With these outcomes, C-AMDATS demonstrated the highest capacity to detect the anomalous and normal patterns present in the multivariate time series. The algorithm revealed the highest and lowest value in all metrics for the mean and standard deviation, respectively. Therefore, with these automatically detected patterns, experts would be able to label a large mass of data in a few moments and with a high accuracy, as well as reduce the many hours of manual and tedious efforts of data annotation tasks.

The SAX-REPEAT and the Bootstrap algorithms performed well for some case studies. The SAX-REPEAT detected the anomalous data in case 2, case 3, and case 4; however, it was unsuccessful in case 1. Bootstrap detected only anomalous data from case 4. The remaining algorithms scored low for all cases, as shown in Table 5.

In addition to evaluating the performance of each algorithm, it is also essential to analyze processing time, which is directly proportional to computational cost. Thus, Table 6 shows this comparison:

Table 6. Processing time of each algorithm (the fastest algorithm is highlighted in bold).

Algorithm	Time of each case study (s)				Mean (s)	SD (s)
	#1	#2	#3	#4		
C-AMDATS	429.2	73.9	703.5	8,439.0	2,411.4	4,026.7
Bitmap	10.3	10.2	20.4	9.8	12.7	5.1
SAX-REPEAT	18.3	201.2	184.8	221.7	156.5	93.3
k-NN	11.5	8.6	14.8	14.3	12.3	2.9
Bootstrap	**3.0**	**3.5**	**3.3**	**3.7**	**3.4**	**0.3**
RRCF	1,625.7	1,519.1	2,449.7	5,849.1	2,860.9	2,035.1

SD: standard deviation

Bootstrap was the fastest algorithm with 3.4 and 0.3 seconds of mean and standard deviation, respectively. A fast algorithm enables real-time analysis. However, the processing time for RRCF and C-AMDATS were very high with an average of 2,860 seconds and 2,411 seconds, respectively. as C-AMDATS showed the best performance (Table 5), it would be interesting to optimize its structure to speed up its processing time, such as parallelization of the calculation of the inverse of the covariance matrix and use of Multiple Graphics Processing Units (GPU).

7. Conclusion

In this work, we have demonstrated the effectiveness of a multivariate analysis using six different unsupervised machine learning algorithms in a real system from an FPSO. Our goal was to compare the performance of the algorithms to automatically identify anomalous patterns in multivariate time series data. The experimental results showed that unsupervised algorithms, such as C-AMDATS, have high ability to recognize and isolate abnormal events of rotating machinery, thus showing their great capacity to automatically label raw data with unsupervised learning.

Therefore, instead of experts spending several hours of hard and tedious work labeling a large amount of data− as shown in Figure 2, Figure 3, Figure 4, and Figure 5 −, they would just have to label a few patterns recognized by unsupervised learning with demonstrated high performance, such as C-AMDATS which had 99% of ACC. However, C-AMDATS

revealed the need to speed up its execution time.

In conclusion, unsupervised learning can leverage the development of DP models for industrial applications, which normally only have unlabeled data. This can be especially applied in the oil and gas industry as it has a large number of multivariate datasets.

Future works include the extension of this analysis to more real cases in the oil and gas industry aiming to develop a broader assesment, as well as an extensive study and investigation of approaches of semi-supervised learning to train DP algorithms to predict and classify unknown data in different mechanical failure modes of rotating machinery for PdM tasks.

7. Acknowledgment

The authors would like to thank the partner companies Repsol Sinopec Brasil and Petrobras for providing the data and all the support given. The authors also thank the CS2I for providing the computational resources needed to develop this work and access to the AIRIS supercomputer, and the Reference Center for Artificial Intelligence, both at SENAI CIMATEC. This work was partially supported by the Brazilian National Agency of Petroleum, Natural Gas and Biofuels (ANP). We would like to express our deeply felt gratefulness to Professor Dr. Davidson Moreira from SENAI CIMATEC and Dr. Leandro Andrade from SENAI CIMATEC for the comprehensive and detailed peer-editing of this document.

