

Article

Incipient Inter-Turn Short Circuit Detection in Induction Motors Using Cumulative Distribution Function and the EfficientNetv2 Model

Carlos Javier Morales-Perez ¹, Laritza Perez-Enriquez ², Juan Pablo Amezcuita-Sanchez ¹,
Jose de Jesus Rangel-Magdaleno ³, Martin Valtierra-Rodriguez ¹ and David Granados-Lieberman ^{4,*}

- ¹ ENAP-Research Group, CA-Sistemas Dinámicos, Facultad de Ingeniería, Universidad Autónoma de Querétaro (UAQ), Campus San Juan del Río, Río Moctezuma 249, Col. San Cayetano, San Juan del Río C.P. 76807, QRO, Mexico; carlosj.morales@ieee.org (C.J.M.-P.); juan.amezcuita@enap-rg.org (J.P.A.-S.); martin.valtierra@enap-rg.org (M.V.-R.)
 - ² Coordinación de Ciencias y Tecnologías del Espacio, Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Luis Enrique Erro #1, Sta. María Tonanzintla, San Andrés Cholula C.P. 72840, PUE, Mexico; laritza@inaoe.mx
 - ³ Digital Systems Group, Coordinación de Electrónica, Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Luis Enrique Erro #1, Sta. María Tonanzintla, San Andrés Cholula C.P. 72840, PUE, Mexico; jrangel@inaoe.mx
 - ⁴ ENAP-Research Group, CA-Fuentes Alternas y Calidad de la Energía Eléctrica, División de Ingeniería en Electromecánica, Tecnológico Nacional de México, ITS Irapuato, Carr. Irapuato-Silao km 12.5, Colonia El Copal, Irapuato C.P. 36821, GTO, Mexico
- * Correspondence: david.granados@enap-rg.org



Citation: Morales-Perez, C.J.; Perez-Enriquez, L.; Amezcuita-Sanchez, J.P.; de Jesus Rangel-Magdaleno, J.; Valtierra-Rodriguez, M.; Granados-Lieberman, D. Incipient Inter-Turn Short Circuit Detection in Induction Motors Using Cumulative Distribution Function and the EfficientNetv2 Model. *Machines* **2024**, *12*, 399. <https://doi.org/10.3390/machines12060399>

Academic Editors: Ahmed Ragab and Mouloud Amazouz

Received: 6 May 2024

Revised: 28 May 2024

Accepted: 6 June 2024

Published: 12 June 2024



Copyright: © 2024 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/>).

Abstract: Induction motors are one of the most used machines because they provide the necessary traction force for many industrial applications. Their easy operation, installation, maintenance, and reliability make them preferred over other electrical motors. Mechanical and electrical failures, as with other machines, can appear at any stage of their service life, making the stator intern-turn short-circuit fault (ITSC) stand out. Hence, its detection is necessary in order to extend and save useful life, avoiding a breakdown and unprogrammed maintenance processes as well as, in the worst circumstances, a total loss of the machine. Nonetheless, the challenge lies in detecting this type of fault, which has made the analysis and diagnosis processes easier. Such is the case with convolutional neural networks (CNNs), which facilitate the development of methodologies for pattern recognition in several areas of knowledge. Unfortunately, these techniques require a large amount of data for an adequate training process, which is not always available. In this sense, this paper presents a new methodology for the detection of incipient ITSC faults employing a modified cumulative distribution function (CDF) of the current stator signal. Then, these are converted to images and fed into a fast and compact CNN model, trained with a small data set, reaching up to 99.16% accuracy for seven conditions (0, 5, 10, 15, 20, 30, and 40 short-circuited turns) and four mechanical load conditions.

Keywords: convolutional neural network; cumulative distribution function; current stator signal; induction motor; incipient fault; short-circuit fault

1. Introduction

Nowadays, maximizing the useful life of induction motors (IMs) is essential because these machines are necessary for industrial applications; they are easy to install and maintain, and are reliable. Even so, these machines can break for diverse reasons, including manufacturing defects, bad installations, or lack of maintenance. In this way, the importance of these machines is evident, given that 60% of industrial electrical consumption is due to the operation of IMs [1,2]. According to the above, several approaches and techniques have been presented and published in the literature to monitor and analyze

the operational signals of IMs in order to assess their conditions, making monitoring and maintenance programs more accurate and sophisticated. On the other hand, electrical failure represents two-fifths of total failures, with the inter-turn short-circuit (ITSC) being the most prevalent failure in this type of machine [3].

Over the years, the scientific and technical community has worked to develop techniques and approaches for creating automatic systems and expert diagnoses for faults in electrical machines. The automatic detection of ITSC failures on IMs is possible due to the existence of signatures or patterns directly related to the nature of the fault [4,5]. Thus, if these anomalies are detected in the physical variables during the machine's operation, it is possible to determine the type of fault and its severity. In this sense, much of the research is focused on the current signal analysis due to the direct relationship between the current behavior with the presence of a short circuit in the motor stator, giving important contributions to the motor current signal analysis (MCSA) [5,6]. According to the above, the statistical features [7] and the wavelet technique for ITSC fault detection are widely applied due to the effectiveness of the signal extraction features in the time domain [8–11]. Besides that, approaches with Fast Fourier transforms (FFTs) have been implemented to search for spectral components related to the failure [12,13]. However, the challenge increases when the machine's natural behavior masks the failure, which is typically presented when the failure is incipient. This results in an inaccurate diagnosis instead of an alert about an initial fault.

Artificial Neural Networks (ANNs) have improved in effectiveness and architecture, allowing their application in various areas of knowledge [14–16]. Today, these exceptional architectures play a crucial role in pattern recognition [17,18], extending to fault detection and classification on IMs [19–21]. In such a way, Skowron et al. [22] present a classification approach based on Self-Organizing Neural Networks (SONNs) to classify electrical winding faults in IMs fed from an industrial frequency converter. In this work, Skowron and collaborators use the instantaneous symmetrical components of the stator current spectra (ISCA) to classify, among others, ITSC faults as minimal one-short-circuited turns (SCTs). On the other hand, Saucedo-Dorantes [11] integrate a SONN method with empirical wavelet transform and statistical indicators to diagnose an IM with various levels of ITSC. Babanezhad et al. [23] introduce a technique using a Probabilistic Neural Network (PNN) to estimate turn-to-turn faults in IMs, evaluating the negative sequence current to estimate up to 1 SCT. Maraaba et al. [24] present another method that applies a multi-layer feedforward neural network (MFNN) to diagnose the stator winding faults mean by statistical coefficients and frequency features, reaching a detection of 5% of SCTs. Also, Guedidi et al. [25] employ a modified SqueezeNet, a convolutional neural network (CNN), to develop the fault detection of ITSC fault in IMs using 3D images generated from image transformation based on Hilbert transform (HT) and variational mode decomposition (VMD), reaching the detection of five SCTs. Alipoor et al. [26] propose a long short-term memory (LSTM) model for detecting five specific ITSCs in an IM. This approach utilizes features derived from empirical mode decomposition alongside 21 statistical indices.

Despite the above, some drawbacks of applying machine learning (ML) or deep learning (DL) techniques are the high computational cost and the necessary large amount of data for the training process [17,27]. While the computational cost is relatively easy to solve, access to data sets with enough samples for training ML or DL architectures is not always available. An alternative is the Transfer Learning (TL) technique, which reduces data and the time required for the training process in DL architectures [28]. In this sense, DL architectures trained previously can transfer knowledge to the development of other specific tasks. Thus, this technique can resolve the problem of a lack of a large amount of data necessary to develop the training process for fault detection, as Shao exposes in [29], and the development of precise ITSC fault detection in combination with the reduction in computational load as exposed by Guedidi et al. [25].

This paper presents a new methodology for incipient ITSC fault detection in IMs based on modified cumulative distribution functions (CDFs) computed from the stator current

and a CNN model implementation. The CNN used belongs to a smaller and faster family of CNNs. In this manner, at least five SCTs are detected at different levels of mechanical load with 100 samples per condition for the training process, reaching the classification of 28 fault conditions (healthy condition and 6 fault severity levels in 4 mechanical load conditions) with an accuracy rate up to 99.16%. This work is organized as follows: Section 2 introduces the overview of the necessary background for the ITSC faults and DL techniques, in addition to the proposed methodology explanation and the description of the used test bench. Section 3 explains the development of the methodology and its results. Subsequently, Section 4 exposes the interpretation of the obtained results. Finally, Section 5 exhibits the conclusions about the developed work.

2. ITSC Faults and DL Techniques

Stator winding faults are one of the most common faults that could be present into the IMs. According to the origin of these faults, these can be principally categorized as follows [30,31]:

1. Turn to turn;
2. Phase to phase;
3. Phase to ground.

These faults can be schematized as depicted in Figure 1. The ITSC fault belongs to the turn-to-turn fault (see Figure 1a), the study object in this work. This fault can be provoked by the degradation of the conductor coating, which can be a product of the coating's natural aging or, in most cases, of the rise in the stator temperature over the design parameters [31]. The circulation of high currents in the stator is the most common factor initiating this condition by the Joule effect, generally provoked by excessive mechanical loads during the start-up cycle. Also, if the fault is derived between stator phases, this fault is the named phase-to-phase fault (see Figure 1b), or if the fault is presented between the phase to chassis, it can create a derivation to the ground, generating the phase-to-ground fault (see Figure 1c). The origin could be the same regardless of the fault type.

