

PROCEEDINGS OF SPIE

SPIDigitalLibrary.org/conference-proceedings-of-spie

A deep learning approach for fault detection and RUL estimation in bearings

Bono, F., Cinquemani, S., Chatterton, S., Pennacchi, P.

F. M. Bono, S. Cinquemani, S. Chatterton, P. Pennacchi, "A deep learning approach for fault detection and RUL estimation in bearings," Proc. SPIE 12049, NDE 4.0, Predictive Maintenance, and Communication and Energy Systems in a Globally Networked World, 1204908 (18 April 2022); doi: 10.1117/12.2607084

SPIE.

Event: SPIE Smart Structures + Nondestructive Evaluation, 2022, Long Beach, California, United States

A deep learning approach for fault detection and RUL estimation in bearings

Bono F. M.^a, Cinquemani S.^a, Chatterton S.^a, and Pennacchi P.^a

^aPolitecnico di Milano, Milano, Italy

ABSTRACT

This paper presents a deep learning approach for detecting early fault in bearings. The identification of bearings defects represents an important problem in the field of rotating machines. Sudden failures may occur, leading to breakdown of the machinery. For this reason, the prediction of possible faults has become a major issue in the study of bearing elements. Different fault diagnosis techniques have been developed during the years based on aggregated parameters (i.e. features) that are computed starting from time domain, frequency domain or time frequency domain analysis, relying on prior knowledge about signal processing. These approaches present major limitations, that can be overcome by adopting a convolutional LSTM (long short-term memory) neural network model. In this case, a more complex architecture is built, and the algorithm can identify effective features from accelerometer signal, that could not be considered by a manual computation approach. The algorithm has been applied on data obtained from a complex test rig to assess bearings failure on high speed trains. The outcome of this work indicates that the adopted approach leads to satisfactory performances.

Keywords: Condition monitoring, deep learning, bearing, LSTM network, RUL estimation

1. INTRODUCTION

In the context of Industry 4.0, the continuous monitoring of automated machineries' health status is a necessity. In the last few decades, condition based maintenance (CBM) has become a trending topic in the industrial context and it is considered as one of the most innovative solutions for predicting incipient failures in machinery.¹ The implementation of CBM strategy can help to understand when maintenance should be performed and offers key-benefits over the traditional strategies, such as run-to-failure or preventive maintenance.² In this field, particular attention is given to bearings since half of the machine breakdown is caused by the failure of these components.^{3,4} Verma and Subramanian⁵ present a cost benefit analysis in rotating machineries which proves that the maximum revenue generation is reached when CBM is adopted over other maintenance strategies.

The widespread application of bearings in industry requires efficient methods for monitoring their health status. Different physical quantities can be considered to develop a condition monitoring system. Most used are: vibration, temperature or acoustic emission. Among these, vibration measurements are the most common.⁶ Starting from the vibration signature of a rolling element bearing, traditional approaches based on prior knowledge about signal processing can be adopted to identify defects.⁷ A set of aggregated parameters (i.e. features), characterized by a physical meaning, can highlight an incipient fault. They are manually extracted in time, frequency, or time-frequency domains.⁸⁻¹⁰ These features are kept under control by threshold-based algorithm, or more in general, a shallow learning model is commonly implemented to assess bearing health status.¹¹ All these methods rely on signal processing theory, but during this "extraction" of features, relevant information contained in the signal could remain unexploited. These limitations can be overcome with the adoption of deep learning (DL) algorithms. In this case, the algorithm can identify effective features from vibration signal, that could not be considered by an approach based on a manual computation of features.¹²

In the last years, Artificial Intelligence (AI) has led to a breakthrough in an enormous variety of human fields, and the automation industry makes no exception.¹³ In industrial environments, AI can turn traditional machineries into smart systems, able to provide real-time insights on production, alongside information on the condition of structural components. In the field of rotating machineries, DL algorithms have been widely applied for intelligent bearing fault diagnosis^{14,15} and researchers achieved prominent results in this research field.¹⁶

Further author information: Francesco Morgan Bono: E-mail: francescomorgan.bono@polimi.it

NDE 4.0, Predictive Maintenance, and Communication and Energy Systems in a Globally Networked World,
edited by Norbert G. Meyendorf, Saman Farhangdoust, Christopher Niezrecki, Proc. of SPIE
Vol. 12049, 1204908 · © 2022 SPIE · 0277-786X · doi: 10.1117/12.2607084

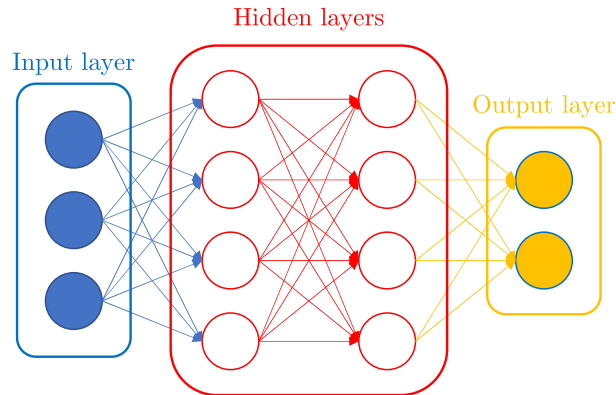


Figure 1. Graphical representation of a basic artificial neural network

Alongside fault detection and diagnostics, another topic of interest is prognostic; it concerns the algorithms for predicting the time progression of a specific failure mode from its incipience to the time of component failure.¹⁷ Remaining useful life (RUL) estimation of a degrading system is the main prognostic task for industry application.¹⁸

This paper proposes an approach for fault detection and RUL estimation in bearings based on convolutional/recurrent neural networks that processes signal acquired by means of an accelerometer and a tachometer. The networks are trained on data acquired from a test rig and validated.

