Hybrid Architecture of Deep Convolutional Variational Auto-encoder for Remaining useful Life Prediction

Ryad Zemouri, Zeina Al Masry, Ikram Remadna, Sadek Labib Terrissa, Noureddine Zerhouni

Hybrid architecture of deep convolutional variational auto-encoder for remaining useful life prediction

Ryad Zemouri

CEDRIC laboratory of the Conservatoire National des Arts et Métiers (CNAM), HESAM université, 292, rue Saint-Martin, 750141 Paris cedex 03, France. E-mail: ryad.zemouri@cnam.fr

Zeina Al Masry

FEMTO-ST institute, Univ. Bourgogne Franche-Comte, CNRS, ENSMM, Besançon, France. E-mail: zeina.almasry@femto-st.fr

Ikram Remadna

LINFI Laboratory, University of Biskra, Biskra, Algeria. E-mail: ikram.remadna@outlook.com

Sadek Labib Terrissa

LINFI Laboratory, University of Biskra, Biskra, Algeria. E-mail: terrissalabib@gmail.com

Noureddine Zerhouni

FEMTO-ST institute, Univ. Bourgogne Franche-Comte, CNRS, ENSMM, Besançon, France. E-mail: noureddine.zerhouni@femto-st.fr

The remaining useful life prediction is a key element in decision-making and maintenance strategies development. Therefore, in practical situation, it is usually affected by uncertainty. The aim of this work is hence to propose a deep learning method which predicts when an in-service machine will fail to overcome the latter problem. It is based on deep convolutional variational autoencoder (CVAE). The proposed approach is validated using the C-MAPSS dataset of the aero-engine. The model's classification performance has reached a superior accuracy compared to existing models and it is used for machine failure prediction in different time windows.

Keywords: Prognostics and Health Management, Predictive Maintenance, Deep Learning, Variational Autoencoder, Data visualisation.

1. Introduction

Nowadays, industries seek to improve their maintenance strategies in order to increase the lifetime of their equipments. Many of them are implementing condition-based maintenance (CBM), which seems to be a very promising strategy to minimize the overall cost and reduce the mean downtime. The maintenance decision within CBM could be based on different characteristics such as the remaining useful life (RUL), the reliability or the cost function. Therefore, the RUL form a key element of the CBM to reduce the cost of preventive and corrective maintenance. Uncertainty in RUL prediction is hence a scientific problem that should be resolved since it affects the accuracy of Prognostics and Health Management (PHM) implementation in industries.

In this paper, a new method based on visual data analysis to predict when an in-service machine will fail is proposed. A deep Convolutional Variational AutoEncoder (CVAE) is used in order to automatically extract performance degradation feature from multiple sensors (Zemouri et al. (2019)). The main aim of using the CVAE is to provide more structured and lower-dimensional representation of the data that showed the best distribution of the class over a 2D-latent space and demonstrates how well the CVAE generalizes (Zemouri et al. (2019)). The encoder, part of the CVAE, is then used for data projection in a 2D-visualisation latent space. The input vectors are encoded and displayed into this 2D-space, which help the expert to visually analyse the spatial distribution of the training dataset. Three degradation classes are then defined according to two thresholds (α_1, α_2). By analysing the spatial distribution, the expert try to find the adapted thresholds setting by minimising the overlapping area between the degradation classes. We then predict the RUL according to the latter degradation classes. The results will be validated using the C-MAPSS dataset of the aero-engine Saxena et al. (2008).

The paper is organized as follows: In Section 2, a theoretical background of the variational autoencoder is presented before the description of the proposed RUL estimation method. In Section 3, the experiment and results analysis are given. Finally, a conclusion is given in Section 4.

2. Methodologies

We here develop the proposed methodology. Firstly, the theoretical background of the variational autoencoders is described. Secondly, the RUL prediction framework based on the CVAE is provided.

2.1. *Variational autoencoders: theoretical background*

An autoencoder (AE) is an unsupervised neural network (NN) trained to reproduce an input vector $\mathbf{X} \in \mathbb{R}^m$ where $m \in \mathbb{N}$ refers to the dimension of $\mathbf{X}$ (Yu and Príncipe (2019), Bengio (2014), Im et al. (2015), Fan (2019)). The AE is composed by two main structures: an encoder and a decoder which are multilayered NNs parameterized by two weight vectors ϕ and θ (see Figure 1). The first one encodes the input data $\mathbf{X}$ into a latent representation $\mathbf{z}$ by the encoder function $\mathbf{z} = f_\phi(\mathbf{X})$, whereas the second one decodes this latent representation onto $\hat{\mathbf{X}} = h_\theta(\mathbf{z})$ which is an approximation or reconstruction of the original data $\mathbf{X}$. In an AE, an equal number of units are used in the input/output layers while less units are used in the latent space. The AEs are usually used for data compression (i.e., feature extraction/reduction), noise removal and pre-trained parameters for a complex network.