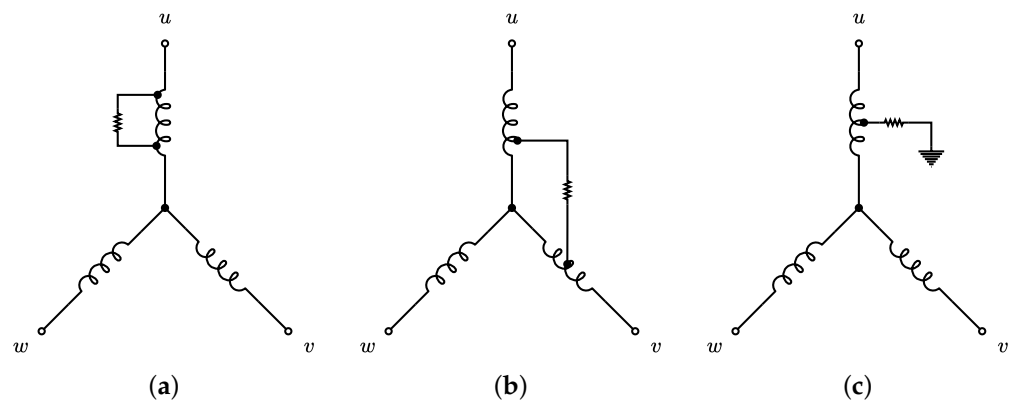


Figure 1. Stator winding faults types: (a) turn to turn; (b) phase to phase; and (c) phase to ground.

On the other hand, the presence of ITSC in the stator creates a disturbance in the magnetic flux distribution in the air gap, inducing spurious currents that inject harmonics into the stator. These harmonic components f_h can be located in the power spectral density as:

$$f_h = f_o \left[\frac{a}{p} (1 - s) \pm b \right] \quad (1)$$

where f_o is the fundamental frequency of the power source, a is the harmonic's number (odd number), p is the pole pair, s is the slip, and b is an index. Since these faults can occur suddenly and gradually worsen over time, the fault presence is not evident until the machine breaks down. In this way, early detection of this fault is desirable to program

corrective action on time. The harmonics induced by the presence of this fault in the current stator generate distortion in the shape of the sine wave provided by the electrical power source. As shown in Figure 2, the amplitude of the current stator signal, experimentally acquired as described in Section 2.3, appears to be increased by the level of damage, implying that it can be misleading since the increase in the mechanical load has a similar effect in the current stator, even in healthy conditions. These effects make the detection hard in the time domain.

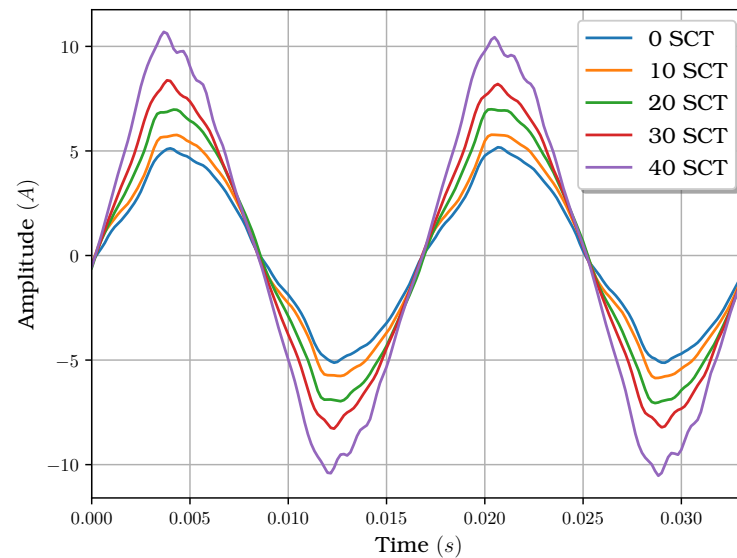


Figure 2. The experimental waveform of distortions related to the ITSC fault in the current stator signal in the time domain, with different levels of severity.

However, if the signal is translated to another representation that allows information preservation, it could be enhanced to detect the ITSC harmonics signature efficiently. So, letting a set $X = [x_0, x_1, \dots, x_{N-1}] \forall x_{N-1} \in \mathbb{R}$, the CDF F_m is defined as (2):

$$F_m = P(X \leq \tilde{x}_m) \quad (2)$$

where $P(X \leq \tilde{x}_m)$ is a value counter which satisfies the inequality condition $X \leq \tilde{x}_m$ and $\tilde{x}_m \in \mathbb{R}$. Thus, the CDF of the current signals shows significant changes in shape (see Figure 3a) concerning the sine wave (see Figure 2), translating the information in a shorter data length, but does not reach enough separability between classes (damage stages at different mechanical loads).

To get through this, removing the typical linear incremental component could be a solution to highlight the features inserted by the different fault conditions, obtaining a kind of distorted one-cycle sine wave (see Figure 3b), in which the distortion could represent the signature of an existing fault. The processing of the CDF described above is presented as follows:

$$y_m = F_m - \frac{N}{M-1}m \quad (3)$$

where $y_m \in Y : Y \in \mathbb{R}^M$, M is the number of classes of the CDF, and F_m is the m -th observation computed from (2). In this sense, the next challenge is developing or implementing a technique to detect small features in the modified CDF.

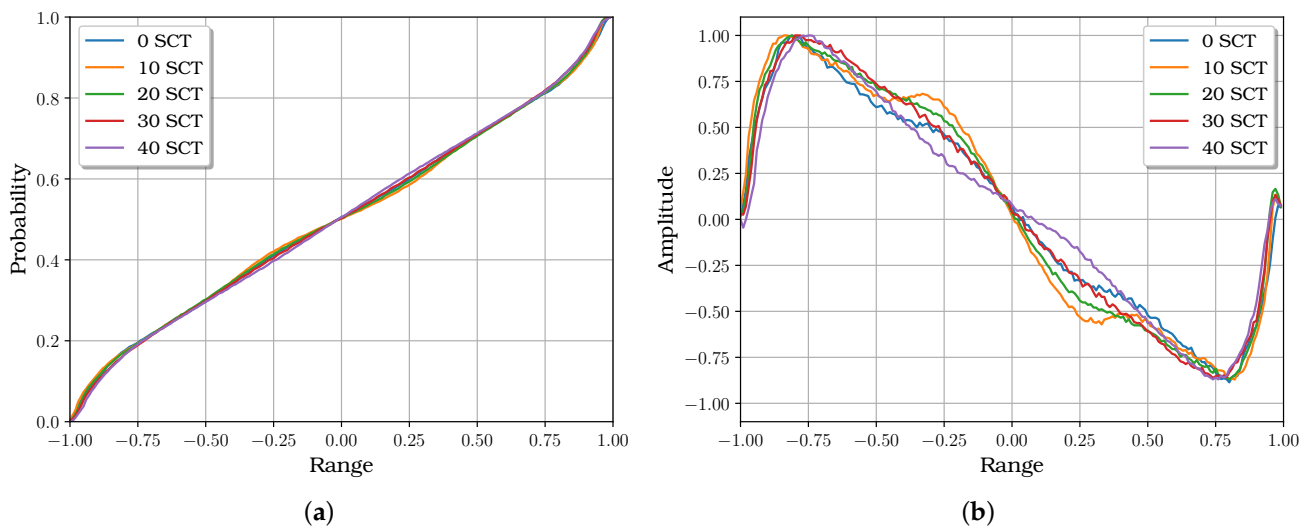


Figure 3. CDFs from current stator signals: (a) original CDFs and (b) modified CDFs.

2.1. Convolutional Neural Network Overview

Proposed by LeCun et al. [32] in 1995, the CNN model is a kind of ANN with self-optimizable neurons and a more complex architecture, designed to emulate the visual cortex of the biological brain, be able to detect oriented edges and endpoint corners, and others.

Its architecture (see Figure 4) often consists of multiple layers; according to their function, these can be classified principally as three layer types [33]: convolutional, pooling, and fully connected. The convolutional layer applies the convolutional function to create feature maps to detect specific features. This layer can be denoted as in (4):

$$c_i = \sigma(\mathbf{w} * \mathbf{x} + \mathbf{b}) \quad (4)$$

where $\mathbf{w}$ is the filter or kernel, $\mathbf{x}$ is the image, $\mathbf{b}$ is the bias, σ is an activation function, and c_i is the i -th feature map. Then, the convolutional layer is connected to the pooling layer, which applies down-sampling to reduce the data from the feature map, saving memory usage and training speed. The pooling layer can be represented as stated in (5):

$$p_i = \text{pool}(c_i) \quad (5)$$

where $\text{pool}(\text{dot})$ is the pooling function and p_i is the i -th network output. The fully connected layer consists of neurons connected between them, and its outputs are directly related to the predicted labels. This layer is usually located after a stack of convolutional layers, and it is represented as:

$$l = a(\mathbf{W} + \mathbf{b}) \quad (6)$$

where $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ is the bias, and l is the predicted label. Due to their intrinsic features, these neural networks can classify wide data diversity. In addition, these architectures have been applied in areas such as computer vision, natural language processing, medical, industrial automation, and robotics, among others. Their versatility and ability to learn complex patterns make them a powerful tool in artificial intelligence.

Since their introduction in 1995, several models have been presented and included in the literature, solving and handling diverse problems and issues exposed by their predecessors. In particular, EfficientNetV2 is an innovative CNN introduced by Tan and Le in 2021 [34] as a new family of minor architecture, parameter efficiency, and faster training neural networks. This CNN solves the bottleneck presented by the predecessor, EfficientNet [35], proposing an improved progressive learning method and increasing the training speed to 11 times over the current architectures. The progressive learning method allows progressive changes in the image size and adaptive regularization adjustment

during the training process, speeding up and making learning simple representations easier in the early stages.