The paper is structured as follow: section 2 describes deep learning fundamentals, section 3 presents the approach in details, section 4 describes the experimental tests for the development of the dataset, section 5 presents the obtained results, section 6 resumes and discusses both methodology and results.

2. DEEP LEARNING FUNDAMENTALS

Artificial neural networks (ANNs) are relatively crude electronic models based on the neural structure of the brain. They are composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.¹⁹ The neurons are grouped in layers, that are stacked together to realize the network architecture, as shown in figure 1. The first layer of an ANN is the **input layer**. This layer has the shape of the objects entering the network (in figure 1 it is composed by three neurons, therefore the input objects are 3x1 vectors). The last layer of an ANN takes the name of **output layer**. This layer's shape strictly depends on what the network has to predict (In figure 1 the 2 neurons in the output suggest that the output object is a 2x1 vector, hence every input object yields 2 values as output). All the layers included between the input and the output layers are called **hidden layers**. Generally speaking, there's no limit to the number of layers and the number of neurons in each layer, and the optimal values for these two numbers cannot be assessed a priori. However, the more the number of layers and neurons, the more weights must be tuned and then, the more the computational burden.

2.1 Layers

Layer is a general term that applies to a collection of neurons operating together at a specific depth within a neural network. Different layers perform different transformations on their inputs, and some of them are better suited for some tasks than others. The five most useful for applications similar to the one here described are briefly explained, to help readers in constructing their own neural network.

2.1.1 Dense layer

Dense layer is the simplest and straight-forward type that can be used. All the neurons in a dense layer are connected to all the neurons in the previous layer. Every connection is characterized by a weight, which multiplies

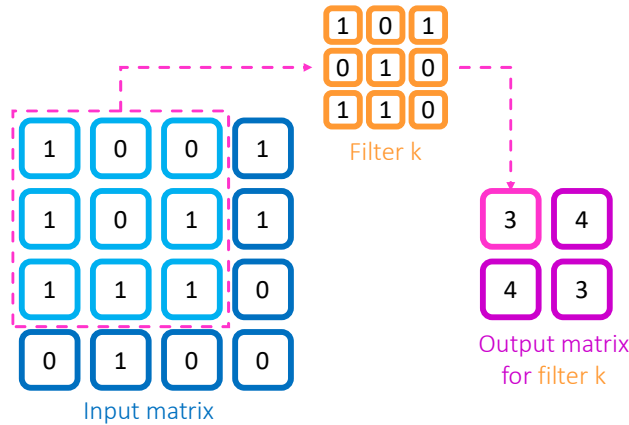


Figure 2. Example of convolution operation

the input value. Dense layer defines a bias b and an activation function f . If x is the input tensor, z is the output tensor and W is the weights tensor, the mathematical equation of a dense layer is:

$$z = f(\tilde{z}) = f(Wx + b) \quad (1)$$

Where \tilde{z} is called weighted sum of the input. The input tensor is a vector $x = [x_1, x_2, \dots, x_n]^T$ (with n representing the number of input values), and the weight tensor is a matrix in the following form:

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{m1} & w_{m2} & \dots & w_{mn} \end{bmatrix}^T$$

Where m is the number of neurons in the layer.

2.1.2 Convolutional layer

The convolutional layer allow to "scan" the input tensor with the purpose of finding local patterns. The convolution operation is made by a linear operator used for feature extraction, where an array of numbers, called filter, "slides" over the input tensor. In particular, the operation consists of an element-wise product between the filter and the input tensor, followed by a sum operation to obtain the output value in the corresponding position of the output tensor. If the input is a $N \times N$ tensor and the filter is a $K \times K$ tensor, the output generated by one filter is a $(N-K+1) \times (N-K+1)$ matrix. In figure 2, a numerical example of the convolution operation performed by one filter is reported: the input is a 4×4 tensor, the filter is a 2×2 and the output is a 3×3 matrix.

If G is the number of filters in a convolutional layer, the output of that layer will count G matrices.

2.1.3 Pooling layer

A limit of convolutional layers is that they greatly increase the number of parameters in the output tensors, compared to the input ones; especially when many filters are involved, the magnitude of tensors grows exponentially. For this reason, a pooling layer usually succeeds a convolutional one. Its purpose is to sub-sample the feature map by retaining only the most attractive information extracted by the convolutional layer. There are many possible pooling functions, but in this work MaxPooling one is adopted, which takes only the max value out of a predefined sub-matrix (an example is shown in figure 3).

