XLI CILAMCE
XLI Ibero-Latin American Congress on Computational Methods in Engineering
16-19 November, 2020 FOZ DO IGUAÇU PARANÁ - BRAZIL

Development of a computational tool in Python language based on Principal Component Analysis, Self-Organizing Map and Support Vector Machines applied to process monitoring

Gabrielle M. da Silva[1], Geovane D. da Silva[1], Lucas O. M. da Silva[1], Dhandara L. C. da Silva[1], Frede O. Carvalho[1]

[1]*Technology Center, Federal University of Alagoas*
Av. Lourival Melo Mota, S/N, 57072-900, Maceió/Alagoas, Brazil
gabriellemelo439@gmail.com, geovanedomi@gmail.com, lucas.mendes149@gmail.com, dhandara24@gmail.com, fredecarvalho@yahoo.com.br

Abstract. Industry is going through a new transformation, called Industry 4.0, in order to absorb recent advances in technology as a way of meeting the constant need for efficient and automated processes. In the chemical industry specifically, these techniques enable to understand the behavior of process guarantee his safety and good performance. There are many methods for implementation of process monitoring system, between techniques of unsupervised and supervised machine learning, such as the Principal Component Analysis (PCA), Self-Organizing Map (SOM) and Support Vector Machines (SVM). In this context, in this paper was studied the development of a computational tool for monitoring the process variables based on cited techniques in order to detect conditions that can affect process performance and, consequently, product quality. The implementation was developed in Python environment and it was applied in a generated fault data from data available in the literature for drying process in an evaporator bed. The statistical metrics F1, accuracy and precision were used to evaluate the techniques and the results indicated that the computational tool showed to be effective on application that was studied. Finally, the tool presented advantages for being free, having an intuitive interface and can be easily used for process monitoring.

Keywords: Process monitoring; Machine Learning; Python Language.

1 Introduction

In the context of the Industry 4.0, the rise in demand for an increasingly safe operation that guarantees the quality of the final product has caused the industries to turn their attention to the monitoring of processes, especially for the fault detection and diagnosis (Nor et al. [1]). This is due to the accompaniment of chemical processes are becoming more complex, with a large number of variables to be analyzed, requiring greater efficiency and safety in operation.

The use of multivariate statistical analysis techniques and supervised and unsupervised machine learning has been the subject of several studies evaluating their potential as tools for monitoring processes (Nor et al. [1]; Zhang et al. [2]). To apply the cited techniques, it is necessary to use some software, generally paid, such as MATLAB®, TIBCO Statistica® and Minitab ®. Some programming languages such as Python and R offer libraries that allow working with these techniques free of charge, however, they have the disadvantage of not offering a friendly interface, which would allow the user to perform analyzes more directly. In reason of this, in this work, the application of the Python language was evaluated to creates a tool for this type of problem, since it is an open source alternative that has been standing out in applications machine learning and science in general.

In the context of fault detection, the work of Zhang et al. [2] explores the application of the Self-Organizing Map (SOM), while the works of Granzotto and Oliveira-Lopes [3] and Barreto et al. [4] study the use of Support Vector Machines (SVM). Among the software mentioned above, only MATLAB® includes support for SOM and SVM. TIBCO Statistica® includes only for SVM and Minitab ® has not support for either techniques. Thus, this paper aimed to develop a computational tool in Python using Principal Component Analysis (PCA), SOM, and

SVM to fault monitoring for industrial applications. So that the user could better interpret and analyze the results, statistical metrics, and confusion matrices were also implemented. The data used for the validation of the tool were from the drying operation of an evaporator bed containing eight monitored variables and 100 samples, available in TIBCO Statistica® 13.5 [5].

