Journal of Sound and Vibration 377 (2016) 331–345

Contents lists available at ScienceDirect

Journal of Sound and Vibration

journal homepage: www.elsevier.com/locate/jsvi

Convolutional Neural Network Based Fault Detection for Rotating Machinery

Olivier Janssens [a,*,1], Viktor Slavkovikj [a,*,1], Bram Vervisch [b], Kurt Stockman [b], Mia Loccufier [b], Steven Verstockt [a], Rik Van de Walle [a], Sofie Van Hoecke [a]

[a] Data Science Lab, Department of Electronics and Information Systems, Ghent University-iMinds, St. Pietersnieuwstraat 41, 9000, Ghent, Belgium

[b] DySC Research Group, Department of Electrical Energy, Systems and Automation - Ghent University

ARTICLE INFO

Article history:
Received 19 November 2015
Received in revised form
10 March 2016
Accepted 17 May 2016
Handling Editor: K. Shin
Available online 24 May 2016

Keywords:
Condition monitoring
Fault detection
Vibration analysis
Machine learning
Convolutional neural network
Feature learning

ABSTRACT

Vibration analysis is a well-established technique for condition monitoring of rotating machines as the vibration patterns differ depending on the fault or machine condition. Currently, mainly manually-engineered features, such as the ball pass frequencies of the raceway, RMS, kurtosis an crest, are used for automatic fault detection. Unfortunately, engineering and interpreting such features requires a significant level of human expertise. To enable non-experts in vibration analysis to perform condition monitoring, the overhead of feature engineering for specific faults needs to be reduced as much as possible. Therefore, in this article we propose a feature learning model for condition monitoring based on convolutional neural networks. The goal of this approach is to autonomously learn useful features for bearing fault detection from the data itself. Several types of bearing faults such as outer-raceway faults and lubrication degradation are considered, but also healthy bearings and rotor imbalance are included. For each condition, several bearings are tested to ensure generalization of the fault-detection system. Furthermore, the feature-learning based approach is compared to a feature-engineering based approach using the same data to objectively quantify their performance. The results indicate that the feature-learning system, based on convolutional neural networks, significantly outperforms the classical feature-engineering based approach which uses manually engineered features and a random forest classifier. The former achieves an accuracy of 93.61 percent and the latter an accuracy of 87.25 percent.

1. Introduction

To reduce operational costs, prolong machines their lifetime and enhance operational uptime, condition monitoring (CM) is required. CM is used to inspect the state of the machine and to detect faulty components. Components that are often the main source of failure in rotating machines, such as wind turbines, are rolling-element bearings [1]. To monitor the condition of machine components, such as rotors, shafts, couplings, gears and also bearings, vibrations are often used. The presence of the rolling elements in the bearings induce vibrations that are inherent to the system. The position of the rolling

* Corresponding authors.
E-mail addresses: odjansse.janssens@ugent.be (O. Janssens), viktor.slavkovikj@ugent.be (V. Slavkovikj).
[1] Contributed equally.

http://dx.doi.org/10.1016/j.jsv.2016.05.027
0022-460X/

elements change continuously with respect to the load, causing a behaviour that depends upon the rotation speed. Furthermore, geometrical imperfections or surface roughness also cause vibrations. Not only are vibrations generated in normal operational conditions, but also due to faults, such as outer-raceway faults, inner-raceway faults, rolling-element faults, cage faults, imbalance and misalignment.

To detect if a fault is present, a frequency spectrum analysis is often done [2]. This technique requires the frequency spectrum to be calculated together with the fundamental frequencies of the bearings. The amplitude at these frequencies can then be monitored for anomalies. However, such a technique has many disadvantages. First, the frequency calculations have the assumption that there is no sliding, i.e., the rolling elements only roll on the raceways. Nevertheless, this is seldom the case. Often, a bearing undergoes a combination of rolling and sliding. As a consequence, the calculated frequencies may differ slightly, i.e. 1–2 percent, compared to the actual frequencies [3]. Second, if multiple faults occur simultaneously, the frequencies generated can add and subtract, obfuscating important frequencies [2]. Third, there is also the possibility that interference is induced due to additional sources of vibration, i.e. bearing looseness, hence obscuring useful features. Lastly, some faults, such as lubrication related faults, do not even manifest themselves as a new cyclic frequency [4], which makes them very hard to detect via traditional vibration analysis techniques. Because of these various challenges, manually-engineered features based on vibration signals can be difficult to interpret, especially in a real-time manner, other than by an experienced vibration analyst [2].

Opposed to feature engineering, recently there has been a considerable effort in machine learning on the development of end-to-end learning methods [5]. Instead of manually devising features that preserve the discriminative characteristics of the data, the goal of end-to-end learning is to learn the discriminative feature representation directly from input data. The latter approach does not require human expertise or prior knowledge of the problem, and is advantageous in tasks where it is challenging to develop characterizing features. Therefore, in this paper we develop a feature learning method to autonomously detect different faults in rotating machinery using vibration data. For comparison reasons, we also develop a more classical method using engineered features for fault detection. The chosen features are determined based on a literature review discussed in the next section. We test both approaches on experimental data generated by different bearing conditions and evaluate the capability of the methods in distinguishing between several fault classes.

The remainder of this article is as follows. In the next section a literature review is given. Subsequently, the data capturing procedure and the data set are discussed. Then, the feature-engineering based approach is presented. Consequently, the feature-learning based approach is discussed. Next, the results of both systems are evaluated and compared. Finally, the conclusions are presented together with possible future work for the presented research.

2. Related literature

To automatically detect faulty components, machine learning algorithms can be used. Machine learning algorithms use data to construct a model that can detect different conditions. Data used to train models are features which are constructed and extracted by an expert from raw data. Raw data, such as vibrations, can be obtained by attaching accelerometers to the machine that has to be monitored.

2.1. Feature engineering

Vibration patterns depend on the machine's condition, and are therefore very suitable to detect specific conditions. For example, imbalance, which is caused due to the shift between the principal axis of inertia and the axis of rotation, results in a high amplitude at the rotation frequency of the machine in the frequency spectrum [6]. Other faults which can be detected in a similar manner are damaged raceways, since these faults generate a peak at a specific fundamental frequency [7]. Besides indicative frequency features, it has also been shown that certain time based statistical features, such as kurtosis and crest factor, are useful in identifying a defect bearing [8]. Furthermore, it was shown that the root-mean-square (RMS), another time-based feature of the vibration signal, is indicative of the amount of separation between the rolling elements and the raceways due to lubrication in a linear bearing [9].

To summarize, several different features with a specific goal can be extracted from vibration data. However, a human expert is still required to interpret the features to identify different machine conditions or anomalies. Hence, machine learning is required to automate this interpretation process.

2.2. Machine learning

Machine learning for machine fault detection focuses on two major topics, i.e.: anomaly detection and fault/condition classification. Anomaly detection is the process of identifying measurements that do not conform to the other patterns of the data set [10]. The assumption here is that these anomalous measurements indicate that the condition of the machine has changed, e.g. a fault has occurred. Anomaly detection does not require samples from the different possible conditions, but only samples taken during normal operational conditions. Hence, anomaly detection is straight-forward to apply. Often, features, as discussed in the previous sub-section, are used by algorithms such as one-class support vector machines (SVM), Gaussian distribution fitting, clustering in combination with principal component analysis, hidden markov models and neural networks [10–13].

Opposed to anomaly detection there is condition/fault detection. The main difference is that additional to samples from normal operational conditions, samples from abnormal operational conditions are used to train the machine learning models. Whereas anomaly detection can only identify deviations from the normal conditions, fault condition/fault detection can identify which fault has occurred. Nevertheless, the disadvantage is that data from the different conditions need to be available.

Condition/fault detection also use features such as discussed in the previous section. These features are processed by machine learning algorithms such as k-nearest neighbor classifiers, naive bayes classifiers, decision trees and multi-layer perceptron classifiers [14–16]. Using these classifiers several types of faults can be accurately detected such as inner-raceway faults, outer-raceway faults and rolling element faults.

