

Article

Detection of Short-Circuit Faults in Induction Motor Winding Turns Using a Neural Network and Its Implementation in FPGA

Luz del Carmen García-Rodríguez ¹, Raúl Santiago-Montero ², Jose de Jesus Rangel-Magdaleno ³,
Francisco Javier Pérez-Pinal ⁴, Rogelio José González-González ⁴, Allan G. S. Sánchez ⁴,
and Alejandro Espinosa-Calderón ^{1,*}

¹ Regional Center for Optimization and Device Development (CRODE), National Technological Institute of Mexico (TecNM), Celaya 38020, Guanajuato, Mexico; carmen.garcia@crodecelaya.edu.mx

² Department of Computer Science, National Technological Institute of Mexico (TecNM), Leon 37290, Guanajuato, Mexico; raul.santiago@itleon.edu.mx

³ Department of Electronics, National Institute of Astrophysics, Optics and Electronics (INAOE), Tonantzintla 72840, Puebla, Mexico; jrangel@inaoe.mx

⁴ Department of Electric and Electronic Engineering, National Technological Institute of Mexico (TecNM), Celaya 38010, Guanajuato, Mexico; francisco.perez@itcelaya.edu.mx (F.J.P.-P.); m1803110@itcelaya.edu.mx (R.J.G.-G.); allan.soriano@itcelaya.edu.mx (A.G.S.S.)

* Correspondence: alejandro.espinosa@crodecelaya.edu.mx

Abstract: Nowadays, induction motors are an essential part of industrial development. Faults due to short-circuit turns within induction motors are “incipient faults”. This type of failure affects engine operation through undesirable vibrations. Such vibrations negatively affect the operation of the system or the products with which said motor is in contact. Early fault detection prevents sudden downtime in the industry that can result in heavy economic losses. The incipient failures these motors can present have been a vast research topic worldwide. Existing methodologies for detecting incipient faults in alternating current motors have the problem that they are implemented at the simulation level, or are invasive, or do not allow in situ measurements, or their digital implementation is complex. This article presents the design and development of a purpose-specific system capable of detecting short-circuit faults in the turns of the induction motor winding without interrupting the motor’s working conditions, allowing online measurements. This system is standalone, portable and allows non-invasive and in situ measurements to obtain phase currents. These data form classified descriptors using a multilayer perceptron neural network. This type of neural network enables agile and efficient digital processing. The developed neural network could classify current faults with an accuracy rate of 93.18%. The neural network was successfully implemented on a low-cost and low-range purpose-specific Field Programmable Gate Array board for online processing, taking advantage of its computing power and real time processing features. The measurement of phase current and the class of fault detected is displayed on a liquid-crystal display screen, allowing the user to take necessary actions before major faults occur.

Keywords: current measurement; multilayer perceptron neural network; non-invasive measurements; online measurements; short-circuit winding turns; incipient faults



Academic Editor: Jie Zhang

Received: 21 December 2024

Revised: 22 February 2025

Accepted: 28 February 2025

Published: 11 March 2025

Citation: García-Rodríguez, L.d.C.; Santiago-Montero, R.; Rangel-Magdaleno, J.d.J.; Pérez-Pinal, F.J.; González-González, R.J.; Sánchez, A.G.S.; Espinosa-Calderón, A. Detection of Short-Circuit Faults in Induction Motor Winding Turns Using a Neural Network and Its Implementation in FPGA. *Processes* **2025**, *13*, 815. <https://doi.org/10.3390/pr13030815>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Induction machines (IMs) are among the most widely used power devices globally due to their robustness, low cost, ease of control, and high efficiency. Ensuring their continuous and safe operation is crucial for maintaining substantial economic benefits and reducing

downtime. However, these machines are subject to various stresses contributing to their degradation and aging. These stresses are categorized under the TEAM (Thermal, Electrical, Ambient, and Mechanical) model [1].

An induction motor comprises multiple mechanical and electrical components, including the motor frame, stator windings, rotor cage, rolling bearings, fan, and rotor shaft. Despite their inherently robust design, induction motors are exposed to external stressors [2]. Short-circuit faults between windings are the most prevalent among the different types of motor faults. Such faults originate from the deterioration of insulating materials, ultimately leading to insulation breakdown between windings and incipient inter-turn short-circuit faults. Early detection of inter-turn short-circuits is essential to prevent further motor damage and mitigate economic losses [3].

Empirical studies [2,4–7] indicate that bearing faults account for approximately 40% of all failures in induction motors, with some research suggesting an even higher rate for small motors. Additionally, studies conducted by the Institute of Electrical and Electronics Engineers (IEEE) and the Electric Power Research Institute (EPRI) report that stator faults constitute 36% and 28% of IM failures, respectively. Furthermore, rotor faults account for 8–10% of induction motor failures [2,4]. Given these statistics, the early detection of such incipient faults has been a significant focus of research to prevent unexpected production downtimes and economic losses.

Over the past decades, various fault detection techniques have been developed, including vibration analysis, temperature monitoring, magnetic field analysis, current analysis, acoustic noise assessment, and torque profiling [2,6,7]. Among these methods, current analysis is widely adopted due to its effectiveness in fault detection, which this study utilizes. Artificial intelligence (AI) techniques have increasingly been integrated into these methodologies in recent years to enhance detection accuracy [7–9].

Several AI-based techniques have been extensively applied in IM fault diagnosis, including artificial neural networks (ANNs), support vector machines (SVM), genetic algorithms (GAs), fuzzy logic, adaptive neuro-fuzzy inference systems (ANFIS), and hybrid models. ANN, in particular, is a powerful data processing system capable of learning input–output relationships from training data. ANN comprises processing elements (neurons) interconnected through weighted links that enable information sharing [7,10].

Artificial neural networks (ANNs) are widely recognized for their capacity to learn, approximate, and analyze complex problems. Single-layer perceptron (SLP) networks, consisting of only input and output layers, represent the simplest form of ANNs but are inadequate for handling nonlinearly separable patterns. To overcome these limitations, researchers have developed multilayer perceptron (MLP) networks, the most frequently employed ANN model in the literature. MLPs offer advantages such as high learning potential, noise robustness, nonlinearity adaptability, parallel processing capability, fault tolerance, and excellent generalization capacity [11].

Recent studies on IM fault detection employ diverse techniques, ranging from mathematical analyses to AI-based methods. Researchers have explored various ANN-based approaches for fault identification. For instance, Khanjani and Ezoji [12] investigated the classification of thermal patterns of electrical faults in three-phase IMs as a non-invasive and non-destructive diagnostic approach, employing SVM due to its strong generalization capability. Osornio et al. [13] proposed a methodology for processing electric current signals using the short-time Fourier transform (STFT). The diagnostic system was developed using digital hardware implementations, integrating an FPGA with a microprocessor (PYNQ Z2, Xilinx, Inc., San Jose, CA, USA) for efficient digital signal and image processing. Their method demonstrated high classification accuracy for various IM faults, including healthy

(HLT), misalignment (MAMT), unbalance (UNB), damaged bearing (BDF), and broken rotor bar (BRB) conditions.

Aib et al. [14] examined the implementation of fuzzy logic on Field Programmable Gate Array (FPGA) (Virtex-4, Xilinx, Inc., San Jose, CA, USA) based programmable logic circuits for diagnosing phase unbalance and phase failure in asynchronous machines. Khoualdia et al. [15] developed a structured and optimized MLP-ANN model for online IM condition monitoring. Their ANN utilized eight indicators representing different motor states (healthy, broken rotor bars, bearing fault, and misalignment), while the output layer employed a coded matrix. The model demonstrated high accuracy. Ewert et al. [16] presented a low-cost system for online IM-bearing condition monitoring and fault diagnosis using neural networks (NNs) in radial vibration measurements.

Bazan et al. [17] proposed an approach for detecting stator winding short-circuit faults using phase-shifted stator current measurements. Their MLP-based online detection system was implemented in Matlab using a data acquisition card (USB-6221, National Instruments, Inc., Austin, TX, USA). Additionally, Bazan et al. [18] developed a two-stage diagnosis method based on mutual information measurements of current signals, principal component analysis, and intelligent systems, achieving high classification accuracies.

Li et al. [19] introduced an improved selective set deep learning method with beetle antenna search for enhancing bearing fault diagnostic performance. Their findings suggest that this approach outperforms conventional base models and ensemble learning methods in accuracy and robustness.

In summary, winding failures represent IMs' second most common incipient faults. Existing fault detection methodologies for AC motors exhibit certain limitations, such as being primarily implemented in simulations, requiring invasive sensors, lacking in situ measurement capabilities, or having complex digital implementations. Few studies focus on developing specialized systems for real-time processing, online and in situ detection of short-circuit faults in IM windings. Most available solutions rely on personal computers, data acquisition cards (DAQ), and invasive sensors, highlighting the need for a dedicated system capable of real-time, efficient fault detection.

Reconfigurable computing, mainly through Field-Programmable Gate Arrays (FPGAs), offers a promising solution. FPGAs balance high-performance hardware and software flexibility, allowing post-manufacturing reprogramming to accommodate application-specific requirements. Compared to Application-Specific Integrated Circuits (ASICs), FPGAs offer cost efficiency, enhanced performance, and reduced power consumption [20]. Additionally, advancements in FPGA-based deep learning accelerators have demonstrated substantial performance improvements in computational efficiency [21].