A variational autoencoder (VAE) has the same functions as the AE in the sense that it is composed by an encoder and a decoder (Figure 1). VAE becomes a popular generative model by combining Bayesian inference and the efficiency of the NNs to obtain a nonlinear low-dimensional latent space (Martin et al. (2018), Canchumuni et al. (2019), Lee et al. (2019), Zhang et al. (2019)). The Bayesian inference is obtained by an additional layer used for sampling the latent vector $\mathbf{z}$ with a prior specified distribution $p(\mathbf{z})$, usually assumed to be a standard Gaussian $\mathcal{N}(0, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. Each element z_i of the latent layer $\boldsymbol{Z}$ is obtained as follow:

$$z_i = \mu_i + \sigma_i.\epsilon \tag{1}$$

where μ_i and σ_i are the i^{th} components of the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ vectors, ϵ is a random variable following a standard Normal distribution ($\epsilon \sim \mathcal{N}(0, 1)$). Unlike the AE which generates the latent vector $\mathbf{z}$, the VAE generates vector of means μ_i and standard deviations σ_i.

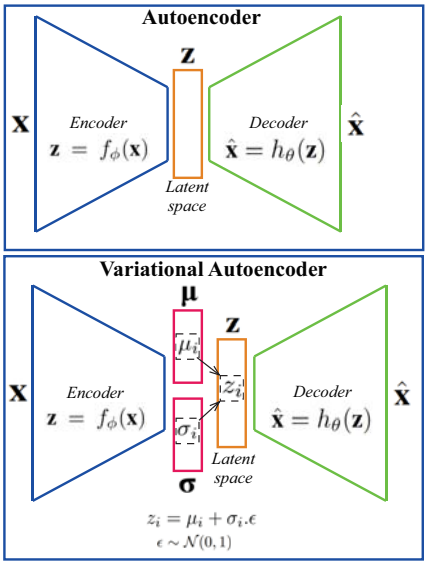

Fig. 1. Schematic architecture of a standard deep autoencoder and a variational deep autoencoder. Both architectures have two parts: an encoder and a decoder.

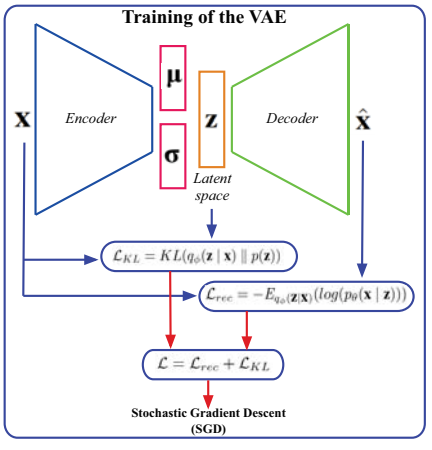

Fig. 2. The VAE loss function. The first term $\mathcal{L}_{rec}$ is the reconstruction loss function. The second term $\mathcal{L}_{KL}$ corresponds to the Kullback-Liebler divergence loss term that forces the generation of a latent vector with the specified Normal distribution. When the VAE is trained, the two functions encoder/decoder can be used separately even to reduce the space dimension by encoding the input data or to generate synthetic samples by decoding new variables from the latent space.

This allows to have more continuity in the latent space than the original AE. The VAE loss function given by the Equation 2 has two terms. The

first term $\mathcal{L}_{rec}$ is the reconstruction loss function (Equation 3). Usually the negative expected log-likelihood (e.g., the cross-entropy function) is used (Hou et al. (2019), Lee et al. (2019), Wang et al. (2019), Sun et al. (2018), Canchumuni et al. (2019)) but the mean squared error can also be used (Zhang et al. (2019)). The second term $\mathcal{L}_{KL}$ (Equation 4) corresponds to the Kullback-Liebler (KL) divergence loss term that forces the generation of a latent vector with the specified Normal distribution (Kingma and Welling (2013), Kingma (2017)). The KL divergence is a theoretical measure of proximity between two densities q and p and it is noted by $KL(q \parallel p)$. The dissimilarities between these densities are asymmetric ($KL(q \parallel p) \neq KL(p \parallel q)$), non-negative and are minimized when $q(x) = p(x)\ \forall x$ (Blei et al. (2016)). Thus, the KL divergence term measures how close is the conditional distribution density $q_\phi(\mathbf{z} \mid \mathbf{x})$ of the encoded latent vectors from the desired Normal distribution $p(\mathbf{z})$. The value of KL is zero when two probability distributions are the same, which forces the encoder of VAE to learn the latent variables that follow a multivariate normal distribution over a k-dimensional latent space.

