WIND TURBINE GEARBOX CONDITION MONITORING USING VIBRATION DATA AND MEL-FREQUENCY CEPSTRAL COEFFICIENTS

CRISTIAN VELANDIA-CARDENAS[*], YOLANDA VIDAL[*†] AND FRANCESC POZO[*†]

[*] Control, Data and Artificial Intelligence (CoDAlab)
Department of Mathematics, Escola d'Enginyeria de Barcelona Est (EEBE)
Campus Diagonal-Besòs (CDB), Universitat Politécnica de Catalunya (UPC)
Eduard Maristany 16, 08019 Barcelona, Spain
e-mail: cristian.velandia@upc.edu, yolanda.vidal@upc.edu, francesc.pozo@upc.edu - Web page: https://codalab.upc.edu/en

[†]Institute of Mathematics (IMTech)
Universitat Politècnica de Catalunya (UPC)
Pau Gargallo 14, 08028 Barcelona, Spain
e-mail: yolanda.vidal@upc.edu, francesc.pozo@upc.edu - Web page: https://imtech.upc.edu/en

Key words: Wind Turbine, Gearbox, Vibration, Early Fault Detection, Neural Network, Autoencoder, Mel-Frequency Cepstral Coefficients (MFCCs)

Abstract. A new methodology for wind turbine condition monitoring that utilizes vibration data and Mel-frequency cepstral coefficients (MFCCs) is being developed to accurately assess the condition of the wind turbine gearbox. Traditional condition monitoring techniques rely on physical inspections, which can be time-consuming and labor-intensive. This new approach offers a more efficient and cost-effective solution. The use of vibration data allows for the identification of subtle changes in a wind turbine's operating condition, providing early warning signs of potential issues. The MFCCs are derived from the vibration data signals and provide a compact representation of the information, allowing for more efficient analysis. These coefficients are used to create a fingerprint of the wind turbine's operating condition, which can then be compared to known healthy operating conditions to identify any deviations or anomalies. This new methodology has been shown to be highly effective in detecting potential issues with wind turbine components such as gearbox failure, and drive train bearing degradation. By providing early warning signs, wind farm operators can take action to address the issue before it leads to significant downtime or damage. In a nutshell, the use of vibration data and MFCCs offers a promising new approach for wind turbine condition monitoring. The proposed approach has been tested on the EISLAB (Lulea, Sweden) dataset concerning the vibration signals from wind turbines of the same type in northern Sweden.

1 INTRODUCTION

According to the Global Wind Energy Council (GWEC), in 2022, the wind energy sector continued to grow, reaching a new installed capacity of 78 GW worldwide, thus reporting its third-best year, behind the records set in 2021 and 2020, despite COVID-19. These new facilities provide an accumulated wind energy capacity of 906 GW in the world, showing a year-over-year growth of 9% compared to the previous period ([1, 2]). According to [3] approximately 57% of the failures and 65% of the inactivity time of the turbines are due to the drivetrain, empowering this work to focus on the condition monitoring (CM) of wind turbines (WTs) gearboxes. The gearbox is one of the most complex subsystems of WTs due to its complicated structure and harsh operating environment ([4]). It is composed of several gears, bearings, and shafts in contrast to other simpler subsystems such as the main bearing or main shafts, which are also well-studied in the literature.

The CM of wind turbines is a fairly broad and studied area, in which a wide variety of techniques and methodologies have been used. Due to the complexity of turbine systems, some researchers tend to analyze its subsystems separately. The combinations between subsystems, algorithms, models, and data sources, allow generating a diversity of methodologies and techniques for the CM of WTs. An extensive selection of data can be extracted to assess this subsystem, including SCADA system data, vibration data, temperatures, stresses, among others. Different studies demonstrate the effectiveness of the reuse of SCADA data to make the gearbox CM. For example, [5] use SCADA system data, processed with signal processing techniques, to feed classic machine learning algorithms and thus be able to detect failures. Considering the fact that SCADA data tends to be aggregated, due to storage and other limitations, and that its minimum agregation window is 10 minutes, a huge loss of information occurs. This was proven by [6], where they compare the efficiency of the 10-minute averaged values with the 100 Hz sampling frequency without averaging and conclude that the use of high-frequency data leads to improved prognostic predictions and more insights about the condition of the wind turbine components, contributing to earlier fault detection. This remarks the importance of the use of high-frequency data, such as vibrations. For example, [7] use vibration signals, captured by an accelerometer, to train dictionaries in an unsupervised way and detect anomalies caused by the gearbox.

Vibration data and signal analysis techniques are widely used also, [8] and [9] use wavelet-based transformations to characterize vibration signals and detect failures. [10] uses time and frequency analysis of vibration signals to characterize failures and [11] use frequency analysis (spectrograms) to characterize signals and detect gearbox anomalies, among other signal analysis techniques for the CM of the gearbox.

