Motor Bearing Failure Identification Using Multiple Long Short-Term Memory Training Strategies

Youcef Atmani[1], Ammar Mesloub[2], Said Rechak[3]

[1]*National High School of Advanced Technologies - ENSTA, LTI Laboratory, Dergana, Algiers, Algeria*
[2]*Polytechnic Military School -EMP, Bordj El Bahri, Algiers, Algeria*
[3]*National Polytechnic School -ENP, El Harrach, Algiers, Algeria*

Abstract

In the context of condition-based maintenance of rotating machines in manufacturing systems, the early diagnosis of possible faults related to rolling elements of the bearing is mainly based on techniques from artificial intelligence, namely, Machine Learning (ML) and Deep Learning (DL). Approaches based on using Deep Learning methods have been the most coveted in recent years. Among a variety of models, the type of architecture known as Long-Short-Term Memory (LSTM) of Recurrent Neural Network (RNN) has both the ability to capture long-term dependencies and to adapt to sequential data modeling. It is therefore able to work on data without any preprocessing. This paper studies using four types of LSTM networks to diagnose bearing faults in a classification approach. It aims to intervene on both the input parameters and the network architecture, to achieve high performance. The proposed method is carried out in two different ways. In the first case, the data inputs are raw frames of vibration signals. However, in the second case, the network inputs are pre-computed time-frequency features. The results clearly showed that LSTMs are more accurate with the latter.

Keywords: *Maintenance, Bearing, Fault, Deep Learning, Diagnosis, LSTM.*

1. Introduction

Guidance and power transmission allow bearings to play a leading role in the operation of rotating machines. These important functions involve the use of all possible means to protect this organ. Indeed, its failure can not only alter its environment but also, in many cases, cause the shutdown of an entire production line. Thus, it is possible to maximize the lifespan of these sensitive organs by applying an adequate predictive maintenance strategy. In this context, Scheffer and Girdhar advocated the use of a technique based primarily on vibration analysis [1]. The diagnostics identified suspected defects inside the rolling organs of the bearing. This result leads to the implementation of maintenance monitoring based on the condition rather than systematic maintenance, which is increasingly neglected for particularly obvious economic reasons. According to O'Donnell [2], statistical studies have shown that 30-40% of rotating machine failures are caused by bearing damage. Randall and Antoni [3] confirms the concern of many specialists about the difficulty of pronouncing on the condition of the bearing. This pretext justifies the development of multiple diagnostic methods. Moreover, some studies have been clearly identified in a review by Henriquez et al. [4].

The use of methods based on Machine Learning (ML) has inspired an era of artificial intelligence. Thus, diagnosis is now approached through a classification problem. It covers the methods and concepts that are automatically extracted from the data, prediction, and decision-making models. In this context and for many years, a variety of algorithms based on the extraction of characteristics from vibration or acoustic signals, as in the

Corresponding author: *Youcef Atmani (youcef.atmani@ensta.edu.dz)*
Received: *30 June 2024*; Revised: *3 October 2024*; Accepted: *8 October 2024*; Published: *11 October 2024*

work of Choudhury and Tandon [5], have been developed for the diagnosis of bearing defects. These methods include the use of Artificial Neural Networks (ANN) by Dubey and Agrawal [6]. In addition, Rojas and Nandi [7] and Benkedjouh et al. [8] used a Support Vector Machine (SVM). Even more, the use of Hidden Markov Models (HMM) by Purushotham et al. in [9], K Nearest Neighbors (KNN) by Tian et al. in [10], Gaussian Mixting Models (GMM) by Atmani et al. in [11], and others like in [12] and [13]. However, with big data, these classical methods have become increasingly unsuitable for this increase in size. In parallel, Deep Learning (DL) has overcome this problem. Furthermore, the larger the data size, the more effective the DL method will be.

Since 2006 to date, a wide variety of deep learning architectures applied to Machine Health Monitoring (MHM) reviewed by Zhao et al. in [14]. With regard to the special use of these methods in bearing fault diagnosis, Zhang et al. summarized them well in [15] and the works carried out in the field. We first find the convolutional neural networks (CNN)-based methods applied by Zhang et al. in [16], Guo et al. in [17] and Chen et al. in [18], on different data sources and at variable rotation speeds, to obtain high performance in bearing fault diagnosis. Methods based on Auto-Encoders (AE) are also in a good position. Their application to bearing diagnoses can be cited in the work of Mao et al. [19]. Other variants of AEs, as in [20] and [21], have been used and have all proven a certain form of performance. Another class named Deep Belief Network (DBN) was also applied. The three-layer DBN used by Chen and Li [22] provides a good example. Staying in continuity, another category called Generative Adversarial Network (GAN) can be mentioned here. Liu et al. proposed an application case in [23] for unsupervised diagnosis of bearing faults. Finally, particular interest has been given to Recurrent Neural Network (RNN). The latter is well-suited for capturing and modelling sequential or time-series data. Applications of RNN were limited owing to the gradient vanishing/exploding issue, until the advent of its version, referred to as Long Short-Term Memory (LSTM), developed by Hochreiter and Schmidhuber in 1997 [24], to overcome this problem. It has a large capacity to memorize and model long-term dependencies in time-series data and textual analyses.

Up to date, recurrent neural network have been successful in many areas, such as voice and handwriting recognition, video analysis, and more. It is worth mentioning the work of Abed et al. in [25], who reported the application of RNNs in bearing fault diagnosis. The data input to the classifier were features extracted using a Discrete Wavelet Transform (DWT). The experimental results have proven the ability of the algorithm to accurately detect and classify bearing defects. Guo et al. [26], propose another method named RNN Health Indicator (RNN-HI) in which, time-frequency features obtained from wind turbines bearings datasets are used. The goal is to predict the Remaining Useful Life (RUL) of bearings using LSTM cells and RNN layers. According to the authors, the proposed RNN-HI method performs better than the Self-Organized Map (SOM) method. Pan et al. [27] presented a bearing fault classification method based on a network architecture composed of a 1-D CNN and LSTM. The network input data were raw signals without preprocessing. A high accuracy was obtained for the different configurations tested. Finally, a recent work of Zhao and Shao [28] proposed an adaptive method for bearing fault diagnosis based on a Deep Gated Recurrent Unit (DGRU). The procedure for learning features obtained from vibration signals is based on an Artificial Fish Swarm Algorithm (AFSA) and an Extreme Learning Machine (ELM). The results were powerful and robust.

While the adaptation of LSTMs neural networks to sequential data and time series has been proven, the literature remains poor in works dedicated to their application to the diagnosis of defects in rotating machinery in general, and rolling bearing elements in particular. Difficulties related to the choice of hyper parameters, overfitting problem, and laborious calculation times contributed to these limitations. However, current interest is growing, and our contribution in this paper attempts to bring solutions to the problems posed. We will investigate the performance of LSTMs with single-layer and deeper layers. Thus, obtaining good results with modest computing resources will greatly open the field to industrial implications in the context of conditional maintenance (in both online and offline modes). The rest of the writing starts in Section 1, with theoretical elements of the proposed method, which includes a classification scheme, used LSTMs architectures, and time-frequency features. Section 2 presents an application case study. The results and discussions are the concerns of Section 3. Finally, the paper ends with a conclusion and perspective.