References

Abid, A., Khan, M. T., & Khan, M. S. (2020). Multidomain Features-Based GA Optimized Artificial Immune System for Bearing Fault Detection. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(1), 348–359. https://doi.org/10.1109/TSMC.2017.2746762

Angiulli, F., & Pizzuti, C. (2002). Fast Outlier Detection in High Dimensional Spaces. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 15–27). https://doi.org/10.1007/3-540-45681-3_2

Ben Ali, J., Saidi, L., Harrath, S., Bechhoefer, E., & Benbouzid, M. (2018). Online automatic diagnosis of wind turbine bearings progressive degradations under real experimental conditions based on unsupervised machine learning. *Applied Acoustics*, *132*, 167–181. https://doi.org/10.1016/j.apacoust.2017.11.021

Caggiano, A., Zhang, J., Alfieri, V., Caiazzo, F., Gao, R., & Teti, R. (2019). Machine learning-based image processing for on-line defect recognition in additive manufacturing. *CIRP*

Annals, *68*(1), 451–454. https://doi.org/10.1016/j.cirp.2019.03.021

Chollet, F. (2018). Deep Learning with Phyton. In *Manning*.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Durbhaka, G. K., & Barani, S. (2016). Fault behaviour pattern analysis and recognition. *2016 International Conference on Information Science (ICIS)*, 191–193. https://doi.org/10.1109/INFOSCI.2016.7845325

Efron, B. (1992). Bootstrap Methods: Another Look at the Jackknife. In *Breakthroughs in Statistics* (Vol. 7, Issue 1, pp. 569–593). https://doi.org/10.1007/978-1-4612-4380-9_41

Efron, B., Rogosa, D., & Tibshirani, R. (2015). Resampling Methods of Estimation. In J. D. Wright (Ed.), *International Encyclopedia of the Social & Behavioral Sciences* (Second Edi, pp. 492–495). Elsevier. https://doi.org/10.1016/B978-0-08-097086-8.42165-3

Figueirêdo, I. S., Guarieiro, L. L. N., & Nascimento, E. G. S. (2020). Multivariate Real Time Series Data Using Six Unsupervised Machine Learning Algorithms. In *Anomaly Detection - Recent Advances, Issues and Challenges [Working Title]* (Issue tourism, p. 13). IntechOpen. https://doi.org/10.5772/intechopen.94944

Goldmeer, J., Sanz, A., Adhikari, M., & Hundal, A. (2018). *Gas to Power: The art of the Possible The Fuel Flexibility of GE Power Aeroderivative Gas Turbines*. https://www.ge.com/content/dam/gepower-pw/global/en_US/documents/GasPower/gasturbines/GEA34108_Aero_Fuel_Flexibility_Whitepaper_Final.pdf

Gombé, B. O., Mérou, G. G., Breschi, K., Guyennet, H., Friedt, J. M., Felea, V., & Medjaher, K. (2019). A SAW wireless sensor network platform for industrial predictive maintenance. *Journal of Intelligent Manufacturing*, *30*(4), 1617–1628. https://doi.org/10.1007/s10845-017-1344-0

Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., & Yang, H. (2019). A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, *115*, 356–372. https://doi.org/10.1016/j.eswa.2018.08.021

Guha, S., Mishra, N., Roy, G., & Schrijvers, O. (2016). Robust random cut forest based anomaly detection on streams. *33rd International Conference on Machine Learning, ICML 2016*, *48*, 3987–3999. http://proceedings.mlr.press/v48/guha16.pdf

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012

Jati, A., & Georgiou, P. (2019). Neural Predictive Coding Using Convolutional Neural Networks Toward Unsupervised Learning of Speaker Characteristics. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *27*(10), 1577–1589. https://doi.org/10.1109/TASLP.2019.2921890

Kakarla, R., Krishnan, S., & Alla, S. (2021). Unsupervised Learning and Recommendation Algorithms. In *Applied Data Science Using PySpark* (pp. 251–298). Apress. https://doi.org/10.1007/978-1-4842-6500-0_7

Li, Y., Du, X., Wan, F., Wang, X., & Yu, H. (2020). Rotating machinery fault diagnosis based on convolutional neural network and infrared thermal imaging. *Chinese Journal of Aeronautics*, *33*(2), 427–438. https://doi.org/10.1016/j.cja.2019.08.014

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery - DMKD '03*, 2–11. https://doi.org/10.1145/882082.882086

Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. In *Mechanical Systems and Signal Processing* (Vol. 108). https://doi.org/10.1016/j.ymssp.2018.02.016

Love, B. C. (2002). Comparing supervised and unsupervised category learning. *Psychonomic Bulletin & Review*, *9*(4), 829–835. https://doi.org/10.3758/BF03196342

Mohammad, Y., & Nishida, T. (2014). Robust learning from demonstrations using multidimensional SAX. *14th International Conference on Control, Automation and Systems*

