Detection of Vibration Faults in Induction Motors Using Automatic Features Selection

Detección De Fallos Por Vibración En Motores De Inducción Utilizando Selección Automática De Características

Salvador Calderon Uribe[1], Uriel Calderon Uribe[2], Irving Armando Cruz Albarran[13]
[1]Laboratory of Artificial Vision and Thermography/Mechatronics, Faculty of Engineering, Autonomous University of Queretaro, Campus San Juan del Rio, San Juan del Río 76807, Mexico.
[2]Instituto Tecnológico Superior del Sur de Guanajuato, Educación Superior 2000, Benito Juárez, 38980, Uriangato, Guanajuato, México.
[3]Artificial Intelligence Systems Applied to Biomedical and Mechanical Models, Faculty of Engineering, Autonomous University of Queretaro, Campus San Juan del Rio, San Juan del Rio 76807, Mexico.
[1]scalderon02@alumnos.uaq.mx,[2]u.calderon@itsur.edu.mx, [13]irving.cruz@uaq.mx

Abstract

The use of machine learning techniques in the diagnosis of induction motors (IM) is becoming increasingly common in modern industry. Employing the right indicators that reflect the behavior of IMs directly impacts the accuracy and effectiveness of diagnostic systems, enabling not only a reduction in maintenance costs but also an improvement in operational efficiency and safety in industrial operations. However, identifying these indicators is complex and often leads to the choice of more robust algorithms, which in turn complicates the implementation of models in real-world environments. Therefore, this work focuses on developing a methodology for fault detection through vibration in IMs using random forest and logistic regression for automatic feature selection, and support vector machines, K-nearest neighbors, and logistic regression as classification models. The results demonstrate the importance of identifying these features and how their synergy improves accuracy and effectiveness in fault classification.

Key Words: Fault Detection, Induction Motors, Feature Selection, Machine Learning algorithms.

Introduction

Induction motors (IM) are one of the most used rotating machines in industrial applications [1]. Due to their widespread use in industry, it is necessary to perform preventive maintenance tasks to ensure optimal performance and reduce the risk of unexpected failures. In this context, advancements in Machine Learning (ML) and Deep Learning (DL) technologies have revolutionized the way IM maintenance is managed, thanks to their ability to analyze large amounts of data and predict potential failures before they occur. Various studies in the literature focus on the analysis and prediction of IM faults using different ML models; for example, Wang [2] developed a fault severity detection model based on K-Nearest Neighbors (K-NN) using redundant statistical features estimated from the wavelet packet transform. Deng et al. [3], on the other hand, proposed the particle swarm optimization algorithm and a least squares-based support vector machine (SVM) model to diagnose motor bearing faults. Similarly, Toma et al. [4] proposed a hybrid approach using statistical features, genetic algorithms, and models such as KNN, decision trees, and random forests to identify bearing faults. It is important to highlight that, although there are numerous studies implementing different classification models, the appropriate features are a crucial part of the model's performance, as they are responsible for capturing and representing the system's behavior. Shen et al. [5] use statistical features from a wavelet packet transform for the detection of multiple faults. Similarly, Glowacz A. [6] employs acoustic signals as the main tool for diagnosing faults in the bearings, stator, and rotor of a single-phase IM. Lizarraga-Morales et al. [7], on the other hand, use homogeneity as the main indicator for detecting and classifying the severity of broken rotor bars in IMs. Likewise, Calderon-Uribe et al. [8] utilize texture features based on the Haralick co-occurrence matrix and SVM to detect imbalances in IMs through vibration signals. Identifying the ideal features for detecting different faults in an IM requires extensive knowledge of the machine's dynamic and operational behavior under various conditions, often necessitating the use of models that allow automatic feature extraction [9-10], or the implementation of automatic feature selection models [11-13], which enable the identification of the most relevant signals indicating malfunction. Given the above, this study focuses on developing a methodology for fault detection through vibration in IMs using statistical features and automatic

selection models. The aim is to observe how these features affect both the performance of classification models and the computational complexity.