2 Materials and methods

2.1 Principal Component Analysis (PCA)

The principal component analysis (PCA) is a multivariate statistical analysis, mainly aimed at the data preprocessing, encompassing the removal of outliers and variables selection. In the technique, a linear transformation is applied to the data set so that each axis, which is now called the principal component (PC), covers the greatest possible variability of the data with the restriction that these axes are orthogonal. This can be seen as an optimization problem to find the axes that maximize the variance subject to orthogonality constraints. The problem is solved by Lagrange multipliers, which is reduced to a problem of eigenvectors and eigenvalues, where eigenvalues are the variances of each axis, and eigenvalues, the vectors that form the basis of the new space (Zhang et al. [2]).

The set of orthogonal vectors obtained are ordered by the amount of variance explained by their directions. Based on this principle, the PCA can be used to reduce the dimensionality of a data set, removing the components with less influence, allowing the reduction of the computational cost for use in learning machines, for example. This characteristic is fundamental when it comes to industrial data since these generally involve a large volume of data with many variables. Moreover, with PCA it is possible to make a qualitative analysis of the weight of each variable in the principal components and, therefore, in the description of the behavior of the data (Nor et al. [1]).

2.2 Self-Organizing Map (SOM)

Based on the self-organization process, as occurs in the human brain, SOM is a type of artificial neural network based on competitive unsupervised learning. The algorithm is capable of mapping a set of high dimensional data through a finite set of neurons normally organized in two-dimensional arrays, called a feature map (Santos et al [6]). Thus, through a neighborhood relationship, data that have similarities are mapped to close regions on the map, making it possible to identify clusters. In the case of a fault detection system, these groupings must correspond to the presence or absence of failures.

The learning process begins with the network initialization, in which random initial values are assigned to the weights. Then, an input vector is presented to the network and the winner neuron (best matching unit, BMU) is identified, which is the one with the greatest similarity between its weight vector and the vector of the presented data. Based on this, the winning neuron and its neighbors, defined according to a neighborhood function, have their weights adjusted to make them more similar to the input vector, this process is repeated until the feature map is complete (Musial and Siqueira [7]).

2.3 Support Vector Machine (SVM)

SVM, developed by Vapnik, is a supervised learning algorithm, which proves to be quite efficient in classification problems, although it can also be used for regression problems. Based on the Theory of Statistical Learning through the minimization of structural risk, the method can be used for both linearly and nonlinearly separable data sets. It has unique advantages in solving small, non-linear, and high-dimensional data classification problems. As explained by Haykin [8], the basic principle of SVM is to map samples to a larger space through a non-linear transformation and to build a hyperplane that acts as a separation surface, in order to maximize the distance between classes and, thus, solve a problem initially of a non-linear nature (Nieto et al [9]; Azimi-Pour et al [10]).

2.4 Description of the data set, data reduction and fault simulation

The data used in the paper are related to the monitoring of a drying process in an evaporator bed, available in a demo of the software TIBCO Statistica® 13.5 [5]. The monitored variables are dewpoint, intake temperature, in-process air temperature, exhaust temperature, mass flow air, bed temperature, filter pressure, and bed pressure, in a set of 100 samples. The data obtained contained outliers, which were removed after statistical treatment using the 3σ-Edit Rule, as used by Borrison et al [11]. In this rule, the sample's absolute deviation from the mean is calculated, if the value is greater than three standard deviations the sample is discarded. This calculation was made for all eight variables and, after that, a set of 97 data was obtained.

Using the PCA, the variables that least influence the variability of the data were analyzed, removing them, and the data set now has only five variables: exhaust temperature, mass flow air, bed temperature, filter pressure and bed pressure. With this reduced set of data, two types of fault were simulated: type 1 fault (F1) occurred in the mass air flow, while type 2 fault (F2) occurred in the filter pressure, as both variables had great influence data variability. Therefore, a set of 40 data were generated for each type of failure, with a change of 15% from its original values. Thus, together with the normal data (N), which has no fault, a total of 177 data was obtained, labeled in N, F1 and F2.

2.5 Evaluation of models

As it is an unsupervised learning machine, the performance analysis of the model obtained from SOM was performed by visual inspection, verifying the distance matrix and how the data were arranged on it, observing if there was grouping data from the same class or overlapping data from different classes. In the case of SVM, as it is a supervised learning machine, it was possible to carry out a quantitative analysis of the results, which was made using the following metrics: accuracy, precision, recall and F1. In addition, the confusion matrix was used.

