

VTT Technical Research Centre of Finland

A Novel Machine Learning-Based Approach for Induction Machine Fault Classifier Development—A Broken Rotor Bar Case Study

Tahkola, Mikko; Szücs, Áron; Halme, Jari; Zeb, Akhtar; Keränen, Janne

Published in:
Energies

DOI:
[10.3390/en15093317](https://doi.org/10.3390/en15093317)

Published: 02/05/2022

Document Version
Publisher's final version

License
CC BY

[Link to publication](#)

Please cite the original version:

Tahkola, M., Szücs, Á., Halme, J., Zeb, A., & Keränen, J. (2022). A Novel Machine Learning-Based Approach for Induction Machine Fault Classifier Development—A Broken Rotor Bar Case Study. *Energies*, 15(9), [3317]. <https://doi.org/10.3390/en15093317>



VTT
<http://www.vtt.fi>
P.O. box 1000FI-02044 VTT
Finland

By using VTT's Research Information Portal you are bound by the following Terms & Conditions.

I have read and I understand the following statement:

This document is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of this document is not permitted, except duplication for research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered for sale.

Article

A Novel Machine Learning-Based Approach for Induction Machine Fault Classifier Development—A Broken Rotor Bar Case Study

Mikko Tahkola ^{1,*}, Áron Szücs ², Jari Halme ¹, Akhtar Zeb ¹ and Janne Keränen ¹

¹ VTT Technical Research Centre of Finland Ltd., P.O. Box 1000, FI-02044 Espoo, Finland; jari.halme@vtt.fi (J.H.); akhtar.zeb@vtt.fi (A.Z.); janne.sami.keranen@vtt.fi (J.K.)

² ABB Motors and Generators, Technology Center, FI-00380 Helsinki, Finland; aron.szucs@fi.abb.com

* Correspondence: mikko.tahkola@vtt.fi

Abstract: Rotor bars are one of the most failure-critical components in induction machines. We present an approach for developing a rotor bar fault identification classifier for induction machines. The developed machine learning-based models are based on simulated electrical current and vibration velocity data and measured vibration acceleration data. We introduce an approach that combines sequential model-based optimization and the nested cross-validation procedure to provide a reliable estimation of the classifiers' generalization performance. These methods have not been combined earlier in this context. Automation of selected parts of the modeling procedure is studied with the measured data. We compare the performance of logistic regression and CatBoost models using the fast Fourier-transformed signals or their extracted statistical features as the input data. We develop a technique to use domain knowledge to extract features from specific frequency ranges of the fast Fourier-transformed signals. While both approaches resulted in similar accuracy with simulated current and measured vibration acceleration data, the feature-based models were faster to develop and run. With measured vibration acceleration data, better accuracy was obtained with the raw fast Fourier-transformed signals. The results demonstrate that an accurate and fast broken rotor bar detection model can be developed with the presented approach.

Keywords: broken rotor bar; condition monitoring; fault classification; feature extraction; induction machine; machine learning; predictive maintenance; supervised learning



Citation: Tahkola, M.; Szücs, Á.; Halme, J.; Zeb, A.; Keränen, J. A Novel Machine Learning-Based Approach for Induction Machine Fault Classifier Development—A Broken Rotor Bar Case Study.

Energies **2022**, *15*, 3317. <https://doi.org/10.3390/en15093317>

Academic Editors: Jose Alfonso Antonino-Daviu, Toomas Vaimann and Anton Rassölnkin

Received: 20 April 2022

Accepted: 22 April 2022

Published: 2 May 2022

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



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Induction machines (IMs) are the most common electrical machine type in industrial applications. In IMs, one fault in a single machine can halt a whole production process and cause more severe financial losses than the value of the machine itself. The focus of this work is on rotor bar failure, which is one of the most common fault types after bearing and stator faults, and one broken bar tends to produce expanding damage in its surroundings [1,2]. The rotor bars may fail due to various stresses, including thermal, magnetic, mechanical, dynamic, residual, and environmental stresses [3].

While machine learning (ML) has been used to develop computationally efficient and accurate models, for example, to simulate the behavior of electrical machines [4], it can also be used to develop accurate fault identification models [5]. Regardless of the application, ML-based modeling involves several steps—data acquisition, data preparation, feature engineering, feature selection, model training, including hyperparameter optimization and model validation. The problem of broken rotor bars (BRBs) in IMs has been investigated in several research papers. However, often model validation is performed with multi-fold cross-validation (CV) [6–10] or by having one separate validation dataset [11], even though the nested CV can provide less biased performance estimation by separating optimization of the model hyperparameters and model evaluation [12]. In addition, many papers present

results obtained with hyperparameters that are chosen manually, randomly, or with grid search, even though more efficient methods such as sequential model-based optimization (SMBO) exist. It has been demonstrated that SMBO outperforms several other methods in hyperparameter optimization [13]. Similarly, various signal processing methods have been applied to detect BRB faults. A common one is the fast Fourier transformation (FFT), which is appropriate for analyzing steady-state signals [14]. A more thorough literature study is presented in Section 2.

In this study, we demonstrate the utilization of sequential model-based optimization (SMBO) and nested cross-validation (CV) procedures in the development of an IM fault identification model. We use SMBO to optimize, e.g., the model hyperparameters, and nested CV to estimate the generalization performance of the model. The software implementation is based on an early version of the ATSC-NEX algorithm presented in [15]. In addition, we evaluate the applicability of the random convolutional kernel transformation (ROCKET) algorithm on FFT data and the resulting fault identification performance and compare the performance to a feature-based approach. To the best of our knowledge, such an approach has not been presented earlier in this context. For the feature-based approach, we develop a technique to use domain knowledge to extract features from specific frequency ranges of the fast Fourier-transformed current and vibration signals.

The proposed approach is demonstrated using both simulation and experimental data to develop models for BRB identification. For this purpose, we compare the use of both vibration and current data to train two types of models, namely logistic regression (LR) and CatBoost (CB) [16] classifiers. The main contributions of this article are:

1. A detailed description of the classifier development workflow ranging from data acquisition to model development using SMBO and nested CV.
2. An evaluation of how the number of samples, the direction of vibration acceleration measurement, and the use of different data processing methods, including ROCKET, affect the model accuracy and development time.
3. An evaluation of the use of pipeline optimization in classifier development with measurement data to partly automate the feature engineering process.

The article is organized as follows. Section 2 briefly presents the related work. Section 3 presents our approach and related methodology, including data acquisition with simulations and actual experiments. In addition, Section 3 presents the classifiers employed in this study and a description of applied data preprocessing and feature extraction methods. Section 4 presents and discusses our numerical results with a focus on a comparison of the accuracy and development time of selected classifiers, as well as the inference time when different input features are used. Finally, conclusions of the study are presented in Section 5.

2. Related Work

Rotor bar failure has traditionally been detected from data obtained with vibration sensors [6,17] and from stator currents [14,18]. The side-band frequency component f_{BRB}^i , which is characteristic of BRB failure in the current spectrum, can be computed from [3]

$$f_{BRB}^i = f_n(1 \pm 2ks), \quad (1)$$

where f_n is the nominal frequency, k is an integer, and s is the slip.

Similarly, if there are BRBs, amplitudes at the rotation frequency, f_r , and its side-band frequencies, f_{BRB}^v , increases in the vibration spectrum and can be computed from [17]

$$f_{BRB}^v = f_r \pm f_p, \quad (2)$$

where f_p is the pole pass frequency. The pole pass frequency can be computed from

$$f_p = (f_s - f_r)p, \quad (3)$$

where f_s is the supply frequency and p is the number of poles. In addition, the amplitudes in the vibration spectrum increase at the side-band frequencies around the higher harmonic frequencies, i.e., $2f_r$, $3f_r$, etc. [3].

The identification of the fault requires a model that distinguishes the condition of the machine based on data. To automate the detection of BRBs, a wide variety of signal processing methods and data-driven models have been proposed in the literature. Table 1 presents some of the ML-based methods and input features that can be used for detecting BRBs. Quabeck et al. [10] examined several ML-based algorithms combined with the motor current signature analysis (MCSA) method and motor square current signature analysis (MSCSA) methods for detecting BRBs in IMs. The subspace k-nearest neighbor (k-NN) algorithm combined with MCSA and MSCSA features and slip information resulted in higher average classification accuracy (97.4%) than that of the fine gaussian support vector machine (SVM) and weighted k-NN algorithms. Cupertino et al. [19] trained supervised and unsupervised neural networks for BRB detection in IMs using fast Fourier-transformed current and voltage data, achieving high accuracy with both. Dias and Pereira [20] evaluated the performance of k-NN, SVM, and MLP classifiers with time-domain features and FFT of air gap flux disturbances as input data. Principal component analysis (PCA) was used to reduce the number of features and over 90% accuracy in CV was obtained using the MLP classifier. Godoy et al. [7] used the normalized maximum current signal values for the k-NN, SVM, MLP, and Fuzzy ARTMAP (FAM) network classifiers and achieved an accuracy of 91.5% with the k-NN algorithm.

Ince [9] applied shallow and 1D convolutional neural networks (CNNs) that utilized raw stator current signals and automatically learned the optimal features, thus there was no need for pre-determined transformation (e.g., FFT, hand-crafted feature extraction, and feature selection). The overall classification accuracy was 97.9%. Ramu et al. [11] applied a Hilbert transform and FFT on three-phase current signals and utilized artificial neural networks (ANNs) for the detection of a BRB fault in an IM drive operating under closed-loop direct torque control. Quiroz et al. [8] extracted thirteen time-domain features from the raw current signals and obtained a maximum accuracy of 98.8% with the random forest (RF) algorithm, which outperformed the decision tree (DT), Naïve Bayes (NB) classifier, LR, ridge regression, and SVM. Skylvik et al. [21] applied the stacked autoencoder (AE) network to extract features from the power spectral density (PSD) of a single-phase current. The algorithm was composed of five layers (i.e., four autoencoders and a softmax layer). The average classification accuracy of the method was 95%, and it performed better than the SVM and k-NN algorithms.

Keskes et al. [22] combined the stationary wavelet packet transform (SWT) and multiclass wavelet SVM (MWSVM) for the BRB diagnosis in IMs. Five different kernel functions were tested and, based on CV, it was found that the Daubechies wavelet kernel function can efficiently detect the faulty condition with 99% accuracy. Nakamura et al. [23] performed the FFT analysis for the healthy and faulty rotors and obtained different clusters by using a self-organizing map (SOM). Their method offered high accuracy in situations where the number of BRBs was more than two. Maitre et al. [24] proposed a hierarchical recognition algorithm based on an ensemble of three different classifiers, i.e., MLP, k-NN, and classification and regression trees (CART). Compared to individual algorithms, the approach was considered robust and gave an accuracy of over 90%.

