Neuroevolution for bearing diagnosis

Rita Sleiman, Amani Raad, Souhayb Kass, Jérôme Antoni

Neuroevolution for Bearing Diagnosis

Rita SLEIMAN[1], Amani RAAD[1], Souhayb KASS[2], Jerome ANTONI[2]
[1]Univ Libanaise, Faculté de Génie, Tripoli, LIBAN
[2]Univ Lyon, INSA LYON, LVA, Lyon, France

Abstract

The monitoring of machinery and especially ubiquitous bearings in all means of transport has gained importance for decades in the industry because of the need to increase the reliability of machines and reduce the possible loss of production due to failures caused by the different faults. Many of the available techniques currently require a lot of expertise to apply them successfully. New techniques are required that allow relatively unqualified operators to make reliable decisions without knowing the mechanism of the system and analyzing the data. Reliability must be the most important criterion of the operation. Artificial intelligence is the revolutionary answer in all areas of industrial control. The main goal of this paper is to propose new solutions for bearing diagnosis based on deep neural networks (DNN). However, in general the optimization of the neural network architecture is done by trial and error, and the features reduction problem is solved by using the principal component analysis. In this paper, the application of the neuro-evolution is proposed for bearing diagnosis where the optimization of the neural network topology as well as the features reduction are done by an evolutionary genetic algorithm. An application of the general procedure is proposed for real signals; that shows the superiority of the combination between neural networks and genetic algorithms for bearing diagnosis.

1 Introduction

Rolling element bearing is one of the most critical components used in rotating machinery and many other mechanical equipment [1]. In fact, most of such machines' malfunctions are linked to bearing faults, such as fatigue, corrosion, overload, etc, that may occur unexpectedly if no predictive maintenance is used. This may lead to significant economic loss: high costs of maintenance and loss of revenue [2]. Therefore, bearing state monitoring and fault diagnosis are very important for discovering early bearing faults, assuring efficient and safe operation of all machines containing bearings.

In general, all bearing condition monitoring approaches in the literature can be classified into two categories: statistical-based approaches and pattern recognition-based approaches. In statistical-based approach, various signal processing tools are used, followed by statistical thresholds to detect the presence of a fault as well as to classify the different types of bearing faults [3,4]. For pattern-recognition-based approaches, several machine learning and artificial intelligence techniques, such as Artificial Neural Networks ANN, Support Vector Machine, fuzzy Expert Systems, Random Forest, and many other, have been successfully employed in fault diagnosis [5,6]. More recently, deep learning algorithms, such as deep neural networks, convolutional neural networks and deep belief networks have shown great capabilities in the field of computer vision [7], speech recognition [8] and natural language processing [9], due to their ability to discover hidden patterns in the data by using architectures composed of several non-linear learning layers. These deep learning algorithms were also applied in the field of industrial diagnosis and have been very useful and effective [10,11].

In recent years, a new artificial intelligence approach known as the 'Neuro-evolution' has attracted considerable attention as they proved to be essential in so many applications. The basic idea is that it applies evolutionary algorithms, and more specifically genetic algorithms GA, in order to construct a well suited artificial neural network for a certain application. Earlier successful applications in the field of Neuro-evolution are in reinforcement learning, evolutionary robotics, and artificial life. Sample applications include evolving behaviors for video games such as evolving new content in real time while the game is played [12],

controlling mobile robots such as evolving the neural networks of robots that were 3D-printed and could move around the real world [13], and investigating the evolution of biologically-relevant behaviors such as investigating abstract evolutionary tendencies, like the evolution of modularity or how biological development interacts with evolution [14].
Meanwhile, in order to benefit from such powerful tool of artificial intelligence in the field of industrial diagnosis, researchers in this domain have tried to apply this combination of neural networks and genetic algorithm on bearing fault classification problem: some have used the genetic algorithm for the weights optimization [15], and others for the features selection problem [16].
The main goal of this paper is to apply this concept of Neuro-evolution in industrial automatic diagnosis without any human intervention, especially after the fourth industrial revolution characterized by the fusion of all modern technologies and the concept of digital factories [17]. In details, the optimization of the number of hidden layers and nodes in an artificial neural network is performed using this concept; in order to guarantee ANN architecture with the highest classification accuracy. In addition, feature reduction is obtained using the genetic algorithm instead of the Principal Component Analysis (PCA) [18], since it needs the tuning of some parameters whereas the GA gives good results without any assumptions.
The main benefit behind this concept is to extend the optimization of the topology as well as the features from one dataset to another without any human intervention to find automatically the best classification accuracy; which could not be obtained using neural networks alone. The extracted features are divided into temporal classical and spectral ones. The spectral features are based on the indicators of probability of presence of faults introduced by S. Kass and Al based on the spectral coherence [19]. These indicators are very powerful due to their ability of condensing the whole information initially displayed in three dimensions into a scalar and returning information in terms of a probability of presence of a fault. They also take into consideration uncertainties in the bearing characteristic frequencies, which is crucial in bearing diagnosis. However, a statistical threshold was derived for decision making. In this paper, this statistical threshold is absent and replaced by the techniques of artificial intelligence already mentioned above, and that will be detailed in the next sections.
This paper is divided into seven sections, where section two introduces briefly the artificial neural networks and the third section describes the genetic algorithm. Section four presents in a general way the effectiveness of the combination between neural networks and genetic algorithm, and section five describes the application done in this paper. In section six, the features used as input to the neural network are presented and described and the proposed method is validated using two datasets in section seven. Conclusions are drawn in the last section.