This study aims to develop a non-invasive, purpose-specific system for real-time and online detection of short-circuit faults in induction motors. The proposed system incorporates current descriptors processed through an MLP neural network for fault classification. The primary advantages of this system include:

- High-speed, energy-efficient processing tailored to purpose-specific diagnostic tasks.
- Simple and reconfigurable digital implementation.
- FPGA-based architecture, enabling hardware upgrades and reuse while maintaining low cost and power consumption.
- Portability and practicality for standalone, online, in situ measurements using non-invasive sensors.
- Real-time acquisition, storage, display, and transmission of current measurement data.
- Utilization of MLP-ANNs, which offer superior learning potential, noise robustness, nonlinearity adaptation, and high fault tolerance.

2. Materials and Methods

2.1. Materials

The present system is described in VHDL code and implemented in a low-cost and low-range FPGA board (Spartan 3A, Xilinx, Inc., San José, CA, USA). This FPGA was selected to demonstrate the minimal computational resources required by the proposed system.

The acquisition system uses all the materials reported in González et al. [22].

Matlab software (R2021b, Mathworks, Natick, MA, USA) was used for NN design, calculation of training patterns, training the multilayer perceptron neural network with back-propagation learning, and calculating polynomial coefficients for FPGA implementation.

Setup for Data Collection

To evaluate the performance of the developed acquisition system, laboratory tests were carried out using the test bench with a 1hp three-phase induction motor (RGZE, Siemens S.A. de C.V., Querétaro, México) with a squirrel cage rotor (Table 1). The tests involved inducing short-circuit faults in the stator winding turns and the variation of the stator load conditions with an electromagnetic brake. The motor has modifications in one of the winding phases that allow us to cause faults intentionally from the outside (Figure 1a); thus, we can extract the data for different operating conditions. An electromagnetic brake added load to the motor through a magnetic field generated by solenoids, which induce eddy currents on an aluminum disc fixed to the motor shaft (Figure 1b). Likewise, we used an rpm meter (revolutions per minute) to measure the engine's speed at all times, thus being able to distinguish the different load conditions added by the electromagnetic brake.

Table 1. Siemens motor technical data sheet.

Technical Data	Value
Voltage	220 V
Current	3.2 A
Nominal Efficiency	82.5%
min^{-1}	1745 rpm
Power	0.746 kW

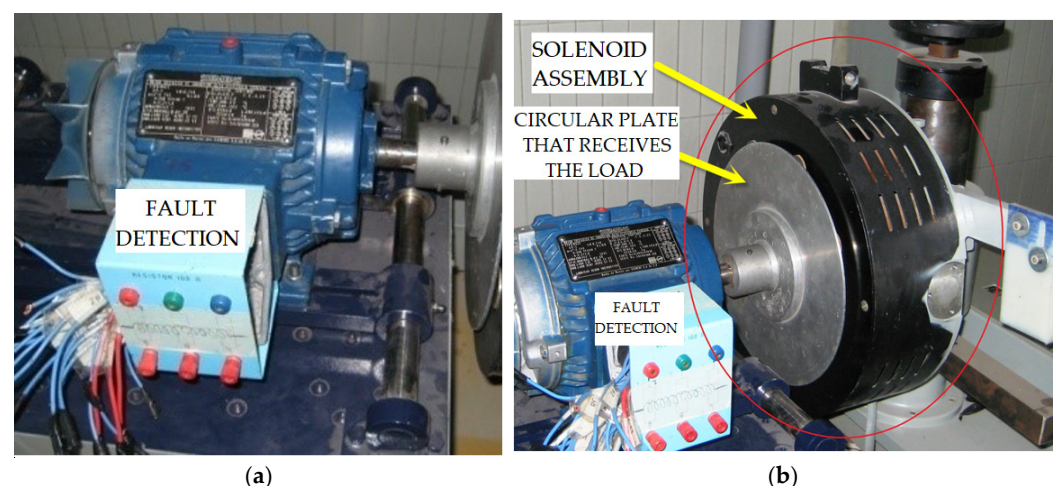


Figure 1. Experimental setup with a Siemens motor. (a) Motor with modification in one of the winding phases, (b) Motor with electromagnetic brake (red circle).

- The motor was monitored under different load and fault conditions during the data extraction process. Six different load states were forced with the electromagnetic brake at 1745 rpm (engine at full load), 1755 rpm, 1765 rpm, 1775 rpm, 1785 rpm, and

1795 rpm (engine at no load). For each of these load states, the motor operating conditions were: Without fault (WoF);

- One turn in short-circuit (1TShC);
- Two turns in short-circuit (2TShC);
- Three turns in short-circuit (3TShC);
- Four turns in short-circuit (4TShC);
- Five turns in short-circuit (5TShC).

Therefore, a total of 6 classes of failure were defined, and for each failure, six load states were used, resulting in 36 different datasets (Table 2). For each case, a 15 s monitoring is carried out at an acquisition frequency of 10 kHz, thus obtaining 150,000 specific values. The current measurement for these datasets was performed in a single winding phase (modified winding).

Table 2. The 36 datasets defined by six fault classes and six load states.

Loading Conditions	Induced Failures					
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5
1745 rpm	WoF	1TShC	2TShC	3TShC	4TShC	5TShC
1755 rpm	WoF	1TShC	2TShC	3TShC	4TShC	5TShC
1765 rpm	WoF	1TShC	2TShC	3TShC	4TShC	5TShC
1775 rpm	WoF	1TShC	2TShC	3TShC	4TShC	5TShC
1785 rpm	WoF	1TShC	2TShC	3TShC	4TShC	5TShC
1795 rpm	WoF	1TShC	2TShC	3TShC	4TShC	5TShC

2.2. Methodology

1. **Motor Current Acquisition System:** The first stage of the proposed methodology consists of acquiring the motor current signal to be used as input for the fault detection system. A non-invasive current sensor (SCT013, Yaohuadechang Electronic Co., Beijing, China) is employed to measure the motor's electrical consumption. The acquired signals are conditioned using analog-to-digital conversion techniques to facilitate further processing [22]. The sampling frequency is selected based on the fundamental frequency of the motor and the Nyquist criterion to ensure sufficient resolution for fault classification.
2. **Neural Network Design:** The development of a neural network for fault detection is divided into the following sub-stages:
 - a. **Calculation of Training Patterns:** Training patterns are generated based on controlled experiments in which different motor operating conditions (healthy and faulty states) are simulated. The feature extraction technique of time-domain statistical analysis is applied to define discriminative descriptors. These descriptors serve as input patterns for training the neural network.
 - b. **Training of the multilayer perceptron neural network with backpropagation learning:** A multilayer perceptron (MLP) neural network is designed and trained using the backpropagation algorithm. The training process involves adjusting the network's weights and biases to minimize classification error. The activation function is chosen based on hardware implementation feasibility, with a polynomial approximation of the sigmoid function being considered for FPGA deployment. The network's performance is evaluated using the standard validation technique of k-fold cross-validation to ensure generalization.
3. **Implementation of the neural network on FPGA:** Once the neural network is trained, it is implemented on an FPGA platform. The network's mathematical operations,

including matrix multiplications and activation function computations, are synthesized using VHDL. Fixed-point arithmetic is employed to optimize hardware resource usage and maintain precision. The FPGA's parallel processing capability is leveraged to achieve real-time fault classification.

4. Data collection: Experimental data collection is performed under various operating conditions, including healthy and faulty motor states. The acquired signals are analyzed to validate their consistency and reliability.
5. Neural network design results: The trained neural network is evaluated based on classification accuracy.
6. FPGA Implementation results: The FPGA-based implementation is assessed in terms of resource utilization and processing time. The processing speed of the implemented system is measured to confirm its capability for on-site fault detection. Comparative analyses between software-based and FPGA-based implementations are conducted to highlight improvements in execution time and system efficiency.

In Section 4, the developed system is compared with existing approaches to determine its advantages. The feasibility of deploying the FPGA-based solution in industrial environments is also discussed.

3. Results

3.1. Motor Current Acquisition System

The neural network requires training. Therefore, generating a database of the motor current in the different load and fault states is necessary.

The current acquisition system proposed by González et al. [22] was used to obtain the current values. The acquired current signal passes through a conditioning circuit, where it is filtered and rectified. The sampling frequency for the current acquisition was set to 10 kHz.

Subsequently, a finite state machine was implemented on an FPGA card to manage data storage, data processing, visualization, and transmission. The state machine algorithm first acquires the current values and stores them in memory until 150,000 pieces of data are obtained. Then, a slide switch determines the status of data transmission. If the decision is not to transmit them, the data are overwritten by new values. If you decide to transmit the data, it is sent through a serial port to a PC using the RS-232 [23] protocol configured at a transmission speed of 19,200 bauds. Finally, the user can monitor the measured effective current and type of fault calculated in situ, both processed in real time through the present standalone system, and displayed using an LCD screen (Figure 2).

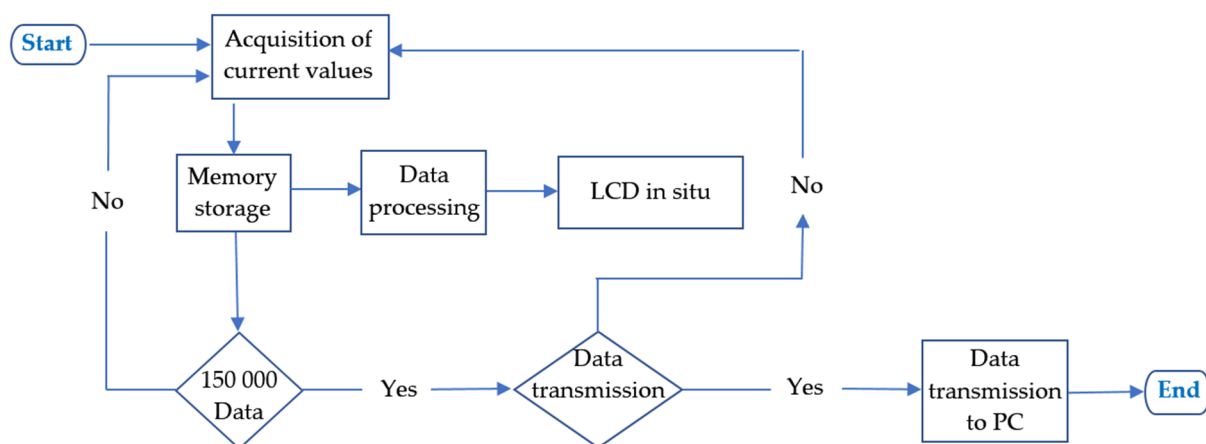


Figure 2. A block diagram of the state machine implemented on an FPGA card to control data storage, data processing, visualization, and transmission.

