COMPUTATIONAL MODEL FOR ELECTRICAL MOTORS CONDITION ANALYSIS AND MONITORING

Álvaro Souza de Araújo[1], Oberdan Pinheiro Rocha[a], Alex Álisson Bandeira Santos[a]

[a] *SENAI CIMATEC, Brazil*

Abstract: Predictive maintenance uses several methods to monitor the conditions of electric motors applied in industrial plants. Among these methods, vibration analysis stands out as a widely used method due to the possibility of identifying a wide chance of failures. This work presents the development of a computational model endowed with a classifying algorithm to receive vibration readings from electric motors and determine if the machine has a fault behavior and, if so, which fault. For the algorithm to detect the type of failure, a dataset with readings from several engines in different failure conditions should be developed for training the model.

Keywords: Vibration analysis; Condition-based monitoring; Feature extraction; Classifier agent.

MODELO COMPUTACIONAL PARA ANÁLISE DE CONDIÇÕES E MONITORAMENTO DE MOTORES ELÉTRICOS

Resumo: A manutenção preditiva utiliza de diversos métodos para acompanhamento das condições dos motores elétricos aplicados em plantas industriais. Dentre estes métodos, destaca-se a análise de vibração como método amplamente utilizado devido à possibilidade de identificação de uma vasta possibilidade de falhas. Este trabalho apresenta o desenvolvimento de um modelo computacional dotado de um algoritmo classificador para receber leituras de vibração de motores elétricos e determinar se a máquina apresenta um comportamento de falha e, em caso positivo, qual falha. Para que o algoritmo seja capaz de detectar o tipo de falha, será desenvolvido um dataset com leituras de diversos motores em variadas condições de falha para treinamento do modelo.

Palavras-chave: Análise de vibração; Monitoramento baseado em condições; Extração de *feature*; Agente classificador.

1. INTRODUCTION

Maintenance is a critical and constantly evolving point in industries. Even with many technological advances, it is still a challenge to predict failures before they happen. Predictive maintenance is essential, especially in the industrial environment, where the failure of certain equipment that results in the need for corrective maintenance can cause the partial or total stoppage of the production plant, which causes inconvenience and losses for the company.

To mitigate the possibility of failure, vibration analysis in electric motors is used. This method is an important tool for diagnosing operating conditions. Many engine problems reflect directly on their vibration due to several factors: misalignment, cavitation, clearance, among others. Each component of a motor has a signature at a different frequency, thus, from the analysis of the behavior of the vibration profile of the equipment, it is possible to detect the type of fault present in the machine.

Current studies show classification algorithms specialized in a specific failure, which does not allow their implementation in an industrial environment to obtain indications of several failures, since the application of an electric motor can induce defects in different components of this equipment.

To cover this gap, this research project aims to develop a classifier algorithm that, from a dataset containing readings from different equipment, will determine the current condition of an electric motor. To make this research feasible, the dataset must be obtained in a laboratory, with the aid of industrial vibration analysis equipment.

Different classification methods will also be evaluated, to obtain what will bring greater precision and accuracy in the detection of the failure, in the shortest execution time. After training the network using the model that best fits the problem solving, the technical feasibility of the proposed model will be analyzed based on laboratory tests.

2. METHODOLOGY

2.1. Classifier algorithms

Classifier algorithms are those that aim to predict the class of a new piece of data, based on learning about similar data. They are a subcategory of supervised learning algorithms, where the objective is to predict the category of a new data based on past observations [1]. The performance, accuracy, and precision of three classification algorithms will be evaluated in this work: Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Artificial Neural Networks (ANN).

2.1.1. SVM

Support Vector Machine (SVM) is a classification and regression tool that uses machine learning theory to maximize prediction accuracy, while automatically avoiding overfitting the network [2]. This algorithm performs the learning by assigning annotations to the data [3]. Also, according to [3], SVM can be used in several applications, from the recognition of fraudulent credit card transactions, detection of

ISSN: 2357-7592

handwriting patterns, through images, to applications in biology, to recognize anomalies in DNA.

