Diagnosis of Failure Modes from Bearing Data via Deep Learning Variational Autoencoder Method

Thais Lucas [a], Lavínia Araújo [a], Diego Aichele [a], Márcio das Chagas Moura [a] and Isis Lins [a]

[a] Universidade Federal de Pernambuco, Recife, Brazil, *thais.lucas@ufpe.br, lavinia.mendes@ufpe.br, diego.aichele@ufpe.br, marcio.cmoura@ufpe.br, isis.lins@ufpe.br*

Abstract: Bearings are indispensable equipment in complex machinery. Many studies developed analyses to improve the effectiveness of predictive maintenance for these components. Thus, Deep Learning (DL) models for diagnosis and prognosis of equipment failure modes can be highlighted. For this purpose, many applications have used supervised learning methods, such as Support Vector Machine, Multilayer Perceptron, and Convolutional neural networks. However, in practice, labelled data connected to the conditions of real-time systems can be more complex and costly to obtain. In this sense, we highlight unsupervised learning models, where the algorithm discovers by itself through data exploration, the possible relationships between data points. Hence, this paper aims to apply the unsupervised Variational Autoencoder method to diagnose failure modes of bearings and gears. The role of data preprocessing is also considered, since the data are subject to Short-time Fourier Transform and Continuous Wavelet Transform and compared to case by using raw data. Two databases available in the literature are used for analyses purposes. Finally, the results will be compared with other methods to validate the models' effectiveness.

1. INTRODUCTION

Higher demands for reliability and operational safety in modern production systems have been driven by technological advancements in machine automation, integration, and precision [1]. Rotating machinery, as a critical piece of mechanical equipment in modern industry, operates in a complicated environment with high temperatures, fatigue, and a large load. Generated failures may result in serious accidents, ensuing in significant financial loss and casualties. Intelligent diagnostics, as a fundamental component of Prognostics Health Management (PHM), is designed to detect failures appropriately for a wide range of rotating machinery, including helicopters, aviation engines, wind turbines, and high-speed trains. Traditional intelligent diagnosis methods include feature extraction using signal processing methods and defect classification adopting Machine Learning (ML) approaches, for which we have seen significant progress [2–4].

Bearing defect detection is effectively served by fault classifiers based on decision trees (DT) [5,6], support vector machines (SVM) [7], k-nearest neighbour (k-NN) [8], convolutional neural networks (CNN) [8], and deep belief networks (DBN) [9]. All of the solutions based on ML that have been suggested require historical, labelled failure data for training, i.e., they are supervised, which is difficult to come by in industry.

On the other hand, the technique of fitting models to unlabelled data is known as unsupervised learning, which could be achieved with a Variational Autoencoder (VAE) that is a type of generative deep learning model. Indeed, a generative model is an unsupervised learning model that can generate new data points that were not present during training. By minimizing the reconstruction error and the Kullback-Leibler Divergence (KLD) between an encoded sample and a Gaussian standard distribution, the VAE performs variational parameter inference using neural networks in an encoder-decoder structure, which is equivalent to maximizing the evidence lower bound (ELBO). Moreover, the "reparameterization trick" can be used to optimize this goal using gradient descent algorithms. The latent variables can be given as a distribution using these generative models [10].

Therefore, this paper aims to apply VAE for bearing failure mode diagnosis. Two databases available in the literature will be used. In addition, three different architectures will be used in the encoder and decoder stages. They are structured from the Multilayer Perceptron (MLP), differentiated by the number of layers. It is worth pointing out that in the classification stage, in all three architectures, the network also consists of an MLP. In addition, we also evaluate the performance of three different input types, namely CWT, STFT and Slice. The results are compared in order to identify the model that best fits the databases.

The remainder of this paper is structured as follows: Section 2 gives a short description of the bearing databases, Case Western Reserve University (CWRU) and Jiangnan University (JNU). Section 3 addresses a theoretical background about VAE, as well as the evaluation metrics. Section 4 describes the architectures of the models and the input types. Session 4 presents the results and discussions. Finally, Section 5 addresses the conclusions.

2. DESCRIPTION OF THE DATABASES

In this paper, two databases were analysed, the CWRU and the JNU. It is important to highlight that they have different difficulty levels, the CWRU being the simplest and the JNU the most complex, as established by Zhao et al. [4]. With that in mind, we can observe the results for different datasets and the model's performance to each one.

