

# An Evolving Fuzzy Classifier for Induction Motor Health Condition Monitoring

Peter Luong<sup>1</sup>, Wilson Wang<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, Lakehead University, Thunder Bay, Canada

<sup>2</sup>Department of Mechanical Engineering, Lakehead University, Thunder Bay, Canada

Email: [pluong@lakeheadu.ca](mailto:pluong@lakeheadu.ca), [wwang3@lakeheadu.ca](mailto:wwang3@lakeheadu.ca)

**How to cite this paper:** Luong, P. and Wang, W. (2019) An Evolving Fuzzy Classifier for Induction Motor Health Condition Monitoring. *Intelligent Control and Automation*, 10, 129-141.

<https://doi.org/10.4236/ica.2019.104009>

**Received:** September 12, 2019

**Accepted:** October 14, 2019

**Published:** October 17, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Induction motor (IM) is commonly used in various industrial applications. Reliable online IM health condition monitoring systems are critically needed in industries to improve operational accuracy and safety of the IMs and the machinery. A new evolving algorithm is proposed to provide more decision-making transparency, as well as better classification and processing efficiency. The effectiveness of the developed intelligent classifier is examined by simulation and experimental tests.

## Keywords

Evolving Fuzzy Classifier, Clustering, Automatic Fault Diagnostics, Induction Motors

---

## 1. Introduction

Induction motors (IMs) are widely used in industrial applications such as pumping stations, manufacturing facilities, electric vehicles, etc. IMs have simple construction and high efficiency relative to other types of motors [1]. On the other hand, IMs consume about 40% of the electrical power generated in the world [2]; consequently, there is a strong incentive to ensure that IMs operate efficiently and do not break down unexpectedly. To achieve this goal, research has been conducted over decades to develop technologies and tools to detect IM faults at their incipient stage, prior to reaching more serious levels, so as to prevent performance degradation, malfunction, or even catastrophic failures of the IMs and the related driven facilities. This active monitoring process is known as condition monitoring [3], which serves as a form of predictive maintenance strategy. IM defects include imperfections in support rolling element bearings, stator systems, rotor bars, etc. Condition monitoring should be performed piece

by piece to improve accuracy. This work focuses on broken rotor bar fault diagnostics.

An IM health condition monitoring system consists of three general modules: 1) data acquisition, 2) fault feature extraction, and 3) automatic diagnostic classification. This work focuses on the automatic diagnostic classification.

Pattern classification is a means to classify features obtained by appropriate signal processing techniques into different IM health categories. To automatically perform pattern classification, multiple soft-computing-based methods have been proposed in literature, such as support vector machines [4], principle component analysis [5], k-nearest-neighbors, and artificial neural networks [6]. However, these methods have a disadvantage of being black box processing where its decision-making process is either unclear or the results are difficult to explain, and hence is less suitable for IM fault diagnosis in industrial applications.

An adaptive neuro-fuzzy-inference-system (ANFIS), has a clear decision-making process by using linguistic fuzzy reasoning structure [7]. It uses error back-propagation to automatically adjust fuzzy inference system (FIS) parameters with its training data. However, expertise is required to set up the number of membership functions (MFs) in advance and for training error control. For example, if a user-specified error threshold is too low, it tends to lead to overtraining. Under these conditions, the FIS cannot accurately assess subsequent testing inputs if these inputs have a large divergence from the training data [8]. In addition, if dynamics of the monitored system change dramatically, parameter adjustment alone may not ensure an ANFIS classifier to have a reasonable decision-making accuracy.

To solve the parameter and expertise-related problems of ANFIS-based classifiers, an evolving FIS system can be used for classification, where both system parameters and linguistic fuzzy reasoning structure are evolved iteratively. Such clustering can be achieved with evolutionary algorithms based on measures such as data potential (a measure of data density) [9] [10]. However, the basis of this data potential calculation on a previous datapoint at  $(k - 1)$  may cause the resulting clusters to be less accurate since they do not reflect the most recent datapoint at  $k$ . In addition, aggressive clustering schemes can cause overly-simplified reasoning structures which in turn, generate a deficit of rules that cannot adequately describe all the possible output classes.

Insufficient reasoning structures will reduce the interpretability of the diagnostic results, which in turn decrease the clarity of reasoning behind false or missed alarms.