2.1.4 GRU and LSTM layers

GRU (gated recurrent unit) and LSTM layer belongs to the family of recurrent neural network (RNN), which are dedicated to the analysis of time sequences. RNN cells differ from the regular neurons since they have an internal hidden state and thus can remember information from the past. They integrate the hidden state h_t ,

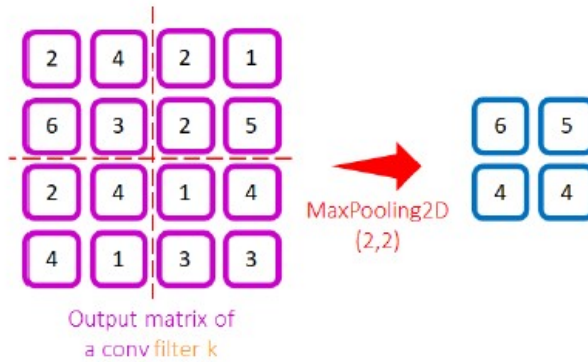


Figure 3. Example of maxpooling operation

which results updated at any time step, and it is expressed as function of the current input X_t and the previous hidden state h_{t-1} :

$$h_t = \Phi(h_{t-1}, X_t) \quad (2)$$

where Φ is a function. The complete set of equations for a RNN cell is not reported since its comprehension, in contrast with the other explained layers, is not so straightforward and requires a detailed description. LSTM layer could handle the problem of long-term dependencies,²⁰ but it requires a high computational effort that ends up in slow training phase of the network. For this reason Cho et Al.²¹ proposed a simplified version of LSTM called GRU, that is characterized by less complex set of equations, ending in lower computational cost with respect to LSTM.

2.1.5 Dropout layer

A common problem in the development of an ANN is the overfitting: it happens when a model learns from particular random feature in the training data so that it is able to perfectly manage that set; but these learned concepts may not apply to new data, leading to poor performances. Dropout is a form of regularization, i.e. an approach that makes the network more robust in training phase, by forcing the network to learn general and recurrent patterns. During training, if a tensor passes through a dropout layer, some of its values are randomly dropped, according to the dropout probability, i.e. the fraction of input's elements whose value is set to zero (figure 4). During test, no values becomes zero, but the output is scaled by a factor equal to dropout probability. Sometimes the values are adjusted by the same fraction only in training, to leave the test and prediction phases untouched. Some guidelines to manage the dropout layer can be found in.²²

2.2 Activation function

The concept of activation function is represented in equation 1: it defines the transformation applied to the weighted sum of the input to obtain the output in a layer of the network. Different types of function can be used

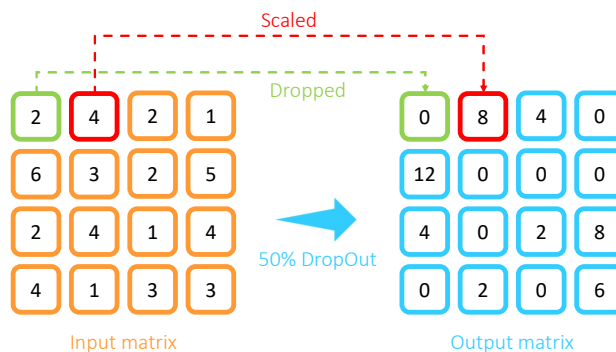


Figure 4. Example of 50% dropout; half of the values are set to zero, the values are doubled to prevent scaling in test.

based on the layer considered and the objective of the network. The activation functions adopted in this work are:

- **Rectified Linear Unit (ReLU)**: it is mainly used for hidden layers in fully connected or CNN neural network. This function is defined as:

$$f = \max(0, \tilde{z}) \quad (3)$$

where \tilde{z} represented the weighted sum of the input presented in equation 1.

- **Sigmoid activation**: it is mainly used for hidden layers in a RNN and for output layer of a binary classification network. This function is defined as:

$$f = \frac{1}{1 + e^{-\tilde{z}}} \quad (4)$$

- **Hyperbolic tangent (tanh)**: it is used for hidden layers in a RNN. it is defined as:

$$f = \frac{e^{\tilde{z}} - e^{-\tilde{z}}}{e^{\tilde{z}} + e^{-\tilde{z}}} \quad (5)$$

- **Linear**: it is used for output layer in a regression problem. it is defined as:

$$f = \tilde{z} \quad (6)$$

2.3 Loss function

A neural network automatically fits its parameters (called weights) during the training phase. The fitting procedure is based on an optimization problem: the objective (or loss) function to minimize is defined as the distance between the output label of the network and the true labels. Different types of loss functions can be chosen depending on problem's nature. The loss functions adopted in this work are:

- **Binary cross-entropy function**: it usually adopted in classification problems. it is defined as follow:

$$L(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (7)$$

where i and N represent the sample's index and the total number of samples respectively, y_i is the label (that assumes 0 or 1 value) and $p(y_i)$ is the predicted probability of belonging to class-1.

- **Mean Square error**: It is usually adopted for regression problems. It averages the squared difference between the predictions and the expected results. it is defined as follow:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (8)$$

i , N and represent the same meaning assumed in equation 7, instead y_i and \hat{y}_i are the actual and predicted values, respectively.

3. PROBLEM STATEMENT AND PROPOSED METHODOLOGY

This paper is intended to provide a general deep learning procedure to realize a CBM strategy for roller bearings. Symptoms of a defective bearing can be found in vibration measurement²³ and can be enhanced by dedicated processing algorithms.²⁴ Real-time monitoring of this physical quantity enables the possibility to detect an early-stage failure, to estimate the residual life of the element and to plan maintenance before the breakdown occurs.