The organization of this work is described as follows. Section 2 outlines the methods used to prepare and train the various feature selection models and fault classification models. Section 3 presents the results and considerations of this study. Finally, Section 4 provides a brief conclusion regarding the work.

Proposed Methodology

This section describes the proposed methodology for feature selection and fault classification. Figure 1 provides an overview of the proposed methodology. As shown in the figure, the classification system is divided into three distinct stages: feature extraction, feature selection, and fault classification. In the first stage, features are extracted from the dataset generated by [14] using sliding windows. In the second stage, using nonlinear methods (random forests) and linear methods (logistic regression), the extracted features are weighted and ranked based on their importance for prediction. Finally, in the third stage, the selected features are evaluated by measuring the performance across different classification models.

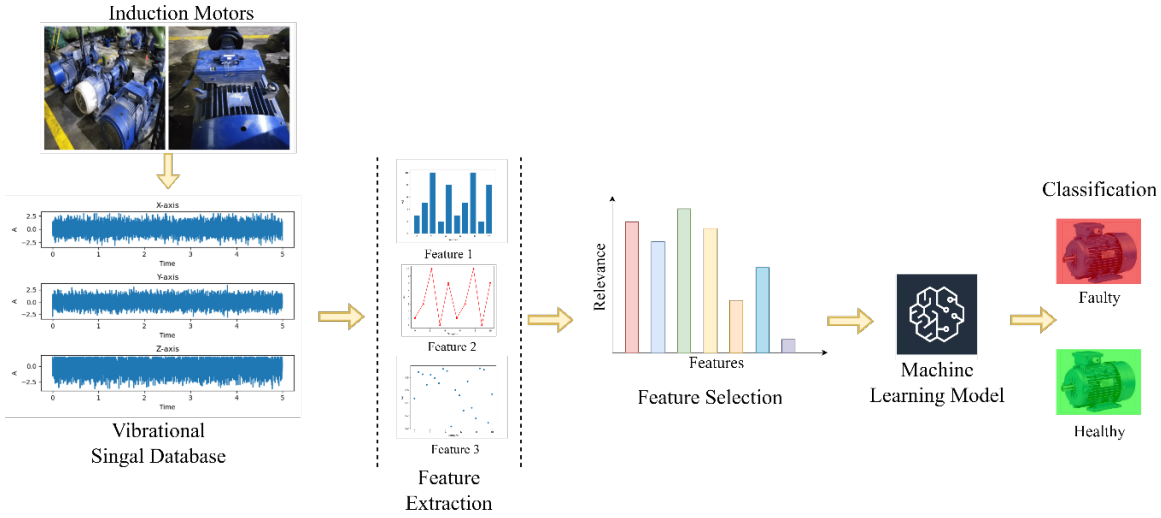

Figure 1. General methodology based on [14] (own authorship).

Dataset Description

This study is based on the dataset documented in [14], which contains 220 vibration signals collected for the purpose of diagnosing faults in large industrial induction motors. These signals were obtained from 45KW three-phase motors coupled with centrifugal water pumps, using a triaxial wireless accelerometer, Beanscape, at a sampling frequency $f_s = 1000\text{Hz}$ for each of the axes (X, Y, Z). The dataset includes two operating conditions of the machine: healthy state and faulty state (including bearing and alignment faults). The healthy state contains 103 signals (for each axis), each with a duration of 5 seconds, 8.5 minutes in total. On the other hand, the faulty state contains 117 signals, also with a duration of 5 seconds, amounting to a total of 9.75 minutes. For this study, only the signals from the X-axis will be used, with the aim of reducing the system's complexity.

