Holistic fault detection and diagnosis system in imbalanced, scarce, multi-domain (ISMD) data setting for component-level prognostics and health management (PHM).

ROHAN, A.

2022

mathematics

MDPI

Article

Holistic Fault Detection and Diagnosis System in Imbalanced, Scarce, Multi-Domain (ISMD) Data Setting for Component-Level Prognostics and Health Management (PHM)

Ali Rohan [1,2]

[1] Department of Mechanical, Robotics, and Energy Engineering, Dongguk University, 30 Pildong 1 Gil, Jung-gu, Seoul 04620, Korea; ali_rohan2003@hotmail.com or alirohan@dongguk.edu or ali.rohan@nottingham.ac.uk
[2] Faculty of Medicine and Health Sciences, University of Nottingham, Sutton Bonington, Loughborough LE12 5RD, UK

Abstract: In the current Industry 4.0 revolution, prognostics and health management (PHM) is an emerging field of research. The difficulty of obtaining data from electromechanical systems in an industrial setting increases proportionally with the scale and accessibility of the automated industry, resulting in a less interpolated PHM system. To put it another way, the development of an accurate PHM system for each industrial system necessitates a unique dataset acquired under specified conditions. In most circumstances, obtaining this one-of-a-kind dataset is difficult, and the resulting dataset has a significant imbalance, a lack of certain useful information, and contains multi-domain knowledge. To address those issues, this paper provides a fault detection and diagnosis system that evaluates and preprocesses imbalanced, scarce, multi-domain (ISMD) data acquired from an industrial robot, utilizing signal processing (SP) techniques and deep learning-based (DL) domain knowledge transfer. The domain knowledge transfer is used to produce a synthetic dataset with a high interpolation rate that contains all the useful information about each domain. For domain knowledge transfer and data generation, continuous wavelet transform (CWT) with a generative adversarial network (GAN) was used, as well as a convolutional neural network (CNN), to test the suggested methodology using transfer learning and categorize several faults. The proposed methodology was tested on a real experimental bench that included an industrial robot created by Hyundai Robotics. This test had a satisfactory outcome with a 99.7% (highest) classification accuracy achieved by transfer learning on several CNN benchmark models.

Keywords: domain knowledge transfer; big industrial data; generative adversarial network (GAN); convolutional neural network (CNN); prognostics and health management (PHM); artificial intelligence (AI)

MSC: 68T40

Citation: Rohan, A. Holistic Fault Detection and Diagnosis System in Imbalanced, Scarce, Multi-Domain (ISMD) Data Setting for Component-Level Prognostics and Health Management (PHM). *Mathematics* **2022**, *10*, 2031. https://doi.org/10.3390/math10122031

Academic Editors: Stefano Carrino, Vicente Rodríguez Montequín, Hatem Ghorbel and Ivana Budinská

Received: 15 February 2022
Accepted: 8 June 2022
Published: 11 June 2022

1. Introduction

In contemporary industrial settings, the majority of tasks are performed by electromechanical devices, such as robots. These robots are made up of several electrical and mechanical components joined together to perform a uniquely engineered operation. Yet, such electromechanical components are vulnerable to degradation due to continued operation. Over time, proper assessment and maintenance strategies are required to prevent irreparable damage. To counter this, prognostics and health management (PHM) has evolved as an attractive method in establishing techniques for system health monitoring, diagnostics, remaining useful life (RUL) prediction, and prognostics. PHM is considered to be an effective approach to providing comprehensive, tailored solutions for health management [1]. PHM has three critical tasks: (1) *fault detection*: detection of fault trigger at the early stage of the component or system degradation; (2) *fault diagnosis*: segregation

and identification of fault and its source; and (3) *prediction*: RUL forecasting. Figure 1 shows the essential tasks of a PHM system. PHM can be applied at the component level, system level, or both. PHM at the component level directs the development of health monitoring strategies for specific components, such as electric motors, electronic devices, bearings, and gear reducers. It determines whether the health of the monitored component is time-degraded due to various environmental, operational, and performance-related parameters [2,3]. In contrast, PHM at the system level evaluates the detailed system health, factoring in system operation, design, and process-related parameters [4].

Figure 1. Basic tasks involved in a typical PHM system.

In recent years, the tremendous progress in artificial intelligence (AI) has strengthened the potential for designing PHM systems that are powerful enough to detect, diagnose, and predict faults at an earlier stage with high precision. Deep learning (DL) and machine learning (ML) have become essential tools in establishing the decision-making capabilities of a PHM system. Numerous studies have been conducted on designing and implementing such a system at the component and system levels. PHM is generally classified into either mathematics- or data-based approaches [5–7]. Awareness of the core knowledge of the component understudies, such as those relating to material characteristics and architectural attributes, supports mathematical model approaches [8,9], while data-driven approaches derive information from statistical data to forecast a component's health [10–14]. The mathematical model-based methods [15–17] require the development of a physics-based model of an element or system beforehand, from which to develop PHM strategies. Yet, they struggle with several issues, such as noisy and dynamic work environments, which affect the requisite precision needed to create the desired model. Furthermore, these models cannot be upgraded with newly recorded data in real-time.

In contrast to the model-based approaches, data-driven approaches [18,19] are becoming popular due to their ability to update and transform in real-time under different scenarios. There have also been substantial improvements in the computing capabilities of devices, with improved sensing technologies that allow efficient data acquisition. Current data-driven approaches require significant information and additional real-time measurements, such as vibration, acoustic emission, laser displacement, temperature, speed, and electrical current [20,21], to design a PHM system. Recently, researchers have suggested data-driven approaches that focus primarily on the DL-based fault diagnosis or prognosis [22,23]. In contrast, others have concentrated on the applicability of a specific item, such as a bearing or an electronic system [24–26]. Other studies have addressed alternative propositions, such as intelligent condition-based monitoring of rotating electrical machines (REMs) using a sparse auto-encoder approach [27], rolling element bearing (REB) PHM based on a deep convolutional neural network (DCNN) [28], and improved DCNN, i.e., the hierarchically adaptive DCNN [29]. CNN-based mechanical bearing fault detection [30] was introduced as a feature-learning basis for health monitoring to freely learn useful features from the data. Meanwhile, my previous study [31] introduced an ML-based fault detection and diagnostic method based on a different feature selection, extraction, and infusion process.

The success of the aforementioned approaches is highly dependent on the following factors: (1) *data availability*: if data are available, or can be acquired for specific components or systems; (2) *data type*: what type of data is known, or can be obtained, such as the vibration, acoustic emission, or electric current; (3) *data quality*: if data are recorded with precision, and they are constitutive of all the information required to analyze the features and behavior of a particular component or system; and (4) *data quantity*: if data are sufficient in quantity for analysis and the creation of an interpolated PHM system. Generally, due to

the complexities of industrial parts and processes, it is difficult to collect data that satisfy all these factors. The raw data collected are often *imbalanced*: the numbers of samples per class are not evenly distributed (sometimes with a significant volume of data for one class, alluded to as the majority class, and much fewer samples for one or two other classes, alluded to as the minority classes); *scarce*: data are not adequate to create sustainable DL models that can be applied at the production stage; *multi-domain*: the dataset is comprised of information regarding several domains, e.g., speed characteristics of a rotating electrical or mechanical component at a different level of speed. Thus, imbalanced, scarce, multi-domain (ISMD) data represent a significant bottleneck in the growth of PHM systems.

Research continues to tackle the challenges related to DL applications for PHM. Previously, in [32], the authors provide a brief introduction to several deep learning models by reviewing and analyzing fault detection, diagnosis, and prognosis applications. Yet, a challenge remains about how to resolve ISMD data, which poses a significant challenge for AI-based applications. Several proposed methods, such as data augmentation and transfer learning, have been used to address the imbalance and scarcity of data. The generative adversarial network (GAN) was frequently used for this purpose. In [33], the authors proposed balancing-GAN (BAGAN) as an augmentation method for restoring balance in imbalanced datasets, while in [34], a concept for learning the discriminative classifier from unlabeled or partly labeled data using categorical-GAN (CatGAN) was established. In another variation, a data augmentation process was proposed [35], producing artificial medical images by using GAN to classify liver lesions. Apart from these methodologies, to address current issues with traditional electromechanical system monitoring approaches used in industrial settings, some researchers focused on incremental learning and novelty detection methodologies [36,37]. These methodologies mainly focus on the development of fault detection systems in industrial settings with a lack of data for faulty conditions. By introducing a systematic approach—of feature selection, extraction, and classification—to retraining, to include new patterns to the novelty detection and fault identification models, these techniques have shown promising results. In recent studies [38,39], GAN was used for multi-domain image translation with various image syntheses. The architecture was called StarGAN. The study implemented a rigorous approach to converting images from two different domains into a single domain, without losing any useful features, thus paving the way for the introduction of domain knowledge transfer using GANs in the field of DL.

ISMD data have been a specific research challenge in creating interpolated DL models, regardless of the type of classification issue. In particular, the problem of multi-domain data has not yet been adequately investigated. Therefore, this specific work expands the ISMD data issue by focusing on multi-domain data, intending to develop a holistic fault detection and diagnosis system for component-level PHM. By adding multiple faults related to a mechanical component, the rotate vector (RV) reducer, data for several domains were recorded in real-time from an industrial robot. The recorded data were then preprocessed using signal-processing techniques such as discrete wavelet transform (DWT) and continuous wavelet transform (CWT). The power of StarGAN was used to transfer domain knowledge using image-to-image translation to generate a synthetic dataset from real data. The final dataset was then validated using transfer learning with a variety of CNN benchmark models.

Relevant literature was examined for this study, finding that such a technique has not yet found any practical application specifically for PHM. So, the prospects of using motor current signature analysis (MCSA) to detect and diagnose mechanical faults were investigated in this study. Previously, mechanical faults were detected by contrasting approaches—vibration signal analysis, acoustic emission, or ferrography analysis. Among these, the most effective method has been vibration analysis. Yet, this method faces certain challenges such as its usage requiring costly vibration sensors, which are challenging to place and install in particular areas to record vibration signals. The ambient environment also generates noise, causing erroneous sensor readings. As an improved option, MCSA has a range of benefits over vibration analysis. For instance, MCSA uses the built-in current

signal of the motor control unit and needs no extra sensors, resulting in a low-cost and less complicated framework. In addition, the current signals are not easily affected by the ambient operating conditions.

The following sections present the details of the proposed fault detection and diagnosis approach: Section 2 defines the materials and methods, including the experimental test bench, and describes the suggested technique; Section 3 presents the results and discussion; Section 4 concludes the paper.