However, some faults are more difficult to detect reliably such as lubricant starvation [4], which can be caused due to grease dry-out. Lubrication has many functions, such as friction control, wear control, contamination control, temperature control and corrosion control. Lack of lubricant is often the root-cause of many bearing failures [17]. If lubricant starvation is not detected in time, other additional faults may be induced, making it more difficult to identify every individual fault.

As discussed by Kankar et al. [16] and Monte et al. [18], when several faults are present in a rotating system at the same time, the detection of the faults is more difficult. Hence, more advanced detection techniques are required. One of the possible techniques is feature learning.

2.3. Feature learning

Feature learning refers to the collection of techniques that learn a transformation or sequence of transformations of the raw data so that the data is optimally represented for the required task. This is different from feature engineering, where features are designed by experts for the required task. This is also different from feature selection. The goal of feature selection is to select the most informative subset of features from all the available features. Therefore, there is no feature learning or feature transformation during feature selection. A schematic representation of the difference between feature engineering, feature engineering in combination with feature selection and feature learning can be seen in Fig. 1. In the feature engineering part of the figure it can be seen that from the input data (X) features are extracted (ϕ) and used to train a classification algorithm ($f_\theta(.)$) that outputs predictions (Y). The learnable parameters of the classification algorithm are denoted here by θ. In the part of the figure about feature engineering in combination with feature selection, a feature selection step is added wherein a subset of the features are selected ($\psi \subseteq \phi$) that are afterwards used in the classification algorithm. In the feature learning part of the figure no hand-crafted features are extracted from the input data, instead, the input data is transformed using $t_{\theta_1}(.)$ wherein θ_1 consists of the learnable parameters of the transformation. The transformation will output a new representation of the input data that should be better suited for the classification task. Transformation steps can be repeated many times—each with their own set of learnable parameters—so that at the classification step the data is transformed optimally, i.e., optimal features are learned for the classification task.

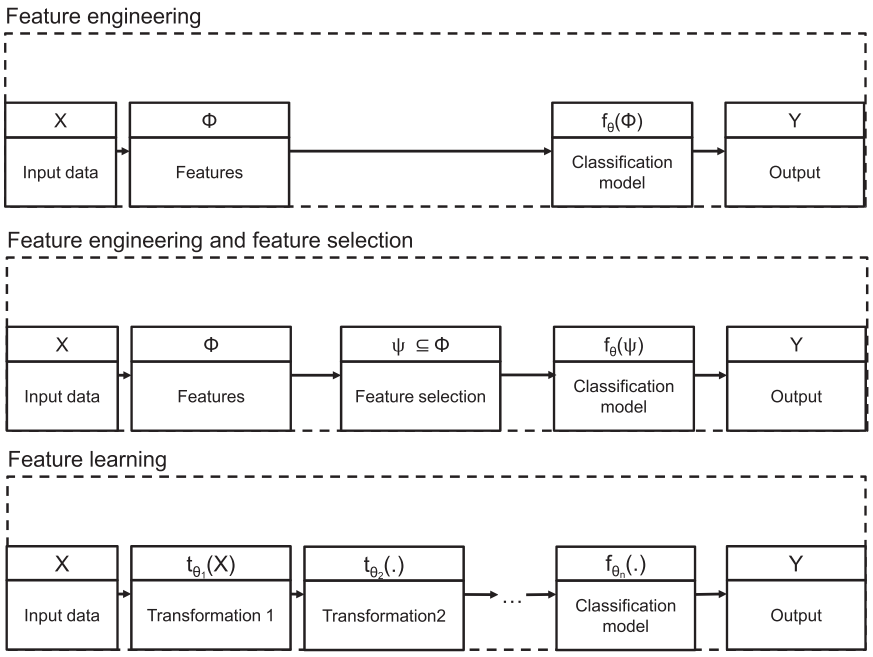

Fig. 1. Schematic representation of feature engineering, feature engineering in combination with feature extraction and feature learning.

One of the feature learning techniques which gained notable attention is sparse coding. The goal of sparse coding is to represent a signal using a linear combination of a few bases from a dictionary. This process uses two steps. The first step consists of constructing the bases of the dictionary. These bases are learned from raw data using algorithms such as k-singular value decomposition [19] or online dictionary learning [20]. The second step consists of determining the sparse coefficients which can be calculated using greedy pursuit algorithms [21] or iterative shrinking algorithms [22].

Some applications of feature learning regarding fault detection have started to emerge in recent years. For example, in the work of Wang et al. [23], dictionary learning is employed for bearing fault detection, illustrating that more descriptive features can be learned from the raw data. Feature learning has also proven to work well for noisy signals. Often, due to vibrations of other mechanical components, vibration signals contain more than just the vibrations generated from the bearing, therefore, making the identification of a fault more difficult. In the work of Deng et al. [24], feature learning is used to extract impulse features from noisy signals generated by an aircraft engine. In their work, features are extracted by fusion sparse coding and online dictionary learning resulting in a de-noised signal which allows for a fault frequency to be accurately identified.

Besides sparse coding and dictionary learning it is also possible to use neural networks on raw data for feature-learning purposes. Neural networks consist of several layers of units or nodes. The units between layers are connected by weights, which represent the adaptive parameters of the model. Each unit from a given layer computes a linear combination of the input to the unit, followed by a nonlinear activation function (such as the sigmoidal function $\sigma(x)=(1+e^{-x})^{-1}$, or $\tanh(x)$). If we represent the vector of inputs to layer k of the network with $\mathbf{x}_{k-1}$, and with $\mathbf{W}_k$ the matrix containing all weights of the connections between layer $k-1$ and k, and with $\mathbf{b}_k$ denoting the vector of bias weights. Then, the output vector $\mathbf{x}_k$ of layer k is given as:

$$\mathbf{x}_k=\sigma(\mathbf{W}_k\mathbf{x}_{k-1}+\mathbf{b}_k), \tag{1}$$

which corresponds to a single transform $t_{\theta_k}(\cdot)$ as depicted in Fig. 1. Given an input $\mathbf{x}$ to a feed-forward network, and a desired output $\mathbf{z}$ of the network, we use an error measure (such as categorical cross-entropy error) to quantify the error between the desired output $\mathbf{z}$, and the predicted output $\mathbf{y}$ of the network. Gradient descent is then used to adjust the learnable parameters of the network so that the error is minimized.

One possible neural network configuration that can be used is called a sparse auto-encoder (SAO). A SAO is a neural network that has the goal to reconstruct the input signal using a limited amount of nodes in the hidden layer. By limiting the number of hidden units, compact, useful features can be learned from the data. Furthermore, when stacking these SAOs, different levels of feature abstractions can be learned. An example of the applications of a stacked SAO for machine fault detection is given by [25].

Our feature learning model is based on convolutional neural networks (CNNs) [26], which have been proven successful in many domains [27–30]. CNNs have several advantages compared to other feature-learning techniques, such as the ones discussed above. First, similar to stacked SAOs, CNNs autonomously learn multiple levels of representations of the data through their layered structure. This enables complex features to be learned [30]. Second, a CNN is an end-to-end learning system, therefore, only a single system has to be optimized. Finally, CNNs are used to exploit the spatial structure in the data. In case of a frequency spectrum of a vibration signal, we define spatial structure as the sequence of frequencies. An example to clarify: due to the combination of sliding and rolling of a rolling element, the energy which is expected to be contained in a fundamental frequency might be partially present in frequencies close to the fundamental frequency. Hence, making use of this information might improve the fault detection. To the best of our knowledge, only one article emerged in recent years which uses convolution neural networks for machine fault detection [31]. In [31], feature extraction is applied to extract features such as skewness, kurtosis, standard deviation, and mean. Afterwards, a convolutional neural network is applied on the extracted features. It should be noted that this is different from the work presented in this article as here, no feature extraction is used and the CNN based model is applied on the raw frequency spectrum of vibration data so that the network can learn features itself.