Feature Extraction

In a ML-based model, feature extraction is the main phase responsible for transforming the input signals into a more representative and relevant dataset for the problem at hand. In the context of fault detection in IMs

through vibration signals, features in both the time domain and frequency domain have been well studied, yielding favorable results in classification, whether by combining features from both domains or using them separately [15, 16, 17]. However, for this study, an analysis will be conducted using features exclusively in the time domain, with the aim of simplifying subsequent stages and reducing computational complexity. Some of the most commonly used time-domain features are the statistical features listed in Table 1, including features such as kurtosis, which describes the "sharpness" or "peakedness" of a signal's distribution compared to a normal (Gaussian) distribution; skewness, which indicates whether the signal's distribution is skewed to the right or left; and standard deviation, which measures the variation with respect to the mean, among others [18].

***Table 1**. Implemented Features (own authorship).*

Feature	Equation
Mean	$\bar{x} = \frac{1}{n}\sum_{i=0} y(i)$
Standard Deviation	$s = \left(\frac{1}{n-1}\right)\sum_{i=0}(y(i) - \bar{x})^2$
RMS	$rms = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n-2} y(i)^2}$
Creast Factor	$Cf = \frac{\max(y)}{RMS(y)}$
Kurtosis	$K = \frac{\frac{1}{n-1}\sum_{i=0}^{n-1}(y(i) - \bar{x})^4}{\left(\frac{1}{n-1}\sum_{i=0}^{n-1}(y(i) - \bar{x})^2\right)^2}$
Skewness	$skw = \frac{\frac{1}{n-1}\sum_{i=0}^{n-2}(y(i) - \bar{x})^3}{\left(\frac{1}{n-1}\sum_{i=0}^{n-2}(y(i) - \bar{x})^2\right)^{3/2}}$

Each of these features was extracted from the signals of both classes using sliding windows. Since the signals were acquired at a sampling period of $\boldsymbol{T_s = 1ms}$, a window of 250 samples was proposed, i.e., a window with a resolution of 250 ms, aiming to identify bearing and alignment faults, which often produce repetitive patterns. A 250 ms window without overlap allows for capturing enough samples to identify these patterns. By obtaining 20 samples from each signal and 6 features from each sample, the resulting feature matrix combining both classes reached the shape of (7).

$$\begin{bmatrix} \boldsymbol{x_{1,1}} & \boldsymbol{x_{1,2}} & \cdots & \boldsymbol{x_{1,n}} \\ \boldsymbol{x_{2,1}} & \boldsymbol{x_{2,2}} & \cdots & \boldsymbol{x_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x_{m,1}} & \boldsymbol{x_{m,2}} & \cdots & \boldsymbol{x_{m,n}} \end{bmatrix} \quad (7)$$

where n denotes the number of features $(n = 6)$ and m denotes the number of samples $(m = 4400)$.

Random Forest Feature Selection

A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. The predictions made by each tree are averaged with the others, significantly improving overall performance [19]. The ability of a decision tree to select features in a classification problem is calculated as the average reduction in impurity (e.g., entropy or Gini index) that each feature achieves in the decision tree. The importance of each feature is weighted with values between 0 and 1, where 0 indicates "not used at all" and 1 indicates "perfectly predicts the target" [19, 20]. From the perspective of a random forest, the importance of the features is calculated as the average reduction in impurity that each feature achieves across the different trees in the forest [19]. In this context, to determine the level of importance of each feature, a random forest composed of 100 trees was employed, using the Gini index as the weighting criterion. In binary classification, the Gini index can be formulated as (8).

$$Gini = p_1(1 - p_1) + p_2(1 - p_2) \quad (8)$$

Where p_1 and p_2 are the probabilities of class 1 and class 2, respectively.

Logistic Regression

Logistic regression is an extension of linear regression that is used when the dependent (or target) variable to be predicted is categorical. Logistic regression models the probability that the dependent variable Y takes a value of 1 given a set of features X [21]. The logistic (or sigmoid) function is defined as:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta X)}} \quad (9)$$

Where $P(Y = 1|X)$ is the probability that Y is 1 given X, β_0 is the intercept or constant term of the model, and β is the vector of coefficients for the set of features X.

