

## Article

# Damage Detection and Isolation from Limited Experimental Data Using Simple Simulations and Knowledge Transfer

Asif Khan <sup>1</sup>, Jun-Sik Kim <sup>2</sup> and Heung Soo Kim <sup>1,\*</sup>

<sup>1</sup> Department of Mechanical, Robotics and Energy Engineering, Dongguk University-Seoul, 30 Pildong-ro 1 Gil, Jung-gu, Seoul 04620, Korea; asif\_dgu@dgu.edu

<sup>2</sup> Department of Mechanical System Engineering, Kumoh National Institute of Technology, Gumi-si 39177, Korea; junsik.kim@kumoh.ac.kr

\* Correspondence: heungsoo@dgu.edu; Tel.: +82-2-2260-8577; Fax: +82-2-2263-9379

**Abstract:** A simulation model can provide insight into the characteristic behaviors of different health states of an actual system; however, such a simulation cannot account for all complexities in the system. This work proposes a transfer learning strategy that employs simple computer simulations for fault diagnosis in an actual system. A simple shaft-disk system was used to generate a substantial set of source data for three health states of a rotor system, and that data was used to train, validate, and test a customized deep neural network. The deep learning model, pretrained on simulation data, was used as a domain and class invariant generalized feature extractor, and the extracted features were processed with traditional machine learning algorithms. The experimental data sets of an RK4 rotor kit and a machinery fault simulator (MFS) were employed to assess the effectiveness of the proposed approach. The proposed method was also validated by comparing its performance with the pre-existing deep learning models of GoogleNet, VGG16, ResNet18, AlexNet, and SqueezeNet in terms of feature extraction, generalizability, computational cost, and size and parameters of the networks.



**Citation:** Khan, A.; Kim, J.-S.; Kim, H.S. Damage Detection and Isolation from Limited Experimental Data Using Simple Simulations and Knowledge Transfer. *Mathematics* **2022**, *10*, 80. <https://doi.org/10.3390/math10010080>

Academic Editors: Maria Luminîța Scutaru and Catalin I. Pruncu

Received: 29 November 2021

Accepted: 24 December 2021

Published: 27 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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/>).

**Keywords:** computer simulations; actual systems; deep learning; transfer learning; autonomous feature extraction; machine learning

## 1. Introduction

Rotating machinery is a common and critical type of mechanical equipment used in a wide variety of modern industrial applications. Catastrophic failure of rotating machinery may result in substantial economic loss and injury to personnel. Turbines are key rotating parts of power plants and are susceptible to mechanical defects, such as unbalance [1,2], misalignment [3,4] rubbing [5,6], oil whirl [7], and oil whip [8,9], during operation. The presence of defects in turbines may cause performance degradation or even collapse of the entire system if not rectified in a timely manner. To ensure safe and reliable operation of rotating machinery, it is imperative that operators be able to promptly detect, isolate, and quantify different faults using vibration signals obtained through accelerometers or proximity sensors.

The most commonly used methods of fault diagnosis include model-based methods and data-driven methods [10–13]. In model-based methods, the physics underlying the system's behavior are modeled and used for fault diagnosis. It is difficult or even impossible to precisely model the behavior of complex systems, owing to the wide range of structural complexities and environmental uncertainties that affect such systems [14]. Data-driven methods use data obtained from sensors in the system to carry out fault diagnosis; these methods do not require much knowledge about the underlying kinematics and physics of the failure of the system [15,16]. In traditional data-driven fault diagnosis methods using machine learning, the signals from sensors are usually subjected to preprocessing (e.g., noise removal, domain transformation (time to frequency), signal decomposition (empirical

mode decomposition)), extraction of discriminative features (e.g., time and frequency domain statistical features), selection of features that are more sensitive to damage (e.g., feature ranking), and processing of the selected features with supervised or unsupervised machine learning algorithms [17]. The performance of machine learning algorithms for fault diagnosis is heavily dependent on the set of discriminative features that is selected [18]. A set of statistical features may work well for one problem and may fail completely for another problem in the same domain but on a different scale [19]. In general, there is no optimized set of processing steps for fault diagnosis using handcrafted statistical features from sensor data and machine learning algorithms. For instance, a data-driven diagnostic strategy that uses simulation data may not be generalized to a dataset from an experimental setup of the same problem without a complex process of model updating [20,21]. Moreover, the extraction of damage-sensitive features is labor-intensive and requires considerable diagnostic skills and domain expertise [22,23]. In addition, even an experienced diagnostic expert may spend a long time optimizing the set of discriminative features to diagnose a certain problem.

Deep learning has been successfully implemented for a variety of applications, such as image classification, speech recognition, computer vision, medical diagnosis, finance, marketing, and a multitude of other applications [23–26]. The inherent capability of deep learning models to automatically extract features from raw data to describe the underlying problem is one of their most celebrated benefits. Additionally, deep learning models can deal with unstructured data in different formats (e.g., texts, images, pdf files, etc.) to uncover the latent relationships between different data types and make important predictions [27,28]. In general, data-driven methods that use deep learning algorithms require sufficient data on the healthy and faulty states of the system for the development of robust and effective fault diagnosis strategies. Although data on the healthy state of a system is generally available in sufficient amounts, data on different defective states can be limited or even completely unavailable due to the high cost associated with running the machinery in the presence of defects. To make up for the dearth of failure data from different defective states of expensive machinery, computer simulations can be employed to generate a sufficient amount of healthy and faulty data using simplified mathematical models of the actual machinery [29]. However, there are gaps between the data from simulations and actual systems and a labor-intensive process is required to identify the parameters of the actual system and tune those parameters to bring the response characteristics of the simulation model closer to those of the actual system [30]. Additionally, despite the process of parameter identification from the actual system, it is not guaranteed that a diagnostic strategy developed from a simulation model will perform equally well for the detection, isolation, and quantification of different defects in the actual system. In general, the better a computer simulation represents the response behavior of an actual system, the greater its computational cost, and vice versa.

One way to bridge the gap between computer simulation and actual systems while keeping the simulation as simple as possible involves transfer learning or cross-domain knowledge transfer [31,32]. The fundamental idea of transfer learning is to leverage the knowledge from a semantically related problem to solve a new problem with a different domain distribution. In the general framework of transfer learning, a learning body learns the required properties and parameters from a source task with a substantial amount of labeled data and transfers/tunes those parameters to a target task with a limited amount of labeled or unlabeled data [12,33]. Generally, the source and target data have different statistical distributions [31,34]. Cao et al. [35] proposed transfer learning from a pretrained deep convolutional neural network (CNN) for fault diagnosis of a gearbox with limited data. The source domain consisted of a large number of labeled natural images and the target domain comprised graphical images from the gearbox vibration signals. Xu et al. [36] presented transfer CNNs for online fault diagnosis of bearings and pumps. In their work, related datasets were used to train several offline CNNs, then their shallow layers were transferred to an online CNN to improve its diagnostic performance. Yan et al. [37]

studied the application of knowledge transfer for fault diagnosis in rotary machines while considering the variation of working conditions, fault locations, types of machines, and different faults. Hasan and Kim [38] studied transfer learning for fault diagnosis of bearings under different working conditions. The only difference between the source and target tasks was the speed of rotation. Li et al. [39] studied the fault diagnosis of rolling bearings via deep convolution domain adversarial transfer learning. Zhang et al. [40] proposed a fault diagnosis strategy for bearings under different working conditions using transfer learning with neural networks. Huang et al. [41] presented a boosted algorithm (SharedBoost) to explore transfer learning for multiple data sources and compared its results with those of other transfer learning methods.

This paper attempted to employ simple simulation models of a rotor system to devise a robust and autonomous diagnostic strategy for actual rotating machinery. First principles were used to develop a simple two degree of freedom model of a rotor system with three types of defects (unbalance, parallel misalignment, and point rubbing). The simple rotor model was used to generate a large amount of vibration data by considering different operation speeds and different defect severity levels. For robustness, the simulation data was also contaminated with different levels of white Gaussian noise. The vibration signals from the simulation models were transformed into scalograms [42,43], which were then used to obtain a pretrained customized deep neural network. The pretrained network was employed as a generalized autonomous feature extractor from the experimental data sets of an RK4 rotor kit and a machinery fault simulator (MFS). The extracted features were processed with several conventional machine learning algorithms and an optimum classifier was identified. The performance of the customized deep learning network for autonomous feature extraction is also compared to that of other existing pretrained models, such as AlexNet [44], GoogleNet [45], ResNet [46], Vgg16 [47], etc. The proposed approach is invariant to the number of health states in the simulation and experimental domains, while no attempts are made to minimize the gap between the two domains.

## 2. Proposed Methodology

The limited nature of data from different defective states of actual machinery prohibits the use of deep learning models for autonomous feature extraction and diagnostics. Development of exact simulation models that replicate the response characteristics of the actual machinery is often computationally expensive. Although simple simulation models can provide insights into the characteristic behaviors of actual machinery in the presence of defects and are less computationally expensive, they do not account for all the uncertainties in the actual system. Transfer learning or cross-domain knowledge transfer could help to leverage the advantages of simple simulation models for fault diagnosis of the actual system. A schematic illustration of the basic idea of transfer learning in the context of the current problem is shown in Figure 1.

A large amount of source data is required to train, validate, and test a deep learning model with the highest possible degree of accuracy. That pretrained model can be used for automatic feature extraction from a limited dataset for a target task, or its weights and bias can be transferred to a limited target dataset using the concept of fine tuning [48,49]. In our case, the parameters of the model trained on simulation data are employed to extract high-level discriminative features from the target task of the experimental data. In transfer learning, the types of defects in the source data and target data are not necessarily the same [35,37,50]. A schematic illustration of the general workflow of the current work, which involves employing simple simulation models to detect, isolate, and quantify different types of defects in actual mechanical systems, is depicted in Figure 2.

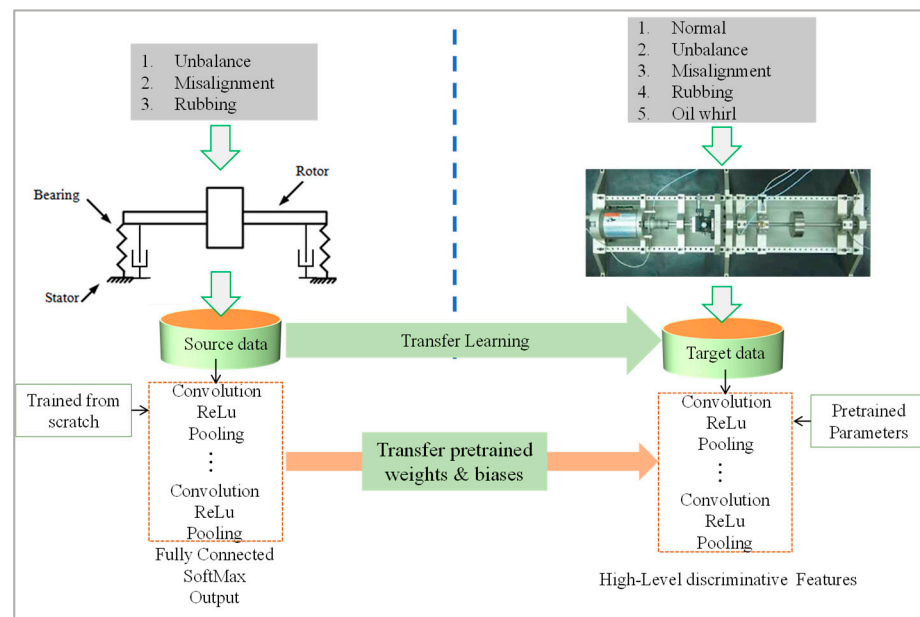


Figure 1. Fundamental idea of transfer learning.