2 Neural Network

Inspired by the human brain, Artificial Neural Networks (ANN) are a family of machine learning models that mimic the structural elegance of the neural system and learn patterns inherent in observations [20]. There are several types of neural networks: Backpropagation networks, Deep Belief networks, Convolutional neural networks, Recurrent networks, Radial Basis Function networks, etc…
One of the most widely used type of neural networks is the Backpropagation network which is a multi-layer perceptron consisting of an input layer with nodes representing input variables to the problem, an output layer with nodes representing the different classes of the corresponding classification problem, and one or more hidden layers containing nodes to help discovering the hidden patterns in the data.
While, in theory, it is possible to apply different types of activation functions for different layers, it is common to apply the same type of an activation function for the hidden layers in the literature [21]. However, it should be a nonlinear function such as the 'logistic sigmoid' function $\sigma(z) = 1/(1+\exp[-z])$, and the 'hyperbolic tangent' function $\tanh(z) = (\exp[z]-\exp[-z]) / (\exp[z]+\exp[-z])$.
In addition, the connections between different layers are weighted: These weights are initially randomly set, and then adjusted between successive training cycles (learning process) in order to increase the classification accuracy. This is generally done by minimizing a cost function using several well-known optimization algorithms: the gradient descent which is the simplest and most popular training algorithm [22], the

stochastic gradient descent which is a lighter algorithm and therefore faster than its all-encompassing cousin [23], the Adaptive Moment, *Adam*, which is based on adaptive estimates of lower-order moments [24], etc.
For K-class classification, it is common to use a cross-entropy cost function defined as follows:

$$E(\omega) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} ln y_{nk} \quad (1)$$

where ω is the weights set, t_{kn} denotes the *kth* element of the target vector t_n and y_{kn} is the *kth* element of the prediction vector y_n for x_n, given a training data set $\{x_n,t_n\}$, n=1…N.

3 Genetic Algorithm

Genetic algorithm (GA) was first introduced by John Holland, from the University of Michigan, in 1975 in its publication '*Adaptation in Natural and Artificial System*' [25]. It is a general-purpose optimization algorithm that is, like neural networks, a bio-inspired artificial intelligence tool. It is a particular class of evolutionary algorithms that are based on the mechanics of natural selection and natural genetics. Unlike other optimization algorithms that improve a single solution at a time, GA uses a strategy of parallel search by working on a population of candidate solutions (also called individuals) to an optimization problem that evolves toward better solutions, enabling extreme exploration and massive parallelization. The basic idea is that over time, evolution will select the most suitable solutions [26]. This is done by evaluating, for each individual in a generation, the fitness function: the higher the fitness, the most probably the individual will be selected.
Technically, when solving an optimization problem using the GA, one must first define:

- Individual (or chromosome): Composed of several genes, it can be binary or real encoded. An individual represents one possible solution of the problem; the collection of multiple chromosomes forms the population which represents a subset of the whole searching space.
- Fitness function: It is the function we tend to maximize; if the problem consists in minimizing a certain function, the latter should be transformed into a fitness function by simply inverting it. The fitness function corresponds to an evaluation of how good the candidate solution is.

Once the fitness function is properly defined, the genetic algorithm generally starts by randomly generating the initial population. It should be large enough so that any solution in the search space can be later engendered [27]. Then, the algorithm loops over an iteration process to make the population evolve. Each iteration consists of the following genetic operators:

- Selection: After evaluating the fitness f(i) for each individual i, Roulette wheel selection is applied in order to give the fittest individuals a higher chance to be selected than weaker ones. This is done by calculating the probability of selection of each individual as follows:

$$p(j) = \frac{f(j)}{\sum_{i=1}^{N} f(i)}; j = 1, \dots, N \quad (2)$$

  where N is the total number of individuals in the population.
- Crossover: Selected individuals should be paired randomly, with a crossover rate Pc, for recombination. The latter is done by exchanging genes between one or more crossover points that are randomly generated.

- Mutation: It randomly alters one or more genes in a chromosome, with a mutation rate Pm, changing it from 1 to 0 and vice versa. It is a powerful operator used to avoid falling into a local optimum.

- Replacement: The new selected, recombined and probably mutated individuals form a new population that replaces the old one.

Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. When the algorithm terminates, the individual with the highest fitness is regard as the approximate optimal solution.

4 Combination of neural networks with genetic algorithm

Generally, when constructing an ANN classification model, choosing its various parameters, such as its topology, activation functions, training sample size, etc, may greatly impact the classification results [28]. Basically, one of the most critical tasks in artificial neural network design is choosing the best topology (architecture) that gives the highest classification accuracy. One starts with no prior knowledge as to the number of hidden layers and number of hidden nodes required [20]; Choosing a small number of hidden layers and nodes will lead to an 'underfitting' problem: ANN will not be able to reveal complex and hidden patterns in the data. In contrast, a network with too many hidden nodes tries to model exactly the training dataset following all its noise, and leading to a poor generalization for additional untrained data: this is known as the 'overfitting' problem. In the literature, several researchers have proposed different methodologies for fixing the number of hidden neurons. Most of the methodologies are presented in a review where the authors also proposed a new method to fix the hidden neurons in Elman networks for wind speed prediction in renewable energy systems [29]. In general, the most used technique for finding the optimal architecture is by trial and error. The latter approach has several limitations such as it is time-consuming and the obtained network structure may not be optimal.

In recent years, new topology optimization techniques based on evolutionary algorithms have gained great interest by the researchers in the domain of artificial intelligence. The combination between artificial neural network and genetic algorithm, also known as 'Neuro-evolution', has shown important capabilities and effectiveness in so many fields as it meets the potential of the increasing high performance computation capabilities in our days [30]. Initially, the use of Neuro-evolution was only restricted to the optimization of neural networks weights in order to overcome some limitations of the backpropagation algorithm. Later, this combination was extended to optimize also the ANN architecture. The following flowchart describes the principal steps of the ANN architecture optimization algorithm.