Camarena-Martinez et al. [25] proposed a methodology based on Shannon entropy and the k-means method for detecting BRBs in IMs. Shannon entropy is used to determine the amount of information associated with the vibration signals. The k-means cluster algorithm is then used to classify the entropy values for automatic BRB diagnosis. In [6], the authors first utilized autocorrelation and the discrete wavelet transform (DWT) to process vibration data and then extracted several statistical features from the processed data. The accuracy of a k-NN model that was trained using these features was 80.5–96.7% depending on the machine condition. In [17], the authors applied the FFT on vibration data and analyzed amplitude changes in it. Ince [9] applied shallow and 1D convolutional

neural networks (CNNs) that utilized raw stator current signals and automatically learned the optimal features, thus there was no need for pre-determined transformation (e.g., FFT, hand-crafted feature extraction, and feature selection). The overall classification accuracy was 97.9%. In addition to shallow CNNs, deep neural networks have been also employed to identify faults in rotating machinery [26], and in general, deep learning has been applied successfully to various time series classification problems [27].

The majority of the found BRB detection literature was using electrical machine data in steady-state, except in a few, e.g., [8,25,28], where start-up transient data was used. Ganesan et al. [28] applied the DWT method to transform IM current signals and extracted several statistical features from the transformed data to be used as training data for a multi-layer perceptron (MLP) ANN. They considered issues in the power quality of the supply in the study.

As shown in Table 1, most of the reviewed studies use experimental data to train their models. We study the use of FE simulation data to detect the faults, as the training data can be produced at a lower cost compared with experimental data. Adapting such models to be used with real measurement data by, e.g., using a transfer learning technique can potentially be conducted with a smaller amount of measurement data than would be required to train a model from scratch [29].

Table 1. Methods for detecting BRBs in IMs.

Input Data	Signal Processing	Model Type	Ref
Air gap flux disturbances ^M	FFT, SF ^b , PCA	k-NN, SVM, MLP	[20]
Current ^M	DWT ^a	FAM, k-NN, SVM, MLP	[7]
Current ^M	SF ^b	DT, NB, LR, Ridge, SVM, RF	[8]
Current ^M	-	CNN	[9]
Current ^M	MCSA, MSCSA	k-NN, SVM	[10]
Current, speed ^{S&M}	Hilbert, FFT	ANN	[11]
Current, voltage ^M	FFT	ANN, SOM	[19]
Current ^M	PSD	Stacked AE, SVM, k-NN	[21]
Current ^M	SWT	MWSVM	[22]
Current, voltage, speed ^M	FFT	SOM	[23]
Current ^S	Best-fit 3D ellipse	Ensemble	[24]
Current ^M	DWT ^a , SF ^b	ANN	[28]
Vibration ^M	AC, DWT ^a , SF ^b	k-NN	[6]
Vibration ^M	FFT, feature trend	-	[17]
Vibration ^M	Shannon entropy	K-means clustering	[25]

^a Discrete wavelet transform. ^b Statistical features. ^S Simulated data. ^M Measured data.

Based on the literature study, different physical quantities and various signal processing methods and model types are applicable to the BRB problem. However, these reviewed studies use multi-fold CV or a fixed validation dataset to evaluate the model performance. In addition, advanced hyperparameter optimization methods have not been used, even though they can provide better results. As mentioned in Section 1, we use nested CV to obtain a less biased estimation of model performance and an SMBO method to efficiently find optimal hyperparameters for the models. In this study, both the simulated and measured data represent a machine in steady-state operation, as it allows the use of the FFT method, which is the most common signal processing method and straightforward to apply. Moreover, the LR model type was chosen to be used together with the ROCKET method, as this combination is effectively similar to a single-layer CNN [30] but without the more complex learning stage of CNN. The CB model was chosen as a more advanced method to evaluate its performance in the BRB detection and to compare its performance with the LR model. To the best of our knowledge, the CB model has not been applied in this context before.

3. Approach and Methodology

In this section, the methodology used to develop ML-based classifiers for BRB detection, covering data preparation and feature engineering, classifiers, and the classifier development, including nested CV and sequential model-based hyperparameter optimization, are discussed.

3.1. Overview

The presented model development approach is based on an early version of the ATSC-NEX algorithm proposed in [15]. An overview of the approach is shown in Figure 1. First, the data is preprocessed and divided into a development and holdout test dataset. The development dataset is used first within the nested CV procedure to estimate the generalization performance of classifiers that are developed for detecting BRB failure in squirrel cage IMs. An SMBO procedure is utilized to optimize classifier hyperparameters and the use of pipeline optimization, instead of only optimizing the classifier hyperparameters, is also evaluated. Multi-fold CV procedure is executed after completing the nested CV procedure to obtain hyperparameters for the detection model. The SMBO procedure is used within both nested CV and multi-fold CV to optimize hyperparameters. After multi-fold CV, the whole development dataset is used to train the detection model with the optimized hyperparameters. This is followed by testing of the detection model with the holdout test dataset. The output of the model development workflow is the final detection model and its performance estimate.

This modeling approach is evaluated with two case studies, in which LR and CB classifiers are developed with different types of input data. The first case study is based on finite element (FE) simulation data and the second case study on measurement data. Simulated electrical current and vibration data are first used to evaluate how the number of training samples affects the accuracy of the classifiers. Next, different datasets that were formed using measured experimental vibration data are used to evaluate the effect of using different sets of input features on the classifiers' accuracy.

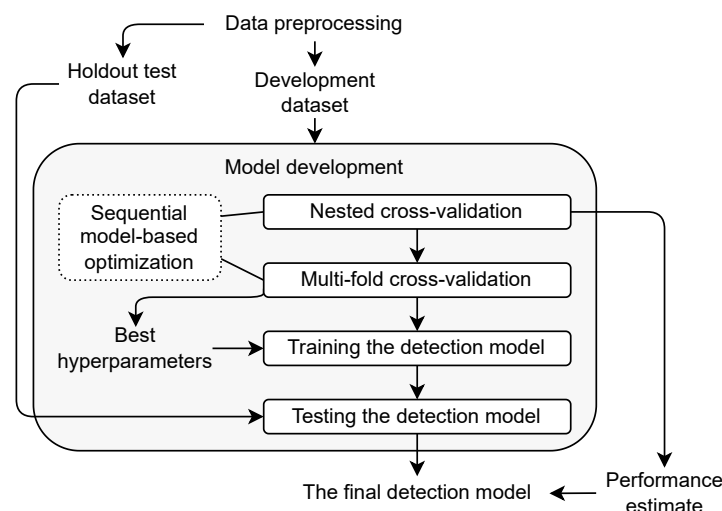


Figure 1. An overview of the model development workflow.

3.2. The Simulation and Experimental Set-Ups

3.2.1. The Simulation Set-Up and Data Generation

The simulation-based results have been computed as a set of electromagnetic 2D FE analyses of a three-phase four-pole squirrel cage IM (shown in Figure 2) using in-house simulation software. The BRBs have been simulated by modifying the rotor cage circuit so that there is an open circuit for broken bars. Forty points, ranging from 0% to 100% load in equal steps, have been computed with a healthy rotor bar and with both one and two BRBs,

resulting in 120 samples for training the classifier. The Figure 2 shows the simulation results in an IM cross-section in the form of the magnetic flux lines and electrical current densities of one case for both a healthy rotor bar and two BRBs. The inner part of the motor (the rotor) rotates around the shaft, including the rotor bars with non-zero current. The outer part of the motor (the stator) is fixed; the rectangular windings are driven by a three-phase current. Even when the net current is zero in broken bars, there still exist positive and negative current densities in the bar, cancelling out each other. Using the magnetic force excitation from the electromagnetic solution, the structural vibrations are computed using unit-wave response-based models [31].

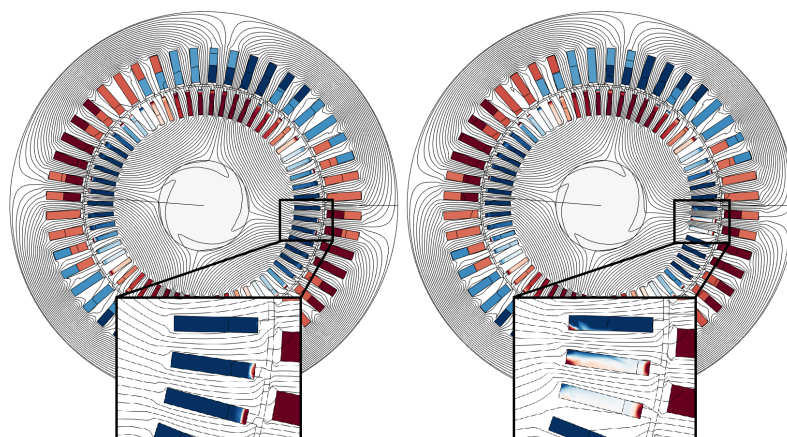


Figure 2. A visualization of magnetic flux lines and electrical current densities in the 2D FE simulation in a cross-section of the IM. On the **left**: a healthy case; on the **right**: two BRBs.

To achieve a higher frequency resolution for the current spectrum-based analyses, the time stepping calculations have been run for more periods than in the vibration-based analyses. The simulations used to generate the current data included 400 periods with 8000 timesteps in total. Figure 3 shows 20 ms of the simulated phase A current in cases with different loads. The vibration level simulation includes five periods with 1000 timesteps in total. The simulation software outputs the vibrations directly in the frequency domain, which is, in this case, the total velocity of vibration at frequencies from 25 to 5000 Hz in 25 Hz steps, as shown in Figure 4. We assume that the rotor bar fault can be detected based on the increased amplitudes at harmonic frequencies in the vibration spectrum. The preprocessing of the current data is presented in Section 3.3.1.

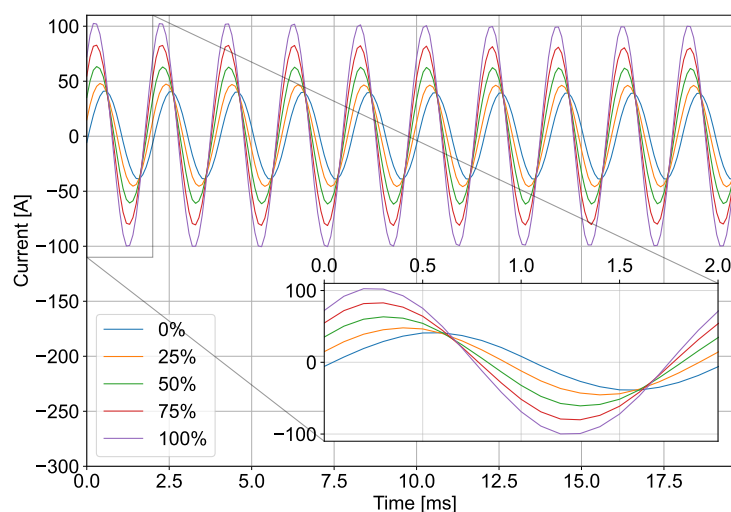


Figure 3. A 20 ms of phase A current in five simulated cases, where the load is approximately 0%, 25%, 50%, 75%, and 100%.