3.2. Neural Network Design

Neural networks have been widely used to detect faults in induction motors, and the multilayer perceptron has stood out for its good results in fault classification [2,15,24–26]. This work used a multilayer perceptron with backpropagation learning. Because the multilayer perceptron works under supervised learning, the input patterns and their corresponding expected outputs must be defined for the correct network training. To shape the input patterns (Equation (1)), a descriptor model was developed and applied to the acquired stream data. This consists of performing the square of the averaged sum of r number of samples of the half-cycles of the acquired signal (Equation (2)), forming the training patterns every six half-cycles (Figure 3).

$$P_n = [C_{6n+1}, C_{6n+2}, C_{6n+3}, C_{6n+4}, C_{6n+5}, C_{6n+6}] \quad (1)$$

$$C_j = \left(\frac{1}{r} \sum_{k=n}^{k=n+r} x_k \right)^2 \quad (2)$$

where P is the training pattern, n is the number of patterns, C_j is the ridge descriptor, and x represents the half-cycle of the acquired current signal.

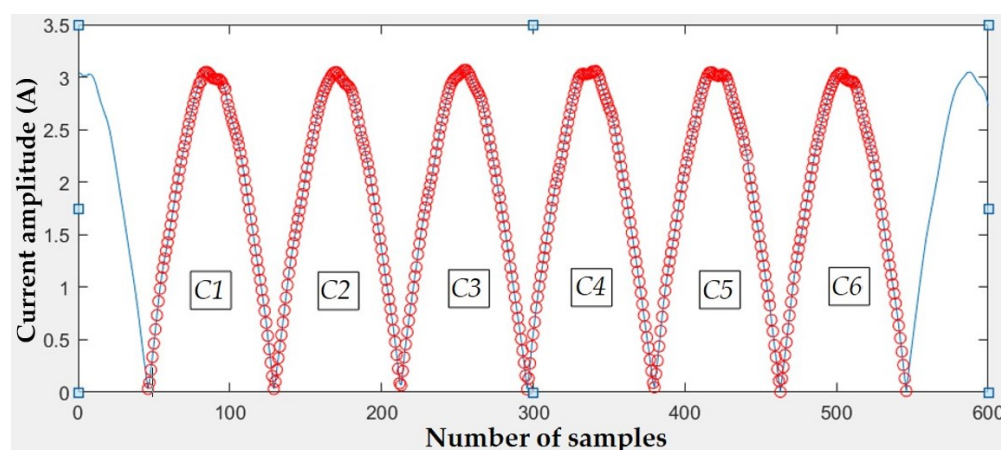


Figure 3. Current half-cycles of the acquired signal $P = [C1, C2, C3, C4, C5, C6]$. The blue line represents the acquired stream data, and the red circles indicate the descriptor models calculated from Equation (1).

After applying the descriptor model to the dataset, 1788 patterns were formed for each failure, thus obtaining the input patterns to the six-dimensional neural network. Therefore, the neural network comprises six neurons in the input layer and six in the output layer, one for each type of failure. So, the architecture is a fully connected network with six nodes at the input, with nodes inside the intermediate layers, and with six nodes at the outputs. The procedure for calculating the number of neurons in the intermediate layers follows the expression suggested in Matlab (R2021b, Mathworks, Natick, MA, USA) (Equation (3)):

$$(\text{number of input neurons} + \text{number of output neurons})^{0.5} + (1 \dots n) \quad (3)$$

where, using trial and error, we find the number from 1 to n that will give the number of neurons in that hidden layer, determined by the minimum mean square error. Two hidden layers were used, the first with 15 neurons and the second with 22 neurons.

The hyperbolic tangent activated all neurons in the multilayer perceptron's training stage. In the network's implementation, the activation function of the output layer was changed to a linear function, where each output neuron represents one kind of failure. The most active output neuron indicates the type of failure the motor presents.

The network used for this proposal is second-generation and has the same convergence principle as the deep learning framed networks, such as backpropagation networks. However, these networks require much more significant amounts of memory and processing time than the network used in this work. In addition, convolutional networks focus on images, and deep networks focus on massive data inputs, which is not the case in this proposal, where we provide only six inputs.

Calculation of Training Patterns

Always remember that only the minimum peaks of a half-cycle are needed to sum up the current values. Therefore, the current signal is multiplied by a negative integer (-1) to generate these minimum peaks, thus obtaining the inverted signal (Figure 4).

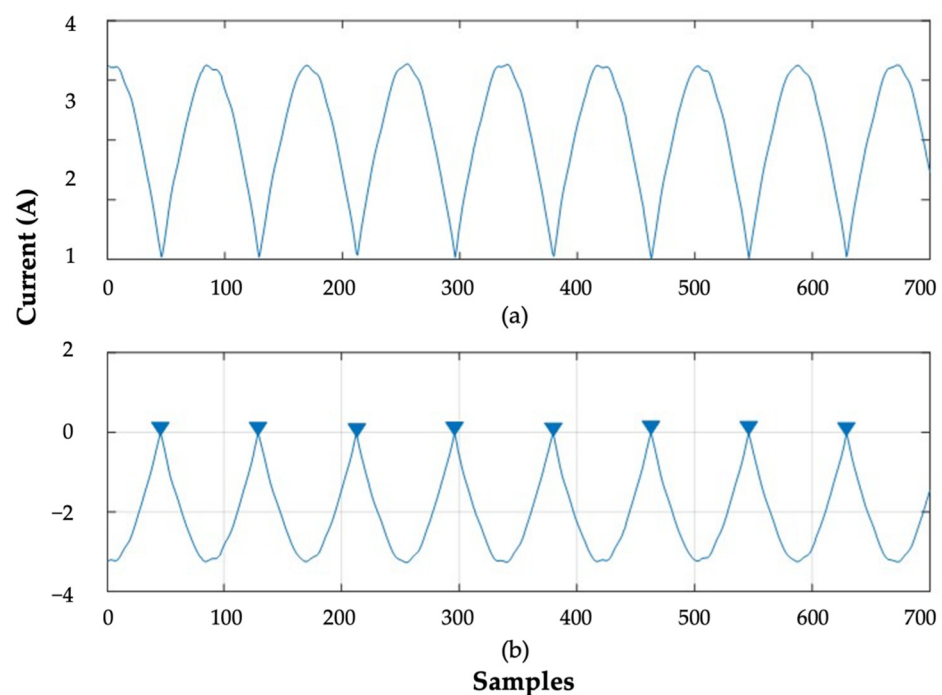


Figure 4. Current signal of the data obtained in the acquisition stage. (a) Data stream signal obtained in the acquisition stage; (b) Inverted current signal, where the blue triangles indicate the moment when the current value is zero.

Once the inverted current signal is obtained, the training patterns are calculated. To calculate the training patterns and describe the model (Equation (2)), the first step is to locate the maximum current peaks of the signal half-cycles obtained during the acquisition stage. This can be achieved using the command “[PKS, LOCS] = findpeaks (data vector)” available in the Matlab tool. This command helps to find the maximum peaks (PKS) and their locations (LOCS) in a data vector.

Once the minimum current peaks are located, the generated vector is analyzed, and Equation (2) is subsequently implemented. The code to perform this calculation in each half-cycle is achieved with a double *for* cycle, one for the complete vector, and the other for each half-cycle within the vector. This code calculates the vector of descriptors, where the training patterns are formed for every six values. This process is performed for each of the 36 datasets. Thus, the 1788 patterns were formed by class and later divided into 70% for training and 30% for testing.

Once the number of neurons in the hidden layers is defined, as well as the training and testing patterns, with the support of the propagation algorithm, all layers’ weight and threshold vectors are initialized randomly. Subsequently, the algorithm allows controlling

the epochs and each pattern with which the network will be trained. Within this last cycle are the three most important steps of the backpropagation algorithm: forward propagation of the input stimulation, backward propagation of the error in each layer, and finally, updating the synaptic weights of each layer. These cycles are repeated up to the number of epochs the user chooses. The root mean square error is plotted during training. Once the epochs have ended, it is verified if the network has been trained correctly. The test patterns are presented, and the percentage of network successes is calculated. As mentioned in Section 3.2, there is a neuron for each type of failure, so the most active neuron indicates the type of failure presented in the motor. According to the above information, if neuron 1 is the most active, there are no faults; on the contrary, if the most active neuron is between 2 and 6, it represents that 1 to 5 loops are shorted, respectively.