From a feature selection perspective, the value of the β coefficients could be interpreted as the level of importance of the features. However, for a more precise evaluation, each β coefficient is assessed with a significance test. This test evaluates the null hypothesis that the coefficient β_i equals zero, meaning that the feature X_i has no effect on the dependent variable. To obtain the significance value, the β coefficients are estimated using the logistic regression model on the feature matrix, and for each β_i, the z-statistic is calculated from (10)

$$z = \frac{\hat{\beta}_i}{SE(\hat{\beta}_i)} \quad (10)$$

Where $\hat{\beta}_i$ is the estimated value of the coefficient for feature X_i, and $SE(\hat{\beta}_i)$ is the standard error associated with that estimate. Once the z-value is obtained, the standard normal distribution is used to calculate the p-value associated with the z-statistic. If the $p < 0.05$, the null hypothesis is rejected, suggesting that feature X_i has a significant impact on the prediction.

Classification Models

To analyze the impact of each feature on the performance of the classification models, three different models were developed: Logistic Regression (LR) [21], SVM [22], and KNN [23]. To find the best configuration for each model, an exhaustive hyperparameter search was conducted using grid search [24], evaluating the optimal combinations described in Table 2. Finally, for each model, the data distribution followed a proportion of 80%-20%, with 80% used for model training and 20% for evaluation.

***Table 2**. Grid search hyperparameters (own authorship).*

Model	Hyperparameters
SVM	Regularization parameter (C): 0.01, 0.1, 1,10 Kernel: linear, rbf, sigmoid
LR	Regularization parameter(C): 0.01, 0.1, 1,10 Max. Iterations: 10,100,500 Optimization Function: liblinear, lbfgs
KNN	Number of Neighbors: 5,7,9,11

Results

Random Forest Feature Selection and Classification Results

Figure 2 graphically shows the importance level of each feature, using the parameters described above.

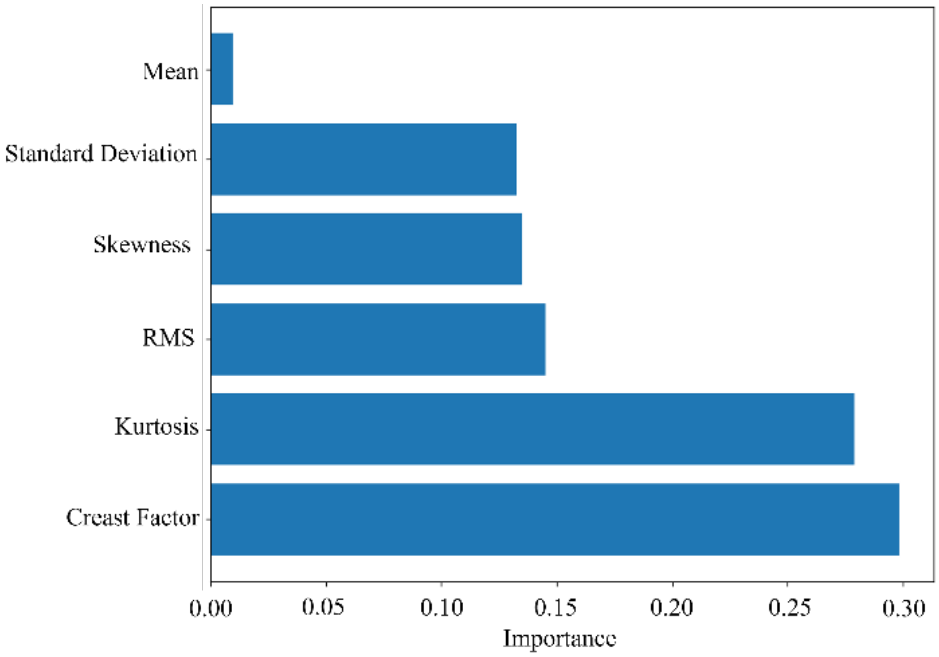

Figure 2. Feature importance using random forest model (own authorship).