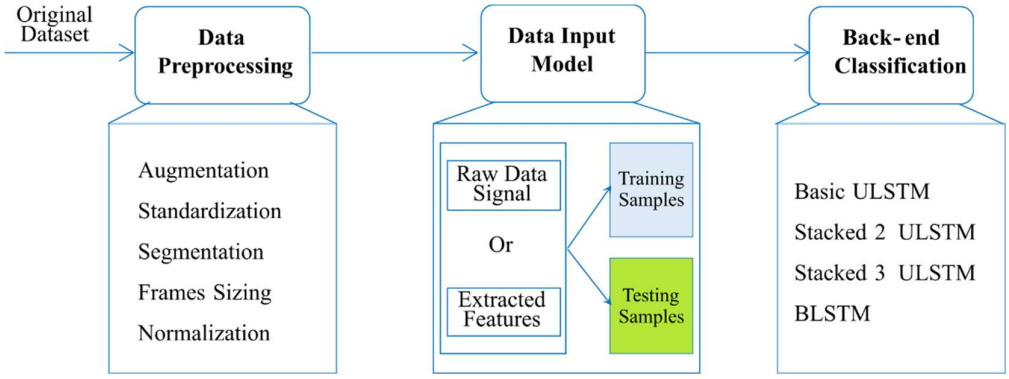

Figure 1. Workflow diagram of the proposed LSTM neural network for classification.

2. Proposed Method

This study aims to explore the possibilities of using LSTM networks in the diagnosis of bearing defects using a multitude of adopted strategies. Thus, two distinct input data formats were applied to four models with different LSTM network architectures: three variants of a unidirectional network with one, two, and three hidden layers (basic ULSTM, stacked 2 ULSTM, and stacked 3 ULSTM, respectively) and a bidirectional network variant (BLSTM) with a single hidden layer. Eight possible scenarios were occasionally tested to achieve the best performance in terms of precision and calculation time.

2.1. Diagram and strategies

The progress of each elaborate classification process goes through three stages, well described by the diagram in Figure 1. The first stage includes preprocessing applied to the original signals to homogenize them in shapes and sizes. For half of the scenarios (4 in number), the raw signal frames obtained are grouped to be introduced to the input layer of the classification network. However, for the other half of the remaining scenarios, extracting spectral features from the obtained signal frames is necessary, followed by normalization of the data before their introduction into the network's input layer. The second step is to randomly distribute training and test data groups. The third and final step consists of choosing the architectural model of the classifier.

2.2. Behavior of a LSTM network cell

Recurrent neural networks are well suited to handling temporal sequence learning problems. They use the backpropagation algorithm in their training process. However, it has been proved in certain works, such as those of Bengio et al. [29], that traditional RNNs cannot capture long-term dependencies due to the strong limitation of the gradient vanishing problem during the backpropagation period. This is why Long-Short-Term Memory (LSTM) proposed in 1997 by Sepp Hochreiter and Juergen Schmidhuber [24], came to overcome this issue.

The concept behind the widespread use of LSTM is that a few gates and a memory cell control the flow of information and can apprehend at each time step, more precise long-term dependencies. The block structure of LSTM consists of one memory cell and three gates. The input gate is used to control the value entered into the LSTM and transfer it to the memory cell. The forgetting gate allows control of the memory cell to retain or delete information. The output gate monitors the output of the current state so that only the desired value will be kept in the memory cell in the next step.

The operation mode of the LSTM can be governed by the following mathematical updated equations:

$$i_t = \sigma(W_i x_t + V_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma\left(W_f x_t + V_f h_{t-1} + b_f\right) \quad (2)$$

$$o_t = \sigma(W_o x_t + V_o h_{t-1} + b_o) \quad (3)$$

$$\check{C}_t = tanh(W_c x_t + V_c h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \check{C}_t \quad (5)$$

$$h_t = o_t \odot tanh(c_t) \quad (6)$$

Where W and V are weight matrices and b are bias vectors. These parameters of model are determined during the training phase and shared at every time step. ⊙ denotes the Hadamard product. σ_x and $tanh_x$ are activation functions, which are defined as follows:

$$\sigma_x = \frac{1}{1+e^{-x}} \quad (7)$$

$$tanh_x = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (8)$$

2.3. Classification based on raw signals

This first strategy concerns four types of LSTM network architecture models designed to classify bearing conditions. It has been used both in unidirectional and in bidirectional modes. A unidirectional case is developed under a basic form (single hidden layer) and under a stacked form (with two and three hidden layers).

The architecture of the different unidirectional LSTM networks (ULSTM) obeys the same functional classification diagram presented in the first block of Figure 2. The data flow starts from the input layer, undergoes modifications through one or more hidden layers formed by LSTM blocks with N time steps, and ends at the end on an output layer (Softmax) of the classifier. In the case of Basic ULSTM, there is only one single hidden layer (j=1) for N time steps. However, for used stacked ULSTMs, two and three hidden layers are stacked forming the network (1<j≤M=4) for N time steps too.

However, the BLSTM network architecture processes sequence data in both forward and backward directions, using two separate hidden layers. This concept taken from the bidirectional RNN of Schuster and Paliwal [30], allows the network to learn long-term dependencies, of the complete time series and at each time step. The structure of a basic BLSTM network (single or non-stacked) is shown in the second block of Figure 2. The output data of the front and back layers are estimated from the standard LSTM updated equations (1) to (6).

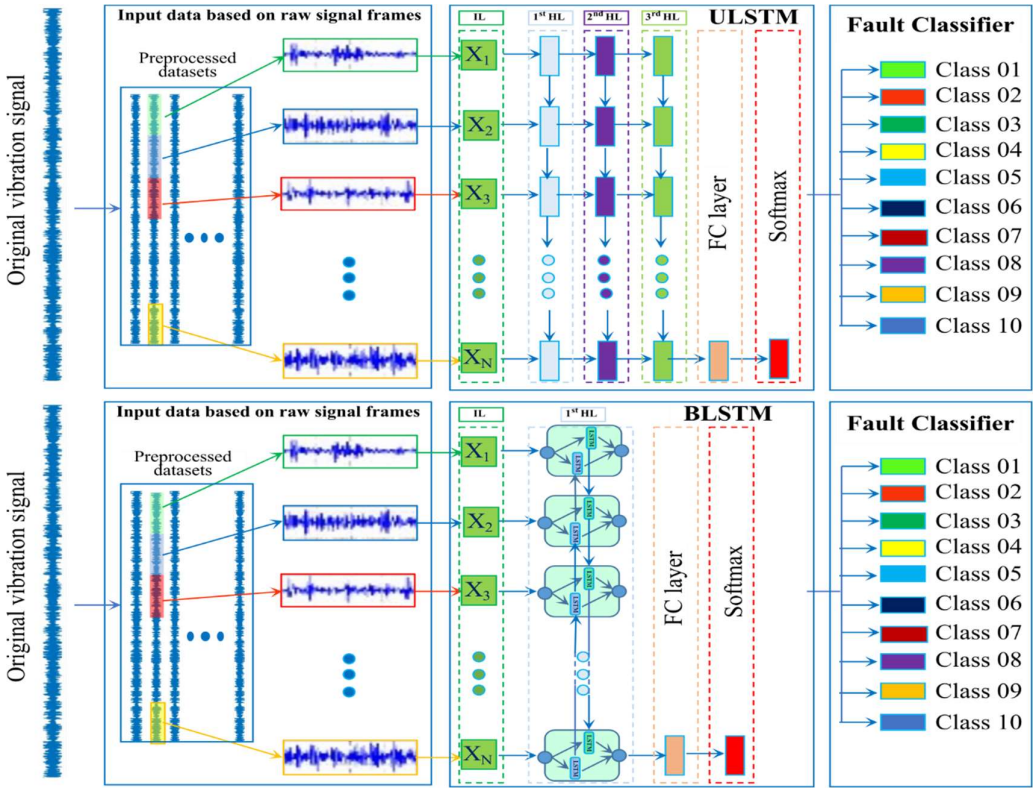

Figure 2. Block diagram of global classification scheme based on raw signals.