Once the neural network is trained, a matrix with fixed weights is generated. Therefore, the accuracy values are subject to statistical evaluation. So, this accuracy was obtained by employing a statistical K-fold test. K-Fold cross-validation is a widely used technique in machine learning and statistics to evaluate the performance of predictive models. Its application is especially valuable in contexts where the amount of data is limited, as it allows for a more robust and reliable assessment of model performance. Through an iterative process, each observation in the dataset is used in both the training and testing phases, thus providing a comprehensive assessment of the model's generalization ability [27]. Compared to traditional validation methods, K-Fold cross-validation offers several advantages:

- More accurate performance estimation: By averaging the results obtained in multiple iterations, variability in the evaluation of the model is reduced, providing a more reliable measure of its performance.
- Efficient use of data: Each dataset instance is used for training and testing, which is particularly useful in small datasets.
- Reduction in bias and variance: A more stable and representative evaluation is obtained by reducing the dependence on the initial partition of the data.
- Improved generalization capacity: It more accurately evaluates the model's performance on unseen data, reducing the risk of overfitting.
- Versatility: It can be applied to various models, including regression, classification, and clustering algorithms, making it a widely adopted strategy in data science.

Given its ability to provide more reliable evaluations and applicability to different models, K-Fold cross-validation is considered an essential technique in developing and optimizing AI models.

3.3. Implementation of the Neural Network on FPGA

FPGAs have advantages such as high-speed processing, high reconfigurability, and parallel acquisition, which ensures that all variables of interest are taken simultaneously. They are reprogrammable devices that create embedded systems, code portability, better architecture, sufficient input–output ports required for the application, and reconfigurable architecture. They are also a purpose-specific system. Due to these advantages, FPGAs are currently used in applications where high-demand computational resources and fast data acquisition are necessary, thus gaining great popularity [28].

The implementation of the neural network in the FPGA is divided into four stages:

- Non-invasive and on-line current acquisition;
- In situ and standalone pattern creation;
- In situ and standalone pattern processing by the network;
- In situ display of the motor status on the LCD screen.

In the first stage, non-invasive and on-line current acquisition, a finite state machine controls the process every moment. As the online processing is performed pattern by

pattern, 700 current samples are taken and stored in RAM in fixed-point binary vectors using an unsigned $Q_{12,10}$ format (12 bits for integers and 10 bits to represent the fractional part). In the second stage, in situ and standalone pattern creation, the current descriptor must be applied to the previously acquired signal. A moving window scrolls through the acquired data, comparing the magnitude of all samples in the window and allowing peaks to be found from a threshold. The address of the minimum value is stored in RAM. This process is repeated until the seven peaks corresponding to the six patterns are found. The moving window depends on the acquisition frequency of the system, as well as the size of the samples that are taken between peaks. It is essential to have an adequate choice regarding the window size, since if larger or smaller windows are considered, the minimums will be incorrect (Figure 5, red windows). In the case of this project, 84 samples were taken based on moving window size (yellow window), as shown in Figure 5.

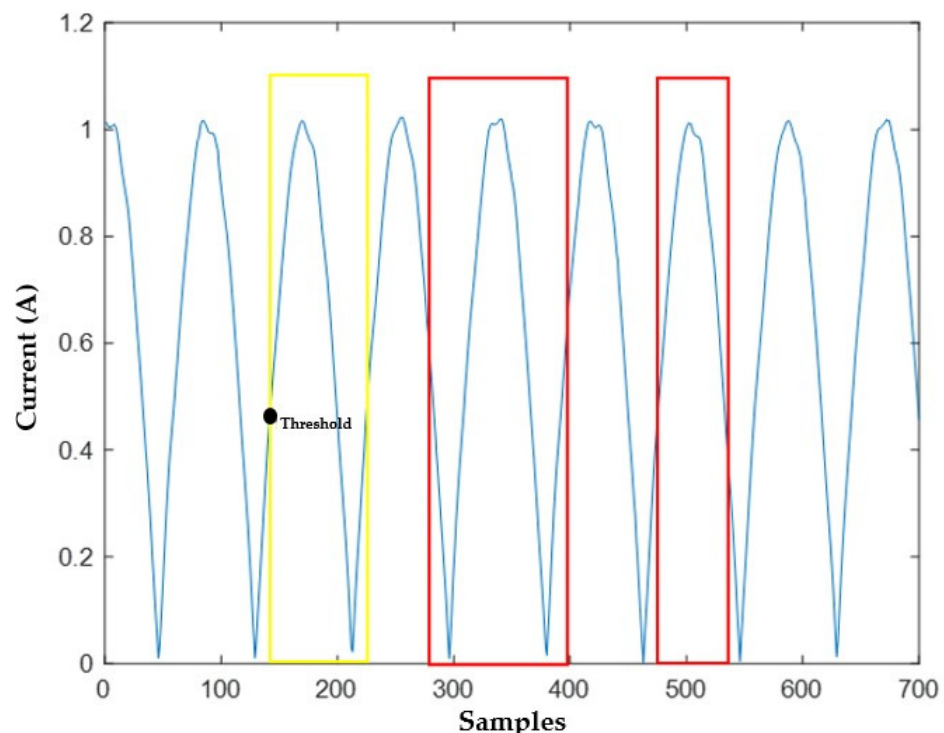


Figure 5. Moving the window to find the minimum current peaks.

Once all the directions of the peak values have been found, we continue implementing Equation (2). The resulting pattern is saved in RAM. The logic inputs and flags of the finite state machine designed in the FPGA control this process. The six pattern attributes are calculated in fixed-point binary vectors using an unsigned $Q_{32,24}$ format (32 bits for the integer section and 24 bits for the fractional part).

The third stage, in situ and standalone pattern processing by the network, evaluates the pattern generated by the neural network. As mentioned above, the network comprises six neurons in the input layer, fifteen in the first hidden layer, twenty-two in the second hidden layer, and six in the output layer. The activation function used in the hidden layer neurons is a tangential sigmoid, while it is a linear function for the output layer. The architecture of the network designed in FPGA is simplified in the diagram in Figure 6. The synaptic weights and polarizations obtained in the training stage using the Matlab tool are stored in ROM memories. On the other hand, the calculation of each of the neurons in the hidden layer is stored in RAM. The calculation of the output layer neurons is recorded and compared, thus determining the number of the most active neurons and, consequently,

the fault classification. A control machine (CM) is responsible for propagating the pattern through the different blocks of the Arithmetic Logic Unit (ALU).

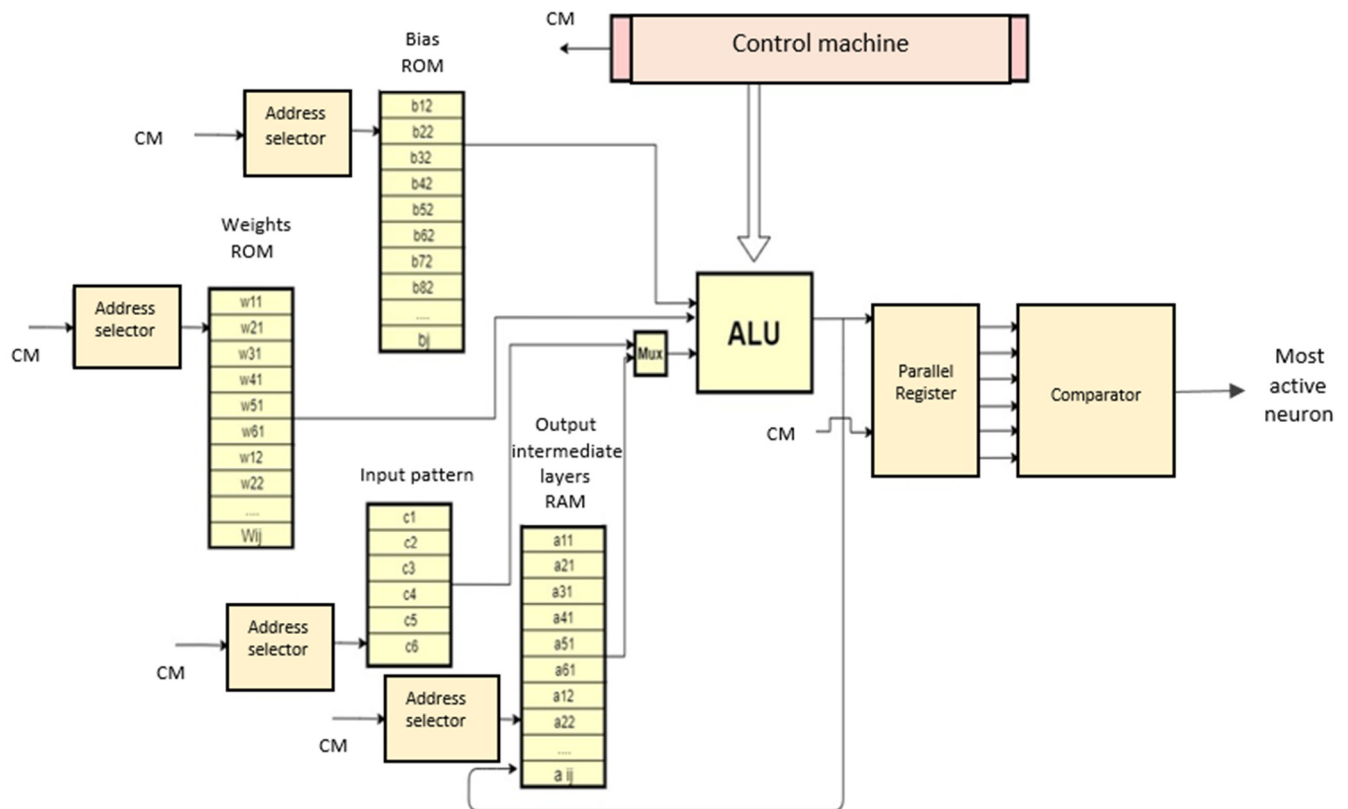


Figure 6. Block diagram of neural network architecture.