To tackle the aforementioned problems, an evolving fuzzy (EF) classifier is developed in this work to integrate features from several fault detection techniques for a more reliable IM rotor bar fault diagnosis. It is new in the following aspects: 1) A new updated clustering algorithm is proposed for an evolving fuzzy system; 2) a new strategy is suggested to implement the proposed EF classifier for an IM rotor bar condition monitoring application.

The remainder of this paper is organized as follows. Section 2 discusses the development of the EF classifier. The effectiveness of the proposed technology is verified in Section 3. Some concluding remarks are summarized in Section 4.

## 2. The Proposed EF Classifier

A first-order Takagi-Kang-Sugeno (TSK-1) fuzzy inference architecture is selected as the reasoning platform in this proposed EF classifier, due to its ability in effective data modeling [7]. For such an intelligent system to be functional, two general processes must be completed, a training phase followed by performance evaluation. These processes will be detailed in the proceeding sections.

### 2.1. System Training Overview

The procedure for training an EF system is illustrated in **Figure 1**.

The training procedures can be described in the following steps:

- 1) Cluster inputs with an evolutionary algorithm.
- 2) Project clusters into fuzzy MFs.
- 3) Process inputs with MFs to determine each rule's firing strength.
- 4) Normalize all firing strengths, as a measure of the rules' contributions to the final output.
- 5) Formulate an input matrix by multiplying normalized firing strengths and inputs into a TSK-1 model.
- 6) Update the TSK-1 consequent parameters using available training data pairs.

Assume that an input to the fuzzy classifier has the form  $x(j, k)$ , where  $j = 1, 2, \dots, J$ , and  $J$  is the number of dimensions or attributes of the input;  $k = 1, 2, \dots, K$  represents the instance of normalized datapoint inputs to the classifier system. Each output class (*i.e.*, healthy motor, faulty motor) corresponds to one or more rules in the fuzzy system. As an illustration, consider a fuzzy classifier with only two dimensions or  $j = 1, 2$ . Then the  $i$ th rule,  $R_i$ , can be represented as [7] [8]:

$$R_i : \quad \begin{array}{l} \text{IF } [x(1, k) \text{ is } M_{F_i}(1)] \text{ AND } [x(2, k) \text{ is } M_{F_i}(2)] \\ \text{THEN } y(k) \text{ is } O_i(k) \end{array} \quad (1)$$

where  $O_j(k)$  is the classifier output of rule  $R_j$ ;  $M_{F_i}$  is a fuzzy MF representing a degree of belongingness of an input along the  $j$ th dimension. Details for obtaining the results of the fuzzy classifier will be discussed in the proceeding sections.

### 2.2. Input Clustering

The clustering takes two steps: 1) evolve cluster centers, and 2) compute cluster spreads after evolution. The clustering in the input spaces can be achieved with an evolutionary algorithm based data potential, a measure of data density. Based on the fundamental definition of potential in [9], a recursive computation of the data potential is proposed as:

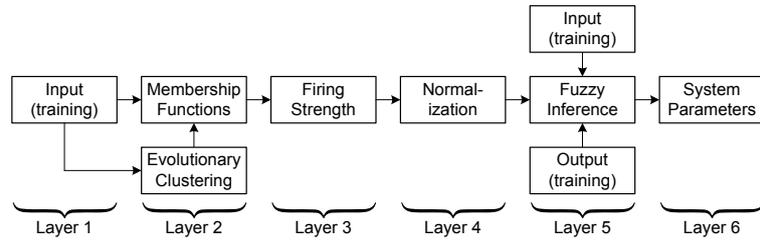


Figure 1. Training with respect to system layers.

$$P(k) = \frac{k}{\left\{ k \left[ 1 + \sum_{j=1}^J [x(j,k)]^2 \right] - 2 \sum_{j=1}^J x(j,k) B(k) + D(k) \right\}}, \quad (2)$$

where,

$$B(k) = \sum_{a=1}^{k-1} x(j,a) + x(j,k), \quad (3)$$

and,

$$D(k) = \sum_{a=1}^{k-1} \sum_{j=1}^J [x(j,a)]^2 + \sum_{j=1}^J [x(j,k)]^2. \quad (4)$$

$B(k)$  and  $D(k)$  are variables representing the relationship between the previous datapoints up to  $k-1$ , and the present data point at  $k$ .