2. Materials and Methods

2.1. Experimental Test Bench

The experimental test bench in this study, as shown in Figure 2, consisted of three main components: an industrial robot, controller, and personal computer (PC). Manufactured by Hyundai Robotics, Deagu, South Korea, the robot used was the model YS080, with a maximum payload capacity of 80 kgf. Figure 3 shows details of the robot as follows: (a) a free body diagram and (b) the Hyundai robot YS080. The robot comprises six axes or joints where an individual axis is mounted with an electric motor (those are of diverse specifications), enabling it to move freely, taking 360-degree turns about each axis.

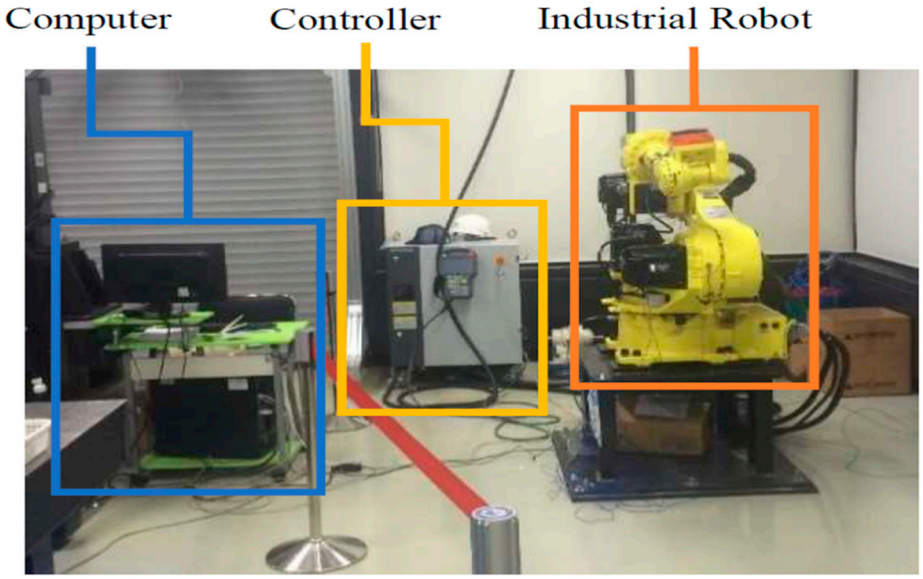

Figure 2. Experimental test bench.

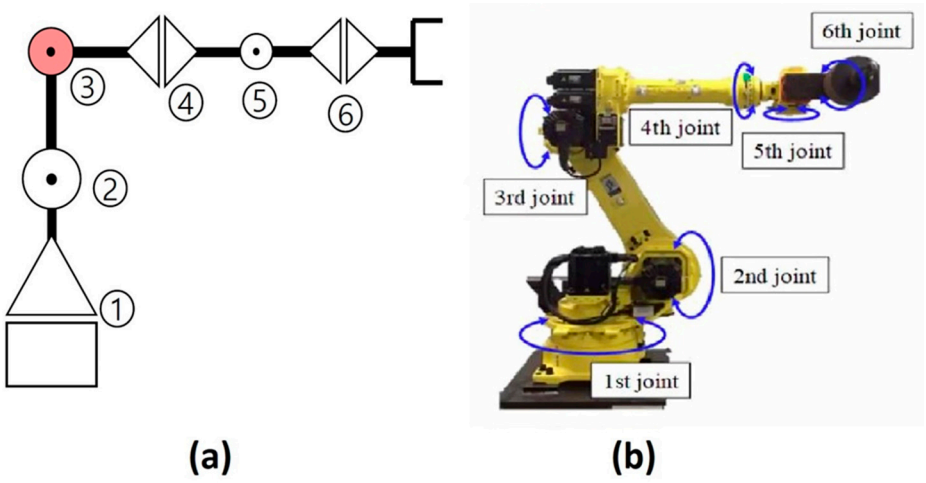

Figure 3. (**a**) Free body diagram, and (**b**) Hyundai robot YS080.

The motors are attached to reducers at each axis to increase or decrease the rotation speed. The robot is operated by sending commands to the controller through a PC, which, in turn, operates electric motors to produce a specific motion. Three-phase servo motors are used on each axis. The power of the motors is configured based on the amount of mechanical load on each axis. The first three axes are equipped with high-specification motors, whereas the others are equipped with lower specifications. Table 1 summarizes the details of the electric motors.

Table 1. Specifications of the electric motors.

Axes No.	Power (kW)	Speed (rpm)	Voltage (V)	Current (A)	Frequency (Hz)
1, 2, 3	5.9	2000	200	25.1	166
4, 5, 6	2	3000	200	11.7	250

2.2. Architecture of the Proposed Methodology

Figure 4 shows the overall basic architecture of the proposed methodology for the fault detection and diagnosis system. The proposed method is divided into three steps:

1. Data analysis;
2. Domain knowledge transfer;
3. Data splitting and classification.

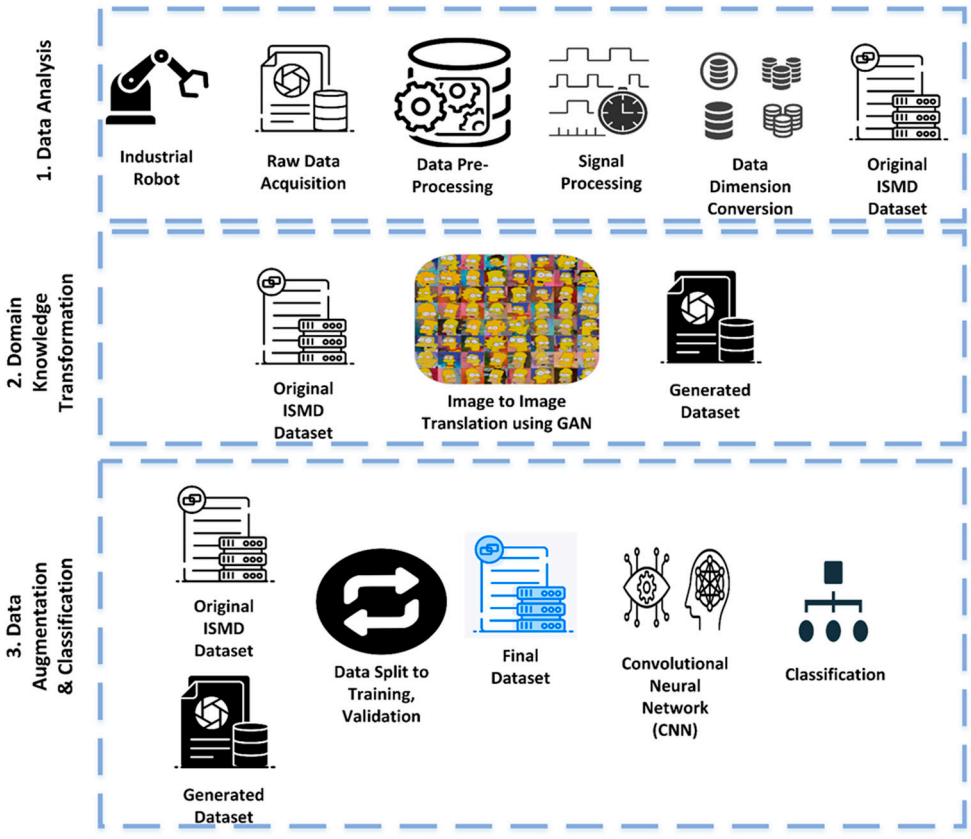

Figure 4. Basic architecture of the proposed fault detection and diagnosis system.

Several experiments were conducted on the experimental test bench shown in Figures 2 and 3 under different conditions to record the data. The recorded ISMD dataset was further used to create a multi-domain infused dataset using GAN and image-to-image translation. Finally, the generated dataset and original ISMD dataset were used, and a CNN was used to perform the classification task between several faults related to the RV reducer. The following subsection contains a detailed explanation of the proposed methodology.

2.2.1. Data Analysis

Data Acquisition

The three-phase current signal data for each axis servo motor were recorded using Hall Effect Base Linear Current Sensors WCS6800 manufactured by Winston Semiconductor Corp., Hsinchu, Taiwan. Sensors were installed at each phase of the electric motor on each axis of the robot. The current signals were recorded using a total of 18 current sensors for six motors. Figure 5 describes the data acquisition system. NI DAQ 9230 modules manufactured by National Instrument (NI), Texas, United States were used to acquire the data. This module for data recording sends the collected signal to a PC with a LabView program installed on it. The received data were processed, and a concluding archive was established containing the signal information for the three-phase current signals of each axis motor.

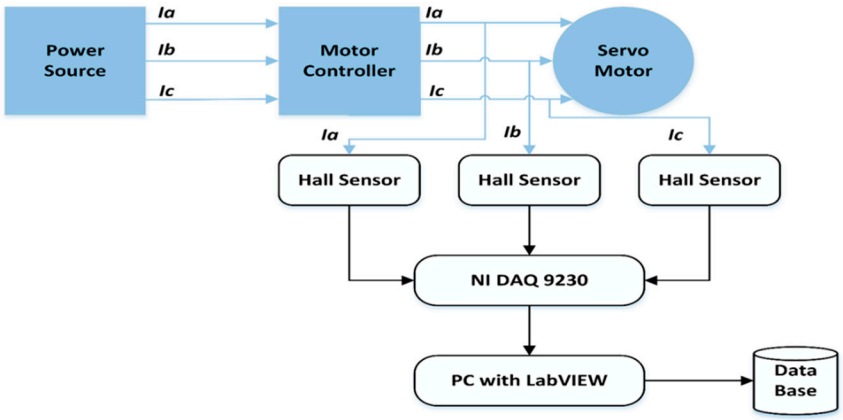

Figure 5. Block diagram of the data collection process.

The data were recorded concurrently for each motor under various fault situations. An RV reducer eccentric bearing fault was introduced into the reducer, coupled with the fourth axis motor in one case. In another case, the fault was introduced by replacing the RV reducer with a degraded one. The data were recorded for a total of three classes: normal, faulty (RV reducer with eccentric bearing fault), and faulty age (RV reducer with aging fault). Figure 6 pinpoints the positions of the faults in the Hyundai Robot with a comprehensive logical view offered. Figure 7 depicts the fault modes using an example of a fault specimen. For several cycles, the robot was operated to move freely around the axis of rotation.