The ALU handles the mathematical calculations and evaluates the activation function (Figure 7). The ALU consists of a multiplier (weights * inputs), an accumulator for adding all weights by the inputs arriving at a neuron, and an adder for the bias. Finally, it is evaluated in the hyperbolic tangent activation function.

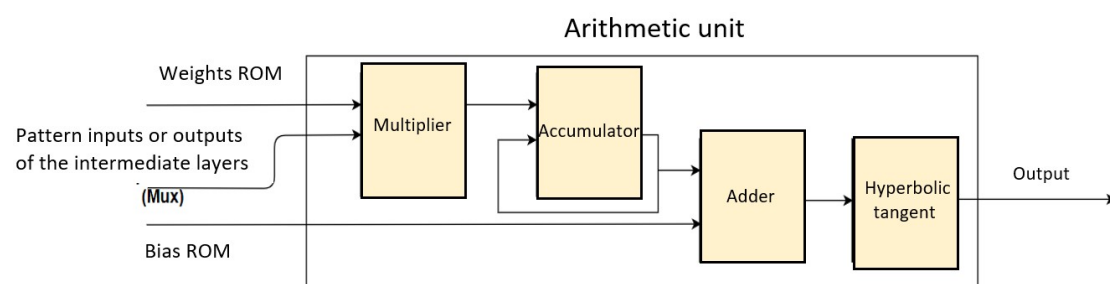


Figure 7. Arithmetic unit.

The digital resource requirements for synthesizing the sigmoid function on an FPGA depend on the chosen mathematical approach. The direct hardware implementation of the tangential sigmoid activation function is impractical due to its inherent division and exponential operations, which demand excessive logic resources and exhibit slow convergence, thereby increasing processing time. To address this limitation, this study employs a polynomial approximation method, $H(x)$, to represent the sigmoid function, as described in Equation (4):

$$H(x) = \begin{cases} -1 & x > 4.4 \\ P_{n1} & -0 \leq x \leq 4.4 \\ P_{n2} & -4.4 \leq x \leq 0000 \\ -1 & x < -4.4 \end{cases} \quad (4)$$

In this approach, P_{n1} and P_{n2} correspond to the polynomial approximations for the positive and negative sections of the function, respectively. Table 3 presents the coefficients of the eighth-order polynomial, obtained using Matlab's *polyfit* function. These coefficients were subsequently transformed and approximated into fixed-point binary vectors using a signed $Q_{32,31}$ format (1 bit for the sign, 31 bits for the integer part, and 31 bits for the fractional section) (Table 4). Such binary vectors are essential for the FPGA's VHDL description and implementation.

Table 3. Coefficients of the eighth-order polynomials P_{n1} and P_{n2} , which were obtained using Matlab's *polyfit* function. These polynomials represent the approximations for the positive and negative sections of the sigmoid function, respectively.

Coefficients	P_{n1}	P_{n2}
a_0	$1.347655395821725 \times 10^{-4}$	$-1.347655395826975 \times 10^{-4}$
a_1	-0.003129272856510	-0.003129272856520
a_2	0.030329383461283	-0.030329383461362
a_3	-0.157298007121535	-0.157298007121864
a_4	0.457402594484345	-0.457402594485108
a_5	-0.665273061840729	-0.665273061841719
a_6	0.113999932754436	-0.113999932755104
a_7	0.984821740623635	0.984821740623438
a_8	$4.197961384300758 \times 10^{-4}$	$-4.197961384465547 \times 10^{-4}$

Table 4. Base 10 approximation of the eighth-order coefficients of polynomials P_{n1} and P_{n2} adjusted into fixed-point binary vectors using a signed $Q_{32,31}$ format.

Coefficients	P_{n1}	P_{n2}
a_0	$1.347656361758709 \times 10^{-4}$	$-1.347656361758709 \times 10^{-4}$
a_1	-0.003129272721708	-0.003129272721708
a_2	0.030329383444041	-0.030329383444041
a_3	-0.157298007048666	-0.157298007048666
a_4	0.457402594387531	-0.457402594387531
a_5	-0.665273061953485	-0.665273061953485
a_6	0.113999932538718	-0.113999932538718
a_7	0.984821740537882	0.984821740537882
a_8	$4.197959788143635 \times 10^{-4}$	$-4.197959788143635 \times 10^{-4}$

Equation (5) shows the resulting polynomial of order 8; however, algebraic transformations allow it to be simplified into Equation (6). This simplification allows basic operations such as addition and multiplication, thus avoiding raising the polynomial to the eighth power, which allows designing a less complex FPGA implementation.

$$a_0 x^8 + a_1 x^7 + a_2 x^6 + a_3 x^5 + a_4 x^4 + a_5 x^3 + a_6 x^2 + a_7 x + a_8 \quad (5)$$

$$((((((((a_0 + 0) x + a_1) x + a_2) x + a_3) x + a_4) x + a_5) x + a_6) x + a_7) x + a_8 \quad (6)$$

Figure 8 shows the architecture of the hyperbolic tangent function. The execution of the architecture consists of the following. When an input value ("x") is presented, a comparator determines in which interval of the function (Equation (4)) this value is located.

Based on this, the polynomial evaluation is not required if “ x ” is more significant than 4.4 or less than -4.4 . A “1” or “ -1 ” is obtained in the output register according to the detected condition. On the other hand, if the value of “ x ” is within the interval established by Equation (4), two output bits of the comparator select the coefficients of the polynomial section to which the input belongs. Such coefficients are stored in the 18-input, 9-output multiplexer block (18–9).

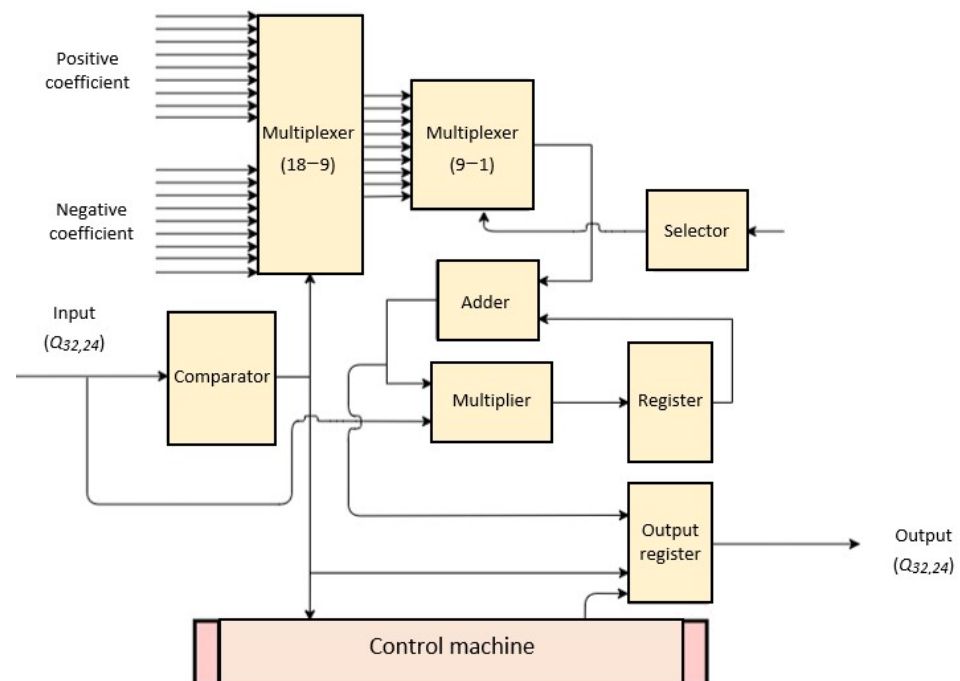


Figure 8. Hyperbolic tangent architecture using polynomial approximations.

Based on Equation (6), a second 9-input, 1-output multiplexer (9–1) chooses the first coefficient to be summed with a zero, then this result is multiplied by the input value “ x ” and consequently recorded. This process executed by the multiplexer 9–1 is repeated until the last coefficient. Once the process is finished, the total sum is recorded in the output, thus obtaining the result of the polynomial evaluation after several iterations.

The last stage of the neural network implementation consists of displaying the type of fault resulting from the network via an interface. For this purpose, a liquid crystal display (LCD) of two lines with 16 characters per line is used. The first line shows the acquired current value in real time, and the second line indicates the number of shorted turns detected by the multilayer perceptron.

The bucket sorting algorithm is among the most popular algorithms based on sorting without comparison [29]. Therefore, a bucket sorting module was implemented in the network output to obtain a device that can detect faults with better results (Figure 9). Its purpose is to sort the classification of various patterns into six buckets, representing the neural network’s six output classes. In the end, the bucket with the most patterns is the one that represents the motor failure. The user can arbitrarily choose the number of patterns to sort, considering that the higher the number, the longer the classification delay. In the case of this study, ten patterns processed and sorted by the network were chosen.

The sorting module was developed with seven counters. Six counters represent the buckets, which are enabled by a selector conditioned by the information coming from the output of the neural network of each pattern. The seventh counter is responsible for numbering the ten patterns to be sorted. Each time the network finishes processing a pattern, a flag conditioned by the network function (NF) activates the 10-pattern counter

and one of the buckets according to the network output. Once the maximum number of patterns to be sorted in the cuvettes has been reached, a cascade of comparators chooses the cuvette containing the most patterns. This information is then saved in a log and displayed in the in situ LCD screen.