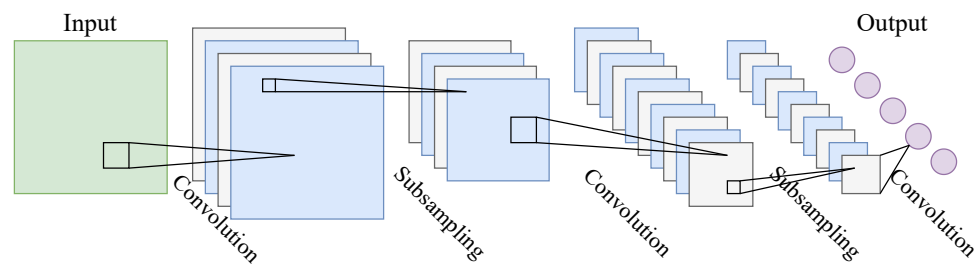


Figure 4. Example of CNN architecture (inspired by [33]).

The main improvement in this architecture over the last is the replacement of the MBConv with the novelty Fused-MBConv blocks (see Figure 5); in the first layers, both are combined, which results in an enriched search space, which increases speed and upholds accuracy. Finally, the authors recommend using the pre-trained model with the ImageNet data set to improve efficiency in future applications.

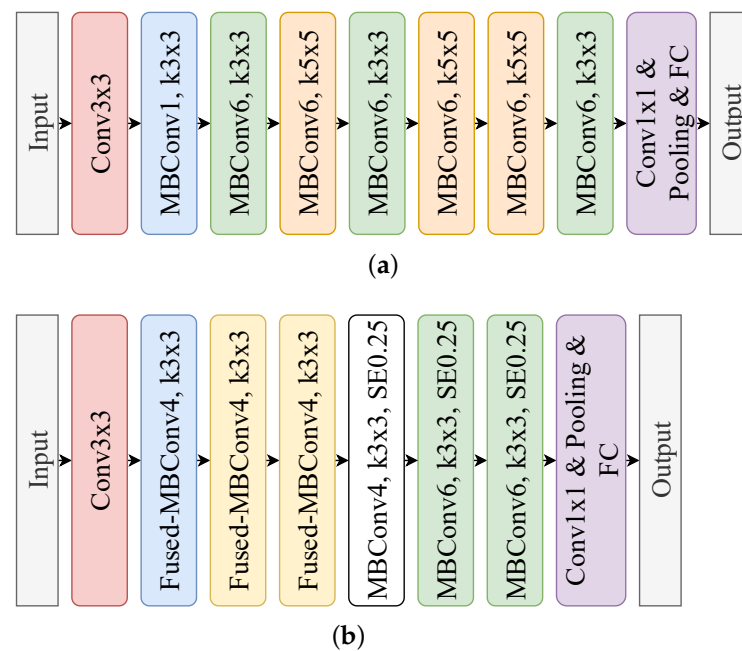


Figure 5. Architecture baselines: (a) EfficientNet [35], and (b) EfficientNetv2 [34].

2.2. Methodology

The proposed methodology is based on analyzing and processing the motor stator's current signal using DL techniques to detect the ITSC fault in the early stages. The proposal is formed by two principal stages: pre-processing and detection. The steps to develop this are listed below.

2.2.1. Pre-Processing

1. First, acquire the current signal from the motor stator of the IM to diagnose in a steady state to save N samples ($X \in \mathbb{R}^N$) for the following stages. Consider a normalization process of the signals between the $(-1, +1)$ range.
2. Obtain the CDF of X with m classes ($F \in \mathbb{R}^M$) in concordance with (2).
3. Delete the linear incremental component, applying (3), and obtain $Y \in \mathbb{R}^M$. Also, consider the normalization of Y in the $(-1, +1)$ range.

4. Finally, generate an image of $3 \times W \times H$ dimension from Y , ensuring the area under the curve is shaded. Note that W is the wide and H is the height.

These steps are depicted in Figure 6.

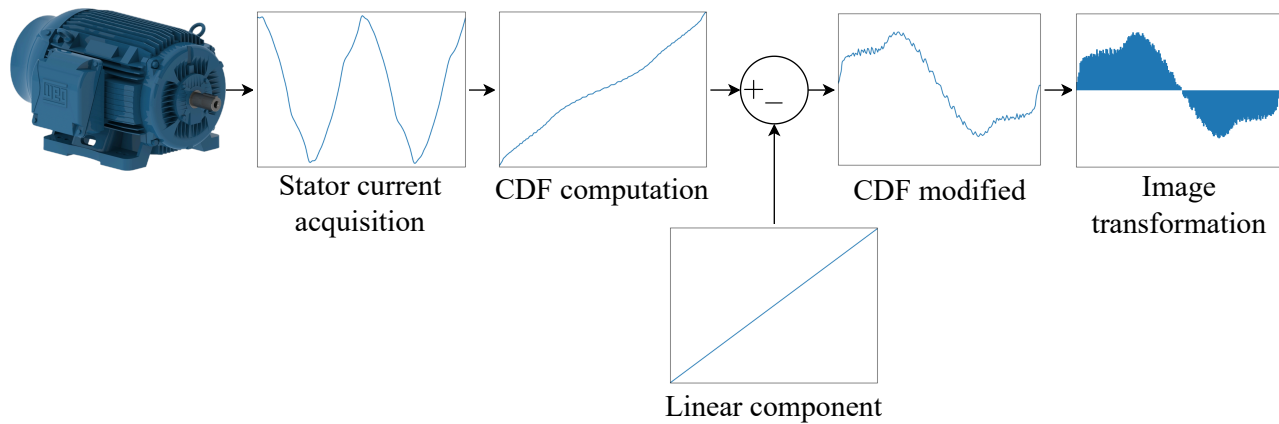


Figure 6. Block diagram of the pre-processing stage.

2.2.2. Detection

The detection process is principally formed of two stages to apply the DL technique correctly: training and testing. The training process is developed once to give the CNN architecture the necessary information about the faults to detect (in this work), learning and applying them in the testing process. This last stage is used as required, performing the detection and finishing the methodology. The steps are listed as follows.

1. Training process.

- (a) Initially, load the EfficientNetv2 model B0 version and realize the model transfer, preserving the early layers and omitting the top layers.
- (b) Then, construct and connect a new output layer in concordance with the desired output labels.
- (c) Next, prepare the data set with limited samples containing the processed CDF with different damage levels. Ensure the generated images (samples) and their labels are compatible with the CNN architecture's input and output formats.
- (d) Launch a training process to adjust the weights with the prepared data set.
- (e) Last, when the training process is finished, save the adjusted weights for the testing process development.

2. Testing process.

- (a) First, create a CNN model as steps a and b indicate in the training process.
- (b) After, load the obtained weights from the training process to the created model.
- (c) The image obtained from the modified CDF is introduced to the CNN, and the evaluation function is run.
- (d) Last, recover the results from the evaluation process to obtain the diagnosis.

The above-mentioned steps are illustrated in Figure 7.

2.3. Test Bench

For validating the proposal efficiency, the IM used is the 218ET3EM145TW model of 2-Hp, 3-phase at 220 VCA @ 60 Hz with a stator of 141 turns per phase, from WEG coupled to a four-quadrant dynamometer model 8540 from Lab-Volt to control the mechanical load applied. The current signals from the stator phases were acquired by current clamps model i200s from Fluke, connected to a data acquisition system (DAS) model NI-USB-6211 from National Instruments. The DAS was connected and communicated to a personal computer (PC) to store the acquired signals. Also, a standard starter was installed to start and stop the motor. Figure 8 depicts the test bench used in this work.

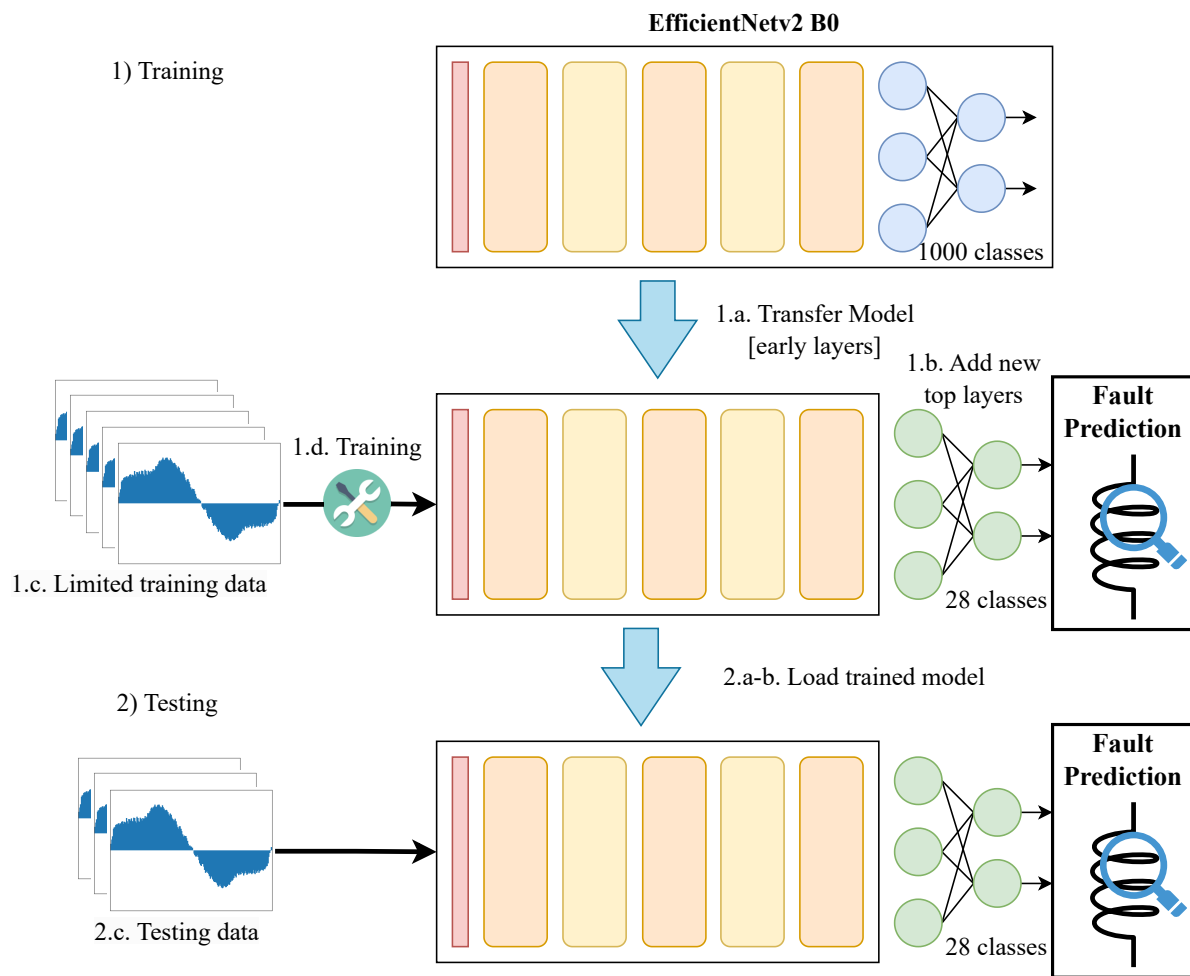


Figure 7. Block diagram of the detection process.