The deep learning algorithms are also used for the WT gearbox CM as a fault detection method. Among them, one of the most used ones for the CM are the convolutional networks (CNN). For example, [12] uses a temporal convolutional network (TCNN) to classify failures found in SCADA data, while [13] uses a TCNN together with a graphical attention network (GAT) to extract spatio-temporal features from SCADA data, revealing the flexibility and applicability of these architectures. There are a set of less used architectures as Deep Capsule Neural Networks (CapsNet) ([14]), graph-based networks ([13, 15, 16]) and Autoencoders ([17, 18]), that create an opportunity to keep researching and experiment with them combied with high-frequency data. For this reason, this work uses an autoencoder and a convolutional autoencoder fed with features extracted by calculating the Mel-frequency cepstral coefficients (MFCCs) of publicly available vibration data. The data was recorded by [19] from the University of Luleå in 2018, of wind turbines located in Sweden, to create an unsupervised anomaly detector for early detection of faults in the gearboxes of the

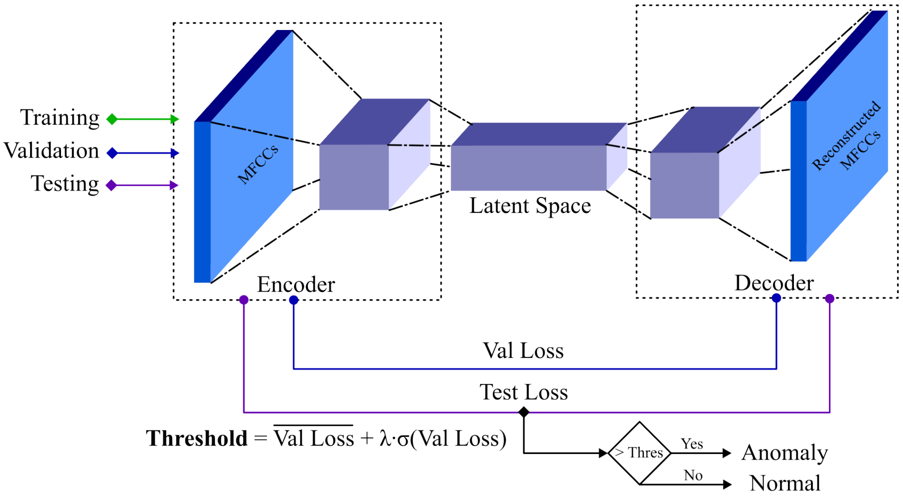


Figure 1: Graphical description of the Condition Monitoring Framework

WT. Those signals were captured from six turbines in the axial direction of an accelerometer mounted on the housing of the bearing of each output shaft. The dataset can be found at `http://www.diva-portal.org/smash/record.jsf?dswid=2475&pid=diva2%3A1244889`.

The article is structured as follows, in Section 1 the context and state of the art of the CM of WTs gearboxes is shown. Section 2 describes the condition monitoring framework, the dataset, the data processing techniques, the feature extraction process, and the algorithm to be used. Then, in Section 3, results are shown and discussed around its capability for the condition of the gearbox. Finally, Section 4 depicts the conclusions of the work and the future steps.

2 MATERIAL AND METHODS

The experiments shown in this section utilize a consumer-grade laptop hardware, demonstrating that the proposed methodology does not require extensive computing resources. By leveraging accessible hardware, this methodology reduces the barriers to exploring AI solutions for the advancement of renewable energy. The following sections highlight efficient training on limited VRAM, enabled by data preparation and model configuration.

2.1 Condition Monitoring Framework

The condition monitoring framework proposed in this paper consists of generating an unsupervised normal (healthy) condition model for each turbine. The model is trained using healthy data and is then tested using data with known anomalies. The anomaly detection process is based on the prediction loss of the validation data, which allows to establish a loss threshold between the input and the reconstructed output of the model. When a signal that was recorded under an abnormal state is fed into the model, it is not able to fully reconstruct the input, therefore, it returns a loss value higher than the threshold indicating that there is an anomaly.

This framework is represented in Figure 1 and explained step by step as follows.

- **Step 1.** Apply data augmentation and data cleaning techniques to optimize data for the training process.

- **Step 2.** Extract features from the data while reducing the dimensionality and capturing the most important information using MFCCs.
- **Step 3.** Train the selected model under different operating conditions, but only with healthy data.
- **Step 4.** Calculate the loss threshold of the normality model (healthy) that is used to classify each testing signal into normal or anomaly by comparing the input and output of the reconstruction algorithm. The threshold is calculated using

  $$\text{threshold} = \overline{\text{loss}_{\text{val}}} + \lambda \cdot \sigma\left(\text{loss}_{\text{val}}\right) \tag{1}$$

  where $\overline{\text{loss}_{\text{val}}}$ corresponds to the mean loss, $\sigma\left(\text{loss}_{\text{val}}\right)$ is the standard deviation of the loss, and λ is the tuning factor to minimize the FPR, commonly used in the literature as 3, but tuned on each turbine model. Therefore, in order to minimize the FPR, the tuning factor λ, must be increased from the base value $\lambda = 3$ until a value that reduces the FPR to the minimal value accepted.

