iomac
International Operational Modal Analysis Conference

11th International Operational Modal Analysis Conference

20 - 23 May 2025 | Rennes, France

Application of pseudo-anomaly generation in unsupervised fault diagnosis of wind turbine bearings under changing operating conditions

Magnus Munk Jensen , Szymon Greś , Rafał Wiśniewski , Dorte Hammershøi

Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark, {magnusmj,sgres,raf,dh}@es.aau.dk

ABSTRACT

In the context of fault detection of bearings, state-of-the-art unsupervised methods struggle to capture anomalies when both speed and loads are a variable. This limitation may arise due to the lack of knowledge about a possibly faulty system behavior and the reliance on a fixed operating point of the baseline system. In this work a machine learning (ML) approach is implemented to address these challenges. To this end, an autoencoder framework is used to extract fault-sensitive features, a support vector machine is employed to classify them, and a pseudo-anomaly scheme is used to ensure a balanced training dataset with an equal distribution of healthy and faulty data. The application concerns a laboratory experiment of a progressively damaged bearing in a wind turbine drivetrain simulator. Different operating points of the drivetrain system related to changing speed and load are considered. The result highlights the potential of ML-based methods for fault diagnosis of wind turbine bearings, offering a more reliable alternative to the conventional fault detection methods.

Keywords: Condition Monitoring, Rotating Machinery, Autoencoder, Support Vector Machine, Pseudo Anomalies

1. INTRODUCTION

Condition monitoring (CM) aims at assessing the operational integrity of rotating machinery in order to provide early warnings about the potential failures. With the recent advancements in sensor technologies and involvement of bearings, gearboxes and drivetrains in many dynamic systems, e.g., wind turbines, CM has received a major attention in the engineering community [1]. Therein, vibration-based methods take the lead, relying on the fact that a potential fault will introduce changes in the machine's dynamics,

which is reflected in the measured vibration signals and consequently in the data-driven features to detect faults. Changes in the operating conditions, e.g., the rotating speed, the load, also affect the machine's dynamics and are characterized by signatures resembling the potential faults, in particular if the latter are at an early stage. This renders the detection of incipient faults in rotating machinery challenging and requires methods that may separate the effects of the varying operating conditions from those due to faults and eliminate false alarms to ensure reliable decision-making. In the context of CM, plethora of methods addressing the problem of robustness of fault detection towards changes in the operating conditions exists. Therein, the approaches are usually categorized in two groups: the explicit and the implicit methods. The explicit methods require measurements of the operating conditions for modeling their influence on the system dynamics and fault-sensitive features. This includes subspace interpolation [2], different regression models, e.g., linear regression [3], non-linear regression [4], Gaussian process models [5] and supervised ML methods [6, 7]. The implicit methods do not directly consider information on the changing operating conditions in the design of fault-sensitive features and usually rely on their normalization with data-driven quantities related to the unknown operating point of the baseline system dynamics, among other techniques. Among the implicit methods are cointegration [8], subspace selection and normalization [9, 10], or unsupervised ML approaches [11]. Although information about the changes of operating conditions decrease the uncertainty of decision-making about the faults, it comes at the price of additional instrumentation. At the same time, no knowledge about the operating conditions oftentimes result in poor generalization to different operating points and in consequence low performance in fault detection.

The goal of this work is to explore the use of ML-based feature estimation and classification methods for application to CM of bearings under changing operating conditions. The proposed approach involves extracting the fault-sensitive features from vibration data using an AutoEncoder (AE) architecture [12], followed by a classic support vector machine (SVM) [13] used for classification. To address the changing operating conditions and the sparse amount of the faulty data, a pseudo-anomaly generation scheme [14] is empirically explored.

2. BACKGROUND THEORY

In this section we provide a brief background of the feature extraction with autoencoders, generation of pseudo-anomaly features, and the usage of supporting vector machines for fault classification.

2.1. Feature extraction with autoencoders

AEs provide a powerful approach to feature extraction by learning two functions: an encoding function that transforms the input data to a set of features, and a decoding function that recreates the input data from the encoded representation. By learning these functions, the AE network captures significant data patterns, while filtering out redundant information. The encoded features represent a latent space that preserves the most significant characteristics of the original data, making them plausible candidates to be used in fault diagnosis. Hereafter we provide a brief overview on the AE principles; an interested reader is referred to [12] for a detailed information.