2.6 Interface preparation and tool construction

Knowing the needs and capabilities of each technique, it was possible to use the Python programming language in the creation of the tool called PyPower PM-Process Monitoring, for this, the libraries were used: Pandas for their ability to provide statistical analysis and handling in sets of data, Numpy that allows the use of vectors, matrices and linear algebra, Scikit-Learn to implement learning machines and the Qt Designer tool for creating GUI's, in which all the properties defined in Qt Designer were dynamically changed within the code in Python language, using the PyQt5 library. With that it was possible to build the interfaces shown in Fig. 1.

Figure 1. Built interface for SOM (a) and SVM (b) analysis

On the home screen, the interface displays the menu, where the user can select between SVM and SOM. In the SOM window, shown in the Fig.1a, it is possible to load the data that will be used and information about the data is shown in the right window. The parameters that must be defined by the user are: the type of normalization, the number of rows and columns on the map, the distance metric, and the neighborhood distance weight. After

that, the program is executed and the graph is generated with the distance matrix, there is the option to save this graph. In the SVM window, shown in the Fig.1b, the data loading is similar to the SOM window. In addition to normalization, the user must define the number of iterations, the tolerance, the test percentage, the kernel, the regularization parameter (C), range, degree, and name of the column containing the classes in the data file. After running the program, pressing the train and test button, the results are shown in two windows, in the first, to the left the following metrics are shown: F1, accuracy, precision, and recall. The confusion matrix is shown in the right window, which can be saved. More information about the developed tool can be found in its repository on Github: https://github.com/GabrielleMelo-Engenharia/PyPower-PM.

3 Results and discussions

The PCA analyzed the variables that most influence the variability of the data without the flaws and with the outliers previously removed. Fig. 2 shows the variables, which were represented by vectors.

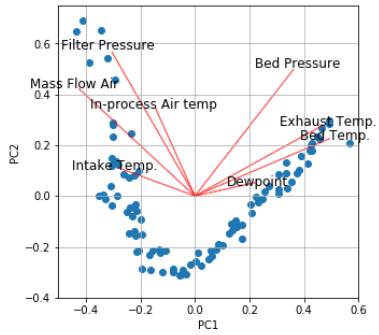

Figure 2. Principal Component Analysis

The vectors of the largest projections of the axes (components) are those that most influence the variability of the data. It is possible to observe by the projection of the vectors on PC1 that the largest were: Exhaust Temperature, Bed Temperature, and Bed Pressure. Similarly, looking at the vectors in PC2, Filter Pressure and Mass Flow Air had higher projections. Therefore, to improve the computational performance in the use of learning machines, the variables that were least influenced (smaller vectors) were removed: In-process Air Temperature, Intake Temperature, and Dewpoint.

With the faults previously simulated, as explained in the methodology, the first menu option takes the window of the SOM unsupervised learning machine in order to cluster the data. In Fig. 3, it is possible to observe the clusters formed in the distance matrix.

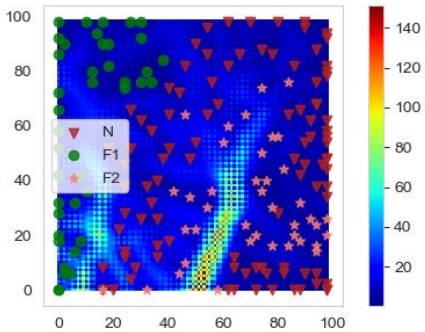

Figure 3. Distance Matrix

It is observed that SOM was able to detect and isolate the two types of simulated faults. The colors with orange tones refer to great distances, in contrast to colors with blue tones, which indicate great similarity between the units. It is possible to identify three groups without overlapping them. Consequently, it can be said that the

developed strategy is sensitive to the types of synthetic flaws used in this investigation.