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{KL} \tag{2}$$

where

$$\mathcal{L}_{rec} = -E_{q_\phi(\mathbf{z}|\mathbf{x})}(log(p_\theta(\mathbf{x} \mid \mathbf{z}))), \tag{3}$$

$$\mathcal{L}_{KL} = KL(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})) \tag{4}$$

with $p_\theta(\mathbf{x} \mid \mathbf{z})$ is the conditional distribution density of the decoded latent vectors.

When the VAE is trained, each function (i.e., the encoder and the decoder) can be used separately, either to reduce the space dimension by encoding the input data, or to generate synthetic samples by decoding new variables from the latent space (Figure 2). Most of the generative applications deal with image processing as in Hou et al. (2019) where a VAE was trained to generate face images with much clearer and more natural noses, eyes, teeth, hair textures as well as reasonable backgrounds. In Canchumuni et al. (2019), a generative model is constructed to create new random realizations of faces that are indistinguishable from samples.

In nonlinear processes monitoring, VAE have been recently used for high-dimensional process fault diagnosis. The most relevant characteristics of the process are extracted by the latent variable space by projecting the high-dimensional process data into a lower-dimensional space (Zemouri et al. (2019), Lee et al. (2019), Zhang et al. (2019), Cheng et al. (2019), Zhang et al. (2018), Wang et al. (2019), ren Wang et al. (2019), Mastroleo et al. (2018), Martin et al. (2018), Mao et al. (2019)).

2.2. *Remaining useful life estimation based on VAE*

Figure 3 provides the proposed methodology for the RUL classes estimation. A CVAE is used to capture the data feature distribution. Recall that CVAE is a particular case of VAE describe in Section 2.1. The CVAE architecture includes two parts, an encoder and a decoder, which are two symmetrical and reversed structures. Each one is composed by two convolutional layers and two fully connected layers. For the encoder, we use convolutional layers with 3×3 kernels and the same padding. The stride was 1×1 for the first convolutional layer and 2×2 for the second. The latent two-dimensional space is represented by two 2D-layers for the encoder: the mean and the standard deviation layers (i.e., $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$), and one 2D-sampling layer ($\mathbf{Z}$) for the decoder.

The first step was to train the whole CVAE architecture for the reconstruction of the input vector. Training the CVAE does not need the label information of the input data. When the training process of the CVAE is successfully done, the encoder part is then used as a 2D-Visualisation tool by a human expert in order to analyse the spatial distribution of the data set. Three degradation classes are then defined according to two thresholds (α_1, α_2) as follows:

- Degradation class 0: $RUL < \alpha_1$,
- Degradation class 1: $\alpha_1 < RUL < \alpha_2$,
- Degradation class 2: $RUL > \alpha_2$.

Each couple (α_1, α_2) will generate a particular overlapping situation between the degradation classes, which is easily visualised by the expert in the 2D-latent space. By analysing the spatial distribution, the expert try to find the adapted thresholds setting by minimising the overlapping area between the degradation classes.

The second step is to train the classifier for the RUL classes estimation. The encoder parameters obtained by the previous step are frozen during the classifier training step.

Finally, in the operating stage, the convolutional encoder is used jointly with the classifier to estimate the RUL based on the degradation classes (as shown by the Figure 3).