Let $\mathbf{x} \in \mathbb{R}^{h \times w \times c}$ denote an observation array with height h, width w, and c channels. Typically, x represents some time series, or time-frequency data, where for the latter h is related to the number of frequency lines and w to the number of time frames.

The AE consists of two main components, namely the encoder and the decoder. The encoder reduces the spatial dimensions (height and width) while transforming the representation into a higher-dimensional feature space by virtually increasing the number of channels. This process maps the input $\mathbf{x}$ to an encoded representation $\mathbf{z} \in \mathbb{R}^{h' \times w' \times c'}$ (where $h' < h$, $w' < w$, and $c' > c$) using $g(\mathbf{x})$

$$\mathbf{z} = g(\mathbf{x}) = \sigma(\mathbf{W}_e * \mathbf{x} + \mathbf{b}_e), \tag{1}$$

where $\mathbf{W}_e$ denotes the convolutional filters, $\mathbf{b}_e$ is a bias term, σ is a non-linear activation function, and $*$ represents the convolution operation.

The decoder reconstructs the input from the encoded space $\mathbf{z}$ using $h(\mathbf{z})$

$$\hat{\mathbf{x}} = h(\mathbf{z}) = \sigma(\mathbf{W}_d * \mathbf{z} + \mathbf{b}_d), \tag{2}$$

where $\mathbf{W}_d$ denotes the convolutional filters, $\mathbf{b}_d$ is a bias term, σ is an activation function, and $*$ again represents the convolution operation.

The parameters $\mathbf{W}_e, \mathbf{b}_e, \mathbf{W}_d, \mathbf{b}_d$ are learned by solving an optimization problem

$$\underset{\mathbf{W}_e,\mathbf{b}_e,\mathbf{W}_d,\mathbf{b}_d}{\arg\min} \frac{1}{N}\sum_{i=1}^{N} \left\|\mathbf{x}_i - h(g(\mathbf{x}_i))\right\|^2, \tag{3}$$

where $\mathbf{x}_i$ is a subset of $\mathbf{x}$ taken over its width and N is the sample length.

2.2. Pseudo-anomaly generation

Pseudo-anomalies efficiently integrate information from both labeled and unlabeled data. When incorporated to a supervised classification schemes, it can be heuristically shown that the pseudo-anomalies facilitate the detection of new, actual, anomalies [14]. In this context, multiple algorithms to generate pseudo-anomalies exist [14–17]. Hereafter, we recapture a recent NNG-mix scheme [14].

To synthesize a pseudo-anomaly, the process begins by randomly sampling an anomaly instance $\mathbf{a_1}$ from the labeled anomaly set $\mathcal{A}$. To identify a contextually similar counterpart for interpolation, a set of k nearest neighbors $\mathcal{M}$ is retrieved using the k-nearest neighbors (kNN) algorithm. The neighborhood is chosen with equal probability 50%, from either the anomaly set $\mathcal{A}$ or the unlabeled data $\mathcal{H}$. This stochastic selection introduces variability in the mixing process and enables the generation of diverse pseudo-anomalies that may lie closer to the decision boundary. Subsequently, a second sample $\mathbf{a_2}$ is drawn from the resulting neighborhood $\mathcal{M}$. This selective pairing strategy is designed to reduce the risk of generating mislabeled or semantically incoherent samples: an issue commonly associated with naive applications of mixup-style interpolation methods [15].

Gaussian noise $\boldsymbol{\epsilon_1}$ and $\boldsymbol{\epsilon_2}$ drawn from a normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\sigma})$ is then added to $\mathbf{a_1}$ and $\mathbf{a_2}$, respectively. A mixing coefficient λ is sampled from a Beta distribution Beta(α, α), and a pseudo-anomaly $\mathbf{d}$ is generated as a linear combination of the a_1 and a_2: $\mathbf{d} = \lambda\mathbf{a_1} + (1-\lambda)\mathbf{a_2}$. This procedure is summarized in Algorithm 1 [14].