From this figure, it is possible to observe that the crest factor is the feature that contributes the most to fault identification, followed by kurtosis and the root mean square (RMS). Similarly, the mean has the lowest importance value. However, a low importance value does not mean that this feature is not informative; rather, the tree likely did not select that feature because another feature probably encodes the same information. Finally, the performance of each model was evaluated using 2, 3, and 6 features, selected based on the importance obtained through the random forest (Table 3).

***Table 3**. Model performance (own authorship).*

Model	Best Hyperparameters	Features	Perfomance
SVM	C: 0.1 Kernel: rbf	Crest factor, Kurtosis	Accuracy: 86.02% Precision: 100% Recall: 74.57% F1 Score: 85.43%
LR	C: 1 Max. iteration:100 Activation function: lbfgs		Accuracy: 85.34% Precision: 93.20% Recall: 78.38% F1 Score: 85.15%
KNN	Neighbors: 5		Accuracy: 85.34% Precision: 88.53% Recall: 83.47% F1 Score: 85.93%
SVM	C: 10 Kernel: rbf	Crest factor, Kurtosis, RMS	Accuracy: 98.18% Precision: 97.89% Recall: 98.72% F1 Score: 98.31%
LR	C: 10 Max. iteration:100 Activation function: lbfgs		Accuracy: 94.43% Precision: 96.28% Recall: 93.22% F1 Score: 94.72%
KNN	Neighbors: 7		Accuracy: 97.84% Precision: 97.68% Recall: 98.30% F1 Score: 97.99%
SVM	C:10 Kernel: rbf	All features	Accuracy: 98.29% Precision: 98.30% Recall: 98.51% F1 Score: 98.41%
LR	C:10 Max. iterations: 100 Activation function: lbfgs		Accuracy: 93.52% Precision: 94.62% Recall: 93.22% F1 Score: 93.91%
KNN	Neighbors: 3		Accuracy: 98.06% Precision: 96.65% Recall: 98.09% F1 Score: 97.37%

From the previous table, it is possible to observe the best hyperparameter configuration that maximizes performance based on the available features. Additionally, in this system, using all the features does not significantly affect the performance of the models compared to using only the top 3 most relevant features identified by the selector.

Logistic Regression Feature Selection and Classification Results

Table 4 indicates the significance obtained for each feature based on its coefficient value β_i. From this table, it can be observed that only Kurtosis and Skewness have a p-value < 0.05, indicating that both variables have a significant influence on the prediction of the target variable, while the others do not contribute statistically significantly to the model.

***Table 4**. Feature selection by logistic regression (own authorship).*

Features	Coef.	SE	z	P>|z|
Kurtosis	-1.3210	0.113	-11.6994	0.000
Skewness	-11.1879	0.411	-27.218	0.000
RMS	46.4860	111.094	0.418	0.676
Creast Factor	-0.0997	0.090	-1.104	0.270
Mean	-3.6576	2.482	-1.474	0.141
Standard Deviation	-47.1011	111.115	-0.424	0.672

Evaluating the models using the features that passed the significance test (Table 5), it was observed that the performance obtained by these features was superior (in some metrics and in most models) to the combination of the most relevant features selected by the random forest model. However, the random forest model provides greater robustness, stability in feature selection, and more flexibility in scenarios where the combination of several features improves model performance.