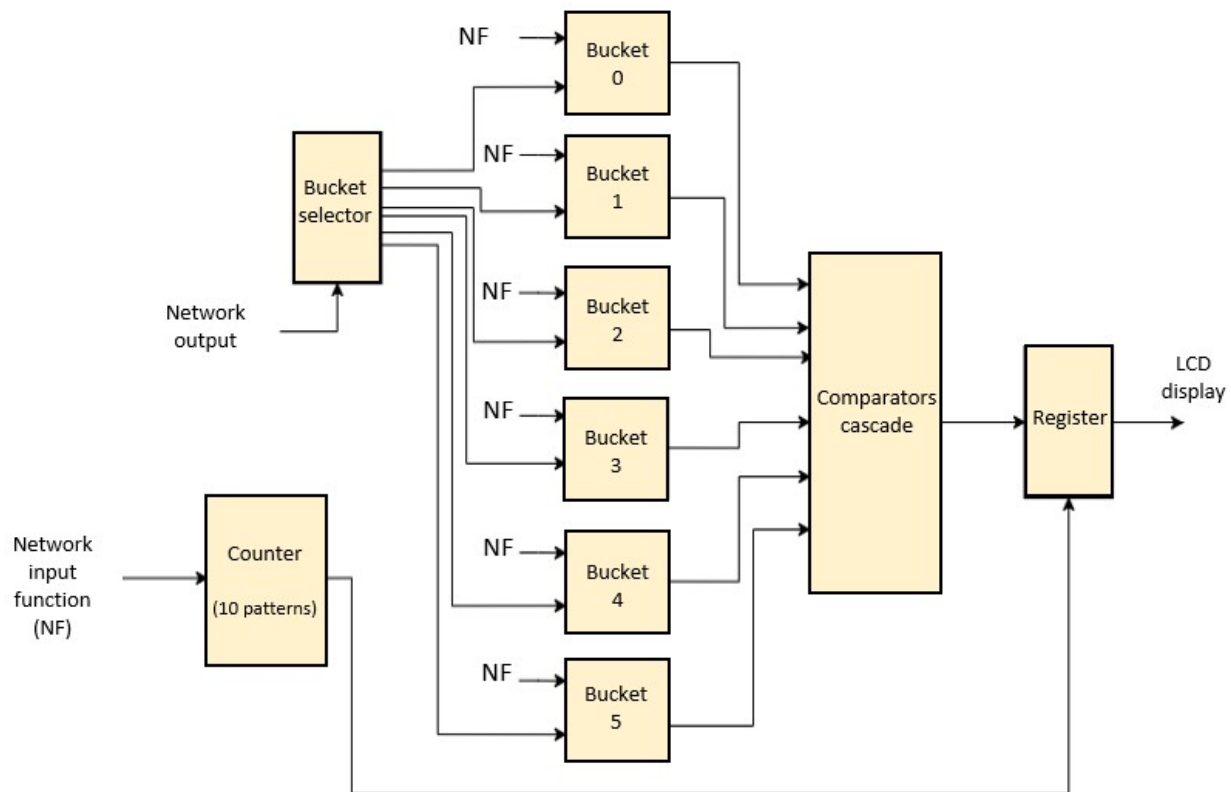


Figure 9. Module for sorting by buckets.

3.4. Data Collection

Six load states were implemented using the electromagnetic brake at different speeds to extract data corresponding to the induction motor current, as mentioned in Section 3.1. Figure 10 shows the current peaks of the measurements taken at a speed of 1745 rpm, without fault induction and with the five types of faults. This figure presents the No-Fault line (WoF), corresponding to the reference sinusoidal signal. As incipient short-circuit faults between turns are introduced in a controlled manner, a progressive attenuation in the signal amplitude is observed. This phenomenon is attributed to the generation of harmonics in the signal, which alters its behavior. Since the tests are conducted in a controlled environment, a direct correlation can be established between these changes and the presence of the motor fault. Furthermore, this figure demonstrates that these alterations are identifiable in the time domain, supporting the feasibility of temporal analysis for early fault detection. This information is recorded and classified according to the neural network processing implemented in the FPGA.

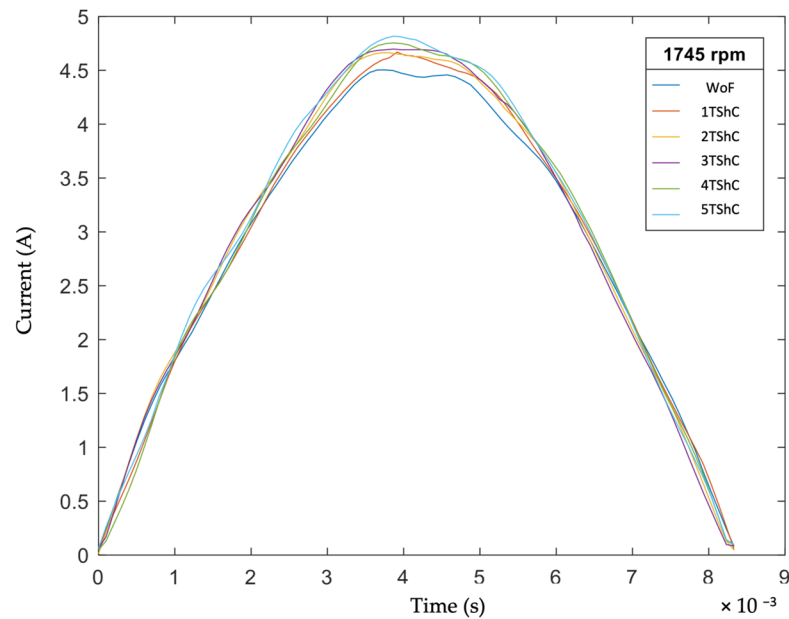


Figure 10. Current values under different fault states at a speed of 1745 rpm, without inducing faults (WoF) in blue line, and with the five types of induced short-circuit faults in the induction motor winding turns (1 TShC, 2 TShC, 3 TShC, 4 TShC and 5 TShC) remaining in colored lines.

3.5. Neural Network Design Results

The developed neural network satisfactorily classified the faults with a hit rate of 88.13%. This percentage was improved to 93.18% by adding the bucket sorting algorithm. It is important to remark that the experiment was performed n -times and the values presented are averaged. The results of the tests based on the percentage of hits for each fault condition were analyzed and are shown in Table 5. This analysis showed that in the non-fault condition (zero short-circuited turns), there was an 81.81% hit rate. As for the state of three and four short-circuited loops, it was possible to reach 100% assertiveness. It was also determined that the higher the number of short-circuited turns, the higher the percentage of correct classification.

Table 5. Hit percentage according to the induction motor fault state classification.

Number of Turns in Short-Circuit	Hit Percentage (%)
0	81.81
1	86.36
2	95.45
3	100
4	100
5	95.45

3.6. FPGA Implementation Results

The algorithm was implemented in an FPGA (Spartan 3A, Xilinx, Inc., San José, CA, USA) to demonstrate the lightness of the final designed architecture. Table 6 shows the resources of such FPGA, on which the neural network was implemented using VHDL code. These resources are considered practically low because no percentage usage value exceeds 65%. Considering that the Xilinx Spartan 3A is a low-resource development FPGA board, the project developed is optimized for implementation on low-resource boards.

Table 6. Resources used for the implementation of the VHDL code on the Xilinx Spartan 3A.

Architectural Overview	Used	Available	Percent of Use (%)
Number of Slices	3755	5888	63
Number of Flip Flop Slices	3147	11,776	26
Number of 4-input LUTs	5299	11,776	44
Number of IOBs	13	372	3
Number of block RAMs	3	20	15
Number of Multipliers	13	20	65

The processing time of the neural network implemented on the FPGA is significantly faster than that achieved using Matlab. The acquisition of the current signal requires approximately 60 ms to collect sufficient data for generating the six descriptors of each input pattern, with this delay being dictated by the signal's period. The pattern generation process is completed in 82.06 μ s, while the classification of each pattern presented to the network is performed in 45.94 μ s. Given these processing times, the proposed system qualifies as a real-time processing solution.

The definition of real-time processing depends on the sampling frequency used in industrial applications. The selected FPGA operates with a 50 MHz crystal, which, combined with its parallel processing capabilities, enables it to meet industry requirements. Notably, short-circuit coil faults are typically detected offline. However, due to the implemented system's high processing speed and parallel processing architecture, data can be acquired and processed, and faults detected in real-time, allowing for online and in situ fault monitoring. It is important to highlight that motor control falls outside the scope of this study.

As for the error that can present the 32-bit arithmetic unit implemented in the FPGA concerning the calculation in Matlab in floating point, it yields an average absolute error of 0.01 and an average relative error of 0.01. On the other hand, the error present in the 32 bits of work processed in the FPGA and in the neural network in Matlab developed at a fixed point is zero bits. For this reason, for any pattern presented to the network, the output neuron with the highest excitation is the same in both platforms, representing the existing fault in the motor.

The last stage of implementing the neural network in the FPGA consists of displaying the type of fault resulting from the network through an LCD. FPGA can perform the neural network processing properly, which allows for visualizing the fault present in the motor in real time.

4. Discussion

Induction motors are widely used and are crucial in modern industrial systems. These motors present common faults such as short-circuits between stator winding turns, bearing wear, or broken rotor bars. Therefore, detection techniques capable of diagnosing these types of faults early on have particular attention from industry and researchers. If a predictive diagnostic method is developed, it will prevent sudden production stoppage and reduce maintenance costs and financial losses.

For predictive diagnostics in IM, computers and data acquisition cards (DAQ) with offline processing and invasive sensors that necessarily interrupt the process of performing current measurements have been implemented. A few works related to developing purpose-specific devices, such as FPGAs for online monitoring and detection of short-circuit faults in induction motor turns, have been reported in the literature.