2.1. CWRU dataset

The CWRU database consists of vibration data collected from a bearing of an engine in the laboratory. Failures are implanted in the motors from electrical discharge machines with test diameter performed with the engine load starting from 0 to 3 HP [11]. Table 1 shows the different failure modes, the diameters used and the proportion of each class in the dataset. Vibration data are collected at a rate of 12k samples per second from accelerometers connected to the equipment, at two points, in the upper and lower turbine of the device. This test bench was built with the objective of building database from the insertion of the main modes of failures present in bearings, which are in general, critical components of rotating equipment (motors, rotors, generators, compressors, pumps, among others) present in most industries [12]. Similar to [4], we used data collected from the drive end, and the sampling frequency is equivalent to 12kHz at 1797 rpm.

Table 1: CWRU Fault Mode Description

Mode Description	Proportion
Health State: the normal bearing at 1797 rpm and 0 HP	20.31%
Inner ring 1: 0.007 inch inner ring fault at 1797 rpm and 0 HP	10.73%
Inner ring 2: 0.014 inch inner ring fault at 1797 rpm and 0 HP	6.51%
Inner ring 3: 0.021 inch inner ring fault at 1797 rpm and 0 HP	8.04%
Rolling Element 1: 0.007 inch rolling element fault at 1797 rpm and 0 HP	8.04%
Rolling Element 2: 0.014 inch rolling element fault at 1797 rpm and 0 HP	7.28%
Rolling Element 3: 0.021 inch rolling element fault at 1797 rpm and 0 HP	8.42%
Outer ring 1: 0.007 inch outer ring fault at 1797rpm and 0 HP	13.79%
Outer ring 2: 0.014 inch outer ring fault at 1797rpm and 0 HP	9.96%
Outer ring 3: 0.021 inch outer ring fault at 1797rpm and 0 HP	6.90%

2.2. JNU dataset

The Jiangnan University (JNU) bearing datasets consist of three bearing vibration datasets with three different rotating speeds (600, 800, and 1000 rpm) collected at 50 kHz. The JNU datasets show one health state and three fault modes (inner ring fault, outer ring fault, and rolling element fault). As a result, the total working conditions classes are twelve, as shown in Table 2 with the respective proportions for each state.

Table 2: JNU Fault Mode Description

Mode	Speed	Proportion	Mode	Speed	Proportion	Mode	Speed	Proportion
Health State	600 rpm	17.95%	Health State	800 rpm	14.81%	Health State	1000 rpm	17.66%
Inner ring	600 rpm	5.13%	Inner ring	800 rpm	5.98%	Inner ring	1000 rpm	4.56%
Outer ring	600 rpm	4.27%	Outer ring	800 rpm	5.41%	Outer ring	1000 rpm	6.27%
Rolling Element	600 rpm	5.13%	Rolling Element	800 rpm	7.69%	Rolling Element	1000 rpm	5.13%

3. THEORETICAL BACKGROUND

3.1. Variational Autoencoder

Autoencoders (AE) are unsupervised methods and were proposed in 1986 as a neural network that is trained exclusively to replicate their input. AE are responsible for reducing the dimensionality of inputs, and then reconstructing the reduced data to get as close as possible to the original input. These structures are composed of two networks, one is called encoder that reduces the amount of data to a latent space smaller than the original. The second network, known as decoder, replicates the input data from the latent space [12,13]. Through the analysis of the encoder and decoder, it is possible to assume that encoders must learn to identify important information and combine it properly, to reduce dimensionality with the least loss of information possible, while decoders are trained to translate the information present in the latent space [13].

The Variational Autoencoder was proposed as a solution for the need for models capable of adjusting data with great dimensionality and are meant to map the data input to a multivariate probabilistic distribution a priori [14]. Therefore, a VAE model seeks to encode the training data, not in a vector space of reduced dimension, but in a probability distribution whose probability density function (PDF) which is defined a priori and, during the optimization process, the best parameters for this distribution are sought. The space of the a priori distribution is defined by equations already known, the encoder converts the data into parameters of the distribution. Comparable to what is done in the AE, the VAE training process is based on the replication error, which is measured in the loss function, but in the VAE, the loss function also contains on and KLD. This new indicator added represents the difference between two probability distributions and is used to assess the difference between the distribution of latent space data from the a priori distribution [12].

