Detection of machine mechanical faults using vibrations and deep autoencoders

Mario Eltabach, Gérard Govaert

Detection of machine mechanical faults using vibrations and deep autoencoders

Mario ELTABACH[1], Gérard GOVAERT[2]

[1]CETIM, 52 Av Félix Louat, 60300 Senlis, France
mario.eltabach@cetim.fr
[2]UTC, Heudiasyc UMR 7253, 57 Av. de Landshut, 60200 Compiègne, France

Abstract

Deep learning models represent a new learning paradigm in artificial intelligence (AI). Recent breakthroughs in image analysis and voice recognition have generated enormous interest in many other areas such as the diagnosis of rotating machinery providing voluminous data during the life of these machines.
A gear endurance test bench was designed to create a database of vibration signals ranging from a healthy state to a degraded state. The aim of our work is to test common methods and the use of unsupervised deep learning in the detection of deviation from the normal operating state of the machine [1].
After a description of the bench, we present the progress of the test over one year. The data collected consists of a representative set of vibration signals corresponding to different operating environments (speed, temperature, etc.).
Then we present the experimental study using deep auto-encoding networks and we compare with usual methods. The principle of the chosen approach is to train the autoencoder with the healthy data, so that it should learn to reconstruct only this type of data. When a new sample of data is supplied to the network, the reconstruction error is calculated and the objective to try to achieve is to obtain a low error for the healthy data and an error which begins to increase as the defect develops. We conclude our paper with the work perspectives.

1 The benchmark

The benchmark consists of a spur gear driven by a motor-gear and followed by a generator. This benchmark simulates a wind turbine. The general view of this benchmark is presented by Figure 1 and the conspectus by Figure 2. This benchmark is used in the accelerated lifetime test by reducing the width of the output wheel of the gear. With the reduction of the tooth width (from 30 to 10mm), the theoretical calculation (lifetime estimation of 480 h) was carried out for a pinion torque of 16 Nm and a constant speed of 1530 rpm at the output of the High-Speed Stage (generator side).
The gear was instrumented by radial and axial accelerometers and by a key phasor (Top-tour IFT200) giving an impulse every round of the high-speed shaft. Three Current sensors were also used from the electrical cabinet of the generator, see
Figure 3.