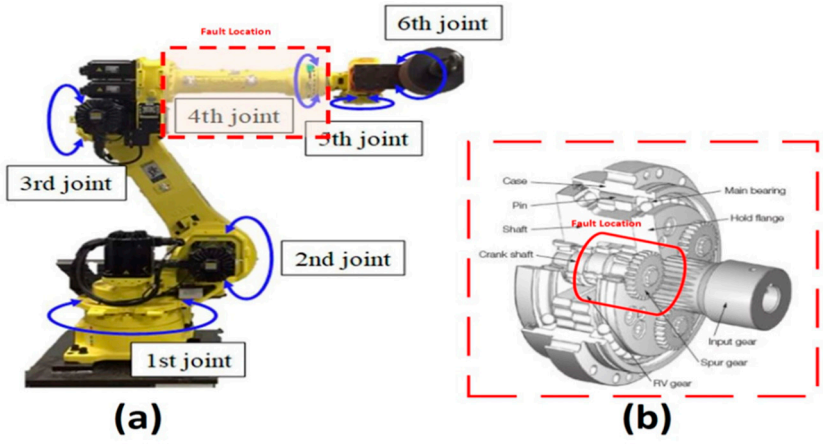

Figure 6. (**a**) Location of the fault, and (**b**) comprehensive logical view.

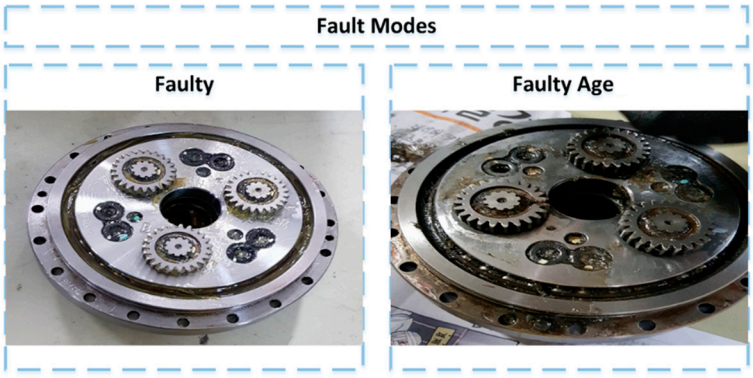

Figure 7. Example of a faulty specimen and fault types.

Data-driven methods typically involve a huge volume of data and more samples, which is the optimal case. However, in this case, the data were recorded for 10 cycles for each axis to generate ISMD data. Each cycle corresponded to the completion of motion along a particular axis. To introduce scarcity into the dataset, 10 cycles were chosen. The data were recorded with an imbalance between the three classes (each class containing a different number of samples). Subsequently, the motors were operated at various speed profiles, ranging from 10 to 100% of the rated speed, to obtain multi-domain data. Each speed level was characterized as a single domain since each speed included information about the various time and frequency components of the original signal. Figure 8 shows the details of the hardware used in the process of data acquisition. For each axis motor, the data were recorded, although the fault was only placed into the RV reducer at axis four, as due to mechanical coupling, a fault in one axis could influence the operation and performance of the other axis motors. Table 2 shows the details of the recorded dataset. There are multiple speed domains from 10 to 100%. The data are imbalanced, with a different number of samples for each class—30, 27, and 24 for normal, faulty, and faulty age, respectively. The data are scarce as only 30 samples were recorded for the single domain at one axis. The final dataset consists of 4860 samples for the three classes.

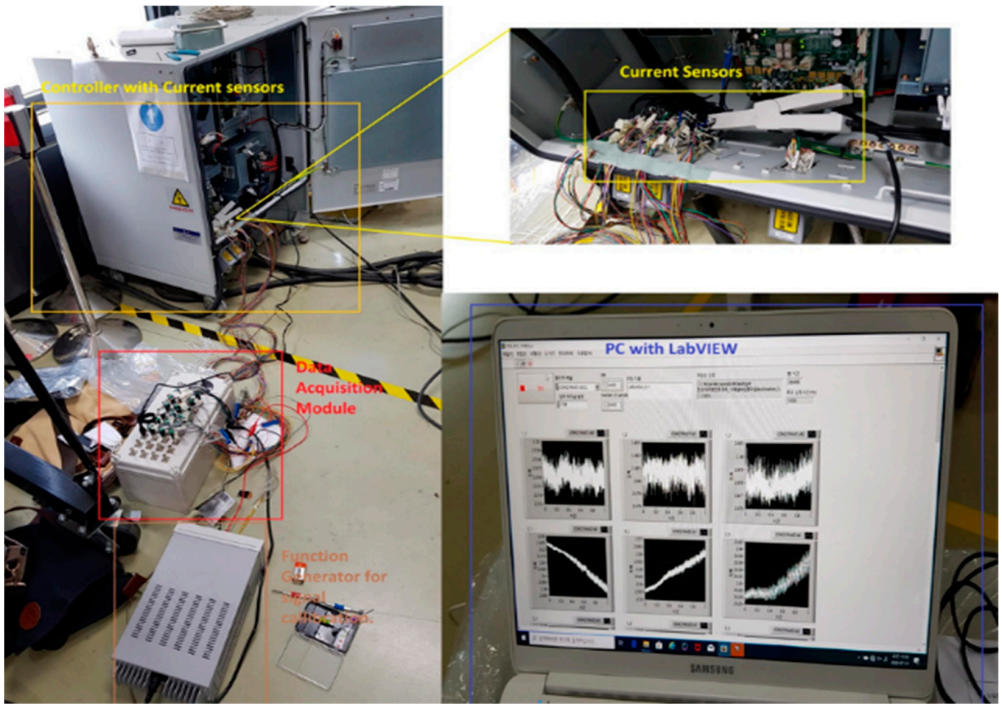

Figure 8. Details of the hardware used in the data collection.

Table 2. Details of the recorded dataset.

Recorded Dataset		Multi-Domain (Speed Profiles)									
		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
		Normal\Faulty\Faulty Age (Number of Samples (Cycles))									
Axis No.	Axis 1	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24
	Axis 2	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24
	Axis 3	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24
	Axis 4	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24
	Axis 5	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24
	Axis 6	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24	30\27\24
Total No. of Samples		180\162\144180\162\144180\162\144180\162\144180\162\144180\162\144180\162\144180\162\144180\162\144180\162\144									
Dataset Size		4860 × 3 (No. of Samples × No. of Classes)									

Data Preprocessing

To reduce the dimensionality and size of the recorded dataset, DQ0 transformation was implemented on the three-phase current signals. The DQ0 transformation was utilized to convert the three-phase current signal to a two-phase current signal in such a way that the information in the remaining two signals was preserved. DQ0 transformation is a well-known technique to reduce the dimensions of the electric signal, transforming three-phase current data to an arbitrary rotating framework of DQ0 by projecting the knowledge from a three-dimensional to a two-dimensional space. The resulting signals can be described by a circle in the projected two-dimensional space. This helps to simplify the frequency analysis since the circle correlates to the signal and helps in conserving the magnitude of the current or voltage signals. In this work, sinusoidal-based DQ0 transformation, which is given in Equation (1), was used. Figure 9 shows the three-phase and two-dimensional representation of the DQ0 transformation.

$$T_{abc-dq} = \sqrt{\frac{2}{3}} \begin{bmatrix} \sin \omega t & \sin\left(\omega t - \frac{2\pi}{3}\right) & \sin\left(\omega t + \frac{2\pi}{3}\right) \\ \cos \omega t & \cos\left(\omega t - \frac{2\pi}{3}\right) & \cos\left(\omega t + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (1)$$

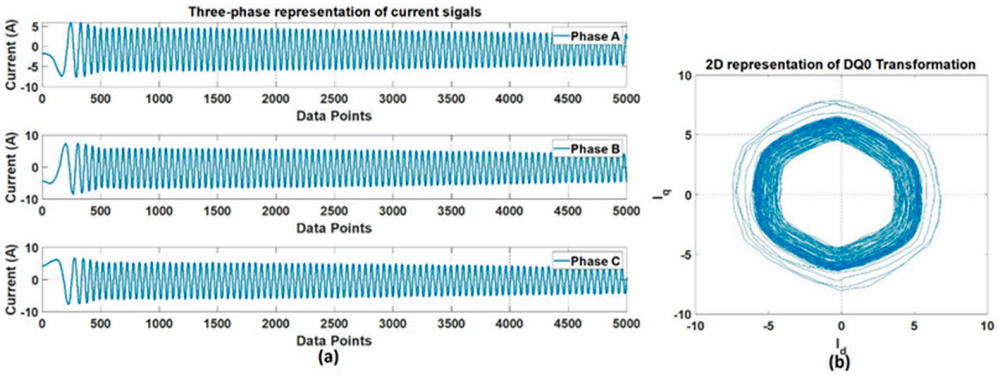

Figure 9. (**a**) Three-phase current signal, and (**b**) DQ0 representation.

Signal Processing and Data Dimension Conversion

Signals extracted from the sensors are in raw form and normally require preprocessing to eliminate noise and unnecessary information. Signal-processing techniques are used for this purpose, which help in analyzing the features of a specific signal in the time, frequency, and time-frequency domains. There are several types of signal-processing techniques based on different domains. These techniques are categorized as time-domain, frequency-domain, and time-frequency domain analyses. Statistical parameters illustrating valuable knowledge in the time domain are extracted from the signal [40–42] in time-domain analysis,

whereas, for frequency-domain analysis, Fourier transform (FT) is the most frequently used method. This decomposes the signals into constituent frequencies. Another technique is the fast-Fourier transform (FFT), which is widely used for the analysis of continuous-time signals. This transformation uses spectral frequencies for the analysis of a signal. However, for the processing of non-stationary signals, such as in this particular work, time- and frequency-domain analysis techniques have certain limitations.

Time-frequency domain analysis, a combination of the frequency and time domains, has been established to reduce these limitations [43]. The standard method used for this purpose is known as short-time Fourier transform (STFT) [44], which segregates the whole signal via FT and windows of small time periods into different segments. Another common method with a similar purpose is wavelet transform (WT) [45,46]. WT is an empirical technique that utilizes the interpretation of wavelet decomposition as a scaling idea for time-varying or non-stationary signals. This is a better approach for analyzing signals with a dynamic frequency spectrum at high resolution in both the time and frequency domains. Unlike other techniques, it not only informs which frequencies are present in a signal but also at which time these frequencies occurred.

WT is mainly divided into two types, DWT and CWT, each with its function in signal analysis. DWT is a non-redundant transformation that mainly focuses on one-to-one interaction between the information in the signal and transform domains. This close interaction makes DWT more appropriate for applications such as signal de-noising and reconstruction. However, CWT is more suitable for scalograms because the analysis window can be sized and configured at any position. This adaptability facilitates the generation of smooth images. Therefore, in this work, to construct a dataset rich with features of both the time and frequency domains, DWT was used for signal de-noising, and CWT for the generation of scalogram representations of each cycle for multiple speed domains. CWT was also used for the conversion of data from a one-dimensional signal to a two-dimensional image. CWT can be mathematically given as:

$$X_{\omega}(a,b) = \frac{1}{|a|^{\frac{1}{2}}} \int_{-\infty}^{\infty} x(t)\overline{\psi}\left(\frac{t-b}{a}\right)dt \tag{2}$$

where $\psi(t)$ is the continuous mother wavelet and the overline represents operation of complex conjugate, which is scaled by a factor of a and translated by a factor of b.