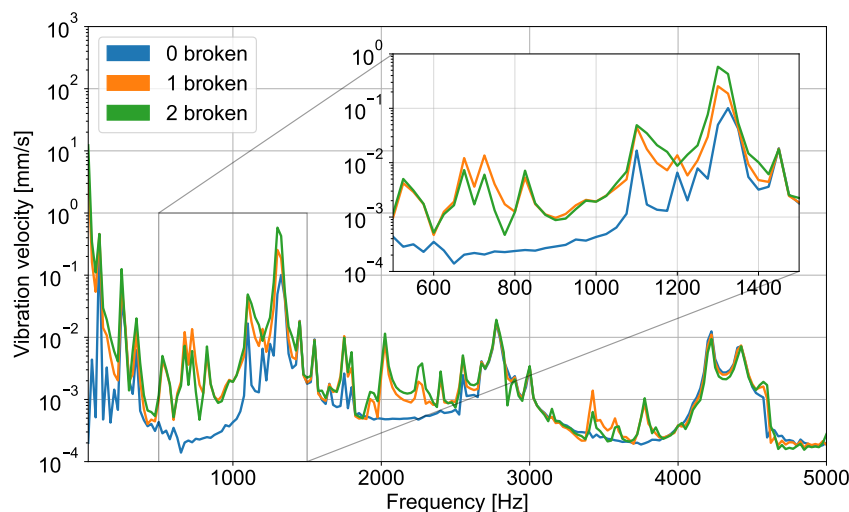


Figure 4. A vibration spectrum of a machine with BRBs and healthy rotor bars.

3.2.2. The Experimental Set-Up and Measurements

The vibration measurements used in this study were carried out at a test bench at the Lappeenranta–Lahti University of Technology (LUT), as part of a wider test arrangement in a joint project between ABB and LUT. The bench consisted of two electrical machines. The test machine was running as a motor and the second machine as a generator, as shown in Figure 5. The actual rotor bar case was tested on the motor side. The motor was an ABB 3-phase 11 kW IM and the generator was an ABB 18 kW IM. The rotation speed of the motor was controlled with an ABB ACS880 frequency converter. In total, six PCB (ICP type model 622B01) vibration acceleration sensors were mounted on the drive-end (DE) and non-drive-end (NDE) shields of the IM in vertical, horizontal, and axial directions. The sensor measurement range is ± 50 g and the frequency range is 0.2–15,000 Hz (± 3 dB). The sensor signals were connected to an ABB AC500-CMS programmable logic controller. The sampling frequency during the analog-to-digital conversion was 50 kHz. The duration for each measured set was 10 s.

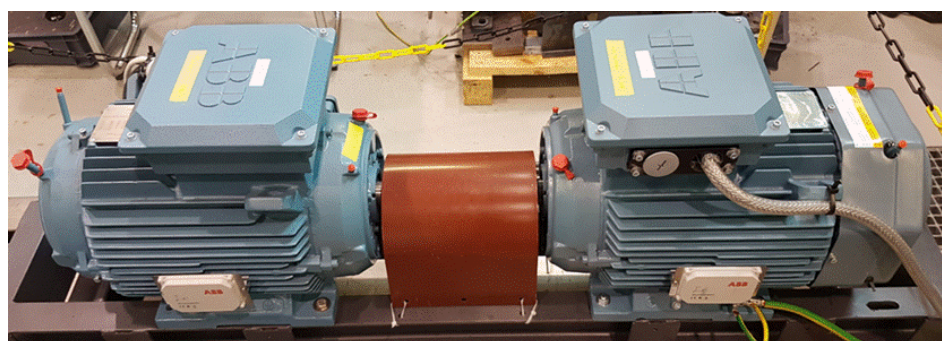


Figure 5. The motor (left) and the generator (right) at a test bench set-up at LUT.

The rotor bar testing was carried out with a healthy and a faulty rotor bar over a predefined test program covering rotation speeds of 900 RPM, 1200 RPM, and 1500 RPM. The loading was from 0% to 100% with a 5% interval in each of the used speeds. In the faulty case, a rotor bar with an artificially made fault was used instead of a healthy one. The artificial fault was made by drilling a hole in the middle of the rotor, as shown in Figure 6. The drilling method has been used by many (e.g., [19,22]) to emulate rotor bar failure.

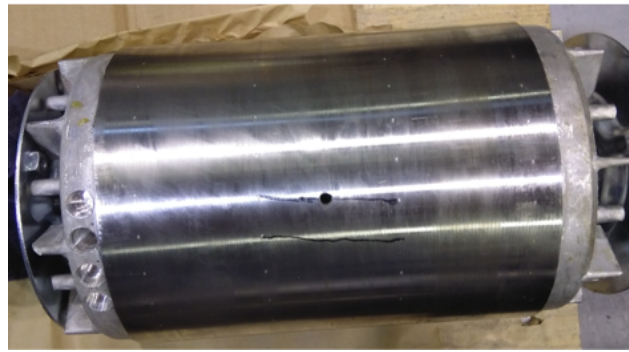


Figure 6. An artificially made fault in the rotor bar on the loading motor.

3.3. Data Preparation and Feature Engineering

The BRB detection models presented in this study are based on either simulation or experimental measurement data. The simulation dataset includes three-phase current signals and the FFT of vibration simulation. The measurement dataset includes vibration acceleration signals from six sensors attached to an IM, as described in Section 3.2.2. The sensors measure acceleration in vertical, horizontal, and axial directions. The data preparation and feature engineering methods are presented in this section and an overview of them is shown in Figure 7.

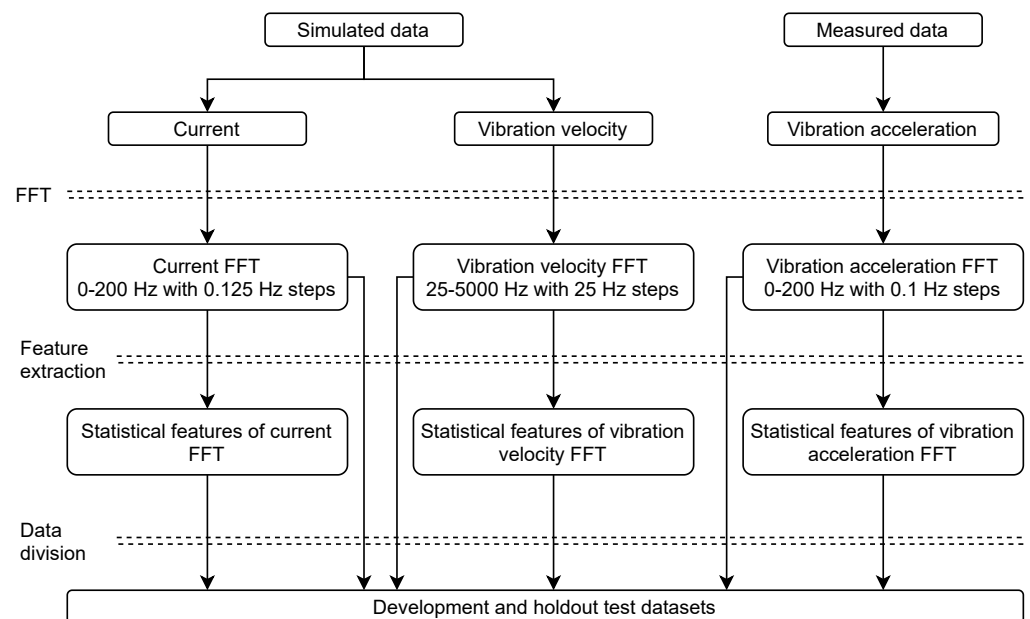


Figure 7. Overview of the data preparation workflow.

3.3.1. Simulation Data

Both the FFT of simulated vibration and three-phase current datasets include 40 load levels, as mentioned in Section 3.2.1. In this study, the FFT of the simulated current and vibration are used directly as inputs for the classifiers. The three-phase current signals are transformed to the frequency domain using an FFT algorithm, and the resulting frequency spectrum is limited to a range of 0–200 Hz, as the BRB failure typically shows as an increased current amplitude at the first and the higher harmonic frequencies and their side-bands, as discussed in Section 2. The resolution of the current frequency spectrum is 0.125 Hz. Figure 8 shows the resulting FFT in one operation point with healthy rotor bars and BRBs. The frequency range for simulated vibration is 25–5000 Hz with steps of 25 Hz.

Next, 12 out of the 40 load levels were excluded from the FFT datasets to be used later as a holdout test dataset to test classifiers. The division was conducted for both the current

and vibration dataset. These test load levels included the lowest and the maximum load levels to measure the extrapolation capability of the classifiers, and the rest of the load levels are there to test the interpolation capability. Three datasets were created using the rest of the data with different numbers of samples for classifier development to study how much the number of samples affects the classification performance. These development datasets included the FFTs of 12, 20, and 28 load levels, corresponding to 30%, 50%, and 70% of all load levels.

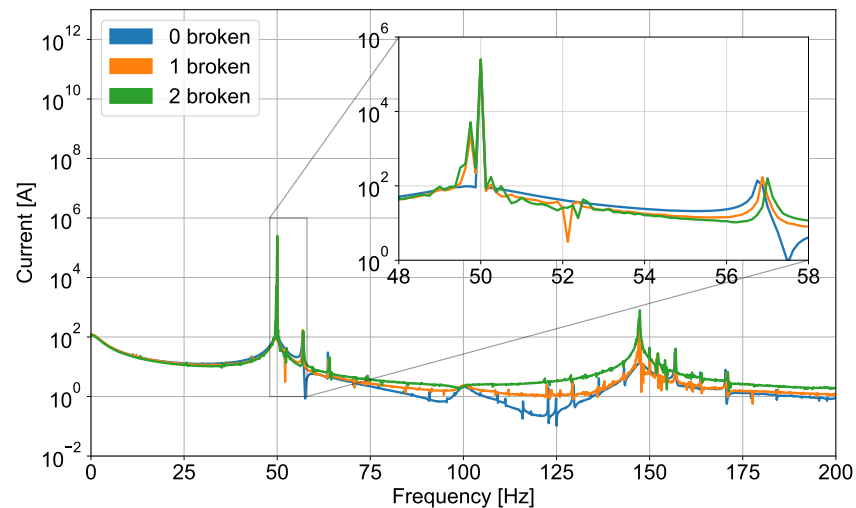


Figure 8. FFT of a simulated current signal (one phase). The frequency range is limited to 0–200 Hz.

In addition to using the raw FFTs as input for the classifiers, another dataset was formed for both a vibration and current-based analysis by computing five statistical features from the corresponding FFT sequences, used as input for the classifier to compare the performance between the two input types. These features were the mean, root mean square, standard deviation, variance, and kurtosis of the vibration velocity spectrum. With the current data, the features were computed from the FFT of each phase current.