Figure 1: the general view of the benchmark

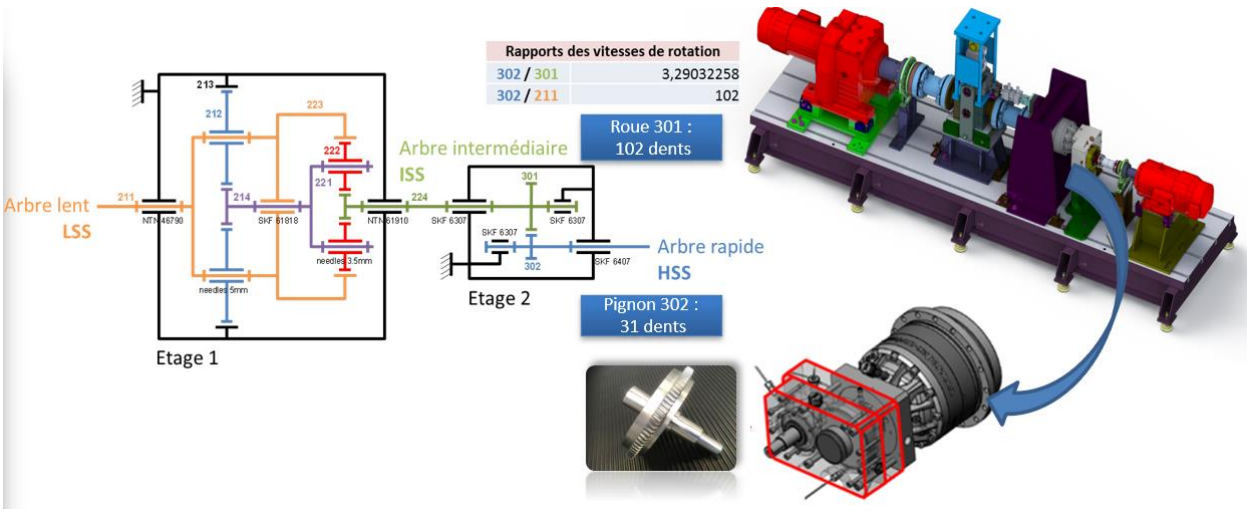

Figure 2: the conspectus of the benchmark

Figure 3: The benchmark instrumentation

The endurance test began on 11 February 2021 at constant speed of 1515 RPM and a constant load of 16Nm then the load was increased to 21 N.m with periodic return to 16 N.m. the lifetime test ended the eight of June 2022 with 6873 hours of operation.

1.1 The data

The data analysed correspond to the spectra of the vibration measurements ranging from zero 2 kHz. During the duration of the endurance, we retained only the signals with a speed and torque corresponding to the nominal speed of 1515 RPM and a torque of 16 +/-0.2 N.m. the number of signals meeting this rule is 72988 Signals. The frequencies calculated with a step of 1/10 Hz, we then obtain a set of 72988 signals with 20000 frequencies.
We note
x_ij: the vibration measurement of signal i for frequency j or the amplitude of the frequency j in the vibration signal i.
x_i=(x_i1,…,x_ip) the signal i spectrum or the frequency vector of signal i.
X=(x_ij) The data set of the vibration measurements.

The 72988 signals are cut as follows:

- a learning set of 27410 spectra measured from the 11 of February 2021 till 30 September 2021 and
- a monitoring set of 45,578 spectra measured from October 1, 2021 to June 8, 2022.

In order to reduce the dimensions of the data we have grouped the frequencies by 4 then by 40 thus giving a second set called G4 and a third set called G40, see the following table.

	Data set Name	Learning	Supervision
Initial data set	G1	27410× 20000	45578× 20000
Frequencies gathered by 4	G4	27410× 5000	45578× 5000
frequencies gathered by 40	G40	27410× 500	45578× 500

Table 1: the different data sets

In the rest of this document the analyzes presented will be those of the two sets G4 and G40

2 Data processing

The measured data in the learning data set is considered coming from flawless system (without any defects) and, remember, that the objective is to represent this data in a parsimonious way so that the reconstruction of this data is the best possible, hoping that the construction of faulty data (data coming system with mechanical defects) is less good. The principle of the chosen approach is therefore to train the auto-encoder with the learning set, so that it should learn to reconstruct only this type of data. When a new sample of data is supplied to the network, the reconstruction error will be calculated and the objective to be achieved is to obtain a low error for the healthy data and an error which increases as the fault progresses.

Having a sample of healthy signals, x1,…,xn, where each xi is a vector of dimension p, the objective is therefore to determine a compression function f and a reconstruction function g of R^p in R^q and of R^q in R^p where p is the initial dimension of the data and q is the dimension of the compressed data such that g∘f is close to the identity function. In another words, if we note y=f(x) the compressed signal, we would like the reconstructed signal g(y) to be close to the initial signal x.

To achieve this objective, two tools were used: The Principal Component Analysis and "deep autoencoders" based on deep neural networks.

2.1 The PCA method

Beyond its use to visualize data, principal component analysis (PCA) can be used to compress and reconstruct initial data. Indeed, if we note

$X^c = (x_{ij}^c)$ the matrix (n, p) of the column-centered initial data defined by

$$X^c = X - \mathbb{1}.\mathbf{g} \quad \textbf{(1)}$$

Which can also be written:

$$x_{ij}^c = x_{ij} - g_j \qquad \forall i, j \quad (2)$$

with $\mathbf{g} = (g_j)$ and $g_j = \sum_i x_{ij}$ is the medium spectrum and

U is the matric of dimension (p, p) of the eigenvectors of the PCA,

the principal component matrix is written:

$$C = X^c.U \quad (3)$$

and is related to the initial data by the formula

$$X^c = C.U^t \qquad \text{and} \qquad X = C.U^t + \mathbb{1}.\mathbf{g}. \quad (4)$$