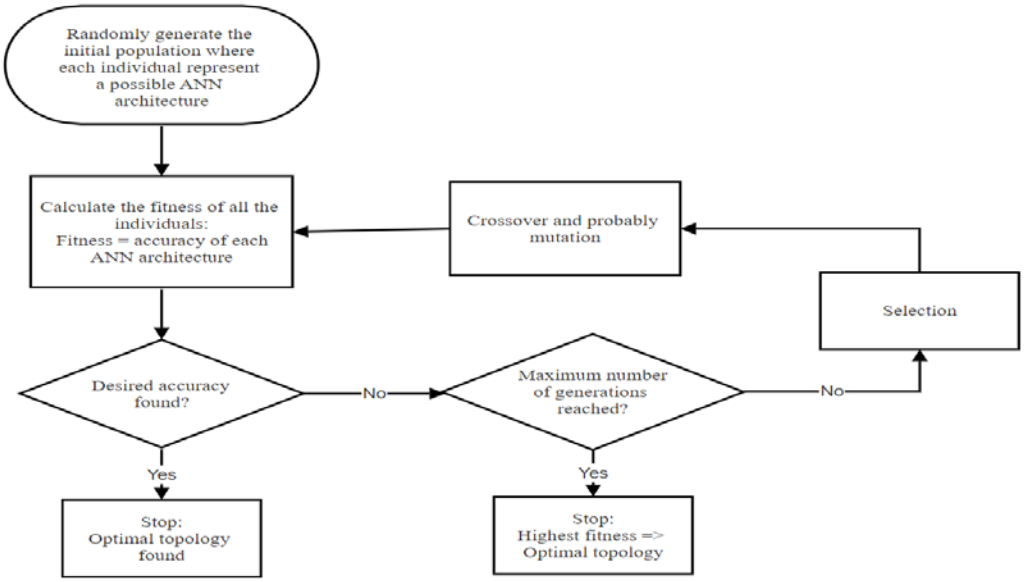
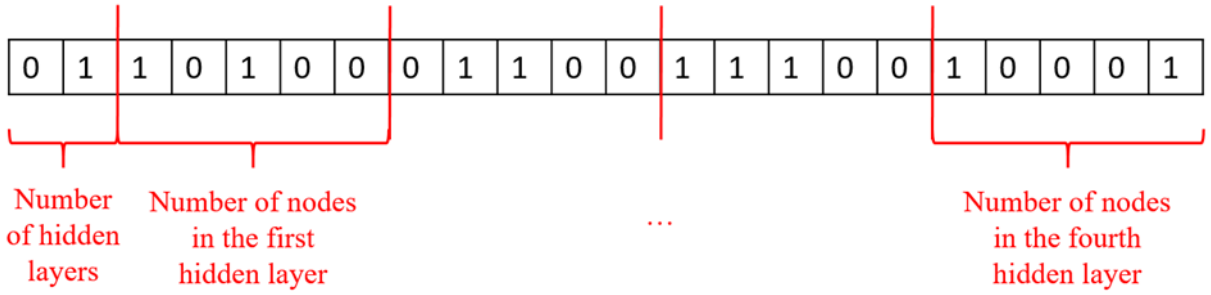


Fig.1 Flowchart of the ANN architecture optimization algorithm

5 Application

As mentioned in section “GA”, in order to apply the GA for any optimization problem, one must first define the encoding way of an individual: the process of representing individual genes. We may have binary or real encoding; The encoding depends mainly on the application. The most common way of encoding is binary strings, which is used in this paper.

For the architecture optimization problem, the chromosome is a binary string composed of 22 bits (genes). The first two bits are reserved for the binary representation of the number of hidden layers, while the other 20 bits are devided into four parts, each of five bits, representing the binary representation of the number of nodes in each hidden layer. This results in a number of hidden layers varying between 1 and 4 layers, and a number of nodes in a layer varying between 1 and 32 nodes. Such chromosome can be illustrated as follows:



On the other hand, such optimization problem tends to maximize the classification accuracy. Therefore, the fitness function must be directly linked to it. The only difference is that a mapping is used in a way to make 90% accuracy worth 10% fitness and 100% accuracy worth 100% fitness. The algorithm that illustrates this idea is as follows:

If accuracy<90%:

Fitness=10%

Else:

*Fitness=((Accuracy-90%)/(100%-90%)) *(100%-10%) + 10%*

The main reason for this mapping is the use of the ‘Roulette wheel’ selection technique that calculate the probability of selection based on the fitness values. Accordingly, mapping is essential in order to ensure that probabilities of selection in a population are not so close in the selection process.

For the features reduction problem, the total algorithm is the same, but with a single difference which is the way of representing an individual. Here, each chromosome in the population is composed of n bits, where n is the total number of features. The value of each bit can be either 1, which indicates the presence of the corresponding feature, or 0, which indicates its absence. In this case, the chromosome can be illustrated as follows:

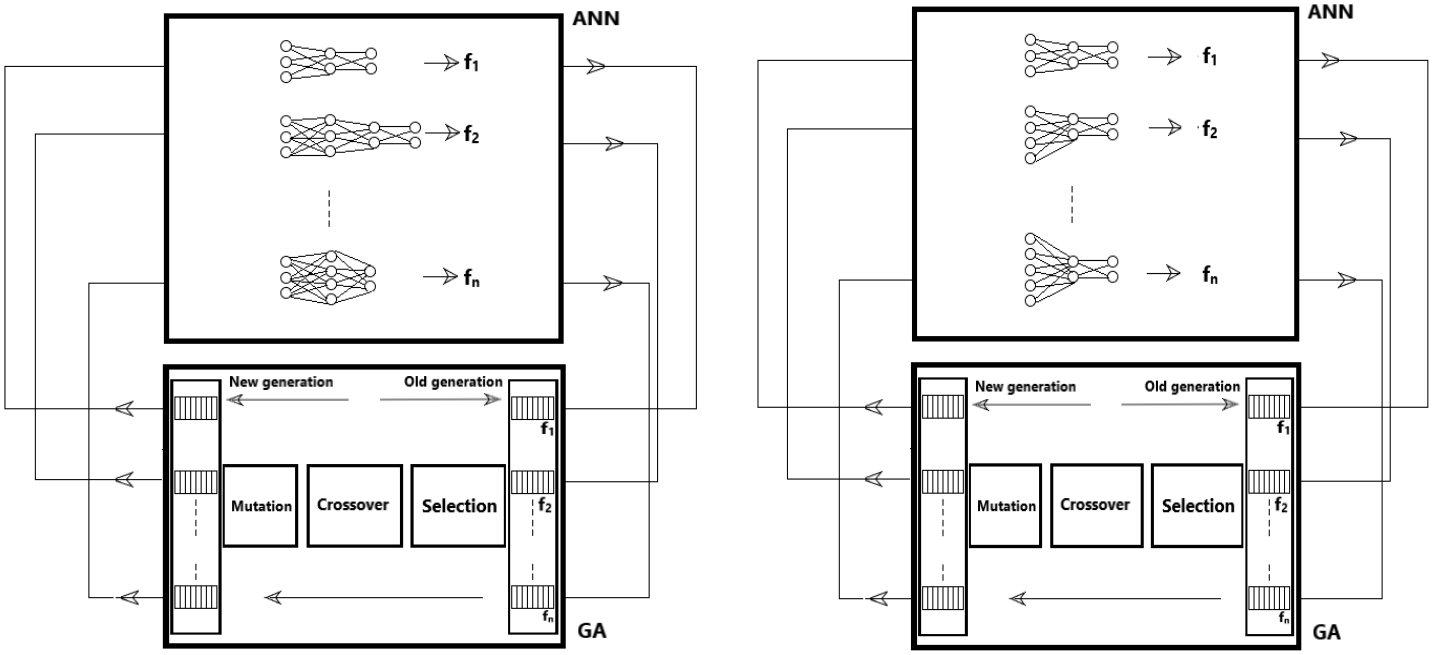


Fig.2 Combination between ANN and GA in two cases of ANN architecture and Feature optimization.

6 Features

6.1 Classical Features

Feature extraction is a very important step in a bearing diagnosis problem. Any feature chosen to be extracted from bearing signals will directly impact the classification results. Therefore, one must properly select the best feature set before moving to the next step of choosing the best classifier.

For detecting the change in bearing signal, traditional statistic features can be used. The advantages of using these features is essentially the ease of implementation and the low computational time. Accordingly, this paper proposes the use of traditional time-domain features presented in the following table:

RMS	$\left(\frac{1}{N}\right)\sum_{i=1}^{N} X_i^2^{\frac{1}{2}}$
Kurtosis	$\frac{1}{N}\sum_{i=1}^{N}\frac{(x_i-\bar{x})^4}{\sigma^4}$
Peak to peak	$x_{max}-x_{min}$
Crest Factor	$\frac{\max\lvert x_i\rvert}{RMS}$
Shape factor	$\frac{RMS}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$
Impulse factor	$\frac{\max\lvert x_i\rvert}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$
Margin factor	$\frac{\max\lvert x_i\rvert}{(\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert^{\frac{1}{2}})^2}$

Tab.1 Temporal features

Along with the above traditional features, extra powerful features for bearing fault classification, based on the second order of cyclostationarity, are used in this paper. These features are deduced from a recently developed indicator that will be detailed in the next section.

6.2 Spectral features

Souhayb and al [19] introduces a new autonomous method of bearing diagnosis in the case where the fault characteristic orders are known, taking into account all factors that may hinder this diagnosis. It is based on the development of new scalar indicators, which can be interpreted in terms of the probability of the presence of a fault. These indicators result from a post-processing of spectral coherence, calculated using the fast version of the spectral coherence algorithm proposed by Antoni et al [31]. It was chosen as a basis for the indicator because it is considered to be the optimal three-dimensional representation in which the bearing failure manifests itself clearly despite extreme operating situations.

In details, the envelope spectrum is calculated, as a first step, by integrating the spectral coherence according to the frequency variable. Therefore, all information initially displayed in three dimensions will be condensed into two dimensions. The next step is to recognize the characteristic peaks of the defects. This recognition must be confirmed by checking the presence of these peaks (theoretical fault frequency and its harmonics) in the envelope spectrum. This step then consists in searching for the maximum amplitude m_i in narrow bands centered on the expected theoretical orders. The search for faults harmonics in bands is intended to compensate for the effects of the sliding phenomenon; The latter is critical since it causes a random deviation from the theoretical orders given by the manufacturer and compromises the automatic tracking of fault harmonics when performed on a single specific order. Therefore, a band B_1 is first defined, centered at the theoretical fault characteristic order α_{c1}, with a deviation tolerance of 100X%, typically between 5% and 10%. The lower and the upper bounds α_1^L and α_1^U of band B_1 are thus defined as:

$$\begin{cases} \alpha_1^L = \alpha_{c1} - X_{\alpha_{c1}} \\ \alpha_1^U = \alpha_{c1} + X_{\alpha_{c1}} \end{cases} \tag{3}$$

After defining the highest peak m_1 in B_1, a measure that represents the probability of the presence of the first harmonic of the fault, PPF_1, is calculated based on the statistical threshold λ_{1-p} already obtained from the histogram of the envelope spectrum with p=0.1. PPF_1 is calculated as follows:

$$PPF_1 = \begin{cases} \dfrac{m_1 - \lambda_{1-p}}{m_1} & m_1 \geq \lambda_{1-p} \\ 0 & m_1 < \lambda_{1-p} \end{cases} \tag{4}$$

The basic idea is that, as with visual inspection, the presence of the characteristic peaks of the defects is confirmed according to the intensity of their amplitudes. If these amplitudes are greater than a statistical threshold, representing background noise, then these peaks are considered symptoms of the defect.