3.3.2. Experimental Data

Like the simulated datasets, the vibration measurement data was first transformed from the time domain to the frequency domain using the FFT algorithm. The FFT dataset contained frequencies from 0–25,000 Hz in steps of 0.1 Hz, i.e., 250,001 samples per signal in total. Similar to the simulation-based current spectrums, the measured vibration spectrums were limited to the range 0–200 Hz, as shown in Figure 9. This input type is referred to as $\text{FFT}_{0-200 \text{ Hz}}$, and it contains the FFTs of each measurement and the frequency-wise sum of these FFTs. Therefore, the number of values per signal was reduced to 2001.

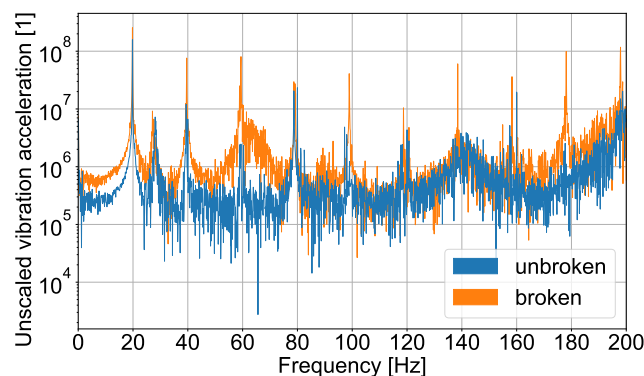


Figure 9. FFT of vibration measurement signals with a BRB and an unbroken rotor bar when the machine was operating at 1200 RPM and under 50% loading. The frequency range is limited to 0–200 Hz.

Similar to the simulated data, the FFT of the vibration acceleration data (0–200 Hz) was first used as input for the classifier without further feature engineering. For the three following experiments, the mean, root mean square, standard deviation, variance, and kurtosis of the vibration velocity spectrum were extracted from the vibration acceleration FFT data for training the classifiers.

In this case, the features were computed in three different ways: (1) they were computed for the whole 0–200 Hz range, (2) they were only computed for ± 6 Hz range at the first harmonic frequency f_1 , or (3) they were computed for the same range around the first three harmonic frequencies: f_1 , f_2 , and f_3 . The last option is shown in Figure 10. These frequency ranges were selected based on the analytical equations shown in Section 2. A similar approach as (2) was used in [32] to take the effect of varying speed on the side-band frequencies into account.

These input datasets are referred to as $\text{FFT}_{f_{200\text{ Hz}}}$, FFT_{f_1} , and $\text{FFT}_{f_{1-3}}$, respectively. The first harmonic (i.e., the fundamental frequency), is estimated based on the no-load RPM of the IM. Although the load affects the rotation speed, its significance in computing the center point (the frequency) of the ± 6 Hz frequency window is negligible. The second and third harmonic frequencies are computed as multiples of the first harmonic. After computing the features from the narrow frequency ranges around the harmonics, the number of input features was reduced from 2001 to 7 or 17 in the two described feature-based datasets, respectively. The input features sets included the no-load speed and load of the machine. These frequency ranges and features were also computed for the frequency-wise sum of the FFTs of signals from the six sensors. Therefore, seven datasets are created for each input type, i.e., 21 datasets in total were used in classifier development. Finally, the datasets were divided into development and holdout test datasets. Cases with load torque levels of 0%, 20%, 40%, 60%, 80%, and 100% were excluded to be used as the holdout test dataset.

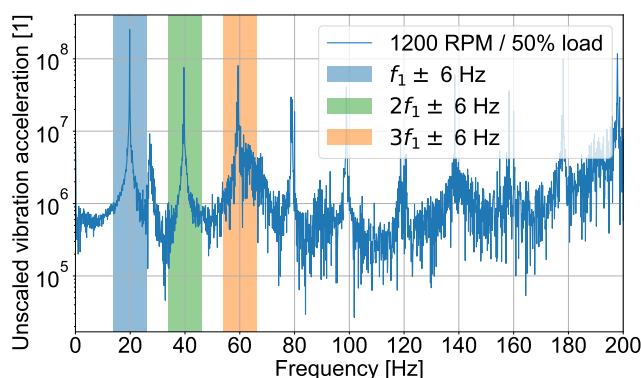


Figure 10. FFT of the measured vibration signal, showing the ± 6 Hz frequency range around the three first harmonic frequencies from which the statistical features are computed.

3.4. Classifiers for Fault Detection

Two classifiers, an LR classifier and a CB gradient boosting classifier, were applied to the BRB modeling problem. The capability of detecting BRBs from simulated and measured current and/or vibration data is evaluated in this study.

An LR classifier is computationally efficient due to its simplicity. It predicts class probabilities Pr_k , as described by

$$Pr_k = \frac{e^{\beta_k^T x}}{1 + \sum_{l=1}^{K-1} e^{\beta_l^T x}}, \quad (4)$$

where k is the class index, K is the number of classes, x is the independent variable value vector, and β^T is the transposed weight vector that is learned during model fitting [33]. In this study, the LR classifier implemented in the Scikit-learn Python library [34] was used.

CatBoost is an open-source ML library for creating gradient boosting ensemble models that are based on using oblivious DTs as base estimators [16]. In oblivious DTs, the decision nodes at the same level evaluate the same splitting criterion, making the tree balanced and less susceptible to overfitting than a regular DT [16,35]. A CB classifier training procedure can be defined to monitor the loss value on an evaluation dataset, which is distinct from the training data, and to output a model with parameters that result in the lowest loss on the evaluation dataset.

3.5. Classifier Development

The LR classifiers were trained on a computer with an Intel Xeon E5-2690 v4 central processing unit. The CatBoost library supports the use of graphics processing units (GPUs) in the training, and in this study, the CB classifiers were trained using an RTX 2080 Ti GPU.

An overview of the model development workflow was shown in Figure 1. The classifiers were developed using the nested CV procedure to estimate the generalization performance, i.e., the performance on data that were not used in the classifier development. The hyperparameters of the classifiers were optimized using the Hyperopt Python library [13]. Hyperopt performs a sequential model-based optimization that is suitable for finding well-performing hyperparameters for the classifiers. Fixed seed values that affect how the data points are split into folds for nested CV and Hyperopt's generation of N random hyperparameter combinations were used to reduce the effect of randomness involved in the model development procedure. In this study, Hyperopt evaluated 20 random hyperparameter combinations at first to build the initial model for optimizing hyperparameters. Then, the algorithm attempted to find well-performing hyperparameters within 20 more evaluations, i.e., the total number of evaluated hyperparameter combinations was 40. The model development procedure was repeated five times with each input data type, and the average of the balanced accuracy (BAC), its standard deviation in nested CV, and BAC on the holdout test dataset are reported.

The hyperparameters and their allowed values for optimization are shown in Table 2. The hyperparameter optimization algorithm was given an option to transform the input data using ROCKET algorithm [30] in the experiments where FFT data was used as input. ROCKET generates a large number of random convolutional kernels that are used to transform sequential data and create features for training. The number of kernels the ROCKET algorithm used was fixed to 2000 in this work. Convolutional kernels are also employed in CNNs but since they involve learning, it can be time-consuming, whereas the ROCKET method aims to take advantage of saved computation time by using random generation.

Table 2. The hyperparameter search space in experiments where FFT data was used as input.

Classifier	Hyperparameter	Values
LR	Maximum iterations	2500
	Solver	lbfgs, liblinear, newton-cg
	C	1×10^{-2} –50
CB	Iterations	5, 10, . . . , 50
	Depth	1–10
	Border count	32–355
	Learning rate	1×10^{-2} –1
	L2 leaf reg	2–10

In the experiments where features extracted from the FFT of measured vibration data were used, feature engineering pipeline optimization was conducted in addition to optimizing the hyperparameters of the classifiers. In practice, the algorithm tries different methods to transform the input data to see which method leads to the best results. The Scikit-learn library was used in constructing the pipeline. Components included in the pipeline optimization are shown in Table 3. Pipeline optimization involves the computation

of polynomial features, scaling or normalizing, feature selection, kernel approximation, and resampling.

Table 3. The pipeline search space in experiments where feature-based input was used. The table follows the order of the components in a pipeline.

Component	Values
PolynomialFeatures	No, Yes
Scaler	StandardScaler, MinMaxScaler, Normalizer, MaxAbsScaler
FeatureSelector	None, RFE ^a
KernelApproximation	None, Nystroem
Resampling	None, SMOTE ^b , Naive oversampling, Naive undersampling

^a Recursive feature elimination. ^b Synthetic minority oversampling technique.

The nested CV procedure is used to estimate the generalization performance, as its result is less biased than that of the flat multi-fold cross-validation [12]. The nested CV includes an outer and inner loop, as shown in Figure 11. The classifier development dataset is first divided into K outer folds in a stratified manner, i.e., in such a way that in each fold there is approximately the same number of examples of each class. Then, hyperparameter optimization is repeated K times, each time using $K - 1$ outer folds for hyperparameter optimization within the inner loop and one fold for evaluating the performance of a model with optimized hyperparameters. In the inner loop, the $K - 1$ folds are further divided into J inner folds in a stratified manner. A model with fixed hyperparameters is then trained J times, each time using $J - 1$ inner folds for training and one fold for validation. For each hyperparameters combination in the inner loop, an average of the J validation losses is computed. Then, the hyperparameters with the lowest average inner validation loss are used to train a model with the outer $K - 1$ training folds, which is followed by the evaluation of the validation loss on the current outer validation fold. In the end, this results in K outer validation loss values, i.e., performance estimates, as shown in Figure 11. The average and the standard deviation of these K outer validation loss values form the estimate for the generalization performance.

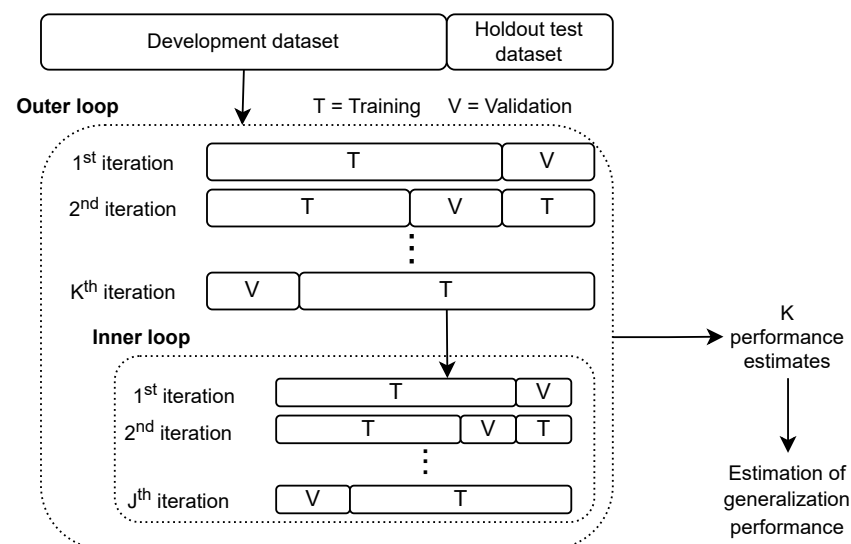


Figure 11. The nested CV procedure, which is used to estimate the generalization performance of the BRB detection model.