In this paper, the approach is applied on a test rig developed for the diagnostic of bearing of a traction system

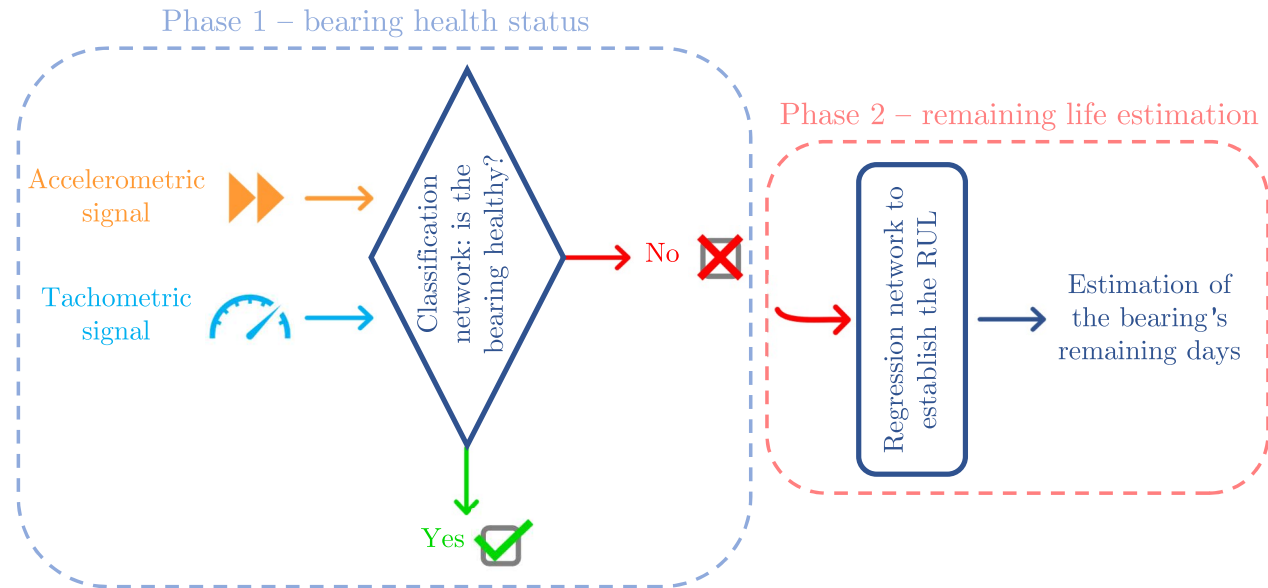


Figure 5. Visual representation of the proposed method to assess bearing health condition, and eventually the remaining life.

for high speed trains, as described in details in section 4. Accelerometric and tachometric signal are acquired for monitoring the bearing. The latter has faced a damage caused by misalignment, which represents the principal failure mode for this particular application. Misalignment happens in bearing due to shaft deformation under load, deflection of the shaft or asymmetric loads; it determines a considerable effect on the performance of the system, until reaching the failure condition.²⁵

The proposed approach relies on a successive characterization of the bearing health status, by analyzing the two signals in two steps, as shown in figure 5:

1. In the first step, the signals enter a neural network (called **classification network**) which classifies the bearing as healthy or not. The network has been trained to recognize whether the bearing is properly working, or a misalignment occurred.
2. If the first network has identified a faulty condition, a second neural network (called **regression network**) estimates the RUL. The architecture used here is close to the configuration of the first network, with the exception that it performs a regression, whereas the first one operated a classification.

The bearing RUL is predicted through a parameter called *score* (s) defined as:

$$s = \frac{day_{failure} - day_i}{day_{failure} - day_{first}} \quad (9)$$

where $day_{failure}$ represents the day in which the bearing has faced the failure and day_{first} the day in which the fault occurs. In this way, the score s assumes values from 100 (in the first day) to 0 (in the failure day). This is done to let the network to be more generic in prediction when a different bearing is considered for the evaluation of the RUL.

3.1 Data preprocessing

As explained in section 3, all the data are considered for the development of the classification network, instead only faulty data are considered for the regression network.

First of all, the dataset is randomly divided into three subsets by following the train/ validation/ test split technique:

- **Train set:** it is composed by 70% of the dataset. It is used to train the network and fit the parameters of the model.
- **Validation set:** it is composed by 15% of the dataset. During the training phase of the model, this set is used to compute the loss (called validation loss) to understand if the model is facing problems like overfitting or underfitting.
- **Test set:** it is composed by 15% of the dataset. Once the training phase is done, this set is used to provide an unbiased evaluation of the model performances.

The following procedure is applied to all the three subsets.

As mentioned in section 4, accelerometric and tachometric signal are available for each acquisition. To feed a ML algorithm, the input data should be brought in a tensor arrangement. For this purpose, a "sliding window" with a width of 2500 samples (0.25s) is considered to split the signal in segments. The following procedure is adopted:

- The two signals of a single acquisition are considered.
- The sliding window saves the segment to a 2500x2 matrix.
- The sliding window "slides" for 2500 samples, it acquires new samples and save them to a new matrix.
- When the end of the signal for an acquisition is reached, a new one is considered and the process restarts from the beginning.

For each acquisition, a number of 20 matrices are created. Then, all the matrices are stacked together in a $N_i \times 2500 \times 2$ tensor, being N_i the number of acquisitions for a particular subset.