3.2. Variational Autoencoder Model Structure

The encoder in the VAE model is developed to learn a variational approximation $q_\varphi(\mathcal{Z}|\mathcal{X})$ to the posterior distribution $p_\theta(\mathcal{Z}|\mathcal{X})$. Where φ and θ denote the encoder and decoder parameters, respectively. The VAE objective is written as [13]:

$$L(\theta,\varphi;x_i) = -D_{KL}(q_\varphi(\mathcal{Z}|\mathcal{X}_i)||p_\theta(\mathcal{Z})) + E_{q_\varphi(\mathcal{Z}|\mathcal{X}_i)}[\log p_\theta(\mathcal{Z}|\mathcal{X}_i) \quad (1)$$

The $D_{KL}(q||p)$ represents the Kullback-Leibler Divergence. The KLD is a measure of the degree of similarity between two probability distributions, while the second term is the data reconstruction error [15–17]. The prior over the latent variables is usually set to be the centred isotropic multivariate Gaussian $p_\theta(\mathcal{Z}) = N(\mathcal{Z};0,1)$. The posterior approximation $q_\varphi(\mathcal{Z}|\mathcal{X}_i)$ are Gaussian $q_\varphi(\mathcal{Z}|\mathcal{X}_i) = N\left(\mathcal{Z};\ \mu^{(i)},(\sigma^i)^2\right)$. Then the KLD component can be expressed as:

$$-D_{KL}(N\left(\mathcal{Z};\ \mu^{(i)},\left(\sigma^i\right)^2\right)||N(\mathcal{Z};0,1) = \frac{1}{2}\sum_{j=1}^{J}(1+\log\left(\left(\sigma_j{}^{(i)}\right)^2\right)-\left(\mu_j{}^{(i)}\right)^2-\left(\sigma_j{}^{(i)}\right)^2) \quad (2)$$

The "J" means the dimensionality of $\mathcal{Z}$, ${\mu_j}^{(i)}$ and ${\sigma_j}^{(i)}$ represents the j-th element of $q_\varphi(\mathcal{Z}|\mathcal{X}_i)$. The difference between the posterior approximation $q_\varphi(\mathcal{Z}|\mathcal{X}_i)$ and $p_\theta(\mathcal{Z}) = N(\mathcal{Z}; 0, 1)$ is measured by the KLD component. The purpose of KLD is for each sample $\mathcal{X}_i$ to calculate a posterior probability distribution $q_\varphi(\mathcal{Z}|\mathcal{X}_i)$ that approaches the standard normal distribution. When the model converges, the encoder's latent variables will follow a normal distribution $p_\theta(\mathcal{Z}) = N(\mathcal{Z}; 0, 1)$. The parameters $\mu^{(i)}$ and $\sigma^{(i)}$ are computed by the encoder networks, which can iteratively update with the optimization process. However, because $\mathcal{Z}^{(i)}$ is sampled from Gaussian $q_\varphi(\mathcal{Z}|\mathcal{X}_i) = N\left(\mathcal{Z};\ \mu^{(i)}, (\sigma^i)^2\right)$, the model is non-differentiable and cannot be optimized due to the latent variables. As a result, a differentiable transformation $g_\varphi(\varepsilon, \mathcal{X})$ of a noise variable ε is used to reparametrize the random variable $\mathcal{Z}$ [13]:

$$\mathcal{Z} = \mu^{(i)} + \sigma^{(i)}\varepsilon \quad with\ \varepsilon \sim N(0,1). \tag{3}$$

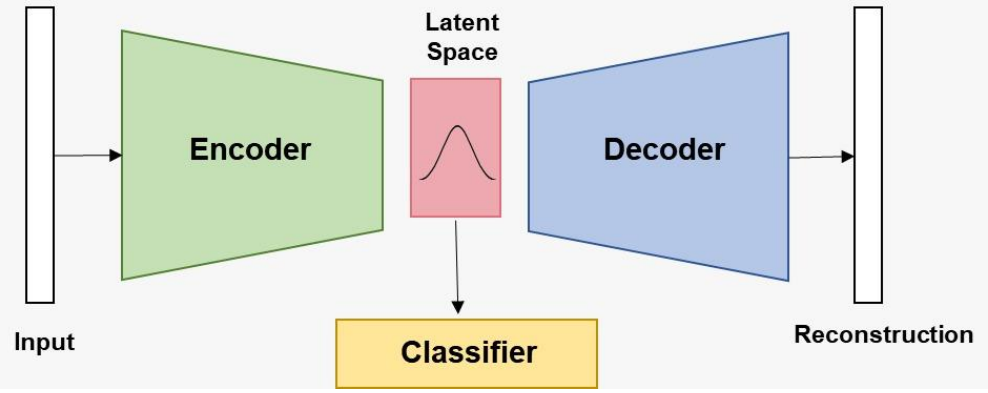
Thus, the Eq. 1 can be expressed as:

$$L(\theta, \varphi; x_i) = \frac{1}{2}\sum_{j=1}^{J}(1 + \log\left(\left({\sigma_j}^{(i)}\right)^2\right) - \left({\mu_j}^{(i)}\right)^2 - \left({\sigma_j}^{(i)}\right)^2) + \frac{1}{L}\sum_{l=1}^{L} log p_\theta(\mathcal{X}^{(i)}|\mathcal{Z}^{(i,l)} \tag{4}$$

And $\mathcal{Z}^{(i,l)} = \mu^{(i)} + \sigma^{(i)}\varepsilon^{(l)} \sim N(0, 1)$. The network can then be used to carry out the optimization procedure. The VAE can learn well-formed latent variables with a significant convergence speed without overfitting and reproduces the samples by sampling and decoding since the KL divergence is introduced as a loss function during the training process [12,13].

In this paper, we design a deep VAE for two-dimension (2D) input data using the MLP. In total, three architectures were used for comparison purposes. Three 2D inputs type was used, namely, Short-time Fourier transform (STFT), Continuous Wavelet Transform (CWT) and Slicing. They will be explained in section 3.3. The latent space dimension used is 8. The basic operating model works as shown in Figure 1 for all the different architectures.

Figure 1: VAE Structure



For the Architecture 1, we use an MLP for the encoder and decoder. The structures and encoder hyper-parameters have 4 layers that have neurons ranging from 500 to 100. With a range of minus 100 neurons between layers. In the encoder, we also have 4 layers ranging from 600 to 100. The last framework presented is the one employed for the classification of failure modes for each one of the models presented earlier. The classification structure is connected to the latent space in both cases, and it's a network composed by 5 layers, and the last one has the number of neurons equal to the number of classes for each dataset, which are 10 for the CWRU and 12 for JNU.

In the Architecture 2 we also use an MLP with more layers to identify if there are improvements in the model. The number of neurons for these layers goes from 50 to 600, and each adjacent layer has a difference of 50 neurons, what means that in the de encoder the first layer has 600 neurons, the second has 550 until the last one, that has 50. For the decoder, the first layer contains 50 neurons and gradually

increases within structure until reaches the last layer with 600 neurons. Architecture 3 has, on the other hand, layers with multiple of 2 neurons. The encoder and decoder of this architecture have 5 layers with neurons ranging from 512 to 32. This type of structure, with multiples of 2, is the most found in VAE applications. But as we will see in the results section, the results, in terms of accuracy, were not the best. Therefore, after testing different structures, it was seen that the structures presented earlier (architectures 1 and 2) are able to classify better.

3.3. Input types

The Short-time Fourier Transform (STFT) is a Fourier-related transform applied to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. To do it, the method for STFT computing is to split a long-time signal into smaller intervals of constant length and then compute the Fourier transform separately on each shorter segment [4,18]. This reveals the Fourier spectrum on each resulting segment. The chosen length of each sample is 33, resulting in 33 × 33 images.

The Continuous Wavelet Transform (CWT) is a widely used technique for signal processing and was first proposed for seismic processing. It comes as a solution to the failures of the Fourier Transform — since its analysis only occurs in the frequency domain — because it uses variable scale windows to process the data in the time-frequency domain [6]. The windows pass through the entire signal and each position of the studied spectrum is calculated. This process is repeated for larger and smaller windows, and then there is the signal representation in the time-frequency domain. The chosen length of each sample is 100, resulting in 100 × 100 images. Finally, the slicing image input consists of reshaping the non-processed data, i.e., the time domain input into a 32 × 32 image.

3.4. Evaluation metrics

The experiments were run with the proportions of 80% of samples as the training set and 20% of samples as the testing set which were randomly selected. To obtain reliable results and show the best overall accuracy that the model can achieve, we repeat each experiment five times. Four indicators are used to assess the performance of models, including the mean and maximum values of the overall accuracy obtained by the last epoch and the mean and maximum values of the maximal overall accuracy. For simplicity, they can be denoted as Last-Mean, Last-Max, Best-Mean, and Best-Max, respectively.