A CNN works as follow: given an input containing multiple channels, such as an image or several vibration signals combined, a CNN layer computes a similar transform as the one in Eq. 1, with the difference that the adjustable parameters of the layer are organized as a set of filters (or filter bank) and convolved over the input to produce the layer's output. The output of a CNN layer is a 3D tensor, which consists of a stack of matrices called feature maps, and can be used as input to a higher level layer of the CNN model. The weights in the filter bank are shared over the input, which effectively exploits the local spatial statistics, while reducing the number of trainable parameters. The operation can be represented as:

$$\mathbf{X}_k^{(m)}=\sigma\left(\sum_{c=1}^{C}\mathbf{W}_k^{(c,m)}*\mathbf{X}_{k-1}^{(c)}+\mathbf{B}_k^{(m)}\right). \tag{2}$$

In Eq. (2) the layer of the network is denoted with k as before, and the $*$ operator is used for the 2D convolution of channel $c=1,\ldots,C$ of the input $\mathbf{X}_{k-1}$ and the filter $\mathbf{W}_k^{(c,m)}$, which is responsible for the m-th output feature map $\mathbf{X}_k^{(m)}$, where $m=1,\ldots,M$. The matrix $\mathbf{B}_k^{(m)}$ contains the bias weights. Finally, a nonlinear activation function σ is applied to the sum of convolutions to obtain the final output.

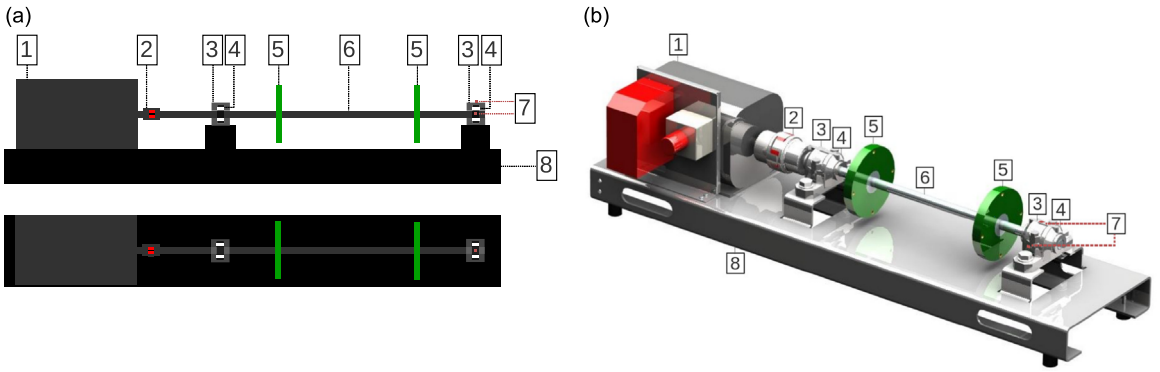

Fig. 2. The used set-up. The labels are 1. servo-motor; 2. coupling; 3. bearing housing; 4. bearing; 5. disk; 6. shaft; 7. accelerometer 8. metal plate. (a) Side-view and top-view of the set-up. (b) 3D image of the set-up.

Table 1
Technical specifications of the test set-up.

Property	Value
Bearing code	FAG 22205-E1-K
Bearing type:	Spherical roller bearing with tapered bore & adapter sleeve
Housing code	SNV052-F-L
Housing type	Closed plummer block
Grease	Molykote BR 2 plus
Rotation speed	25 Hz
Sample frequency	51200 Hz
Accelerometer type	IEPA
Accelerometer product type	4534-B

3. Methodology

Two methods for bearing fault detection are described in this section, i.e., one based on feature engineering and the other based on feature learning. In order to compare these techniques, we performed experiments on a data set which was created using the test set-up discussed below.

3.1. Test set-up

A visualization of our set-up is shown in Fig. 2 and the technical specifications are summarized in Table 1. In the set-up there are two bearing housings. Out of the two housings, the housing farther from the motor contained the different fault induced bearings during the CM tests. On this housing, two accelerometers were mounted perpendicular to one another to measure the vibrations in the x- and y-direction, i.e., one on top of the housing, and one on the back of the housing. The faults and conditions introduced are:

1. Healthy bearing (HB)
2. Mildly inadequately lubricated bearing (MILB)
3. Extremely inadequately lubricated bearing (EILB)
4. Outer-raceway fault (ORF)
5. Healthy bearing during imbalance (HB-IM)
6. Mildly inadequately lubricated bearing during imbalance (MILB-IM)
7. Extremely inadequately lubricated bearing during imbalance (EILB-IM)
8. Outer-raceway fault during imbalance (ORF-IM)

Some images of the induced conditions are presented in Fig. 3. To imitate an ORF, three small shallow grooves were added mechanically on the bearing's outer-raceway Fig. 3. Also grease was added to the bearing as lubricant. To calculate the amount of grease required, Eq. (3) was used, where D is the outer diameter of the bearing and B the inner diameter [32]. In our set-up we used bearing with diameter D=52 mm and B=18 mm.

$$m = D \cdot B \cdot 0.0027\ [g] \tag{3}$$

Both the HBs and those with an ORF contain 2.5 g of grease, in addition to the 20 g of grease in the grease reservoir within the housing. The amount of grease is determined so that the housing cavities are filled to the recommended 60

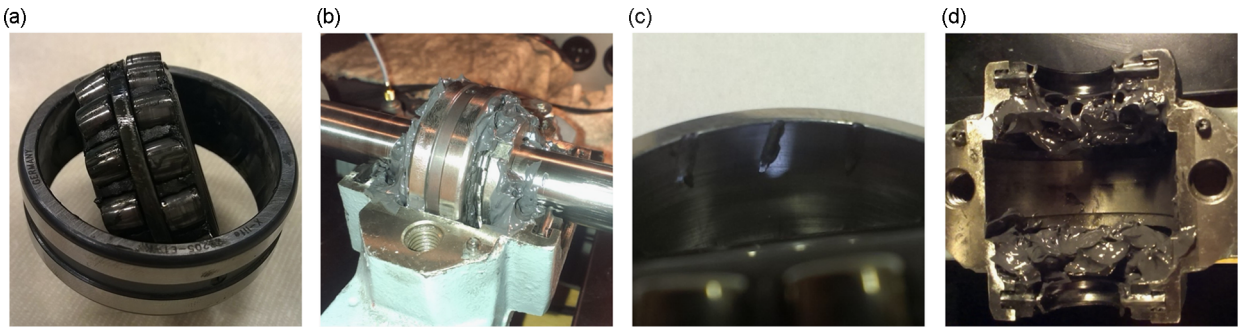

Fig. 3. Examples of different bearing conditions and the grease reservoir. (a) Mildly inadequately lubricated bearing outside the housing. (b) Healthy Bearing in an open housing. (c) Three small, shallow grooves to imitate an outer-raceway fault. (d) Grease reservoir in the bottom of the housing.

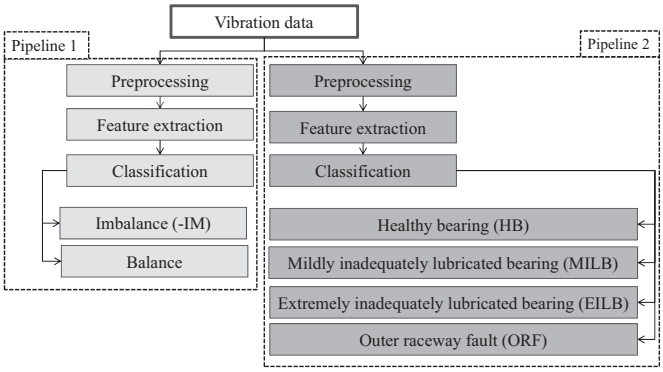

Fig. 4. High-level representation of the architecture.

percent [33]. For the MILBs, the grease reservoir is removed and the grease on the bearing is diluted. Similarly, for the EILBs no reservoir is present, and the grease in the bearings is diluted further. All four conditions are also tested during rotor imbalance. The imbalance is created by adding a 13 g bolt to the outer disk at a radius of 5.4 cm. By means of this set-up, a data set was created incorporating the introduced healthy and faulty conditions.