The stator motor was modified to create specific fault conditions in the IM, generating six ITSC fault conditions plus the healthy condition. The conditions with ITSC faults are 0, 5, 10, 15, 20, 30, and 40 short-circuited turns, with 0 SCTs being the healthy condition. The mechanical loads were simulated as fixed loads in 0%, 33.33%, 66.66%, and 100%, in concordance with the motor's nominal power. These values were chosen to maintain homogeneity across the entire range of the motor's nominal power. Regarding the above, 28 conditions were created: seven ITSC conditions at four different levels of mechanical load each. The current stator signals were acquired at a 6 kHz sampling rate in a steady state, and the motor was run at intervals of 3.5 s to avoid severe damage to the stator. Then, each acquired signal was segmented into seven parts to obtain seven testing signals of 0.5 s duration as depicted in Figure 9. All tests accumulated 1960 s and 11,760,000 signal points. It is important to say that the test bench's configuration was prepared to be fed with the commercial electrical power grid in order to verify the feasibility of applying the proposed methodology under typical operating conditions.

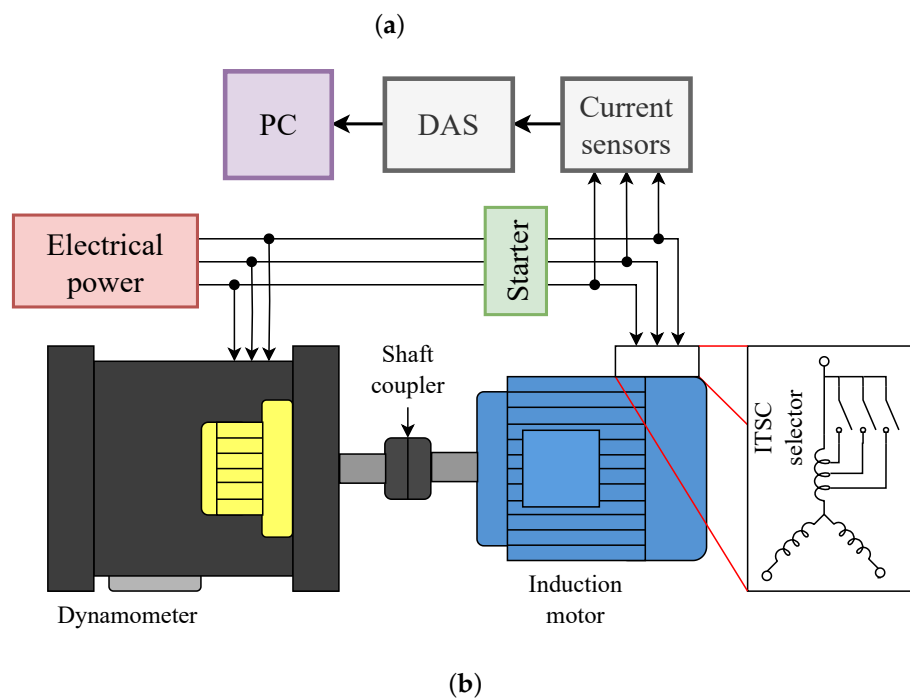
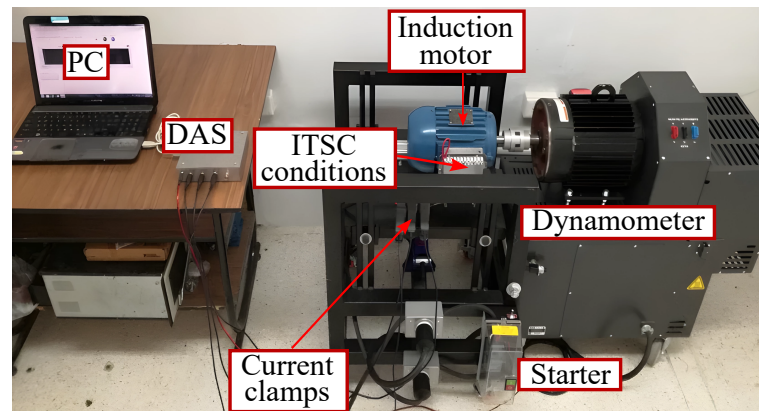


Figure 8. Test bench used for testing development: (a) real station and (b) blocks diagram.

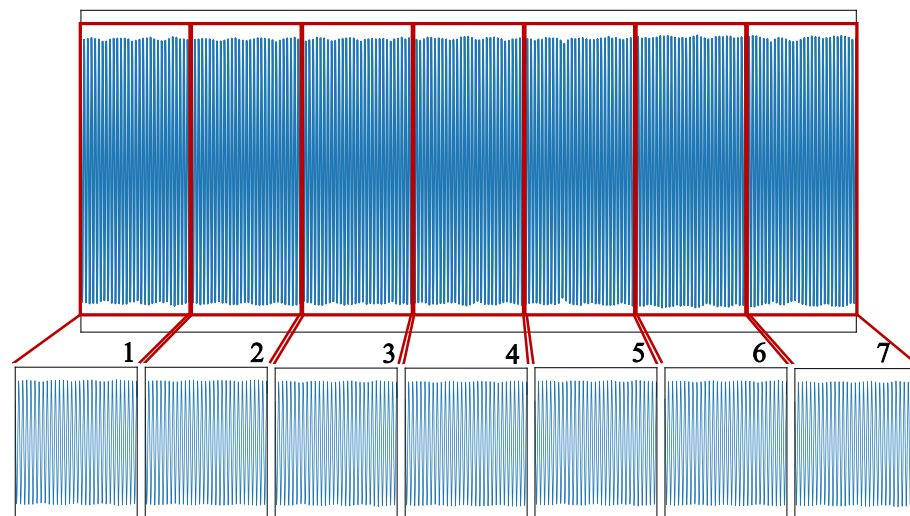


Figure 9. Example of an acquired signal segmentation (3.5 s) to obtain seven segments of testing signals (0.5 s).

3. Tests and Results

The current stator signals were acquired and pre-processed according to the earlier methodology. In this sense, a data set was created from the data generated during the acquisition stage. So, each fault condition was integrated by 140 signals of 0.5 s duration, which is enough acquisition time to detect the fault following preliminary testings. All signals were acquired from a motor fed by the commercial electrical power grid, and the application of fixed mechanical load levels and damage conditions was applied as specified in Section 2.3. This entails a data set of 3920 signal segments, converted to the modified CDFs with 200 classes and transformed into images of $3 \times 224 \times 224$ pixels. The dimensions of the images were selected according to the input format of the selected CNN model to maintain its simplicity. In addition, the primary purpose of this representation is to reduce the number of points of the signals from 3000 to 200, comprising and preserving the information on the status of the stator motor, as well as less complex CNN applications.

Per each condition fault, 100 images were selected for the training process and 40 for the testing process; in other words, 71.4% of the data set (2800 images) was selected for training, a meager amount of data in contrast to the traditional training process in DL. Furthermore, the 5-fold cross-validation was applied [36] to avoid bias and explore the methodology's effectiveness. This cross-validation consists of the random sample selection to be part of the training set, and the remaining is destined for the testing set. In this work, this process was repeated five times. Figure 10 graphically resumes the essence of this validation technique.

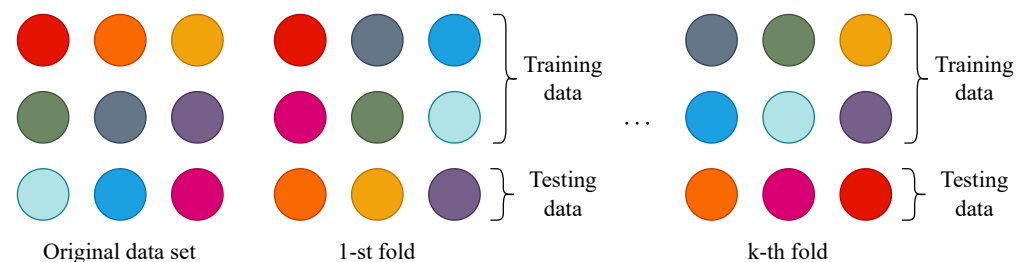


Figure 10. Graphical k-fold cross-validation process.

On another note, the necessity of using a CNN relies on the requirements of incipient fault detection. For example, this paper's most challenging fault conditions are the 0, 5, 10, and 15 SCTs. Traditional classification often confuses these faults, especially at low or no mechanical loads, due to the difficulty of identifying the small features and differences related to them [19]. Figure 11 depicts an example with ten modified CDFs per the mentioned damage level. As can be shown, these lack remarkable differences. However, slight differences exist between positive and negative lobes, generally. This makes the identification possible.