In the case of the non-zero value of PPF_1, the algorithm then searches for the presence of the second harmonic in a new band B_2. In order to properly define B_2, the center of the first band is first corrected to account for the possible mismatch between the actual and the theoretical fault order by defining $\alpha_{C_1}^{corr}$ such that $\alpha_{C_1}^{corr}$=α where the highest peak m_1 was found. Thus, B_2 is centered on α_{C_2}= 2 * $\alpha_{C_1}^{corr}$ and given the same bandwidth as B_1, PPF_2 is calculated. The algorithm terminates when either PPF_i is equal to zero, or a maximum number of harmonics is set. The general formula describing the theoretical fault frequency correction is:

$$\alpha_{C_i} = \frac{i}{i-1} \sum_{n=1}^{i-1} \frac{\alpha_{C_n}^{corr}}{n}, \quad i > 1 \tag{5}$$

Once the PPF_is in all concerned bands have been calculated, an overall indicator PPF is calculated as the mean value.

$$PPF = \frac{1}{n}\sum_{i=1}^{n} PPF_i \quad (6)$$

It should be noted that PPF increases with the severity of the fault. This is because the severity of the fault affects the amplitudes and number of peaks that characterize it.

In [19], defect detection was performed using a non-parametric hypothesis test using the proposed indicator. However, this article follows a different direction in which four of the proposed indicator, each on one of the four fault frequencies: Ball Pass Frequency Inner race (BPFI), Ball Pass Frequency Outer race (BPFO), Ball Spin Frequency (BSF) and Fundamental Train Frequency (FTF), are used as input parameters for the neural network. The four features, PPF_BPFI, PPF_BPFO, PPF_BSF and PPF_FTF, are used to calculate the probability of presence of different types of faults; inner race (PPF_BPFI), outer race (PPF_BPFO) and ball fault (PPF_BSF and PPF_FTF).

7 Experimental Results

7.1 Bearing Data Center

The performance of the proposed algorithm is now evaluated on the bearing signals provided by the CWRU database. The CWRU database has been used in many references and can be considered as a reference to test newly proposed [32].

The datasets are divided into four categories: 48k baseline, 12k drive end fault, 48k drive end fault and 12k fan end fault – according to the sample frequency and the fault's location. The experimental setup consists of a 1.4914 kW, reliance electric motor driving a shaft on which a torque transducer and encoder are mounted. Torque is applied to the shaft via a dynamometer and electronic control system. Four types of vibration signals are collected (normal, ball fault, inner-race fault, and outer-race fault), acquired by accelerometer sensors under different operating loads and speeds. The bearing type is a deep groove ball bearing SKF6205-2RS JEM.

BPFI	BPFO	FTF	BSF
5.415	3.585	0.3983	2.357

Tab.2 Multiplicative factors to calculate four fault frequencies

The table below presents some of the signals from this dataset with some of their corresponding features values (calculated for the four fault frequencies).

Signal	**PPF_BPFI**	**PPF_BPFO**	**PPF_BSF**	**PPF_FTF**	**Kurtosis**	**RMS**	**Peak to peak**	**Margin factor**
Inner fault	0.99	0.53	0.15	0.54	5.38	0.29	3.11	10.42
Outer fault	0.76	0.96	0.04	0.48	6.94	0.31	4.64	14.99
Ball fault	0.45	0.1	0.55	0.89	3.77	2.14	20.36	7.83
Normal	0.18	0.06	0.2	0.03	2.9	0.06	0.66	7.56

Tab. 3 Some of the signals with their features

As indicated in this table, for an inner race faulty signal, the probability of presence of a fault indicator that searches for the presence of BPFI (PPF_BPFI) has a high value=0.99, which insure the presence of the inner fault. The same is for the outer race and inner race faulty signals that have [PPF_BPFO=0.96] and

[PPF_BSF=0.55 & PPF_FTF=0.89] respectively. However, the temporal features values clearly show their ability for detection.

7.2 Second Dataset

In order to validate the proposed algorithm, the latter is evaluated also on another dataset acquired by John Stokes in the University of New South Wales (UNSW) in Australia. The test bench has a gearbox composed of two shafts, one of which is driven by a three-phase motor. The power flows through a hydraulic motor and pump. The two input and output shafts are placed parallel to each other, and are connected to the gearbox by two bearings each. An accelerometer was installed above the defective bearing, and the signals were temporally sampled with a sampling frequency of 48 KHz. The defects installed are localized defects in the form of a small superficial notch placed either on the outer ring, or on the inner ring or on one of the balls. The vibration signals were acquired under different conditions of rotational speed, 3, 6 and 10Hz, and load torque, 25, 50, 75, 100 Nm. The bearing under-test, *Koyo 1205*, has the following fault frequencies:

BPFI	BPFO	FTF	BSF
7.11	4.89	0.41	2.65

Tab.4 Multiplicative factors to calculate four fault frequencies

The table below presents some of the signals from this dataset with some of their corresponding features values (calculated for the four fault frequencies).