The general equations which induce the update for any Mth arbitrary layer, of unidirectional stacked form are listed below:

$$i_t^M = \sigma\left(W_i^M h_t^{M-1} + V_i^M h_{t-1}^M + b_i^M\right) \quad (9)$$

$$f_t^M = \sigma\left(W_f^M h_t^{M-1} + V_f^M h_{t-1}^M + b_f^M\right) \quad (10)$$

$$o_t^M = \sigma(W_o^M h_t^{M-1} + V_o^M h_{t-1}^M + b_o^M) \quad (11)$$

$$\check{C}_t^M = tanh(W_c^M h_t^{M-1} + V_c^M h_{t-1}^M + b_c^M) \quad (12)$$

$$c_t^M = f_t^M \odot c_{t-1}^M + i_t^M \odot \check{C}_t^M \quad (13)$$

$$h_t^M = o_t^M \odot tanh(c_t^M) \quad (14)$$

It should be noted that in this hierarchical configuration, each intermediate hidden layer of the LSTM network constitutes an input for the next one and an output for the previous one. Raw vibration signals (or possibly the manually extracted features) form the data of the first input layer of the network. The output data from the last hidden LSTM layer is sent to a fully connected layer and then to the Softmax layer for classification.

2.4. Classification based on spectral features

This second classification strategy is applied to the four LSTM network architecture models intended to detect rolling conditions. Once the original signals have been preprocessed and segmented, during this phase we proceed to the manual extraction of two types of spectral features, namely the instantaneous frequency (IF) and the spectral entropy (SE). As the difference between their values is considerable, a normalization process is necessary, before sending them to the input of each architectural model. Figure 3 clearly shows the overall classification scheme.

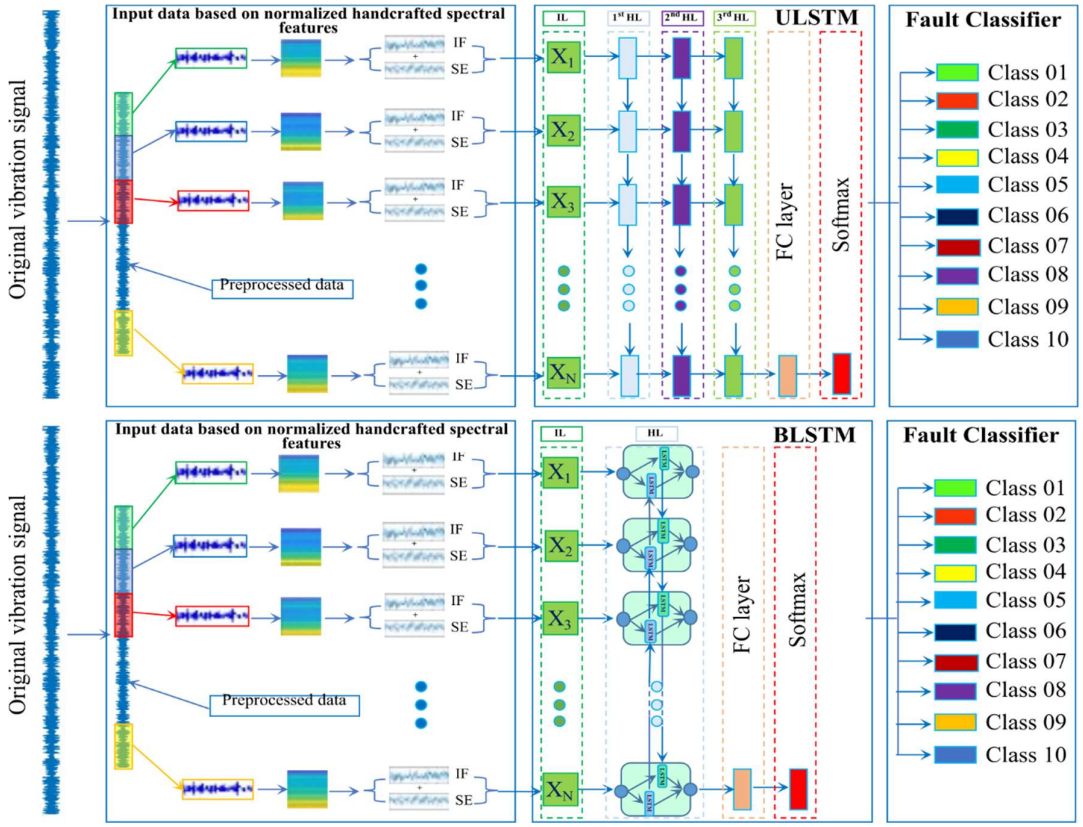

Figure 3. Block diagram of global classification scheme based on spectral features.

2.1.1. Estimate instantaneous frequency

The Instantaneous Frequency (IF) of a non-stationary signal is a parameter linked to the average of the frequencies present in the signal and which varies over time [31] and [32]. It can be estimated as the first conditional spectral moment of the time-frequency distribution of the input signal, or from the derivative of the phase shift of the input analytical signal. In this work, the first approach was used and it is done as follows:

First step: the power spectrum (spectrogram) S(t, f) of the input signal is computed and used as a time-frequency distribution

Second step: Estimation of the instantaneous frequency using the expression:

$$f_{inst}(t) = \frac{\int_0^\infty f\ S(t,f)\ df}{\int_0^\infty S(t,f)\ df} \quad (15)$$

2.1.2. Spectral Entropy

Spectral entropy (SE) is a measure of the normalized spectral power distribution of a signal, treated as a probability distribution and from which the Shannon entropy from information theory is calculated. This property gives it wide use in many fields and is useful for feature extraction in fault detection and diagnosis [33]. Its equations follow both from those of the power spectrum and the probability distribution of a signal.

Thus, for a signal x(n), the power spectrum is:

$$S(m) = |X(m)|^2 \quad (16)$$

Where $X(m)$ is the discrete Fourier transform of $x(n)$. The probability distribution $P(m)$ becomes:

$$P(m) = \frac{S(m)}{\sum_i S(i)} \quad (17)$$

The spectral entropy *H* is expressed by:

$$H = -\sum_{m=1}^{N} P(m)\ log_2\ P(m) \quad (18)$$

By normalizing:

$$H_n = \frac{-\sum_{m=1}^{N} P(m)\ log_2\ P(m)}{log_2\ N} \quad (19)$$

Where *N* represents the total number of frequency points. In the denominator, $\log_2 N$ is the maximum spectral entropy of white noise, uniformly distributed in the frequency domain.

If a time-frequency power spectrogram $S(t,f)$ is known, then the probability distribution becomes:

$$P(m) = \frac{\sum_t S(t,m)}{\sum_f \sum_t S(t,f)} \quad (20)$$

Spectral entropy is still given by Eq. (18).