In addition, the conversion to modified CDFs increases the differences between damage conditions over the time-domain signals. In Figure 12 note that time-domain signals highlight their amplitude, which can be easily mistaken with other damage conditions. However, modified CDFs show sensitivity lobes to the change in the damage and load conditions. In this way, the modified CDFs are preferred for fault detection in this work.

The implementation of the CNN models is described below. Due to its fast training process and relatively small architecture, the EfficientNetv2 family model, specifically the most miniature model, the B0 version, was selected. This model has only six convolutional blocks and can be implemented with pre-trained ImageNet classification weights. As a result, the top layers must be replaced because, initially, the model could classify 1000 classes, and the purpose of the proposed methodology is to classify 28 classes only. In the case of the inputs, these were conserved as initially proposed by their authors: $3 \times 224 \times 224$, to be compatible with RGB images. Finally, the model was constructed by using a similar top

layers type, following the new number of classes in a categorical format and including a softmax activation in the last layer. The architecture summary is shown in Table 1.

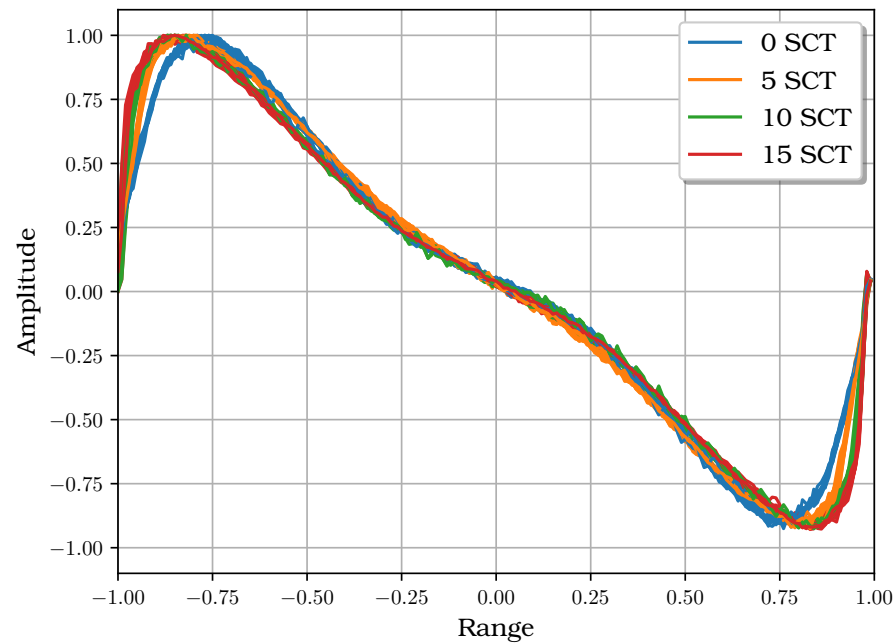
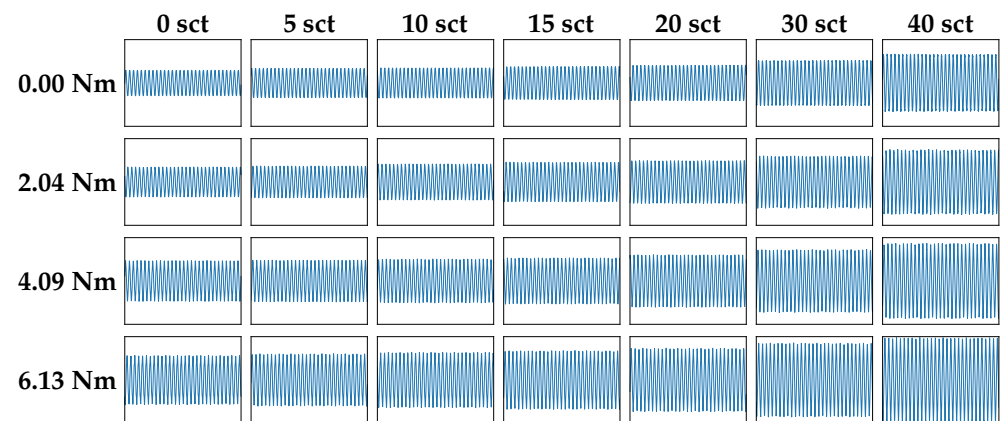
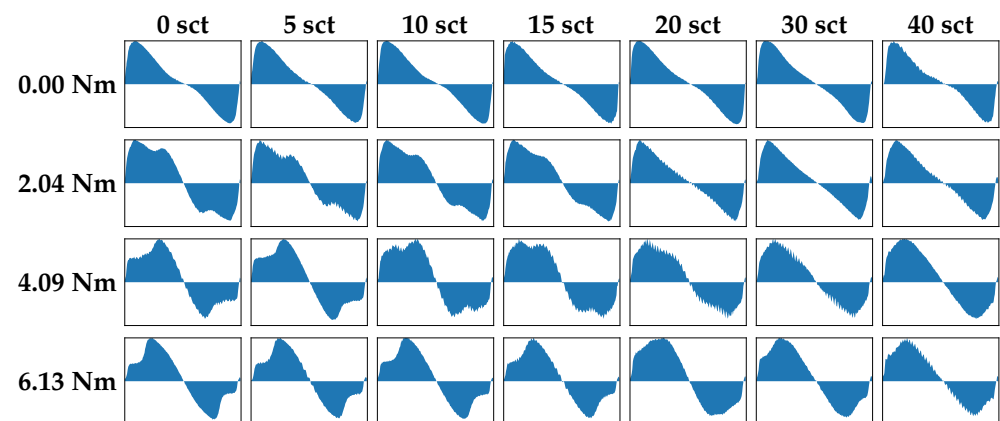


Figure 11. Example of modified CDFs with no load at 0, 5, 10, and 15 SCT.



(a)



(b)

Figure 12. Examples of differences in signals due to the damage and load conditions: (a) time-domain; and (b) modified CDFs.

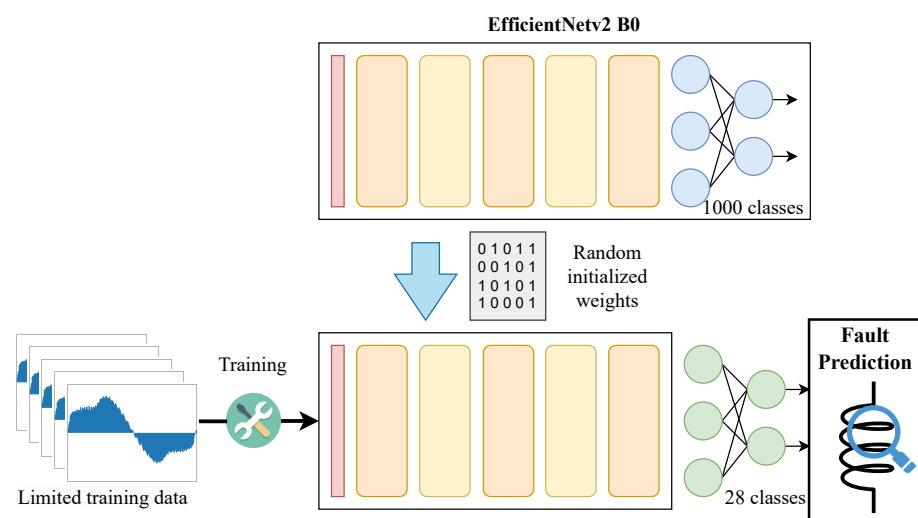
Table 1. Summary of implemented CNN architecture.

Layer	Output Shape	No. of Parameters	Basic Configuration
Input	(224, 224, 3)	0	—
EfficientNetv2-B0	(7, 7, 1280)	5,919,312	No top layers
Global Average Pooling 2D	(1280)	0	—
Dropout	(1280)	0	rate = 0.2
Dense	(28)	35,868	units = 28, activation = softmax

Three testing scenarios were established for the CNN effectiveness evaluation, described below. It is important to note that the CNN model only needs the RGB image and does not receive additional information, such as mechanical load.

3.1. Training with Randomly Initialized Weights

Implementing the architecture mentioned above, the training process was accomplished with a configuration of 30 epochs and a batch size of 5. The batch size was selected to minimize memory use and save the accuracy and speed of the process. In addition, the compiled model was developed with the Adamax optimizer, with categorical cross-entropy as the loss, and accuracy as the metric. All the trainable weights of the architecture were initialized randomly. Figure 13 shows the present training process. Note that it transfers only the convolutional blocks, and the replacement of the top layer is to be compatible with the number of classes in this work.

**Figure 13.** Training process with randomly initialized weights.

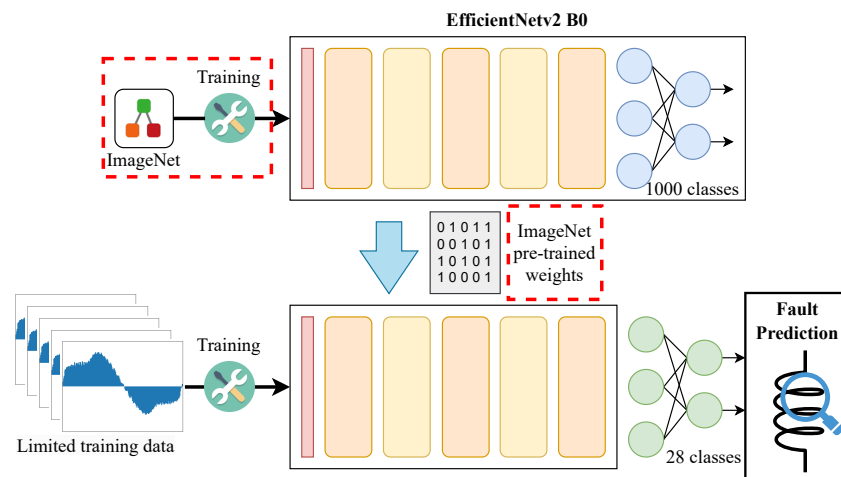
The results presented in Table 2 demonstrate the high accuracy of the method, with the worst accuracy rate of 97.50% and the best at 98.21% and an average of 97.95%. This means that in the worst case, only 28 of 1120 images were wrongly classified, and in the best case, 20 images. This behavior is due mainly to the complex detection of the incipient's damages, as exposed above. Also, the presented results emphasize the fast convergence of the CNN model since it only takes 30 epochs to reach the presented rates, a feature highlighted in this methodology and reported by their authors.