Figure 10 shows the overall flow of the signal processing and data dimension conversion techniques implemented in this work. Figure 11 shows an example of the de-noised signal using DWT where the blue signal is the original signal while the green signal is the de-noised signal. The signal is for one cycle of mechanical rotation of the robot along one axis, rather than an electrical cycle. On the other hand, Figure 12 shows an example of the scalogram images after the implementation of CWT on the de-noised cycles for a single domain of 10% rotation speed.

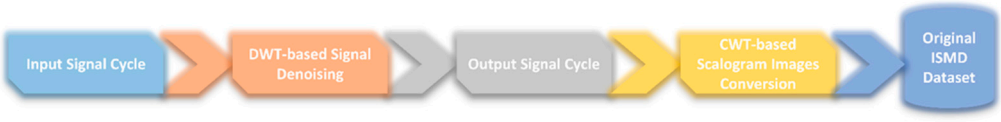

Figure 10. Flow of signal processing and data dimension conversion technique.

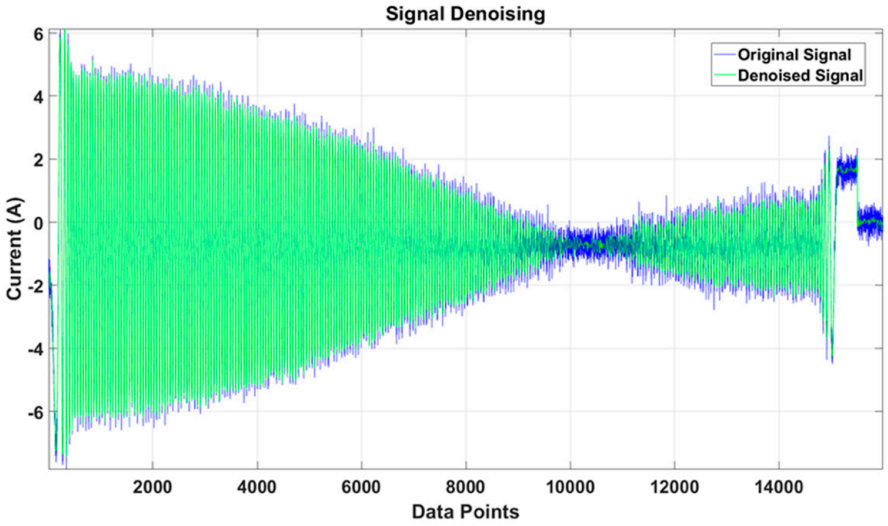

Figure 11. Example of the de-noised signal cycle after DWT implementation.

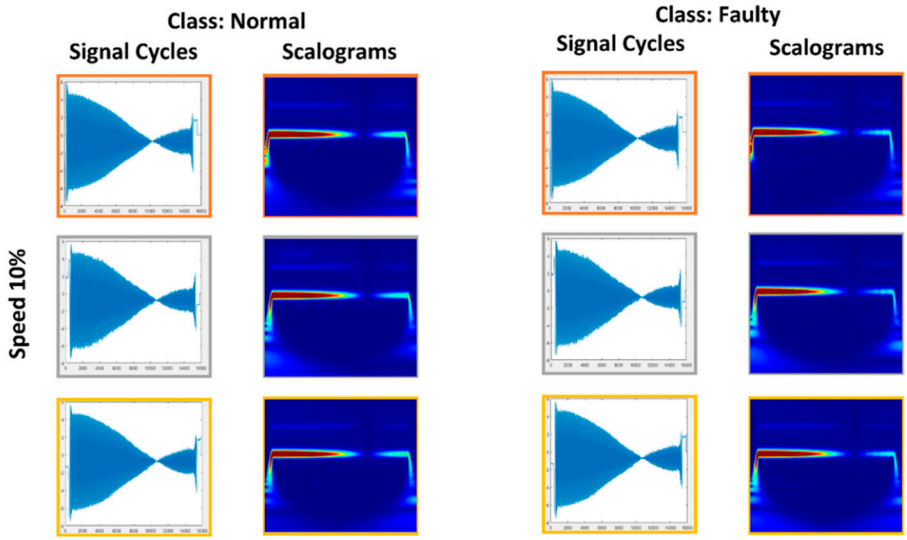

Figure 12. Scalogram images after CWT implementation for a single domain of 10% rotation speed.

Original ISMD Data

The final dataset after preprocessing and signal processing was stored in a database of scalogram images reflecting the original current signal cycles in various speed domains.

As stated earlier, to produce the dataset in this work, a single parameter of speed was the focus, to construct multi-domain data. Preliminary findings and signal processing aided the simplification of the data collection by removing redundant data from the dataset presented in Table 2. Hypothetically, when faults are imitated in axis four of the robot, there is a risk that the current signals of the other axis motors may be affected, which would entail simultaneous MCSA concentrating on the data of all axes at the same time to identify the faults. However, this was not the case with the Hyundai Robot; it was found that a fault in one axis would not affect the operation or signal pattern of the other axis motors. This meant the data for the axis where the fault lay could be solely relied on. Table 3 illustrates the size of the data after preprocessing and signal analysis. This dataset is presented in a scalogram representation, rather than a signal cycle. Each signal cycle is translated to a scalogram representation, and the number of samples is reduced from 30/27/24 to 20/17/14 for the normal/faulty/faulty age class, respectively, thanks to DQ0 transformation.

Table 3. Original ISMD data.

Refined Original ISMD Dataset	Multi-Domain (Speed Profiles)									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
	Normal\Faulty\Faulty Age (Number of Samples (Cycles))									
Total No. of Samples (Axis 4) Dataset Size	20\17\14	20\17\14	20\17\14	20\17\14	20\17\14	20\17\14	20\17\14	20\17\14	20\17\14	20\17\14
	510 × 3 (No. of Samples × No. of Classes)									

Figure 13 shows the detailed scalogram obtained after CWT implementation. The exemplary scalogram is for the speed of 10%. The *x*-axis shows the time in milliseconds and the *y*-axis shows the scales chosen based on the frequency spectrum of the input signal. The color bar on the right-hand side shows the magnitude of a specific scale at a specific instance. Note that the *x*- and *y*-axis values for each speed case were kept similar for better comparison. Figure 14 shows the scalogram images for each speed domain for a single cycle. There is a clear difference in the scalogram images: as the speed of the rotation increases, the frequency also increases with different time scales. Figure 15 shows the comparison of three classes in RGB and greyscale formats at a speed of 10%.

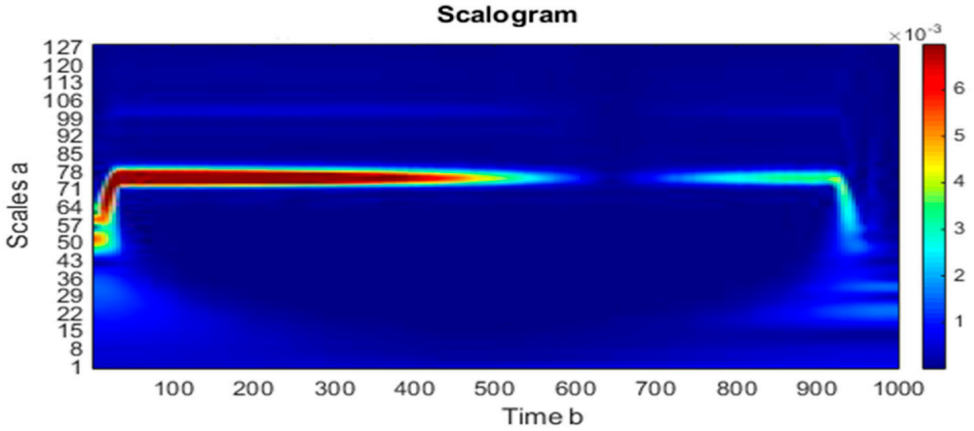

Figure 13. Example of a scalogram obtained after CWT implementation for the speed of 10%.

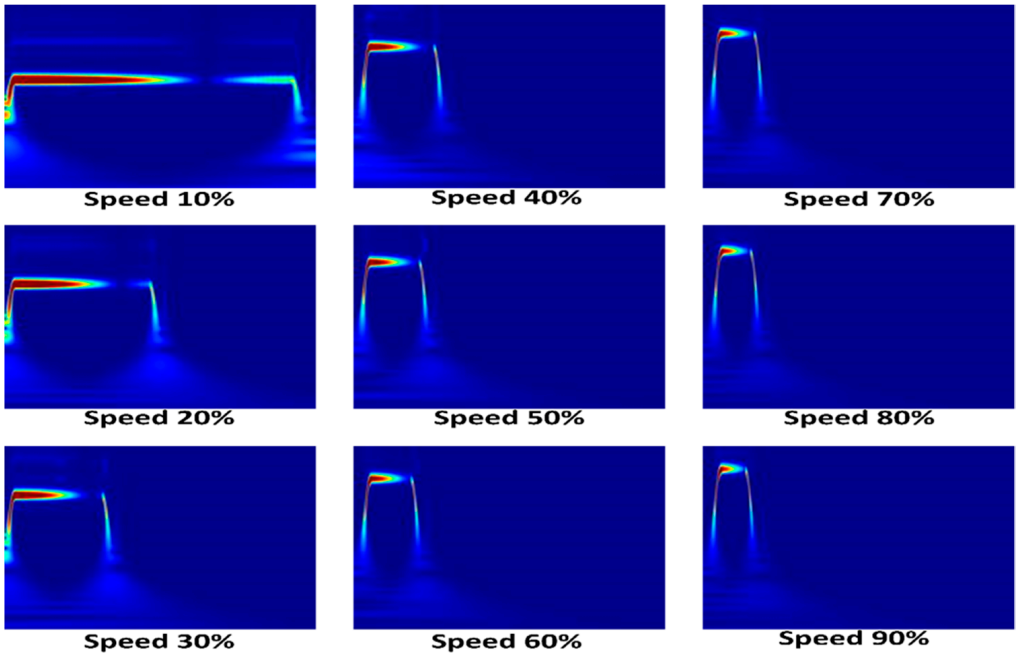

Figure 14. Scalogram images for multiple speed domains for a single cycle.