  It is worth recalling that this is calculated over a normality model, this means that it is trained *only* on healthy data, and so validated only with healthy data. This threshold is not calculated using data with faults, as it is assumed that these data are scarce or not available when developing the methodology, thus faulty data is only used for testing.

- Step 5. Using the threshold value that minimizes the FPR at the validation process, it is possible to detect anomalies in the test data by:

  $$\text{anomaly} = \begin{cases} 1 & \text{if } \text{loss}_{\text{test}} > \text{threshold}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

2.2 Data set Description and Processing

The selected data set is made up of vibration (acceleration) data collected by the EISLAB research group of the University of Luleå in Sweden. By means of an accelerometer mounted on the gearbox output shaft cover, they record the data measured in its axial direction.

The data set is composed of, on average, $2,700$ signals recorded with a sampling rate of 12.8 kilo samples/s per turbine and for 1.28 seconds or $16,384$ data points each. The time interval between the recordings is approximately 12 hours from each other for a period of 46 months. They also recorded the speed, and time variables to keep track of the signals. From now on, the dataset is represented as D, which is of the form:

$$D_X = (d_{ij})_{\substack{i=1,\ldots,N\\ j=1,\ldots,P}} = \begin{array}{c} \begin{array}{ccccc} \boldsymbol{x_1} & \cdots & \boldsymbol{x_j} & \cdots & \boldsymbol{x_P} \end{array} \\ \begin{pmatrix} d_{1,1} & \cdots & d_{1,j} & \cdots & d_{1,P} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{i,1} & \cdots & d_{i,j} & \cdots & d_{i,P} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{N,1} & \cdots & d_{N,j} & \cdots & d_{N,P} \end{pmatrix} \end{array} \in \mathcal{M}_{N\times P}(\mathbb{R}), \tag{3}$$

where $i = 1, \ldots, N$ and $j = 1, \ldots, P$. N are the number of signals recorded for the WT X, and P the number of columns in the dataset, composed of a time (years) column, speed (RPM) column, and $16,384$ more columns corresponding to each captured point of the vibration signal.

It is known that three of the turbines remained healthy (they did not present any type of failure or fault) throughout the analyzed period, two that had some electrical faults, one turbine that had several faults, and a damaged gearbox that was replaced at a particular time during the analysis frame. For this work, healthy turbines (3 and 4) are used to validate the model, and the turbine with damage, turbine number 5, is used for unsupervised anomaly detection. The faults that affected the turbine 5 are presented below:

- At time 1.2 years, the bearing of section 2 of the gearbox was replaced due to faults and wear.
- Approximately in year 2 the gearbox was replaced.

There is no deeper description of the faults, but it is enough to know that they occurred at an approximate time to try to detect them using an unsupervised deep learning technique. The known healthy signal number is very low, and not enough to train a deep-learning algorithm. Therefore, there is a need for data augmentation. Thus, a data augmentation technique is proposed in Section 2.2.1.

2.2.1 Sliding Window Data Augmentation

The data augmentation process consists of using the available data to create more signals for training by shifting a window over each recorded signal every δ sampling periods. The window must have a fixed length, called window size. From now on, this method is referred to as the sliding window method. The visual representation of the proposed method is shown in Figure 2 and deeply explained as follows.

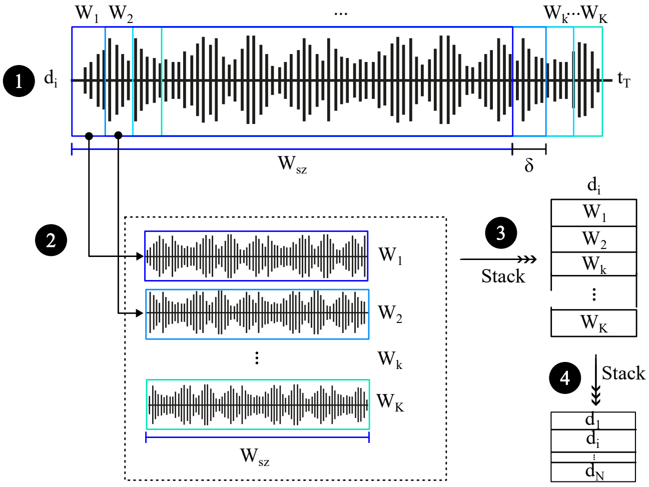


Figure 2: Graphical description of the sliding window method for data augmentation.