In cluster center identification, from Equation (2), the initial (*i.e.*, the first) cluster center is established at the first datapoint. With subsequent datapoints, the potential of the  $m$ th existing cluster center can be recursively updated by:

$$P_m(k) = \frac{k}{\left\{ k \left[ 1 + \sum_{j=1}^J [x(j,c_m)]^2 \right] - 2 \sum_{j=1}^J x(j,c_m) B(k) + D(k) \right\}}, \quad (5)$$

where  $x(j,c_m)$  is the datapoint corresponding to a cluster center.

A new cluster center is established when the data potential at datapoint  $k$ ,  $P(k)$  is larger than the data potential of any other existing cluster center, or  $\exists m = 1, 2, \dots, P(k) \geq P_m(k)$ . Once the data potential and cluster centers of every datapoint have been computed, the spread is determined with an algorithm [10] expressed as:

$$\sigma_m = \sqrt{\frac{\sum_s \sum_{j=1}^J [x(j,c_m) - x(j,s)]^2}{S_d}}, \quad (6)$$

where  $S_d$  is the data scatter, or the number of datapoints that have the shortest Euclidean distance to a cluster center  $x(j,c_m)$ . From the known training output data, each cluster center is assigned to its respective class, for example, the cluster “1” represents a healthy motor condition, cluster “2” represents a faulted motor condition, etc.

Once the initial cluster centers and spreads have been computed, post-processing is undertaken to generate a single representative cluster per rule.

### 2.3. Membership Function and Firing Strength Formulation

To perform fuzzy reasoning, the inputs are fuzzified with Gaussian MFs, expressed as:

$$M_{F_i}(j) = \exp\left(\frac{-(x(j,k) - x(j, c_m))^2}{2\sigma_m^2}\right) \in (0,1], \quad (7)$$

where  $i = 1, 2, \dots, I$  is the rule associated with a cluster. From Equation (7), the MFs are derived from the cluster centers and spreads. To implement the fuzzy reasoning structure of Equation (1), the firing strength of the  $i$ th rule,  $w_i(k)$  is as follows:

$$w_i(k) = \min\{M_{F_i}(1), M_{F_i}(2), \dots, M_{F_i}(J)\}, \quad (8)$$

where the min operator is a fuzzy  $t$ -norm operator (e.g., AND) [8].

### 2.4. Consequent Parameters for the Evolving Fuzzy Inference System

The output of the classifier system,  $O(k)$ , is computed as:

$$O(k) = \sum_{i=1}^{i=I} \{\bar{w}_i(k) f_i[x(j,k)]\}; \quad \bar{w}_i(k) = w_i(k) / \sum_{i=1}^I w_i(k), \quad (9)$$

where  $\bar{w}_i(k)$  is the normalized firing strength, which represents the contribution of the firing strength of the  $i$ th rule to the output.  $f_j$  is the TSK-1 consequent function of the  $i$ th rule, represented as:

$$f_i[x(j,k)] = C_{i0} + \sum_{j=1}^J x(j,k) C_{ij}, \quad (10)$$

where  $C_{i0}, C_{i1}, \dots, C_{iJ}$  are the consequent parameters of the  $i$ th rule of the EF classifier. These unknown linear consequent parameters can be estimated by training. For example, if  $T(k)$  is the target of the  $k$ th, training data pair, then:

$$T(k) = \sum_{i=1}^{i=I} \{\bar{w}_i(k) f_i[x(j,k)]\}, \quad (11)$$

which can be expanded to,

$$T(k) = \begin{bmatrix} \bar{w}_1(k) & \bar{w}_1 x(1,k) & \dots & \bar{w}_1 x & \dots & \bar{w}_I(k) & \bar{w}_I x(1,k) & \dots & \bar{w}_I x(J,k) \end{bmatrix} \begin{bmatrix} C_{10} & C_{11} & \dots & C_{1J} & \dots & C_{I0} & C_{I1} & \dots & C_{IJ} \end{bmatrix}^T, \quad (12)$$