3. Experiment and results analysis

3.1. *C-MAPSS Dataset*

We begin by describing the used dataset for our experimental study. We here use a dataset from Nasa repository which is the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) generated data. It concerns the simulation of

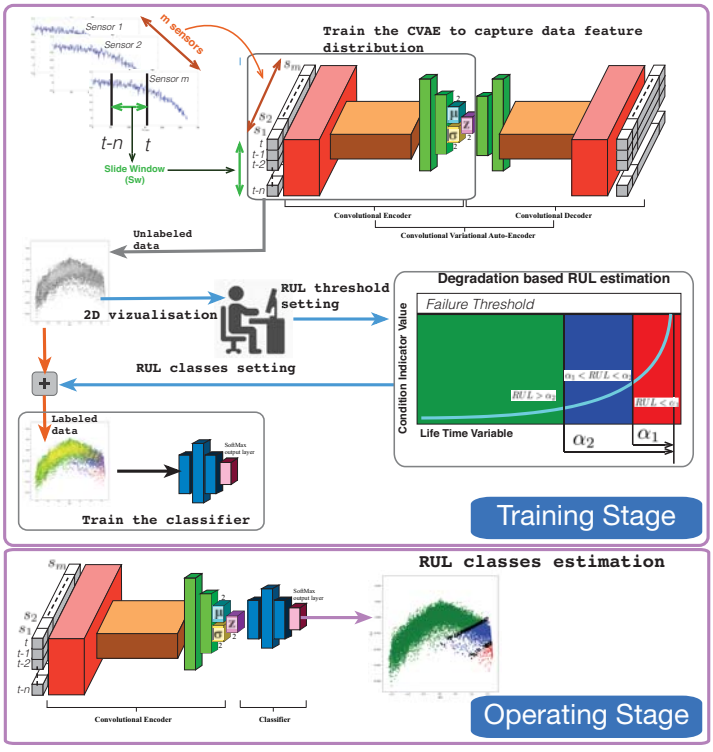

Fig. 3. The framework of the proposed RUL estimation.

a turbofan engine degradation data-sets. These datasets Saxena et al. (2008) are organized in four data subsets with 26 criteria and many data rows. In the following, we only used the failure set FD001.

3.2. *Results*

Several sizes of the slide window (SW) were tested and evaluated according to different degradation thresholds α_1 and α_2. Table 1 gives the detail of the used architecture. Figure 4 shows the confusion matrices obtained for each test. For each confusion matrix, the ordinate represents the true label while the abscissa represents the predicted one.

One should point out that if the thresholds of α_1 and α_2 are too close, the classifier fails to predict the middle class due to its reduced area. For example, despite the size of the SW, the predicted degradation $\#1$ is almost nonexistent for $\alpha_1 = 10$ and $\alpha_2 = 20$.

However, by increasing the gap between the two degradation thresholds, the performance of the RUL prediction is improved. The better estimation is obtained by the slide window SW = 16 with $\alpha_1 = 10$ and $\alpha_2 = 90$. The proportion of the true positive classification obtained for the degradation classes $\#0$, $\#1$ and $\#2$ are respectively 69%, 60% and 91%. Note that the worst results are obtained with the thresholds $\alpha_1 = 40$ and $\alpha_2 = 150$ and the most critical degradation class $\#2$ was correctly predicted (except for the last threshold couple).

We can remark form the obtained results that the proposed CVAE approach for RUL estimation is quite interesting for preventive maintenance purposes. To be more precise, except for the first and last threshold couples, all the false positive predictions are belonging to the less critical degradation class. As it can be seen in the confusion matrix obtained by the slide window SW = 6 and $(\alpha_1 = 10, \alpha_2 = 70)$ with the below performances:

- Degradation 0: 56% are correctly classified as Degradation 0, and 44% are incorrectly classified as Degradation 2,
- Degradation 1: 43% are correctly classified as Degradation 1, and 51% are incorrectly classified as Degradation 3,
- Degradation 2: 98% are correctly classified as Degradation 2.

Finally, from the maintenance point of view, this approach could be the adapted one to alert the cases with critical degradation levels. This means

Table 1. The Proposed Hybrid deep convolutional variational auto-encoder architectures.

Layer	Type	Neurons	Kernels
Encoder			
0	Input vector	26 × Sw × 1	-
1	Convolution	26 × Sw × 32	4 × 1
2	Convolution	13 × Sw × 64	4 × 1
3	Fully connected	200	-
4	Fully connected	100	-
5	Mean layer	2	-
5	Standard deviation layer	2	-
Decoder			
0	Sampling layer	2	-
1	Fully connected	100	-
2	Fully connected	200	-
3	Deconvolution	13 × Sw × 64	4 × 1
4	Deconvolution	26 × Sw × 32	4 × 1
5	Output vector	26 × Sw × 1	-
Classifier			
0	Input	2	-
1	Fully connected	50	-
2	Fully connected	100	-
3	Fully connected	50	-
4	SoftMax output layer	3	-

that the predicted RUL is less then the true one ($RUL_{predicted} \leq RUL_{True}$).