In this work, a logistic loss function is used to evaluate the predictive performance of the classifiers within hyperparameter optimization in the inner loop of the CV. The classifier's generalization performance is estimated using BAC (i.e., the average of the recall values obtained for each individual class) as the metric. In this study, the number of

folds in both the inner and outer loop of nested CV was six. After having an estimation of the generalization performance, a six-fold CV is run to find hyperparameters for the final classifier using the whole development dataset. The best hyperparameters are then used to set up the final classifier, which is trained using the whole development dataset.

With the measured vibration acceleration data, the best signal source (i.e., the sensor and the direction of measurement) for each input dataset and both classifiers is selected based on computing the weighted BAC, i.e., BAC_w , using

$$BAC_w = (BAC_{nCV} - 0.01e^{\sigma/0.075} + BAC_{test})/2, \quad (5)$$

where BAC_{nCV} is the BAC obtained in nested CV, σ is the standard deviation of the BACs obtained on the outer test folds in nested CV, and BAC_{test} is the BAC on the holdout test dataset. The coefficients 0.01 and 0.075 in the equation have no physical meaning and were chosen so that a slightly higher penalty is given for higher standard deviations than for lower ones.

4. Results and Discussion

The results of the FE simulation and measurement data-based model development are presented in this section. LR and CB classifiers were developed in each experiment to compare the performance of the two. The experiments shown here were repeated five times and the average values are reported. The simulated current and vibration velocity data-based modeling was conducted using three different numbers of samples in nested CV to compare how much the number of samples affects model performance. However, the main focus is on evaluating how different input features affect the model performance. The reported model development times are real-time, and it should be noted that the fitting of the LR models utilizes a central processing unit (CPU), whereas the CB training makes use of a GPU, as mentioned in Section 3.5.

4.1. Simulated Current Data

The classifier development using simulated current data was conducted separately for the FFTs of the three-phase currents, here referred to as I_A , I_B , and I_C . In addition, statistical features computed from the FFTs of the three-phase currents (dataset I_{feat_200Hz}) were used to develop classifiers.

The results of the current-based classifiers are shown in Table 4. With the raw FFTs of individual phase currents as the input, the CB classifier achieved 99.2–100.0% BAC with a standard deviation of 0–1.9% in nested CV with only 30% of the samples used in the training. These CB classifiers had a BAC of 99.6–100.0% on the holdout test dataset, already showing excellent generalization performance on unseen data with a small number of training samples. The corresponding LR classifier, on the other hand, had a BAC of 72.9–86.7% with a standard deviation of 16.3–20.3% in nested CV when 30% of the samples were used. Still, these LR classifiers had 92.9–100.0% BAC on the holdout test dataset. However, the nested CV score of LR with raw FFT input increased when the number of samples was increased to 50% and did so even more with 70% of the samples where the BAC was 98.3–99.7% with a standard deviation decreased to 0.7–3.3%. The results with FFT-based data demonstrate that the standard deviation of BAC_{nCV} decreases with the LR model when more samples are used to develop the model. With the CB model, the standard deviation is relatively low already with the lowest number of samples. The results with feature-based data demonstrate, on the other hand, that the standard deviation decreases with both model types when more samples are used to develop the model. This suggests that there was not enough data used in the development of the models that had high variance.

There are several reasons why the nested CV score can be lower than the corresponding score on the holdout test dataset. The nested CV score is based on evaluating each sample in the development dataset, i.e., the majority of the whole dataset, whereas the holdout test dataset is a minor part of the whole dataset. Thus, the nested CV provides a better estimation of how the model works on data that has not been used in the model

development. On the other hand, there are fewer samples available for training a model within the outer loop of the nested CV procedure than there are available for training the final model, which can affect the prediction accuracy.

Table 4. The BAC of classifiers trained with FFTs of three phases of simulated current (I_A , I_B , and I_C) or features computed from the FFT data ($I_{\text{feat_200Hz}}$). The experiment was repeated with 30%, 50%, and 70% of samples in the training dataset.

Samples	Input	LR			CB		
		BAC _{nCV}	BAC _{test}	t _{dev}	BAC _{nCV}	BAC _{test}	t _{dev}
30%	I_A	72.9% ± 20.3%	100.0%	12.8 min	100.0% ± 0.0%	100.0%	49.7 min
	I_B	84.2% ± 19.8%	92.9%	11.9 min	99.2% ± 1.9%	99.6%	50.1 min
	I_C	86.7% ± 16.3%	94.6%	8.0 min	99.2% ± 1.9%	100.0%	49.1 min
	$I_{\text{feat_200Hz}}$	92.8% ± 11.2%	100.0%	0.4 min	88.3% ± 12.0%	93.6%	19.9 min
50%	I_A	97.4% ± 4.0%	100.0%	17.6 min	100.0% ± 0.0%	100.0%	56.7 min
	I_B	98.0% ± 3.6%	97.5%	14.9 min	99.4% ± 1.2%	100.0%	55.4 min
	I_C	93.7% ± 8.8%	98.5%	14.9 min	97.8% ± 5.0%	100.0%	55.0 min
	$I_{\text{feat_200Hz}}$	98.0% ± 3.8%	100.0%	0.4 min	92.0% ± 8.3%	98.3%	20.8 min
70%	I_A	99.7% ± 0.7%	99.2%	20.0 min	99.7% ± 0.7%	100.0%	62.2 min
	I_B	98.3% ± 3.2%	97.5%	17.2 min	98.7% ± 2.8%	100.0%	61.9 min
	I_C	98.5% ± 3.3%	97.5%	17.5 min	100.0% ± 0.0%	100.0%	62.6 min
	$I_{\text{feat_200Hz}}$	100.0% ± 0.0%	100.0%	0.4 min	96.0% ± 4.7%	98.9%	20.6 min

Table 4 also shows that the performance of the CB classifier trained on the FFTs of individual phase currents remained approximately the same when the number of samples used in the development was increased, although the training time measured as real-time increases. However, using the feature-based input $I_{\text{feat_200Hz}}$ to train the CB classifiers requires 70% of the samples to be used in the development to reach almost as high BAC in nested CV (96.0% ± 4.7%) and when using the holdout test dataset (98.9%). Still, one should note that the model development is approximately more than three times faster with the feature-based dataset compared with raw FFT data, as the number of inputs is lower. In addition, the lower number of inputs affects the computation time of the model itself when it is used to make predictions.

In contrast to the CB classifier, the LR classifier performed better when using the feature-based input rather than raw FFT input. The development of the LR classifier took only 0.4 min with feature-based input, regardless of the number of samples, which is approximately 47 times faster compared with the corresponding CB models. Compared to the development of LR and CB models using raw FFT data, the feature-based LR model was respectively 22–46 and 114–144 times faster to train, depending on the number of samples used. From the application point of view, the best choice from these options would be to develop an LR model that takes features computed from FFT data as input as that model is both fast to train and achieves 100% BAC in nested CV and when using the holdout test dataset. This LR model extrapolates well, as the holdout test dataset included lower and higher load points than the development dataset.

4.2. Simulated Vibration Velocity Data

The simulated vibration velocity FFT data was used to form two datasets, namely $v_{v_5000\text{Hz}}$ and $v_{v_feat_5000\text{Hz}}$. The former contains unprocessed FFT data (vibration spectrum), and the latter only contains statistical features computed from the FFT data. Using the simulated vibration spectrum as the input for the classifiers, high BACs are obtained with both classifiers, as shown in Table 5. With raw FFT data, the LR classifier achieved 98.3% BAC with a standard deviation of 3.7% in nested CV and 100% BAC with the holdout test dataset using only 30% of the samples. Improvement was nevertheless obtained when 70% of the samples was used to train the LR model as the standard deviation of BAC in

nested CV decreased to zero while the BAC remained at 100.0% in nested CV and using the holdout test dataset. With the CB classifier trained on raw FFT data, 70% of the samples were required to obtain 97.2% BAC in nested CV, but the standard deviation was still higher than with the LR classifiers. The nested CV BAC of feature-based LR and CB classifiers only increased slightly (from 81.1% to 85.2% and from 81.7% to 84.8%, respectively), when the number of samples was increased from 30% to 70%. Similarly, as with the simulated current data, the standard deviations decrease with simulated vibration data when more samples are used in the model development.

With simulated vibration velocity data, the feature-based LR and CB models were, respectively, 5–20 and 7–10 times faster to develop compared with the pure FFT-based classifiers. The time required to develop the CB classifiers remained approximately the same regardless of the number of samples used in the training. The development of feature-based CB classifiers was approximately eight times faster compared with raw FFT. Based on these results, it can be concluded that the LR model trained with the FFT of vibration velocity data works the best in this case, and the model extrapolates well, as 100% BAC was obtained on the holdout test dataset that included lower and higher load points than the development dataset. Although its development time is higher compared with that of the same model trained on the feature-based input, it is still reasonable.

Table 5. The BAC of classifiers trained using FFT of simulated vibration velocity v_v ($v_{v_5000\text{Hz}}$) or features computed from the FFT data ($v_{v_feat_5000\text{Hz}}$). The experiment was repeated with 30%, 50%, and 70% of samples in the training dataset.

Samples	Input	LR			CB		
		BAC _{nCV}	BAC _{test}	t _{dev}	BAC _{nCV}	BAC _{test}	t _{dev}
30%	$v_{v_5000\text{Hz}}$	98.3% ± 3.7%	100.0%	1.9 min	88.3% ± 17.0%	97.8%	18.9 min
	$v_{v_feat_5000\text{Hz}}$	81.1% ± 15.3%	85.7%	0.3 min	81.7% ± 13.7%	86.0%	1.8 min
50%	$v_{v_5000\text{Hz}}$	97.5% ± 5.6%	100.0%	5.1 min	94.9% ± 6.5%	98.5%	20.0 min
	$v_{v_feat_5000\text{Hz}}$	82.5% ± 8.6%	87.0%	0.4 min	83.4% ± 10.3%	86.3%	2.3 min
70%	$v_{v_5000\text{Hz}}$	100.0% ± 0.0%	100.0%	7.2 min	97.2% ± 5.2%	97.5%	20.1 min
	$v_{v_feat_5000\text{Hz}}$	85.2% ± 6.3%	86.1%	0.4 min	84.8% ± 8.7%	85.0%	2.9 min

These results suggest that the extracted statistical features fail to capture all the relevant information from the raw FFT vibration velocity data, whereas with the simulated current data, the features led to better results. The results in Section 4.1 demonstrate that with the simulated current data, the number of samples has a greater effect on the accuracy of the LR classifier compared with the CB model. With CB, BAC of 100% was already obtained in nested CV and with the holdout test dataset with the lowest amount of samples used, whereas the LR model required the highest amount of samples tested to achieve the same. However, with the latter, it was the feature-based approach that was not only the most accurate but also the fastest to develop and one of the fastest to make predictions. With simulated vibration velocity data, on the other hand, the feature-based approach did not yield as high accuracies as the FFT-based approach. Still, BAC of 100% was obtained in nested CV and with the holdout test dataset with the vibration velocity spectrum as the input for the LR model, although it does this with a higher computational cost compared with the best current-based model. In general, the input feature set had a more dominant effect on the accuracy and the computational efficiency than the number of training samples.