If we reduce to the matrix $\hat{C} = X^c.\hat{U}$. with dimension (p, q) of the first q eigenvalues, we obtain the first q principals components, which grouped in the matrix $\hat{C}$ of dimension (n, q), checking

$$\hat{C} = X^c.\hat{U}. \quad (5)$$

Then using the reconstruction

$$\hat{X}^c = \hat{C}\hat{U}^t \quad (6)$$

And then

$$\hat{X} = \hat{C}.\hat{U}^t + \mathbf{g}, \quad (7)$$

one can show that we thus obtain a linear approximation of dimension q minimizing the criterion of the least squares

$$||\hat{X}^c - X^c||^2 = ||\hat{X} - X||^2 = \sum_{i,j} (\hat{x}_{i,j} - x_{i,j})^2. \quad (8)$$

We thus have a compression function $\mathbf{y} = f(\mathbf{x})$ defined by the relation

$$\mathbf{y} = \mathbf{x}^c.\hat{U} \quad (9)$$

and a reconstruction function $\hat{x} = g(\mathbf{y})$ matrix defined by the relation

$$\hat{x} = \mathbf{y}\hat{U}^c + \mathbf{g} \quad (10)$$

which provide the optimal linear reconstruction in the sense of least squares.

2.2 The autoencoders

2.2.1 Definition

An autoencoder is a non-supervised artificial neural network in which the input layer has the same number of neurons as the output layer. The interest is to learn how to reduce the dimension of a data set. In practice, an auto-encoder breaks down into two parts.

• The first part is the encoder. The encoder will make it possible to condense the information initially available (image, text, audio, etc.) by extracting characteristics that best define the initial information. The vector that results from the encoder is much smaller in size than the initial vector.

•The decoder is the second part of an auto-encoder. It is responsible for reconstructing the initial information, from the condensed vector. For example, when working with images, when well trained, an autoencoder can take an input image, condense it into a small-sized vector through the encoder, and then recreate it only through to this small vector, via the decoder.

Many examples of auto-encoder networks have been developed. We can cite, for example, the following examples:

The denoising auto-encoders [2] are auto-encoders where the input data is processed through a random noise filter rendering, for example, a grainy image. The output is always compared to the input, so the network learns to ignore some of the detailed features that are irrelevant.

The deep auto-encoders (DAE) or stacked auto-encoders (SAE) are deep auto-encoder models. The architecture of an "SAE" is built by stacking several auto-encoders to form a deep model with many layers. The learning often proposed in this situation, of the greedy type, is done step by step without questioning [3]

The autoencoder parameters are the parameters of the functions used in each layer of the autoencoder.

Once the input data is encoded, it is present in a new form in a particular space called latent space. The latent space corresponds to a new representation of our data. In this new representation, only the most important information contained in the training data is condensed, while filtering the noise (feature extraction).

To train an autoencoder, i.e. to determine the parameters of the autoencoder, it is provided with input data that it must be able to encode in a space of a fixed dimension, then to decode it from the resulting encoding. During training we expect the model to give as output the same input.

2.2.2 Tools

To implement these autoencoders, we select the PyTorch package and use the graphics processors (GPU) of the Cassio and cassio2 servers.

2.2.3 The autoencoder Hyper-parameters

As with the neural network in general, the stochastic gradient descent algorithm is often used to train an autoencoder. Several hyper-parameters must be taken into account to train an autoencoder:

The number of hidden layers: in our experiments, three types of networks were tested: 1 hidden layer, 3 hidden layers and finally 5 hidden layers.

The number of neurons per layer decreases in the encoder and increases in the decoder (symmetry with respect to the code layer of the encoder and the decoder). The number of neurons in the different layers will depend on the tests and will depend on the dimension of the latent space (1, 5 or 10) which correspond to the number of the retained variables.