SVM algorithms are well known for their excellent performance in the sphere of statistical classification. Still, the high computational cost due to the complexity of the cubic runtime is problematic for big data sets: training the SVM classifier requires solving a quadratic optimization problem [4].

Given an N number of training data (x_i, y_i) for $i = 1\ to\ N$, where xi is the input data and $y_i \in \{+1, -1\}$ corresponds to its target value, the algorithm tries to find the hyper plane in such a way that the margin (distance perpendicular to the hyper plane) between support vectors is maximized [5]. If the data are linearly separable, there is a hyper plane, which can be expressed in the equation (1), considering $w \in R^N$ as the weight parameter.

$$f(x) = w^T x + b = \sum_{i=1}^{N} w_i x_i + b = 0 \quad (1)$$

The performance of SVM is most influenced by regularization parameter C and kernel parameter y [6]. The C is used to control the trade-off between maximizing the margin and minimizing the training error. For a given problem, if C is too large, SVM may store many support vectors and it may be over fit. If C is too small, SVM may again not fit properly, or it is under fit [6].

2.1.2. RNA

Artificial Neural Networks can be defined as machines designed to model the working principle of a brain to perform a task [7]. The basic elements of this algorithm are the input connections, where each one has a weight to be determined, subsequent layers of neurons, an accumulator element to concentrate the signals and, finally, an activation function, which can take different formats, to then, present the output value.

One of the biggest difficulties encountered in the use of neural networks is choosing the best architecture, since this process is experimental and requires a great deal of execution time [8]. Thus, the use of this technique requires extensive tests with different configurations to, experimentally, obtain the model best adapted to the problem.

Neural networks are similar to the human brain in two basic aspects: knowledge is acquired by the network from its environment, through the learning process, and connection forces between neurons (synaptic weights) are used to store the acquired knowledge [9]. At the output of each neuron, the generated signal goes through an activation function, responsible for weighing the effect of each output on the subsequent layer.

2.1.2. KNN

For pattern recognition, the KNN algorithm is a method to classify objects based on training examples that are closer in space [10]. This model was proposed by [11] and is a simple to implement classifier that can get very accurate results depending on the application.

This algorithm sorts data in a dataset based on proximity to already sorted data. Thus, the number of neighbors that must be considered for the classification of a later data is determined as a parameter. This rule simply retains the entire training set during learning and assigns each query a class represented by the label of most of its closest neighbors in the training set [10].

The KNN classification method also has good accuracy and precision compared to the previously mentioned methods. The algorithm consists of estimating the distance between data to determine classification limits. It is important to emphasize that it is possible to graphically represent the data separation limits represented by up to three dimensions. With higher dimensions, the method can obtain the classification, however, the representation of more dimensions does not make physical sense.

To determine the degree of similarity of an entry in question with its closest neighbors, there are different methods for calculating the distance between the data, including the Euclidean distance, Hamming distance and Manhattan distance, the first being the most used with input variables of the same type.