Equation (12) can be represented in a matrix/vector form:

$$\vec{T} = \mathbf{Z} \vec{C}, \quad (13)$$

where  $\vec{T}$ ,  $\mathbf{Z}$ , and  $\vec{C}$  are the target vector, input matrix, and consequent parameter vector, respectively. Since  $\mathbf{Z}$  is likely a non-square matrix and its inverse may not be computed directly, the singular value decomposition (SVD) will be used to solve for  $\vec{C}$ . The SVD breaks down the  $\mathbf{Z}$  matrix into three components:

$$\mathbf{Z} = \mathbf{U} \mathbf{D} \mathbf{V}^T, \quad (14)$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are the respective left and right singular values, and  $\mathbf{D}$  are the

singular eigenvalues of  $\mathbf{Z}$ . From Equation (14), the Moore-Penrose pseudo-inverse [11] is computed by,

$$\mathbf{Z}^+ = \mathbf{V}\mathbf{D}^+\mathbf{U}^T, \quad (15)$$

where  $D^+$  is the reciprocal of all non-zero elements of  $D$ . With Equation (15), the consequent parameters  $\vec{C}$  can then be estimated by

$$\mathbf{Z}^+\vec{T} = \mathbf{Z}^+\mathbf{Z}\vec{C} \cong \vec{C}. \quad (16)$$

It is noted that the entire training process is one-pass, without the need for a back propagation of error. Upon solving the consequent parameters of Equation (16), it is applied to Equation (9)-(10) for the testing inputs to determine the EF classifier's output.

### 3. Performance Evaluation

The effectiveness of the proposed EF classifier is first assessed with benchmark datasets [12] by simulation. Then it is implemented for IM rotor bar condition monitoring. For comparison, variants of both the proposed and evolving fuzzy classifier proposed in [9] will be evaluated. The variants differ primarily based on the requirements to form clusters: 1) Loose clustering: New clusters are formed when the data potential is larger than any existing cluster center [10];

2) Strict clustering: New clusters are formed when the data potential is larger than all existing cluster centers [9].

The classifiers evaluated will be represented as follows:

- Classifier #1: The proposed EF classifier using the proposed clustering algorithm and a loose clustering variant;
- Classifier #2: A comparison classifier, using the EF clustering algorithm and a strict clustering scheme;
- Classifier #3: A comparison classifier using a classical clustering algorithm and a loose clustering scheme;
- Classifier #4: A comparison classifier using a classical clustering algorithm and a strict clustering scheme.

#### 3.1. Simulation, Iris and Wine Datasets

The Iris and wine datasets have four and thirteen inputs respectively, representing pertinent physical measurements, with more details found in Reference [12]. Both datasets have three outputs, representing the different classes of iris flower or wine types. The datasets used for simulation are summarized in **Table 1**.

**Table 1.** Summary of datasets used for simulation.

Dataset	Number of classifications	Number of attributes	Total Number of datapoints used
Iris [12]	3	4	150
Wine [12]	3	13	144

### 3.1.1. Clustering and Identified Fuzzy Model

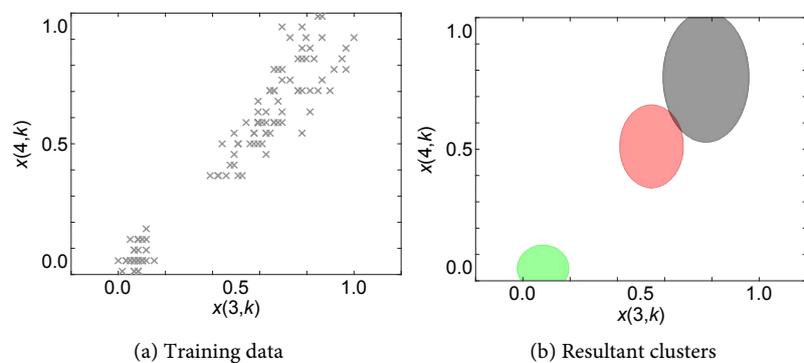
The data and clustering across all attributes in one representative trial with the iris dataset are shown in **Figure 2**. It can be observed that the clusters closely correspond to the original data, showing clear distinctions between the different classes. Based on these clustering results, the resulting recognized fuzzy model is shown in **Figure 3**.