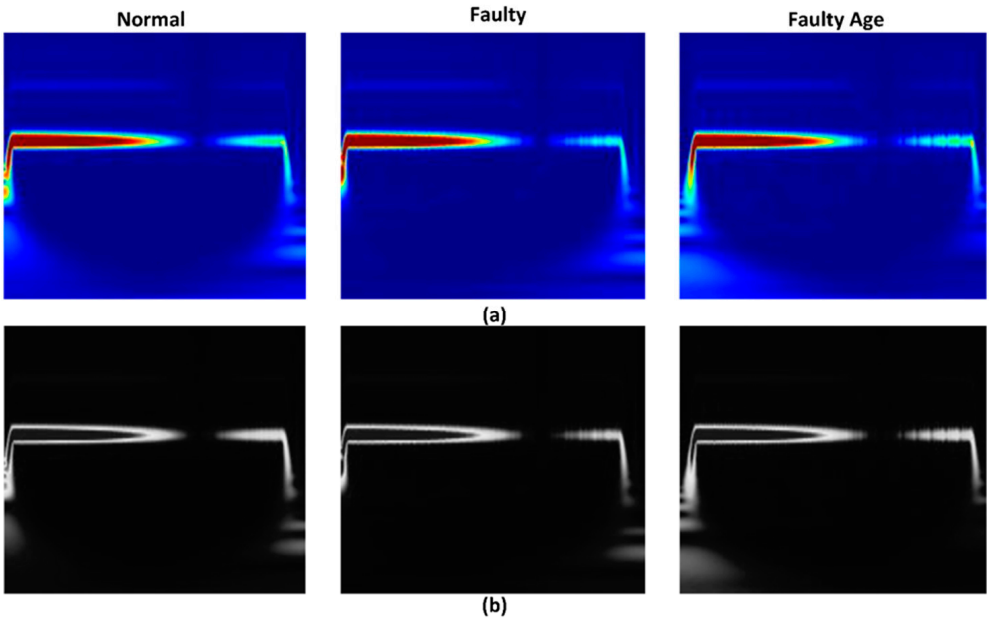

Figure 15. Comparison among different classes; (**a**) RGB images, and (**b**) greyscale images.

2.2.2. Domain Knowledge Transfer

The original ISMD dataset, which had several speed domains with scarce and imbalanced characteristics, could not be considered to perform well for the classification of faults. Yet, if the classification was carried out at a fair stage, the system would still be limited to an application-specific environment, with little potential for interpretability. Therefore, it was necessary to either collect further data with thousands of samples, which are, in most cases, very complex for industrial robots and systems, or to adopt a method in which data features with minimal samples across several domains can be translated, and based on those translated features, create a new dataset of translated knowledge, which can be further used to establish DL-based classification models. The latter was not feasible until recently when studies [38,39] suggested methods for image-to-image translation, where a GAN with an input source and the reference image is used to learn the features of the source and reference, generating a new image with features from both. This GAN has shown positive results in the field of computer vision, being particularly efficient in areas such as image synthesis [47–49], translation [50–52], and creation [53–56]. The previous GAN image-to-image translation framework was limited to a single domain, which could be used with a single network. To translate images through multiple domains, a new network was required, increasing the computational cost for the generation of systems with the multi-domain image-to-image translation.

Figure 16 demonstrates the difference between the previous cross-domain models and the StarGAN [38] used in this work, in (a), addressing how 12 independent generator networks were trained to interpret images from only four areas, whereas, in (b), the model demanded training data from several environments, and learned to map among all domains using only one generator. Figure 17 displays the StarGAN architecture. The first part of the network is a generator, which takes an input image and converts it into an output image with a domain-specific style code. The second part is a mapping network that converts hidden code to style codes for various domains randomly chosen during the training process. The third part is a style encoder, which derives an image style code that enables the generator to execute reference-guided image fusion. The last part, like other GANs, is the discriminator, which differentiates between original and fake images from a variety of domains.

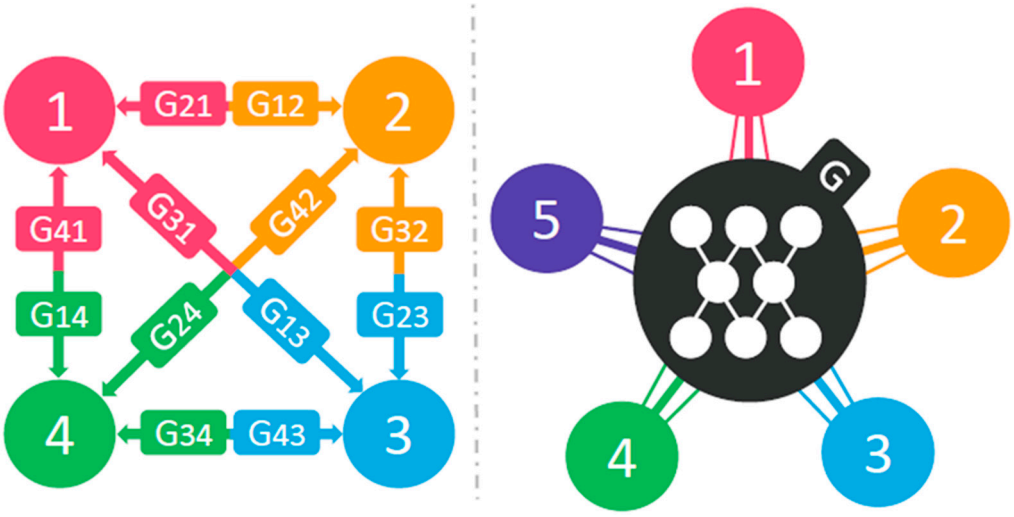

Figure 16. Comparison between (**a**) previous cross-domain models, and (**b**) StarGAN [39].

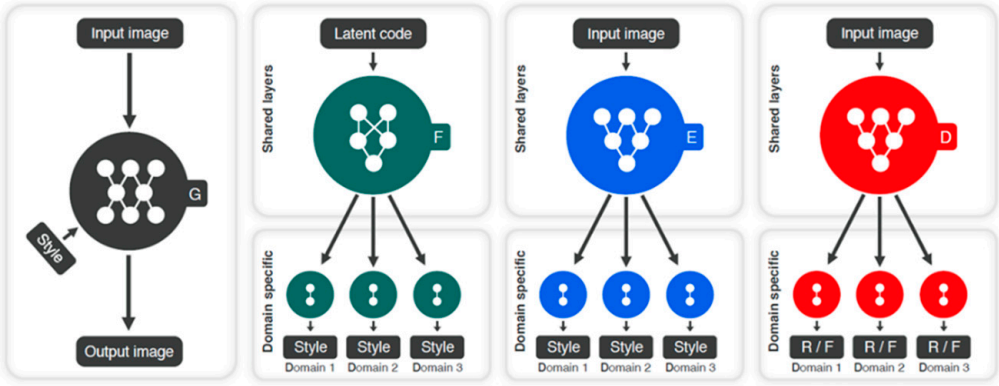

Figure 17. StarGAN architecture; (**a**) generator, (**b**) mapping network, (**c**) style encoder, and (**d**) discriminator [38].

2.2.3. Data Splitting and Classification

Using domain knowledge transfer with the StarGAN mentioned in the previous section, a dataset comprising scalogram images from each class and translated images across each speed domain was created. Initially, three datasets for the normal, faulty, and faulty age classes were generated using three StarGANs trained specifically to tackle a single class with multi-domain data. Those, along with the original ISMD dataset, were used as the final data, split into training and validation datasets. Figure 18 shows the details of the data generation and domain knowledge transfer for a single domain of 10% speed. Note that the digits on the blocks show the total number of images present at the specific stage. The size of the generated images is 224 × 224 pixels. The same operation was carried out for each domain, and the final dataset was obtained. The total number of images for domain knowledge transfer among two domains can be calculated using Equation (3):

$$N_{ab} = (n_a \times n_b) \times (n_d \times N_d) \quad (3)$$

where N_{ab} is the total number of images after knowledge transfer between domains a and b, n_a is the number of images in domain a, n_b is the number of images in domain b, n_d is the total number of domains, and N_d is the domain number = 1, 2, 3, . . . , 10 for the speed (10, 20, 30, . . . , 100) percentage. Table 4 shows the possible combinations of domain knowledge transfer between different domains, with the total number of generated images

(calculated using Equation (3)) and the overall final dataset size. Note that domain 10 with 100% speed is not present in the following table since it has already been compensated for in the data generation for the other domains.

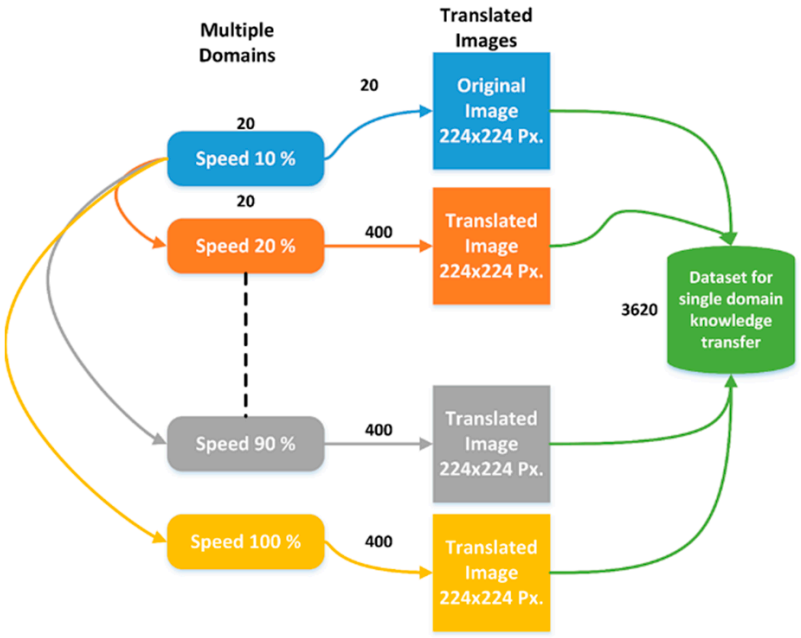

Figure 18. Details of domain knowledge transfer and data generation for a single domain of 10% speed.

Table 4. Final dataset with possible combinations of domain knowledge transfer.