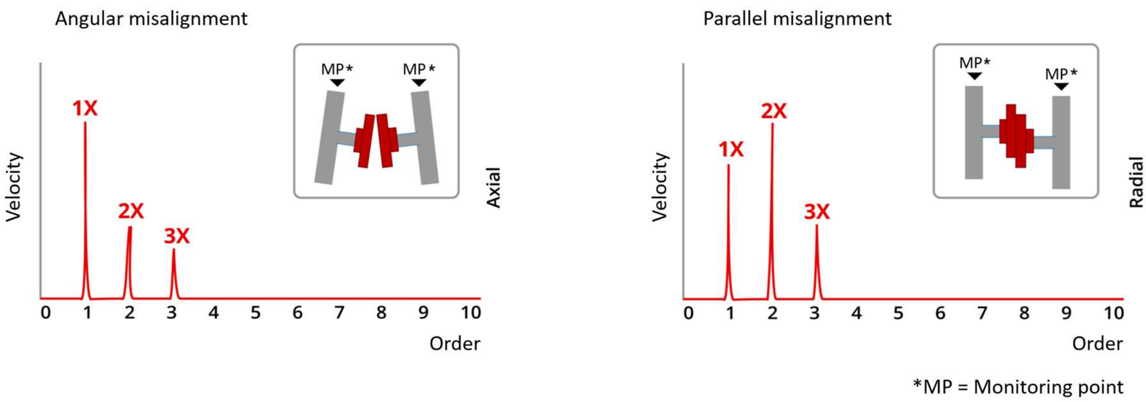
2.2. Vibration analysis

Every machine has noise and vibration due to its own operation and external excitations. However, a portion of the vibrations is due to small defects that compromise the performance of the equipment [12]. In this way, each failure behavior has a different frequency spectrum, which allows the distinction of the type of failure and, also, knowing the machine operation specifications, the degree of severity.

For the detection of failures of mechanical origin, vibration analysis is used, where, by means of equipment equipped with an accelerometer sensor, a vibration signal in acceleration, velocity or displacement is measured, to then observe these data in the spectrum of time or frequency.

In Figure 1 it is possible to visualize the behavior in the frequency spectrum of two faults: the angular and parallel misalignment, where 1x represents the dominant operating frequency of the motor. If it is powered directly from a 60 Hz electrical grid, this will be its dominant frequency.

Figure 1. Angular and parallel misalignment and their frequency behaviors



2.3. Dataset

To implement and train this model, a dataset MaFaulDa [13] was selected, which contains 1951 multivariable series containing vibration sensor readings. This dataset covers the following equipment conditions: normal operation, unbalance, axial and radial misalignment, and internal and external bearing failures.

Each reading contained in this dataset is composed of three single-axis accelerometers, model IMI 601A01, each positioned on the radial, axial, and tangential axes, a triaxial accelerometer, model IMI 604B31, an analog tachometer, model MT-190 and a microphone Shure SM81 model. The dataset contains data referring to multiple rotation frequencies, ranging from 737 to 3686 rpm.

The first sensor, the uniaxial accelerometer, has a sensitivity of 100 mV/g, being able to measure accelerations between -50 g and +50 g, and with a frequency spectrum from 0.27 Hz to 10,000 Hz. The operating principle of this equipment is a piezoelectric sensor, where a mass is in direct contact with the piezoelectric element and, when subjected to an acceleration, this mass exerts a mechanical force on the load that can be converted into an electrical signal, thus being readable by means of a converter analog to digital connected to a controller. This type of sensor is suitable for industrial applications that require high frequency response while maintaining stable responses in varying temperature environments.

The second sensor, the triaxial accelerometer, also has the same sensitivity as the previous one (100 mV/g) and the same measurement range (-50 g to +50g), however its frequency response varies between 0.5 and 5,000 Hz. it has a ceramic sensor element, which results in low interference from external noise, in addition to suffering minimal reading variations when subjected to different temperatures.

2.3. Proposed model

The first step to process the entire dataset is to perform the Fast Fourier Transform in all signals to extract the frequency spectrum. Then, the preprocessing algorithm will retrieve statistic features, such as standard deviation, kurtosis, mean, skewness, and variance. All these information will be stored in an ARFF file that will be used to train the classifiers.

With the aid of the Weka machine learning software [14], the next step is to perform an analysis to determine which features are relevant to the models. Therefore, another ARFF file is generated including the most relevant features. Then, the filtered dataset is used to train classifiers using KNN, SVM and RNA algorithms.

The RNA algorithm was configured with a 0.3 learning rate, using one hidden layer containing 23 neurons, as per described in the equation (2). Furthermore, the classifier was limited to 500 epochs as stopping criteria.

$$n_{neurons} = \frac{(classes + attributes)}{2} \tag{2}$$

Since the KNN classifier is a simpler type of classifier, it has fewer parameters to adjust. The model used 5 nearest neighbors, with the Euclidian distance. At last, the SVM model used the polynomial kernel, with the C parameter set to 0.1.

ISSN: 2357-7592

3. RESULTS AND DISCUSSION

After evaluating the accuracy of all three models, the SVM presented the best accuracy for the dataset, as per Table 1. It is important to note that the time required to train the dataset using this classifier is comparable to the RNA, however considerably larger than KNN, due to the nature of algorithm, that demands higher computational power as the number of neurons and layers increase.