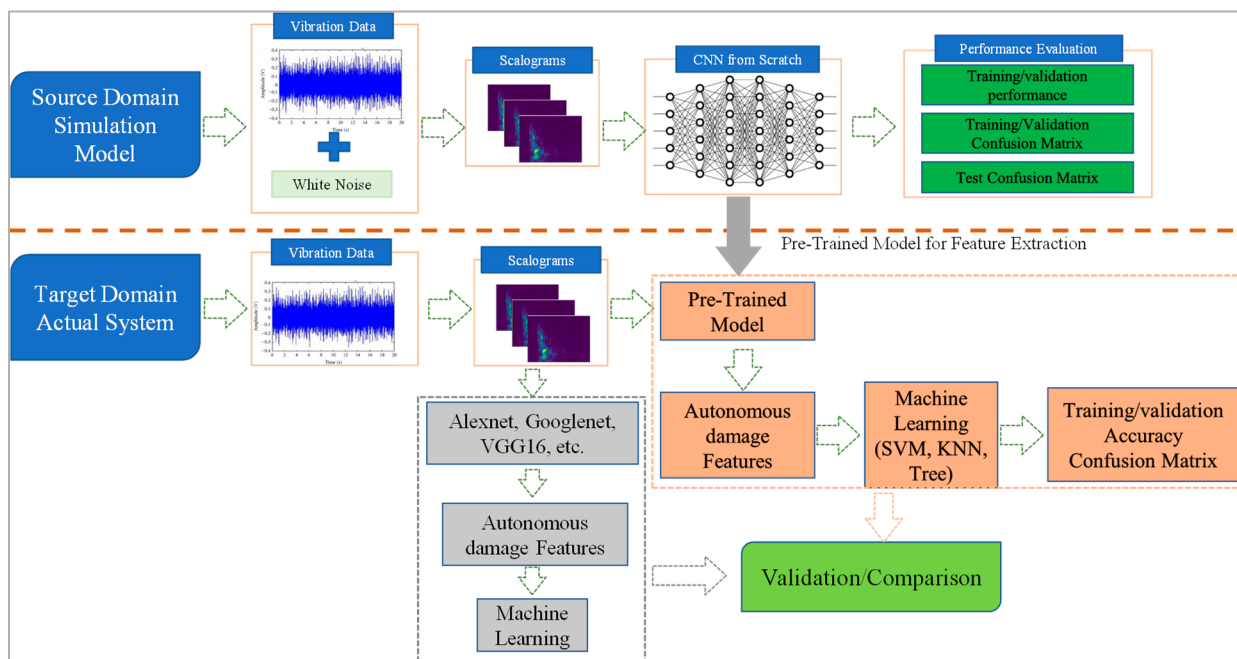


Figure 2. Overall workflow of the proposed methodology.

Herein, a simple simulation model was employed to generate a large amount of source data for the representative health states of the source task. For robustness, the simulation data was contaminated with different levels of white Gaussian noise. The noisy source data was transformed into scalograms via MATLAB and the scalograms were used to train, validate, and test a customized CNN.

The neural network trained on simulation data was used to automatically extract discriminative features from the response scalograms of the experimental data of real machines. The discriminative features were processed using traditional machine learning algorithms, such as support vector machine (SVM), tree classifier, K-nearest neighbor (KNN), etc. The results of the autonomous feature extraction via a customized neural network trained on simulation data are also compared with the results of feature extraction



via available pretrained deep learning models (e.g., Alexnet, GoogLeNet, VGG16) in terms of classification accuracy, generalization, computational cost, hardware requirement, etc. The proposed approach was validated for two datasets from an RK4 rotor kit by GE Bently Nevada (1631 Bently Parkway South, Minden, Nevada USA 89423) and a machinery fault simulator (MFS) by SpectraQuest (8227 Hermitage Road, Richmond, VA 23228 USA). Although, in the current work, the proposed approach was employed for the diagnosis of rotating machinery, this approach could be extended to the damage assessment of laminated composites, civil infrastructures, industrial robots, gearboxes, and others, where simple simulations could be developed to gain insights into the fault characteristics of the actual systems.

2.1. Simulation Model and Source Data Generation

As described in the previous sections, developing a simulation model that precisely matches the response characteristics of an actual system in the absence and presence of defects is either too computationally expensive or completely impossible for complex systems. Although simple simulation models of different types of actual machinery (e.g., a turbine simplified as a shaft-disk system) have been used to gain insight into the characteristics of various defects in the actual system, it is never a guarantee that the simulation models can be employed to assess damage in the actual system using a conventional approach. To bridge the gap between the actual systems and their simulated counterparts, transfer learning or cross-domain knowledge transfer provides a natural solution. However, transfer learning requires a large amount of data from the source task. This section describes the simple mathematical models of a shaft-disk system with different defects that were used to generate a large dataset for the source task. The simple rotor system considered in this work consists of a single disk of mass  $m$  mounted at the center of a shaft with length  $L$ , as shown in Figure 3.

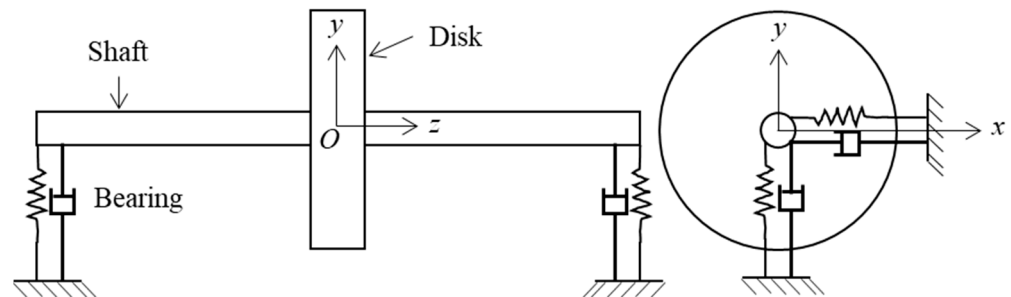


Figure 3. Simple rotor-disk system for the generation of a large amount of source data.

The shaft is supported by two bearings at the ends; the bearings are linearized, ideally with stiffness and damping. The support and/or foundation are assumed to be rigid. The dynamic response of the shaft-disk system is represented by a fixed coordinate system at the center of the disk. The system is characterized in terms of transverse displacements, and the vibration along the axis of the shaft is ignored. For the isotropic properties of the bearings at the two ends, and the disk mounted at the center of the shaft, the dynamics of the system in Figure 3 can be defined by a time-dependent equation, as follows:

$$\begin{aligned}
 m\ddot{x}(t) + c_{xT}\dot{x}(t) + k_{xT}x(t) &= F_x(t) \\
 m\ddot{y}(t) + c_{yT}\dot{y}(t) + k_{yT}y(t) &= F_y(t)
 \end{aligned}
 \tag{1}$$

where  $x$  and  $y$  are the displacements at the disk along the  $x$  and  $y$  axes, respectively,  $m$  is the mass of the disk,  $c_{xT}$  and  $c_{yT}$ , respectively, denote the total damping at the two bearings along the  $x$  and  $y$  axes, and  $k_{xT}$  and  $k_{yT}$  denote the total stiffness at the two bearings along the  $x$  and  $y$  axes, respectively. The terms  $F_x$  and  $F_y$  refer to the general forces acting on the system along the  $x$  and  $y$  direction, respectively. The anisotropic supports can be modeled using the approach proposed by Filippi et al. [51]. The characteristic behavior of

the forcing functions acting on the system depends on the type of defect in the system. In this work, three defects (unbalance, misalignment, and rubbing) of different magnitudes were considered in the system shown in Figure 3 to generate a large amount of source data. In practice, the pristine or healthy state of the rotating machinery has a small amount of residual unbalance that cannot be completely removed despite efforts at balancing. This small amount of residual unbalance is considered to be within the acceptable range (i.e., the system is considered to be healthy) if the amplitude of the vibration signals is within a certain level of the root mean square (rms) as prescribed by the standards of ISO 7919-2 [52–54]. Residual unbalance in the system exists when the center of mass is not coincident with the center of rotation. The motion equation used to simulate the residual unbalance in the rotor system is shown as follows in Equation (2):

$$\begin{aligned} m\ddot{x}(t) + c_{xT}\dot{x}(t) + k_{xT}x(t) &= me_r\Omega^2 \cos(\Omega t + \alpha) \\ m\ddot{y}(t) + c_{yT}\dot{y}(t) + k_{yT}y(t) &= me_r\Omega^2 \sin(\Omega t + \alpha) \end{aligned} \quad (2)$$

where  $e_r$  is the eccentricity between the center of mass and center of rotation,  $\alpha$  is the phase angle of residual unbalance, and  $\Omega$  is the rotational speed of the shaft.

The presence of unbalance, misalignment, and rubbing can be simulated as additional forces along with the residual unbalance. The forcing functions for the three defects are shown by Equations (3)–(5), respectively, as follows:

$$\begin{aligned} F_{x\_unb} &= m_a e_a \Omega^2 \cos(\Omega t + \beta) \\ F_{y\_unb} &= m_a e_a \Omega^2 \sin(\Omega t + \beta) \end{aligned} \quad (3)$$

$$\begin{aligned} F_{x\_mis} &= FX_2 \cos(\Omega t + \psi) + FX_2 \cos(2\Omega t + \psi) \\ F_{y\_mis} &= FY_2 \sin(\Omega t + \psi) + FY_2 \sin(2\Omega t + \psi) \end{aligned} \quad (4)$$

$$\begin{aligned} F_{x\_rub} &= -k_r(x - \delta_0)H(x - \delta_0) \\ F_{y\_rub} &= f k_r(x - \delta_0)H(x - \delta_0) \end{aligned} \quad (5)$$

where  $m_a$  is the added unbalance to the disk with an eccentricity of  $e_a$  and phase angle of  $\beta$ . The term  $\Omega$  denotes the speed of rotation. The terms  $FX_i$  and  $FY_i$  ( $i = 1, 2$ ) are the external forces due to parallel misalignment with a phase angle of  $\psi$ . The term  $k_r$  is the stiffness of the axial rub-impact rod,  $f$  is the friction coefficient of between the two parts,  $H$  is the Heaviside function, and  $\delta_0$  is the gap between the rotor and stator. Further details on the mathematical modeling can be found in Appendix A.

The mathematical models of unbalance (Equation (3)), misalignment (Equation (4)), and rubbing (Equation (5)) were employed to generate the large amount of source data necessary for the transfer learning strategy shown in Figure 2. The basic parameters of the three simulation models are given in Table 1.

Herein, the added unbalance was simulated with a fixed value of eccentricity ( $e_a$ ) by varying the value of the added mass from 1 to 20 g at increments of 2 g, misalignment was simulated with a parallel misalignment along the  $y$ -bending angular flexibility rate axis from 8 to 26 mm at increments of 2 mm, and rubbing was simulated by reducing the values of clearance between the rotor and stator from  $9.2 \times 10^{-8}$  to  $4.7 \times 10^{-8}$  m at decrements of  $0.5 \times 10^{-8}$  m. For the three defects (unbalance, misalignment, and rubbing) of the shaft-disk system, ten different levels of severity were considered, and each defective case of the system was operated at 50 different speeds of rotation from 300 to 6810 rpm at increments of 120 rpm. The steady-state vibration responses of the system were obtained along the  $x$  and  $y$  axes at the disk location by solving the differential equation (Equations (3)–(5)) of each defect via Newmark's time integration algorithm [55]. The number of steady-state responses for each defect with all severity levels was  $20 \times 50 = 1000$  samples.