- **Step 1.** Each signal $d_i,\ i = 1, \ldots, N_X$, for each turbine X, of the dataset described in Equation (3) is divided into small pieces of data called windows $W_k,\ k = 1, \ldots, K$. The number of data points captured by each window is called the window size W_{sz}. It is defined by the sampling rate of the signal (S_r) and the amount of time selected for the window (W_t). In this case, the sampling rate is 12.8 kilosamples/s and a window of 1 second is defined in order to slide it over the remaining 0.28 seconds of the signal. This results in $W_{sz} = 12,800$ points captured per window.

- **Step 2.** All information within the window $W_k,\ k = 1, \ldots, K$, of size W_{sz} is captured and stored as a new signal. The new signals keep the timestamp and speed of the root signal to avoid disturbing the time dependency and keep track of the generated samples.

- **Step 3.** Each new signal is stacked to create an augmented matrix of dimensions $[K, W_{sz} + 2]$, where K is the number of windows that can be generated by sliding the window through the signal. The number of windows generated depends on the sampling rate of the signal (S_r), the total length of the signal in seconds (t_T), the window length in seconds (W_t) and the amount of sampling periods to hop, or as it is usually denoted, the hop size (δ). The "+2" is due to the two extra columns of speed and time. Thus, K is defined as

$$K = \frac{(t_T[s] - W_t[\mathrm{s}]) \cdot S_r \left[\frac{\mathrm{samples}}{\mathrm{s}}\right]}{\delta}. \tag{4}$$

  Considering that the signal length is 1.28 seconds, with a window of 1 second, and $\delta = 2$, to slide the window every two steps or every two sampling periods $(\Delta = \delta \cdot T_s = 156.25\mu s)$ through the remaining time of the signal, $0.28s$, it is possible to generate $K = 1,792$ windows or new training samples.

- **Step 4.** Each augmented matrix d_i with dimensions $[K, W_{sz} + 2]$ is stacked to create a new augmented data set with dimensions $[N \times K, W_{sz}]$.

Recall that this is not a synthetic oversampling method, as it is not generating synthetic data, but creating new samples based on each real signal.

2.3 Feature Extraction (MFCCs)

Features are extracted from vibration signals using the Mel frequency cepstral coefficients (MFCCs), this serves a pair of purposes, first, reduce the dimension of the signal and, second, extract the time and frequency characteristics of the signal to help the deep learning algorithm extract the most of the normal operating conditions of the turbine. The MFCCs are commonly used for speech and audio characterization and have been deeply studied and used to train machine learning (ML) models for audio classification or speech information extraction [20]. To calculate them, the following steps must be followed:

1. Split the signal into fixed-width overlapping windows.

2. Apply the time-frequency transform to each of the extracted frames and apply the DFT to calculate the periodogram (it is an estimate of the spectral density of a signal).

3. Compute a Mel-spaced filter bank for each of the frames. The filter bank is a series of 20-40 triangular filters to get a filtered spectrogram. Before applying the filters, the frames must be transformed into Mel-scale using Equation (5). Then, apply the log to the resulting spectrogram.

$$M(f) = 2,595.0 \times \log\left(1.0 + \frac{f}{700.0}\right). \tag{5}$$

4. The spectrogram of the previous step contains coefficients highly correlated, to avoid that the DCT can be applied at this point to decorrelate them and obtain the MFCCs which are very efficient for computation.

Finally, a data scaling process is applied to the resulting MFCCs. Data scaling methods are used to improve the training of the algorithm by transforming all the values into a lower magnitude scale to avoid exploding gradients and bias. The selected scaling method is known as min-max scaling, and can be used by applying the following Equation (6):

$$D'(x) = a + \frac{(D(x) - \min D(x))(b - a)}{\max D(x) - \min D(x)}. \tag{6}$$

The data is scaled between 0 and 1, thus, the scaling factors are $a = 0$ and $b = 1$. Once the dataset is rescaled, it is ready for training. To calculate the MFCCs the library Librosa ([21]) written in Python programming language is used.

2.4 Model Selection

The model selected for the framework described in Section 2.1, is the autoencoder. It is an unsupervised learning technique based on neural networks. It is commonly trained to learn patterns of input data (encode) and reconstruct (decode) them at the output, suppressing its noise. Two architectures of autoencoders were selected for training, namely the 1D autoencoder and the 2D version, called convolutional autoencoder. For both of them, an image-denoising design is selected, and the metric selected to measure their performance is the mean squared error (MSE). It is the most common metric used to assess the performance of this kind of algorithm. It is also commonly used to calculate the threshold needed for the anomaly detection process. Due to the size of the VRAM of the available computer, both algorithms are trained using a data sequencer.