4. RESULTS

4.1. CWRU dataset

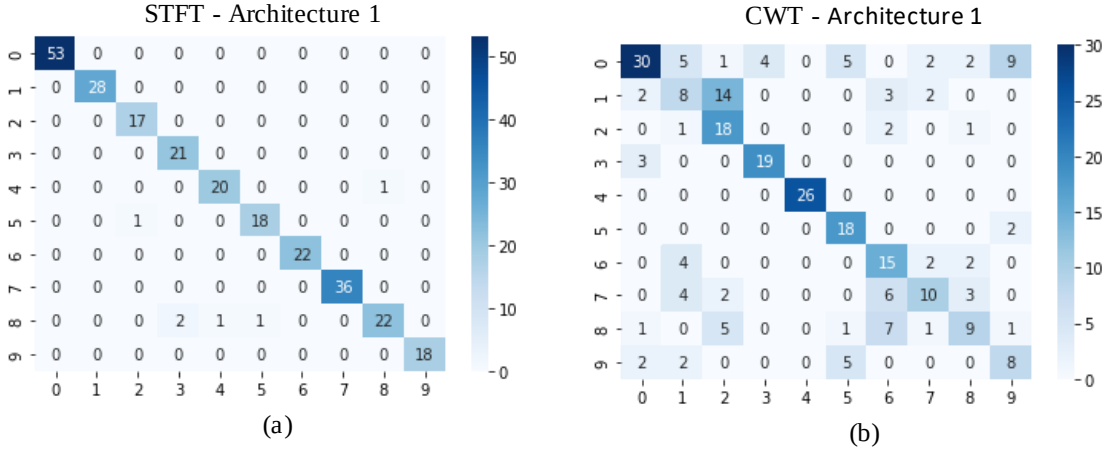
Table 3 presents the metrics for evaluating the model used according to the type of data processing. For CWRU we can identify that the model with the best accuracy is STFT, followed by CWT, and finally Slice. In fact, the Slice model is expected to be the worst among the three, since the raw data is only transformed into images, without major transformations. Therefore, there is more noise in the diagnosis of failure modes. We can also make a comparison between the different architectures used in VAE. STFT and CWT had better results in architecture 1, i.e., increasing the layers of the MLP does not result in greater assertiveness of the model. While architecture 2 excelled, in terms of accuracy, in relation to Slice. Architecture 3 has better results compared to STFT in relation to Architecture 2, but Architecture 1 still emerges. Finally, we can perform a comparison between the results of our architectures with those obtained by Zhao et al. [4] using a CNN. Regarding CWT and Slice, the supervised method has better metrics. As for STFT, on the other hand, the results are quite close. It shows that, with this type of data pre-processing, the VAE performs as well as the CNN.

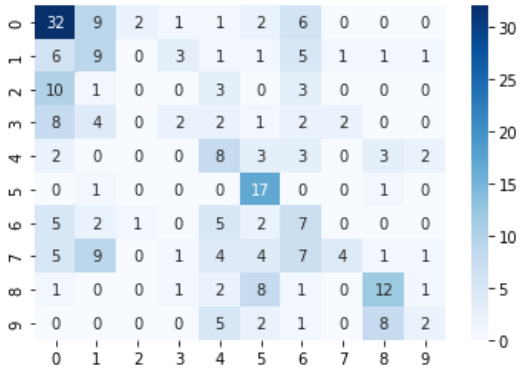
Table 3: CWRU Accuracy Results for different architectures

Architecture 1			
Metric	**STFT**	**CWT**	**Slice**
Last-Mean	0.9925	0.8127	0.3789
Last-Max	0.9947	0.8199	0.3908
Best-Mean	0.9966	0.8233	0.3968
Best-Max	0.9989	0.8262	0.4057
Architecture 2			
Metric	**STFT**	**CWT**	**Slice**
Last-Mean	0.7821	0.5757	0.4509
Last-Max	0.7934	0.5858	0.4792
Best-Mean	0.8090	0.5892	0.4792
Best-Max	0.7966	0.5921	0.4641
Architecture 3			
Metric	**STFT**	**CWT**	**Slice**
Last-Mean	0.9218	0.1769	0.1802
Last-Max	0.9297	0.1769	0.1810
Best-Mean	0.9282	0.1771	0.1857
Best-Max	0.9361	0.1779	0.1863
CNN – Zhao et al. [4]			
Metric	**STFT**	**CWT**	**Slice**
Last-Mean	0.9931	0.9885	0.8552
Last-Max	1.0000	0.9907	0.9387
Best-Mean	0.9946	0.9935	0.9464
Best-Max	1.000	0.9940	0.9655