The Loss function: it is the function that the autoencoder will seek to minimize. The latter is generally the root mean square error in the case of continuous data, which is our case. This error is defined as follows:

$$\mathrm{MSE} = \frac{\sum_{i,j} (\hat{x}_{ij} - x_{ij})^2}{n \times p} \qquad (11)$$

The learning rate: this rate sets the level of modification of the parameters of the autoencoder at each step of the gradient descent. Three learning rates were used: 10^{-3}, $5\ 10^{-4}$ et 10^{-4}.

Batch size: this size corresponds to the number of samples taken into account at each step of updating the parameters of the gradient method.

The number of epochs (epochs): this is the number of times the network is trained with all the data. The number of epochs retained was 1000.

2.2.4 Link with the PCA

The ACP can be seen as a linear autoencoder with only 1 hidden layer and without an activation function. The quality of spatial representation E_k provided by the first k principal components is measured by the inertia fraction Q_k which varies between 0 and 1. We can notice that the maximization of the criterion Q_k of the PCA is equivalent to the minimization of the mean square error criterion of the autoencoders.

3 Results

3.1 The learning phase.

3.1.1 PCA and the autoencoders reconstruction method

The reconstruction quality obtained by the first 10 axes is about 90% for the G40 data set and about 70% for the G4 data set, see figure below.

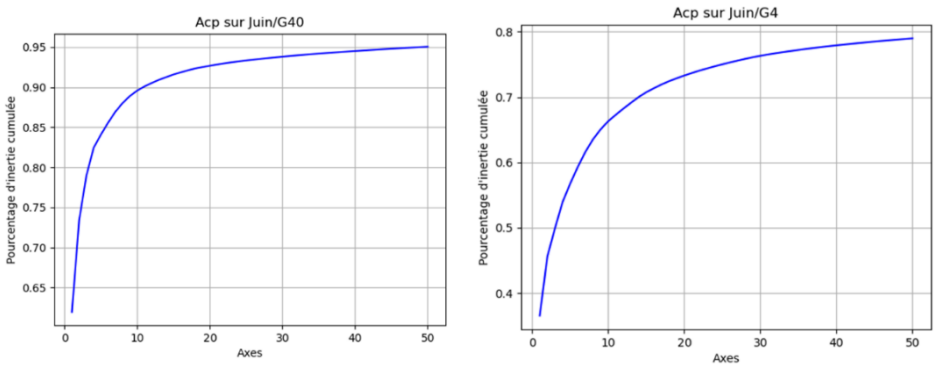

Figure 4: The PCA spaces quality

In the following table, we have reported the values of the reconstruction error of the different data sets. We then obtain the following results (after multiplying the values by10^8) :

Data set	Retained variables	PCA	Autoencoder 1 layer	Autoencoder 3 layers	Autoencoder 5 layers
G40	1	7.47	20.10	20.04	19.99
	5	3.11	7.81	3.08	20.06
	10	2.04	3.062	3.43	2.34
G4	1	32.47	52.04	52.07	52.05
	5	22.13	51.42	52.07	22.23
	10	17.24	33.42	26.92	20.63

Table 2: the reconstruction error of the different data sets in the learning phase

We can notice that the best results, except for one situation, are those obtained by PCA.
We also notice that the error of reconstruction decreases when the number of the retained variable increases.

3.2 The supervision phase.

In order to study the evolution of the data after the learning phase, we chose to measure the root mean square difference between the reconstituted spectra and the initial spectra.

$$\text{eqm}_i = \frac{1}{p}\sum_j (\hat{x}_{ij} - x_{ij})^2 = \frac{1}{p}\sum_j (\hat{x}_{ij}^c - x_{ij}^c)^2 \quad (12)$$

In order to facilitate the reading of these results, we used the following normalized standard deviation.

$$e_i = \frac{1}{\sigma}\sqrt{\text{eqm}_i} \quad (13)$$

where σ is the standard deviation of the learning data.
Furthermore, we reported these discrepancies for both the training data and the supervision data. In all these figures, the border between these two types of data is indicated by a vertical red line. The two red longitudinal lines are the upper control limit and the lower control limit defined as the mean value +/- 3* standard deviation of the e_i values in the learning period. The alert should turn on when data in the supervision phase exceeds one of these two control limits.