To calculate instantaneous spectral entropy from a time-frequency power spectrogram $S(t,f)$, the probability distribution at time *t* is:

$$P(t,m) = \frac{S(t,m)}{\sum_f S(t,f)} \quad (21)$$

Thus, the spectral entropy at time *t* is given by the expression:

$$H(t) = -\sum_{m=1}^{N} P(t,m)\ log_2\ P(t,m) \quad (22)$$

3. Application Case

To study its effectiveness, the proposed approach is applied on a bearing dataset obtained from the Case Western Reserve University (CWRU) Bearing Data Center. This choice is motivated by its wide use in the literature for the validation of new algorithms. To date, it has become a reference in the field and hundreds of articles based on its data have been published. This is why our results can be reproducible and therefore verifiable.

2.5. Data description

The CWRU data collection contains normal (healthy) and faulty bearings vibration signals. The faulty ones have defects located at the ball, inner race, and outer race. For each type, there are three fault diameters, 0.18 mm, 0.36 mm, and 0.53 mm, respectively. Thus, there are 10 fault conditions (classes), to be considered in this dataset, as given in Table 1. The raw signals are recorded in Matlab files format (xxx.mat) in which names are defined by three digits. The selected ones for our work have a sampling frequency of 48 kHz and are classified according to four load conditions (0, 1, 2, and 3). Each file contains two signals issued from the Drive End (DE) and Fan End (FE) bearings.

Table 1. The 48 kHz selected vibration signals files.

Health condition	Classes (Labels)	Fault diameter [mm]	Motor load (HP) Vs. File names			
			0	1	2	3
Inner race fault	IR07	0.18	109	110	111	112
	IR14	0.36	173	175	176	177
	IR21	0.53	213	214	215	217
Outer race fault Centred	OR07	0.18	135	136	137	138
	OR14	0.36	201	202	203	204
	OR21	0.53	238	239	240	241
Ball fault	BA07	0.18	122	123	124	125
	BA14	0.36	189	190	191	192
	BA21	0.53	226	227	228	229
Healthy (Normal)	HE00	/	097	098	099	100

The details of the experimental environment and the corresponding data are well described in [34].

The selected data set covers 40 scenarios relating to the ten fault conditions considered (10 classes described above). By adding the two signals relating to the measurement points (DE and FE of each scenario), we obtain 80 signals to be processed as described in the classification scheme.

2.6. Dataset organization

The implementation of the approach begins with the preparation of the data sets. In this step, we proceeded to oversample the data to guarantee the dimensional homogeneity of the classes. It is a form of data augmentation used in deep learning. Then, targets from each class of the obtained dataset are divided randomly into training and testing partitions (sets). During the training stage, the network divides the data into mini-batches. To make all signals have the same length in the same mini-batch it pads or truncates them. This padding or truncating operation is very sensitive and can affect network performance because it can misinterpret a signal based on the added or removed information.

4. Results and Discussion

Throughout this study, two criteria, ranked in order, are used for performance evaluation. The first is naturally the average classification rate (accuracy) deduced from the confusion matrices. The higher it is, the better the performance. The second criterion is the computational time. The lower it is, the better the performance. However, the latter is used only for information purposes, as a complementary argument for the optimal choice of classification hyperparameters. This restriction is mainly caused by its strong dependence on the hardware configuration used consisting of a single CPU: Intel Core i7-3632QM, 2.20 GHz, which may therefore limit the generalization of the results.

All the obtained results are grouped into two main categories. The first contains classifications based on data in the form of raw vibration signal frames, introduced at the input layer of the network. The second groups together classifications based on two time-frequency moment features manually extracted beforehand from signal frames and then introduced at the input layer of the network.

2.7. Classification with LSTM using raw signals

To reduce excessive calculation time, the basic ULSTM architecture was used to optimize the hyperparameters of the classification process. For the same reasons, the number of “Max Epochs” is limited to 5. Among the training options specified for the classifier is the “mini-batch size” set to 32, which instructs the network to examine 32 training signals at a time. The “initial learning rate” is set to 0.01 to promote an accelerated training process. The “gradient threshold” is taken equal to 1, for stability of the training process by preventing too large gradients. Then the choice was made on the adaptive momentum estimation (Adam) solver, because it is more suitable for recurrent neural networks

(LSTM) than the Stochastic Gradient Descent Momentum (SGDM) solver, as specified in [35]. For training and testing, there are the same number of signals in each class. However, this number depends on frame analysis length, as shown in Table 2. To have reliable experimental results, a random selection strategy is applied to the selection of samples (training and testing).

Before considering using each of the four LSTM network architectures, it is best to understand the decisions that must be made during its construction, in particular how to choose certain classification parameters. The latter must first be adjusted to specific values, to allow the classifier to achieve the right performance. However, in the literature, there are not yet clearly established rules or recommendations relating to the fixed choices of these values. Unfortunately, studies including even partial optimization of hyperparameters as in [27] are rare. Moreover, such optimizations will likely require a separate study. It therefore seems more interesting to first carry out a set of tests allowing certain parameters of the LSTM classifier to be optimized. Thus, a process of optimizing the frame length, the mini-batch size, and the initial learning rate was initiated.

Table 2. Distribution of training and testing data subsets according to frame length.

Frame length (Samples)	Training set (Frames)	Testing set (Frames)
2048	1481	165
1024	2963	329
512	5931	659
256	11864	1318
128	23733	2637
64	47473	5275

4.1.1. Frame length

A frame length or sequence length specifies the partitioning of the signal into smaller segments to prevent the machine memory from being overwhelmed by examining too much data at once. The tested range of vibration signal frames fed to the LSTMs as input sequences with varying dimensions, between 64 and 1024 samples. The classification results presented in Figure 4, show that the accuracy fluctuates between extreme values of 44% and 66%. This is due to the number of epochs initially set to only 5, to reduce the exorbitant computation times for epoch numbers greater than 100. However, it was found that the smaller the frame length, the longer the classification time. This counter-intuitive appearance is probably due to causes such as processing overhead, hidden state management, recursion loop, gradient update, etc. Fortunately, classification accuracy (performance) does not depend on computation time, which remains indicative. Thus, the value of 128 samples seems to be the best compromise, and will therefore be assigned to the sequence length parameter.

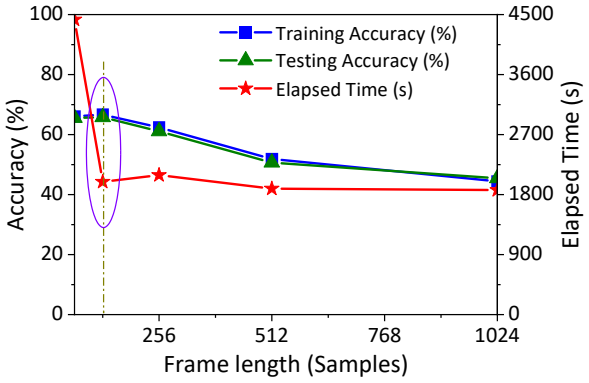

Figure 4. Effect of frame length on accuracy for Number of Epochs = 5.

4.1.2. Mini-batch size

The batch size is a hyperparameter of the model execution. It corresponds to the number of samples in a batch. It influences the speed of training and the efficiency of the model. Assigning a small value to this parameter can lead to a noisier estimation of the gradients. This can be beneficial to escape local minima, but can also make training slower. While a large value allows it to use resources efficiently, but can make training less dynamic and cause generalization problems. The values tested for this parameter are 8, 12, 16, 20, 24, 28, 32 and 64. The classification results obtained in Figure 5 show that the corresponding accuracy values are close. The value of 24 seems to be the best compromise because it corresponds to an acceptable number of iterations, and therefore to a relatively reduced classification time.