**Table 1.** Material properties and parameters of the simulation models.

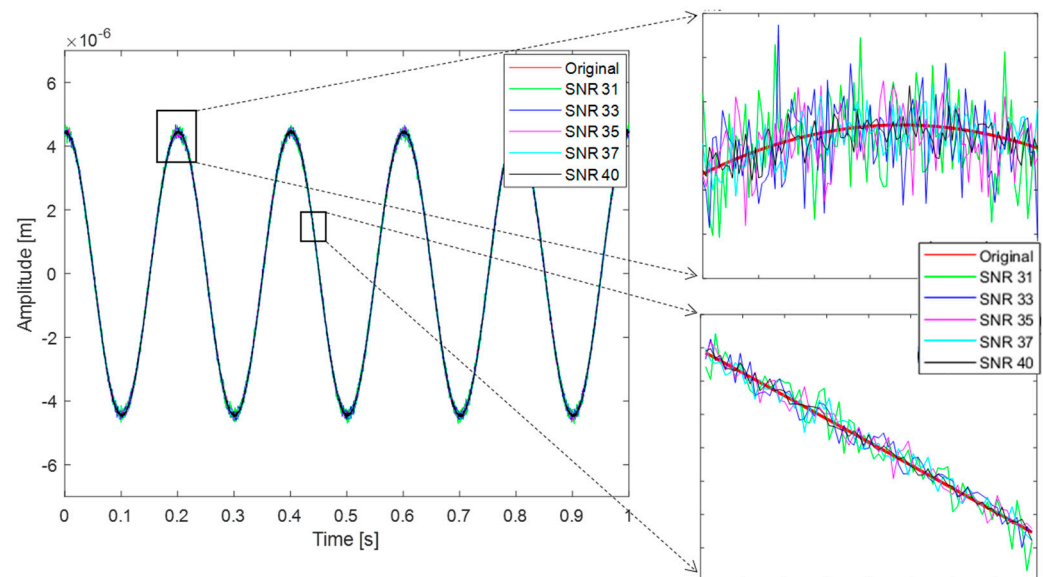
Model	Parameter	Value
General	Length of shaft ( $L$ )	1 m
	Modulus of elasticity ( $E$ )	$211 \times 10^9$ Pa
	Modulus of Rigidity ( $G$ )	$81.1 \times 10^9$ Pa
	Diameter of shaft ( $d_s$ )	0.01 m
	Diameter of disk ( $d$ )	0.075 m
	Thickness of disk ( $h$ )	0.0254 m
	Density of shaft and disk ( $\rho$ )	$7810 \text{ kg/m}^3$
	Mass of disk $m = \rho h \pi d^2/4$	0.8764 kg
	Stiffness at bearing 1 along $x$ -axis ( $k_{x1}$ )	$1.0 \times 10^6$ N/m
	Stiffness at bearing 1 along $y$ -axis ( $k_{y1}$ )	$1.0 \times 10^6$ N/m
	Stiffness at bearing 2 along $x$ -axis ( $k_{x2}$ )	$1.0 \times 10^6$ N/m
	Stiffness at bearing 2 along $y$ -axis ( $k_{y2}$ )	$1.0 \times 10^6$ N/m
	Damping at bearing 1 along $x$ -axis ( $c_{x1}$ )	1000 Ns/m
	Damping at bearing 1 along $y$ -axis ( $c_{y1}$ )	1000 Ns/m
Damping at bearing 2 along $x$ -axis ( $c_{x2}$ )	1000 Ns/m	
Damping at bearing 2 along $y$ -axis ( $c_{y2}$ )	1000 Ns/m	
Residual Unbalance	Mass eccentricity ( $e_r$ )	0.000015 m
	Phase angle ( $\alpha$ )	$0^\circ$
Unbalance	Added masses ( $m_a$ )	(1:2:20) g
	Phase angle ( $\beta$ )	$0^\circ$
Misalignment	Misalignment along $x$ -axis ( $\Delta X_1 = -\Delta X_2$ )	0 m
	Misalignment along $y$ -axis ( $\Delta Y_1 = -\Delta Y_2$ )	(8:2:26) mm
	Center of articulation ( $Z_3$ )	0.024 m
	Bending angular flexibility rate ( $K_b$ )	0.35 degree/Nm
	Power of Motor ( $P$ )	700 Watt
Rubbing	Clearance between rotor and stator ( $\delta_0$ )	$(9.2: -0.5:4.7) \times 10^{-8}$ m
	Stiffness of axial rub-impact rod ( $k_r$ )	$1.2 \times 10^7$ Pa
	Coefficient of friction ( $f$ )	0.7

To account for noise in the signals from actual systems, all steady-state responses of the three defects were added to white Gaussian noise with a signal-to-noise ratio (SNR) of 31 to 40 using the MATLAB function *awgn* (add white Gaussian noise to signal). The basic mathematical form of the *awgn* function is shown by Equation (6).

$$\begin{aligned}
 S_{noise} &= S + Z \\
 Z &\sim N(0, \mu)
 \end{aligned}
 \tag{6}$$

where  $S$  is the original signal without noise and  $Z$  refers to the random noise having normal/Gaussian distribution with zero mean and  $\mu$  variance.  $S_{noise}$  is the output signal contaminated with noise. Additional mathematical details of adding white Gaussian noise can be found in the MATLAB documentation and the published literature, as shown in the references [56–58]. The decision to use a range of SNR from 31 to 40 was made after looking at the effect of different SNR values on the original signals obtained from simulations. The

effect of different values of SNR on the original signal of 9 g unbalance at 300 rpm is shown in Figure 4.



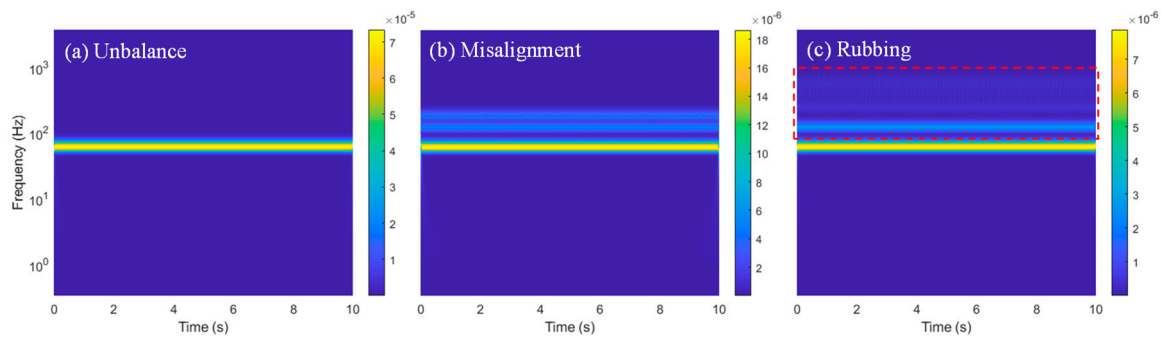
**Figure 4.** Effect of different values of sound-to-noise-ratio (SNR) on the signals obtained from the simulation model.

As shown in Figure 4, it was observed that the SNR range of 31 to 40 accounts for the higher and lower levels of noise in the source simulation data.

This noise contamination of the 1000 steady-state signals of each defect resulted in  $20 \times 50 \times 10 = 10,000$  samples for each defect. The steady-state response signals of the three defects with and without noise were combined, resulting in 33,000 samples (11,000 samples for each defect) that served as the source data from the simulation model.

## 2.2. Deep Learning Model for Simulation Data

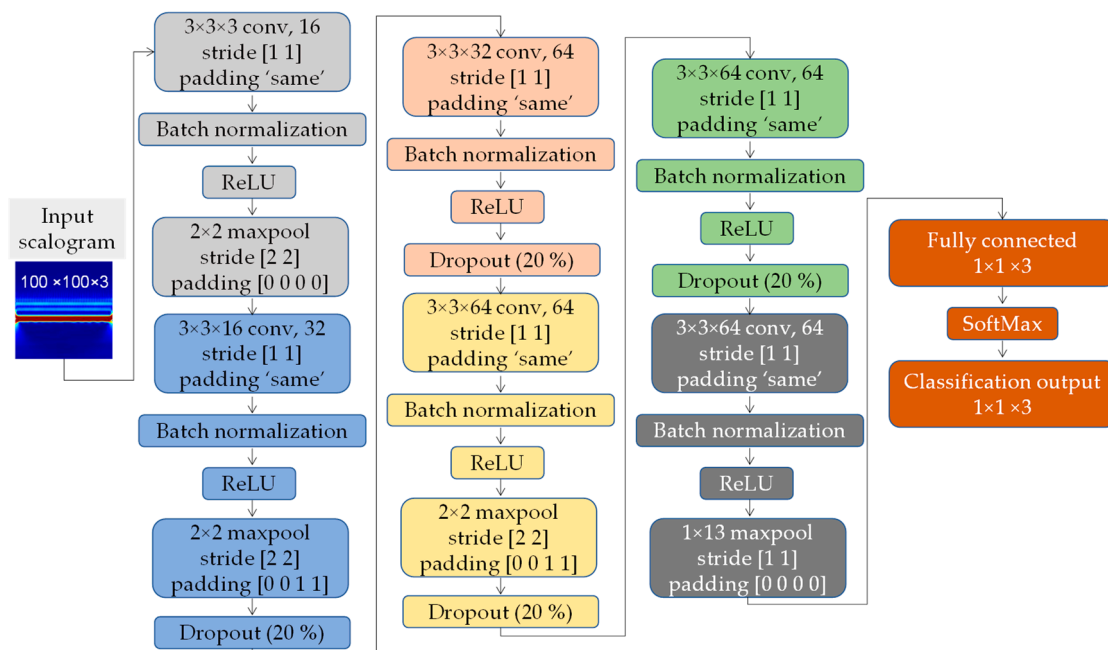
The 33,000 response signals from the simulation model were transformed into scalograms using continuous wavelet transform (CWT). A scalogram is essentially a time-frequency representation of a time domain signal that is generated from the absolute value of the CWT coefficients of that signal. The mathematical details regarding the transformation of a time series to a scalogram using wavelet analysis can be found in the references [59,60]. In this work, MATLAB was used to design a CWT filter bank with a sampling frequency of 8500 Hz (the same as the signal acquisition frequency) and the default number of voices per octave (10 wavelet bandpass filters per octave) [61]. The analytic Morse wavelet with the default values of the symmetry parameter and time-bandwidth product was used in the filter bank [62–64]. More details on the parametric study of the effect of the parameters of wavelet transform can be found in the references [65–67]. The filter bank was used to transform all the time series from the simulation models to scalograms. Figure 5 depicts some samples of unbalance, misalignment, and rubbing scalograms for a given speed of rotation out of 33,000 scalograms from the simulation data.



**Figure 5.** Sample unbalance, misalignment, and rubbing scalograms in the simulation model at a steady state of 3660 rpm with  $y$ -axis on a logarithmic scale; (a) Unbalance; (b) Misalignment; (c) Rubbing.

The scalograms of the three defects have distinct characteristics in the time-frequency domain. In general, the presence of unbalance, misalignment, and rubbing in a rotating system are characterized by the presence of distinct frequency spectra at 1X (speed of rotation) [68], frequency spectra at 1X and the integral multiples thereof (2X, 3X), and frequency spectra at 1X and its sub- and super-harmonics depending on the speed of rotation [69,70], respectively. The general characteristics of unbalance, misalignment, and rubbing can be observed in the scalograms in Figure 5, where the presence of unbalance, misalignment, and rubbing are shown by a distinct frequency component at the speed of rotation, the speed of rotation and its integral multiples, and by super harmonics (dashed red rectangle), respectively. In addition, note that the  $y$ -axis is a logarithmic scale.

The scalograms of the source data from the simulation were used to train, validate, and test a convolutional neural network (CNN). Figure 6 depicts the detailed architecture of the CNN used in the current work.



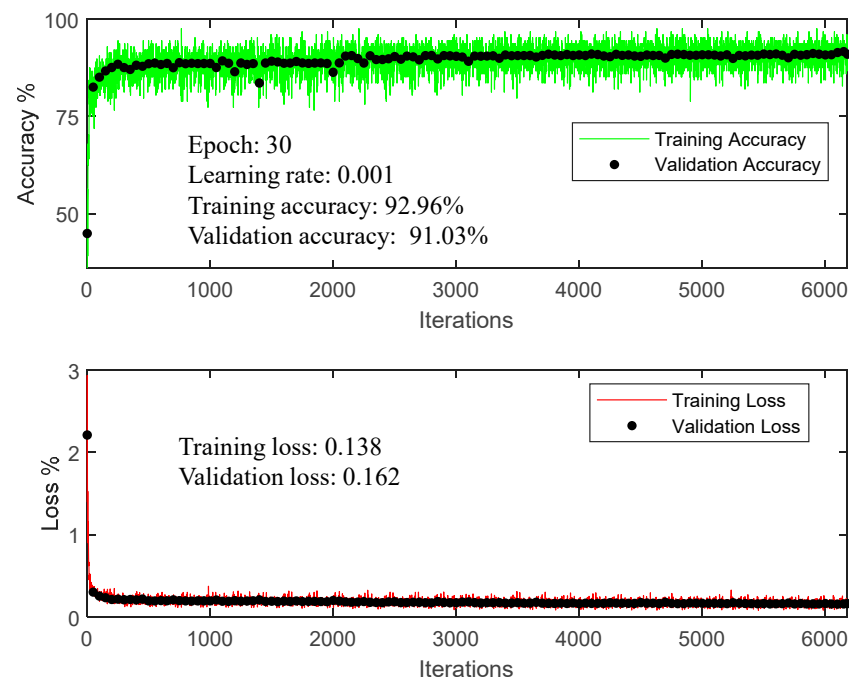
**Figure 6.** Architecture of the convolutional neural network used in the current study.