***Table 5**. Results of the evaluation using the most significant features (own authorship).*

Model	Best Hyperparameters	Features	Performance
SVM	C:0.1 Kernel: rbf	Kurtosis and Skewness	Accuracy: 89.77% Precision: 93.21% Recall: 87.28% F1 Score: 90.15%
LR	C:10 Max iteration: 100 Solver: lbfgs		Accuracy: 82.61% Precision: 85.05% Recall: 82.00% F1 Score: 83.49%
KNN	Neighbors: 7		Accuracy: 88.86% Precision: 91.00% Recall: 87.92% F1 Score: 89.43%

Complexity assessment

Finally, Table VI shows how the number of features influences the execution time during model testing.

***Table 6**. Time execution evaluation (own authorship).*

Model	Features	Time (s)
SVM	Creast Factor, Kurtosis, RMS	0.053
LR		0.011
KNN		0.0089
SVM	All Features	0.064
LR		0.041
KNN		0.012

Although the addition of three more features has a minimal impact on increasing computational complexity, it is important to consider that as more features are added, this cumulative effect could become significant, affecting not only processing time but also increasing the possibility of overfitting.

Conclusions

This study proposes the use of feature selection algorithms to reduce complexity in classification systems, particularly in fault detection systems for induction motors (IM). Using many features in these systems significantly reduces the feasibility of implementation in real-world environments and, at the same time, increases the tendency for system overfitting. Similarly, this study contributes by proposing a simple pipeline focused on the development of automatic classification systems using ML models, incorporating automatic feature selection algorithms and exhaustive hyperparameter search, achieving classification performance above 97% using SVM. Future work will focus on analyzing and identifying the relevance of frequency domain features and addressing a larger number of faults.

Referencias

[1] Riera-Guasp, M., Antonino-Daviu, J. A., & Capolino, G. A. (2014). Advances in electrical machine, power electronic, and drive condition monitoring and fault detection: State of the art. IEEE Transactions on Industrial Electronics, 62(3), 1746-1759. Doi: 10.1109/TIE.2014.2375853

[2] Wang D (2016) K-nearest neighbors based methods for identification of different gear crack levels under different motor speeds and loads: revisited. Mech Syst Signal Process 70:201–208. Doi: 10.1016/j.ymssp.2015.10.007

[3] Deng W, Yao R, Zhao H, Yang X, Li G (2019) A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm. Soft Comput 23(7):2445–2462.Doi: 10.1007/s00500-017-2940-9

[4] Toma, R. N., Prosvirin, A. E., & Kim, J. M. (2020). Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers. Sensors, 20(7), 1884. Doi: 10.3390/s20071884

[5] Shen Z, Chen X, Zhang X, He Z (2012) A novel intelligent gear fault diagnosis model based on EMD and multi-class TSVM. Measurement 45(1):30–40. Doi: 10.1016/j.measurement.2011.10.008

[6] Glowacz, A. (2019). Fault diagnosis of single-phase induction motor based on acoustic signals. Mechanical Systems and Signal Processing, 117, 65-80. Doi: 10.1016/j.ymssp.2018.07.044

[7] Lizarraga-Morales, R. A., Rodriguez-Donate, C., Cabal-Yepez, E., Lopez-Ramirez, M., Ledesma-Carrillo, L. M., & Ferrucho-Alvarez, E. R. (2017). Novel FPGA-based methodology for early broken rotor bar detection and classification through homogeneity estimation. IEEE Transactions on Instrumentation and Measurement, 66(7), 1760-1769. Doi: 10.1109/TIM.2017.2664520

[8] Calderon-Uribe, U., Lizarraga-Morales, R. A., & Guryev, I. V. (2023). Unbalance Detection in Induction Motors through Vibration Signals Using Texture Features. Applied Sciences, 13(10), 6137. Doi: 10.3390/app13106137

[9] Lee, J. H., Pack, J. H., & Lee, I. S. (2019). Fault diagnosis of induction motor using convolutional neural network. Applied Sciences, 9(15), 2950. Doi: 10.3390/app9152950

[10] Hsueh, Y. M., Ittangihal, V. R., Wu, W. B., Chang, H. C., & Kuo, C. C. (2019). Fault diagnosis system for induction motors by CNN using empirical wavelet transform. Symmetry, 11(10), 1212. Doi: 10.3390/sym11101212.