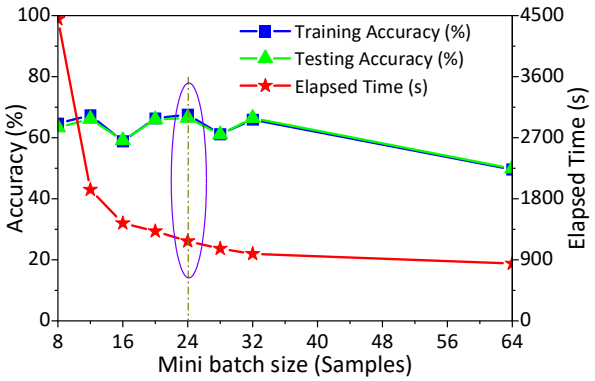

Figure 5. Accuracy versus mini batch size for Number of Epochs = 5.

4.1.3. Initial learning rate

It is used to compile the neural network model when a gradient descent optimizer trains it. It indicates the weights needed to be introduced in the reverse direction of the gradient, for a mini-batch. In our case, the Adam optimizer is used. The values tested for this hyperparameter are 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005 and 0.0001. According to the results presented in Figure 6, the value of 0.005 seems to be the best compromise. It is noted that the training diverges in the case of the extreme values of 0.1 and 0.0001. This is due either to significant changes in the weights or to the acceleration of the training, which causes an increase in the loss.

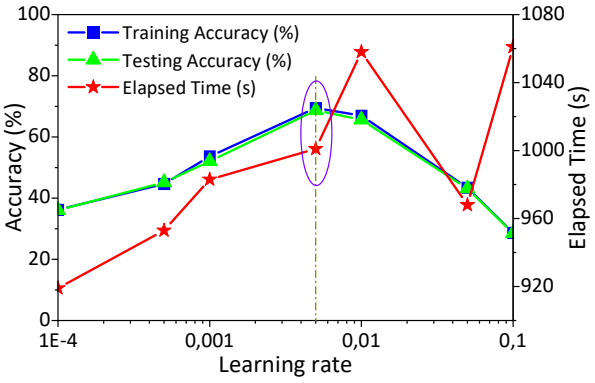

Figure 6. Impact of learning rate on performance.

4.1.4. Epochs number

The number of epochs is also one of the key parameters that affect the training process and the final performance of the neural network. It affects both the speed and the quality of the training process. It influences both the learning curve and the generalization ability. One epoch means one pass through all the training samples.

This important hyperparameter will not only be used like its predecessors for the evaluation of the performance of the basic ULSTM network but will also be extended to the rest of the architectures studied as shown in Figure 7, for comparison purposes. The values tested for the number of epochs are set to 5, 10, 15, 20, 30, 50, 75 and 100. This choice is imposed by an enormous calculation time necessary for values greater than 100.

In addition, as strongly indicated for deeper ULSTMs, a "Dropout" layer is recommended for each LSTM layer. This helps avoid overfitting, and will thus reduce the sensitivity to the specific weights of neurons. A value of 0.2 (20%) is usually used. The performances of the different network-tested architectures are represented by the evolution of the accuracy obtained from the confusion matrices as a function of the epoch's number.

The curves clearly show that increasing the number of epochs further improves the classification. Hence the highest precision achieved for all the used architectures is estimated at 80 ± 3 %. It corresponds to the number of epochs limit set at 100. To hope to further improve the score for this scenario (input in the form of raw signals and the same hardware configuration used), we must expect an exponential increase in time Calculation. Moreover, for information purposes, the basic ULSTM took approximately 8.5 hours for 100 epochs. While the BLSTM network exceeds 19 hours under the same conditions.

This seems logical since the BLSTM network processes data in both directions. The classification process therefore takes longer than that of the basic (unidirectional) ULSTM network.

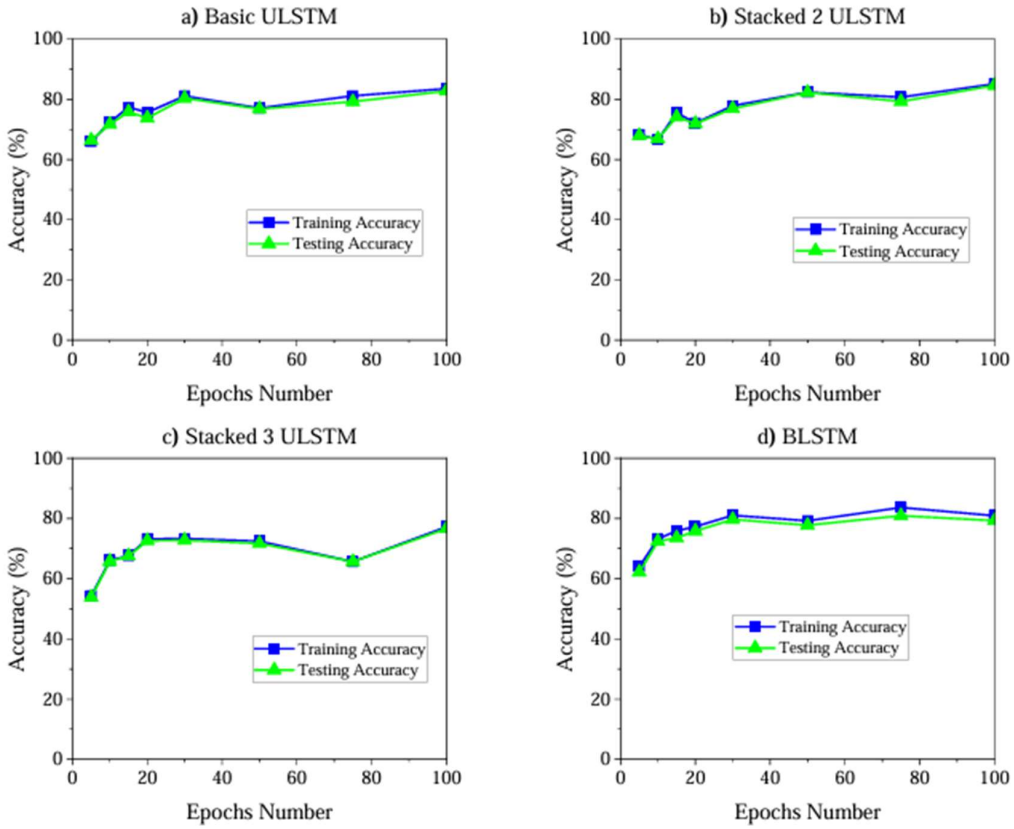

Figure 7. Accuracy versus Epochs Number for all tested LSTM architectures: a) Basic ULSTM, b) Stacked2 ULSTM, c) Stacked3 ULSTM and d) BLSTM.

2.8. LSTM classification based on time frequency moment features

The last and most important part of the experiment conducted in this paper was to use manual feature extraction hoping to improve the performance of the LSTM model in bearing fault detection and classification. The objective is to explore the effectiveness of the model and possible improvements in terms of precision and calculation time. To decide on the choice of features to extract, this study referred to the work of other researchers, notably those of Grzegorz Kłosowski & al. [36].

Therefore, feature extraction is based on the use of the short-term Fourier transform. It consists of converting the vibration signals relating to each class into spectrograms. Then, the time-frequency images are again converted into two vectors, namely instantaneous frequency and spectral entropy. The latter are subsequently intended for training LSTM networks. However, initially, they have very different averages. Furthermore, the instantaneous average frequency may be too high for the LSTM to learn effectively. It is therefore imperative to standardize the training and test sets, through a normalization called Z-Score.