2.3. Classification with Support Vector Machines

Support Vector Machine (SVM) are supervised learning models primarily used for binary classification. They operate by finding a hyperplane that maximizes the margin between two classes. Let $\{(\cdot,\cdot)\}$ denote a data set. Given training data $\{(\mathbf{z}_i, y_i)\}$, $i = 1 \ldots N$, with feature vectors $\mathbf{z}_i \in \mathbb{R}^d$ and binary labels $y_i \in \{-1, 1\}$, a linear SVM defines the decision boundary as:

$$f(\mathbf{z}) = \mathbf{w}^T\mathbf{z} + b = 0, \tag{4}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the normal vector and $b \in \mathbb{R}$ is the bias term. For data that are not linearly separable, a soft-margin SVM introduces slack variables $\xi_i \geq 0$ to allow for misclassifications, leading to the optimization problem:

$$\underset{\mathbf{w},b,\xi}{\arg\min} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i, \;\; \text{subject to} \;\; y_i(\mathbf{w}^T\mathbf{z}_i + b) \geq 1 - \xi_i, \quad \forall i, \tag{5}$$

where $C > 0$ controls the trade-off between maximizing the margin and minimizing classification errors [13]. The conceptual idea of the feature classification with SVM is illustrated in Figure 1.

Algorithm 1 NNG-mix scheme [14]

Input: Anomaly data $\mathcal{A}$, unlabeled data $\mathcal{H}$, generated pseudo-anomaly $\mathcal{D}$, hyper-parameters α, k and $\boldsymbol{\sigma}$.

$\mathcal{D} = \{\}$
for $i = 1, \ldots, N$ **do**
  Sample anomaly $\mathbf{a_1}$ from $\mathcal{A}$
  if Uniform(0, 1) > 0.5 **then**
    $\mathcal{M}$ = kNN($\mathbf{a_1}, \mathcal{A}, k$)
  else
    $\mathcal{M}$ = kNN($\mathbf{a_1}, \mathcal{H}, k$)
  end if
  Sample $\mathbf{a_2}$ from $\mathcal{M}$
  $\boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma})$
  $\mathbf{a_1} = \mathbf{a_1} + \boldsymbol{\epsilon_1}$
  $\mathbf{a_2} = \mathbf{a_2} + \boldsymbol{\epsilon_2}$
  λ = Beta(α, α)
  $\mathbf{d} = \lambda \mathbf{a_1} + (1 - \lambda)\mathbf{a_2}$
  $\mathcal{D} = \{\mathcal{D}; \mathbf{d}\}$
end for
return $\mathcal{D}$

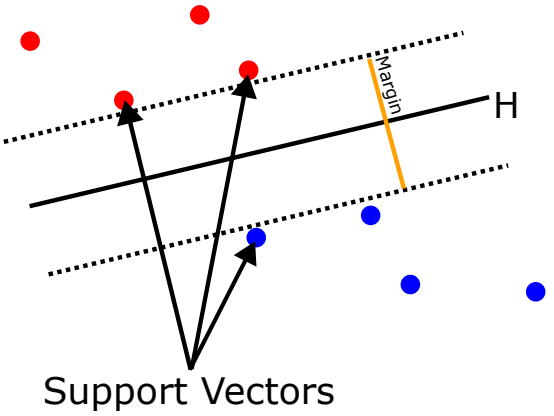


Figure 1: A dummy example of how a Support Vector Machine (SVM) finds a hyperplane, that maximize the margin between straight lines relying on the Support Vectors.

3. APPLICATION

This section is dedicated to the application of the ML methodology outlined in Sections 2.1.-2.3. to vibration data collected from a drivetrain simulator. The central aim is to examine how varying the proportion of faulty data stemming from the pseudo-anomaly injection influences both the stability and generalizability of SVM when confronted with a new, unseen fault conditions.

3.1. Experimental test setup

The ML methodology is applied on an experimental Drivetrain Diagnostics Simulator (DDS) illustrated in Figure 2. The setup consists of a motor with speed/torque controller, a single two-stage parallel-axis gearbox, a single one-stage planetary gearbox, a magnetic break and a set of 7 piezoelectric uniaxial accelerometers located at different mechanical components of the drivetrain. Due to the proximity to the faulty mechanical component, two accelerometers mounted on the horizontal and vertical axes of the bearing 1 are used, see the next paragraph. To emulate changes in the operating operating conditions, varying load a rotational speeds of the drivetrain are considered, i.e., load 0%, 20%, 40% and the rotational frequency of the shaft 10 Hz, 12 Hz, 16 Hz, 18 Hz and 20 Hz. For each operating point, the vibration data is collected with a sampling frequency of 5.12 kHz for 30 minutes. To generate faulty data, these vibration tests are repeated with the drivetrain where the bearing 1 is replaced with a damaged bearing of the same type. For a detailed description of the tests, see [18].