Figure 2**: CWRU Models Confusion Matrix, (a) STFT – Architecture 1, (b) CWT – Architecture 1, (c) Slice – Architecture 1** confirms the information presented so far. For Architecture 1, we can see that in the confusion matrices, there are more matchings in Figure 2**: CWRU Models Confusion Matrix, (a) STFT – Architecture 1, (b) CWT – Architecture 1, (c) Slice – Architecture 1**a, that is, there were more right classifications in this scenario. The STFT model was able to correctly classify all normal health state data, i.e., state "0". It also correctly diagnosed all signals of failure modes 1, 6, and 9, i.e., Inner ring 1, Rolling Element 2, and Outer ring 3. In CWT, there was a complete matching only for failure mode 4, i.e., Inner ring 3. Finally, in Slice, there was at least one error in each of the classes. Thus, the STFT input is the most appropriate for this database for the scope of this analysis.

Figure 2: CWRU Models Confusion Matrix, (a) STFT – Architecture 1, (b) CWT – Architecture 1, (c) Slice – Architecture 1.





(c)

4.2. JNU dataset

The results obtained for this dataset are shown in **Erro! Fonte de referência não encontrada.**, and the same accuracy pattern present in the CWRU dataset results appears in the JNU, where the best results are found in the STFT pre-processed data, followed by CWT and Slice. But in this case, the accuracy is not as high as found in the CWRU data. In terms of architecture, the former was the one with the best results in terms of accuracy for all input types. Although the architecture 2 performed a bit better for the Slice data in comparison to the CWT, its results did not surpass the ones found in the first model. Similarly, architecture 3 performed better than architecture 2, but the first one is still better. Finally, we can compare our results with those of a supervised method (CNN). The results obtained by Zhao et al. [4] show that the use of CNN is preferable when pre-processing the data with CWT or Slice. However, our three architectures were more effective when preprocessing data in STFT.