### 3.1.2. Comparison and Discussion

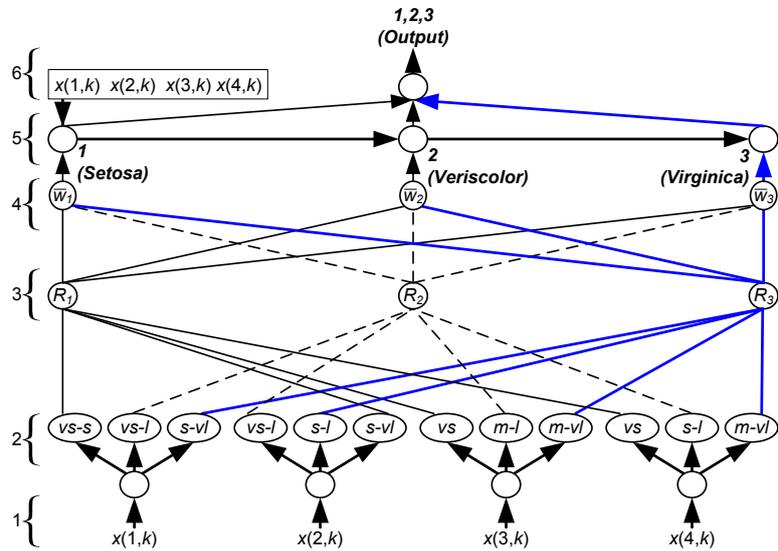
Results across the simulated datasets are shown in **Table 2**.

**Table 2.** Summary of results for the Iris and Wine datasets each with 36 test samples per trial. All results are averaged over 100 trials.

Evolving Fuzzy Classifier Variant	Dataset	Average	
		Iris	Wine
<b>Classifier #1 Proposed (loose clustering)</b>	Accuracy (%)	<b>99.81</b>	<b>94.92</b>
	No. of clusters (average)	<b>2.99</b>	<b>2.95</b>
	Classes represented (%)	<b>99.67</b>	<b>98.33</b>
	Time/Sample (us)	<b>84.95</b>	<b>91.74</b>
Classifier #2 Comparison (strict clustering)	Accuracy (%)	99.97	90.89
	No. of clusters (average)	2.02	1.27
	Classes represented (%)	67.33	42.33
	Time/Sample (us)	78.01	84.78
Classifier #3 Comparison (loose clustering)	Accuracy (%)	99.67	97.25
	No. of clusters (average)	26.10	18.81
	Classes represented (%)	99.33	95.67
	Time/Sample (us)	119.95	127.79
Classifier #4 Comparison (strict clustering)	Accuracy (%)	99.92	91.06
	No. of clusters (average)	5.26	3.66
	Classes represented (%)	66.67	41.00
	Time/Sample (us)	89.02	95.58



**Figure 2.** (a) Training data; (b) Clustering results of the iris dataset, where green, red, and black, correspond to the three flower types.



**Figure 3.** Recognized fuzzy reasoning model of the Iris dataset. All inputs are normalized from [0.0, 1.0]. Approximations of membership functions are represented by “vs” (very small), “s” (small), “m” (medium), “l” (large), and “vl” (very large), denoting input ranges of [0.0, 0.2), (0.2, 0.4), (0.4, 0.6), (0.6, 0.8], (0.8, 1.0], respectively.

From these processing results, the accuracy of the proposed Classifier #1 is comparable to that of Classifiers #2, #3 and #4, but with additional significant advantages. Although Classifier #2 takes less processing times than Classifier #1, due to its the generation of fewer clusters, its resulting clusters do not sufficiently represent the classes, with much lower class representation. As a result, Classifier #2 does not have a fully transparent fuzzy rule base for decision making. Furthermore, the proposed Classifier #1 outperforms Classifiers #3 and #4, in terms of class representation, with a significant advantage of having a more transparent rule base. In addition, Classifier #1 has lower processing times than #3 and #4 due to having fewer clusters.

The proposed EF Classifier #1 demonstrates improvement in terms of processing time and class representation of the clusters, making it more suitable for an IM condition monitoring application as will be discussed in the next section. Hence, the proposed EF classifier has been successfully validated with simulation results.