(ICCAS), 64–71. https://doi.org/10.1109/ICCAS.2014.6987960

Nascimento, E. G. S., Tavares, O., & De Souza, A. (2015). A Cluster-based Algorithm for Anomaly Detection in Time Series Using Mahalanobis Distance. *ICAI'2015 - International Conference on Artificial Intelligence*, 622–628. https://www.researchgate.net/publication/282330724_A_Cluster-based_Algorithm_for_Anomaly_Detection_in_Time_Series_Using_Mahalanobis_Distance

Nguyen, V. H., Cheng, J. S., Yu, Y., & Thai, V. T. (2019). An architecture of deep learning network based on ensemble empirical mode decomposition in precise identification of bearing vibration signal. *Journal of Mechanical Science and Technology*, *33*(1), 41–50. https://doi.org/10.1007/s12206-018-1205-6

Parrella, I., Bardi, F., Salerno, G., Gronchi, D., Cannavò, M., & Sparacino, E. (2019, November 11). Using Analytics to Assess Health Status of DLE Combustion Gas Turbines. *Abu Dhabi International Petroleum Exhibition & Conference*. https://doi.org/10.2118/197679-MS

Perera, L. P., Machado, M. M., Valland, A., & Manguinho, D. A. P. (2019). Failure intensity of offshore power plants under varying maintenance policies. *Engineering Failure Analysis*, *97*(February 2018), 434–453. https://doi.org/10.1016/j.engfailanal.2019.01.011

PETROBRAS. (2019). *Petrobras informa sobre ocorrência na P-50*. PETROBRAS; Wiley Online Library. https://www.agenciapetrobras.com.br/Materia/ExibirMateria?p_materia=981253&p_editoria=8

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. *SIGMOD Record (ACM Special Interest Group on Management of Data)*. https://doi.org/10.1145/335191.335437

Soualhi, A., Razik, H., Clerc, G., & Doan, D. D. (2014). Prognosis of bearing failures using hidden markov models and the adaptive neuro-fuzzy inference system. *IEEE Transactions on Industrial Electronics*, *61*(6), 2864–2874. https://doi.org/10.1109/TIE.2013.2274415

Souza, R. M., Nascimento, E. G. S., Miranda, U. A., Silva, W. J. D., & Lepikson, H. A. (2021). Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Computers & Industrial Engineering*, *153*(N/A), 107060. https://doi.org/10.1016/j.cie.2020.107060

Torabi Jahromi, A., Er, M. J., Li, X., & Lim, B. S. (2016). Sequential fuzzy clustering based dynamic fuzzy neural network for fault diagnosis and prognosis. *Neurocomputing*, *196*(N/A), 31–41. https://doi.org/10.1016/j.neucom.2016.02.036

Vargas, R. E. V., Munaro, C. J., Ciarelli, P. M., Medeiros, A. G., Amaral, B. G. do, Barrionuevo, D. C., Araújo, J. C. D. de, Ribeiro, J. L., & Magalhães, L. P. (2019). A realistic and public dataset with rare undesirable real events in oil wells. *Journal of Petroleum Science and Engineering*, *181*, 106223. https://doi.org/10.1016/j.petrol.2019.106223

Wachowiak, M. P., Moggridge, J. J., & Wachowiak-Smolikova, R. (2019). Clustering Continuous Wavelet Transform Characteristics of Heart Rate Variability through Unsupervised Learning. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 4584–4587. https://doi.org/10.1109/EMBC.2019.8857515

Wang, X.-B., Zhang, X., Li, Z., & Wu, J. (2019). Ensemble extreme learning machines for compound-fault diagnosis of rotating machinery. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2019.105012

Wei, L., Kumar, N., Lolla, V. N., Keogh, E. J., Lonardi, S., & Ratanamahatana, C. (Ann). (2005). Assumption-Free Anomaly Detection in Time Series. *17th International Conference on Scientific and Statistical Database Management, SSDBM*, *5*, 237–242. https://pdfs.semanticscholar.org/909b/8226968d41f76cc14d0eef2d365572e7a37b.pdf?_ga=2.68813208.804195361.1594613685-826585445.1591805583

Yiakopoulos, C. T., Gryllias, K. C., & Antoniadis, I. A. (2011). Rolling element bearing fault detection in industrial environments based on a K-means clustering approach. *Expert Systems with Applications*, *38*(3), 2888–2911. https://doi.org/10.1016/j.eswa.2010.08.083

Zhu, A. Z., Yuan, L., Chaney, K., & Daniilidis, K. (2019). Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 989–997. https://doi.org/10.1109/CVPR.2019.00108