The work of Mejia et al. [30] presents a model based on multilayer neural networks to reproduce the current signatures associated with inter-turn short-circuit fault conditions. The model considers five fault severities. This model was implemented in a simulation

based on a power hardware scheme, which is a tool for testing electrical and electronic systems based on a virtual representation of a physical system. Similarly, Ewert et al. [16] present a system that allows online monitoring of induction motor bearings and subsequent fault diagnosis based on the analysis of vibration measurement data. A suitably trained neural network (NN) performs the bearing condition assessment based on spectral and envelope analysis of mechanical vibrations. It was also shown that the developed NN-based detectors are insensitive to other motor faults, such as stator winding short-circuits, broken rotor bars, or motor misalignment. Like our proposal, Mejia et al. [30] and Ewert et al. [16] employ an MLP-based diagnostic system for fault detection and identification. However, their approaches focus on different aspects, such as vibration measurement data analysis [16]. Both studies are limited to simulation-only implementations, either using a power hardware scheme [30] or the LabVIEW environment [16]. In contrast, the approach presented in this article extends beyond software implementation by incorporating a portable, non-invasive, in situ, online, and purpose-specific deployment on a standalone, real-time digital system based on a low-cost, low-range FPGA.

Osornio et al. [13] present a methodology based on the processing of electric current signals using time-domain and time-frequency-domain techniques in combination with an analysis of thermographic images that was achieved through various image processing techniques applying the short-time Fourier transform. The development of the diagnostic system has been performed with digital hardware implementations using a sensor platform based on an FPGA microprocessor running a series of algorithms developed for the online detection of multiple faults in electric motors. The system uses infrared thermograms and current signals to detect five different fault states: healthy, misaligned, unbalanced, damaged bearing, and broken bar, with an accuracy close to 99%. This work presented by Osorio et al. [13] is similar to the one presented in this manuscript in that it is a non-invasive system, and its implementation is on an FPGA. However, the method used for fault detection is based on image processing and cannot detect short-circuit faults in the winding turns of induction motors.

The contribution of the research work developed by Aib et al. [14] consists in the use of FPGA programmable logic circuits for fault diagnosis in an asynchronous machine for cases of phase unbalance and missing phase faults. This contribution introduces a fuzzy inference system in the analysis algorithm of the motor current signal by taking the effective signal of the stator phase current as the fault indicator signal. In that study, Aib et al. [14] adapted the fuzzy logic for optimal implementation. The implementation must guarantee efficiency, speed of execution, and minimum possible space in the FPGA circuit. As can be seen, this study also uses an FPGA and analyzes the motor current signals for fault detection; however, it introduces a fuzzy inference system to determine unbalance in the stator phases. According to previous research, most stator-related faults are short-circuiting faults between turns of the same phase due to insulation degradation and are very difficult to detect. Therefore, the analysis of the current between the turns is more effective for fault detection than the starting vibration of a motor. For this reason, our research is focused on detecting this type of fault.

As mentioned, even though our research uses the data acquisition system presented in González et al. [22], specific improvements are reached in the present article. The existing problem in creating the descriptors when choosing the parameters is eliminated since the summation of the complete half-cycle is performed and not from an empirically determined threshold. Also, the MLP implemented in this article has a higher classification rate than the network developed by Gonzales et al. [22]. This classification presents a higher accuracy when applying the bucket sorting algorithm. Finally, it was possible to obtain a system embedded in a standalone, portable, non-invasive and purpose-specific device capable of

performing online processing and showing, in real time, through an interface, the fault present in the engine.

Regarding fault diagnosis in induction motor coils using artificial intelligence, recent studies have explored advanced strategies, such as the integration of traditional regression approaches with machine learning techniques [31]. Likewise, Long Short-Term Memory (LSTM) neural networks have been investigated for fault prediction [32]. These architectures represent an advanced variant of ANN, with a more complex structure and functionality. While these methodologies offer high potential in data processing and analysis, they have certain limitations, such as higher computational cost, significant demand for memory and training time, and a propensity for overfitting. In addition, their interpretation is more complex compared to simpler approaches, such as the MLP used in the present study, due to the greater number of layers and hidden states involved in its operation.

In this context, although the current trend in AI favors massive data expansion and the exponential growth of neurons in intermediate layers—leveraging the computational power of cloud servers from companies such as Google (Mountain View, CA, USA) or IBM (Armonk, NY, USA), among others, via collaborative platforms—these strategies are not directly applicable to this work. The approach presented in this study focuses on the implementation of embedded systems for real-time applications, where large-scale data processing is not feasible. In this scenario, the use of a MLP is sufficient for fault detection, offering advantages in terms of energy efficiency, hardware requirements, computational resources, and processing time. Furthermore, relying on large-scale computing infrastructures would significantly increase operational costs in industrial applications, limiting their practicality in real-world production environments.

In summary, the present article proposes a portable system, which offers several advantages over other solutions presented in Section 4. These include a short response time due to real-time processing, high accuracy in fault detection, and a digitally simple algorithm that enables implementation on a low-cost, low-range, purpose-specific FPGA board. Additionally, the system operates independently, without requiring external devices, making it a fully stand-alone solution.

5. Conclusions

This study presents a standalone, non-invasive, and portable system capable of detecting short-circuit faults in winding turns. The proposed system performs real-time processing, in situ and online measurements of variations in stator load conditions of induction motors without interrupting their operation.

A multilayer perceptron neural network (MLP) was successfully developed to classify current faults, achieving an accuracy rate of 93.18%. The performance analysis, based on the percentage of correct classifications for each fault condition, demonstrated that for the non-fault condition (zero short-circuit turns, WoF), the classification accuracy was 81.81%. For a single short-circuit turn, an accuracy of 86.26% was obtained, while for three and four short-circuit turns, the system achieved an accuracy of 100%. Additionally, for two and five short-circuit turns, the accuracy reached 95.45%.

The system incorporates a data acquisition stage, which proved to be effective and sufficiently accurate to obtain phase current values in a non-invasive manner. Furthermore, the descriptor model applied to the acquired current data (Equation (2)) successfully captured the relevant current characteristics, allowing the formation of patterns that could be effectively classified by the neural network. The final digital algorithm is computationally efficient, allowing its implementation on a low-cost, low-range, purpose-specific FPGA board for real-time classification of short-circuit faults in induction motor winding turns. The classification results are displayed on an LCD screen.

The findings of this study highlight that FPGA-based implementations offer high real-time processing capabilities at relatively low cost. The open and reconfigurable architecture of the FPGA facilitates upgrades and modular integration to adapt to changing requirements. Furthermore, its internal hardware parallelism enables fast and efficient processing for online diagnostic systems. In summary, this work successfully demonstrates the development of a portable electronic device capable of detecting and displaying fault classifications in real-time without interrupting engine operation, allowing users to take timely preventive measures. While the proposed approach offers a functional solution for engine fault diagnosis, certain limitations prevent it from being considered a fully general and comprehensive solution. Nevertheless, the results suggest that the proposed method has significant potential for effective and efficient application to other engine types and real-world operating conditions. Future work may focus on extending the system's capability to detect more than five short-circuit loops, as well as investigating its applicability to detect other types of faults.

Author Contributions: Conceptualization, R.S.-M. and A.E.-C.; methodology, F.J.P.-P. and A.G.S.S.; software, R.S.-M., R.J.G.-G. and L.d.C.G.-R.; validation, J.d.J.R.-M. and R.J.G.-G.; formal analysis, A.G.S.S.; investigation, A.E.-C. and J.d.J.R.-M.; resources, A.E.-C. and R.J.G.-G.; writing—original draft preparation, L.d.C.G.-R. and R.J.G.-G.; writing—review and editing, L.d.C.G.-R., A.E.-C. and F.J.P.-P.; visualization, L.d.C.G.-R. and A.E.-C.; supervision, A.E.-C. and R.S.-M.; project administration, A.E.-C. All authors have read and agreed to the published version of the manuscript.

Funding: Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCyT), M.Sc.grant (Number 728531) to R.J.G.-G.

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study. Requests to access the datasets should be directed to alejandro.espinosa@crodecelaya.edu.mx.

Acknowledgments: The authors thank Tecnológico Nacional de México (TecNM) and Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCyT), Mexico, for the M.Sc.grant to R.J.G.-G.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclatures

A	Amperes unit of measure
AC	Alternating current
AI	Artificial intelligence
ALU	Arithmetic Logic Unit
ANFIS	Adaptive neuron–fuzzy inference system
ANN	Artificial neural network
ASICs	Application-Specific Integrated Circuit
C_j	Ridge descriptor
CM	Control machine
CNNs	Convolutional neural networks
DAQ	Data acquisition cards
DSP	Digital signal processor
EPRI	Electric Power Research Institute
FPGA	Field-programmable gate arrays
GA	Genetic algorithm
hp	Power measurement unit
$H(x)$	Polynomial approximation method

IEEE	Institute of Electrical and Electronics Engineers
IM	Induction motors
IOBs	Input and output blocks
k	Constante
kW	Kilowatt unit of measure
LCD	Liquid-crystal display
LOCS	Locations in a data vector
LSTM	Likewise, Long Short-Term Memory neural networks
LUTs	Matrix of numbers that can be referred to by subscripts (Look up Tables)
MLP	Multilayer perceptron
MLPARD	Multilayer perceptron automatic relevance determination
MLPANN	Multilayer perceptron artificial neural network
ms	Unit of time in the International System of Units (SI) equal to one thousandth of a second.
NF	Network function
PC	Personal computer
PKS	Maximum peaks
P_n	P is the training pattern, n is the number of patterns
$Q_{12,10}$	Unsigned 12-bit format, of which 10 represent the fractional part
$Q_{32,24}$	32-bit unsigned format with 24 bits for the fractional part
$Q_{32,31}$	31 bits for fractional representation and one bit for the sign
R	Number of samples of the half-cycles of the acquired signal
RA	Correlation rate of learning
RAM	Random access memory
RG	Overall correlation rate
I_{rms}	Root mean square current value
ROM	Read Only Memory
rpm	Revolutions per minute
RS-232	Recommended Standard 232
RV	Validation correlation rate
SLP	Single-layer perceptron
SoCs	System on a chip
SVM	Support vector machine
TShC	Turns in short-circuit
USB	Universal Serial Bus
V	Volts unit of measure
VHDL	Very High Speed Integrated Circuits
WoF	Without fault operating conditions
x	Represents the half-cycle of the acquired current signal
μs	Unit of time in the International System of Units (SI) equal to one millionth of a second.