In the CNN architecture, convolutional, batch normalization, and ReLU layers are used to extract high-level features from the input scalograms, and max pooling layers are employed to down-sample those features [35]. A dropout layer is inserted to minimize the chances of overfitting during the training process [71]. The classification layer adopts a SoftMax function [72,73] to classify the extracted features into three different classes:



unbalance, misalignment, and rubbing in the shaft-disk system. In the current architecture, the max pooling layers in the first, second, and fourth hidden layers were used to account for invariances in the simulation scalograms. Since the pretrained model was to be employed as a generalized feature extractor from the experimental data, the max pooling layers in the third and fifth hidden layers were excluded to accommodate the local variations in the autonomous features of the target task.

To train the CNN, the weights were randomly initialized and tuned from scratch using Adam optimizer as an optimization function. The data set of 33,000 scalograms was split into 80% training, 10% validation during the training, and 10% independent test datasets. To avoid memory problems, the scalograms were loaded in the form of an image data store using the function “imageDatastore” in MATLAB. Figure 7 shows the accuracy and loss for the training and validation of the network.



**Figure 7.** Training and validation of the customized deep learning model using simulation data.

Here, 80% of the data (training data) was used to train the CNN, while 10% of the data (validation data) was used to evaluate the performance of the model at each iteration of the training process. The training/validation accuracy refers to the classification/validation accuracy for each mini batch of the training/validation dataset. The training/validation loss indicates the performance of the model after each iteration of optimization and denotes the sum of errors for each example of the training/validation data. From Figure 7, the overlap between the training and validation accuracies and losses as well as the validation accuracy of 91.5% imply that the network is optimally learning from the training data and could be generalized to unseen instances.

To verify the generalization of the pretrained CNN to an unseen data set, the model was tested on the remaining 10% of the dataset (testing data). Figure 8 shows the test confusion matrix.

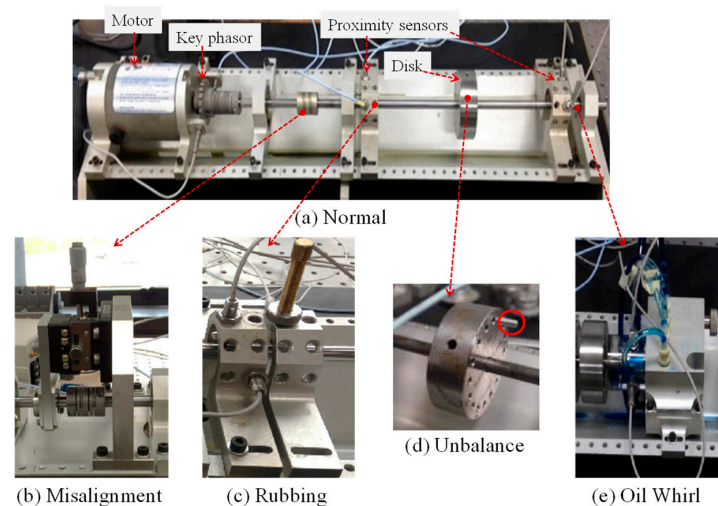


**Figure 8.** Test confusion matrix showing the performance of the pretrained deep learning model on unseen simulation test data (90.27% accuracy on test data).

As shown in Figure 8, the pretrained network successfully identified the presence of misalignment and rubbing with 100% accuracy; however, it confused 29.2% instances (321 observations) of unbalance with misalignment. The reason for the confusion between unbalance and misalignment is that the misalignment model in Equation (4) only simulates parallel misalignment along the  $y$ -axis, resulting in misaligned response characteristics along the  $y$ -axis and unbalance response characteristics along the  $x$ -axis. For lower values of added unbalance, the unbalance response from the misalignment model along the  $x$ -axis and the actual added unbalance will be confused. Thus, 29.2% instances of unbalance were confused with misalignment.

*2.3. Experimental Data*

Two experimental data sets were employed to validate the effectiveness of the proposed approach. The first experimental vibration data for different health states of the shaft-disk system was obtained from an RK4 rotor kit, a product of GE Bently Nevada (1631 Bently Parkway South, Minden, Nevada USA 89423). The vibration signals were obtained via proximity sensors for the following health states: normal (residual unbalance), unbalance, rubbing, misalignment, and oil whirl. The experimental configuration of the different health states of the rotor system is shown in Figure 9.



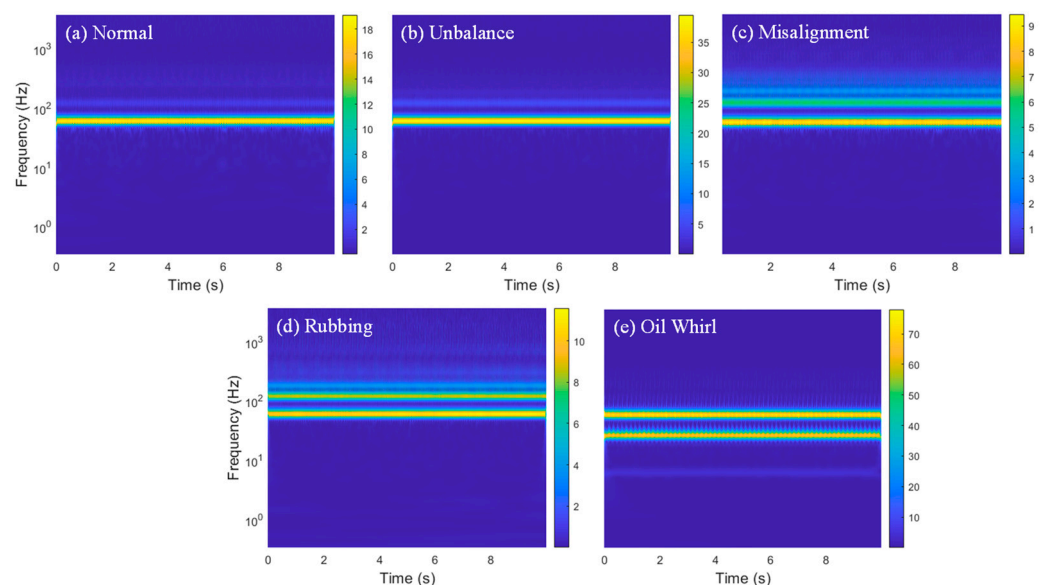
**Figure 9.** RK4 Rotor kit with different health states; (a) Normal; (b) Misalignment; (c) Rubbing; (d) Unbalance; (e) Oil Whirl.

Despite efforts to perfectly balance the system in the normal state, there existed a small amount of unbalance in the system; the amplitude of the resulting vibration signal was within the acceptable range of  $10\ \mu\text{m}$  of the root mean square (rms) level, as determined by the ISO standard 7919-2.

The unbalance state was induced by attaching a 15 g screw to the disk (Figure 9d). A special jig (Figure 9b) was employed to induce a parallel misalignment of  $20\ \mu\text{m}$  along the  $y$ -axis at the coupling location. The rubbing state was simulated with a rubbing screw (Figure 9c) that contacted the shaft when a 15 g mass was attached to the disk. The position of the rubbing screw was adjusted such that the shaft contacted the rubbing screw once per revolution at 3600 rpm (steady-state condition for all health conditions). An additional tool kit (Figure 9e) was used to induce the oil whirl phenomenon at an oil pressure of 35 kPa.

Two sets of proximity sensors placed near each bearing were used to acquire the vibration response signals for all the health states of the rotor kit; for each set of proximity sensors, the two were installed at right angles along the  $x$  and  $y$  axes. To ensure repeatability and account for experimental uncertainty, each health state was executed five times, and the rotor kit was reassembled before each experiment. All five health states of RK4 were studied at a steady-state condition of 3600 rpm. The dataset for all health states consisted of 100 signals, with 20 signals for each case (4 signals for each health state  $\times$  5 executions of each experiment).

The CWT filter bank designed for the simulation data was used to transform the vibration signals from the RK4 rotor kit for all health states to scalograms without any preprocessing. We aimed to gain insight into the characteristics of different defects and compare the results with the outcomes of the simple simulations. Figure 10 depicts sample scalograms of the experimental data for the different health states.



**Figure 10.** Sample scalograms of different health states in the RK4 Rotor kit at steady state of 3600 rpm: (a) normal; (b) unbalance; (c) misalignment; (d) rubbing; and (e) oil whirl ( $y$ -axis is a logarithmic scale).

In Figure 10, some high-frequency contents are observed in the scalogram of the normal state, implying either the presence of noise or some other small unavoidable defects alongside the small unbalance in the system. Additionally, comparing the scalograms from the experimental data with their simulated counterparts shows that the scalograms of the experimental data demonstrate more complex behavior in terms of time-frequency content, which confirms that extremely simple mathematical models cannot replicate the exact dynamic response behavior of an actual system with and without defects. Furthermore, the health states of normal and oil whirl were not considered in the source simulation data.

In the next section, the CNN model pretrained on simulation data is used to automatically extract discriminative features from the scalograms of experimental data.

2.4. Autonomous Feature Extraction Using Pretrained Models

In transfer learning, a model developed and trained for one task is reused as a starting point for another related task, without expending much time or computational resources [33]. As stated previously, the inner layers of a CNN autonomously extract high-level features from the input images and use those features in the last fully connected and classification layers to distinguish between different classes of input images. In the architecture of a pretrained network, some layers can be eliminated to retrieve the high-level features from layer activation, and those features can be processed with traditional machine learning algorithms, as depicted in Figure 11.

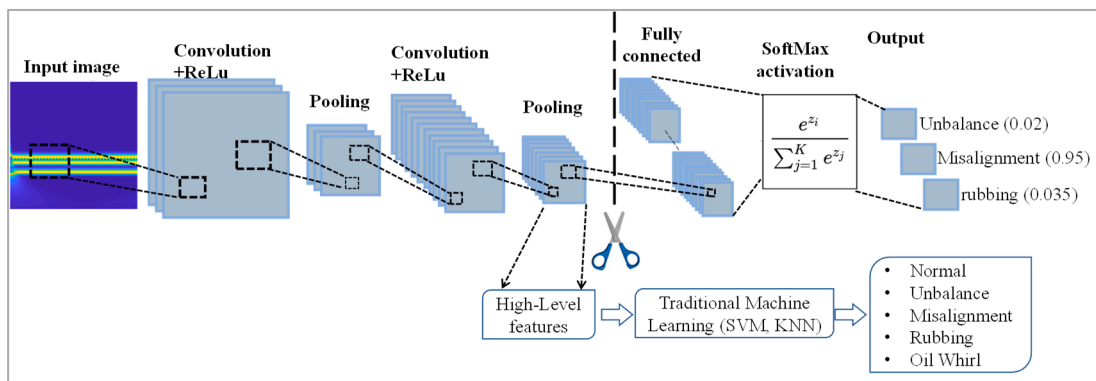


Figure 11. Autonomous feature extraction via a pretrained deep learning model.

The automatically extracted high-level features can be used to train, validate, and test traditional machine learning algorithms, such as SVM, tree classifier, KNN, etc. In this work, the activations from the last max pooling layer of the CNN trained on simulation data were used as discriminative features for the scalograms of the experimental data from the RK4 rotor kit. The autonomously extracted features were processed with several different machine learning classifiers; Figure 12 shows a comparison of the different classifiers in terms of overall classification accuracy and area under the ROC (receiver operating characteristic) curve. The ROC area is obtained by graphing the true and false positive rates and its value implies a tradeoff between recall and fallout. An ROC area close to 1 indicates that the model is able to achieve a high recall (true positive rate) while maintaining a low fallout (false positive rate) [74].