Number of Combinations/ Operations	Domain 1 (10% Speed)	Domain 2 (20% Speed)	Domain 3 (30% Speed)	Domain 4 (40% Speed)	Domain 5 (50% Speed)	Domain 6 (60% Speed)	Domain 7 (70% Speed)	Domain 8 (80% Speed)	Domain 9 (90% Speed)
1	(10, 20)	×	×	×	×	×	×	×	×
2	(10, 30)	(20, 30)	×	×	×	×	×	×	×
3	(10, 40)	(20, 40)	(30, 40)	×	×	×	×	×	×
4	(10, 50)	(20, 50)	(30, 50)	(40, 50)	×	×	×	×	×
5	(10,60)	(20,60)	(30,60)	(40,60)	(50,60)	×	×	×	×
6	(10, 70)	(20, 70)	(30, 70)	(40, 70)	(50, 70)	(60, 70)	×	×	×
7	(10, 80)	(20, 80)	(30, 80)	(40, 80)	(50, 80)	(60, 80)	(70, 80)	×	×
8	(10, 90)	(20, 90)	(30, 90)	(40, 90)	(50, 90)	(60, 90)	(70, 90)	(80, 90)	×
9	(10, 100)	(20, 100)	(30, 100)	(40, 100)	(50, 100)	(60, 100)	(70, 100)	(80, 100)	(90, 100)
Generated Images/Classes	3620	3220	2820	2420	2020	1620	1220	820	420
Final Dataset Size	40847 × 3 (No. of Samples × No. of Classes)								

The final dataset mentioned in Table 4 was used for the classification of the faults using transfer learning on some of the prominent benchmark CNN models. In this work, finetuning, a transfer learning principle that substitutes the pre-trained output layer with a layer containing the number of final dataset classes, was used. The last three layers were substituted with a fully connected layer, a softmax layer, and a classification layer. The primary objective of using pre-trained CNN models is to achieve quick and accurate training on a CNN's manipulation of a random initialization of weights and to accomplish a low training error. The effectiveness of the GoogLeNet [57], SqueezeNet [58], AlexNet [59], VGG16 [60], Inception v3 [61], and ResNet50 [62] architectures was analyzed for this specific fault detection and diagnosis system. Table 5 presents the characteristics of these CNN architectures.

Table 5. Characteristics of the benchmark CNN models.

Network	Depth/No. of Layers	Input Image Size
GoogLeNet	22	224 × 224
SqueezeNet	18	227 × 227
AlexNet	8	227 × 227
VGG16	16	224 × 224
Inceptionv3	48	299 × 299
ResNet50	50	224 × 224

3. Results and Discussion

3.1. Domain Knowledge Transfer for ISMD Data/Data Generation

Table 6 shows the specifications of the PC used in this work. For domain knowledge transfer, StarGAN was trained for 100,000 iterations. The recorded training time was approximately 5 h for a single class and 15 h for three classes. The number of iterations was carefully selected after observing the loss for the generator and the descriptor module of StarGAN. Figure 19 shows some results of the latent images generated during the training process at 10,000, 20,000, 30,000, and 40,000 iterations. As can be seen, at different iterations, the network learns more and more features, generating images with a similar pattern to the input. For precise analysis, results are presented with a focus on only one class: normal. The results for the other classes had a similar pattern, but with different features and information. In Figure 19, (a) represents the speed domain of 30%, (b) 50%, (c) 80%, and (d) 100%.

Table 6. PC specifications.

Component	Detail
CPU	Intel ® Core™ i7-8700k CPU Eight-core @ 3.0 GHz
Memory	32 GB
GPU	NVIDIA GeForce RTX 2080 Ti
Operating System	Linux

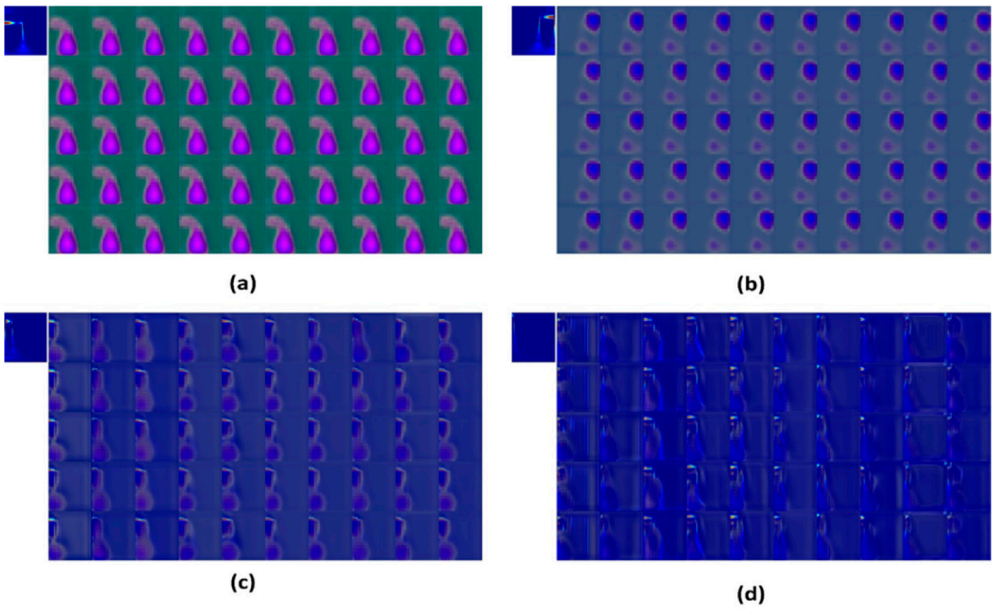

Figure 19. Latent images generated during the training process for the normal class (**a**) at 10,000 iterations for 30% speed, (**b**) 20,000 iterations 50% speed, (**c**) 30,000 iterations for 80% speed, and (**d**) 40,000 iterations for 100% speed.

Figure 20 shows some of the reference images generated during the training process of the network for different domains and a different number of iterations for the normal class. The red rectangular box in (a) represents the scalogram image for domain one at 40% speed, while the orange one represents domain two at 20% speed; the yellow one, meanwhile, is the generated output image after domain knowledge transfer at 10,000 training iterations. (b) shows the domain knowledge transfer between 10 and 30% speed at 20,000 training iterations, (c) 60 and 100% speed at 30,000 training iterations, and (d) 80 and 100% speed at 40,000 training iterations.

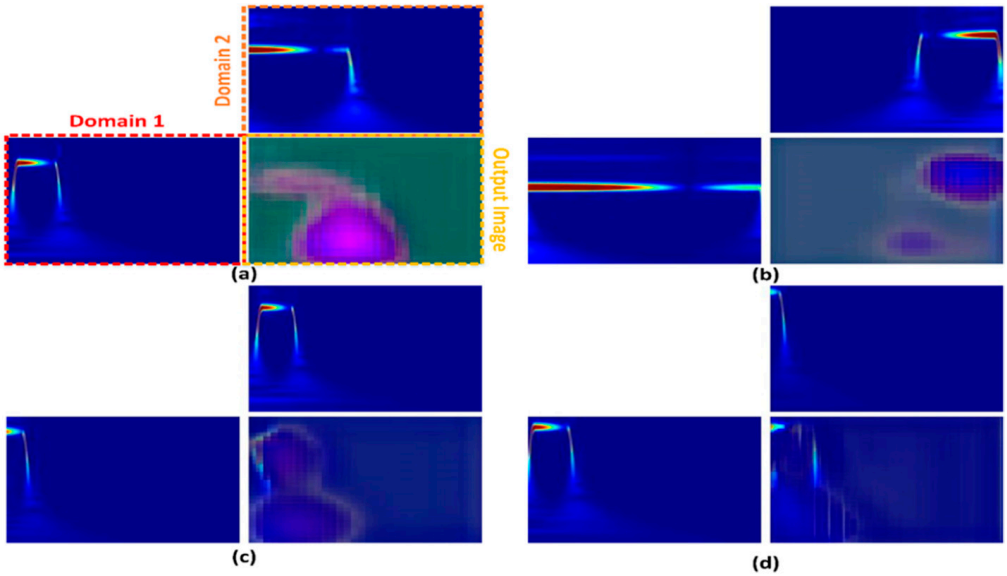

Figure 20. Reference images generated during the training process with domain knowledge transfer (**a**) speed (40 and 20)% at 10,000 iterations, (**b**) (10 and 30)% at 20,000 iterations, (**c**) (100 and 60)% at 30,000 iterations, and (**d**) (80 and 100)% at 40,000 iterations.

Figure 21 shows the results of the generated images after training the network. The reference images are those provided to the network as a reference, with source images acting as the input at a specific time. The network takes the source image, compares it with the reference image, and based on the learned features from both sources and references, generates the output image. For multiple speed domains, 10 to 100%, a relationship is established, and several images are generated based on the combinations previously presented in Table 4. Note that there are combinations in Figure 21, such as (10, 10), (20, 20), (30, 30), (40, 40), (50, 50), (60, 60), (70, 70), (80, 80), (90, 90), and (100, 100), which act as 'do not care' conditions (similar to in Table 4). These combinations are segregated from the final dataset since the goal is to transfer knowledge across multiple domains with the infusion of several frequency- and time-domain features at different levels of speed, rather than the same set of speed domains. This also reduces the size and redundancy of the final dataset. As previously mentioned, similar images as in Figure 21 are generated for the other two classes, faulty and faulty age, to create a complete dataset. The implementation of StarGAN is available as opensources for both the PyTorch and Tensorflow libraries in the GitHub repository [63]. In this particular work, the TensorFlow implementation was used.

The dataset obtained from StarGAN with the original dataset recorded in real-time was used by keeping the resource parameters intact. Each class of the generated dataset with the corresponding class of the recorded dataset was used for training and validation purposes. The size of the final dataset (total dataset) was 40,847 $\times$ 3 (Number of samples/images $\times$ Number of classes).

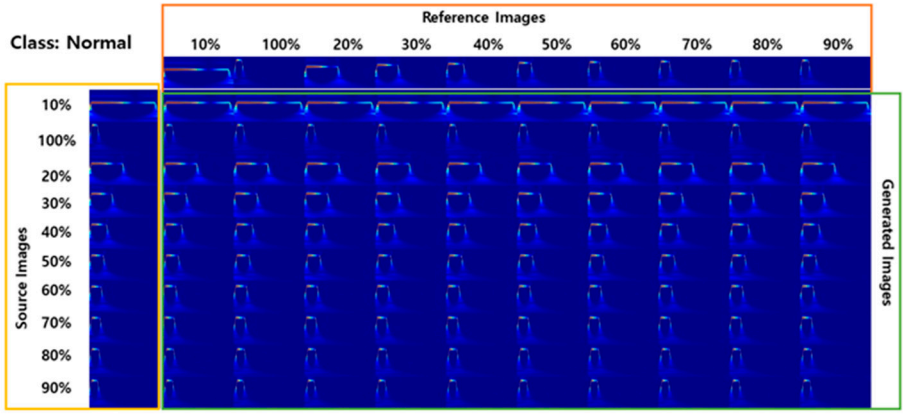

Figure 21. Generated images for the normal class with domain knowledge transfer.