Table 2. Resume of results of training with randomly initialized weights.

Cross-Validation	Accuracy (%)		Loss (%)	
	Train	Test	Train	Test
1	99.39	98.21	2.147×10^{-2}	8.228×10^{-2}
2	99.54	97.77	1.940×10^{-2}	1.069×10^{-1}
3	99.46	97.50	2.012×10^{-2}	1.178×10^{-1}
4	99.64	98.04	1.604×10^{-2}	6.515×10^{-2}
5	99.43	98.21	1.865×10^{-2}	6.679×10^{-2}
Average	99.49	97.95	1.914×10^{-2}	8.778×10^{-2}

3.2. Training with Pre-Trained Weights

This test's launch was conducted conclusively with the same configuration previously exposed; the difference relies on loading the ImageNet pre-trained weights to the selected CNN model (red dashed rectangles in Figure 14) as their authors recommend. In this manner, the information extracted earlier is readjusted, and the model can converge quickly and improve its performance. So, the model is compiled with the same number of epochs, batch size, optimizer, etc., to compare the rates it will reach. Results are resumed in Table 3.

**Figure 14.** Training process with ImageNet pre-trained weights.**Table 3.** Resume of results with ImageNet pre-trained weights.

Cross-Validation	Accuracy (%)		Loss (%)	
	Train	Test	Train	Test
1	99.86	98.75	5.644×10^{-2}	4.588×10^{-2}
2	99.82	98.75	7.189×10^{-2}	5.934×10^{-2}
3	99.82	97.67	6.737×10^{-2}	10.960×10^{-2}
4	99.89	98.93	6.778×10^{-2}	4.837×10^{-2}
5	99.89	99.16	3.849×10^{-2}	2.998×10^{-2}
Average	99.86	98.65	6.039×10^{-3}	5.863×10^{-2}

As expected, this process enhanced the training process, improving the accuracy rate by 0.7% over the last average and covering up to 99.16% in the best of cases, failing nine images in the correct class.

3.3. Transfer Learning and Fine-Tuning

TL is a powerful technique used in CNN as an alternative when access to a data set with many samples is unavailable. This technique uses pre-trained CNN, and the information learned will be applied to new applications. This implies transferring the

architecture without the top layer and replacing it with another in concordance with the application. The development of TL in this work used the application of the typical workflow, described as follows.

1. A new architecture implementation shown in Table 1 is performed and ensures the loading of pre-trained weights. The transferred layer is frozen to avoid re-adjusting their weights in the new training process. Therefore, only the weights of the top layer will be trained.
2. Then, the model compilation is performed with the configuration applied in the latest procedures, ensuring the base model is instantiated in inference mode.
3. So, the training process is run with 20 epochs and a batch size of 5.

Continuing development of the FT technique is recommended if the scores obtained are not as high as expected. In the case of this work, the score obtained by TL is about 90%, so the FT is applied, following the next steps.

1. Unfreeze the layers of the base model.
2. Recompile the model as performed previously and set a very low learning rate to ensure the weights are updated slowly and lightly (fine-tuning). In this work, a rate of 1×10^{-5} was configured.
3. Run the re-training process. The configuration of the training process for fine-tuning is 20 epochs and a batch size of 5.

A simplified schematic of this process is depicted in Figure 15.

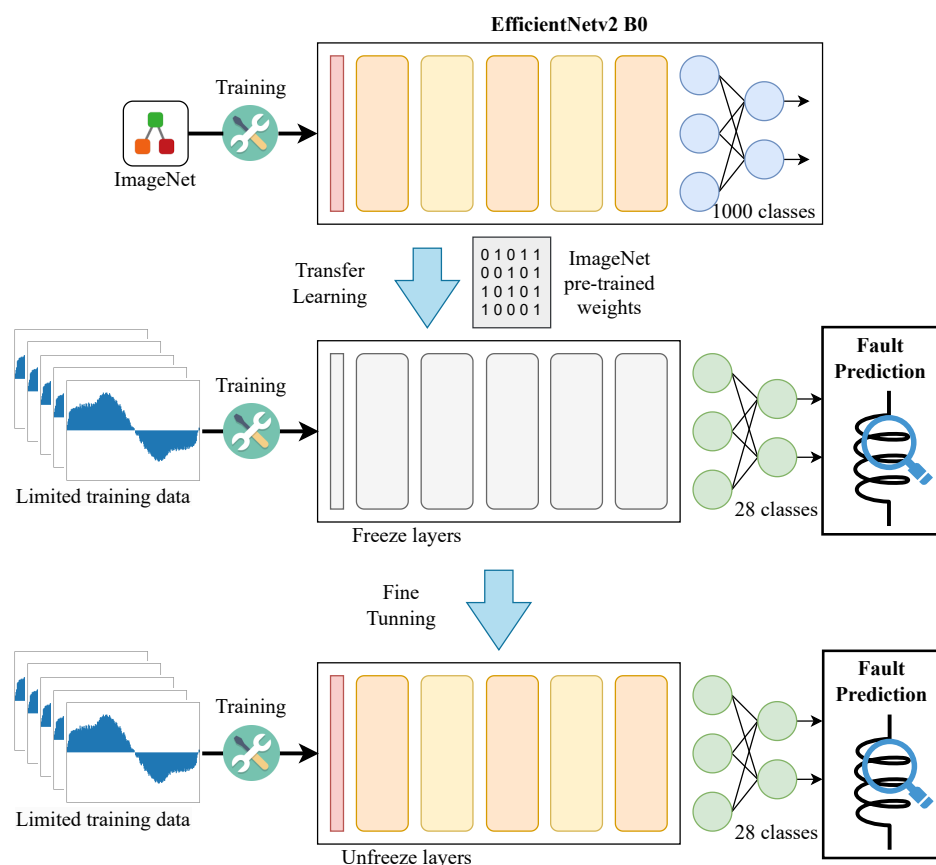


Figure 15. Training process with transfer learning and fine-tuning application.

The results obtained from the TL and FT development have scant differences (see Table 4). While it appears to be the lowest testing rate, the accuracies differ by less than 2%. So, this approach could be an alternative because one of the main assumptions of this technique is fast convergence. Table 5 presents the average times of each training process.

Table 4. Resume of results with transfer learning and fine-tuning technique.

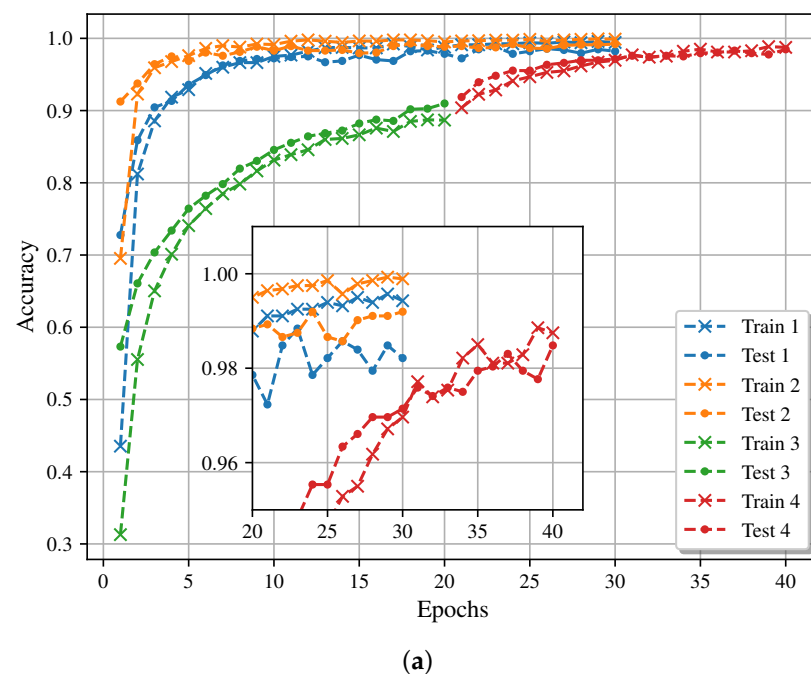
Cross-Validation	Accuracy (%)		Loss (%)	
	Train	Test	Train	Test
1	98.75	98.48	5.275×10^{-2}	5.454×10^{-2}
2	98.82	96.87	4.428×10^{-2}	8.990×10^{-2}
3	99.11	97.50	4.195×10^{-2}	11.757×10^{-2}
4	98.82	97.32	4.869×10^{-2}	7.838×10^{-2}
5	98.75	97.14	4.856×10^{-2}	8.546×10^{-2}
Average	98.85	97.46	4.725×10^{-3}	8.517×10^{-2}

Table 5. Average times of training approaches.

Approach	Average Time (s)
Randomly initialized weights	1379
ImageNet pre-trained weights	1292
TL and FT	316 and 728

On the other hand, the testing results ensure that the detection process takes 0.107 s (including signal-to-image conversion) once the signal is acquired (0.5 s). Thus, the method can provide two fault diagnoses per second or one diagnosis in 0.607 s. It is important to note that the training process is developed once and generally accomplished offline, allowing the detection process time to remain unaffected.