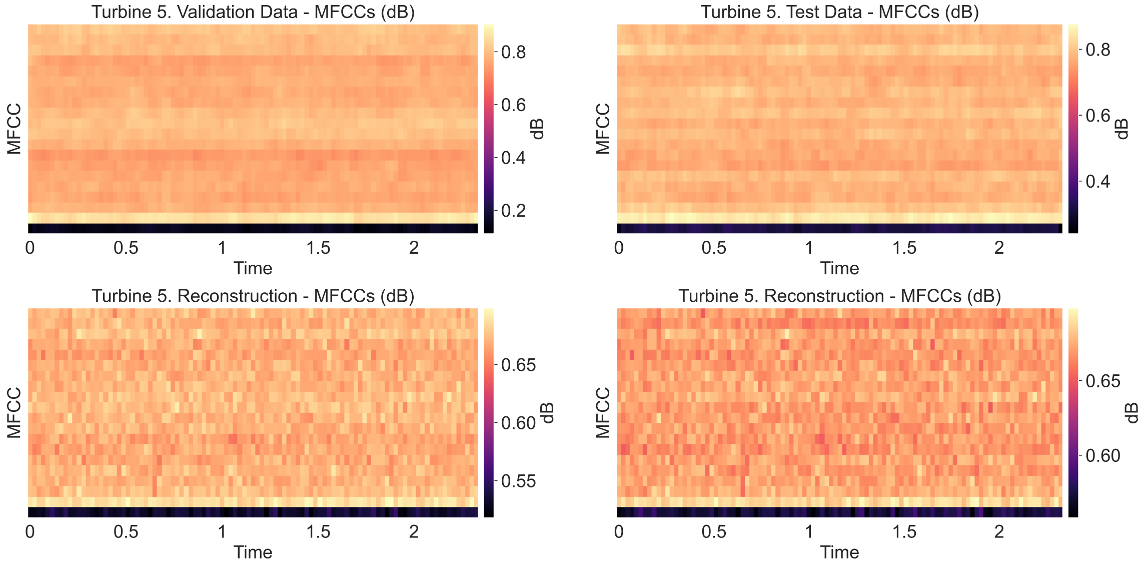
3 Results and Discussion

Using the method described in Subsection 2.1, the trained models are able to learn the particular characteristics of MFCCs to create a normality model of the turbine. The first model trained was the 1D autoencoder.

The experiments carried out revealed that when training a more complex model with more parameters, the model stabilized quickly and stopped early, resulting in a low loss (as seen in Table 1, experiment 1). On the other hand, when training a less complex model with half of the parameters, it achieved a lower loss (Table 1, experiment 3) than experiment 1, but not as low as the middle model (Table 1, experiment 2). All experiments stopped early, stabilizing on the loss value. The best-performing model was that of experiment 2, which achieved a minimum of $1.70 \cdot 10^{-4}$ validation MSE.

Table 1: Training performance - 1D Autoencoder (Turbine 5)

Experiment	H1, H2, H3	Lr	Time	Parameters	Loss	Val Loss	Epochs
1	64, 32, 16	10^{-3}	14m31.7s	265,908	$2.76{\cdot}10^{-4}$	$2.32{\cdot}10^{-4}$	8
2	64, 32, 8	10^{-3}	32m34.5s	265,388	$2.02{\cdot}10^{-4}$	$1.70{\cdot}10^{-4}$	18
3	32, 16, 8	10^{-3}	14m22.8s	132,684	$2.55{\cdot}10^{-4}$	$2.33{\cdot}10^{-4}$	8



(a)

(b)

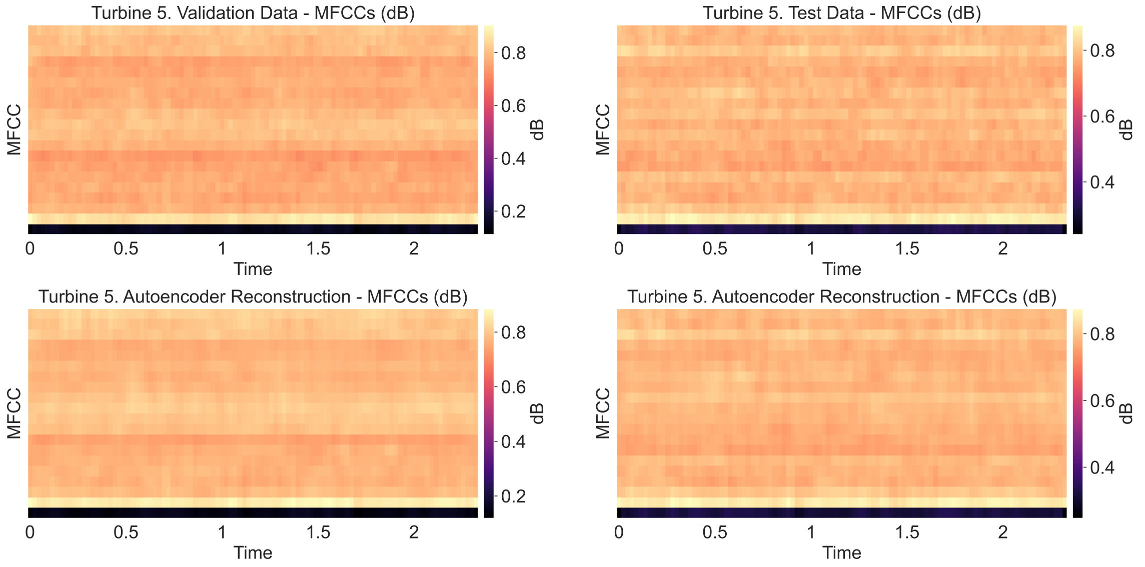
Figure 3: Reconstructed Turbine 5 MFCCs matrix by the trained 1D autoencoder - (a) Validation, (b) Testing Data Example