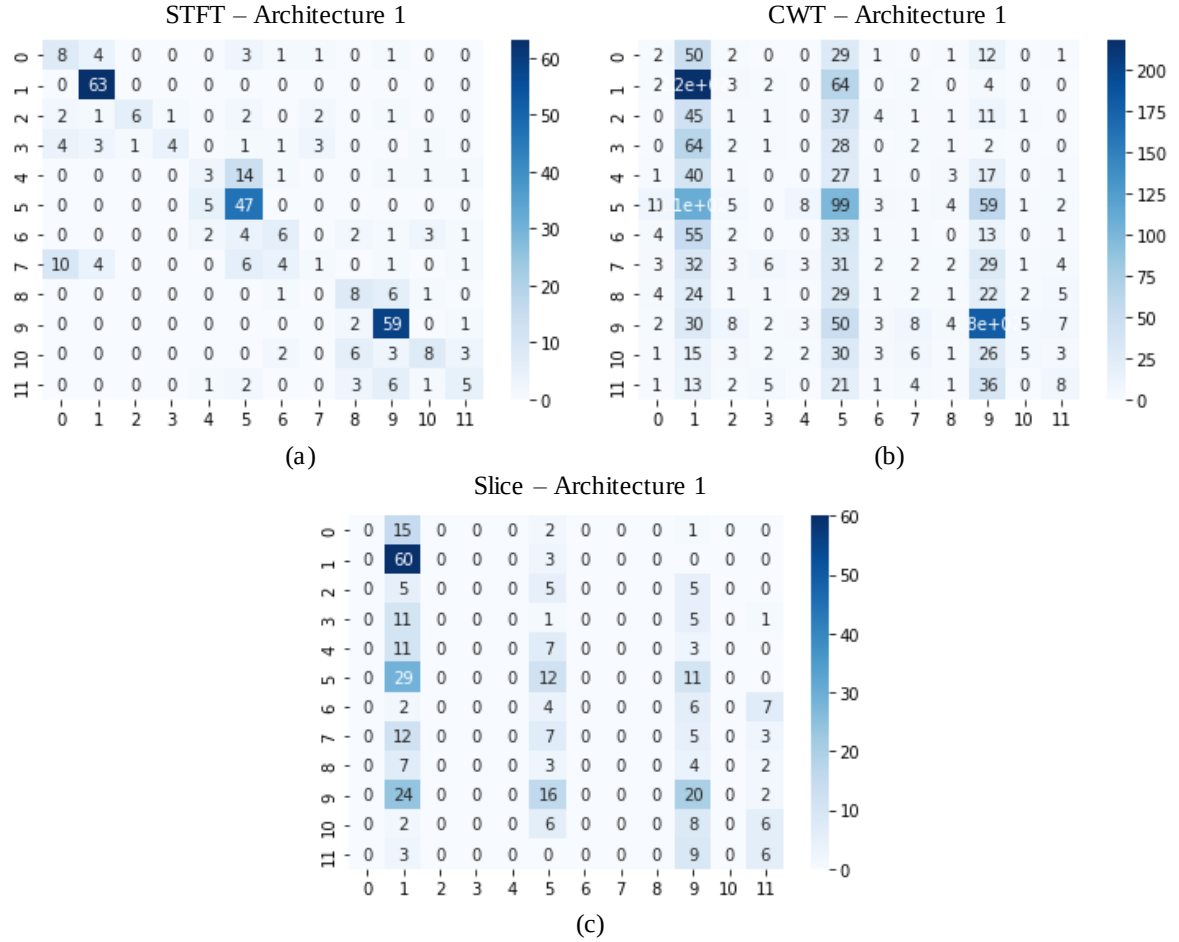
Table 4: JNU Accuracy Results

Architecture 1			
Metric	**STFT**	**CWT**	**Slice**
Last-Mean	0.6690	0.3780	0.2730
Last-Max	0.6659	0.3748	0.2671
Best-Mean	0.6690	0.3717	0.2587
Best-Max	0.6598	0.3684	0.2543
Architecture 2			
Metric	**STFT**	**CWT**	**Slice**
Last-Mean	0.5960	0.1660	0.1708
Last-Max	0.6016	0.1720	0.1825
Best-Mean	0.6082	0.1762	0.1913
Best-Max	0.6111	0.1798	0.1937
Architecture 3			
Metric	**STFT**	**CWT**	**Slice**
Last-Mean	0.6306	0.1670	0.2557
Last-Max	0.6468	0.1757	0.2619
Best-Mean	0.6399	0.1766	0.2717
Best-Max	0.6468	0.1782	0.2746
CNN – Zhao et al. [4]			
Metric	**STFT**	**CWT**	**Slice**
Last-Mean	0.4370	0.6974	0.4285
Last-Max	0.4811	0.7236	0.4644
Best-Mean	0.4716	0.7231	0.4866
Best-Max	0.5166	0.7550	0.4957

The confusion matrixes in Figure 3 gives the information regarding the classification for each class, and for the architecture 1 we can observe that the normal health states for each working condition (“1”, “5” and “9”) are the ones with more hits, like which is seen in the previous dataset. But the model is not good at classifying the fault states in the JNU case, and only class “1”, which corresponds to the health state for rotating speed 600 rpm, was a full hit. And for the second architecture, the performance is

inferior for all data, and for the CWT and the Slide data, the model classified all the points in only one class.

Figure 3: JNU Models Confusion Matrix, (a) STFT – Architecture 1, (b) CWT – Architecture 1, (c) Slice – Architecture 1.



5. CONCLUSIONS

In this paper, a VAE model for classification was proposed and two publicly available datasets were selected to evaluate its performance. Three data pre-processing techniques were considered in the analysis for each dataset, and the STFT input showed the best accuracy. For both datasets, the health state modes were the ones best classified by the model, while it struggled on how to determine which type of failure was present in the data. Probably the accuracy can be improved if the faults in distinct working conditions are consolidated in only one class instead of many, which is possible future step for this work. Although the model was able to make a considerable match for the CWRU dataset, this has not happened to JNU dataset, what shows that the difficulty presents in the datasets influenced considerably the performance. An important remark is that the increase in the number of layers in the encoder and in decoder was not directly converted to an improvement for the model's learning ability, which might mean that more sophisticated models are needed. A possible way is to test different machine learning models such as Convolution Neural Networks within the VAE structure instead of MLP or even combinations of both models. The use of Generation Adversarial Networks can be also investigated to generate more data and boost the performance.

Acknowledgements

The authors thank the Humans Resource Program (PRH 38.1) titled "Risk Analysis and Environmental Modeling in the Exploration, Development and Production of Oil and Gas", financed by the Brazilian Agency for Petroleum, Natural Gas and Biofuels (ANP) and managed the Brazilian Funding Authority

for Studies and Projects (FINEP), for the financial support in this research. This study was also financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES), Finance Code 001. The first author thanks the Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco (FACEPE) and the second author thanks the Brazilian National Agency for Research (CNPq).