For all the graphs of the figure below, the normalized error evolution during the learning and the monitoring data, shows a small drift from the month of October 2021 to the month of February 2022 then a clearer drift from the month of April 2022 till June 2022.

we can also notice that if we increase the dimension of the PCA space (increase the number of principal components retained) then the decision model will be more sensitive to variations and will alert more and more in a precocious manner. The same observation is observed by increasing the number of variables retained. In another terms, and as noticed in the learning phase, the model with 10 retained variables is more sensitive than that of 5 variables and the latter is more sensitive than the model with a single retained variable.

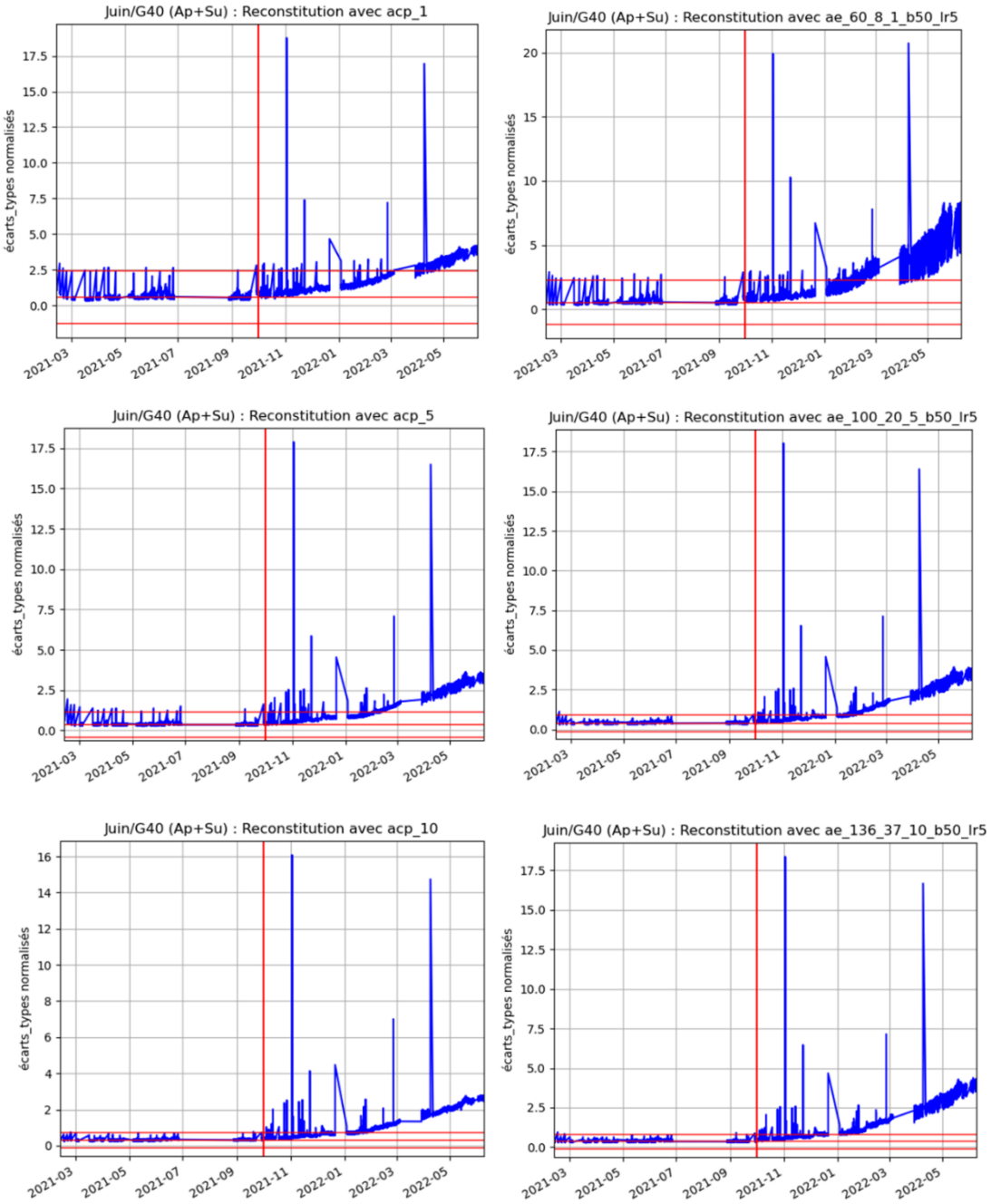

Figure 5: Results of computing the normalized standard deviation in the learning and supervision phases using the G40 data set and for different PCA and autoencoder types.

4 Gear inspection

After gear removal and microscopic examinations, we notice an incipient pitting on the active sides of the pinion and a small seizing area on two teeth of the gear wheel, see figure below.

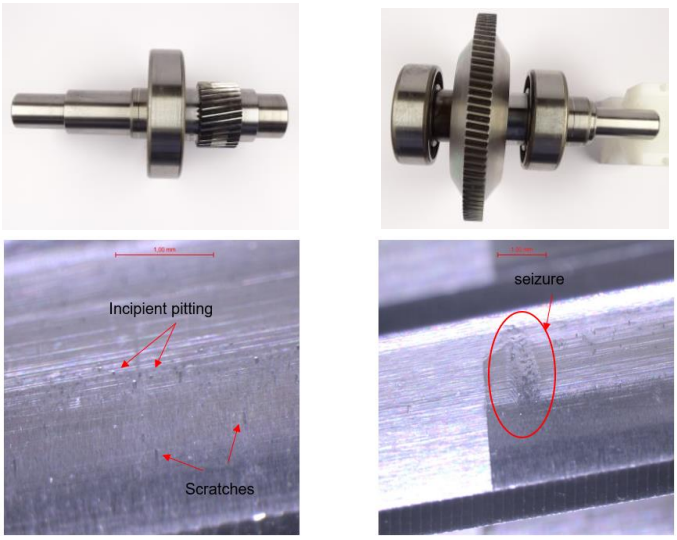

Figure 6: Inspection of the gear components

5 Conclusion