Signal	**PPF_BPFI**	**PPF_BPFO**	**PPF_BSF**	**PPF_FTF**	**Kurtosis**	**RMS**	**Peak to peak**	**Margin factor**
Inner fault	0.95	0.08	0.01	0.00	2.89	9.89	65.88	5.83
Outer fault	0.2	0.82	0.14	0.10	3.25	4.17	3.65	1.35
Ball fault	0.16	0.19	0.83	0.18	3.00	5.65	45.5	5.59
Normal	0.055	0.18	0.27	0.57	4.01	4.26	9.25	2.21

Tab. 5 Some of the signals with their features

7.3 Results

The same code was applied on both of the datasets; Even the GA parameters were set the same. These parameters were chosen based on GA logic and concept of starting with a large population in order to be able to obtain a large variety of solutions, and evolving the population over too many generations so we attend the desired solution. Accordingly, the total number of individuals was set to 20, and the maximum number of generations to 30. These numbers may differ from one application to another. In addition, the crossover probability was set to 0.7, which indicates that only 70% of the individuals will be recombined to form the new population. And finally, it is well-known that the mutation probability must not be high, since it will, in that case, negatively influence the main GA concept of evolving the population towards fittest solutions. Consequently, it was set to 0.03 (3%).

For the Bearing Data Center, results have shown that there may be several architectures giving the highest accuracy = 99%. This result is very logic since we are not solving an optimization problem with a well-defined equation and a unique solution. Having several topologies could be very helpful and essential in many applications especially when the data is too large and requires many hidden layers. The architectures found by the GA are formed by two or three hidden layers:

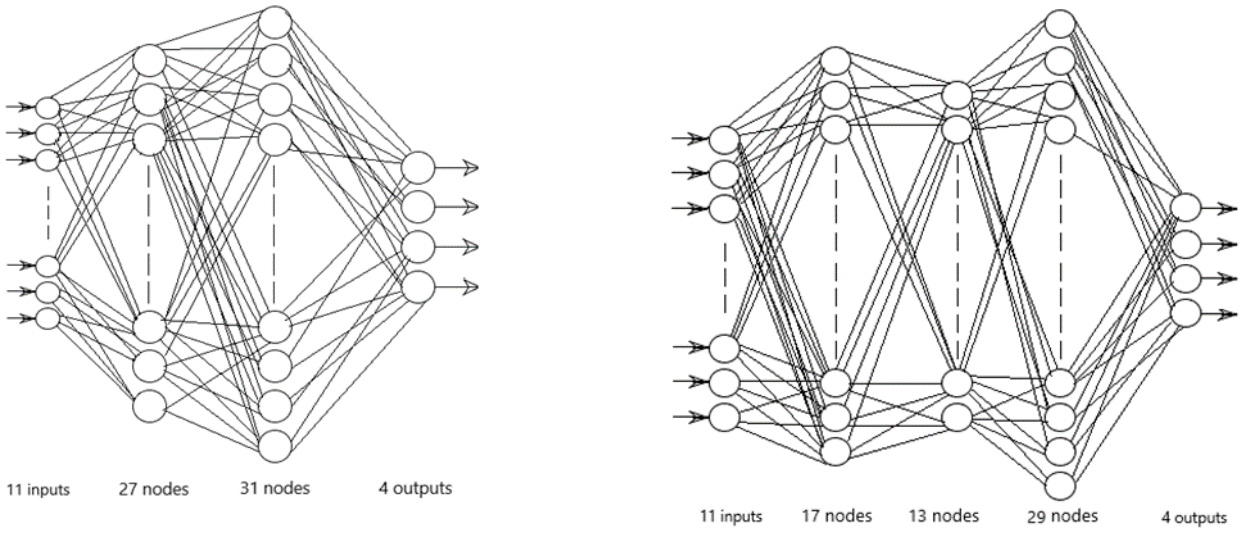


Fig.3 Two possible architectures obtained by the GA

For the features optimization, only six features out of eleven are relevant for the ANN. Those features are: PPF_BPFI, PPF_BPFO, PPF_FTF, Margin Factor, Peak to Peak, RMS. They are able to give the same classification accuracy 98% when fed into the ANN having the best architecture. The results were very convincing: this combination of both temporal features, capable of detecting the presence of a fault, and the spectral features, capable of classifying the different fault types, will surely be enough for getting good classification results. The figure below is an example of an envelope spectrum obtained for an inner race faulty signal. It explains clearly why the spectral indicator was able to classify the different faults by searching on the theoretical fault frequencies taking into consideration the slip phenomenon that may occur.

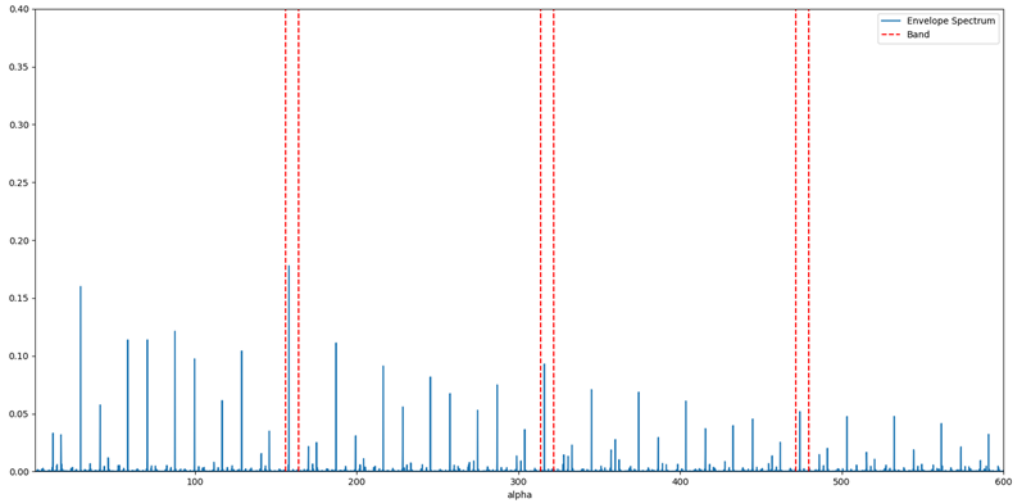


Fig.4 Envelope Spectrum for an inner race faulty signal

Below is the confusion matrix of the resulting classification model, where 0 indicates normal signals, 1 inner race fault, 2 outer race fault and 3 ball fault.