The second option takes the window of the supervised learning machine SVM in order to classify the data. The performance of the developed classifier was satisfactory for the analysis of the data set with the simulated failures, as illustrated by the confusion matrix of Fig. 4, as well as the model evaluation metrics.

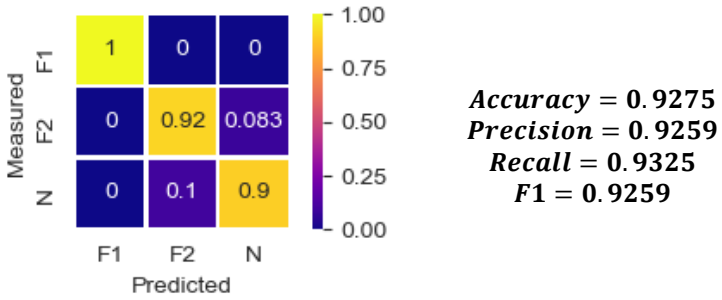

Figure 4. Confusion Matrix and evaluation metrics

4 Conclusions

Therefore, Python proved to be a powerful programming language for being free, for implementing learning machine models, and for creating user-friendly interfaces, such as PyPower PM- Process Monitoring created in this article. It was possible to notice the potential of PCA in assessing the influence of variables on data variability by reducing dimensionality. Moreover, it was observed that SOM proved to be a robust and reliable method in the detection and isolation of failures, in addition to the evaluation metrics used in the SVM had satisfactory results, as they were close to 1. At last, the tool created can be used in an industrial environment to monitor processes, with the advantage of being free and having a user-friendly interface, making it possible to achieve the status of Industry 4.0 more and more.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] N. M. Nor et al. A review of data-driven fault detection and diagnosis methods: Applications in chemical process systems. *Reviews in Chemical Engineering*, v. 36, n. 4, p. 513-553, 2019.
[2] Y. Zhang et al. Intelligent Fault Diagnosis of Engine Based on PCA-SOM. In: *Journal of Physics: Conference Series*. p. 012022, 2020.
[3] M. H. Granzotto; L. C. Oliveira-Lopes; Desenvolvimento de Sistema de Detecção de Falhas Baseado em Aprendizado Estatístico de Máquinas de Vetores de Suporte, p. 11819-11828. In: *Anais do XX Congresso Brasileiro de Engenharia Química - COBEQ 2014*. São Paulo: Blucher, 2015.
[4] L. C. Barreto et al. Aplicação de Support Vector Machine na Detecção de Falhas em Sensores de uma Coluna Debutanizadora, p. 3003-3009. In: *Anais do XIII Congresso Brasileiro de Engenharia Química em Iniciação Científica.* São Paulo: Blucher, 2019.
[5] TIBCO Statistica®. *Industrial Evaporador Data File*. TIBCO Inc, 2018.
[6] A. B. Santos et al. Previsão de surtos epiléticos usando Mapas Auto Organizáveis executados diretamente em hardware. In: *Anais do XIV Brazilian e-Science Workshop*. SBC, 2020. p. 57-64.
[7] J. E. Musial; P. H. Siqueira. Aplicação de caos em redes neurais auto-organizáveis para resolver problemas de otimização. In: *Proceedings of the XXXVI Ibero-Latin American Congress on Computational Methods in Engineering-CILAMCE*. 2015.
[8] S. Haykin. *Neural Networks and Learning Machines*. 3ª ed. Pearson Prentice Hall, 2009.
[9] P. J. G. Nieto et al. A new predictive model for the cyanotoxin content from experimental cyanobacteria concentrations in a reservoir based on the ABC optimized support vector machine approach: A case study in Northern Spain. *Ecological Informatics*, v. 30, p. 49-59, 2015.
[10] M. Azimi-Pour et al. Predição linear e não linear de SVM para propriedades frescas e resistência à compressão de concreto autoadensável de cinza volante de alto volume. *Construção e Materiais de Construção*, v. 230, 2020.
[11] R. Borrison et al. Data Preparation for Data Mining in Chemical Plants using Big Data. In: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN).* IEEE, 2019. p. 1185-1191.