4. Conclusion & Future work

In this paper, we have presented the framework of a new approach for RUL prediction based on CVAE techniques. The efficiency of the proposed method has been highlighted in the numerical analysis using C-MAPSS dataset. Based on this work, one could think about some theoretical approaches to choose the adapted threshold couple (α_1, α_2) to formalize the major analysis which is focused on the fact that the false positive predictions are all in the least critical degradation classes. On the other hand, it could be interesting to integrate this CVAE method with some condition-based maintenance strategies (in the spirit of Al Masry et al. (2017)) to reduce the cost and to increase systems safety. Finally, to highlight the effectiveness of the proposed Hybrid architecture of deep convolutional variational auto-encoder, some comparison with other NNs architectures will be developed.

References

Al Masry, Z., S. Mercier, and G. Verdier (2017). A condition-based dynamic maintenance policy for an extended gamma process. In *MMR 2017 (International Conference on Mathematical Methods on Reliability)*.

Bengio, Y. (2014). How auto-encoders could provide credit assignment in deep networks via target propagation. *CoRR abs/1407.7906*.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2016, Jan). Variational inference: A review for statisticians. *arXiv e-prints*, arXiv:1601.00670.

Canchumuni, S. W., A. A. Emerick, and M. A. C. Pacheco (2019). Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother. *Computers & Geosciences 128*, 87 – 102.

Cheng, F., Q. P. He, and J. Zhao (2019). A novel process monitoring approach based on variational recurrent autoencoder. *Computers & Chemical Engineering 129*, 106515.

Fan, Y. J. (2019). Autoencoder node saliency: Selecting relevant latent representations. *Pattern Recognition 88*, 643 – 653.

Hou, X., K. Sun, L. Shen, and G. Qiu (2019). Improving variational autoencoder with deep feature consistent and generative adversarial training. *Neurocomputing 341*, 183 – 194.

Im, D. J., S. Ahn, R. Memisevic, and Y. Bengio (2015). Denoising criterion for variational auto-encoding framework. *CoRR abs/1511.06406*.

Kingma, D. (2017, October). *Variational inference & deep learning: A new synthesis*. Ph. D. thesis, Faculty of Science (FNWI), Informatics Institute (IVI), University of Amsterdam.