4.3. Measured Vibration Acceleration Data

Four different sets of features ($a_{v_200\text{Hz}}$, $a_{v_feat_200\text{Hz}}$, $a_{v_feat_f1}$, and $a_{v_feat_f123}$) were separately formed from the signals of six accelerometers and used to develop LR and CB classifiers to identify a BRB in IM. The data acquisition of the measurement data was discussed in Section 3.2.2. Vibration acceleration sensors were mounted on the drive-end and non-drive-end shields of the IM in vertical, horizontal, and axial directions. In this

section, these sensors are referred to as DE_{hor} , DE_{vert} , DE_{ax} , NDE_{hor} , NDE_{vert} , and NDE_{ax} . Classifiers were also trained using the frequency-wise sum of the fast Fourier-transformed vibration signals of the six sensors and with the statistical frequency domain features computed from the FFT data, as discussed in Section 3.3.2. The model development procedure was repeated five times, as discussed in Section 3.5, and the results shown in Table 6 are the average values obtained from these five repetitions. The best signal source for each dataset was selected based on BAC_w as described in Section 3.5. The best signal sources are shown in bold in Table 6.

Table 6. Comparison of BAC in nested CV BAC_{nCV} , holdout test dataset BAC_{test} , and weighted BAC_w , as well as the computation time required to develop the classifiers when training with different measurement-based datasets. The models were trained using the FFTs of the six vibration acceleration a_v signals and statistical features of these FFTs. For each input type, the results of the sensor data which resulted in the highest BAC_w are shown in bold.

Input	Sensor	LR				CB			
		BAC_{nCV}	BAC_{test}	BAC_w	t_{dev}	BAC_{nCV}	BAC_{test}	BAC_w	t_{dev}
a_v_{200Hz} ^a	DE_{hor}	86.9% ± 11.0%	97.6%	90.1%	9.7 min	86.1% ± 11.2%	90.7%	86.2%	67.8 min
	DE_{vert}	91.3% ± 10.2%	83.3%	85.3%		90.0% ± 11.2%	77.8%	81.7%	
	DE_{ax}	90.0% ± 9.5%	88.7%	87.6%		84.2% ± 10.7%	86.5%	83.2%	
	NDE_{hor}	87.0% ± 11.0%	90.8%	86.7%		83.7% ± 11.5%	87.1%	83.1%	
	NDE_{vert}	85.8% ± 11.4%	89.0%	85.1%		86.7% ± 11.1%	84.5%	83.4%	
	NDE_{ax}	88.1% ± 11.0%	90.4%	87.1%		84.8% ± 11.6%	87.7%	83.9%	
	Sum	88.4% ± 8.8%	88.0%	86.6%		88.6% ± 10.3%	84.6%	84.6%	
$a_v_{feat_200Hz}$ ^b	DE_{hor}	73.0% ± 12.4%	75.7%	71.8%	5.2 min	71.0% ± 15.0%	85.7%	74.7%	3.6 min
	DE_{vert}	84.9% ± 8.9%	81.4%	81.5%		87.6% ± 9.1%	88.1%	86.2%	
	DE_{ax}	81.4% ± 7.4%	77.7%	78.2%		77.7% ± 11.3%	77.3%	75.2%	
	NDE_{hor}	71.1% ± 12.5%	72.8%	69.3%		69.3% ± 11.9%	84.9%	74.7%	
	NDE_{vert}	87.0% ± 8.4%	82.4%	83.2%		87.8% ± 7.6%	79.2%	82.1%	
	NDE_{ax}	82.4% ± 9.9%	77.2%	78.0%		78.6% ± 15.3%	79.4%	75.1%	
	Sum	78.1% ± 11.3%	79.3%	76.4%		82.8% ± 10.8%	90.9%	84.7%	
$a_v_{feat_fl}$ ^c	DE_{hor}	78.9% ± 9.3%	88.4%	81.9%	5.6 min	79.0% ± 10.8%	92.5%	83.6%	3.3 min
	DE_{vert}	74.8% ± 12.0%	62.1%	66.0%		77.7% ± 14.7%	83.5%	76.7%	
	DE_{ax}	76.6% ± 12.8%	81.9%	76.5%		80.3% ± 13.0%	89.4%	82.1%	
	NDE_{hor}	78.7% ± 12.1%	88.6%	81.2%		79.8% ± 12.9%	91.3%	82.8%	
	NDE_{vert}	78.8% ± 13.4%	87.9%	80.3%		78.0% ± 13.0%	89.7%	81.0%	
	NDE_{ax}	80.1% ± 12.8%	90.5%	82.6%		81.3% ± 11.8%	85.5%	81.0%	
	Sum	78.1% ± 11.6%	87.9%	80.6%		78.5% ± 13.8%	89.5%	80.9%	
$a_v_{feat_fl23}$ ^d	DE_{hor}	87.6% ± 7.8%	86.9%	85.8%	6.7 min	85.5% ± 9.6%	89.3%	85.6%	7.3 min
	DE_{vert}	87.9% ± 9.5%	88.4%	86.4%		89.8% ± 11.4%	87.3%	86.2%	
	DE_{ax}	86.0% ± 9.4%	86.3%	84.4%		83.0% ± 8.0%	89.6%	84.9%	
	NDE_{hor}	87.4% ± 9.6%	84.4%	84.1%		86.5% ± 8.7%	86.2%	84.7%	
	NDE_{vert}	84.1% ± 9.2%	81.7%	81.2%		81.8% ± 11.3%	88.9%	83.1%	
	NDE_{ax}	86.0% ± 8.4%	89.4%	86.2%		83.7% ± 11.2%	91.4%	85.3%	
	Sum	89.1% ± 7.4%	88.2%	87.3%		87.7% ± 8.6%	87.3%	85.9%	

^a FFT of vibration acceleration (0–200 Hz). ^b Features of vibration acceleration FFT (0–200 Hz). ^c Features of vibration acceleration FFT ($f_1 \pm 6$ Hz). ^d Features of vibration acceleration FFT ($f_{1-3} \pm 6$ Hz).

The best BAC_w score, 90.1%, was obtained with the LR classifier trained on FFT data (a_v_{200Hz}), computed from the sensor DE_{hor} signal. However, the BAC_w for the LR classifier trained on the feature-based $a_v_{feat_fl23}$ dataset was almost as high (87.3%), while the computation time required to develop the feature-based classifier was approximately 1.5 times shorter than that of the FFT-based classifier. The slightly longer development time of the FFT-based classifier is not only caused by the higher number of input variables but also due to the different feature engineering options in the hyperparameter optimization, which were discussed in Section 3.5. In particular, having ROCKET transformation as one option to process the data caused slightly longer computation times with the FFT-based datasets.

The highest BAC_w score for the CB classifier was 86.2%, which was obtained with three of the four input types (excluding $a_v_{feat_fl}$). Although the highest BAC_{test} with

CB was obtained with $a_{v_feat_f1}$ and DE_{hor} , the corresponding nested CV BAC was only $79.0\% \pm 10.8\%$. The possible reason for such a result was discussed in Section 4.1.

In this case, the standard deviations of BAC_{nCV} were 7.4–13.4% with the LR model and 7.6–15.3% with the CB model. However, the standard deviations of the models trained with a specific input are relatively close to each other regardless of the sensor, i.e., the measurement direction. This suggests that there might be some samples in the dataset with an information value that is not so good, i.e., they are challenging to learn from and to classify. This could be confirmed by looking at the individual samples one by one and checking whether samples of some specific operation area are systematically misclassified. In such case, obtaining more data for development could help, as the results in Sections 4.1 and 4.2 demonstrate.

The development time of the CB classifiers was in the range of 3.3–7.3 min with the feature-based approach and 67.8 min with the FFT data. The CB model was faster to train than the LR model with the feature-based datasets $a_{v_feat_200Hz}$ and $a_{v_feat_f1}$, but a bit slower with the $a_{v_feat_f123}$ dataset. However, with the FFT-based dataset a_{v_200Hz} , the LR model was almost seven times faster to develop than the CB model, suggesting that with these datasets, the LR model scales better to a higher number of input features than the CB model. One must keep in mind that the number of training samples is constant in each of the experiments shown in this section.

When using the raw FFT data as input for either classifier, the optimization algorithm found that applying the ROCKET transformation on the FFT data results in a smaller logistic loss. Analyzing the hyperparameters of the LR model, the inverse of regularization strength C obtained higher values with the feature-based dataset compared with the raw FFT. This is logical, as raw FFT data contain many more variables than the feature datasets, and thus stronger regularization is needed to prevent overfitting the model. Overfitting is especially a problem when the number of features is higher than the number of samples. With L2 regularization applied, the values of the coefficients of irrelevant features achieve values closer to zero than without regularization, which means that the regularized model does not respond so strongly to changes in these features.

The average computation times required to develop the classifiers and the corresponding BAC_w with different input features are visualized in Figure 12. It summarizes the discussed findings and demonstrates that while the feature-based datasets mean a short development time with both classifiers, the maximum weighted BAC with them is lower than 88%. However, Figure 12 also shows that the LR model scales better in terms of development time and can detect the bar failures more accurately than the CB classifier.

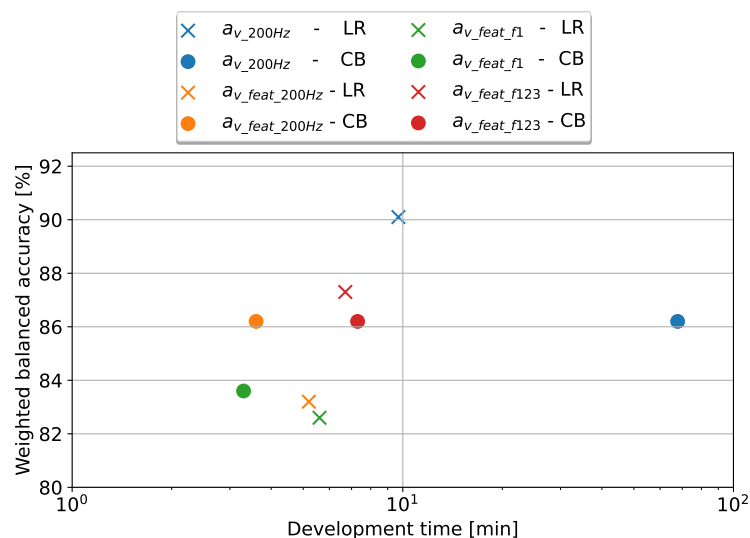


Figure 12. The computation time required to develop a classifier vs. the weighted BAC of classifiers trained with different input datasets.