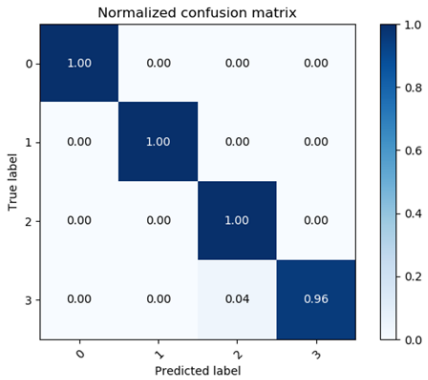


Fig.5 Normalized confusion matrix

Applying the same code on the second dataset also gave important results. The same GA was able to find ANN architectures giving a 100% classification accuracy. These architectures are formed by a single hidden layer containing 31 nodes, or by 2 hidden layers having number of nodes higher than 20. The GA convergence was much faster in this case than in the bearing data center dataset. The main reason is that the envelope spectrum for this database was more clear and without noise which directly influenced the spectral indicators values. This is also the reason of obtaining a higher classification accuracy. Below is an envelope spectrum of an inner race faulty signal from the second database:

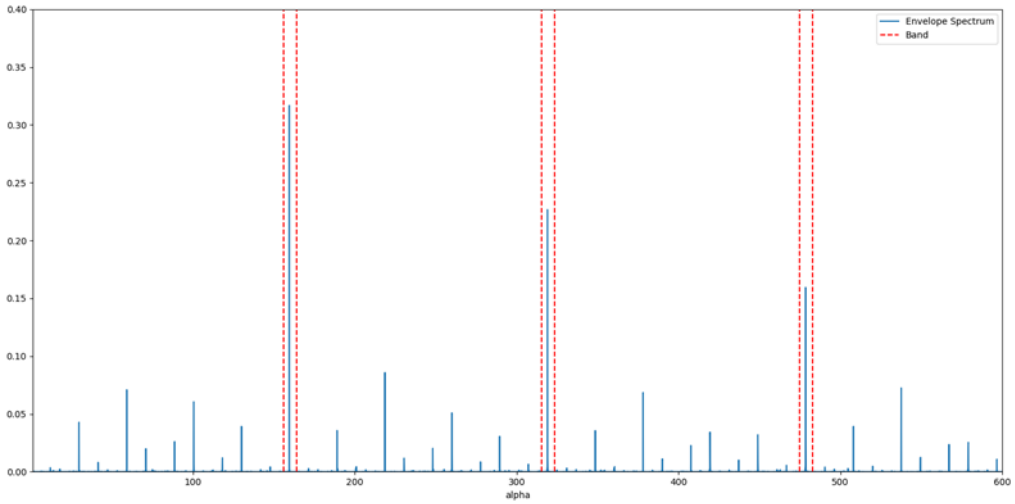


Fig.6 Envelope Spectrum for an inner race faulty signal

For the features optimization problem, here also, the number of features was reduced from 11 to 6: PPF_BPFI, PPF_BPFO, PPF_BSF, Margin Factor, Peak to Peak, RMS, with a single difference of having PPF_BSF instead of PPF_FTF. This result was a little bit confusing, for both of the datasets, since it is well-known that a signal with a ball fault is characterized by the presence of the combination of the ball spin and the fundamental train frequencies. Below is the confusion matrix of the resulting classification model:

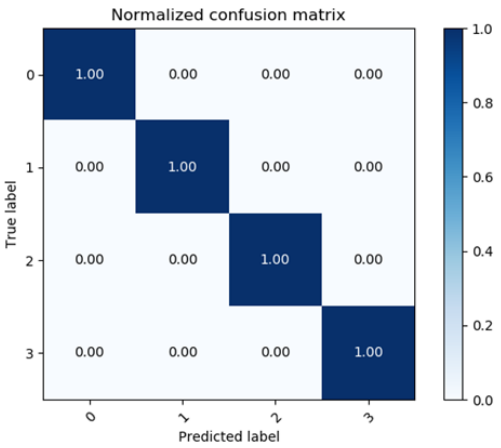


Fig.7 Normalized confusion matrix

8 Conclusion

This study presents an algorithm based on neuro-evolution for bearing fault classification problems. Although it is widely known that Neural Networks alone are used for such applications, the fact that its architecture is fixed makes it not adaptive to change. This study suggests optimizing the neural network architecture and reducing the number of features using the genetic algorithm. This technique was tested on bearing fault detection and classification with two datasets and gave promising results. This concept of applying the neuro-evolution is very powerful and effective in so many fields especially when having a huge dataset to train and a big number of features. Additional work can be done with neuro-evolution like weight optimization in order to overcome some limitations of the classical gradient descent method.