3.2. Fault Classification

To classify the faults and validate the efficacy of the proposed data-generation methodology, transfer learning on some of the prominent benchmarks of CNN was adopted. Six CNN models, where each had a different number of layers and input image size, and each was developed with a different analogy to carry out image classification tasks, were trained. For comparison purposes, each of these models was trained with the same number of epochs, and with similar batch sizes and hyperparameters. Table 7 presents the values of the hyperparameters used during the training of the CNNs. The training was performed on the PC with the specifications mentioned in Table 6. The final generated dataset was used as a training dataset. The validation dataset contained the data from the original ISMD dataset, whereas the training dataset was composed of the data generated using StarGAN. This was done to verify the interpolation capacity of the trained CNN models and to avoid overfitting. The validation was performed on unseen data not generated by StarGAN. The training dataset was further divided into the training and testing datasets with the ratio of 70:30%. The split strategy was chosen carefully to counter the computationally intensive task of classification.

Table 7. Hyperparameters used for training.

Hyperparameters	Value
Batch Size	32
Epochs	30
Learning Rate	0.0001
L2 Regularization	0.00001
Optimization Algorithm	Stochastic Gradient Descent

The performance of the six CNN models (GoogLeNet, SqueezeNet, AlexNet, VGG16, Inception v3, and ResNet50) was evaluated using the following performance metrics. Some of these metrics are widely used to test the performance of a trained ML/DL model. Among these metrics available, the accuracy, sensitivity, specificity, precision, and F-score were used. Accuracy is the most important metric, which reveals the number of samples that are correctly classified out of all the samples. Generally, it is expressed as the ratio of true positives (TPs) and true negatives (TNs) divided by the sum of the total number of TPs, TNs, false positives (FPs), and false negatives (FNs). A TP or TN is a data sample that the algorithm accurately classifies as true or false. On the other hand, an FP or FN is a data sample that the algorithm falsely classifies. Equation (4) represents this metric:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

The sensitivity metric is also known as the recall. It is defined as the number of accurately classified positive samples, which implies how many samples of the positive classes are identified correctly. It is given in Equation (5):

$$\text{Sensitivity/Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

Specificity refers to the conditional possibility of the TN in the given class; this implies a prediction of the possibility of a negative label becoming true. It can be represented as in Equation (6):

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{6}$$

Precision, calculated as the percentage of TPs, divided by the number of TPs plus the number of FPs, is given by Equation (7). This metric is about consistency, i.e., it measures the prediction performance of the algorithm. It tells us how precise a model is based on what is expected to be positive, and given how much of that is truly positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

Finally, the F-score is defined as the relative average of precision and recall, as shown in Equation (8). It relies on positive class evaluation. A high value of this parameter suggests that the model performs the best in the positive class. Table 8 presents the results achieved when finetuning the CNNs based on the performance metrics.

$$\text{F} - \text{score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \tag{8}$$

Table 8. Results of the evaluation metrics and CNN performance.

CNN Model	Metric				
	Accuracy	**Sensitivity**	**Specificity**	**Precision**	**F-Score**
GoogLeNet	99.71325864	99.61988796	99.75994398	99.61988796	99.619888
SqueezeNet	99.61988796	99.47983193	99.68991597	99.48506224	99.4798152
VGG16	99.3331	99.0997	99.4499	99.102	99.1
AlexNet	98.01269841	97.21904762	98.40952381	97.37777778	97.220979
Inceptionv3	97.9994	97.2992	98.3496	97.366	97.298
ResNet50	95.7055	94.2583	96.4921	94.7586	94.2398

The results are significantly promising for all six CNN models, starting with GoogLeNet, which achieved an accuracy of 99.7%, with high predictability for the positive and negative classes, and with a sensitivity of 99.6% and specificity of 99.7%. It was undoubtedly the best model to classify the faults. Following it was SqueezeNet with an accuracy, sensitivity, and specificity of 99.6%, 99.4%, and 99.6%, respectively. SqueezeNet performed considerably well for the overall classification. The third best was VGG16, which achieved an accuracy, sensitivity, and specificity of 99.3%, 99.0%, and 99.4%, respectively, followed by AlexNet, which showed less accurate results compared to the top three models. The maximum accuracy was recorded as 98.0%, with a sensitivity and specificity of 97.2% and 98.4%, respectively. Inceptionv3 and ResNet50, meanwhile, performed considerably less effectively than the other models but were still better at solving the fault classification problem.

The architectures of these CNN models differ from one another. GoogLeNet is composed of 22 layers, where there are multiple convolutional layers, followed by the inception module and more convolutional layers. The greater number of layers helps the network to extract useful, distinguishable, and prominent features from a given training dataset. The model is heavy compared to the others but performs well in learning features with multiple filters. On the other hand, SqueezeNet is composed of 18 layers, designed with a

different analogy from GoogLeNet meaning it has relatively fewer parameters. VGG 16 and AlexNet are composed of 16 and 8 layers, respectively, where convolutional layers are stacked one after another. While they were proven to be promising models for image classification, in this study, they were inefficient for fault classification. While the number of layers can be an important parameter to consider when designing the architecture of a neural network, the relationship between the different layers of the network can also play a significant role in improving the learning capacity of a network. Inceptionv3 and ResNet50 are examples of such networks. Even though both these networks have a high number of layers compared to GoogLeNet, they nevertheless do not perform well for the fault classification problem. Inceptionv3 is composed of 48 layers, while ResNet50 is composed of 50 layers. The poor performance of these architectures has to do with the way the layers are related to one another in the network. In Inceptionv3, multiple inception modules are used that are comprised of multiple convolutional layers, whereas, in ResNet50, the skip connection concept of residual networks is used, where the output from a layer can be directly fed as an input to the other layer at a specific level of the architecture, while skipping the layers between them. The results of this work also prove that such a network fails to show more prominent results due to the design of the architecture and specific implementation methodologies.

Figure 22 shows the confusion matrices for the top two models. The rows correspond to the expected output class, while the columns correspond to the real target class. The diagonal cells correspond to the groups that are precisely classified. Off-diagonal cells correspond to groups that are falsely classified. Each cell displays the number of samples and their percentile. The far-right column represents the percentiles of all the samples expected for each class, correctly and incorrectly classified. These metrics are also known as the precision and the false discovery rate. In both the cases of GoogLeNet and SqueezeNet, more confusion is found between the normal and faulty classes. Figure 15, which shows the scalogram images of each class, also confirms this. The difference between the Normal and Faulty Age class is more visible than the Faulty class, making it easier for the CNN model to classify with high accuracy. Almost 100% accuracy between the Normal/Faulty Age and Faulty/Faulty Age class is achieved.

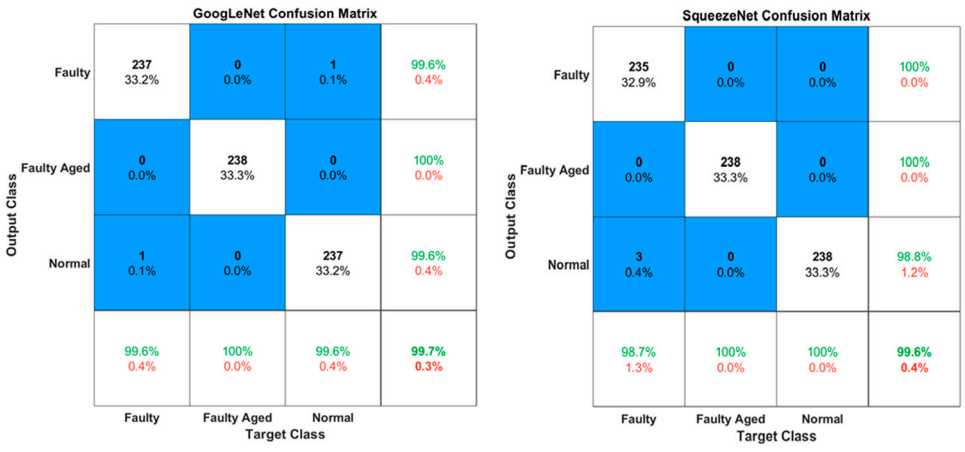

Figure 22. Confusion matrices for GoogLeNet and SqueezeNet.

4. Conclusions

This study provides a holistic approach to the detection and diagnosis of faults related to the mechanical component of an industrial robot in ISMD data settings for component-level PHM. The obtained results were impressive for six CNN benchmark models: GoogLeNet, SqueezeNet, VGG16, AlexNet, Inceptionv3, and ResNet50, with accuracies of 99.7%, 99.6%, 99.3%, 98.0%, 97.9%, and 95.7%, respectively. The results achieved demonstrate that the suggested methodology can overcome the key problem of ISMD data. The benefit will be the ability to design potential methods for detecting and diagnosing

faults before they worsen, at times when we have less knowledge about the type of fault. Furthermore, interpolation between various types of faults and different robots will be made feasible by the transition of domain knowledge. Collecting or providing a limited amount of data could become the basis for constructing DL models that are highly efficient, and that can be applied in real-time with a realistic framework. In future work, there is a need to concentrate on further faults related to electrical and mechanical components in various types of robots, to examine and build a system that can recognize not just faults but also the robots where specific faults occur.