The computation times required to make predictions (i.e., the model run time), with FFT and feature-based classifiers and the corresponding BAC_w are visualized in Figure 13. It shows that the FFT-based classifiers are slower to use for predicting the bar failures than the corresponding feature-based models. To analyze the reasons behind this, Table 7 shows a breakdown of the total computation time required for predicting with these classifiers, including the computation time that the data processing requires as well as the time required to run the actual model to obtain a prediction.

With raw FFT data, the data processing step takes approximately the same amount of time with both classifiers. However, with the LR model, the actual prediction can be obtained in a significantly shorter time than with the CB model, as it is 481 times faster. Both classifiers trained with raw FFT data make use of the ROCKET transformation, which makes their data processing time longer compared to the feature-based approach. This suggests that the LR model scales better, not only in terms of development time when the number of features increases, but also in terms of the computation time required to make predictions. The feature-based LR model has a more than four times faster data processing pipeline and computes the actual prediction almost ten times faster than the corresponding CB model. In total, the feature-based LR model is over five times faster in computing a prediction than CB, but their accuracy is similar.

Even though the raw FFT-based LR classifier achieved the highest accuracy in this study, Figures 12 and 13 show the importance of feature engineering. The feature-based classifiers are not only significantly faster to train but also to use in operation, and thus it may be beneficial to study the more extensive extraction of statistical features. While the most accurate model (i.e., the FFT-based LR model) can make approximately 17 predictions each second, the feature-based LR model reaches a speed of over 900 predictions per second. Each of the developed models is computationally fast enough to be used for real-time fault monitoring during operation. Naturally, depending on the hardware used (e.g., in edge computation), the computation time of the slowest models might limit the frequency of analyzing the bar condition, which should be considered when selecting the methods.

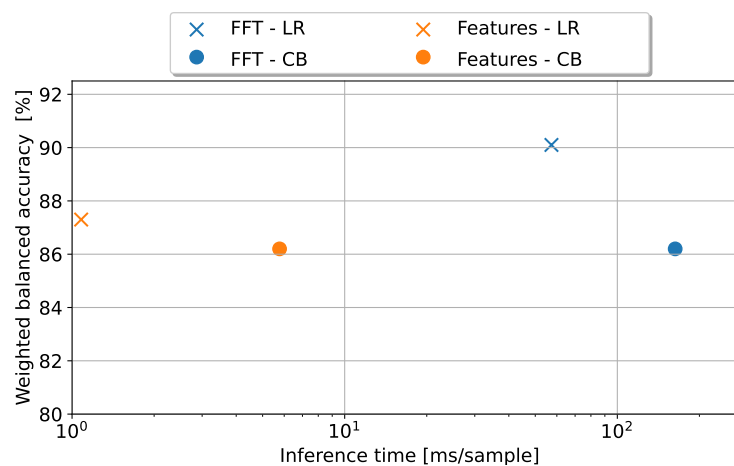


Figure 13. The computation time required to make a prediction vs. the weighted balanced accuracy of classifiers trained with different input datasets.

Table 7. The average computation time (including the time required to process input data and run a model to obtain a result) of selected classifiers trained with FFT or feature-based vibration acceleration measurement data. The results are computed using samples in the test dataset.

Classifier	Input	Computation Time (ms/Sample)		
		Data Processing	Prediction	Total
LR	FFT	57.12	0.21	57.33
	Features	0.83	0.25	1.08
CB	FFT	61.95	101.05	163.00
	Features	3.40	2.37	5.77

Figures 14 and 15 show classifications computed on the holdout test dataset with the best feature-based LR and CB models, respectively. In both, the x-axis and y-axis indicate the operation point of the machine (i.e., the rotation speed and load, respectively), while the color of the markers shows whether the classification was correct or not. There were four measurements available in the holdout test dataset for most of the operation points—two with both a BRB and a healthy rotor bar.

Figure 14 shows that the LR model classifies all but two samples correctly. This LR model was trained using features computed from the frequency-wise sum of six FFTs of measured vibration acceleration signals ($a_{v_feat_f123}$ dataset). The first is at the operation point, where the speed is 1500 RPM with zero load, in which case one of the two samples with a healthy rotor bar is classified as broken. At this operation point, the model is extrapolating, as the lowest load included in the model development data was 5%. The challenge in the zero load condition might be caused by the fact that when the load is low, the slip is low too, which in turn means that the side-bands in the vibration spectrum that are characteristic of the rotor bar failure are closer to the harmonic frequencies in comparison with the high slip values [17]. The second wrongly classified sample is at speed is 900 RPM with a 60% load, in which a BRB is classified as *healthy*.

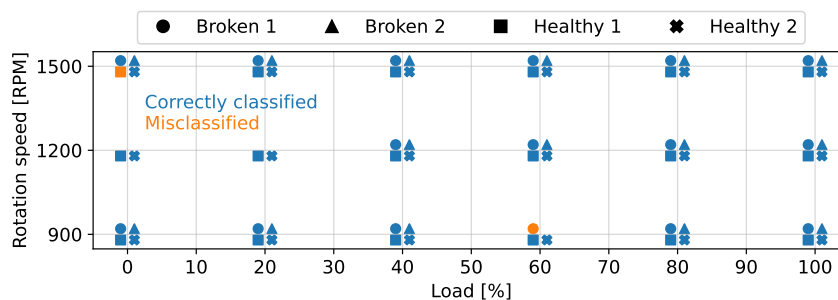


Figure 14. The results of the LR model used with the holdout test dataset shows which samples with different no-load rotation speeds and loads were classified correctly.

The CB classifier, which was trained using features computed from the FFTs of measured vibration acceleration sensor DE_{vert} ($a_{v_feat_f123}$ dataset), failed to correctly classify seven samples out of 67 samples in the holdout test dataset, as shown in Figure 15. As with the LR model, a broken bar was also detected as healthy at a 60% load and with 900 RPM with CB. Two of the misclassified samples represented extrapolating operation points with a load of 100% and a speed of 900 RPM where BRBs were classified as *healthy*. The same misclassification was made for samples with a load of 20% and at 900 RPM, and at the same load level but at 1500 RPM speed, healthy bars were classified as *broken*. Since the raw FFT-based LR model classified these operation points correctly, it might be that the difference between the faulty and healthy case is not so clear in the FTT frequency response, and hence the few selected statistical features fail to capture it, whereas the FFT-based LR model is sensitive enough to recognize the difference. Regardless of the model type, interpreting the classifiers is challenging, as various feature transformations

are applied to the input data (ROCKET applied to FFT data or various methods applied to statistical features).

The results demonstrate that one specific measurement direction is not significantly better than any other regarding how accurately the bar failure can be detected. Interestingly, for each dataset, there is still a visible pattern regarding what is the best and worst measurement direction, as they are the same for both classifiers. For example, with raw FFT data, on average the horizontal measurement direction resulted in slightly higher accuracy than other directions, whereas the vertical direction is a bit worse than other directions. The horizontal direction is also a slightly better option with the $a_{v_feat_f1}$ dataset. With the $a_{v_feat_200Hz}$ dataset, the vertical measurement direction is accuracy-wise better than other directions. The frequency-wise sum of FFTs computed from all signals was found to be best with the $a_{v_feat_f123}$ dataset, with a minor margin over individual signals. However, it requires all six measurements to be available for monitoring. Based on these findings, it seems that the most accurate rotor bar failure detection can be obtained with an LR classifier trained with the raw FFT data of vibration acceleration measured in a horizontal direction, and by transforming the FFT data using the ROCKET algorithm.

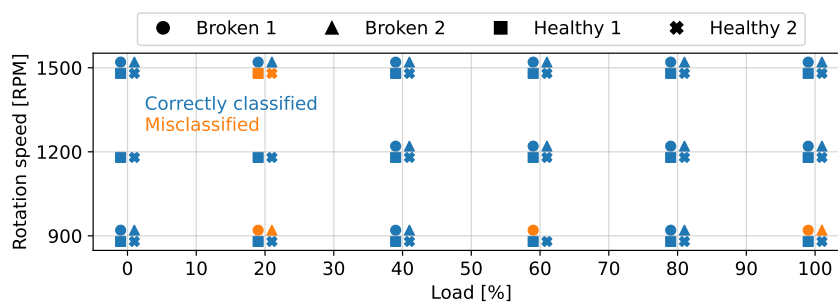


Figure 15. The results of the CB model used with the holdout test dataset, showing which samples with different no-load rotation speeds and loads were classified correctly.

The experiments presented in this section included two additional input feature sets where domain knowledge was utilized to compute the statistical features of FFT, only within a narrow frequency range around the first or the first three harmonic frequencies, and not from the whole FFT sequence. The computation of the features around the first three harmonic frequencies resulted in almost as high accuracy as was achieved with the FFT-based input data but with 96 times shorter development time with the LR model, which demonstrates the potential of the feature-based approach even though only five features were extracted from each of the narrow frequency ranges. Focusing the analysis on the relevant frequency ranges reduces the amount of noise and redundant or irrelevant input features, which might be one reason for lower standard deviations in nested CV scores with the feature-based dataset. This highlights the importance of feature engineering. Still, the highest BAC_w score was obtained by using the data that were transformed using the fast Fourier and the ROCKET methods to train an LR model. In this study, the LR model performed overall slightly better than the CB model when both the accuracy and the computational efficiency are considered.

5. Conclusions

In this article, we have presented a novel approach for broken rotor bar fault identification model development for an induction machine. The presented approach utilizes nested cross-validation to deliver a reliable estimation of the model performance, and sequential model-based optimization to effectively find optimal model hyperparameters. The cost of the more reliable performance estimate is that more computational resources are required compared to, e.g., multifold CV, as many more models are trained. However, some computations in the nested CV procedure can be parallelized to mitigate this. The outer loop and inner loop of the nested CV procedure as well as the initial random iterations of

the SMBO algorithm to initialize the surrogate model can be parallelized, as these are all independent steps in the algorithm.

We have also described the workflow starting from data acquisition to the use of various data preparation methods. While various models and feature engineering and transformation approaches have been discussed in the literature, optimization of the feature engineering pipeline as a part of the hyperparameter optimization procedure or the use of the ROCKET method on fast Fourier-transformed data has not been presented before, to the best of our knowledge. We have demonstrated how to use domain knowledge to extract statistical features from specific frequency ranges of fast Fourier-transformed signals and compared the results with those obtained with the data that were transformed with the fast Fourier and ROCKET methods. In this study, there were no simulation and measurement data representing the same machine available, and a comparison of the results could not be made. This limitation shall be addressed in future work.

The logistic regression model performed better than the more advanced method CatBoost model. With simulated vibration velocity and measured vibration acceleration, using data transformed with the fast Fourier with the ROCKET methods as the input led to the best results, whereas with simulated current data, statistical features extracted from the fast Fourier-transformed data performed the best. Although the models trained with the fast Fourier-transformed data were significantly slower in making predictions when compared with feature-based models, they are fast enough for fault identification. The set of input features of the models affected the model accuracy and development time more than the number of samples, although increasing the number of training samples improved the fault detection accuracy. The evaluation of the classifiers' accuracy with respect to the measurement direction of vibration acceleration data demonstrated that data from horizontally installed sensors yielded the best results when transformed with the fast Fourier and the ROCKET methods. The predictions made with the holdout test dataset proved that the models extrapolate reasonably well as most of the samples at the minimum and maximum loads were classified correctly.