Figure 16 shows the accuracy and loss results of the CNN training process with the different applied training approaches. Note the fast convergence and the high accuracy and low loss rate reached. Also, the TL and FT techniques use more epochs, but these are executed faster than the others.

**Figure 16.** Cont.

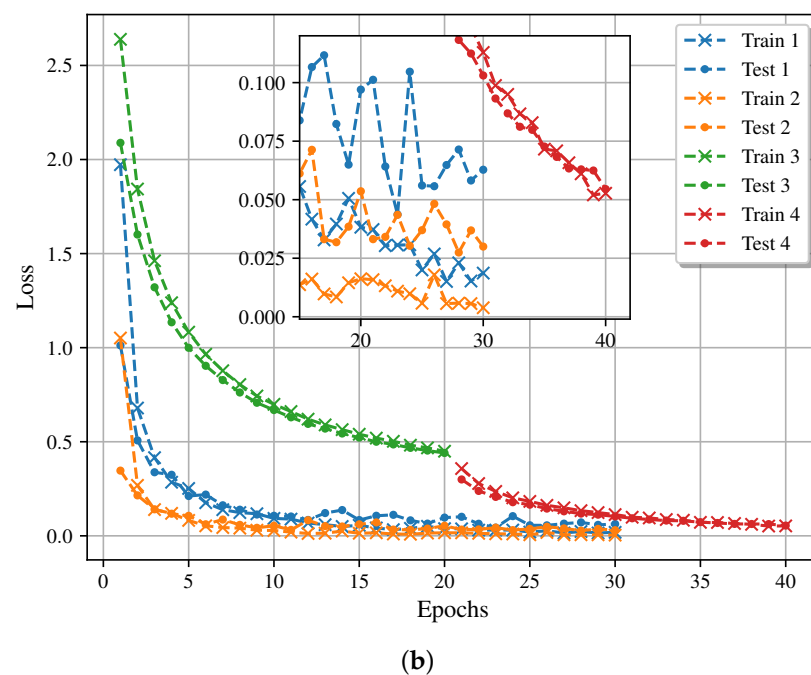


Figure 16. Comparative plots of applied training approaches: (a) accuracy; and (b) loss. [1: weights randomly initialized; 2: ImageNet pre-trained weights; 3: TL; and 4: FT].

In addition, testing using 25, 50, and 75 images per condition (700, 1400, and 2100 images in total, respectively) for the training process was developed, as Section 3.2 specifies, to explore the effectiveness with smaller data set applications. The final accuracy of the tests was 94.73, 97.05, and 98.39%, respectively. Compared to the best result with 100 images per condition of 99.16% (see Table 3), the higher the data set, the higher the accuracy rate. These comparisons are depicted in Figure 17, where the high compatibility of the selected CNN model and the modified CDFs and the high effectiveness of the model with smaller data sets are also demonstrated since a high accuracy rate can be obtained (over 90%) even in a data set with a significant lack of samples.

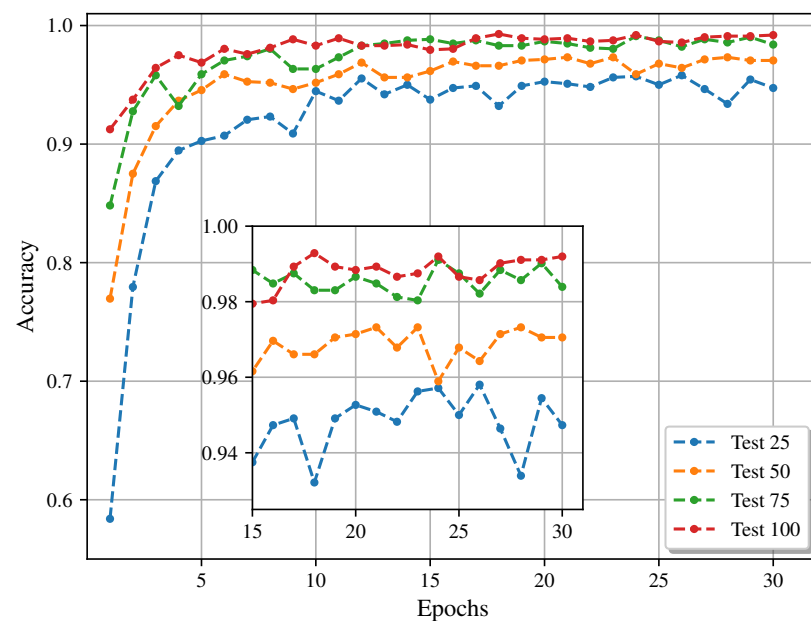


Figure 17. Comparative plots of accuracy behavior of the EfficientNetv2 B0 during the training process, with different sizes of data sets.

The tests were developed in Python version 3.11, using Tensorflow version 2.15 and the Keras version 2.10 library. The used hardware was a Dell G15 with an Intel Core i7-13650HX, 16 GB of RAM, and an NVIDIA GeForce RTX 4060 GPU board with 8 GB of dedicated RAM. The configuration of TensorFlow and Keras for GPU hardware usage was accomplished according to the published procedure on the TensorFlow web page [37].

4. Discussion

The results of implementing the proposed methodology indicate that it successfully detected the 28 ITSC conditions, regardless of the applied mechanical load. Thus, the reached accuracy rate of 98.75–99.16% and an average of 98.65% proves the effectiveness of the methodology for detecting incipient damage up to 5 SCT as minimal, damage that represents 3.5% of the total turns in the motor stator. As planned, loading the ImageNet pre-trained weights shows the best results, one of the most highlighted model features reported in the literature [34]. Conversely, the training process with weights randomly initialized retrieved a high accuracy rate of 97.50–98.27% and an average of 97.95%, much better than expected. This process typically uses a high amount of data and epochs to reach a high accuracy rate, contrasting with the model used in this work under previously expressed conditions. Also, TL and FT promised great potential in this application; concerning the development, this approach was a better fit for this application. However, it gave the lowest rates of 97.14–98.48% and an average of 97.46% in this work. Despite that, the results are comparable with those previously discussed, and it is an alternative to small data sets. In addition, Figure 16 presents an exceptional performance of the CNN model. However, the graph suggests stopping the training at 20 epochs since the changes are insignificant (less than 1%) for the two first approaches. On another note, the TL and FT techniques could improve the results if the epochs number is increased; however, this can be unproductive since reaching overfitting is possible.

Table 6 presents a comparative analysis of the results obtained by the proposed method alongside the latest research utilizing similar techniques presented in the literature for detecting ITSCs in IM. Specifically, it details the techniques or methods employed in each study, the mechanical load conditions, fault severity levels, and the efficacy reached in assessing the IM condition. According to this table, it is possible to note that the ANNs at different levels of complexity are applied to the incipient ITSC fault diagnosis. While other techniques present effectiveness in lower levels of ITSC damage with diverse levels of mechanical load [10,24,25] or under a single or no mechanical load [7,11,23,26], they apply elaborated preprocessing stages, which increase their computational complexity and could limit their implementation in an industrial process. In this regard, for example, Guedidi et al. [25] investigate a SqueezeNet model, a lightweight CNN network, for detecting ITSCs in an IM, reaching high accuracy rate. Their proposal relies on the use of several signal processing techniques (e.g., Hilbert transform, variational mode decomposition, and correlation) prior to introducing the identified features to SqueezeNet, thereby increasing its computational resources. In addition, a more significant amount of data is used for the training process in all works of Table 6. In this way, the proposed methodology reaches accuracies that can compete and, in some cases, improve that presented in the literature, being that its main features or advantages are (1) a low computational complexity because it only uses a preprocessing stage and (2) that it can be employed regardless of the mechanical load since the lightweight CNN is trained from different load levels and does not need additional information or adjustments to accomplish the diagnosis. Although promising results have been achieved by the proposal, further research is necessary in order to enhance its robustness. In this regard, it is important to continue exploring other case studies that address various scenarios: (1) a wider range of faults, e.g., bearing damage, broken rotor bars, among others, (2) time-varying regimens, i.e., those produced by a frequency variator, and perturbations in the electric network or noise, (3) a greater variety of load levels and dynamic profiles of mechanical load, and (4) exploring other lightweight CNN networks, such as WearNet, ShuffleNet, RasNet, among others [38]. Calibrating the proposal under

these circumstances will enable the investigation of a wider range of scenarios. At present, the performance of the proposed method may be compromised in such situations.

Table 6. Comparison of the proposed methodology with similar techniques in the literature.

Work	Technique	Mechanical Load (%)	Severities	Accuracy (%)
[7]	(a) Statistical features from a quaternion analysis (b) Decision tree	No load	0, 6, 12, 18, and 24 SCTs	99
[10]	(a) Discrete Wavelet Coefficients (b) L1 and L2 norms (c) ANN	0, 25, 50, 75, 100, and 125	0, 1, 3, 5, 7, 10, 13, 15, 17, 20, 23, 25, 35, 45, 55, 65, 80, and 90 SCTs	95.29
[11]	(a) Empirical wavelet transform (b) statistical features (c) Linear discriminant analysis (d) Self-Organizing Neural Network	20	0, 2, 4, and 6 SCTs	100
[23]	(a) Negative sequence current analysis (b) Probabilistic Neural Network	No specified	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10 SCTs	90
[24]	(a) Statistical features from time-based from torque (b) Magnitude of spectra estimation (c) Multi-layer feed-forward neural network	0, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100	0, 1.9, 3.9, 7.9, 11.9, 15.8, 19.8, 23.8, and 27.7%	88–99
[25]	(a) Hilbert transform (b) Variational mode decomposition (c) Correlation (d) SqueezeNet model and TL	0, 30, and 50	5, 7, 13, 15, and 20 SCTs	99.8
[26]	(a) Empirical mode decomposition (b) Statistical features (c) Long Short-Term Memory.	No specified	0, 1, 2, 5, 7, and 15 SCTs	95
This work	(a) CDFs obtention from current signals (b) EfficientNetv2-B0 model CNN	0, 33.33, 66.66, and 100	0, 5, 10, 15, 20, 30, and 40 SCTs	98.75–99.16

5. Conclusions

This paper presented a novel methodology for ITSC fault detection in IM based on DL and the development of a training process with a small data set. The CDF of stator current signals was obtained, and the typical linear component was subtracted to highlight the fault signatures. So, these modified CDFs were converted to images to apply DL techniques using the EfficientNetv2-B0 model, whose compact architecture and fast training process are highlighted, reaching an accuracy rate of up to 99.16%. Also, a re-training process was developed with barely 2800 images for 28 fault conditions with barely 30 epochs. This allowed the effective detection of the fault up to 5 SCTs regardless of the applied mechanical load. That way, it is demonstrated the detection of incipient ITSC with modified CDFs and the versatility of the followed workflow, solving the problem of limited data sets, an important constraint in this area. This opens the possibility of expanding this methodology to other areas of fault detection in IMs.