Table 1. Accuracy of the classifier algorithms

Algorithm	Accuracy
SVM	95.6433%
RNA	94.0031%
KNN	91.3378%

Comparing the results, it is possible to note that the SVM algorithm performed the best, considering the accuracy. Furthermore, in the confusion matrix, the entries, in general, were classified with minimal error, with the class IMBALANCE reporting the lowest score of 93.09% of correct classifications.

Other classes such as UNDERHANG_BEARING_CAGE_FAULT and NORMAL contained, respectively, 93.61% and 93.09% of accuracy, which slightly decreased the overall accuracy of the model, at 95.64%. However, the other classes were correctly classified with over 96% of certainty, as per the Figure 2.

Figure 2. Confusion matrix from the results of the SVM classifier

```
  a   b   c   d   e   f   g   h   i   j   <-- classified as
 46   1   1   0   1   0   0   0   0   0 |   a = NORMAL
  5 187   1   0   2   0   0   2   0   0 |   b = HORIZONTAL_MISALIGNMENT
  0  10 285   0   6   0   0   0   0   0 |   c = VERTICAL_MISALIGNMENT
  1   6   7 310   8   0   0   1   0   0 |   d = IMBALANCE
  4   4   0   1 176   1   0   1   1   0 |   e = UNDERHANG_BEARING_CAGE_FAULT
  2   0   0   0   0 182   0   0   0   0 |   f = UNDERHANG_BEARING_OUTER_RACE
  1   0   0   0   0   0 185   0   0   0 |   g = UNDERHANG_BEARING_BALL_FAULT
  4   1   0   0   0   0   0 178   5   0 |   h = OVERHANG_BEARING_CAGE_FAULT
  1   1   0   0   2   0   0   4 180   0 |   i = OVERHANG_BEARING_OUTER_RACE
  0   0   0   0   0   0   0   0   0 137 |   j = OVERHANG_BEARING_BALL_FAULT
```

The second most accurate result was provided by the RNA classifier. In its confusion matrix, it is shown that the algorithm obtained 55.11% of correct classifications for the class NORMAL, which impacted negatively to the lower global accuracy of the model. However, the other classes presented over 91% of accuracy, as show on the confusion matrix for this algorithm at Figure 3.

Figure 3. Confusion matrix from the results of the RNA classifier

a	b	c	d	e	f	g	h	i	j	<-- classified as
27	7	3	6	5	0	0	0	1	0	a = NORMAL
4	175	11	0	2	1	0	2	1	1	b = HORIZONTAL_MISALIGNMENT
0	4	286	1	4	0	3	2	0	1	c = VERTICAL_MISALIGNMENT
0	3	1	323	1	0	2	1	1	1	d = IMBALANCE
1	3	5	6	167	1	0	4	1	0	e = UNDERHANG_BEARING_CAGE_FAULT
0	0	0	0	0	184	0	0	0	0	f = UNDERHANG_BEARING_OUTER_RACE
0	0	0	0	0	0	186	0	0	0	g = UNDERHANG_BEARING_BALL_FAULT
0	1	7	2	3	2	0	172	1	0	h = OVERHANG_BEARING_CAGE_FAULT
1	1	1	0	2	0	0	6	177	0	i = OVERHANG_BEARING_OUTER_RACE
0	0	0	0	0	0	0	0	0	137	j = OVERHANG_BEARING_BALL_FAULT

At last, the KNN classifier presented the lowest accuracy from the three, but close to the previous RNA values. Differently from the previous results, this model could classify 42.89% of the entries from the NORMAL class correctly, however this still indicates that this model encounter difficulty distinguishing NORMAL and other classes such as HORIZONTAL_MISALIGNMENT, with 78.71% of accuracy. The other classes presented over 89% of accuracy, as shown on Figure 4.