To show the reconstruction capability of the 1D autoencoder, the MFCCs of one of the signals are plotted, comparing the input and the reconstructed output of the model of the experiment number 2 (see Figure 3). It is evident that, despite being the most successful model, it is unable to accurately replicate the input data. Consequently, an alternative or different model that can truly comprehend all the features of the signal is necessary.

That alternative is the convolutional autoencoder. Its results show a better generalization and a better reconstruction of the input data.

Table 2: Training performance - Convolutional Autoencoder (Turbine 5)

Experiment	Lr	Time	Parameters	Loss	Val Loss	Epochs
1	10^{-2}	19m 16.6s	12,193	$1.29{\cdot}10^{-4}$	$1.34{\cdot}10^{-4}$	10
2	10^{-3}	19m 16.5s	12,193	$8.74{\cdot}10^{-5}$	$7.96{\cdot}10^{-5}$	10
3	10^{-4}	28m 32.2s	12,193	$7.58{\cdot}10^{-5}$	$7.28{\cdot}10^{-5}$	15
4	10^{-3}	28m 50.1s	12,193	$5.94{\cdot}10^{-5}$	$4.36{\cdot}10^{-5}$	15



(a)

(b)

Figure 4: Reconstructed Turbine 5 MFCCs matrix by the trained trained convolutional autoencoder - (a) Validation, (b) Testing Data Example

Using the resulting convolutional autoencoder model of experiment number 4, which obtained the best performance, the same visualization exercise is performed showing an improved recosntruction capability. In addition, it achieves its results in approximately 4 fewer minutes of training and with under 95% of the parameters used by the optimal 1D autoencoder (Table 1).

Using the best-performing model, the convolutional autoencoder model number 4, feeding it with the data with known anomalies and tuning the threshold to minimize false positives ($\lambda = 5$), it is possible to identify several anomalies that resemble the failures depicted by the authors of the dataset described in section 2.2. The final results are shown in Figure 5.

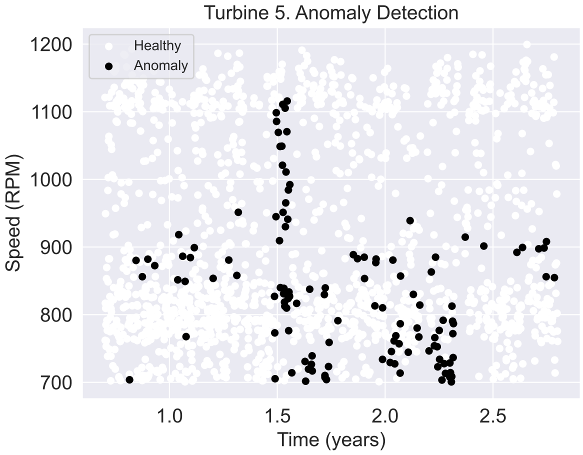
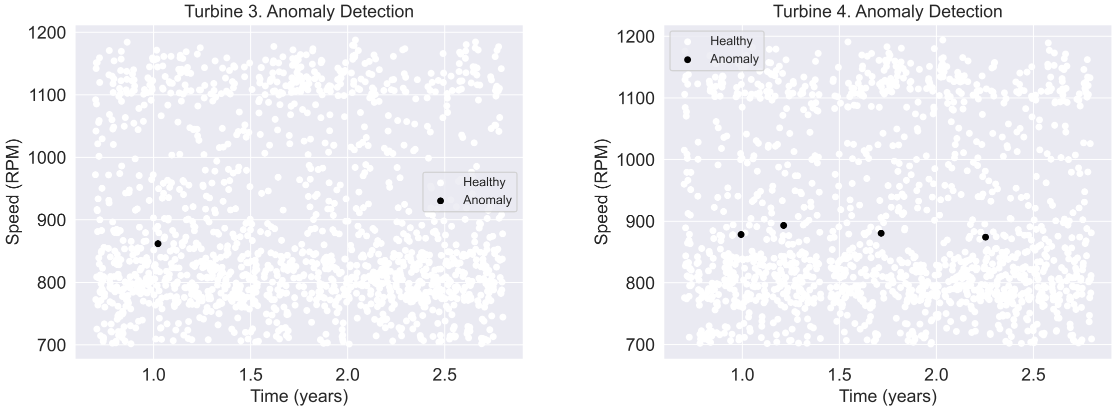