### 3.2. IM Condition Monitoring for Rotor Bar Fault Diagnosis

#### 3.2.1. Experimental Setup

Five fractional horsepower IMs are tested, with varying defects: healthy #1, healthy #2, 1-bar, 2-bar, 3-bar fault tested under 4 loading conditions: decoupled, low, medium, and heavy load. These defects are representative of a motor with a gradually worsening state prior to reaching a catastrophic failure condition. In addition, the tests are conducted under 2 line frequencies of 50 Hz and 60 Hz, respectively. Data acquisition is performed with split-core current transformers with a custom-built wireless data acquisition system sampling 64,000 samples at

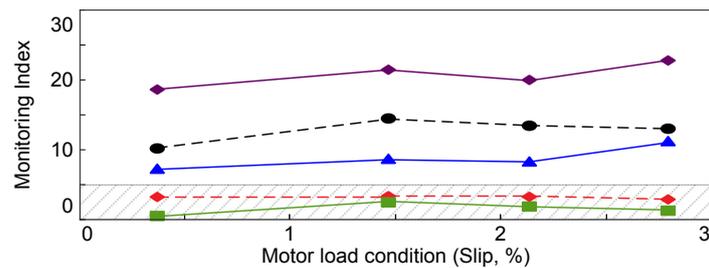
1 kHz. The fault feature extraction is based on the current spectrum analysis, involving sidebands to the fundamental line frequency [13] as well as higher magnetomotive force-based harmonics [14].

To serve as the inputs to the EF classifier, monitoring indices are created on the signal-to-noise ratio of the extracted fault features, where a higher index indicates a higher severity of a faulted condition. An example of such a monitoring index is illustrated in Figure 4. The datasets used for implementation are summarized in Table 3.

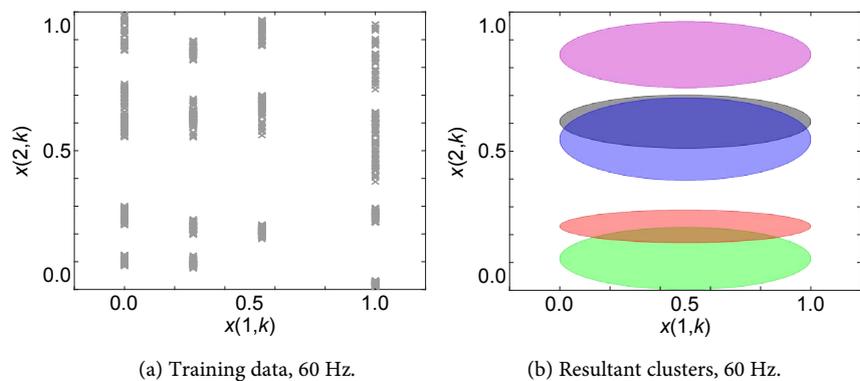
In addition, it can be noted that the influence of noise of this EF classifier for this application has been mitigated by two factors: 1) the long data acquisition period that effectively averages out the influences of non-periodic noise signals, and 2) the clustering algorithm, where an outlier data input would have an insufficient data potential to form the cluster required to influence the fuzzy reasoning of the EF classifier.

### 3.2.2. Clustering and Identified Fuzzy Model

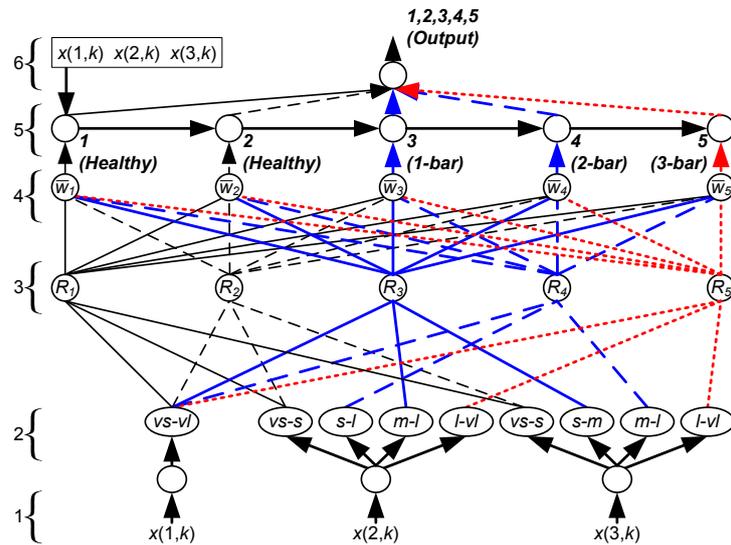
The data and clustering across all attributes in one representative trial with the 60 Hz rotor bar dataset are shown in Figure 5. It can be observed that the clusters closely correspond to the original data, showing clear distinctions between the different classes. Based on these clustering results, the corresponding recognized fuzzy models are shown in Figure 6.