[11] Agrawal, S., Giri, V. K., & Tiwari, A. N. (2018). Induction motor bearing fault classification using WPT, PCA and DSVM. Journal of Intelligent & Fuzzy Systems, 35(5), 5147-5158.

[12] Patel, R. K., Agrawal, S., & Giri, V. K. (2020). Induction motor bearing fault classification using PCA and ANN. In Computing Algorithms with Applications in Engineering: Proceedings of ICCAEEE 2019 (pp. 269-284). Springer Singapore. Doi: 10.1007/978-981-15-2369-4_23

[13] Saberi, A. N., Sandirasegaram, S., Belahcen, A., Vaimann, T., & Sobra, J. (2020, August). Multi-sensor fault diagnosis of induction motors using random forests and support vector machine. In 2020 International Conference on Electrical Machines (ICEM) (Vol. 1, pp. 1404-1410). IEEE. Doi: 10.1109/ICEM49940.2020.9270689

[14] Kafeel A, Aziz S, Awais M, Khan MA, Afaq K, Idris SA, Alshazly H, Mostafa SM. An Expert System for Rotating Machine Fault Detection Using Vibration Signal Analysis. Sensors. 2021; 21(22):7587. https://doi.org/10.3390/s21227587.

[15] Palácios, R. H. C., Godoy, W. F., Goedtel, A., da Silva, I. N., Moríñigo-Sotelo, D., & Duque-Perez, O. (2017, August). Time domain diagnosis of multiple faults in three phase induction motors using inteligent approaches. In 2017 IEEE 11th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED) (pp. 85-89). IEEE. Doi: 10.1109/DEMPED.2017.8062338

[16] Misra, S., Kumar, S., Sayyad, S., Bongale, A., Jadhav, P., Kotecha, K., ... & Gabralla, L. A. (2022). Fault detection in induction motor using time domain and spectral imaging-based transfer learning approach on vibration data. Sensors, 22(21), 8210. Doi: 10.3390/s22218210

[17] Delgado-Arredondo, P. A., Morinigo-Sotelo, D., Osornio-Rios, R. A., Avina-Cervantes, J. G., Rostro-Gonzalez, H., & de Jesus Romero-Troncoso, R. (2017). Methodology for fault detection in induction motors via sound and vibration signals. Mechanical Systems and Signal Processing, 83, 568-589. Doi: 10.1016/j.ymssp.2016.06.032

[18] Shrivastava, A., & Wadhwani, S. (2013). Development of fault detection system for ball bearing of induction motor using vibration signal. International Journal Of Scientific Research.

[19] Müller, A. C., & Guido, S. (2016). Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".

[20] Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., & Hamprecht, F. A. (2009). A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC bioinformatics, 10, 1-16. Doi: 10.1186/1471-2105-10-213

[21] Anish Kumar, J., Jothi Swaroopan, N. M., & Shanker, N. R. (2022). Induction motor's rotor slot variation measurement using logistic regression. Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije, 63(2), 288-302. Doi: 10.1080/00051144.2022.2031541

[22] Pérez, R., Aguila, A., & Vásquez, C. (2016, May). Classification of the status of the voltage supply in induction motors using support vector machines. In 2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D) (pp. 1-5). IEEE. Doi: 10.1109/TDC.2016.7520012

[23] Samanta, S., Bera, J. N., & Sarkar, G. (2016, January). KNN based fault diagnosis system for induction motor. In 2016 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC) (pp. 304-308). IEEE. Doi: 10.1109/CIEC.2016.7513791

[24] Belete, D. M., & Huchaiah, M. D. (2022). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. International Journal of Computers and Applications, 44(9), 875-886. Doi: 10.1080/1206212X.2021.1974663