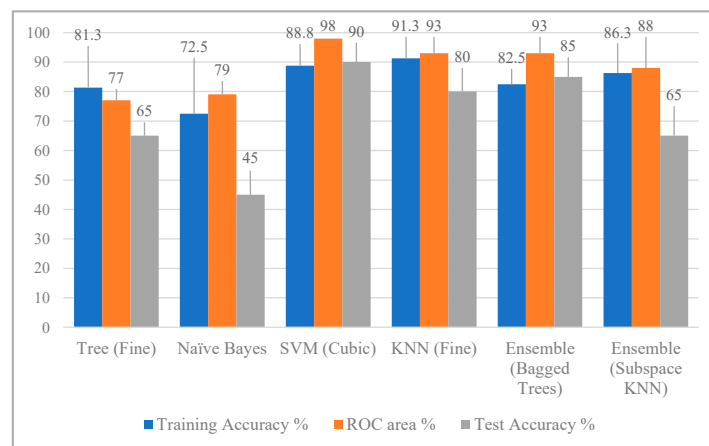
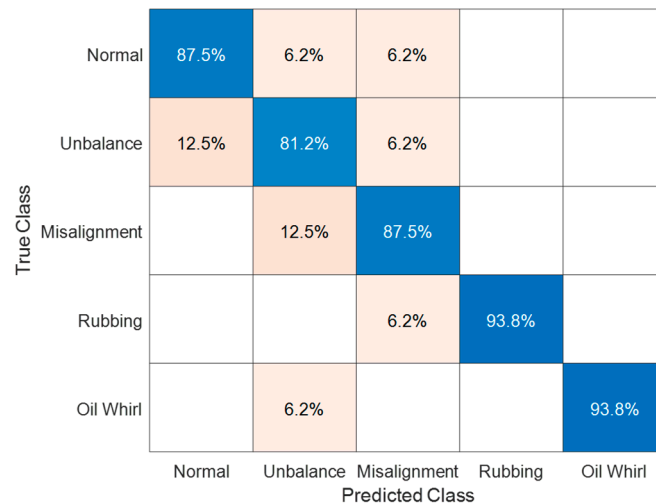


Figure 12. Performance of different classifiers using the automatically extracted features.

As shown in Figure 12, the minimum and maximum training accuracies were 72.5% and 91.3%, for the naïve Bayes and KNN classifiers, respectively. However, the overall training accuracy could be deceiving, and the model may have overfitted the training data. The matrices of ROC area and prediction results on an independent test set would help to fully explore the behavior of the supervised learning classifiers.

In Figure 12, SVM stands out as the optimum classifier in terms of training accuracy (88.8%), ROC area (98%), and test accuracy (90%). Figure 13 shows the confusion matrix of SVM on the 80% training dataset, created to gain further insight into the classification performance of SVM.



**Figure 13.** Training/validation confusion matrix of cubic SVM on the features automatically extracted by the pretrained deep learning model from the original RK4 data.

During the training process, the classifier confused 6.2% of the instances of normal as unbalance and misalignment, 12.5% of the instances of unbalance as normal, 6.2% of the instances of unbalance as misalignment, 12.5% of the instances of misalignment as unbalance, 6.2% of the instances of rubbing as misalignment, and 6.2% of the instances of oil whirl as unbalance. Here, 6.2% and 12.5% instances refer to one and two observations, respectively. According to the training confusion matrix, the loss of accuracy was mainly due to the confusion of 12.5% instances of misalignment with unbalance and 12.5% instances of unbalance with the normal state. The physical reason for this confusion is that only parallel misalignment was induced along the *y*-axis, while the response along the *x*-axis is purely due to residual unbalance that may coincide with the added unbalance. Similarly, a possible explanation for confusing unbalance with the normal state is that the two share the same response characteristics and only differ in amplitude. The domain and class invariance of the proposed approach is verified from the high classification accuracy of the health states of normal and oil whirl, which were not considered in the source simulation domain.

The results of the pretrained cubic SVM on the unseen test dataset are shown in the form of a confusion matrix in Figure 14.



True Class	Normal	100.0%				
	Unbalance		100.0%			
	Misalignment			75.0%	25.0%	
	Rubbing			25.0%	75.0%	
	Oil Whirl				100.0%	
		Normal	Unbalance	Misalignment	Rubbing	Oil Whirl
		Predicted Class				

Figure 14. Test confusion matrix of cubic SVM on features automatically extracted by the pretrained deep learning model from the original RK4 data.

As shown in Figure 14, 25% of instances of misalignment were confused with rubbing and 25% of the instances of rubbing were confused with misalignment. The results of the test confusion matrix are within an acceptable range, as 25% of instances is equivalent to one observation out of four from the 20% test data.

Given the above discussion, the results of autonomous feature extraction via a CNN that was pretrained on simulation data are physically reasonable. However, the limited size of the training and the test datasets make it difficult to draw a general conclusion. One option is to obtain more data from the testbed by repeating the experiments; however, the experiments have already been repeated five times.

Another option is to employ the concept of virtual sensors around the shaft, as introduced by Jung et al. [75], to artificially augment the data without performing any further experiments. In this work, the concept of virtual sensors is adopted to synthetically augment the experimental data. The main idea of virtual sensors is to obtain synthetic vibration signals from the vibration signals of the actual orthogonal proximity sensors by rotating the cartesian coordinate system with respect to the z-axis, as depicted in Figure 15.

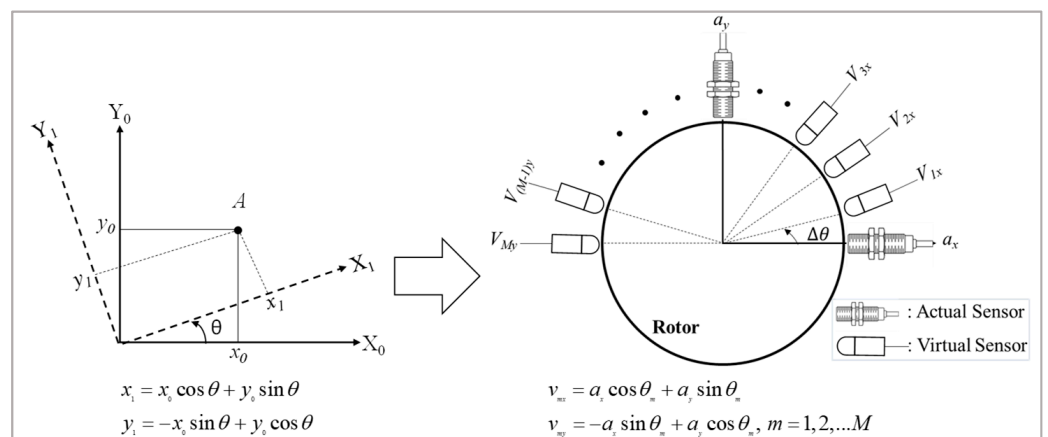
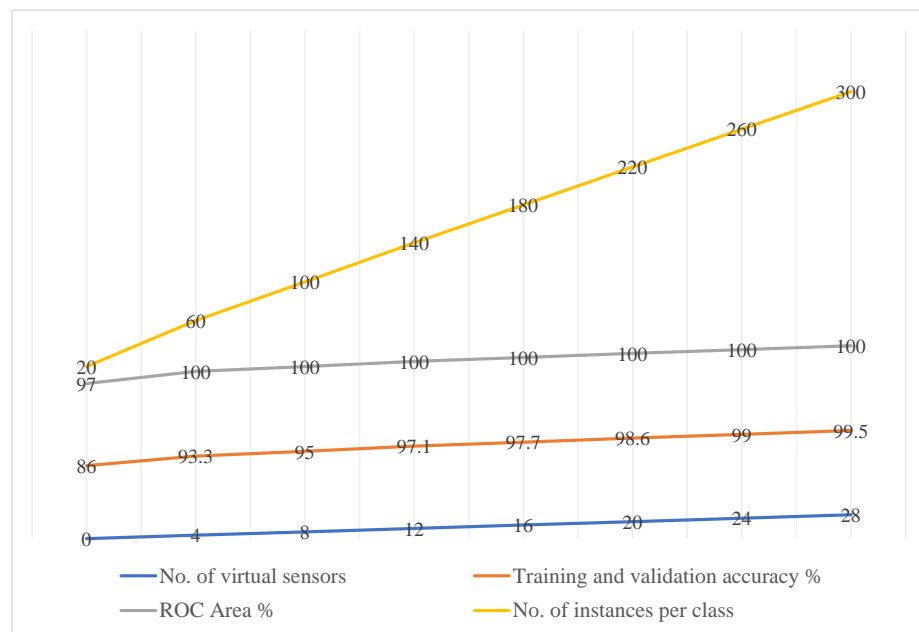


Figure 15. Concept of virtual sensors based on simple transformation of coordinates.

The virtual signals are obtained from the actual signals using the following coordinate transformation:

$$\begin{aligned}
 x_{Vm} &= \cos(m\Delta\theta)x_a + \sin(m\Delta\theta)y_a \\
 y_{Vm} &= -\sin(m\Delta\theta)x_a + \cos(m\Delta\theta)y_a \\
 (m &= 1, 2, \dots, M)
 \end{aligned}
 \tag{7}$$

where  $x_{Vm}$  and  $y_{Vm}$  are virtual signals along the rotated  $x$  and  $y$  axes, respectively. The terms  $a_x$  and  $a_y$  refer to the actual signals obtained via the proximity sensors along the original  $x$  and  $y$  axes, respectively,  $\Delta\theta$  is the angle of rotation for the coordinate system of virtual signals, and  $M$  denotes the number of virtual signals. Owing to symmetry around the shaft, the maximum number of virtual sensors is  $M = \pi/\Delta\theta$ . As shown in a previous paper [75],  $x_{Vm}$  is equal to  $y_{Vm+M/2}$ ; hence, in this work only  $x_{Vm}$  was retained from Equation (6) for synthetic data augmentation. To identify the optimum number of virtual sensors for the current task, a parametric study was carried out for different numbers of virtual sensors and the effect was evaluated terms of training/validation accuracy, ROC area, and the number of instances per class as a result of data augmentation, as shown in Figure 16.



**Figure 16.** Different numbers of virtual sensors and their effect on the classification performance and size of dataset.

To obtain the results shown in Figure 16, the original and augmented datasets were transformed into scalograms and processed via the pretrained CNN to extract discriminative features. A cubic SVM was employed to classify the extracted features into different classes using 10-fold cross-validation. The results showed that the training and validation accuracy could be increased to 99.5% by synthetic data augmentation using virtual sensors; however, the increase in the evaluation matrices of training/validation accuracy and ROC are relatively small compared with the increase in the size of the augmented dataset. Thus, because of this tradeoff between the size of the augmented dataset and classification accuracy, the number of virtual sensors was set at 12 for further analysis. To provide more insight into the problem, the augmented data was split into 80% training and 20% test data. Figure 17 shows the per class training/validation performance of the cubic SVM on the training data in the form of a confusion matrix with 97.3% accuracy.

True Class	Normal	96.4%		0.9%	2.7%	
	Unbalance	4.5%	93.8%	0.9%	0.9%	
	Misalignment			99.1%	0.9%	
	Rubbing			1.8%	98.2%	
	Oil Whirl					100.0%
		Normal	Unbalance	Misalignment	Rubbing	Oil Whirl
		Predicted Class				

**Figure 17.** Training/validation confusion matrix of SVM (cubic) on automatically extracted features from the pretrained deep learning model with 12 virtual sensors on RK4.

The cubic SVM was trained via 10-fold cross-validation on the synthetically augmented data using 12 virtual sensors. In Figure 17, the higher true positive rate and the lower false positive rate for each class demonstrate the optimum performance of the proposed methodology on the experimental data set. Additionally, note that the per class classification performance increased compared with the results from the data without augmentation in Figure 13. To show that the vibration signals synthesized through virtual sensors did not cause overfitting of the machine learning model, the cubic SVM pretrained on the augmented data (synthesized and measured) was employed to make predictions on the 20% unseen test data. Here, the unseen data describes a data set not seen by the network during the training/validation process. The pretrained model showed a test accuracy of 97.14%, with the confusion matrix shown in Figure 18.

True Class	Normal	96.4%		3.6%		
	Unbalance	3.6%	89.3%	3.6%	3.6%	
	Misalignment			100.0%		
	Rubbing				100.0%	
	Oil Whirl					100.0%
		Normal	Unbalance	Misalignment	Rubbing	Oil Whirl
		Predicted Class				

**Figure 18.** Test confusion matrix of SVM (cubic) on automatically extracted features from the pretrained deep learning model with 12 virtual sensors on RK4.

As shown in Figure 18, the test accuracy on the augmented data increased from 90% to 97.14% in comparison with the performance on the measured data, which would not have been possible in the case of overfitting due to synthesized signals.