References

[1] Y. Liu, H. Jiang, Y. Wang, Z. Wu, S. Liu, "*A conditional variational autoencoding generative adversarial networks with self-modulation for rolling bearing fault diagnosis*", Meas. J. Int. Meas. Confed., 192, 11088, (2022).
[2] S. Wang, X. Chen, C. Tong, Z. Zhao, "*Matching Synchrosqueezing Wavelet Transform and Application to Aeroengine Vibration Monitoring*", IEEE Trans. Instrum. Meas. 66, pp. 360–372, (2017).
[3] Z. Zhao, S. Wu, B. Qiao, S. Wang, X. Chen, "*Enhanced sparse period-group lasso for bearing fault diagnosis*", IEEE Trans. Ind. Electron., 66, pp. 2143–2153m (2019).
[4] Z. Zhao, T. Li, J. Wu, C. Sun, S. Wang, R. Yan, X. Chen, "*Deep learning algorithms for rotating machinery intelligent diagnosis: An open-source benchmark study*", ISA Trans. 107, pp. 224–255, (2020).
[5] M. Saimurugan, K.I. Ramachandran, V. Sugumaran, N.R. Sakthivel, "*Multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine*", Expert Syst. Appl., 38, pp. 3819–3826, (2011).
[6] L. Song, H. Wang, P. Chen, "*Vibration-Based Intelligent Fault Diagnosis for Roller Bearings in Low-Speed Rotating Machinery*", IEEE Trans. Instrum. Meas., 67, pp. 1887–1899, (2018).
[7] A. Soualhi, K. Medjaher, N. Zerhouni, "*Bearing Health Monitoring Based on Hilbert – Huang Transform, Support Vector Machine, and Regression*", IEEE Trans. Instrum. Meas., 64, pp. 52–62, (2015).
[8] M.M.M. Islam, J.M. Kim, "*Automated bearing fault diagnosis scheme using 2D representation of wavelet packet transform and deep convolutional neural network*", Comput. Ind., 106, pp. 142–153, (2019).
[9] L. Chen, Z. Zhu, "*An End-to-End Model Based on Improved Adaptive Deep Belief Network and Its Application to Bearing Fault Diagnosis*", IEEE Access., 6, pp. 63584–63596, (2018).
[10] M. Hemmer, A. Klausen, H. Van Khang, K.G. Robbersmyr, T.I. Waag, "*Health Indicator for Low-Speed Axial Bearings Using Variational Autoencoders*", IEEE Access., 8, pp. 35842-35852, (2020).
[11] CWRU PAGE, Case West. Reserv. Univ. Bear. Data Cent. Website. (2020). https://csegroups.case.edu/bearingdatacenter/pages/welcome-case-western-reserve-university-bearing-data-center-website.
[12] S. Zhang, F. Ye, B. Wang, T.G. Habetler, "*Semi-Supervised Bearing Fault Diagnosis and Classification Using Variational Autoencoder-Based Deep Generative Models*", IEEE Sens. J., 21, pp. 6476–6486, (2021).
[13] D. Zhao, S. Liu, D. Gu, X. Sun, L. Wang, Y. Wei, H. Zhang, "*Enhanced data-driven fault diagnosis for machines with small and unbalanced data based on variational auto-encoder*", Meas. Sci. Technol., 31, pp. 035004, (2019).
[14] N. Sivaramakrishnan, V. Subramaniyaswamy, A. Viloria, V. Vijayakumar, N. Senthilselvan, "*A deep learning-based hybrid model for recommendation generation and ranking*", Neural Comput. Appl., 33, pp. 10719–10736, (2021).
[15] Y. Mehrali, M. Asadi, "*Parameter estimation based on cumulative kullback–leibler divergence*", Revstat Stat. J., 19, pp. 111–130, (2021).
[16] G. Yari, Z. Tondpour, "*Estimation of the exponential pareto ii distribution parameters*", Commun. Stat. Simul. Comput., 46, pp. 6889–6906, (2017).
[17] G. Yari, A. Mirhabibi, A. Saghafi, "*Estimation of the Weibull parameters by Kullback-Leibler divergence of survival functions*", Appl. Math. Inf. Sci., 7, pp. 187–192, (2013).
[18] C. Mateo, J.A. Talavera, "*Short-time Fourier transform with the window size fixed in the frequency domain*", Digit. Signal Process. A Rev. J., 77, pp. 13–21, (2018).