4.1.5. Optimal sequence length

In this second part of the study, the classification is based on user-defined characteristics. Thus, the choice of the nature and number of attributes necessary for the input sequence potentially determines the result and the accuracy of the classification. If the choice of the duo composed of the instantaneous frequency and the spectral entropy initially fixed the nature of the characteristics. All that remains is to determine the optimal signal analysis frame length on which the feature extraction process will be applied. The analysis frame length values tested are 128, 256, 512, 1024, and 2048 samples, as shown in Figure 8. This choice is dictated in part, by the sizes of the segmented training and testing data sets.

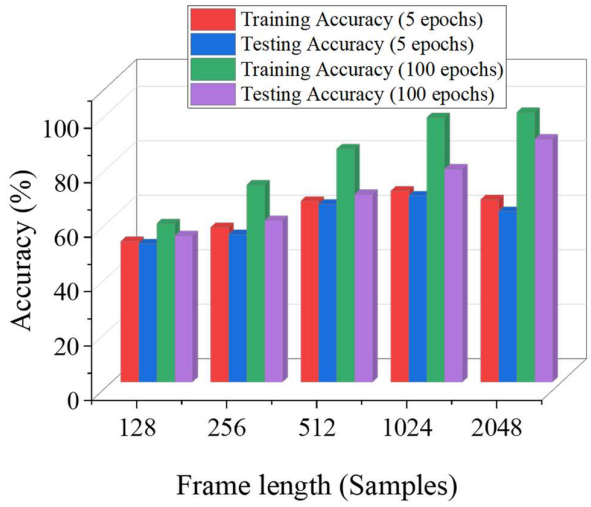

Figure 8. Accuracy versus sequence length for basic ULSTM.

The results show that for a value of the number of epochs set to 100, the classification rate relating to training is worth 99%. While that of the test, they correspond to 89%. This 10% gap clearly shows a data-overfitting problem. The strong point to remember is that the precision increases with the length of the analysis frame, and advantageously this improvement is also accompanied by a reduction in calculation time. In other words, the value of 2048 samples corresponds well to the optimal analysis frame length adopted.

4.1.6. Accuracy improvement

To supervise the classification process, it is wise to vary the number of epochs. Since the calculation times are short compared to those of the first part of the study, the maximum value of the number of epochs is doubled. Thus, the range of values tested for this parameter is 10, 20, 30, 50, 75, 100, 150, and 200, as shown in Figure 9.

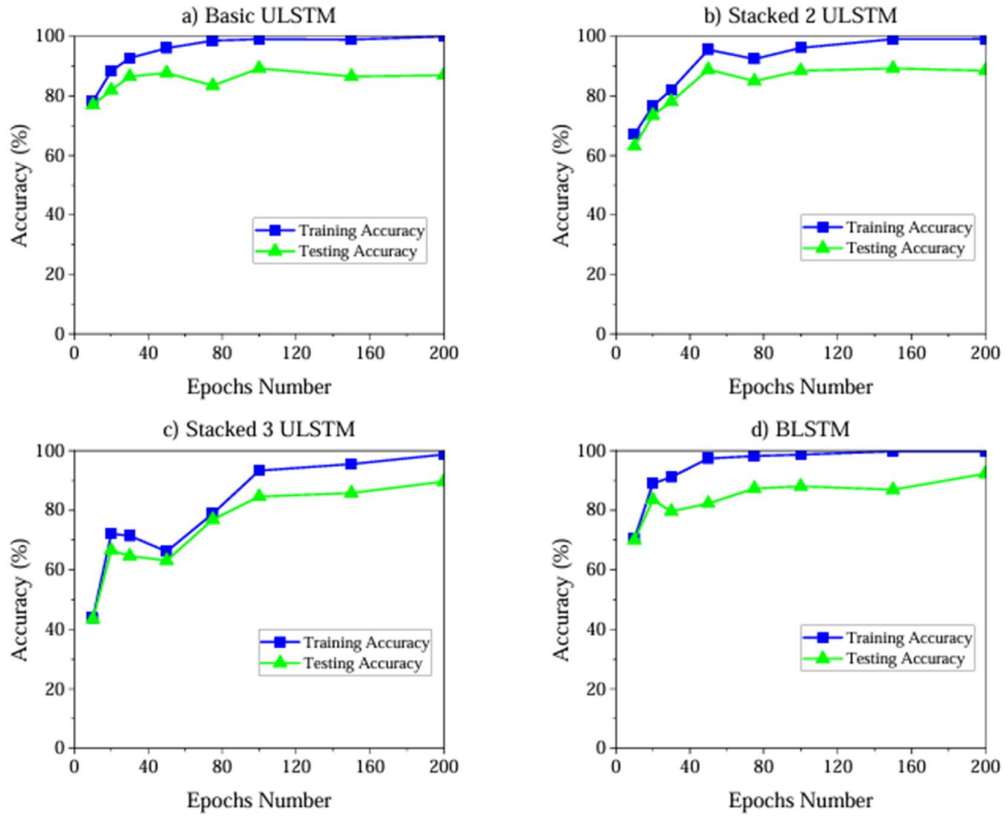

Figure 9. Accuracy versus Epochs Number for the four tested LSTM architectures: a) Basic ULSTM, b) Stacked2 ULSTM, c) Stacked3 ULSTM and d) BLSTM.

As expected, the results clearly show that increasing the number of epochs further improves the classification. However, we also know that this is not a norm. Indeed, the problem of data overfitting appeared from the value of 50. Beyond that, we note an increase in the training accuracy which reaches 100%, while that of the test remains fluctuating around 85%. In the same figure, we also notice that the calculation time increases in a practically linear manner. This suggests that to hope for an improvement, it is necessary to carry out more tests on other parameters or probably to first pre-process the signals from the database considered, which are known to be noisy.

2.9. Summary of key points

The results clearly show that the classification improves significantly with increasing number of epochs. However, this is not the rule. The problem of data overfitting can arise at any time. This is why it is necessary to carry out tests with maximum values and then search for the optimal values, which must remain constant for subsequent tests.

The two classification schemes under study were successfully applied to the four selected LSTM network architecture models. However, their modes of application have followed different directions. Indeed, in the first classification case, the frame of the raw vibration signal is not only long but also noisy. The result is less efficient classification accompanied by costly computational time. In the second scenario, the input sequence is relatively short and denoised thanks to the applied Fourier transformation processes. Thus, classification performance is maximum crowned by potentially reduced calculation time. While it exceeds 19 hours in the case of classification based on raw signals, it is of the order of 23 minutes for the worst case in the classification based on manual extraction of time-frequency features.

In the case of classification based on time-frequency moment characteristics, the appearance of the training progression and loss curves in Figure 10 confirms that the model completely converges in only about 50 epochs, which is rather fast for an LSTM and the results can be considered impressive. In conclusion, it can be said that the LSTM network is considered effective for the detection and classification of anomalies in data from vibration signals.