To summarize the study, we have:

1. Described and applied a novel approach to efficiently develop an accurate and reliable BRB detection model;
2. Demonstrated that a well-extrapolating BRB detection model can be developed with both simulated and measured current and vibration data;
3. Demonstrated how, e.g., the application of the ROCKET method and utilization of domain knowledge, affect the model performance;
4. Demonstrated the automation of the feature engineering process.

In an industrial setting, utilizing the measurements of multiple quantities to detect faults leads to more confident decision making. Although the model development approach here was presented in the context of broken rotor bar identification, it applies to other faults as well. Since different induction machine faults can have distinct frequency domain characteristics, it could be beneficial to further automate the exploration of various data feature engineering and transformation methods to find the most optimal one for a specific fault. Such an approach would take the modeling for fault detection purposes towards the world of automatic machine learning. Moreover, applying convolutional neural networks for fault detection without feature engineering and the application of transfer learning to improve the data efficiency of the model development process are intriguing topics for future research.

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

Funding: This work has received funding from the Arrowhead Tools project, funded by the European Commission through the European H2020 Research and Innovation programme, ECSEL Joint Undertaking, and national funding authorities from the 18 involved countries under the grant agreement number 826452.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Jenni Pippuri-Mäkeläinen and Jussi Kiljander (VTT Technical Research Centre of Finland) for their valuable comments that greatly improved the manuscript.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Nandi, S.; Bharadwaj, R.M.; Toliyat, H.A.; Parlos, A.G. Study of three phase induction motors with incipient rotor cage faults under different supply conditions. In Proceedings of the Conference Record—IAS Annual Meeting (IEEE Industry Applications Society), Phoenix, AZ, USA, 3–7 October 1999; Volume 3, pp. 1922–1928. [\[CrossRef\]](#)
2. Nandi, S.; Toliyat, H. Condition monitoring and fault diagnosis of electrical machines—A review. *IEEE Trans. Energy Convers.* **2005**, *20*, 197–204. [\[CrossRef\]](#)
3. Gangsar, P.; Tiwari, R. Signal based condition monitoring techniques for fault detection and diagnosis of induction motors: A state-of-the-art review. *Mech. Syst. Signal Process.* **2020**, *144*, 106908. [\[CrossRef\]](#)
4. Tahkola, M.; Keranen, J.; Sedov, D.; Far, M.F.; Kortelainen, J. Surrogate Modeling of Electrical Machine Torque Using Artificial Neural Networks. *IEEE Access* **2020**, *8*, 220027–220045. [\[CrossRef\]](#)
5. Kudelina, K.; Asad, B.; Vaimann, T.; Rassölkin, A.; Kallaste, A.; Van Khang, H. Methods of condition monitoring and fault detection for electrical machines. *Energies* **2021**, *14*, 7459. [\[CrossRef\]](#)
6. Rangel-Magdaleno, J.; Peregrina-Barreto, H.; Ramirez-Cortes, J.; Morales-Caporal, R.; Cruz-Vega, I. Vibration analysis of partially damaged rotor bar in induction motor under different load condition using DWT. *Shock Vib.* **2016**, *2016*, 3530464. [\[CrossRef\]](#)
7. Godoy, W.F.; da Silva, I.N.; Goedel, A.; Palacios, R.H.C.; Scalassara, P.; Morinigo-Sotelo, D.; Duque-Perez, O. Detection of broken rotor bars faults in inverter-fed induction motors. In Proceedings of the International Conference on Electrical Machines (ICEM), Alexandroupoli, Greece, 3–6 September 2018; pp. 1803–1808. [\[CrossRef\]](#)
8. Quiroz, J.C.; Mariun, N.; Mehrjou, M.R.; Izadi, M.; Misron, N.; Mohd Radzi, M.A. Fault detection of broken rotor bar in LS-PMSM using random forests. *Measurement* **2018**, *116*, 273–280. [\[CrossRef\]](#)
9. Ince, T. Real-time broken rotor bar fault detection and classification by shallow 1D convolutional neural networks. *Electr. Eng.* **2019**, *101*, 599–608. [\[CrossRef\]](#)
10. Quabeck, S.; Shangguan, W.; Scharfenstein, D.; De Doncker, R.W. Detection of broken rotor bars in induction machines using machine learning methods. In Proceedings of the International Conference on Electrical Machines and Systems (ICEMS), Hamamatsu, Japan, 24–27 November 2020; pp. 620–625. [\[CrossRef\]](#)
11. Ramu, S.K.; Irudayaraj, G.C.R.; Subramani, S.; Subramaniam, U. Broken rotor bar fault detection using Hilbert transform and neural networks applied to direct torque control of induction motor drive. *IET Power Electron.* **2020**, *13*, 3328–3338. [\[CrossRef\]](#)
12. Varma, S.; Simon, R. Bias in error estimation when using cross-validation for model selection. *BMC Bioinform.* **2006**, *7*, 91. [\[CrossRef\]](#)
13. Bergstra, J.; Yamins, D.; Cox, D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In Proceedings of the Machine Learning Research, Atlanta, GA, USA, 17–19 June 2013; Dasgupta, S., McAllester, D., Eds.; PMLR: Atlanta, GA, USA, 2013; Volume 28, pp. 115–123.
14. Hassan, O.E.; Amer, M.; Abdelsalam, A.K.; Williams, B.W. Induction motor broken rotor bar fault detection techniques based on fault signature analysis—A review. *IET Electr. Power Appl.* **2018**, *12*, 895–907. [\[CrossRef\]](#)
15. Tahkola, M.; Guangrong, Z. ATSC-NEX: Automated Time Series Classification with Sequential Model-Based Optimization and Nested Cross-Validation. *IEEE Access* **2022**, *10*, 39299–39312. [\[CrossRef\]](#)
16. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Gradient Boosting With Categorical Features Support. *arXiv* **2018**, arXiv:1810.11363.
17. Kanovic, Z.; Matic, D.; Jelicic, Z.; Rapaic, M.; Jakovljevic, B.; Kapetina, M. Induction motor broken rotor bar detection using vibration analysis—A case study. In Proceedings of the 2013 9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, SDEMPED 2013, Valencia, Spain, 27–30 August 2013; pp. 64–68. [\[CrossRef\]](#)
18. Godoy, W.F.; da Silva, I.N.; Goedel, A.; Palacios, R.H.C.; Lopes, T.D. Application of intelligent tools to detect and classify broken rotor bars in three-phase induction motors fed by an inverter. *IET Electr. Power Appl.* **2016**, *10*, 430–439. [\[CrossRef\]](#)

19. Cupertino, F.; Giordano, V.; Mininno, E.; Salvatore, L. Application of supervised and unsupervised neural networks for broken rotor bar detection in induction motors. In Proceedings of the IEEE International Conference on Electric Machines and Drives, San Antonio, TX, USA, 15 May 2005; pp. 1895–1901. [\[CrossRef\]](#)
20. Dias, C.G.; Pereira, F.H. Broken rotor bars detection in induction motors running at very low slip using a hall effect sensor. *IEEE Sens. J.* **2018**, *18*, 4602–4613. [\[CrossRef\]](#)
21. Skylvik, A.J.; Robbersmyr, K.G.; Khang, H.V. Data-driven fault diagnosis of induction motors using a stacked autoencoder network. In Proceedings of the 2019 22nd International Conference on Electrical Machines and Systems (ICEMS), Harbin, China, 11–14 August 2019; pp. 1–6. [\[CrossRef\]](#)
22. Keskes, H.; Braham, A.; Lachiri, Z. Broken rotor bar diagnosis in induction machines through stationary wavelet packet transform and multiclass wavelet SVM. *Electr. Power Syst. Res.* **2013**, *97*, 151–157. [\[CrossRef\]](#)
23. Nakamura, H.; Pandarakone, S.E.; Mizuno, Y. A novel approach for detecting broken rotor bar around rated rotating speed using frequency component and clustering. *IEEE Trans. Electr. Electron. Eng.* **2016**, *11*, S116–S122. [\[CrossRef\]](#)
24. Maitre, J.; Bouzouane, A.; Gaboury, S. A hierarchical approach for the recognition of induction machine failures. *J. Control Autom. Electr. Syst.* **2018**, *29*, 44–61. [\[CrossRef\]](#)
25. Camarena-Martinez, D.; Valtierra-Rodriguez, M.; Amezcua-Sanchez, J.P.; Granados-Lieberman, D.; Romero-Troncoso, R.J.; Garcia-Perez, A. Shannon entropy and k-means method for automatic diagnosis of broken rotor bars in induction motors using vibration signals. *Shock Vib.* **2016**, *2016*, 4860309. [\[CrossRef\]](#)
26. Zhao, Z.; Li, T.; Wu, J.; Sun, C.; Wang, S.; Yan, R.; Chen, X. Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study. *ISA Trans.* **2020**, *107*, 224–255. [\[CrossRef\]](#)
27. Ismail Fawaz, H.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Deep learning for time series classification: A review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [\[CrossRef\]](#)
28. Ganesan, S.; David, P.W.; Balachandran, P.K.; Samithas, D. Intelligent starting current-based fault identification of an induction motor operating under various power quality issues. *Energies* **2021**, *14*, 304. [\[CrossRef\]](#)
29. Tercan, H.; Guajardo, A.; Heinisch, J.; Thiele, T.; Hopmann, C.; Meisen, T. Transfer-learning: Bridging the gap between real and simulation data for machine learning in injection molding. *Procedia CIRP* **2018**, *72*, 185–190. [\[CrossRef\]](#)
30. Dempster, A.; Petitjean, F.; Webb, G.I. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min. Knowl. Discov.* **2020**, *34*, 1454–1495. [\[CrossRef\]](#)
31. Roivainen, J. Unit-Wave Response-Based Modeling of Electromechanical Noise and Vibration of Electrical Machines. Ph.D. Thesis, Helsinki University of Technology, Espoo, Finland, 2009. Available online: <http://lib.tkk.fi/Diss/2009/isbn9789512299119/isbn9789512299119.pdf> (accessed on 14 January 2022).
32. Brito, L.C.; Susto, G.A.; Brito, J.N.; Duarte, M.A. An explainable artificial intelligence approach for unsupervised fault detection and diagnosis in rotating machinery. *Mech. Syst. Signal Process.* **2022**, *163*, 108105. [\[CrossRef\]](#)
33. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*, 2nd ed.; Springer, New York, NY, USA, 2008. [\[CrossRef\]](#)
34. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
35. Langley, P.; Sage, S. Oblivious decision trees and abstract cases. In Proceedings of the Working Notes of the AAAI94 Workshop on Case-Based Reasoning, Seattle, WA, USA, 31 July–4 August 1994; pp. 113–117.