The last step concerns the data scaling by means of a standardization technique. For each signal, all the samples in the training set are concatenated in a single vector, and then mean and standard deviation are computed. Hence, all the samples belonging to the three partitions can be normalized by computing the Z-score:

$$Z = \frac{x - \mu}{\sigma} \quad (10)$$

In this way, the distribution of samples will be centered in zero with a standard deviation of 1.

On the other hand, the output data are generated during the above described procedure for the input data. Different outputs are computed for the two neural networks considered in this work:

- **Classification network:** bearing health status is known for each acquisition. It is assigned 0 value for nominal condition and 1 value for faulty condition. No scaling is required for this output.
- **Regression network:** the parameter s (explained in section 3 is computed for each signal considered as faulty by applying equation 9. In this case, also the output s is scaled by applying a standardization technique.

The preprocessing pipeline for input and output data is resumed in figure 6.

3.2 Models architecture

The structure of the neural networks adopted in this paper is extremely similar to the one proposed and consolidate for speech recognition and text analysis.²⁶ The neural network's structure selected is made of a combination of convolutional neural networks (CNN) and recurrent cells (like LSTM or GRU). This solution has proved to be more effective than the direct application of recurrent neural networks.²⁷ In particular, recurrent layers are usually extreme prone to noise, while convolutional layers have denoising properties, so that the unique local patterns in signals have less influence on the final results.²⁸

Since the models deal with one-dimensional signals, almost all of the layers are used in a 1D version. Furthermore,

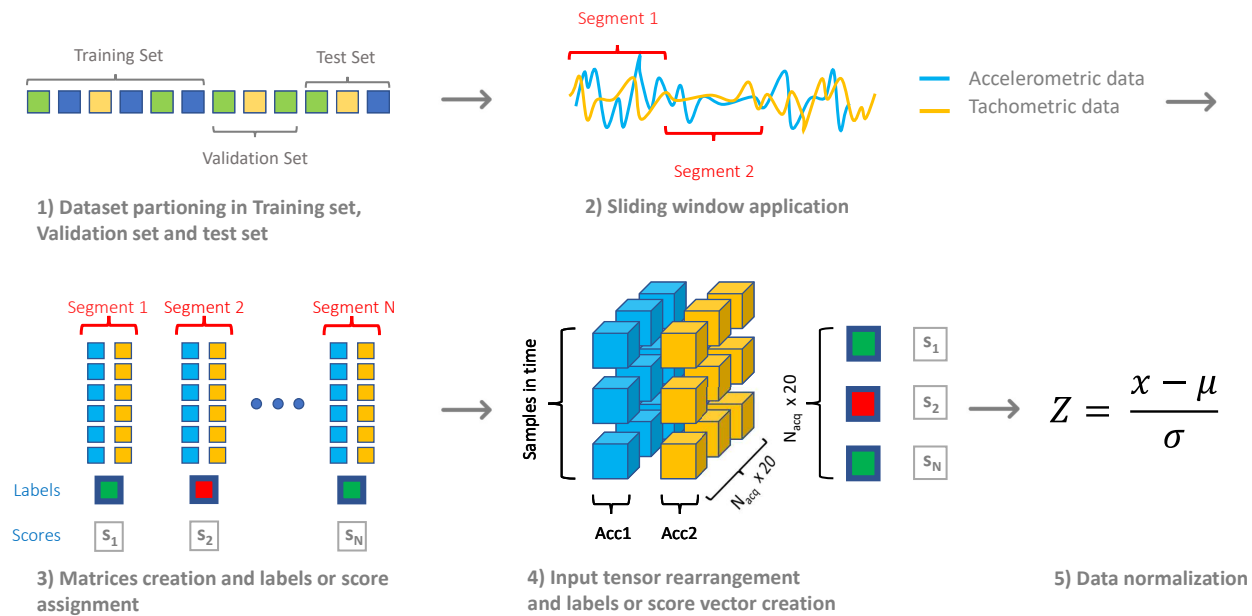


Figure 6. Preprocessing pipeline

instead of 1D convolutional layers, the models contain 1D separable convolutional layers. This layer accomplishes the convolution operation separately on the two channels, i.e. accelerometric and tachometric signal, by creating and applying dedicated filters for each of them. In this way, a depth-wise convolution separately on the two channels is performed, followed by a point-wise convolution that mixes the two. The models architecture are described in further detail below. It is important to underline that the proposed architectures are the final outcome of an iterative process of hyperparameter optimization, where the number and types of layers are optimized, as well as the learning rate, the momentum, the batch size and the number of training epochs.²⁹

The following notation is introduced to describe the adopted layer: $C(n, m)$ represents an 1D convolutional layer with n filters of kernel size $m \times 1$, $P(n)$ represents a maxpooling layer with a sub-sampling factor of n , $D(n)$