3.2. Data set

For every condition, five bearings were tested, resulting in 40 test runs in total. Each test had a runtime of one hour from which the last 10 minutes of vibration data in the x and y plane were captured using the accelerometers. In the next section, the feature engineering techniques, applied on these measurements are discussed in detail.

3.3. Feature engineering

Each measurement is assigned two labels, i.e., one for the machine condition and one for the bearing condition. Hence, we regard the fault detection task as a combination of both a binary classification problem and a multi-class classification problem. Every 10-minute vibration recording is classified by the binary classifier as balanced or imbalanced, and by the multi-class classifier according to HB, MILB, EILB, or ORF. This solution is depicted by the architecture in Fig. 4. The raw data is used in two pipelines, each with their own respective feature extraction step, classification models and labels. By using an architecture with two pipelines, the combination of the two labels generated for each sample give the final fault and condition classification (one of the eight classes as listed in Section 3). It is determined experimentally that a two pipeline system works better than a single pipeline system, which was also confirmed by our previous work [34].

3.3.1. Pipeline one

The goal of pipeline one is to determine if there is rotor imbalance, regardless of the presence of a bearing fault (MILB, EILB, ORF, or HB). As described in the previous section, the first step of the pipeline is feature extraction.

3.3.2. Feature extraction

As discussed in the literature review, imbalance is detectable by observing if there is a high amplitude at the rotation frequency of the machine. Note here that the sampling frequency of the accelerometers is very high to merely capture the

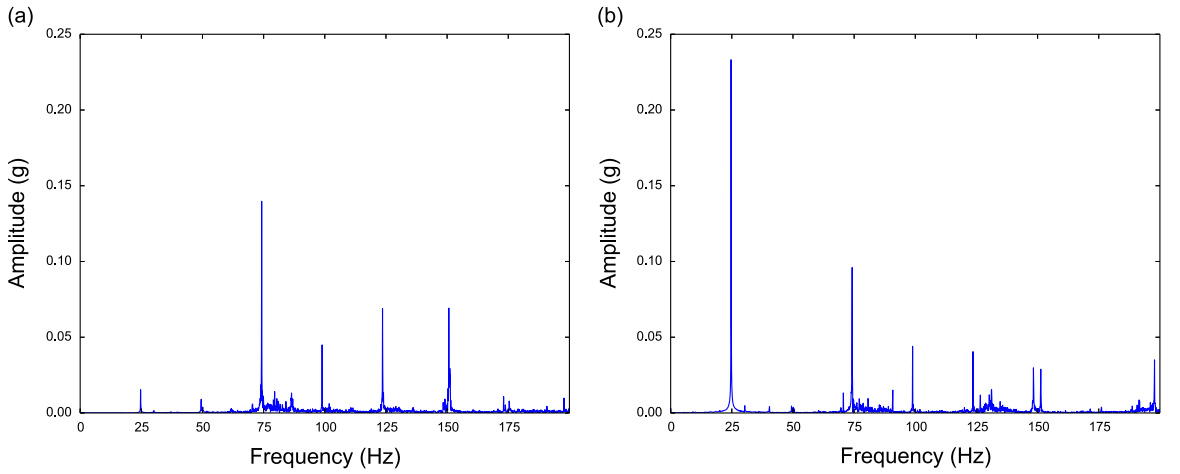

Fig. 5. Frequency plots during imbalance and balance. (a) Frequency spectrum of a bearing when the system is in balance. (b) Frequency spectrum of a bearing during imbalance. The peak at ~25 Hz is clearly visible.

rotation frequency of 25 Hz. However, to detect bearing faults, which is done in the second pipeline, the resonant frequency of the bearings should be measurable. This frequency is usually above 10 kHz, hence, the sampling frequency has to be at least 20 kHz. Nevertheless, imbalance can also be detected using the chosen accelerometers. The first step to extract the amplitude at the rotation frequency is windowing. A window contains one minute of vibration data, and overlaps by 50 percent with its neighbouring window. This means that from every 10-minute vibration data recording, 19 windows are extracted, each containing 60 seconds · 51,200 Hz = 3,072,000 samples. As there are two accelerometers mounted on the bearing housing, there are in fact twice as many samples. The window length is experimentally determined, and provides the most optimal results. Also, a relatively large window is preferred as it enables a small bin resolution for the Discrete Fourier Transform (DFT), which is used in the second step. In fact, when applying the DFT, the frequency resolution is 1 / length of the window or 1 / 60 seconds = 0.0166 Hz/bin, allowing for small frequency differences to be detected. An example of a frequency plot is given in Fig. 5. As can be seen, a peak close to the rotation frequency can be observed when there is imbalance. In the final step, from this frequency spectrum the frequency bellow 90 Hz corresponding to the highest amplitude is chosen as a feature. This feature is extracted for the two vibration signals per window, resulting in 19 samples per test run, each containing two features. After these features are calculated, classification is applied.

3.3.3. *Classification:*

The samples gathered from the measurements during imbalance are expected to have a maximum amplitude around 25 Hz. Therefore, a simple classifier suffices to classify the samples. Logistic regression is chosen here as it functions as a very fast linear and binary classifier. Nevertheless, other options such as a linear support vector machine or a decision tree are equally valid but have a higher computational complexity. An example of the decision boundary found by logistic regression can be seen in Fig. 6.

3.3.4. *Pipeline Two*

The goal of pipeline two is to identify the specific bearing condition regardless of balance or imbalance. As this is a more difficult task compared to the task of pipeline one, a larger set of features is used.

3.3.5. *Features extraction*

Similar to pipeline one, windowing is applied in this pipeline too. From every window, several features are calculated. First of all, three statistical features are calculated: RMS, kurtosis, and crest factor. These features are chosen as they have been proven useful for bearing fault detection [8,9]. The RMS, kurtosis and crest factor are calculated according to Eqs. (4), (5), (6) respectively, where $\mathbf{x}$ is a vector of N samples in a window, and μ and σ respectively denote the mean and the standard deviation of $\mathbf{x}$.

$$RMS = \sqrt{\frac{1}{N}\sum_{i}^{N} x_i^2} \tag{4}$$

$$Kurtosis = \frac{\sum_i^N (x_i - \mu)^4}{N\sigma^4} \tag{5}$$

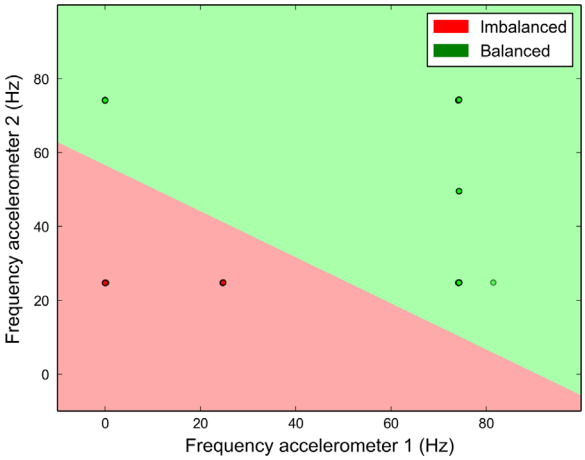

Fig. 6. Scatter plot of the frequencies with maximum amplitude, extracted for both accelerometers for all the samples. Also, the decision boundary determined by logistic regression is plotted. Accelerometer 1 = accelerometer on top of the housing. Accelerometer 2 = accelerometer on the side of the housing.