Figure 4. Confusion matrix from the results of the KNN classifier

a	b	c	d	e	f	g	h	i	j	<-- classified as
21	11	6	0	11	0	0	0	0	0	a = NORMAL
9	155	20	2	5	0	0	4	2	0	b = HORIZONTAL_MISALIGNMENT
0	8	288	3	0	0	1	1	0	0	c = VERTICAL_MISALIGNMENT
0	2	14	313	2	0	0	1	1	0	d = IMBALANCE
3	10	1	7	160	1	0	3	3	0	e = UNDERHANG_BEARING_CAGE_FAULT
0	0	0	0	0	183	0	1	0	0	f = UNDERHANG_BEARING_OUTER_RACE
0	0	0	0	0	0	186	0	0	0	g = UNDERHANG_BEARING_BALL_FAULT
0	7	0	0	2	1	0	169	9	0	h = OVERHANG_BEARING_CAGE_FAULT
1	2	0	1	1	0	0	13	170	0	i = OVERHANG_BEARING_OUTER_RACE
0	0	0	0	0	0	0	0	0	137	j = OVERHANG_BEARING_BALL_FAULT

4. CONCLUSION

The proposed model was able to correctly identify different types of mechanical faults on electrical motors using vibration analysis readings. To obtain this result, an initial processing step was required to transform the signals to the frequency spectrum, extract relevant statistical features and to filter the amount of data provided to train the classifiers.

The model was trained to detect faults in multiple rotational speeds on the motor, to provide a better accuracy in a real-world scenario, since in many applications, the equipments' speed is controlled by frequency inverters.

Furthermore, the classifier was tested under three different algorithms to evaluate which one provides a better result: SVM, KNN and RNA. Tests were done to determine the best parameters for all three classifiers.

From the results, it is possible to note that the SVM algorithm is better suited for classifying this dataset and detect with better accuracy which failure a vibration signal contains, despite the higher computational requirements, especially since the other algorithms such as KNN and RNA presented a lower accuracy with some classes that are relevant to the failure detection.

5. REFERENCES

[1] RASCHKA, Sebastian. Python machine learning. Packt publishing ltd, 2015.

[2] JAKKULA, Vikramaditya. Tutorial on support vector machine (svm). School of EECS, Washington State University, v. 37, 2006.

[3] NOBLE, William S. What is a support vector machine?. Nature biotechnology, v. 24, n. 12, p. 1565-1567, 2006.

[4] CHAPELLE, Olivier et al. Choosing multiple parameters for support vector machines. Machine learning, v. 46, n. 1, p. 131-159, 2002.

[5] KUMAR, Anil; KUMAR, Rajesh. Time-frequency analysis and support vector machine in automatic detection of defect from vibration signal of centrifugal pump. Measurement, v. 108, p. 119-133, 2017.

[6] CHANG, Chih-Chung; LIN, Chih-Jen. LIBSVM: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), v. 2, n. 3, p. 1-27, 2011.

[7] JUNIOR, Ronny Francis Ribeiro; DE ALMEIDA, Fabricio Alves; GOMES, Guilherme Ferreira. Fault classification in three-phase motors based on vibration signal analysis and artificial neural networks. Neural Computing and Applications, v. 32, n. 18, p. 15171-15189, 2020.

[8] HAYKIN, Simon S.; VAN VEEN, Barry. Sinais e sistemas. Bookman, 2001.

[9] FLECK, Leandro et al. Redes neurais artificiais: Princípios básicos. Revista Eletrônica Científica Inovação e Tecnologia, v. 1, n. 13, p. 47-57, 2016.

[10] IMANDOUST, Sadegh Bafandeh; BOLANDRAFTAR, Mohammad. Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. International Journal of Engineering Research and Applications, v. 3, n. 5, p. 605-610, 2013.

[11] FUKUNAGA, Keinosuke; NARENDRA, Patrenahalli M.. . A branch and bound algorithm for computing k-nearest neighbors. IEEE transactions on computers, v. 100, n. 7, p. 750-753, 1975.

[12] DA SILVA, Luiz Eduardo Borges et al. Desenvolvimento de um Sistema de Detecção de Defeitos Precoces em Motores de Indução.

[13] RIBEIRO, Felipe ML et al. Rotating machinery fault diagnosis using similarity-based models, 2017.

[14] FRANK, Eibe et al. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.