In this paper we have presented the follow-up of an endurance test on a gear. This test was monitored by an intelligent system comprising vibration sensors as well as a "machine learning" method for the automatic detection of deviation from the normal operating state of the bench.
Despite the difficulty encountered in revealing pitting and fatigue wear, the monitoring system was quite sensitive to changes in gear condition. Few but sufficient alerts were issued to trigger an inspection and to notice incipient pitting and seizing.
However, the interest expressed by industrials leads us to propose continuing this study with another endurance test by reducing the duration of the tests through the following:
1- Reduce the width of the wheel
2- Possibly modify the materials of the wheel
3- Increase the capacity of the endurance bench by replacing the epicyclic reduction gear with another that can withstand much greater torques.
It is also planned to use a more elaborate monitoring system developed by CETIM comprising advanced sensor signal processing functions as well as on-board ‘Edge’ decision-making methods.

References

[1] G. Gouvaert & all, *Suivi des études en intelligence artificielle 2021*, Projet 267224, Cetim, 2012 April.
[2] P. Vincent & all, *Extracting and composing robust features with denoising autoencoders*, Proceedings of the 25th international conference on Machine learning, 2008, pp 1096–1103
[3] M. Heydarzadeh & all, *In-bed posture classification using deep autoencoders*, 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 2016, pp.3839-3842.