Funding: This research was financially supported by the Ministry of Trade, Industry, and Energy (MOTIE) and the Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program (project no. P059500003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The author is grateful for the support provided by the team of SMD Lab, Dongguk University, South Korea including Heung-soo Kim, Hyewon Lee, and Izaz Rauf in helping with communication with the industry partner and providing the resources and funding for the project.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Lee, J.; Wu, F.; Zhao, W.; Ghaffari, M.; Liao, L.; Siegel, D. Prognostics and Health Management Design for Rotary Machinery Systems—Reviews, Methodology and Applications. *Mech. Syst. Signal Process.* **2014**, *42*, 314–334. [CrossRef]
2. Lall, P.; Lowe, R.; Goebel, K. Prognostics and Health Monitoring of Electronic Systems. In Proceedings of the 2011 12th Interantional Conference on Thermal, Mechanical & Multi-Physics Simulation and Experiments in Microelectronics and Microsystems, Linz, Austria, 18–20 April 2011. [CrossRef]
3. Carvalho Bittencourt, A. Modeling and Diagnosis of Friction and Wear in Industrial Robots. Ph.D. Dissertation, Linköping University Electronic Press, Linköping, Sweden, 2014.
4. Abichou, B.; Voisin, A.; Iung, B. Bottom-up Capacities Inference for Health Indicator Fusion within Multi-Level Industrial Systems. In Proceedings of the 2012 IEEE Conference on Prognostics and Health Management, Denver, CO, USA, 18–21 June 2012. [CrossRef]
5. Sheppard, J.; Kaufman, M.; Wilmer, T. IEEE Standards for Prognostics and Health Management. *IEEE Aerosp. Electron. Syst. Mag.* **2009**, *24*, 34–41. [CrossRef]
6. Yang, J.; Kim, J. An Accident Diagnosis Algorithm Using Long Short-Term Memory. *Nucl. Eng. Technol.* **2018**, *50*, 582–588. [CrossRef]
7. Zhang, L.; Lin, J.; Karim, R. Adaptive Kernel Density-Based Anomaly Detection for Nonlinear Systems. *Knowl.-Based Syst.* **2018**, *139*, 50–63. [CrossRef]
8. Fan, J.; Yung, K.C.; Pecht, M. Physics-of-Failure-Based Prognostics and Health Management for High-Power White Light-Emitting Diode Lighting. *IEEE Trans. Device Mater. Reliab.* **2011**, *11*, 407–416. [CrossRef]
9. Pecht, M.; Gu, J. Physics-of-Failure-Based Prognostics for Electronic Products. *Trans. Inst. Meas. Control.* **2009**, *31*, 309–322. [CrossRef]
10. Tsui, K.L.; Chen, N.; Zhou, Q.; Hai, Y.; Wang, W. Prognostics and Health Management: A Review on Data Driven Approaches. *Math. Probl. Eng.* **2015**, *2015*, 793161. [CrossRef]
11. Gao, Z.; Cecati, C.; Ding, S. A Survey of Fault Diagnosis and Fault-Tolerant Techniques Part II: Fault Diagnosis with Knowledge-Based and Hybrid/Active Approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [CrossRef]
12. Rohan, A.; Kim, S.H. RLC Fault Detection Based on Image Processing and Artificial Neural Network. *Int. J. Fuzzy Log. Intell. Syst.* **2019**, *19*, 78–87. [CrossRef]
13. Rohan, A.; Kim, S.H. Fault Detection and Diagnosis System for a Three-Phase Inverter Using a DWT-Based Artificial Neural Network. *Int. J. Fuzzy Log. Intell. Syst.* **2016**, *16*, 238–245. [CrossRef]
14. Rohan, A.; Rabah, M.; Kim, S.H. An Integrated Fault Detection and Identification System for Permanent Magnet Synchronous Motor in Electric Vehicles. *Int. J. Fuzzy Log. Intell. Syst.* **2018**, *18*, 20–28. [CrossRef]
15. Ding, S.X. Model-Based Fault Diagnosis Techniques. In *Advances in Industrial Control*; Springer: New York, NY, USA, 2013; pp. 405–440.
16. Liu, J.; Luo, W.; Yang, X.; Wu, L. Robust Model-Based Fault Diagnosis for PEM Fuel Cell Air-Feed System. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3261–3270. [CrossRef]
17. Ding, S.X.; Zhang, P.; Jeinsch, T.; Ding, E.L.; Engel, P.; Gui, W. A Survey of the Application of Basic Data-Driven and Model-Based Methods in Process Monitoring and Fault Diagnosis. *IFAC Proc. Vol.* **2011**, *44*, 12380–12388. [CrossRef]

18. Yin, S.; Ding, S.X.; Xie, X.; Luo, H. A Review on Basic Data-Driven Approaches for Industrial Process Monitoring. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6418–6428. [CrossRef]
19. Hu, C.; Youn, B.D.; Wang, P.; Taek Yoon, J. Ensemble of Data-Driven Prognostic Algorithms for Robust Prediction of Remaining Useful Life. *Reliab. Eng. Syst. Saf.* **2012**, *103*, 120–135. [CrossRef]
20. Zhou, W.; Habetler, T.G.; Harley, R.G. Bearing Condition Monitoring Methods for Electric Machines: A General Review. Available online: https://ieeexplore.ieee.org/abstract/document/4393062 (accessed on 28 April 2020). [CrossRef]
21. Hamadache, M.; Lee, D.; Veluvolu, K.C. Rotor Speed-Based Bearing Fault Diagnosis (RSB-BFD) under Variable Speed and Constant Load. *IEEE Trans. Ind. Electron.* **2015**, *62*, 6486–6495. [CrossRef]
22. Xu, Y.; Sun, Y.; Wan, J.; Liu, X.; Song, Z. Industrial Big Data for Fault Diagnosis: Taxonomy, Review, and Applications. *IEEE Access* **2017**, *5*, 17368–17380. [CrossRef]
23. Liao, L.; Kottig, F. Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction. *IEEE Trans. Reliab.* **2014**, *63*, 191–207. [CrossRef]
24. Pham, M.T.; Kim, J.M.; Kim, C.H. Deep Learning-Based Bearing Fault Diagnosis Method for Embedded Systems. *Sensors* **2020**, *20*, 6886. [CrossRef]
25. Cerrada, M.; Sánchez, R.V.; Li, C.; Pacheco, F.; Cabrera, D.; Valente de Oliveira, J.; Vásquez, R.E. A Review on Data-Driven Fault Severity Assessment in Rolling Bearings. *Mech. Syst. Signal Process.* **2018**, *99*, 169–196. [CrossRef]
26. Wang, D.; Tsui, K.L.; Miao, Q. Prognostics and Health Management: A Review of Vibration Based Bearing and Gear Health Indicators. *IEEE Access* **2018**, *6*, 665–676. [CrossRef]
27. Verma, N.K.; Gupta, V.K.; Sharma, M.; Sevakula, R.K. Intelligent Condition Based Monitoring of Rotating Machines Using Sparse Auto-Encoders. In Proceedings of the 2013 IEEE Conference on Prognostics and Health Management (PHM), Gaithersburg, MD, USA, 24–27 June 2013. [CrossRef]
28. Zhuang, Z.; Lv, H.; Xu, J.; Huang, Z.; Qin, W. A Deep Learning Method for Bearing Fault Diagnosis through Stacked Residual Dilated Convolutions. *Appl. Sci.* **2019**, *9*, 1823. [CrossRef]
29. Guo, X.; Chen, L.; Shen, C. Hierarchical Adaptive Deep Convolution Neural Network and Its Application to Bearing Fault Diagnosis. *Measurement* **2016**, *93*, 490–502. [CrossRef]
30. Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; Van de Walle, R.; Van Hoecke, S. Convolutional Neural Network Based Fault Detection for Rotating Machinery. *J. Sound Vib.* **2016**, *377*, 331–345. [CrossRef]
31. Rohan, A.; Raouf, I.; Kim, H.S. Rotate Vector (RV) Reducer Fault Detection and Diagnosis System: Towards Component Level Prognostics and Health Management (PHM). *Sensors* **2020**, *20*, 6845. [CrossRef]
32. Zhang, L.; Lin, J.; Liu, B.; Zhang, Z.; Yan, X.; Wei, M. A Review on Deep Learning Applications in Prognostics and Health Management. *IEEE Access* **2019**, *7*, 162415–162438. [CrossRef]
33. Mariani, G.; Scheidegger, F.; Istrate, R.; Bekas, C.; Malossi, C. BAGAN: Data Augmentation with Balancing GAN. *arXiv* **2018**, arXiv:1803.09655. [CrossRef]
34. Springenberg, J.T. Unsupervised and Semi-Supervised Learning with Categorical Generative Adversarial Networks. *arXiv* **2016**, arXiv:1511.06390. [CrossRef]
35. Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. Synthetic Data Augmentation Using GAN for Improved Liver Lesion Classification. *arXiv* **2018**, arXiv:1801.02385. [CrossRef]
36. Carino, J.A.; Delgado-Prieto, M.; Iglesias, J.A.; Sanchis, A.; Zurita, D.; Millan, M.; Ortega Redondo, J.A.; Romero-Troncoso, R. Fault Detection and Identification Methodology under an Incremental Learning Framework Applied to Industrial Machinery. *IEEE Access* **2018**, *6*, 49755–49766. [CrossRef]
37. Calabrese, F.; Regattieri, A.; Bortolini, M.; Galizia, F.G.; Visentini, L. Feature-Based Multi-Class Classification and Novelty Detection for Fault Diagnosis of Industrial Machinery. *Appl. Sci.* **2021**, *11*, 9580. [CrossRef]
38. Choi, Y.; Uh, Y.; Yoo, J.; Ha, J.W. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8185–8194. [CrossRef]
39. Choi, Y.; Choi, M.; Kim, M.; Ha, J.W.; Kim, S.; Choo, J. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-To-Image Translation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
40. Lei, Y.; Zuo, M.J. Gear Crack Level Identification Based on Weighted K Nearest Neighbor Classification Algorithm. *Mech. Syst. Signal Process.* **2009**, *23*, 1535–1547. [CrossRef]
41. Bechhoefer, E.; Kingsley, M. A Review of Time Synchronous Average Algorithms. In Proceedings of the Annual Conference of the Prognostics and Health Management Society, San Diego, CA, USA, 27 September–1 October 2009; Volume 1.
42. Braun, S. The Synchronous (Time Domain) Average Revisited. *Mech. Syst. Signal Process.* **2011**, *25*, 1087–1102. [CrossRef]
43. He, Q.; Liu, Y.; Long, Q.; Wang, J. Time-Frequency Manifold as a Signature for Machine Health Diagnosis. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 1218–1230. [CrossRef]
44. Portnoff, M. Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Analysis. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 55–69. [CrossRef]
45. Tarasiuk, T. Hybrid Wavelet-Fourier Spectrum Analysis. *IEEE Trans. Power Deliv.* **2004**, *19*, 957–964. [CrossRef]
46. Hassanpour, H.; Shahiri, M. Adaptive Segmentation Using Wavelet Transform. In Proceedings of the 2007 International Conference on Electrical Engineering, Lahore, Pakistan, 11–12 April 2007. [CrossRef]

47. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv* **2018**, arXiv:1809.11096. [CrossRef]
48. Lucic, M.; Tschannen, M.; Ritter, M.; Zhai, X.; Bachem, O.; Gelly, S. High-Fidelity Image Generation with Fewer Labels. *arXiv* **2019**, arXiv:1903.02271. [CrossRef]
49. Donahue, J.; Simonyan, K. Large scale adversarial representation learning. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 10542–10552.
50. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2242–2251.
51. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. *arXiv* **2017**, arXiv:1703.05192. [CrossRef]
52. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-To-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
53. Yu, B.; Ding, Y.; Xie, Z.; Huang, D. Stacked Generative Adversarial Networks for Image Compositing. *EURASIP J. Image Video Process.* **2021**, *2021*, 10. [CrossRef]
54. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434. [CrossRef]
55. Zhao, J.; Mathieu, M.; LeCun, Y. Energy-Based Generative Adversarial Network. *arXiv* **2016**, arXiv:1609.03126. [CrossRef]
56. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv* **2017**, arXiv:1710.10196. [CrossRef]
57. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015. [CrossRef]
58. Landola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and < 0.5 MB Model Size. *arXiv* **2016**, arXiv:1602.07360. [CrossRef]
59. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
60. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556. [CrossRef]
61. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
62. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
63. StarGAN-v2. Available online: https://github.com/clovaai/stargan-v2-tensorflow (accessed on 1 November 2021).