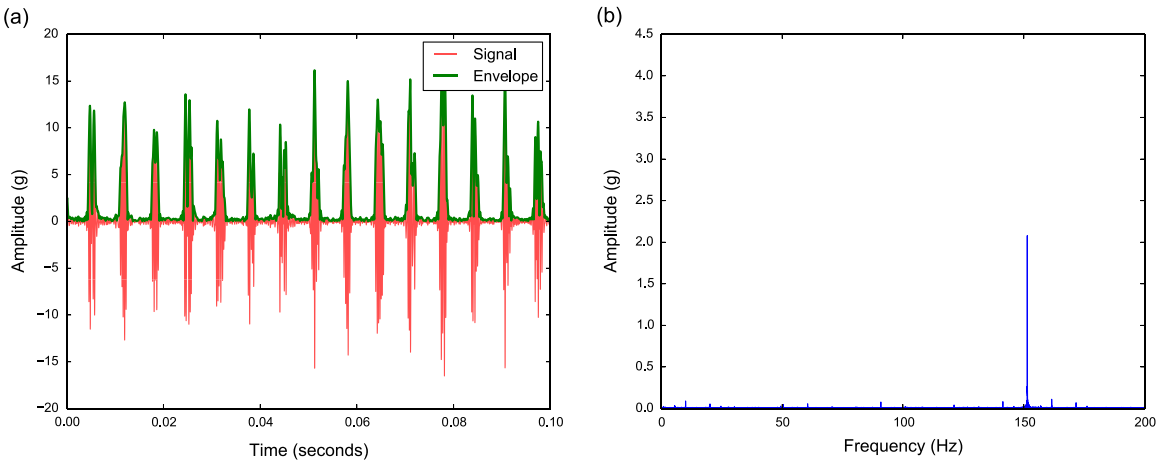

Fig. 7. Vibration signal generated by an outer-raceway fault together with its frequency spectrum. (a) The signal together with the corresponding envelope. (b) Frequency spectrum of the envelope signal.

$$Crest = \frac{\max(|\mathbf{x}|)}{RMS} \tag{6}$$

When a rolling element hits a fault in the outer raceway, the natural frequency of the raceway is excited, resulting in a high frequency burst of energy which decays and is then excited again as the next rolling element hits the fault. This high frequency impulse is superimposed, that is, amplitude modulated, on a carrier signal which originates from the rotating machine. To identify a fault, it is necessary to detect the frequency of occurrence of these high energy bursts. Therefore, envelope detection is applied. First a band pass filter is used. All frequencies bellow 1 kHz, such as the carrier frequency, are removed. Also, frequencies above 20 kHz that interfere with high frequency signals originated from the impact are filtered out. After this filter process, the high frequency impacts should be better isolated. The final step is to determine the envelope signal which will have a frequency equal to the frequency of occurrence of the high energy bursts. The envelope is determined by taking the magnitude of the analytical signal, which is computed using the Hilbert Huang transform. An example of this envelope signal can be seen in Fig. 7a. When there is an outer-raceway fault, the frequency of the envelope signal will manifest itself at the ball pass frequency of the outer raceway (BPFO).

The BPFO can be calculated using Eq. (7), where n is the amount of rolling elements, f the rotation frequency, d the diameter of the rolling elements, D the diameter of the rolling-element cage and α the contact angle. This results in a BPFO at 150.41 Hz for the chosen bearings. To summarize, if the frequency of the envelope signal is near the BPFO and has a high amplitude, it can be concluded that an outer-raceway fault is present. An example of this can be seen in Fig. 7b. From the envelope frequency, the maximum amplitude and the corresponding frequency are extracted as features. These two features

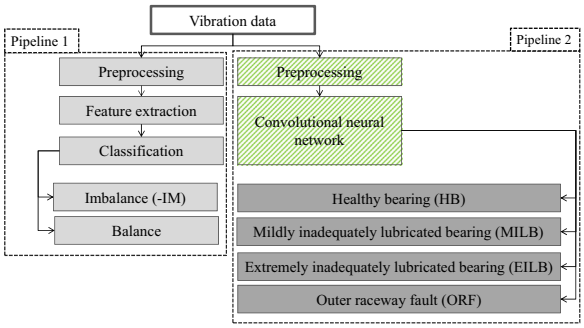

Fig. 8. High-level representation of the proposed feature-learning architecture. The modules in green indicate the modified parts compared to the feature-engineering based approach.

are calculated for both vibrations signals.

$$BPFO = \frac{1}{2} nf \left(1 - \frac{d}{D} \cos \alpha\right) \tag{7}$$

As in pipeline one, all these features are extracted from overlapping windows on the vibration signals, resulting in 19 samples per measurement. Each sample consists of 14 features (seven features per accelerometer), i.e.: RMS, kurtosis, crest factor, frequency of the highest amplitude of the envelope-signal spectrum, the maximum amplitude in the envelope-signal spectrum, rotation frequency, and amplitude of the rotation frequency. The rotation frequency and it's amplitude are also added as they improve the classification results.

3.3.6. *Classification*

Classifying between the four different faults is a more difficult task, hence a random forest (RF) classifier is chosen [35]. A RF classifier is a non-linear, multi-class ensemble classifier based on decision trees. Due to this ensemble technique, parallelism is inherent to a RF, enabling a fast training phase. Also, a RF is easy to use since it requires a minimal amount of meta parameters to tune. The most important parameter to tune is the number of individual decision trees contained in the forest, which we fix at 200 trees, as adding more trees does not improve the results further.

To put the results of the RF in perspective, tests are also done using a support vector machine (SVM) with a linear kernel, polynomial kernel and a radial basis kernel. For the SVM the hyper parameters C, which determines the penalty on misclassifications, γ, which determines how far the influence of a single training example reaches, and the *degree* for the polynomial kernel, are empirically determined using grid-search. Grid-search is a hyper-parameter optimization technique wherein all feasible combinations of hyperparameters, such as those mentioned above, are tested.

3.4. *Feature learning*

Similar to the feature-engineering based approach, the feature-learning based approach uses a two pipeline system as depicted in Fig. 8. As can be seen from the feature-engineering based approach, the binary classification problem of balanced versus imbalanced samples can already be effectively solved using pipeline one. Therefore, we reuse this pipeline here. Nevertheless, for the detection of the four specific bearing conditions: HB, MILB, EILB, and ORF a feature learning model is proposed, which forms the second pipeline.

3.4.1. *CNN model*

Our proposed feature-learning approach is based on a convolutional neural network model. More specifically, a CNN model similar to the one proposed by Slavkovikj et al. [36] is used. However, the model applied here leverages the capacity of the network for exploiting the spatial structure in data to effectively capture the covariance of the frequency decomposition of the accelerometer signals. Note that the two accelerometers are placed perpendicular to one another, and the goal here is to differentiate between the complex bearing conditions by learning the patterns of changes of the joint accelerometer signals. Fig. 9 shows a diagram of the proposed CNN architecture. The convolutional layer corresponds to Eq. 2, and the fully connected layer to Eq. (1). The different variables of the network and their dimensionality are listed in Table 2.