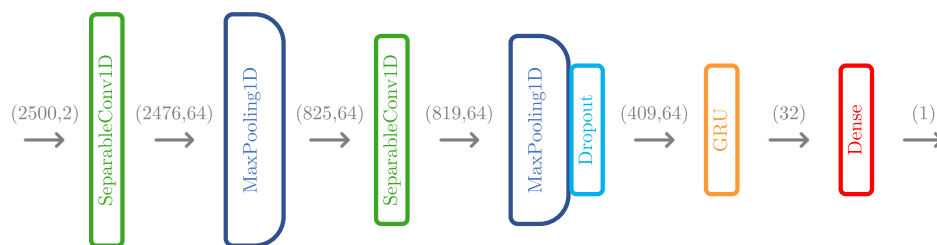


Figure 7. Structure of the classification network

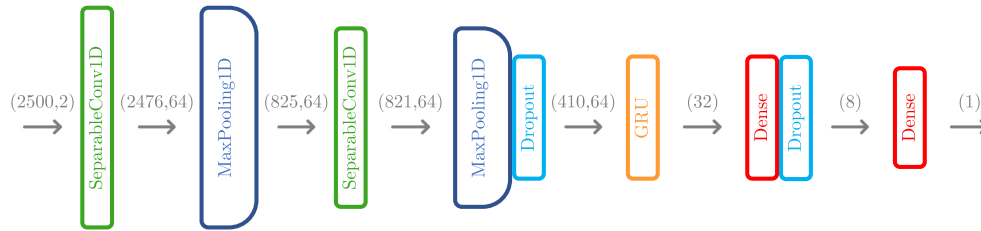


Figure 8. Structure of the regression network

represents a dense layer with n neurons, $DR(p)$ represents a dropout layer with a fraction of the input unit to drop equal to p and $G(n)$ represents a GRU layer with n units that determines the dimensionality of the output space.

3.2.1 Classification network

The structure of the classification network is represented in figure 7. The input layer receives 2500x2 tensor, while the output is a single number, that can assume 0 or 1 values. The architecture can be summarized as: $C(64,25)$, $P(3)$, $C(64,7)$, $P(2)$, $DR(0.1)$, $G(32)$, $D(1)$. Convolutional and GRU layers are operated with ReLU activation function, instead dense layer presents the sigmoid activation function. Binary cross-entropy is considered as loss function.

3.2.2 Regression network

The structure of the classification network is represented in figure 8. The input layer receives 2500x2 tensor, while the output is a single number, i.e. the score s . The architecture can be summarized as: $C(64,25)$, $P(3)$, $C(64,5)$, $P(2)$, $DR(0.1)$, $G(32)$, $D(8)$, $DR(0.1)$, $D(1)$. Convolutional and GRU layers are operated with ReLU activation function, as well as the first dense layer, instead the second one presents the linear activation function. In this network, MSE is considered as loss function.



Figure 9. Damaged bearing.³⁰

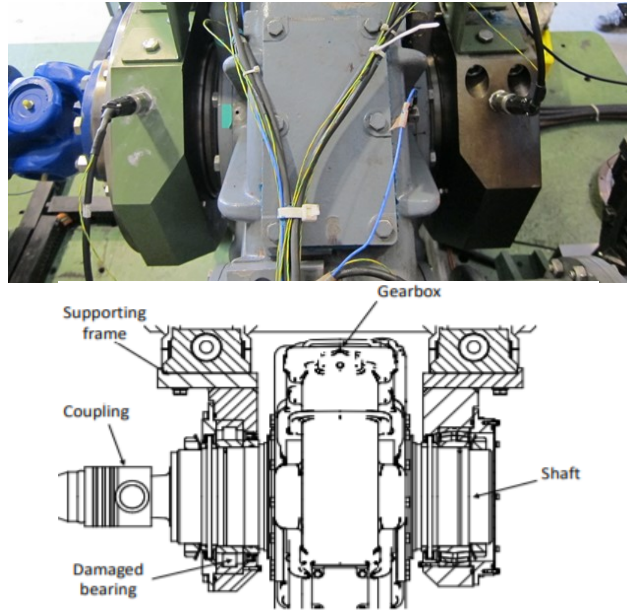


Figure 10. Test rig overview and position of the damaged bearing.³¹

4. EXPERIMENTAL SETUP

The proposed methodology is applied to a real case consisting of a test rig developed for the diagnostic of rolling element bearing of a traction system for high speed trains (shown in figure 10). Detailed information about the test rig and the experiment can be found in articles.³⁰⁻³³

A failure occurred on the cylindrical roller bearing (FAG NU1040M1) of the frame that support the output shaft of the gearbox under test (Fig. 3). Even if the damaged bearing is an auxiliary component of the test rig, it was monitored by an accelerometer and, after a period of about 39 millions of bearing cycles, high level of vibration has been detected by the sensor. A visual analysis was performed to understand if possible damages had been occurred; the inspection revealed a damage in the inner ring of the bearing with brinelling marks as shown in figure 9, caused by an angular misalignment of the shaft with respect to the bearing house due to looseness of the frame support. In this operating condition, a local bearing overload occurred leading to the failure. After the failure, the bearing was replaced and the acquisition continued for 37 days.

The bearing ran in this faulty condition for 8 hours a day for 91 days at constant speed of about 16.17 Hz and constant radial load until the failure. Data have been acquired with a sampling rate equal to 20 kHz, and a five-second vibration signal was collected every hour. Considering the effect of thermal expansion during the run-up from stand-still condition and the variation of environment condition, such as the room temperature, only four records from the second hour to the fifth hour are selected every day. Furthermore, a tachometric signal was available for each acquisition. The available dataset is labelled as follow for training, validating and testing the algorithms:

- **Nominal data:** these data are related to the first bearing before the misalignment had been happened and to the new bearing (that replaced the first one once the failure had been occurred) for 37 days.
- **Faulty data:** these data are related to the first bearing from the time the misalignment had been happened until the failure (91 days).