Figure 5: Testing data example for detected anomalies (true alarms) in Turbine 5.

Finally, to validate the proposed framework, a last experiment is performed. It consists of using the data captured from turbines 3 and 4, which were two of the turbines that stayed healthy, and applying the proposed framework to its data. The models are trained and tuned, depicting great results. As expected, the false positives are reduced to a low percentage, showing just a few anomalies (see Figure 6).



(a)

(b)

Figure 6: Testing data with detected anomalies (false alarms) in Turbines 3 (a) and 4 (b).

4 CONCLUSIONS

This paper proposes a wind turbine gearbox condition monitoring technique based on vibration data descriptors, data augmentation, and deep learning. The method augments the existing normal condition vibration data and uses it to calculate the MFCCs as signal descriptors to train a convolutional autoencoder. The model is able to accurately reproduce the signal descriptors passed to it, capturing the normal parameters of the wind turbine and detecting any abnormal conditions in an unsupervised manner. The proposed technique is able to alert about faults before they cause failure, as the degradation of the mechanical parts is progressive. The method was tested on publicly available vibration data from wind turbines in Sweden, and it is completely unsupervised in the sense that only healthy data is needed to develop and validate it, thus greatly extending its range of application in any wind turbine (even when no faulty data is available). It could be further improved by testing it on real data from condition monitoring systems (CMS) of actual wind turbines in operation. Unfortunately, these data are usually proprietary and not easily accessible to researchers. Therefore, it is essential that in the near future asset owners and researchers work together in a mutually beneficial way, where asset owners can benefit from the latest data analytics and researchers can access the necessary asset data to train and validate their models.

Acknowledgments

This work is partially funded by the Spanish Agencia Estatal de Investigación (AEI) - Ministerio de Economía, Industria y Competitividad (MINECO), and the Fondo Europeo de Desarrollo Regional (FEDER) through the research projects PID2021-122132OB-C21 and TED2021-

129512B-I00; and by the Generalitat de Catalunya through the research project 2021-SGR-01044.

REFERENCES

[1] Global Wind Energy Council. *Global Wind Report 2022*. en-US. Apr. 2022. URL: https://gwec.net/global-wind-report-2022/ (visited on 02/21/2023).

[2] Global Wind Energy Council. *Global Wind Report 2023*. en-US. Feb. 2023. URL: https://gwec.net/globalwindreport2023/ (visited on 06/14/2023).

[3] Sebastian Pfaffel, Stefan Faulstich, and Kurt Rohrig. "Performance and Reliability of Wind Turbines: A Review". In: *Energies* 10.11 (2017). ISSN: 1996-1073. DOI: 10.3390/en10111904.

[4] Tianyang Wang et al. "Vibration based condition monitoring and fault diagnosis of wind turbine planetary gearbox: A review". In: *Mechanical Systems and Signal Processing* 126 (July 2019), pp. 662–685. ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2019.02.051. URL: https://www.sciencedirect.com/science/article/pii/S0888327019301426.

[5] Cristian Velandia-Cardenas, Yolanda Vidal, and Francesc Pozo. "Wind Turbine Fault Detection Using Highly Imbalanced Real SCADA Data". In: *Energies* 14.6 (2021). ISSN: 1996-1073. DOI: 10.3390/en14061728.

[6] Ayush Verma et al. "Wind turbine gearbox fault prognosis using high-frequency SCADA data". In: *Journal of Physics: Conference Series* 2265.3 (May 2022), p. 032067. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/2265/3/032067. URL: https://iopscience.iop.org/article/10.1088/1742-6596/2265/3/032067 (visited on 07/12/2023).

[7] Sergio Martin-del-Campo, Fredrik Sandin, and Daniel Strömbergsson. "Dictionary Learning Approach to Monitoring of Wind Turbine Drivetrain Bearings:" en. In: *International Journal of Computational Intelligence Systems* 14.1 (2020), p. 106. ISSN: 1875-6883. DOI: 10.2991/ijcis.d.201105.001. URL: https://www.atlantis-press.com/article/125946150 (visited on 06/15/2023).