The architecture which yielded the best results in the experiments consists of one convolutional layer with width 64, followed by a fully-connected layer with 200 units. Several configurations of the network were tested. Table A1 (Appendix A) contains a list of configurations of other well performing networks. The height of the convolutional layer corresponds to the two signals originating from the accelerometers. The input signals are preprocessed in order to train the model. First, the accelerometer signals are scaled to have zero mean and unit variance. Then, from the training set signals, non-overlapping windows are extracted containing one second of measurement samples. For each window of extracted samples, the DFT is calculated. The amplitudes of the frequency decompositions are then used as training samples for the neural network model. It was determined that the accelerometer's sampling resolution could be lowered without affecting the output of the

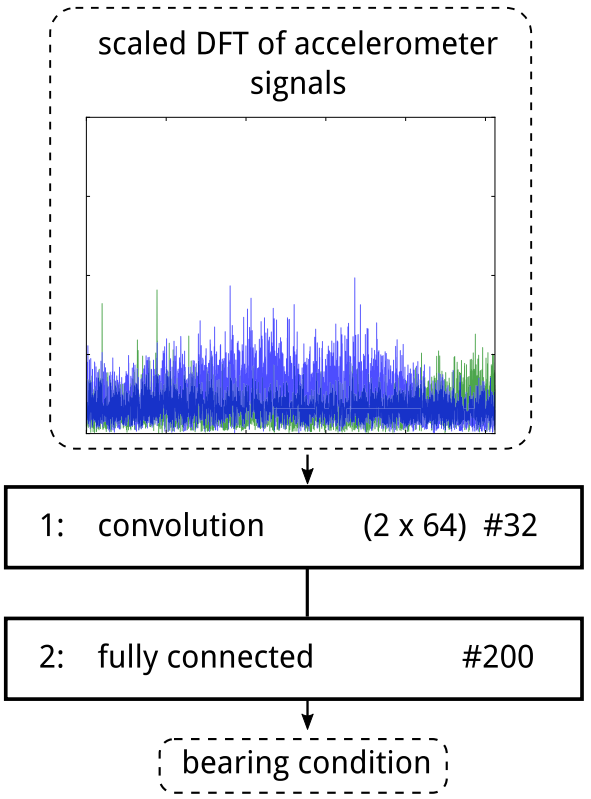

Fig. 9. Architecture of the CNN model for bearing fault detection. The size of the convolutional filter is denoted as ($h \times w$), and # denotes the number of convolutional filters, and the number of hidden units in the fully-connected layer.

Table 2
Dimensions of the variables in the convolutional neural network. X, W, b, O denote the input to a layer, the weights, the bias and the output of the layers.

Variable	Dimensionality
X_0	$\mathbb{R}^{(S,C,h,w)}$ S=100, C=1, h=2, w=5120
W_1	$\mathbb{R}^{(S,C,h,w)}$ S=32, C=1, h=2, w=64
b_1	$\mathbb{R}^{(S)}$ S=32
O_1	$\mathbb{R}^{(S,C,h,w)}$ S=100, C=32, h=1, w=5057
X_1	$\mathbb{R}^{(S,C,h,w)}$ S=100, C=32, h=1, w=5113
W_2	$\mathbb{R}^{(S,N)}$ S=161824, N=200
b_2	$\mathbb{R}^{(S)}$ S=200
O_2	$\mathbb{R}^{(S,N)}$ S=100, N=200
X_2	$\mathbb{R}^{(S,N)}$ S=100, N=200
W_3	$\mathbb{R}^{(S,N)}$ S=200, N=4
b_3	$\mathbb{R}^{(S)}$ S=4
O_3	$\mathbb{R}^{(S,N)}$ S=100, N=4

model. Therefore, 5-fold subsampling is applied on the original accelerometer data. The CNN model was trained using minibatch gradient descent and momentum [37], using 100 training examples per minibatch.

It has been shown that by using a deep architecture, i.e., a network with many layers, the network becomes more robust to the variation in the data [38]. Hence, if the dataset has a lot of variation, a deep architecture is required. As the manifestation of the different faults considered here shows little variation, a shallow architecture suffices. Furthermore, the initial layers of CNNs learn the fastest, hence a short training time is sufficient to achieve convergence [38]. Several variations of the proposed network were tested by varying the number of convolutional and fully connected layers, and the number of units per layer. For our particular use case, it was determined that a deep version of the proposed architecture does not yield better results.

4. Results

To assess the feature-engineering based approach and the feature-learning based approach, the evaluation metrics, the evaluation procedure, and the obtained results are discussed in this section.

4.1. Evaluation metrics

To quantify the performance of the different classifiers, four error measurements are calculated: accuracy, precision, recall and f1-score for which the formulas can be seen in Eqs. (8)–(11), with $|TP|$ being the amount of true positive classifications; $|TN|$, the amount of true negative classifications; $|FP|$, the amount of false positive classifications, e.g. a false alarm, and $|FN|$, the amount of false negative classifications, e.g. missed faults. These different metrics are chosen because they directly reflect the impact on CM requirements. If a CM system triggers an alarm when the classifier supposedly detects a fault, it is more interesting to be alerted of all the faults, even if among the fault there are some false alarms. Nevertheless, the operator does not want to have too many false alarms since this increases the operational cost due to unnecessary downtime. In other words, if many alarms are triggered, a lot of faults are brought to the operator's attention (higher recall), nevertheless, there are also more false alarms (lower precision). On the other hand, if only real faults are flagged, and some are missed and there are no false alarms, there will be a high precision, but a low recall. A good classifier will maximize both, so that an alarm is only trigged when there is an actual fault, no faults are missed and there are no false alarms. This combination is expressed directly in the f1-score. Also the accuracy is chosen because it is easy to interpret as it is the ratio between the amount of correctly classified samples and the total amount of samples.

$$accuracy = \frac{|TP| + |TN|}{|TP| + |FP| + |FN| + |TN|} \tag{8}$$

$$precision = \frac{|TP|}{|TP| + |FP|} \tag{9}$$

$$recall = \frac{|TP|}{|TP| + |FN|} \tag{10}$$

$$f1 - score = 2\frac{precision * recall}{precision + recall} \tag{11}$$

4.2. Evaluation procedure

To objectively evaluate the performance of the systems, leave-one-bearing-out cross-validation is used. Hence, from the 40 recordings, 32 recordings, are used to train the system, and 8 recordings, i.e., one recording for each condition from a single bearing, are used to test the system. This procedure is done five times so that every bearing is used once for testing, assuring that the system provides a generalized solution. For the convolution neural network this means that the training sets have the following dimensionality: $\mathbb{R}^{(S,C,h,w)}$ where S=19200, C=1, h=2, w=5120, and the test sets: $\mathbb{R}^{(S,C,h,w)}$ where S=4800, C=1, h=2, w=5120.

4.3. Feature engineering results

4.3.1. Pipeline 1

As Fig. 6 illustrates, classifying between balance and imbalance is a trivial task. Therefore, the mean accuracy, recall, precision and f1-score, achieved by the RFC during the leave-one-bearing-out cross-validation, are 100 percent (σ= 0 percent). To determine the importance of the features the classification task was repeated using a random forest classifier. For the rotation frequency extracted from the accelerometer on top of the housing the importance of the feature is 73.28 percent (σ= 5.90 percent) and for the accelerometer on the back of the bearing housing 26.72 percent (σ= 5.90 percent). As only the rotation frequency for the two accelerometers are used as features, it is possible to see which of the accelerometers provides more discriminative information.

4.3.2. Pipeline 2

The results achieved by pipeline two are summarized in Table 3. As can be seen, the RFC classifier outperforms the different SVMs. Generally, it can be stated that classifying between the different bearing condition is more difficult. When visualizing the confusion matrix of the RFC Fig. 10a, it can be seen that the system can perfectly identify a healthy bearing. Also, outer-raceway faults are nearly always detectable. Generally, it can be stated that a MILB is the most difficult to detect as it can be confused as an EILB or HB. This is possibly due to the fact that the manually-engineered features do not contain enough information, or do not represent the required information in such a way that the MILB samples can be distinguished from other conditions.

Due to the use of RFCs, the feature importance can be calculated for the features used in this pipeline too. The results are listed in Table 4. Several observations can be made regarding this table. First, the amplitude of the fault frequencies are

Table 3
Performance results of the two-pipeline system, which uses hand-crafted features, using leave-one-bearing-out cross validation, executed 10 times. RFC=random forest classifier, SVM 1=SVM using a linear kernel ($C=10^5$), SVM 2=SVM using a polynomial kernel ($degree=3$, $C=1$, $\gamma=1$), SVM 3=SVM using a radial basis function kernel ($C=10$, $\gamma=0.3$).