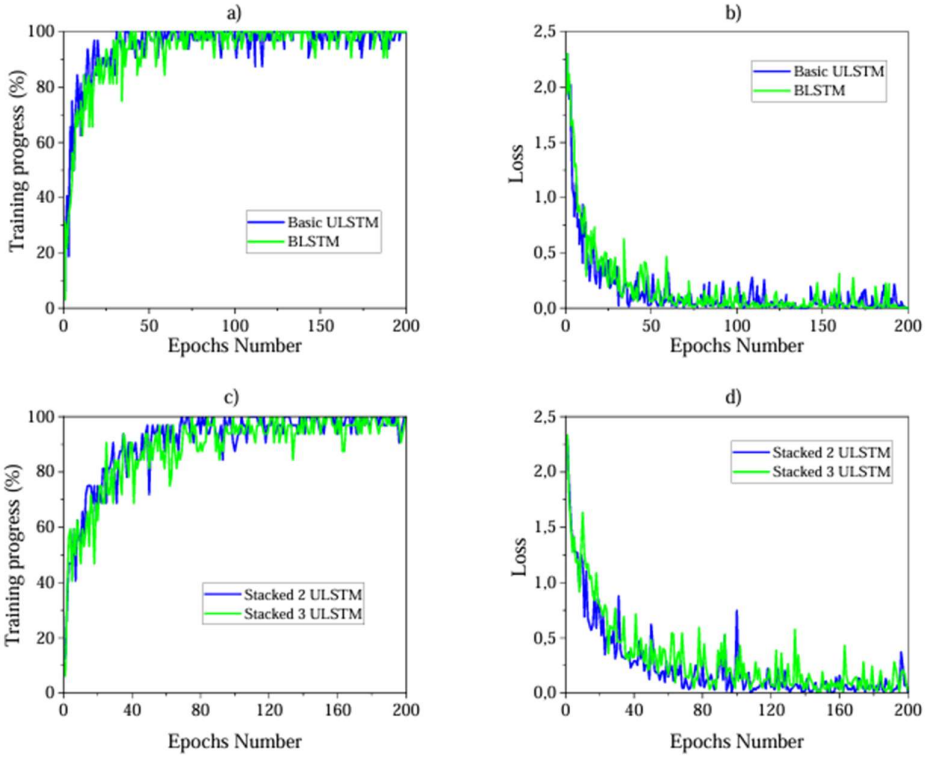

Figure 10. Training progress and loss curves respectively: a) & c) for Basic ULSTM & BLSTM and b) & d) for Stacked2 & Stacked3 ULSTMs.

Across many variations of LSTMs networks, bearing faults have been successfully classified both with input data as raw signals and with others introduced as previously and manually extracted feature vectors. Certainly, the performances of the classifications carried out are different. Nevertheless, LSTM networks deserve to be recognized, in agreement with the work of Verner [37], as having many advantages. One can cite, among others, the possibility of storing information over a large number of steps, the possibility of managing noise, the possibility of easily processing time series of variable lengths, the possibility of managing very long sequences and the ability to parallelize and accelerate calculations using a multitude of CPUs and GPUs.

As for the drawbacks illustrated mainly by the loss of effectiveness in certain situations increased by a fairly long training time. In our case, when the input data are frames of raw vibration signals, the moderate performance is a good illustration. Nevertheless, Reimers and Gurevych [38] recommend that these limitations can be overcome, possibly by making modifications relating to the size of a hidden state, to the number of LSTM layers in the stacked LSTM model, to the algorithm of optimization, on the method and rate of variational dropout regularization technique and LSTM architectures with integration layer, stacked BLSTM, etc.

5. Conclusion

In this paper, different variants of LSTM networks are proposed for bearing fault diagnosis. Thus, four different types of architectures were successfully tested. This method appears promising and has relatively low computational costs because it can be executed online on machines with limited processing power, as part of condition-based maintenance. This advantage increases its industrial implication and its practical applications in the context of condition-based maintenance. In addition, and compared to traditional methods, it can use raw vibration signals as input to the classifier without any pre-processing beforehand. Unfortunately, in such cases and within the limits of this study, training the LSTM network results in modest classification accuracy and quite long computational times. One of the solutions to the problem demonstrated during this study consists of using time-frequency-moment features that can be extracted manually beforehand, to be sent to the input of the classifier. Training the network using these two features significantly improves classification performance and reduces training time. Alternatively, the need to consider in perspective the testing of many other hyperparameters and other architectural models capable of improving the performance in question.

Acknowledgments

The authors express their sincere gratitude to Professor K.A. Loparo of Case Western Reserve University for permission to use their rolling vibration signal database.

Competing Interest Statement

The authors declare no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Data and Materials Accessibility

The datasets generated and/or analysed during the current study are available in the CWRU repository [34], https://engineering.case.edu/bearingdatacenter/welcome. Any other information about the findings of this study is available from the corresponding author, M. ATMANI (youcef.atmani@ensta.edu.dz), upon request.

Authors' Contributions

Mr. ATMANI as the first and the corresponding author, made substantial contributions to the problem statement, idea proposal, implementations, drafting of the work, paper writing, conception, and design of the work.

Mr. MESLOUB as one of the co-authors, provided significant additional means and calculation tools, discussed the results, revised and improved the paper writing, and revised the draft for important intellectual content.

Mr. RECHAK as one of the co-authors, checked and approved all aspects of the work.

Consent to Participate and for Publication

The authors declare their consent to participate in this work as well as to its publication

References

[1] C. Scheffer, P. Girdhar, "Practical Machinery Vibration Analysis and Predictive Maintenance", *Newnes, Elsevier*, 2004. https://doi.org/10.1016/b978-0-7506-6275-8.x5000-0.

[2] P. O'Donnell, "Report of Large Motor Reliability Survey of Industrial and Commercial Installations Part I," *IEEE Transactions on Industry Applications*, vol. IA-21, no. 4, pp. 853-864, 1985. https://doi.org/10.1109/TIA.1985.349532.

[3] R. B. Randall, J. Antoni, "Rolling Element Bearing Diagnostics - A Tutorial," *Mechanical Systems and Signal Processing*, vol. 425, no. 2, pp. 485-520, 2011. https://doi.org/10.1016/j.ymssp.2010.07.017.

[4] P. Henrıquez, J. B. Alonso, M. A. Ferrer, C. M. Travieso, "Review of Automatic Fault Diagnosis Systems Using Audio and Vibration Signals," *in IEEE Transactions on Systems,* Man, and Cybernetics, Systems, vol. 44, no. 5, pp. 642-652, 2014. https://doi.org/10.1109/TSMCC.2013.2257752.

[5] Choudhury, N. Tandon, "Application of acoustic emission technique for the detection of defects in rolling element bearings," *Tribology International*, vol. 33, no. 1, pp. 39–45, 2000. https://doi.org/10.1016/S0301-679X(00)00012-8.

[6] R. Dubey, D. Agrawal, "Bearing fault classification using ANN-based Hilbert footprint analysis," *IET Science, Measurement & Technology*, vol. 9, no. 8, pp. 1016-1022, 2015. https://doi.org/10.1049/iet-smt.2015.0026.

[7] Rojas, A. K. Nandi, "Detection and classification of rolling element bearing faults using support vector machines," *2005 IEEE Workshop on Machine Learning for Signal Processing,* Mystic, CT, USA, pp. 153-158, 2005. https://doi.org/10.1109/MLSP.2005.1532891.

[8] T. Benkedjouh, K. Medjaher, N. Zerhouni, S. Rechak, "Fault prognostic of bearings by using support vector data description," *2012 IEEE Conference on Prognostics and Health Management,* Denver, CO, USA, 2012, pp. 1-7. https://doi.org/10.1109/ICPHM.2012.6299511.

[9] V. Purushotham, S. Narayanan, S. A. N. Prasad, "Multi-fault diagnosis of rolling bearing elements using wavelet analysis and hidden Markov model-based fault recognition," *NDT&E International*, vol. 38, no. 8, pp. 654–664, 2005. https://doi.org/10.1016/j.ndteint.2005.04.003.