[8] Xingjia Yao et al. "Wind Turbine Gearbox Fault Diagnosis Using Adaptive Morlet Wavelet Spectrum". In: *2009 Second International Conference on Intelligent Computation Technology and Automation*. Vol. 2. Journal Abbreviation: 2009 Second International Conference on Intelligent Computation Technology and Automation. Oct. 2009, pp. 580–583. DOI: 10.1109/ICICTA.2009.375.

[9] Qingqing Liu and Jiangtian Yang. "Fault Diagnosis of Wind Turbine Gearbox Based on Dual-tree Complex Wavelet and Information Entropy". In: *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*. Journal Abbreviation: 2018 Prognostics and System Health Management Conference (PHM-Chongqing). Oct. 2018, pp. 1194–1199. ISBN: 978-1-5386-5380-7. DOI: 10.1109/PHM-Chongqing.2018.00210.

[10] Zeyuan Wang, Hongwei Wang, and Wei Liu. "Fault Diagnosis of Wind Turbine Gearbox Based on Vibration Data". In: *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. Journal Abbreviation: 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE). Oct. 2018, pp. 234–238. DOI: 10.1109/ICEBE.2018.00045.

[11] Peter F. Odgaard and Amir R. Nejad. "Frequency based wind turbine gearbox fault detection applied to a 750 kW wind turbine". In: *2014 IEEE Conference on Control Applications (CCA)*. Journal Abbreviation: 2014 IEEE Conference on Control Applications (CCA). Oct. 2014, pp. 1383–1388. ISBN: 978-1-4799-7409-2. DOI: 10.1109/CCA.2014.6981517.

[12] Shahabodin Afrasiabi et al. "Wind Turbine Fault Diagnosis with Generative-Temporal Convolutional Neural Network". In: *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. Journal Abbreviation: 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe). June 2019, pp. 1–5. DOI: 10.1109/EEEIC.2019.8783233.

[13] Jun Zhan et al. "Condition monitoring of wind turbines based on spatial-temporal feature aggregation networks". In: *Renewable Energy* 200 (Nov. 2022), pp. 751–766. ISSN: 0960-1481. DOI: 10.1016/j.renene.2022.09.102. URL: https://www.sciencedirect.com/science/article/pii/S0960148122014616.

[14] Pengfei Liang et al. "A deep capsule neural network with data augmentation generative adversarial networks for single and simultaneous fault diagnosis of wind turbine gearbox". In: *ISA Transactions* (Oct. 2022). ISSN: 0019-0578. DOI: 10.1016/j.isatra.2022.10.008. URL: https://www.sciencedirect.com/science/article/pii/S0019057822005365.

[15] Guoqian Jiang et al. "TempGNN: A Temperature-based Graph Neural Network Model for System-level Monitoring of Wind Turbines with SCADA Data". In: *IEEE Sensors Journal* (2022), pp. 1–1. ISSN: 1558-1748. DOI: 10.1109/JSEN.2022.3213551.

[16] Xiaoxia Yu, Baoping Tang, and Lei Deng. "Fault diagnosis of rotating machinery based on graph weighted reinforcement networks under small samples and strong noise". In: *Mechanical Systems and Signal Processing* 186 (Mar. 2023), p. 109848. ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2022.109848. URL: https://www.sciencedirect.com/science/article/pii/S0888327022009165.

[17] Guoqian Jiang et al. "Stacked Multilevel-Denoising Autoencoders: A New Representation Learning Approach for Wind Turbine Gearbox Fault Diagnosis". In: *IEEE Transactions on Instrumentation and Measurement* 66.9 (Sept. 2017), pp. 2391–2402. ISSN: 1557-9662. DOI: 10.1109/TIM.2017.2698738.

[18] Guoqian Jiang et al. "Wind Turbine Fault Detection Using a Denoising Autoencoder With Temporal Information". In: *IEEE/ASME Transactions on Mechatronics* 23.1 (Feb. 2018), pp. 89–100. ISSN: 1941-014X. DOI: 10.1109/TMECH.2017.2759301.

[19] Sergio Martin del Campo Barraza, Fredrik Sandin, and Daniel Strömbergsson. *Dataset concerning the vibration signals from wind turbines in northern Sweden.* 2018.

[20] Tom Bäckström et al. "Introduction to Speech Processing: 2nd Edition". en. In: (July 2022). Publisher: Zenodo Version Number: v2. DOI: 10.5281/ZENODO.6821775. URL: https://zenodo.org/record/6821775 (visited on 06/17/2023).

[21] Brian McFee et al. *librosa/librosa: 0.10.0.post2*. Mar. 2023. DOI: 10.5281/ZENODO.7746972. URL: https://zenodo.org/record/7746972 (visited on 06/18/2023).