**Figure 4.** IM fault monitoring index examples, where green, red, black, blue, and purple lines correspond to healthy #1, healthy #2, 1-bar, 2-bar, and 3-bar faults, respectively.



**Figure 5.** (a) Training data; (b) Clustering results of 60 Hz rotor bar faulted data, where green, red, black, blue, and purple correspond to healthy #1, healthy #2, 1-bar, 2-bar, and 3-bar faults, respectively.



**Figure 6.** Recognized fuzzy reasoning model of the 60 Hz faulted rotor bar dataset. All inputs are normalized from [0.0, 1.0]. Approximations of membership functions are represented by “vs” (very small), “s” (small), “m” (medium), “l” (large), and “vl” (very large), denoting input ranges of [0.0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1.0], respectively.

**Table 3.** Summary of datasets used for implementation of IM health condition monitoring.

Dataset	Number of classifications	Number of Attributes	Total Number of datapoints used
Rotor Bar, 50 Hz	5	3	1000
Rotor Bar, 60 Hz	5	3	1000

### 3.2.3. Comparison and Discussion

Results across the implementation datasets are summarized in **Table 4**.

For these processing results, it is seen that the accuracy of the proposed EF Classifier #1 performs the best in comparison with all the other three classifiers. This can be attributed to the new clustering algorithm’s improved tracking to changing data. Although Classifier #2 has faster average processing speed than Classifier #1, due to fewer clusters. As a result, Classifier #2 does not have a fully transparent fuzzy rule base for decision making. Likewise, the proposed EF Classifier #1 outperforms Classifiers #3 and #4, in terms of class representation of the clusters and the number of generated clusters; it also has a significant advantage of having a more transparent rule base and faster processing efficiency due to having less clusters.

In summary, the proposed Classifier #1 has the best classification accuracy, improved processing time efficiency and class representation. With better transparent cluster representation, missed and false alarms in a diagnostic application can be investigated upon, where it is possible to track the monitoring indices responsible for an incorrect classifier output.

**Table 4.** Summary of results for the 50 Hz and 60 Hz faulted rotor bar datasets, each with 250 test samples per trial. All results are averaged over 100 trials.

Evolving Fuzzy Classifier Variant	Dataset	Average	
		50 Hz	60 Hz
	Accuracy (%)	<b>99.37</b>	<b>99.45</b>
<b>Classifier #1 Proposed (loose clustering)</b>	No. of clusters (average)	<b>5.00</b>	<b>4.75</b>
	Classes represented (%)	<b>100.00</b>	<b>95.00</b>
	Time/Sample (us)	<b>19.79</b>	<b>16.80</b>
	Accuracy (%)	92.30	88.47
Classifier #2 Comparison (strict clustering)	No. of clusters (average)	2.86	2.00
	Classes represented (%)	57.20	40.00
	Time/Sample (us)	14.20	13.18
	Accuracy (%)	98.90	98.90
Classifier #3 Comparison (loose clustering)	No. of clusters (average)	115.02	106.99
	Classes represented (%)	100.00	87.60
	Time/Sample (us)	148.93	126.73
	Accuracy (%)	94.44	86.25
Classifier #4 Comparison (strict clustering)	No. of clusters (average)	9.13	8.47
	Classes represented (%)	58.40	40.00
	Time/Sample (us)	23.54	21.59
	Accuracy (%)	94.44	86.25

The processing time for each sample across all datasets is in the order of tens of microseconds. This is significantly faster than the generation of the inputs, which require processes such as data acquisition, signal processing, and monitoring index generation. Hence, this classifier demonstrates suitability for use in real industrial condition monitoring applications.

### 3.2.4. Assumptions and Limitations

For this application of the proposed EF classifier to IM condition monitoring the following are assumed: 1) The inputs are representative of the condition of the motor being monitored, 2) During the training process, there are known outputs corresponding to given inputs, so that the consequent parameters of Equation (16) can be estimated.

The developed EF classifier could have the following limitations to be improved: 1) The monitoring indices are based on the signal-to-noise ratio (SNR) of representative features, which could vary with the use of the data acquisition system and motor dynamics. 2) When the developed EF classifier is used in a new application, some test data are needed to update the initial classification architecture.