Kingma, D. P. and M. Welling (2013, Dec). Auto-

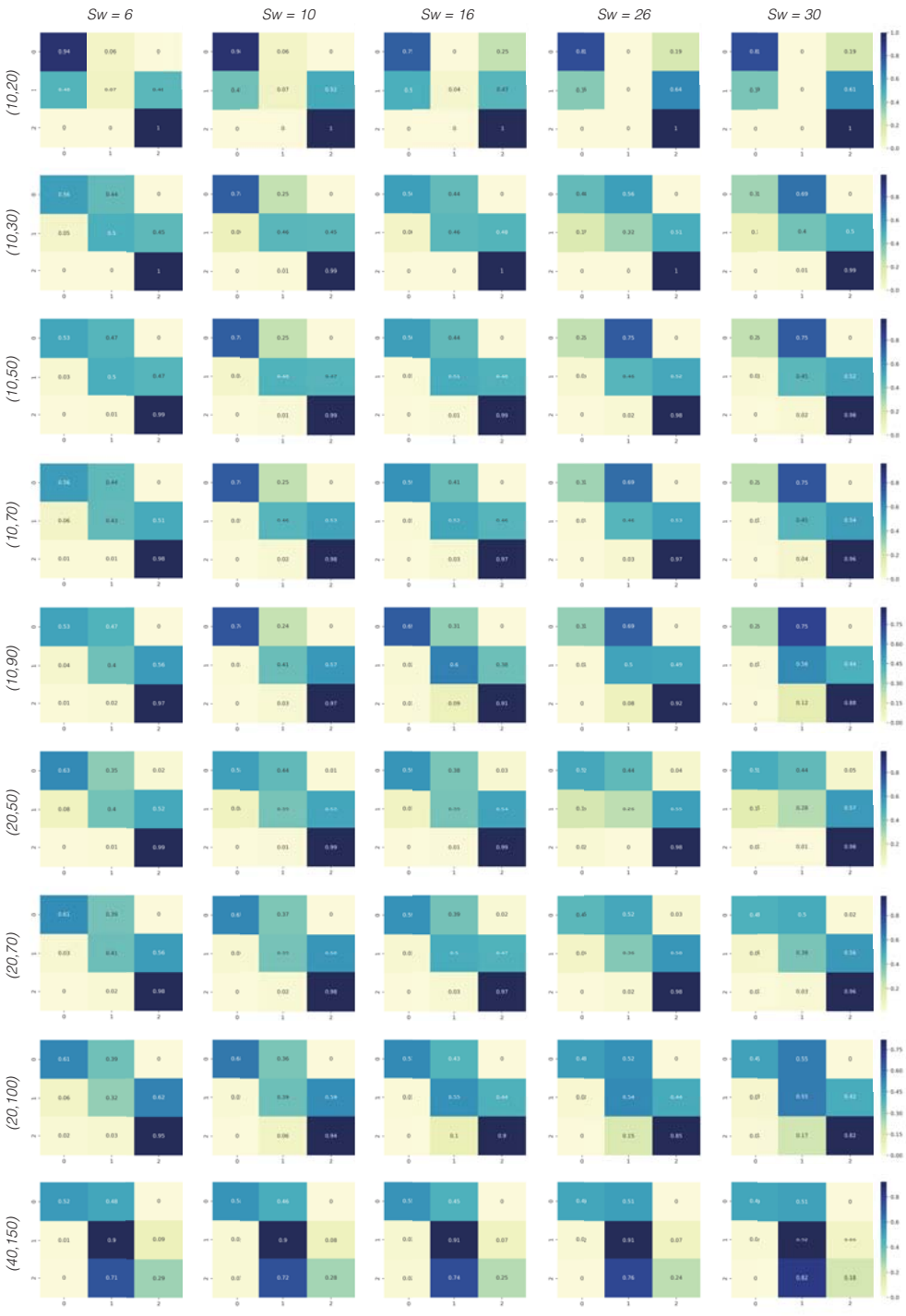

Fig. 4. The confusion matrices obtained on test set FD001 with different slide windows (SW) and different thresholds (α_1, α_2).

Encoding Variational Bayes. *arXiv e-prints*, arXiv:1312.6114.

Lee, S., M. Kwak, K.-L. Tsui, and S. B. Kim (2019). Process monitoring using variational autoencoder for high-dimensional nonlinear processes. *Engineering Applications of Artificial Intelligence 83*, 13 – 27.

Mao, W., Y. Liu, L. Ding, and Y. Li (2019). Imbalanced fault diagnosis of rolling bearing based on generative adversarial network: A comparative study. *IEEE Access 7*, 9515–9530.

Martin, G. S., E. L. Droguett, V. Meruane, and M. das Chagas Moura (2018). Deep variational auto-encoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis. *Structural Health Monitoring 0*(0), 1475921718788299.

Mastroleo, M., R. Ugolotti, L. Mussi, E. Vicari, F. Sassi, F. Sciocchetti, B. Beasant, and C. McIlroy (2018). Automatic analysis of faulty low voltage network asset using deep neural networks. *The Journal of Engineering 2018*(15), 851–855.

ren Wang, Y., Q. Jin, G. dong Sun, and C. fei Sun (2019). Planetary gearbox fault feature learning using conditional variational neural networks under noise environment. *Knowledge-Based Systems 163*, 438 – 449.

Saxena, A., K. Goebel, D. Simon, and N. Eklund (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management*, pp. 1–9. IEEE.

Sun, J., X. Wang, N. Xiong, and J. Shao (2018). Learning sparse representation with variational auto-encoder for anomaly detection. *IEEE Access 6*, 33353–33361.

Wang, K., M. G. Forbes, B. Gopaluni, J. Chen, and Z. Song (2019). Systematic development of a new variational autoencoder model based on uncertain data for monitoring nonlinear processes. *IEEE Access 7*, 22554–22565.

Yu, S. and J. C. Príncipe (2019). Understanding autoencoders with information theoretic concepts. *Neural Networks 117*, 104 – 123.

Zemouri, R., M. Lévesque, N. Amyot, C. Hudon, O. Kokoko, and A. Tahan (2019). Deep convolutional variational autoencoder as a 2d-visualization tool for partial discharge source classification in hydrogenerators. *IEEE Access*, 1–1.

Zhang, Z., T. Jiang, S. Li, and Y. Yang (2018). Automated feature learning for nonlinear process monitoring – an approach using stacked denoising autoencoder and k-nearest neighbor rule. *Journal of Process Control 64*, 49 – 61.

Zhang, Z., T. Jiang, C. Zhan, and Y. Yang (2019). Gaussian feature learning based on variational autoencoder for improving nonlinear process monitoring. *Journal of Process Control 75*, 136 – 155.