5. RESULTS

Both the two networks are built on *Tensorflow* platform that is a widespread open source library for developing and training ML and DL models.³⁴ The metrics adopted to evaluate the performances of the networks are the

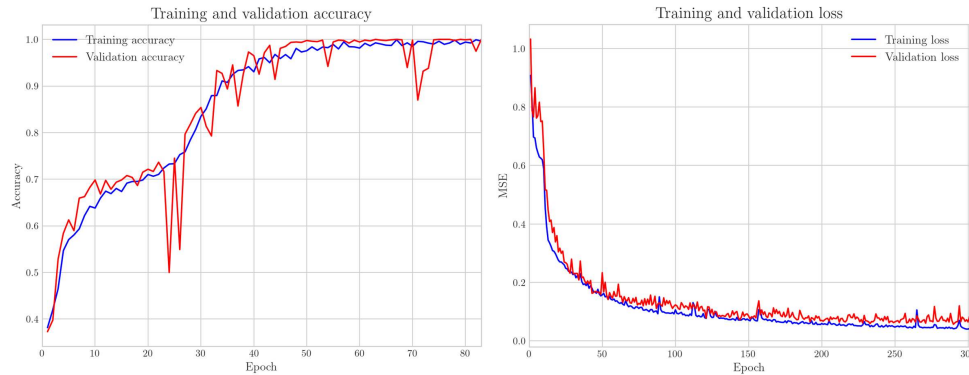


Figure 11. Training histories of the two networks; on the left for the classification network and on the right for the regression network.

accuracy and the MSE respectively for the classification and the regression problems. The two networks are trained for a maximum of 200 epochs with the implementation of a callbacks method to regulate the fitting method in order to prevent overfitting.³⁵ In particular, an "Earlystopping criteria" is adopted to stop the training process when some condition are met, i.e. the process is stopped when the performance of the model in validation does not improve for 15 epochs.

In figure 11 the training histories of the networks are showed; the classification network achieved an accuracy of 99.97% in training and 99.69 % in validation after 83 epochs of training. This model is then evaluated on the test set, reaching an accuracy of 99.72%. It is possible to assess that the first network is able to reliably predict the presence of a misalignment.

For the regression network, since the output was normalized as explained in section 3, the network predicts the normalized value of the score s ; the true values of this parameter can be restored by inverting the equation 10 for each sample. Then the MAE of the scores can be computed to evaluate the performances by adopting the following equation:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (11)$$

The resulting values for three partitions are respectively 1.309, 1.362, 1.356. Considering that score s ranges from 0 to 100, it is possible to affirm that the network is extremely robust in predicting the RUL.

6. CONCLUSIONS

In this paper, a deep learning based approach for bearing fault detection and RUL estimation is presented. The method relies on two different neural networks that processes accelerometric and tachometric signal produced by the bearing; the first network is responsible for the determination of the bearing health status and, once a damage is detected, the second one estimates the RUL of the rolling element. The presented method is validated with data collected from a test rig developed for the diagnostic of bearing of a traction system for high speed trains. The results have shown that the presented method is effective for the implementation of a CBM maintenance approach. The capability of a convolutional/recurrent neural network of automatically extracting features from raw data let to obtain information about fault severity. For this reason, this approach can be adopted in a wide range of application in modern industry, where bearing diagnosis represents a primary need.

REFERENCES

- [1] Jauregui Correa, J. C. A. and Lozano Guzman, A. A., "Chapter eight - condition monitoring," in [*Mechanical Vibrations and Condition Monitoring*], Jauregui Correa, J. C. A. and Lozano Guzman, A. A., eds., 147–168, Academic Press (2020).
- [2] Bogue, R., "Sensors for condition monitoring: A review of technologies and applications," *Sensor Review* **33** (09 2013).