References

1. Gyftakis, K.N.; Spyropoulos, D.V.; Mitronikas, E.D. Advanced Detection of Rotor Electrical Faults in Induction Motors at Start-Up. *IEEE Trans. Energy Convers.* **2021**, *36*, 1101–1109. [\[CrossRef\]](#)
2. Garcia-Calva, T.; Morinigo-Sotelo, D.; Fernandez-Cavero, V.; Romero-Troncoso, R. Early detection of faults in induction motors—A review. *Energies* **2022**, *15*, 7855. [\[CrossRef\]](#)
3. Smirnov, A.I.; Voytyuk, I.N. Diagnostics of inter-turn short-circuit in the stator winding of the induction motor. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *643*, 012023. [\[CrossRef\]](#)
4. Alawady, A.A.; Yousof MF, M.; Azis, N.; Talib, M.A. Phase to phase fault detection of 3-phase induction motor using FRA technique. *Int. J. Power Electron. Drive Syst.* **2020**, *11*, 1241. [\[CrossRef\]](#)
5. Gangsar, P.; Tiwari, R. Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review. *Mech. Syst. Signal Process.* **2020**, *144*, 106908. [\[CrossRef\]](#)
6. Zhang, S.; Wang, B.; Kanemaru, M.; Lin, C.; Liu, D.; Miyoshi, M.; Teo, K.H.; Habetler, T.G. Model-based analysis and quantification of bearing faults in induction machines. *IEEE Trans. Ind. Appl.* **2020**, *56*, 2158–2170. [\[CrossRef\]](#)

7. Gundewar, S.K.; Kane, P.V. Condition monitoring and fault diagnosis of an induction motor. *J. Vib. Eng. Technol.* **2021**, *9*, 643–674. Available online: <https://link.springer.com/article/10.1007/s42417-020-00253-y> (accessed on 1 September 2024). [CrossRef]
8. Nath, A.G.; Udmale, S.S.; Singh, S.K. Role of artificial intelligence in rotor fault diagnosis: A comprehensive review. *Artif. Intell. Rev.* **2021**, *54*, 2609–2668. [CrossRef]
9. Kumar, P.; Hati, A.S. Review on machine learning algorithm based fault detection in induction motors. *Arch. Comput. Methods Eng.* **2021**, *28*, 1929–1940. [CrossRef]
10. Toma, R.N.; Prosvirin, A.E.; Kim, J.M. Bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers. *Sensors* **2020**, *20*, 1884. [CrossRef]
11. Heidari, A.A.; Faris, H.; Mirjalili, S.; Aljarah, I.; Mafarja, M. Ant lion optimizer: Theory, literature review, and application in multi-layer perceptron neural networks. In *Nature-Inspired Optimizers: Theories, Literature Reviews, and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 23–46. Available online: https://link.springer.com/chapter/10.1007/978-3-030-12127-3_3 (accessed on 1 August 2024).
12. Khanjani, M.; Ezoji, M. Electrical fault detection in three-phase induction motor using deep network-based features of thermograms. *Measurement* **2021**, *173*, 108622. [CrossRef]
13. Osornio-Rios, R.A.; Cueva-Perez, I.; Alvarado-Hernandez, A.I.; Dunai, L.; Zamudio-Ramirez, I.; Antonino-Daviu, J.A. FPGA-Microprocessor Based Sensor for Faults Detection in Induction Motors Using Time-Frequency and Machine Learning Methods. *Sensors* **2024**, *24*, 2653. [CrossRef]
14. Aib, A.; Khodja, D.E.; Chakroune, S.; Rahali, H. Fuzzy current analysis-based fault diagnostic of induction motor using hardware co-simulation with field programmable gate array. *Electr. Eng. Electromechanics* **2023**, *6*, 3–9. [CrossRef]
15. Khoualdia, T.; Lakehal, A.; Chelli, Z.; Khoualdia, K.; Nesaib, K. Optimized multi layer perceptron artificial neural network based fault diagnosis of induction motor using vibration signals. *Diagnostyka* **2021**, *22*, 65–74. [CrossRef]
16. Ewert, P.; Kowalski, C.T.; Orlowska Kowalska, T. Low-cost monitoring and diagnosis system for rolling bearing faults of the induction motor based on neural network approach. *Electronics* **2020**, *9*, 1334. [CrossRef]
17. Bazan, G.H.; Scalassara, P.R.; Endo, W.; Goedel, A.; Palacios RH, C.; Godoy, W.F. Stator Short Circuit Diagnosis in Induction Motors Using Mutual Information and Intelligent Systems. *IEEE Trans. Ind. Electron.* **2018**, *66*, 3237–3246. [CrossRef]
18. Bazan, G.H.; Goedel, A.; Duque-Perez, O.; Morinigo-Sotelo, D. Multi-fault diagnosis in three-phase induction motors using data optimization and machine learning techniques. *Electronics* **2021**, *10*, 1462. [CrossRef]
19. Li, X.; Jiang, H.; Niu, M.; Wang, R. An enhanced selective ensemble deep learning method for rolling bearing fault diagnosis with beetle antennae search algorithm. *Mech. Syst. Signal Process.* **2020**, *142*, 106752. [CrossRef]
20. Babu, P.; Parthasarathy, E. Reconfigurable FPGA architectures: A survey and applications. *J. Inst. Eng. Ser. B* **2021**, *102*, 143–156. [CrossRef]
21. Shawahna, A.; Sait, S.M.; El-Maleh, A. FPGA-based accelerators of deep learning networks for learning and classification: A review. *IEEE Access* **2019**, *7*, 7823–7859. [CrossRef]
22. González, R.J.G.; Calderón, A.E.; Montero, R.S.; Ramírez, A.Z.; Magdaleno, J.d.J.R. FPGA embedded system for non-invasive current measurement in induction motors. In Proceedings of the 2021 IEEE International Summer Power Meeting/International Meeting on Communications and Computing (RVP-AI/ROC&C), Acapulco, Mexico, 14–18 December 2021; pp. 1–5. [CrossRef]
23. RS-323; Standard for Serial Data Communication. EIA: Arlington, VA, USA, 1969.
24. Principi, E.; Rossetti, D.; Squartini, S.; Piazza, F. Unsupervised electric motor fault detection by using deep autoencoders. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 441–451. [CrossRef]
25. Cherif, H.; Benakcha, A.; Laib, I.; Chehaidia, S.E.; Menacer, A.; Soudan, B.; Olabi, A.G. Early detection and localization of stator inter-turn faults based on discrete wavelet energy ratio and neural networks in induction motor. *Energy* **2020**, *212*, 118684. [CrossRef]
26. AlShorman, O.; Irfan, M.; Saad, N.; Zhen, D.; Haider, N.; Glowacz, A.; AlShorman, A. A review of artificial intelligence methods for condition monitoring and fault diagnosis of rolling element bearings for induction motor. *Shock. Vib.* **2020**, *2020*, 8843759. [CrossRef]
27. Zhang, X.; Liu, C.A. Model averaging prediction by K-fold cross-validation. *J. Econom.* **2023**, *235*, 280–301. [CrossRef]
28. García-Rodríguez, L.d.C.; Prado-Olivarez, J.; Guzmán-Cruz, R.; Heil, M.; Guevara-González, R.G.; Diaz-Carmona, J.; López-Tapia, H.; Padierna-Arvizu, D.d.J.; Espinosa-Calderón, A. Black-Box Mathematical Model for Net Photosynthesis Estimation and Its Digital IoT Implementation Based on Non-Invasive Techniques: *Capsicum annuum* L. Study Case. *Sensors* **2022**, *22*, 5275. [CrossRef]
29. Gill, S.K.; Singh, V.P.; Sharma, P.; Kumar, D. A comparative study of various sorting algorithms. *Int. J. Adv. Stud. Sci. Res.* **2019**, *4*, 367.
30. Mejia-Barron, A.; Tapia-Tinoco, G.; Razo-Hernandez, J.R.; Valtierra-Rodriguez, M.; Granados-Lieberman, D. A neural network-based model for MCSA of inter-turn short-circuit faults in induction motors and its power hardware in the loop simulation. *Comput. Electr. Eng.* **2021**, *93*, 107234. [CrossRef]

31. Shih, K.J.; Hsieh, M.F.; Chen, B.J.; Huang, S.F. Machine learning for inter-turn short-circuit fault diagnosis in permanent magnet synchronous motors. *IEEE Trans. Magn.* **2022**, *58*, 1–7. [[CrossRef](#)]
32. Samanta, S.; Bera, J.N.; Biswas, A. Induction Motor Inter-turn Short Circuit Fault Classification by Extracting Auto Features Using LSTM Neural Network. In Proceedings of the 2024 IEEE 3rd International Conference on Control, Instrumentation, Energy & Communication (CIEC), Kolkata, India, 25–27 January 2024; pp. 180–185.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.