To further explore the robustness of the proposed approach and its ability to bridge the gap between simple computer simulations and actual experiments, the deep learning model trained on simulation data was compared with pre-existing deep learning mod-

els of GoogleNet, Vgg16, ResNet18, AlexNet, and SqueezeNet [76] in terms of feature extraction. The pre-existing deep learning models are trained and optimized on natural images and have fixed network architectures [31]. The image dataset that is commonly employed to train the existing pretrained networks is usually a subset of the ImageNet database [77]. For instance, Vgg16 is pretrained on approximately 1.5 million images with 41 layers, and Alexnet is pretrained on approximately 1.2 million images with eight layers and 60 million parameters. To verify the performance of the customized deep learning model for autonomous feature extraction from a limited amount of experimental data, the performance of the cubic SVM on autonomously extracted features from the simulation model was compared with the performance of Googlenet, Vgg16, Resnet18, Alexnet, and Squeezenet, as shown in Table 2.

**Table 2.** Comparison of the customized pretrained simulation model with existing pretrained deep learning models (12 virtual sensors and cubic SVM).

	Training/Validation Accuracy %	ROC Area%	Testing Accuracy %
Simulation Model	97.5	100	97.14
GoogleNet	93.8	99	83.57
Vgg16	95	100	83.5
Resnet18	97.1	100	86.43
Alexnet	95.2	99	85.71
SqueezeNet	90.2	99	81.43

For the results in Table 2, the features extracted by all the deep learning models were split into 80:20 for training and testing, respectively. The 80% training data was used to train a cubic SVM through 10-fold cross-validation, and the resulting trained model was employed to make predictions on the 20% test dataset. According to the results shown in Table 2, all the deep learning models performed reasonably well in terms of training/validation accuracy, ROC area, and test accuracy, which validates the performance of the customized deep learning model.

The results shown in Table 2 bring up an obvious question: if the existing pretrained models perform equally well on the limited experimental dataset, then why bother using a simulation dataset and a customized deep learning model?

The motivation behind the customized deep learning model is that the existing pretrained networks (AlexNet, VGG16) have fixed architectures, a fixed number of parameters, and limited flexibility for controlling the dimensions of the extracted discriminative features, whereas the customized deep learning model offers more flexibility in terms of network size, number of parameters, and dimensions of the extracted discriminative features. Furthermore, as seen from the test classification accuracy of the simulation model in Table 2, a deep learning model pretrained on source data that resembles the target data of the transfer learning scheme would provide better generalizability. To further support the effectiveness of the proposed approach, the autonomous feature extraction through a customized deep learning model for the data of 12 virtual sensors was compared with the feature extraction through the pre-existing deep learning models in terms of size of the network, parameters of the network, and computational time, as shown in Table 3.

**Table 3.** Comparison of customized deep learning model with pre-existing deep learning models.

Name	Size of Network (Bytes)	Number of Network Parameters	Computational Time (sec)	
			CPU *	GPU **
Simulation Model	631,037	139,587	4.542	1.65
GoogleNet	29,670,809 (191.7%)	6,698,552 (192.2%)	25.66 (139.8%)	2.55 (42.7%)
Vgg16	554,895,306 (199.5%)	138,357,54 (199.6%)	149.47 (188.2%)	6.33 (117.2%)
Resnet18	47,156,446 (194.7%)	11,694,312 (195.2%)	23.56 (135.3%)	2.33 (34.1%)
Alexnet	245,283,524 (198.9%)	60,965,224 (199.0%)	10.14 (76.3%)	1.86 (12.12)
SqueezeNet	5,232,394 (156.9%)	1,235,496 (159.4%)	17.59 (117.9%)	1.89 (13.3%)

\* CPU: Intel i7-4790 with 32 GB RAM, \*\* GPU: NVIDIA GeForce RTX 2080 Ti.

In Table 3, the percentage value in each cell is the percentage of the difference between the value for the customized deep learning model and that of a pre-existing deep learning model. One can observe that the customized deep learning model with relatively simple architecture outperformed the pre-existing deep models developed and trained by experts with a massive amount of training data.

In addition, as seen from the computation time, the problem-specific customized deep learning model has more potential for practical implementation with less hardware requirements than pre-existing deep learning models. Furthermore, in the framework of transfer learning, the input data to the pretrained models must be of the same size as that of the original data used during the pretraining of the network (e.g., image size, number of channels etc.); resizing a data set as per the requirement of pre-existing deep learning models may remove significant information in terms of the image size reduction or image size increment. However, such issues could be easily handled with a customized deep learning model specifically designed, trained, and transferred for a given engineering problem as achieved in the current work.

In the previous discussion, the experimental data set from RK4 rotor kit consisted of five health states at a steady state speed of 3600 rpm and a single severity level of each health state. To further verify the robustness of the proposed approach, a more extensive data set from SpectraQuest's machinery fault simulator (MFS) [78] kit was employed. In this work, five health states (normal, horizontal misalignment, unbalance, outer race fault in bearing, and rolling element fault in bearing) with different speeds of operation (49 speeds for each health state) and different severity levels (two severity levels) of each health state were considered. Furthermore, the bearing defects were studied in the presence of a 6 g and 35 g unbalanced mass. A detailed discussion of the data set can be referred to in reference [79] and it is available for download at the website in reference [80]. The vibration signals from the MFS were transformed into scalograms using the same filter bank as used for the data from the simulation models and RK4 rotor kit. The deep learning model pretrained on simulation data was employed to extract discriminative features from the scalograms of the experimental data from the MFS kit and SVM was employed to classify those features into different classes. The SVM classifier was trained through 10-fold cross-validation and Figure 19 shows its training/validation confusion matrix on the 90% training data with a classification accuracy of 97.8%.



Nor	99.0%	0.1%	0.1%	0.4%	0.1%	0.1%		0.1%	0.1%
UB 10	0.8%	94.6%	2.7%	0.5%	0.8%	0.2%	0.1%	0.3%	0.1%
UB 25	0.2%	2.9%	95.6%	0.5%	0.3%		0.3%	0.1%	0.1%
HM 0.5	0.4%	0.3%	0.2%	98.1%	0.6%	0.2%	0.1%	0.1%	
HM 20	0.1%	0.4%	0.3%	0.6%	98.2%	0.1%		0.1%	0.1%
BRE 6U	0.1%	0.1%		0.1%		99.3%	0.1%	0.3%	0.1%
BRE 35U	0.1%	0.1%	0.1%	0.1%		0.3%	98.3%	0.2%	0.9%
BOR 6U	0.1%	0.3%		0.1%	0.1%	0.2%		98.7%	0.4%
BOR 35U		0.2%		0.2%	0.1%	0.2%	0.5%	0.6%	98.3%
	Nor	UB 10	UB 25	HM 0.5	HM 20	BRE 6U	BRE 35U	BOR 6U	BOR 35U

Figure 19. Training/validation confusion matrix of SVM on automatically extracted features through the pretrained deep learning model from the data of the MFS kit.

To verify that the SVM did not overfit the training data, Figure 20 shows the testing performance in the form of a confusion matrix on the 10% independent test data with 97.31% accuracy.

Nor	97.7%	0.6%	0.6%	0.6%	0.6%				
UB 10		97.1%	2.9%						
UB 25		6.5%	91.7%	0.6%		0.6%			0.6%
HM 0.5		0.6%		98.3%					1.1%
HM 20			1.1%	0.6%	97.2%	0.6%			0.6%
BRE 6U				0.6%		98.9%			0.6%
BRE 35U						1.4%	98.0%		0.7%
BOR 6U		1.1%			0.6%			97.7%	0.6%
BOR 35U				0.7%					99.3%
	Nor	UB 10	UB 25	HM 0.5	HM 20	BRE 6U	BRE 35U	BOR 6U	BOR 35U

Figure 20. Test confusion matrix of SVM on features automatically extracted by the pretrained deep learning model from the data of the MFS kit.

In Figure 20, the tags should be interpreted as follows: HM 0.5: horizontal misalignment of 0.5 mm; HM 20: horizontal misalignment of 20 mm; UB 10: unbalance with 10 g mass; UB 25: unbalance with 25 g mass; Nor: normal; BRE 35U: bearing with rolling element fault and 35 g unbalance mass; BRE 6U: bearing with rolling element fault and 6 g unbalance mass; BOR 35U: bearing with outer race fault and 35 g unbalance mass; and BOR 6U: bearing with outer race fault and 6 g unbalance mass.

The results show that the model can distinguish different health states and their severity levels with a minimum accuracy of 94.6% for UB 10 and a maximum accuracy of 99.3% for BRE 6U. The misclassification results are within the acceptable range. The high accuracy of the model on the features extracted through the simulation model shows that the model is robust to different speeds of operations (49 different speeds for each health state) and that the extracted features are only sensitive to the presence of defects in the rotor system. Additionally, the high accuracy on the bearing faults in the presence of different unbalance loads confirms the robustness of the model to different loads. Furthermore,

the target domain class invariance of the customized deep learning model is verified from the high accuracy on the bearing faults, which were not considered in the source simulation data.

From the test confusion matrix of Figure 20, the high accuracy of 97.31% on the 10% independent test data shows that the SVM model pretrained on the discriminative features of the deep learning model did not overfit the training data. The essence of the current work is that a domain invariant generalized feature extractor developed from simple simulations can accommodate the gap in the response characteristics and new health states in the target domain in a supervised learning framework.

### 3. Conclusions

This work proposed a domain and class invariant generalized feature extractor using a supervised learning framework of transfer learning. A source simulation domain with three health states was employed to detect, isolate, and quantify five health states in the target experimental domain without minimizing the gap in the response characteristics of the two domains. The source domain was comprised of the simulation model of a few representative health states of the target domain, and simulation models were not required for all prospective health states of the actual target system. The proposed methodology relies on transfer learning, where a customized deep learning model is trained, validated, and tested on a substantial set of simulation data, and then the pretrained model is employed to autonomously extract discriminative features from a small experimental target dataset. This work also discussed the synthetic augmentation of the limited experimental data using virtual sensors, where the output from the virtual sensors was defined in terms of the actual sensors using the concept of coordinate transformation. Synthetic augmentation of the experimental data enhanced the performance of the proposed approach in terms of training/validation accuracy (from 88.8% to 99.5%), test accuracy (90% to 97.14%), and ROC area (from 97% to 100%). The effectiveness of the proposed approach was validated by comparing its results with the pre-existing deep learning models of GoogleNet, VGG16, ResNet18, AlexNet, and SqueezeNet in terms of training, testing, generalization, size of the network, parameters of the network, and computational time. The current approach was found to perform relatively better in terms of generalizability and computation cost with more flexibility for a given engineering problem.

The proposed approach autonomously extracts discriminative features from the vibration-based scalograms of a limited experimental dataset and eliminates the need for labor-intensive hand-crafted statistical features. In addition, the source simulation signals and target experimental signals are directly transformed into scalograms using a single filter bank that eliminates the need for complex preprocessing. The generalized autonomous discriminative features are robust to variations in the operating conditions, severity levels of different health states, and scale of the source and target domains. This work could be extended to assess faults in laminated composites, gearboxes, industrial robots, civil infrastructures, etc.

**Author Contributions:** Conceptualization, H.S.K., J.-S.K. and A.K.; methodology, A.K. and H.S.K.; software, A.K.; validation, A.K. and H.S.K.; formal analysis, A.K.; investigation, A.K.; resources, H.S.K.; writing—original draft preparation, A.K.; writing—review and editing, A.K., H.S.K. and J.-S.K.; supervision, H.S.K.; project administration, H.S.K.; funding acquisition, H.S.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF-2020R1A2C1006613), funded by the Ministry of Education, and was also conducted as part of a research project (R17GA08) of the Korea Electric Power Corporation.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

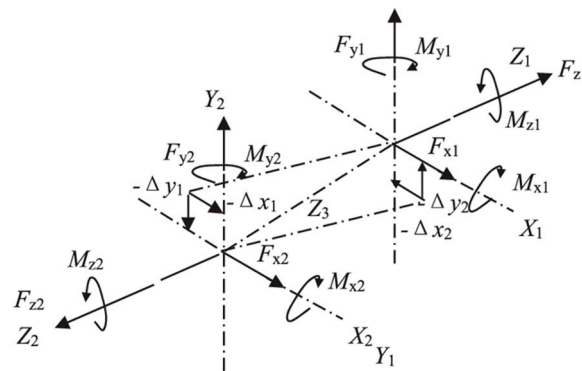
**Conflicts of Interest:** The authors declare no conflict of interest.