- [3] Zhou, W., Habetler, T. G., and Harley, R. G., "Bearing condition monitoring methods for electric machines: A general review," in [2007 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives], 3–6 (2007).
- [4] Nandi, S., Toliyat, H., and Li, X., "Condition monitoring and fault diagnosis of electrical motors—a review," *IEEE Transactions on Energy Conversion* **20**(4), 719–729 (2005).
- [5] Verma, N. K. and Subramanian, T., "Cost benefit analysis of intelligent condition based maintenance of rotating machinery," in [2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)], 1390–1394 (2012).
- [6] Tandon, N., Yadava, G., and Ramakrishna, K., "A comparison of some condition monitoring techniques for the detection of defect in induction motor ball bearings," *Mechanical Systems and Signal Processing* **21**(1), 244–256 (2007).
- [7] Tandon, N. and Choudhury, A., "A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings," *Tribology International* **32**(8), 469–480 (1999).
- [8] McInerny, S. and Dai, Y., "Basic vibration signal processing for bearing fault detection," *IEEE Transactions on Education* **46**(1), 149–156 (2003).
- [9] Malla, C. and Panigrahi, I., "Review of condition monitoring of rolling element bearing using vibration analysis and other techniques," *Journal of Vibration Engineering & Technologies* **7** (05 2019).
- [10] Patil, M. S., Mathew, J., and RajendraKumar, P. K., "Bearing Signature Analysis as a Medium for Fault Detection: A Review," *Journal of Tribology* **130** (12 2007). 014001.
- [11] Wu, D., Wang, J., Wang, H., Liu, H., Lai, L., He, T., and Xie, T., "An automatic bearing fault diagnosis method based on characteristics frequency ratio," *Sensors* **20**(5) (2020).
- [12] Zhong, G., Ling, X., and Wang, L.-N., "From shallow feature learning to deep learning: Benefits from the width and depth of deep architectures," *WIREs Data Mining and Knowledge Discovery* **9**(1), e1255 (2019).
- [13] Lee, J., Davari, H., Singh, J., and Pandhare, V., "Industrial artificial intelligence for industry 4.0-based manufacturing systems," *Manufacturing Letters* **18**, 20–23 (2018).
- [14] Hoang, D.-T. and Kang, H.-J., "A survey on deep learning based bearing fault diagnosis," *Neurocomputing* **335**, 327–335 (2019).
- [15] Zhang, S., Zhang, S., Wang, B., and Habetler, T. G., "Deep learning algorithms for bearing fault diagnostics—a comprehensive review," *IEEE Access* **8**, 29857–29881 (2020).
- [16] Verstraete, D., Ferrada, A., Droguett, E., Meruane, V., and Modarres, M., "Deep learning enabled fault diagnosis using time-frequency image analysis of rolling element bearings," *Shock and Vibration* **2017**, 1–17 (10 2017).
- [17] Sikorska, J., Hodkiewicz, M., and Ma, L., "Prognostic modelling options for remaining useful life estimation by industry," *Mechanical Systems and Signal Processing* **25**(5), 1803–1836 (2011).
- [18] Yu, W., Kim, I. Y., and Mechefske, C., "An improved similarity-based prognostic algorithm for rul estimation using an rnn autoencoder scheme," *Reliability Engineering & System Safety* **199**, 106926 (03 2020).
- [19] Schmidhuber, J., "Deep learning in neural networks: An overview," *CoRR abs/1404.7828* (2014).
- [20] Yu, Y., Si, X., Hu, C., and Zhang, J., "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation* **31**(7), 1235–1270 (2019).
- [21] Cho, K., van Merriënboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR abs/1406.1078* (2014).
- [22] Srivastava, N., "Improving neural networks with dropout," *University of Toronto* **182**(566), 7 (2013).
- [23] Saruhan, H., Sandemir, S., Çiçek, A., and Uygur, I., "Vibration analysis of rolling element bearings defects," *Journal of Applied Research and Technology* **12**(3), 384–395 (2014).
- [24] McInerny, S. and Dai, Y., "Basic vibration signal processing for bearing fault detection," *Education, IEEE Transactions on* **46**, 149 – 156 (03 2003).
- [25] He, Z.-P., Zhang, J.-H., Xie, W.-S., Li, Z.-Y., and Zhang, G.-C., "Misalignment analysis of journal bearing influenced by asymmetric deflection, based on a simple stepped shaft model," *Journal of Zhejiang University: Science A* **13**(9), 647–664 (2012). cited By 26.

- [26] Chien, J.-T., “Deep bayesian natural language processing,” in [*Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*], 25–30 (2019).
- [27] Yin, W., Kann, K., Yu, M., and Schütze, H., “Comparative study of cnn and rnn for natural language processing,” (2017).
- [28] Sainath, T. N., Vinyals, O., Senior, A. W., and Sak, H., “Convolutional, long short-term memory, fully connected deep neural networks,” *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* , 4580–4584 (2015).
- [29] Aszemi, N. M. and Dominic, P., “Hyperparameter optimization in convolutional neural network using genetic algorithms,” *Int. J. Adv. Comput. Sci. Appl* **10**(6), 269–278 (2019).
- [30] Chatterton, S., Borghesani, P., Pennacchi, P., and Vania, A., “Optimal frequency band selection for the square envelope spectrum in the diagnostics of rolling element bearings,” *Proceedings of the ASME Design Engineering Technical Conference* **8** (08 2014).
- [31] Xu, L., Pennacchi, P., and Chatterton, S., “A new method for the estimation of bearing health state and remaining useful life based on the moving average cross-correlation of power spectral density,” *Mechanical Systems and Signal Processing* **139**, 106617 (2020).
- [32] Pennacchi, P., Chatterton, S., Vania, A., Ricci, R., and Borghesani, P., “Experimental evidences in bearing diagnostics for traction system of high speed trains,” *Chemical Engineering Transactions* **33**, 739–744 (Jul. 2013).
- [33] Pennacchi, P., Bruni, S., Chatterton, S., Borghesani, P., Ricci, R., Marinis, D., Didonato, A., and Unger-Weber, F., “A test rig for the condition-based maintenance application on the traction chain of very high speed trains,” *Proceedings. of the 9th World Congress on Railway Research - WCRR* , 1–11 (01 2011).
- [34] Goldsborough, P., “A tour of tensorflow,” (2016).
- [35] Caruana, R., Lawrence, S., and Giles, L., “Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping,” in [*Proceedings of the 13th International Conference on Neural Information Processing Systems*], *NIPS’00*, 381–387, MIT Press, Cambridge, MA, USA (2000).