References

[1] K.A. Loparo, M.L. Adams, W. Lin, M.F. Abdel-Magied, N. Afshari, Fault detection and diagnosis of rotating machinery, IEEE Trans. Ind. Electron. 47 (5) (2000) 1005–1014

[2] C. Bianchini, F. Immovilli, M. Cocconcelli, R. Rubini, A. Bellini, Fault detection of linear bearings in brushless AC linear motors by vibration analysis, IEEE Trans.Ind. Electron. 58 (5) (2011) 1684–1694

[3] Raad, J. Antoni, M. Sidahmed, Indicators of cyclostationarity: Theory and application to gear fault monitoring, Mech. Syst. Signal Process. 22 (3) (2008) 574–587

[4] A. Klausen, K.G. Robbersmyr, H.R. Karimi, Autonomous bearing fault diagnosis method based on envelope spectrum, IFAC-PapersOnLine 50 (1) (2017) 13378–13383

[5] Widodo, A.; Yang, B.-S. Support vector machine in machine condition monitoring and fault diagnosis. Mech. Syst. Signal Process. 2007, 21, 2560–2574

[6] Kuo, R.J. Intelligent diagnosis for turbine blade faults using artificial neural networks and fuzzy logic. Eng. Appl. Artif. Intell. 1995, 8, 25–34

[7] A. Cruz-Roa, A. Basavanhally, F. Gonzalez, H. Gilmore, M. Feldman, S. Ganesan, N. Shih, J. Tomaszewski, A. Madabhushi, Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks, In: Medical Imaging 2014: Digital Pathology, vol. 9041

[8] J. Huang, B. Kingsbury, Audio-visual deep learning for noise robust speech recognition, In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 7596–7599

[9] R. Sarikaya, G.E. Hinton, A. Deoras, Application of deep belief networks for natural language understanding, IEEE/ACM Trans. Audio Speech Lang. Process. 22 (4) (2014) 778–784
[10] Tao, J.; Liu, Y.; Yang, D. Bearing fault diagnosis based on deep belief network and multisensor information fusion. Shock Vib. 2016, 2016
[11] Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufier, M.; Verstockt, S.; van de Walle, R.; van Hoecke, S. Convolutional neural network based fault detection for rotating machinery. J. Sound Vib. 2016, 377, 331–345
[12] Togelius, J., Yannakakis, G. N., Stanley, K. O. & Browne, C. Search-based procedural content generation: a taxonomy and survey. IEEE Trans. Comput. Intell. AI Games 3, 172–186 (2011)
[13] Lipson, H. & Pollack, J. B. Automatic design and manufacture of robotic lifeforms. Nature 406, 974–978 (2000)
[14] Clune J, Mouret J-B, Lipson H. 2013 The evolutionaryorigins of modularity. Proc R Soc B 280: 20122863. http://dx.doi.org/10.1098/rspb.2012.2863
[15] J. Montana, David & Davis, Lawrence. (1989). Training Feedforward Neural Networks Using Genetic Algorithms.. 762-767
[16] Fröhlich, Holger & Chapelle, O & Scholkopf, B. (2003). Feature Selection for Support Vector Machines by Means of Genetic Algorithms. Int Conf on Tools with Artificial Intelligence. 142 - 148. 10.1109/TAI.2003.1250182
[17] Trans. Ind. Electron. 2015, 62, 657–667, doi:10.1109/tie.2014.2308133. 2. Jeschke, S.B.C.; Song, H.; Rawat, D.B. Ind
[18] Malhi, A.; Gao, R.X. PCA-based feature selection scheme for machine defect classification. IEEE Trans. Instrum. Meas. 2004, 53, 1517–1525
[19] Kass, Souhayb & Raad, Amani & Antoni, Jerome. (2019). Self-Running Bearing Diagnosis Based on Scalar Indicator using Fast Order Frequency Spectral Coherence. Measurement. 138. 467-484. 10.1016/j.measurement.2019.02.046
[20] Basheer, Imad & Hajmeer, M.N.. (2001). Artificial Neural Networks: Fundamentals, Computing, Design, and Application. Journal of microbiological methods. 43. 3-31. 10.1016/S0167-7012(00)00201-3
[21] Suk, H.-I. “An Introduction to Neural Networks and Deep Learning,” in Deep Learning for Medical Image Analysis, London, Academic Press - Elsevier, 3-24 (2017)
[22] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," in IEEE Transactions on Neural Networks, vol. 5, no. 6, pp. 989-993, Nov. 1994
[23] L. Bottou. Stochastic gradient learning in neural networks. In Proceedings of Neuro-Nˆımes. EC2, 1991
[24] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations
[25] John H. Holland. 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge, MA, USA
[26] Sivanandam, S.N and S.N. Deepa. 2007. Introduction to Genetic Algorithms. Springer Berlin Heidelberg.
[27] Xu, Q., Xu, Z.Q. and Wang, T. (2015) A Data-Placement Strategy Based on Genetic Algorithm in Cloud Computing. International Journal of Intelligence Science, 5, 145-157. http://dx.doi.org/10.4236/ijis.2015.53013
[28] Tiwari, K.C.. (2001). Neural network parameters affecting image classification. Defence Science Journal. 51. 263-278. 10.14429/dsj.51.2237
[29] Gnana Sheela, K & Deepa, S N. (2013). Review on Methods to Fix Number of Hidden Neurons in Neural Networks. Mathematical Problems in Engineering. 2013. 10.1155/2013/425740
[30] Stanley, Kenneth & Clune, Jeff & Lehman, Joel & Miikkulainen, Risto. (2019). Designing neural networks through neuroevolution. Nature Machine Intelligence. 1. 10.1038/s42256-018-0006-z
[31] J. Antoni, G. Xin, N. Hamzaoui, Fast computation of the spectral correlation,Mech. Syst. Signal Process. 92 (2017) 248–277
[32] Smith, Wade & Randall, R.B.. (2015). Rolling Element Bearing Diagnostics Using the Case Western Reserve University Data: A Benchmark Study. Mechanical Systems and Signal Processing. 64-65. 10.1016/j.ymssp.2015.04.021