**Appendix A**

To simulate the added unbalance in the system, which represents unbalance that is commonly encountered in practice, a small mass of magnitude  $m_a$  is attached at an eccentricity of  $e_a$  and phase angle of  $\beta$  to the disk, causing a harmonic centrifugal force of magnitude  $m_a \times e_a \times \Omega^2$  along the  $x$  and  $y$  axes of the system when the system rotates at speed  $\Omega$ . The motion of the system in the presence of added unbalance is expressed as follows:

$$\begin{aligned} m\ddot{x}(t) + c_{xT}\dot{x}(t) + k_{xT}x(t) &= me_r\Omega^2 \cos(\Omega t + \alpha) + m_a e_a \Omega^2 \cos(\Omega t + \beta) \\ m\ddot{y}(t) + c_{yT}\dot{y}(t) + k_{yT}y(t) &= me_r\Omega^2 \sin(\Omega t + \alpha) + m_a e_a \Omega^2 \sin(\Omega t + \beta) \end{aligned} \tag{A1}$$

Misalignment is another common defect in rotating machinery. Misalignment in the coupled machine shafts generates reaction forces in the coupling [4]. In this work, the Gibbons [5] model was adopted to simulate the presence of misalignment in the rotor-disk system of Figure 3. A schematic of the Gibbons model of parallel misalignment is shown in Figure A1 [81,82].



**Figure A1.** Schematic of the Gibbons parallel misalignment model. Reprinted with permission from ref. [82]. Copyright 2021 Elsevier.

Here,  $Z_1$  and  $Z_2$ , respectively, denote the centerlines of the driver and driven shafts, which are offset by  $\Delta Y$  along the vertical direction and by  $\Delta X$  along the horizontal direction. The term  $Z_3$  denotes the coupling center of articulation;  $M_X$ ,  $M_Y$ , and  $M_Z$  are the three moments; and  $F_X$ ,  $F_Y$ , and  $F_Z$  are the three reaction forces. The moments and forces exerted by coupling on the driver and driven shafts are shown by Equation (A2).

$$\begin{aligned} \theta_1 &= \sin^{-1}(\Delta X_1/Z_3), \theta_2 = \sin^{-1}(\Delta X_2/Z_3) \\ \phi_1 &= \sin^{-1}(\Delta Y_1/Z_3), \phi_2 = \sin^{-1}(\Delta Y_2/Z_3) \\ M_{X1} &= T_q \sin \theta_1 + K_b \phi_1, M_{X2} = T_q \sin \theta_2 - K_b \phi_2 \\ M_{Y1} &= T_q \sin \phi_1 - K_b \theta_1, M_{Y2} = T_q \sin \phi_2 + K_b \theta_2, \\ F_{X1} &= (-M_{Y1} - M_{Y2})/Z_3, F_{X2} = -F_{X1} \\ F_{Y1} &= (M_{X1} + M_{X2})/Z_3, F_{Y2} = -F_{Y1} \end{aligned} \tag{A2}$$

where  $K_b$  is the bending angular flexibility rate of the flexible coupling and  $T_q$  is the torque of the rotor shaft, which is calculated in terms of motor power  $P$  and speed of rotation  $\Omega$ , as given by Equation (A3).

$$P = T_q \times \Omega \tag{A3}$$

The moments and forces of Equation (A2) appear as periodic forces with  $1\Omega$  and  $2\Omega$  components, and the equation of motion in the presence of parallel misalignment is modified as follows:

$$\begin{aligned}
 m\ddot{x}(t) + c_{xT}\dot{x}(t) + k_{xT}x(t) &= me_r\Omega^2 \cos(\Omega t + \alpha) + FX_2 \cos(\Omega t + \psi) \\
 &\quad + FX_2 \cos(2\Omega t + \psi) \\
 m\ddot{y}(t) + c_{yT}\dot{y}(t) + k_{yT}y(t) &= me_r\Omega^2 \sin(\Omega t + \alpha) + FY_2 \sin(\Omega t + \psi) \\
 &\quad + FY_2 \sin(2\Omega t + \psi)
 \end{aligned}
 \tag{A4}$$

where  $\psi$  is the phase angle. In addition, note that, besides the misalignment forces, residual unbalance is present in the system, as shown by the first term on the right side of Equation (A4).

To simulate the rubbing phenomenon between the rotor and stator, it is assumed that a single rub-impact occurs at the disk location, as shown by the schematic in Figure A2.

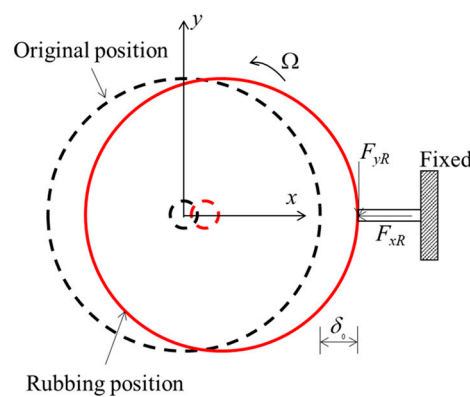


Figure A2. Schematic of a single rub-impact between the rotor and stator.

It is assumed that there is a small gap of  $\delta_0$  between the rotor and stator. The rub-impact occurs when the axial displacement of the shaft due to unbalance is larger than  $\delta_0$ . The equation of motion in the presence of a single-rub impact is given by Equation (A5):

$$\begin{aligned}
 m\ddot{x}(t) + c_{xT}\dot{x}(t) + k_{xT}x(t) &= me_r\Omega^2 \cos(\Omega t + \alpha) + F_{xR}(x, y) \\
 m\ddot{y}(t) + c_{yT}\dot{y}(t) + k_{yT}y(t) &= me_r\Omega^2 \sin(\Omega t + \alpha) + F_{yR}(x, y)
 \end{aligned}
 \tag{A5}$$

where the terms  $F_{xR}$  and  $F_{yR}$  denote the nonlinear forces along the  $x$  and  $y$  axes, respectively, due to the single rub-impact between the rotor and stator and are expressed as follows [83]:

$$\begin{aligned}
 F_{xR} &= -k_r(x - \delta_0)H(x - \delta_0) \\
 F_{yR} &= fk_r(x - \delta_0)H(x - \delta_0)
 \end{aligned}
 \tag{A6}$$

where  $k_r$  is the stiffness of the axial rub-impact rod,  $f$  is the coefficient of friction between the rotor and stator, and  $H$  is the Heaviside function, which is expressed as follows:

$$H(x - \delta_0) = \begin{cases} 0 & \text{if } x < \delta_0 \\ 1 & \text{if } x \geq \delta_0 \end{cases}
 \tag{A7}$$

References

1. Sudhakar, G.; Sekhar, A.S. Identification of Unbalance in a Rotor Bearing System. *J. Sound Vib.* **2011**, *330*, 2299–2313. [CrossRef]
2. Jain, J.R.; Kundra, T.K. Model Based Online Diagnosis of Unbalance and Transverse Fatigue Crack in Rotor Systems. *Mech. Res. Commun.* **2004**, *31*, 557–568. [CrossRef]
3. Tonks, O.; Wang, Q. The Detection of Wind Turbine Shaft Misalignment Using Temperature Monitoring. *CIRP J. Manuf. Sci. Technol.* **2017**, *17*, 71–79. [CrossRef]
4. Verma, A.K.; Sarangi, S.; Kolekar, M.H. Experimental Investigation of Misalignment Effects on Rotor Shaft Vibration and on Stator Current Signature. *J. Fail. Anal. Prev.* **2014**, *14*, 125–138. [CrossRef]

5. Al-bedoor, B.O. Transient Torsional and Lateral Vibrations of Unbalanced Rotors with Rotor-to-Stator Rubbing. *J. Sound Vib.* **2000**, *229*, 627–645. [[CrossRef](#)]
6. Jiang, J.; Ulbrich, H.; Chavez, A. Improvement of Rotor Performance under Rubbing Conditions through Active Auxiliary Bearings. *Int. J. Non-Linear Mech.* **2006**, *41*, 949–957. [[CrossRef](#)]
7. Luo, L.-Y.; Fan, Y.-H.; Tang, J.-H.; Chen, T.-Y.; Zhong, N.-R.; Feng, P.-C.; Kao, Y.-C. Frequency Enhancement of Oil Whip and Oil Whirl in a Ferrofluid–Lubricated Hydrodynamic Bearing–Rotor System by Magnetic Field with Permanent Magnets. *Appl. Sci.* **2018**, *8*, 1687. [[CrossRef](#)]
8. El-Shafei, A.; Tawfick, S.H.; Raafat, M.S.; Aziz, G.M. Some Experiments on Oil Whirl and Oil Whip. *J. Eng. Gas Turbines Power* **2007**, *129*, 144–153. [[CrossRef](#)]
9. Schweizer, B. Oil Whirl, Oil Whip and Whirl/Whip Synchronization Occurring in Rotor Systems with Full-Floating Ring Bearings. *Nonlinear Dyn.* **2009**, *57*, 509–532. [[CrossRef](#)]
10. Gao, Z.; Cecati, C.; Ding, S.X. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis with Model-Based and Signal-Based Approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [[CrossRef](#)]
11. Mansouri, M.; Harkat, M.-F.; Nounou, H.N.; Nounou, M.N. *Data-Driven and Model-Based Methods for Fault Detection and Diagnosis*; Elsevier: Amsterdam, The Netherlands, 2020.
12. Wang, X.; Shen, C.; Xia, M.; Wang, D.; Zhu, J.; Zhu, Z. Multi-Scale Deep Intra-Class Transfer Learning for Bearing Fault Diagnosis. *Reliab. Eng. Syst. Saf.* **2020**, *202*, 107050. [[CrossRef](#)]
13. Xu, Z.; Saleh, J.H. Machine Learning for Reliability Engineering and Safety Applications: Review of Current Status and Future Opportunities. *Reliab. Eng. Syst. Saf.* **2021**, *211*, 107530. [[CrossRef](#)]
14. Lu, B.; Li, Y.; Wu, X.; Yang, Z. A Review of Recent Advances in Wind Turbine Condition Monitoring and Fault Diagnosis. In Proceedings of the 2009 IEEE power electronics and machines in wind applications, Lincoln, NE, USA, 24–26 June 2009; IEEE: Piscataway Township, NJ, USA, 2009; pp. 1–7.
15. 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](#)]
16. Gómez, M.J.; Castejón, C.; García-Prada, J.C. Automatic Condition Monitoring System for Crack Detection in Rotating Machinery. *Reliab. Eng. Syst. Saf.* **2016**, *152*, 239–247. [[CrossRef](#)]
17. Khan, S.; Yairi, T. A Review on the Application of Deep Learning in System Health Management. *Mech. Syst. Signal Process.* **2018**, *107*, 241–265. [[CrossRef](#)]
18. Shukla, S.; Yadav, R.N.; Sharma, J.; Khare, S. Analysis of Statistical Features for Fault Detection in Ball Bearing. In Proceedings of the 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICIC), Madurai, India, 10–12 December 2015; pp. 1–7.
19. Oh, H.; Jung, J.H.; Jeon, B.C.; Youn, B.D. Scalable and Unsupervised Feature Engineering Using Vibration-Imaging and Deep Learning for Rotor System Diagnosis. *IEEE Trans. Ind. Electron.* **2017**, *65*, 3539–3549. [[CrossRef](#)]
20. Mottershead, J.E.; Friswell, M.I. Model Updating in Structural Dynamics: A Survey. *J. Sound Vib.* **1993**, *167*, 347–375. [[CrossRef](#)]
21. Cavalini, A.A., Jr.; Lobato, F.S.; Koroishi, E.H.; Steffen, V., Jr. Model Updating of a Rotating Machine Using the Self-Adaptive Differential Evolution Algorithm. *Inverse Probl. Sci. Eng.* **2016**, *24*, 504–523. [[CrossRef](#)]
22. Immovilli, F.; Bianchini, C.; Cocconcelli, M.; Bellini, A.; Rubini, R. Bearing Fault Model for Induction Motor with Externally Induced Vibration. *IEEE Trans. Ind. Electron.* **2012**, *60*, 3408–3418. [[CrossRef](#)]
23. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
24. Bakator, M.; Radosav, D. Deep Learning and Medical Diagnosis: A Review of Literature. *Multimodal Technol. Interact.* **2018**, *2*, 47. [[CrossRef](#)]
25. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep Learning for Computer Vision: A Brief Review. *Comput. Intell. Neurosci.* **2018**, *2018*, 7068349. [[CrossRef](#)]
26. Pierson, H.A.; Gashler, M.S. Deep Learning in Robotics: A Review of Recent Research. *Adv. Robot.* **2017**, *31*, 821–835. [[CrossRef](#)]
27. Morozov, V.; Petrovskiy, M. An Approach for Complex Event Streams Processing and Forecasting. In Proceedings of the 2020 26th Conference of Open Innovations Association (FRUCT), Yaroslavl, Russia, 20–24 April 2020; IEEE: Piscataway Township, NJ, USA, 2020; pp. 305–313.
28. Dey, L.; Meisheri, H.; Verma, I. Predictive Analytics with Structured and Unstructured Data—a Deep Learning Based Approach. *IEEE Intell. Inform. Bull.* **2017**, *18*, 27–34.
29. Gecgel, O.; Ekwaro-Osire, S.; Dias, J.P.; Serwadda, A.; Alemayehu, F.M.; Nispel, A. Gearbox Fault Diagnostics Using Deep Learning with Simulated Data. In Proceedings of the 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 17–20 June 2019; IEEE: Piscataway Township, NJ, USA, 2019; pp. 1–8.
30. Tiwari, R. *Rotor Systems: Analysis and Identification*; CRC Press: Boca Raton, FL, USA, 2017.
31. Zheng, H.; Wang, R.; Yang, Y.; Yin, J.; Li, Y.; Li, Y.; Xu, M. Cross-Domain Fault Diagnosis Using Knowledge Transfer Strategy: A Review. *IEEE Access* **2019**, *7*, 129260–129290. [[CrossRef](#)]
32. Li, C.; Zhang, S.; Qin, Y.; Estupinan, E. A Systematic Review of Deep Transfer Learning for Machinery Fault Diagnosis. *Neurocomputing* **2020**, *407*, 121–135. [[CrossRef](#)]
33. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]



34. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Transfer Learning Using Deep Representation Regularization in Remaining Useful Life Prediction across Operating Conditions. *Reliab. Eng. Syst. Saf.* **2021**, *211*, 107556. [CrossRef]
35. Cao, P.; Zhang, S.; Tang, J. Preprocessing-Free Gear Fault Diagnosis Using Small Datasets with Deep Convolutional Neural Network-Based Transfer Learning. *IEEE Access* **2018**, *6*, 26241–26253. [CrossRef]
36. Xu, G.; Liu, M.; Jiang, Z.; Shen, W.; Huang, C. Online Fault Diagnosis Method Based on Transfer Convolutional Neural Networks. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 509–520. [CrossRef]
37. Yan, R.; Shen, F.; Sun, C.; Chen, X. Knowledge Transfer for Rotary Machine Fault Diagnosis. *IEEE Sens. J.* **2020**, *20*, 8374–8393. [CrossRef]
38. Hasan, M.J.; Kim, J.-M. Bearing Fault Diagnosis under Variable Rotational Speeds Using Stockwell Transform-Based Vibration Imaging and Transfer Learning. *Appl. Sci.* **2018**, *8*, 2357. [CrossRef]
39. Li, F.; Tang, T.; Tang, B.; He, Q. Deep Convolution Domain-Adversarial Transfer Learning for Fault Diagnosis of Rolling Bearings. *Measurement* **2021**, *169*, 108339. [CrossRef]
40. Zhang, R.; Tao, H.; Wu, L.; Guan, Y. Transfer Learning with Neural Networks for Bearing Fault Diagnosis in Changing Working Conditions. *IEEE Access* **2017**, *5*, 14347–14357. [CrossRef]
41. Huang, P.; Wang, G.; Qin, S. Boosting for Transfer Learning from Multiple Data Sources. *Pattern Recognit. Lett.* **2012**, *33*, 568–579. [CrossRef]
42. Byeon, Y.-H.; Pan, S.-B.; Kwak, K.-C. Intelligent Deep Models Based on Scalograms of Electrocardiogram Signals for Biometrics. *Sensors* **2019**, *19*, 935. [CrossRef] [PubMed]
43. Peng, Z.K.; Chu, F.L.; Tse, P.W. Detection of the Rubbing-Caused Impacts for Rotor–Stator Fault Diagnosis Using Reassigned Scalogram. *Mech. Syst. Signal Process.* **2005**, *19*, 391–409. [CrossRef]
44. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: New York, NJ, USA, 2012; pp. 1097–1105.
45. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. 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; IEEE: Piscataway Township, NJ, USA, 2015; pp. 1–9.
46. 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; IEEE: Piscataway Township, NJ, USA, 2016; pp. 770–778.
47. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
48. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How Transferable Are Features in Deep Neural Networks? In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: New York, NK, USA, 2014; pp. 3320–3328.
49. Guo, Y.; Shi, H.; Kumar, A.; Grauman, K.; Rosing, T.; Feris, R. Spottune: Transfer Learning through Adaptive Fine-Tuning. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4805–4814.
50. Shao, S.; McAleer, S.; Yan, R.; Baldi, P. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2446–2455. [CrossRef]
51. Filippi, M.; Carrera, E. Stability and Transient Analyses of Asymmetric Rotors on Anisotropic Supports. *J. Sound Vib.* **2021**, *500*, 116006. [CrossRef]
52. Standard, I.S.O. *Mechanical Vibration-Evaluation of Machine Vibration by Measurements on Non-Rotating Parts*; ISO/IS: 1996; p. 10816. Available online: <https://www.iso.org/obp/ui/#iso:std:iso:7919:-1:ed-2:v1:en> (accessed on 18 February 2021).
53. Standardization, I.O. for ISO 20816-1: 2016 (En), *Mechanical Vibration—Measurement and Evaluation of Machine Vibration—Part 1: General Guidelines*. 2016. Available online: <https://www.iso.org/obp/ui/#iso:std:iso:20816:-1:ed-1:v1:en> (accessed on 18 February 2021).
54. ISO 16084:2017(En), *Balancing of Rotating Tools and Tool Systems*. Available online: <https://www.iso.org/obp/ui/#iso:std:iso:16084:ed-1:v1:en> (accessed on 15 January 2021).
55. Hashamdar, H.; Ibrahim, Z.; Jameel, M. Finite Element Analysis of Nonlinear Structures with Newmark Method. *IJPS* **2011**, *6*, 1395–1403. [CrossRef]
56. Jondral, F.K. White Gaussian Noise—Models for Engineers. *Frequenz* **2018**, *72*, 293–299. [CrossRef]
57. Liu, N.; Schumacher, T. Improved Denoising of Structural Vibration Data Employing Bilateral Filtering. *Sensors* **2020**, *20*, 1423. [CrossRef] [PubMed]
58. Add White Gaussian Noise to Signal—MATLAB Awgn. Available online: <https://www.mathworks.com/help/comm/ref/awgn.html> (accessed on 23 December 2021).
59. Benítez, R.; Bolós, V.J.; Ramírez, M.E. A Wavelet-Based Tool for Studying Non-Periodicity. *Comput. Math. Appl.* **2010**, *60*, 634–641. [CrossRef]
60. Türk, Ö.; Özerdem, M.S. Epilepsy Detection by Using Scalogram Based Convolutional Neural Network from EEG Signals. *Brain Sci.* **2019**, *9*, 115. [CrossRef]
61. Lee, D.T.; Yamamoto, A. Wavelet Analysis: Theory and Applications. *Hewlett Packard J.* **1994**, *45*, 44–52.

62. Olhede, S.C.; Walden, A.T. Generalized Morse Wavelets. *IEEE Trans. Signal Process.* **2002**, *50*, 2661–2670. [[CrossRef](#)]
63. Lilly, J.M.; Olhede, S.C. Higher-Order Properties of Analytic Wavelets. *IEEE Trans. Signal Process.* **2008**, *57*, 146–160. [[CrossRef](#)]
64. Lilly, J.M.; Olhede, S.C. Generalized Morse Wavelets as a Superfamily of Analytic Wavelets. *IEEE Trans. Signal Process.* **2012**, *60*, 6036–6041. [[CrossRef](#)]
65. Silik, A.; Noori, M.; Altabey, W.A.; Ghiasi, R.; Wu, Z. Comparative Analysis of Wavelet Transform for Time-Frequency Analysis and Transient Localization in Structural Health Monitoring. *Struct. Durab. Health Monit.* **2021**, *15*, 1–22. [[CrossRef](#)]
66. De Moortel, I.; Munday, S.A.; Hood, A.W. Wavelet Analysis: The Effect of Varying Basic Wavelet Parameters. *Sol. Phys.* **2004**, *222*, 203–228. [[CrossRef](#)]
67. Bolós, V.J.; Benítez, R. The Wavelet Scalogram in the Study of Time Series. In *Advances in Differential Equations and Applications*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 147–154.
68. Yamamoto, G.K.; da Costa, C.; da Silva Sousa, J.S. A Smart Experimental Setup for Vibration Measurement and Imbalance Fault Detection in Rotating Machinery. *Case Stud. Mech. Syst. Signal Process.* **2016**, *4*, 8–18. [[CrossRef](#)]
69. Chen, S.; Yang, Y.; Peng, Z.; Wang, S.; Zhang, W.; Chen, X. Detection of Rub-Impact Fault for Rotor-Stator Systems: A Novel Method Based on Adaptive Chirp Mode Decomposition. *J. Sound Vib.* **2019**, *440*, 83–99. [[CrossRef](#)]
70. Ma, H.; Tai, X.; Sun, J.; Wen, B. Analysis of Dynamic Characteristics for a Dual-Disk Rotor System with Single Rub-Impact. *Adv. Sci. Lett.* **2011**, *4*, 2782–2789. [[CrossRef](#)]
71. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
72. Nwankpa, C.; Ijomah, W.; Gachagan, A.; Marshall, S. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *arXiv* **2018**, arXiv:1811.03378.
73. Khan, A.; Sohail, A.; Zahoor, U.; Qureshi, A.S. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artif Intell Rev* **2020**, *53*, 5455–5516. [[CrossRef](#)]
74. Hanley, J.A.; McNeil, B.J. The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology* **1982**, *143*, 29–36. [[CrossRef](#)]
75. Jung, J.H.; Jeon, B.C.; Youn, B.D.; Kim, M.; Kim, D.; Kim, Y. Omnidirectional Regeneration (ODR) of Proximity Sensor Signals for Robust Diagnosis of Journal Bearing Systems. *Mech. Syst. Signal Process.* **2017**, *90*, 189–207. [[CrossRef](#)]
76. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size. *arXiv* **2016**, arXiv:1602.07360.
77. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
78. SpectraQuest Inc. Available online: <https://spectraquest.com/> (accessed on 18 February 2021).
79. Marins, M.A.; Ribeiro, F.M.; Netto, S.L.; da Silva, E.A. Improved Similarity-Based Modeling for the Classification of Rotating-Machine Failures. *J. Frankl. Inst.* **2018**, *355*, 1913–1930. [[CrossRef](#)]
80. MAFAULDA: Machinery Fault Database. Available online: [http://www02.smt.ufrj.br/~{}offshore/mfs/page\\_01.html](http://www02.smt.ufrj.br/~{}offshore/mfs/page_01.html) (accessed on 24 November 2021).
81. Gibbons, C.B. Coupling Misalignment Forces. In Proceedings of the Proceedings of the 5th Turbomachinery Symposium, College Station, TX, USA, 12–14 October 1976; Texas A&M University, Gas Turbine Laboratories: College Station, TX, USA, 1976.
82. Jalan, A.K.; Mohanty, A.R. Model Based Fault Diagnosis of a Rotor-Bearing System for Misalignment and Unbalance under Steady-State Condition. *J. Sound Vib.* **2009**, *327*, 604–622. [[CrossRef](#)]
83. Tai, X.; Ma, H.; Liu, F.; Liu, Y.; Wen, B. Stability and Steady-State Response Analysis of a Single Rub-Impact Rotor System. *Arch. App. Mech.* **2015**, *85*, 133–148. [[CrossRef](#)]