Metric	RFC	SVM 1	SVM 2	SVM 3
Accuracy	87.25 % ($\sigma=$ 8.10 %)	72.5 % ($\sigma=$ 18.37 %)	80 % ($\sigma=$ 20.31 %)	77.5 % ($\sigma=$ 18.37 %)
Precision	89.83 % ($\sigma=$ 8.21%)	73.75 % ($\sigma=$ 19.70 %)	80 % ($\sigma=$ 23.78 %)	82.08 % ($\sigma=$ 15.78 %)
Recall	87.25 % ($\sigma=$ 8.10 %)	72.5 % ($\sigma=$ 18.37 %)	80 % ($\sigma=$ 20.31 %)	77.5 % ($\sigma=$ 18.37 %)
F1-score	86.73 % ($\sigma=$ 8.14 %)	73.12 % ($\sigma=$ 19.01 %)	80 % ($\sigma=$ 21.91 %)	79.73 % ($\sigma=$ 16.98 %)

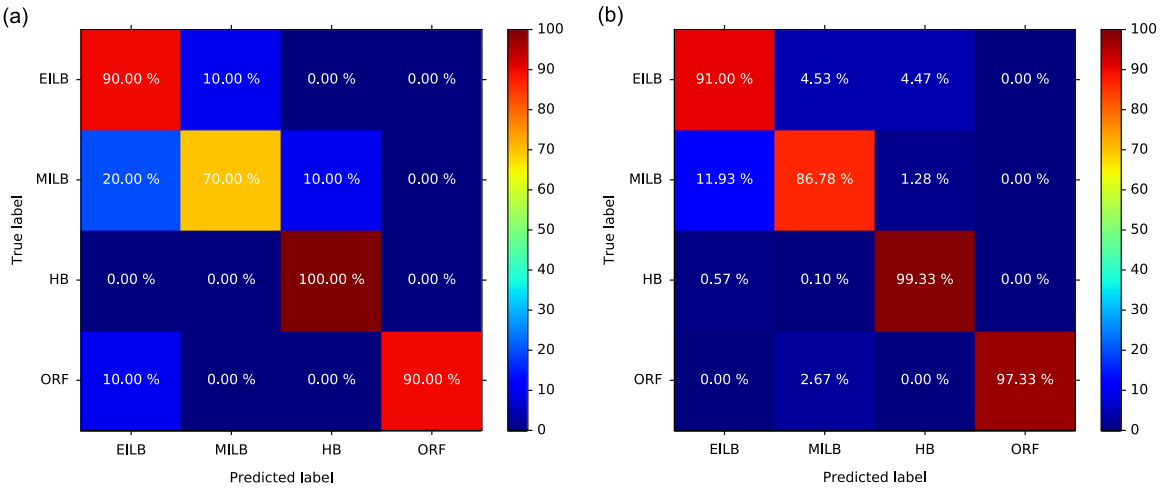

Fig. 10. (a) Confusion matrix of the bearing faults classification using hand-crafted features. (b) Confusion matrix of the bearing faults classification using the convolutional neural network.

Table 4
Feature importance scores according to the RFCs for pipeline two using hand-crafted features.

Feature	Score
RMS (1)	18.49 % (σ=1.34)
RMS (2)	13.14 % (σ=1.48)
Fault frequency amplitude (1)	8.30 % (σ=0.80)
Fault frequency amplitude (2)	14.88 % (σ=2.10)
Fault freq (1)	4.86 % (σ=1.08)
Fault freq (2)	8.18 % (σ=0.70)
Kurtosis (1)	4.99 % (σ=1.24)
Kurtosis (2)	3.48 % (σ=1.20)
Crest (1)	7.01 % (σ=0.89)
Crest (2)	1.44 % (σ=0.40)
Rotation frequency (1)	2.95 % (σ=0.39)
Rotation frequency (2)	3.07 % (σ=0.96)
Rotation frequency amplitude (1)	2.84 % (σ=0.37)
Rotation frequency amplitude (2)	6.37 % (σ=0.83)

important to the model, which is due to the direct relation to the outer-raceway fault. Second, the RMS is very important to the model, which is to be expected as the RMS is possibly indicative of the separation between the rolling elements and raceways due to lubricant (as discussed for a linear bearing in Section 2). Third, the crest features seem to be less important. A likely explanation is that since crest incorporates the RMS, which is already directly available to the model, it does not provide much additional information. Finally, the kurtosis features are also less important to the model. As kurtosis is indicative of bearing damage [7], possibly some of the required information is already directly provided to the model by the fault frequencies and the rotation frequencies.

4.3.3. Combination

The first pipeline distinguishes between imbalance and balance, whereas the second pipeline distinguishes between, HB, MILB, EILB and ORF. Although the two pipelines work independently from one-another and have their own conditions to

Table 5
Performance results of convolutional neural network based fault detection system using leave-one-bearing-out cross validation, executed 10 times.

Metric	Score
Accuracy	93.61 % (σ= 6.97 %)
Precision	94.52 % (σ= 6.0 %)
Recall	93.6 % (σ= 7.03 %)
F1-score	94.06 % (σ= 6.47 %)

distinguish, in the end they need to be combined to get the most accurate fault diagnosis. The eventual system needs to distinguish between HB, HB-IM, MILB, MILB-IM, EILB, EILB-IM, ORF and ORF-IM. As pipeline one has an accuracy of 100 percent and pipeline two 87.25 percent, the eventual system is able to distinguish between the 8 conditions with an overall accuracy of 87.25 percent.

A single-pipeline system was also tested where the goal was to immediately classify between the 8 faults/conditions. The test indicated that a single pipeline-system significantly under performs compared to a two-pipeline system.

To put the results of the two-pipeline system in perspective, the results of the feature-learning based approach are presented in the next section.

4.4. Feature learning results

The feature-learning based approach also uses a two-pipeline system, as it also outperforms a single-pipeline system. The feature-learning based approach reuses pipeline one, which detects imbalance. Hence, pipeline one also has an accuracy, precision, recall and f1-score of a 100 percent (σ=0 percent). For the second pipeline, which makes a distinction between HB, MILB, EILB and ORF, the results can be seen in Table 5. As can be seen, for every metric the results are better. A paired two-tailed t-test was also performed based on the cross-validation results, and it can be concluded that for every metric the convolutional neural networks perform significantly ($p < 0.05$) better compared to the feature-engineering based approach. However, as can be seen in Fig. 10b, the classifier does still make some errors related to lubrication degradation, i.e. EILB and MILB, leaving room for future research and optimization.

5. Conclusion and future work

In this article feature learning is used in the form of a convolutional neural network model, which is an end-to-end machine learning system. This CNN model is not applied on extracted features such as kurtosis, skewness, mean or standard deviation but on the raw amplitudes of the frequency spectrum of the vibration data. By applying the convolutional neural network on this raw data, the network learns transformations on the data that result in better representation of the data for the eventual classification task in the output layer. The major advantage of an end-to-end machine learning system which uses feature learning is that less domain expertise is required to achieve very good results, as has been shown for computer vision research in the past. We illustrated this by comparing the convolutional neural network approach to a classical approach which uses feature extraction and a random forest classifier. Our results show that by using the proposed convolutional neural network model, better results can be achieved for the detection of different faults, such as outer-raceway faults, and different levels of lubricant degradation. Compared with a classical manual feature extraction approach, the CNN based method yields an overall increase in classification accuracy of approximately 6 percent, without relying on extensive domain knowledge for detecting faults.

Future work will consist of testing the convolutional neural network approach on more conditions. Furthermore, as there are still some misclassifications possible, additional sensors will be considered. An example of a useful sensor is a thermal camera. It has been shown that by the use of thermal cameras, lubrication degradation is easily detectable, possibly enabling a strong multi-sensor based fault detection system [34].

Acknowledgements

This work was partly funded by the O&M Excellence project, a VIS project of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT), and has been performed in the framework of the Offshore Wind Infrastructure Application Lab (http://www.owi-lab.be).

Appendix A. Parameter tuning

The convolutional neural network were trained on a GPU. The used GPU was a Nvidia GeForce GTX TITAN Black. The results for different network configurations which yielded good results can be found in Table A1.

Table A1
Parameter tuning of the convolutional neural network.