## 4. Conclusion

An EF classifier has been developed in this work for IM health condition moni-

toring. A new updated evolutionary clustering algorithm is proposed for fuzzy inference reasoning and to formulate a rule structure accounting for multiple clusters belonging to different output classes. Its effectiveness is verified by simulation tests using some benchmark datasets. In addition, the EF classifier is implemented for IM rotor bar fault diagnosis. Test results have shown that the developed EF classifier has improved classification accuracy, processing efficiency, and ability to produce distinct fuzzy clusters/rules to clearly indicate the reasoning process behind every classification. Due to these factors, the developed intelligent IM monitoring system has the potential to be used in real industrial predictive maintenance applications.

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Zeraoulia, M., Benbouzid, M.E.H. and Diallo, D. (2006) Electric Motor Drive Selection Issues for HEV Propulsion Systems: A Comparative Study. *IEEE Transactions on Vehicular Technology*, **55**, 1756-1764. <https://doi.org/10.1109/TVT.2006.878719>
- [2] Saidur, R. (2010) A Review on Electrical Motors Energy Use and Energy Savings. *Renewable and Sustainable Energy Reviews*, **14**, 877-898. <https://doi.org/10.1016/j.rser.2009.10.018>
- [3] Randall, R. B. (2011) *Vibration-Based Condition Monitoring: Industrial, Automotive and Aerospace Applications*. John Wiley & Sons Ltd., Chichester. <https://doi.org/10.1002/9780470977668>
- [4] Matić, D., Kulić, F., Pineda-Sánchez, M. and Kamenko, I. (2012) Support Vector Machine Classifier for Diagnosis in Electrical Machines: Application to Broken Bar. *Expert Systems with Applications*, **39**, 8681-8689. <https://doi.org/10.1016/j.eswa.2012.01.214>
- [5] Esfahani, E.T., Wang, S. and Sundararajan, V. (2014) Multisensor Wireless System for Eccentricity and Bearing Fault Detection in Induction Motors. *IEEE/ASME Transactions on Mechatronics*, **19**, 818-826. <https://doi.org/10.1109/TMECH.2013.2260865>
- [6] Moosavian, A., Ahmadi, H., Tabatabaefar, A. and Khazaei, M. (2013) Comparison of Two Classifiers; K-Nearest Neighbor and Artificial Neural Network, for Fault Diagnosis on a Main Engine Journal-Bearing. *Shock and Vibration*, **20**, 263-272. <https://doi.org/10.1155/2013/360236>
- [7] Jang, J.R. (1993) ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man, and Cybernetics*, **23**, 665-685. <https://doi.org/10.1109/21.256541>
- [8] Karray, F.O. and De Silva, C. (2004) *Soft Computing and Intelligent Systems Design: Theory, Tools, and Applications*. Pearson/Addison Wesley, Harlow, England.
- [9] Angelov, P. (2004) An Approach for Fuzzy Rule-Based Adaptation Using On-Line Clustering. *International Journal of Approximate Reasoning*, **35**, 275-289. <https://doi.org/10.1016/j.ijar.2003.08.006>
- [10] Angelov, P. and Zhou, X. (2008) Evolving Fuzzy-Rule-Based Classifiers from Data Streams. *IEEE Transactions on Fuzzy Systems*, **16**, 1462-1475.

<https://doi.org/10.1109/TFUZZ.2008.925904>

- [11] Hadrien, J. (2018) Deep Learning Book Series 2.9 the Moore Penrose Pseudoinverse. <http://archive.ics.uci.edu/ml>
- [12] Dua, D. and Graff, C. (2019) UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA. <http://archive.ics.uci.edu/ml>
- [13] Filippetti, F., Franceschini, G., Tassoni, C. and Vas, P. (1998) AI Techniques in Induction Machines Diagnosis Including the Speed Ripple Effect. *IEEE Transactions on Industry Applications*, **34**, 98-108. <https://doi.org/10.1109/28.658729>
- [14] Kliman, G.B. and Stein, J. (1992) Methods of Motor Current Signature Analysis. *Electric Machines & Power Systems*, **20**, 463-474. <https://doi.org/10.1080/07313569208909609>