The work will expand to detect minor faults over 5 SCTs in the future, applying TL and FT to explore the technique's performance, including additional operating conditions and motor configurations. Also, combined fault detection will be explored. These goals will contribute to developing more robust methods that can be integrated into online and reliable fault tolerant-control strategies.

Author Contributions: Conceptualization and methodology, C.J.M.-P., J.P.A.-S. and M.V.-R.; software, formal analysis, resources, and data curation, C.J.M.-P. and L.P.-E.; writing—review and editing, all authors; supervision, project administration, and funding acquisition, J.P.A.-S., J.d.J.R.-M., M.V.-R. and D.G.-L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The author, C.J. Morales-Perez, thanks the “Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT)—México” for supporting a postdoctoral stay at the “Universidad Autonoma de Queretaro (UAQ, México)”. The authors would like to thank the (CONAHCYT)—México and the “Sistema Nacional de Investigadoras e Investigadores (SNII)—CONAHCYT—México” for their support in this research.

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

References

- de Souza, D.F.; Salotti, F.A.M.; Sauer, I.L.; Tatizawa, H.; de Almeida, A.T.; Kanashiro, A.G. A Performance Evaluation of Three-Phase Induction Electric Motors between 1945 and 2020. *Energies* **2022**, *15*, 2002. [\[CrossRef\]](#)
- Gonzalez-Abreu, A.D.; Osornio-Rios, R.A.; Jaen-Cuellar, A.Y.; Delgado-Prieto, M.; Antonino-Daviu, J.A.; Karlis, A. Advances in Power Quality Analysis Techniques for Electrical Machines and Drives: A Review. *Energies* **2022**, *15*, 1909. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Jaen-Cuellar, A.Y.; Elvira-Ortiz, D.A.; Saucedo-Dorantes, J.J. Statistical Machine Learning Strategy and Data Fusion for Detecting Incipient ITSC Faults in IM. *Machines* **2023**, *11*, 720. [\[CrossRef\]](#)
- Niu, G.; Dong, X.; Chen, Y. Motor Fault Diagnostics Based on Current Signatures: A Review. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 3520919. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Cardenas-Cornejo, J.J.; Ibarra-Manzano, M.A.; González-Parada, A.; Castro-Sanchez, R.; Almanza-Ojeda, D.L. Classification of inter-turn short-circuit faults in induction motors based on quaternion analysis. *Measurement* **2023**, *222*, 113680. [\[CrossRef\]](#)
- Akhil Vinayak, B.; Anjali Anand, K.; Jagadanand, G. Wavelet-based real-time stator fault detection of inverter-fed induction motor. *IET Electr. Power Appl.* **2020**, *14*, 82–90. [\[CrossRef\]](#)
- Almounajjed, A.; Sahoo, A.K.; Kumar, M.K. Diagnosis of stator fault severity in induction motor based on discrete wavelet analysis. *Measurement* **2021**, *182*, 109780. [\[CrossRef\]](#)
- Almounajjed, A.; Sahoo, A.K.; Kumar, M.K.; Subudhi, S.K. Stator Fault Diagnosis of Induction Motor Based on Discrete Wavelet Analysis and Neural Network Technique. *Chin. J. Electr. Eng.* **2023**, *9*, 142–157. [\[CrossRef\]](#)
- Saucedo-Dorantes, J.J.; Jaen-Cuellar, A.Y.; Perez-Cruz, A.; Elvira-Ortiz, D.A. Detection of Inter-Turn Short Circuits in Induction Motors under the Start-Up Transient by Means of an Empirical Wavelet Transform and Self-Organizing Map. *Machines* **2023**, *11*, 958. [\[CrossRef\]](#)
- Hussain, M.; Kumar, D.; Kalwar, I.H.; Memon, T.D.; Memon, Z.A.; Nisar, K.; Chowdhry, B.S. Stator Winding Fault Detection and Classification in Three-Phase Induction Motor. *Intell. Autom. Soft Comput.* **2021**, *29*, 869–883. [\[CrossRef\]](#)
- Ghanbari, T.; Mehraban, A.; Farjah, E. Inter-turn fault detection of induction motors using a method based on spectrogram of motor currents. *Measurement* **2022**, *205*, 112180. [\[CrossRef\]](#)
- Wu, Y.C.; Feng, J.W. Development and Application of Artificial Neural Network. *Wirel. Pers. Commun.* **2018**, *102*, 1645–1656. [\[CrossRef\]](#)
- Angelov, P.P.; Soares, E.A.; Jiang, R.; Arnold, N.I.; Atkinson, P.M. Explainable artificial intelligence: An analytical review. *Wires Data Min. Knowl. Discov.* **2021**, *11*, e1424. [\[CrossRef\]](#)
- Zhang, C.; Lu, Y. Study on artificial intelligence: The state of the art and future prospects. *J. Ind. Inf. Integr.* **2021**, *23*, 100224. [\[CrossRef\]](#)
- Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Umar, A.M.; Linus, O.U.; Arshad, H.; Kazaure, A.A.; Gana, U.; Kiru, M.U. Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access* **2019**, *7*, 158820–158846. [\[CrossRef\]](#)
- Patil, T.; Pandey, S.; Visrani, K. A Review on Basic Deep Learning Technologies and Applications. In *Data Science and Intelligent Applications*; Kotecha, K., Piuri, V., Shah, H.N., Patel, R., Eds.; Springer: Singapore, 2021; pp. 565–573. [\[CrossRef\]](#)
- Liu, R.; Yang, B.; Zio, E.; Chen, X. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mech. Syst. Signal Processing* **2018**, *108*, 33–47. [\[CrossRef\]](#)

20. 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. [\[CrossRef\]](#)
21. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [\[CrossRef\]](#)
22. Skowron, M.; Wolkiewicz, M.; Orłowska-Kowalska, T.; Kowalski, C.T. Application of Self-Organizing Neural Networks to Electrical Fault Classification in Induction Motors. *Appl. Sci.* **2019**, *9*, 616. [\[CrossRef\]](#)
23. Babanezhad, H.; Yaghobi, H.; Hamidi, M. Stator Turn-to-Turn Fault Estimation of Induction Motor by Using Probabilistic Neural Network. *Modeling Simul. Electr. Electron. Eng.* **2021**, *1*, 35–40. [\[CrossRef\]](#)
24. Maraaba, L.; Al-Hamouz, Z.; Abido, M. An Efficient Stator Inter-Turn Fault Diagnosis Tool for Induction Motors. *Energies* **2018**, *11*, 653. [\[CrossRef\]](#)
25. Guedidi, A.; Laala, W.; Guettaf, A.; Arif, A. Early detection and localization of stator inter turn faults based on variational mode decomposition and deep learning in induction motor. *Diagnostyka* **2023**, *24*, 2023401. [\[CrossRef\]](#)
26. Alipoor, G.; Mirbagheri, S.J.; Moosavi, S.M.M.; Cruz, S.M.A. Incipient detection of stator inter-turn short-circuit faults in a Doubly-Fed Induction Generator using deep learning. *IET Electr. Power Appl.* **2023**, *17*, 256–267. [\[CrossRef\]](#)
27. Saufi, S.R.; Ahmad, Z.A.B.; Leong, M.S.; Lim, M.H. Challenges and Opportunities of Deep Learning Models for Machinery Fault Detection and Diagnosis: A Review. *IEEE Access* **2019**, *7*, 122644–122662. [\[CrossRef\]](#)
28. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In *Artificial Neural Networks and Machine Learning—ICANN 2018*; Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I., Eds.; Springer: Cham, Switzerland, 2018; pp. 270–279. [\[CrossRef\]](#)
29. Shao, S.; McAleer, S.; Yan, R.; Baldi, P. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2446–2455. [\[CrossRef\]](#)
30. Zorig, A.; Hedayati Kia, S.; Chouder, A.; Rabhi, A. A comparative study for stator winding inter-turn short-circuit fault detection based on harmonic analysis of induction machine signatures. *Math. Comput. Simul.* **2022**, *196*, 273–288. [\[CrossRef\]](#)
31. Sadeghi, R.; Samet, H.; Ghanbari, T. Detection of Stator Short-Circuit Faults in Induction Motors Using the Concept of Instantaneous Frequency. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4506–4515. [\[CrossRef\]](#)
32. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **1995**, *3361*, 1995.
33. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. *arXiv* **2021**, arXiv:2104.00298. [\[CrossRef\]](#)
35. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2020**, arXiv:1905.11946. [\[CrossRef\]](#)
36. Marcot, B.G.; Hanea, A.M. What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? *Comput. Stat.* **2021**, *36*, 2009–2031. [\[CrossRef\]](#)
37. TensorFlow. Build from Source on Windows. 2014. Available online: https://www.tensorflow.org/install/source_windows (accessed on 4 May 2024).
38. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [\[CrossRef\]](#)

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.