Number	Width	Feature maps	Accuracy	Time per epoch
1	8	8	91.75 % ($\sigma = 10.76\%$)	3.13 s ($\sigma = 0.11$ *s*)
2	8	16	92.62 /% ($\sigma = 9.31\%$)	4.94 s ($\sigma = 0.21$ *s*)
3	8	32	92.51 % ($\sigma = 9.55\%$)	8.87 s ($\sigma = 0.41$ *s*)
4	8	64	92.80 % ($\sigma = 9.14\%$)	16.99 s ($\sigma = 0.64$ *s*)
5	8	128	92.48 %($\sigma = 9.73\%$)	37.18 s ($\sigma = 1.31$ *s*)
6	2	32	91.55 %($\sigma = 11.13\%$)	8.96 s ($\sigma = 0.32$ *s*)
7	4	32	92.21 % ($\sigma = 9.97\%$)	8.99 s ($\sigma = 0.30$ *s*)
8	16	32	92.28 % ($\sigma = 9.25\%$)	8.49 s ($\sigma = 0.36$ *s*)
9	32	32	92.46 % ($\sigma = 9.58\%$)	9.04 s ($\sigma = 0.30$ *s*)
10	64	32	93.61 % ($\sigma = 6.97\%$)	9.73 s ($\sigma = 0.35$ *s*)
11	128	32	90.78 % ($\sigma = 10.46\%$)	11.21 s ($\sigma = 0.49$ *s*)

References

[1] E.J. Terrell, W.M. Needelman, J.P. Kyle, Wind turbine tribology, in: M. Nosonovsky, B. Bhushan (Eds.), Green Tribology, Green Energy and Technology, Springer Berlin Heidelberg, 2012, pp. 483–530.
[2] J. Lacey, An overview of bearing vibration analysis, *Maintenance & asset management* 23 (6) (2008) 32–42.
[3] W.A. Smith, R.B. Randall, Rolling element bearing diagnostics using the case western reserve university data: *a benchmark study*, *Mechanical Systems and Signal Processing* 64-65 (2015) 100–131.
[4] P. Boškoski, J. Petrovčič, B. Musizza, Dani Juričić, Detection of lubrication starved bearings in electrical motors by means of vibration analysis, *Tribology International* 43 (9) (2010) 1683–1692.
[5] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
[6] M. Monte, F. Verbelen, B. Vervisch, The use of orbitals and full spectra to identify misalignment, in: IMAC XXXII, Proceedings, Springer International Publishing, 2014, pp. 215–222.
[7] B.P. Graney, K. Starry, Rolling element bearing analysis, *Materials Evaluation* 70 (1) (2012) 78–85.
[8] R. Heng, M. Nor, Statistical analysis of sound and vibration signals for monitoring rolling element bearing condition, *Applied Acoustics* 53 (1–3) (1998) 211–226.
[9] H. Ohta, Y. Nakajima, S. Kato, H. Tajimi, Vibration and acoustic emission measurements evaluating the separation of the balls and raceways with lubricating film in a linear bearing under grease lubrication, Journal of Tribology 135 (4).
[10] A. Purarjomandlangrudi, A.H. Ghapanchi, M. Esmalifalak, A data mining approach for fault diagnosis: *an application of anomaly detection algorithm*, *Measurement* 55 (2014) 343–352.
[11] O. Geramifard, J. Xu, C. Pang, J. Zhou, X. Li, Data-driven approaches in health condition monitoring – a comparative study, in: 8th IEEE International Conference on Control and Automation (ICCA), 2010, pp. 1618–1622.
[12] Q. Miao, D. Wang, M. Pecht, A probabilistic description scheme for rotating machinery health evaluation, *Journal of Mechanical Science and Technology* 24 (12) (2010) 2421–2430.
[13] A. Verma, Z. Zhang, A. Kusiak, Modeling and prediction of gearbox faults with data-mining algorithms, *Journal of Solar Energy Engineering* 135 (3) (2013) 1–11.
[14] D. Kateris, D. Moshou, X.-E. Pantazi, I. Gravalos, N. Sawalhi, S. Loutridis, A machine learning approach for the condition monitoring of rotating machinery, *Journal of Mechanical Science and Technology* 28 (1) (2014) 61–71.
[15] J.B. Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, F. Fnaiech, Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals, *Applied Acoustics* 89 (2015) 16–27.
[16] P. Kankar, S.C. Sharma, S. Harsha, Fault diagnosis of ball bearings using machine learning methods, *Expert Systems with Applications* 38 (3) (2011) 1876–1886.
[17] J. Fitch, The hidden dangers of lubricant starvation, Online, ⟨http://www.machinerylubrication.com/Read/29040/lubricant-starvation-dangers⟩2012.
[18] M. Monte, F. Verbelen, B. Vervisch, Detection of coupling misalignment by extended orbits, in: Experimental Techniques, Rotating Machinery, and Acoustics, Volume 8, Springer, 2015, pp. 243–250.
[19] M. Aharon, M. Elad, A. Bruckstein, K -svd: *an algorithm for designing overcomplete dictionaries for sparse representation*, *Signal Processing, IEEE Transactions on* 54 (11) (2006) 4311–4322.
[20] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 689–696.
[21] S. Bahmani, P. Boufounos, B. Raj, Greedy sparsity-constrained optimization, in: Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on, 2011, pp. 1148–1152.
[22] S. Beygi, M. Kafashan, H.R. Bahrami, D.H. Mugler, The iterative shrinkage method for impulsive noise reduction from images, *Measurement Science and Technology* 23 (11) (2012) 1–7.
[23] C. Wang, M. Gan, C. Zhu, Fault feature extraction of rolling element bearings based on wavelet packet transform and sparse representation theory, *Journal of Intelligent Manufacturing* (2015) 1–15.
[24] S. Deng, B. Jing, S. Sheng, Y. Huang, H. Zhou, Impulse feature extraction method for machinery fault detection using fusion sparse coding and online dictionary learning, *Chinese Journal of Aeronautics* 28 (2) (2015) 488–498.
[25] N. Verma, V. Gupta, M. Sharma, R. Sevakula, Intelligent condition based monitoring of rotating machines using sparse auto-encoders, in: IEEE Conference on Prognostics and Health Management (PHM), 2013, pp. 1–7.
[26] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86 (11) (1998) 2278–2324.
[27] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.
[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions 2014. URL ⟨http://arxiv.org/abs/1409.4842⟩.
[29] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition, *IEEE/ACM Transactions on Audio, Speech and Language Processing* 22 (10) (2014) 1533–1545.
[30] S. Dieleman, K.W. Willett, J. Dambre, Rotation-invariant convolutional neural networks for galaxy morphology prediction, *Monthly Notices of the Royal Astronomical Society* 450 (2) (2015) 1441–1459.
[31] Z. Chen, C. Li, R.-V. Sanchez, Gearbox fault identification and classification with convolutional neural networks, Shock and Vibration, 2015 (2015) 10.
[32] SKF, Spherical roller bearings, Online October 2009.

[33] Schaeffler, Fag split plummer block housings of series snv, Online 2015.
[34] O. Janssens, R. Schulz, V. Slavkovikj, K. Stockman, M. Loccufier, R.V. de Walle, S.V. Hoecke, Thermal image based fault diagnosis for rotating machinery, *Infrared Physics & Technology* 73 (2015) 78–87.
[35] L. Breiman, Bagging predictors, *Machine Learning* 24 (2) (1996) 123–140.
[36] V. Slavkovikj, S. Verstockt, W. De Neve, S. Van Hoecke, R. Van de Walle, Hyperspectral image classification with convolutional neural networks, in: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, MM'15, 2015, pp. 1159–1162.
[37] I. Sutskever, J. Martens, G.E. Dahl, G.E. Hinton, On the importance of initialization and momentum in deep learning, in: Proceedings of the 30th International Conference on Machine Learning (ICML-13), Vol. 28, 2013, pp. 1139–1147.
[38] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Computer VisionECCV 2014, Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 818–833.