[10] J. Tian, C. Morillo, M. H. Azarian, M. Pecht, "Motor bearing fault detection using spectral Kurtosis-based feature extraction coupled with K-nearest neighbor distance analysis," *in IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793–1803, 2016. https://doi.org/10.1109/TIE.2015.2509913.

[11] Y. Atmani, S. Rechak, A. Mesloub, L. Hemmouche, "Enhancement in bearing fault classification parameters using Gaussian mixture models and Mel Frequency Cepstral Coefficients features," *Archives of Acoustics*, vol. 45, no. 2, pp. 283-295, 2020. https://doi.org/10.24425/aoa.2020.133149.

[12] Q. Tong, J. Cao, B. Han, X. Zhang, Z. Nie, J. Wang, Y. Lin, W. Zhang, "A Fault Diagnosis Approach for Rolling Element Bearings Based on RSGWPT-LCD Bilayer Screening and Extreme Learning Machine," *in IEEE Access*, vol. 5, pp. 5515–5530, 2017. https://doi.org/10.1109/ACCESS.2017.2675940.

[13] Z. Wang, Q. Zhang, J. Xiong, M. Xiao, G. Sun, J. He, "Fault diagnosis of a rolling bearing using wavelet packet denoising and random forests," *in IEEE Sensors Journal*, vol. 17, no. 17, pp. 5581–5588, 2017. https://doi.org/10.1109/JSEN.2017.2726011.

[14] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213 – 237, 2019. https://doi.org/10.1016/j.ymssp.2018.05.050.

[15] S. Zhang, S. Zhang, B. Wang, T. G. Habetler, "Deep Learning Algorithms for Bearing Fault Diagnostics-A Comprehensive Review," *in IEEE Access*, vol. 8, pp. 29857-29881, 2020. https://doi.org/10.1109/ACCESS.2020.2972859.

[16] W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mechanical Systems and Signal Processing*, vol. 100, pp. 439 – 453, 2018. https://doi.org/10.1016/j.ymssp.2017.06.022.

[17] S. Guo, T. Yang, W. Gao, C. Zhang, Y. Zhang, "An intelligent fault diagnosis method for bearings with variable rotating speed based on Pythagorean spatial pyramid pooling CNN," *Sensors*, vol. 18, no. 11, 3857, 2018. https://doi.org/10.3390/s18113857.

[18] Y. Chen, G. Peng, C. Xie, W. Zhang, C. Li, S. Liu, "ACDIN: Bridging the gap between artificial and real bearing damages for bearing fault diagnosis," *Neurocomputing*, vol. 294, pp. 61 – 71, 2018. https://doi.org/10.1016/j.neucom.2018.03.014.

[19] W. Mao, J. He, Y. Li, Y. Yan, "Bearing fault diagnosis with auto-encoder extreme learning machine: A comparative study," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 231, no. 8, pp. 1560 – 1578, 2017. https://doi.org/10.1177/0954406216675896.

[20] F. Xu, W. P. Tse, Y. L. Tse, "Roller bearing fault diagnosis using stacked denoising auto-encoder in deep learning and Gath-Geva clustering algorithm without principal component analysis and data label," *Applied Soft Computing*, 73, pp. 898 – 913, 2018. https://doi.org/10.1016/j.asoc.2018.09.037.

[21] C. Li, W. Zhang, G. Peng, S. Liu, "Bearing fault diagnosis using fully connected winner-take-all auto-encoder," *in IEEE Access*, vol. 6, pp. 6103 – 6115, 2018. https://doi.org/10.1109/ACCESS.2017.2717492.

[22] Z. Chen, W. Li, "Multisensor feature fusion for bearing fault diagnosis using sparse auto-encoder and deep belief

network," *in IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 7, pp. 1693 – 1702, 2017. https://doi.org/10.1109/TIM.2017.2669947.

[23] H. Liu, J. Zhou, Y. Xu, Y. Zheng, X. Peng, W. Jiang, "Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks," *Neurocomputing*, vol. 315, pp. 412 – 424, 2018. https://doi.org/10.1016/j.neucom.2018.07.034.

[24] S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735 – 1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735.

[25] W. Abed, S. Sharma, R. Sutton, A. Motwani, "A robust bearing fault detection and diagnosis technique for brushless DC motors under non-stationary operating conditions," *Journal of Control, Automation and Electrical Systems*, vol. 26, pp. 241 – 254, 2015. https://doi.org/10.1007/s40313-015-0173-7.

[26] L. Guo, N. Li, F. Jia, Y. Lei, J. Lin, "A recurrent neural network-based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98 – 109, 2017. https://doi.org/10.1016/j.neucom.2017.02.045.

[27] H. Pan, X. He, S. Tang, F. Meng, "An improved bearing fault diagnosis method using one-dimensional CNN and LSTM," *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 64, no. 7–8, pp. 443 – 452, 2018. https://doi.org/10.5545/sv-jme.2018.5249.

[28] K. Zhao, H. Shao, "Intelligent Fault Diagnosis of Rolling Bearing Using Adaptive Deep Gated Recurrent Unit," *Neural Processing Letters*, vol. 51, pp. 1165 – 1184, 2019. https://doi.org/10.1007/s11063-019-10137-2.

[29] Y. Bengio, P. Simard, P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *in IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157 – 166, 1994. https://doi.org/10.1109/72.279181.

[30] M. Schuster, K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673 – 2681, 1997. https://doi.org/10.1109/78.650093.

[31] B. Boashash, "Estimating and Interpreting the Instantaneous Frequency of a Signal - Part 1: Fundamentals," *in Proceedings of the IEEE*, vol. 80, no. 4, pp. 520 – 538, 1992. https://doi.org/10.1109/5.135376.

[32] B. Boashash, "Estimating and Interpreting the Instantaneous Frequency of a Signal - Part 2: Algorithms and Applications," *in Proceedings of the IEEE*, vol. 80, no. 4, pp. 540 – 568, 1992. https://doi.org/10.1109/5.135378.

[33] Y. N. Pan, J. Chen, X. L. Li, "Spectral Entropy: A Complementary Index for Rolling Element Bearing Performance Degradation Assessment," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 223, no. 5, pp. 1223 – 1231, 2009. https://doi.org/10.1243/09544062JMES1224.

[34] K. A. Loparo, "Bearings Vibration Data Sets," Case Western Reserve University, 2012. http://csegroups.case.edu/bearingdatacenter/home, Accessed 24 Nov 2021.

[35] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, N. Freitas, "Learning to learn by gradient descent by gradient descent," *arXiv:1606.04474v2 [cs.NE]*, 2016. https://doi.org/org/10.48550/arXiv.1606.04474.

[36] G. Kłosowski, T. Rymarczyk, D. Wójcik, S. Skowron, T. Cieplak, P. Adamkiewicz, "The Use of Time-Frequency Moments as Inputs of LSTM Network for ECG Signal Classification," *Electronics*, vol. 9, no. 9, 1452, 2020. https://doi.org/10.3390/electronics9091452.

[37] Verner, "LSTM Networks for Detection and Classification of Anomalies in Raw Sensor Data," Doctoral dissertation. Nova Southeastern University. Retrieved from NSUWorks, College of Engineering and Computing, (1074). 2019. https://nsuworks.nova.edu/gscis_etd/1074. Accessed 04 Oct 2023.

[38] N. Reimers, , I. "Gurevych, Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 338-348, 2017. https://doi.org/ 10.18653/v1/D17-1035.