3.2. Training the ML models

The devised ML methodology consists of an AE, used for feature extraction, a pseudo-anomaly generation scheme used for creating sufficient labeled data before training the SVM, and a SVM used for classification. Its block diagram can be viewed in Figure 3, where class 1 represents anomalous instances and class 0 corresponds to normal observations.

To extract the fault-sensitive features, an AE is trained on a subset of the reference data. To verify whether the model has generalization capabilities, the reference data is limited to 50 % of the measured cases, chosen randomly. The data sets used for training are outlined with the ✓symbol in Table 1.

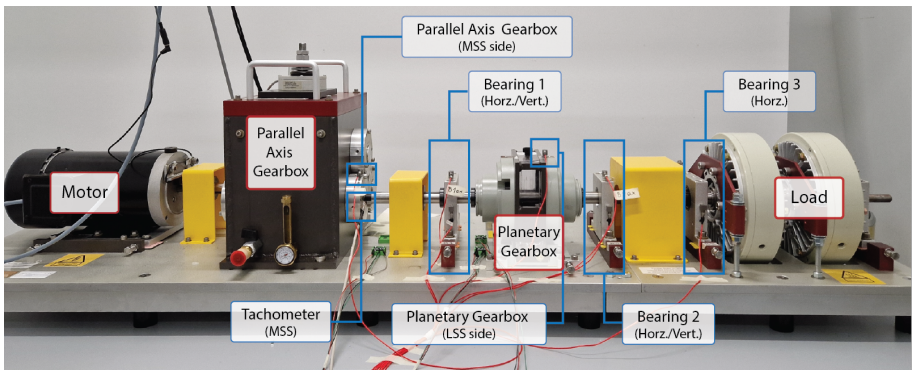


Figure 2: Test rig, after Figure 2.a in [18].

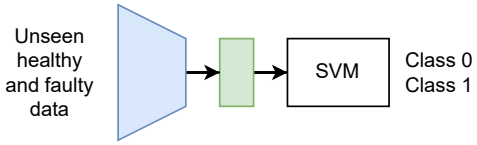


Figure 3: The proposed ML model uses the AE to extract features, and the SVM to predict if data is healthy or faulty.

Table 1: Training data availability at various loads and speeds.

Load/Speed	Frequency					
	10 Hz	12 Hz	14 Hz	16 Hz	18 Hz	20 Hz
0	✓	✓	✓	✓	×	✓
20	×	✓	✓	✓	×	×
40	✓	×	×	×	×	×

The AE is trained with a decoder weaker than the encoder, to ensure that the encoded space has all relevant features. The proposed model is an Short Time Fourier Transform (STFT)-based convolutional autoencoder designed to capture spectral-temporal features of input data represented as the magnitude STFT spectrograms, without the phase information. The spectrograms are zero-padded afterwards, to ensure equal length of all signals, and normalized between [0, 1]. A sketch of the principle behind the AE is illustrated in Figure 4.

The encoder consists of three convolutional layers with increasing channel depths (from 2 channels in the input up to 64 channels). Each convolutional layer uses a kernel size of 3×3, followed by a ReLU activation function. This progressively reduces the spatial dimensions while extracting hierarchical features. The decoder reconstructs the original input from the encoded representation using transposed convolutional layers. It comprises two transposed convolutional layers designed to reverse the dimensionality reduction performed by the encoder. The first decoder layer upsamples aggressively, using a stride of 4, followed by another transposed convolutional layer with a stride of 2. Both layers employ a kernel size of 3×3 and carefully selected output padding parameters ((0, 3) and (0, 1), respectively) to match the original input dimensions. The decoder utilizes ReLU activations, except the final layer, which employs a Sigmoid activation function to map outputs into the normalized input data range.

For the training process, an arbitrary batch size of 32 is chosen, and 20 % of the training dataset is used for validation. The depth and accompanying hyper parameters are all arbitrarily chosen.

The encoder is used to make new datasets, containing the feature space from the labeled healthy data, the labeled faulty data and the unlabeled data, where, the considered labeled faulty data correspond to the cases marked with ✓ in Table 2. It is assumed that the unlabeled data contains all running conditions with both healthy and faulty data. In real life scenarios, the availability of faulty data is limited. To

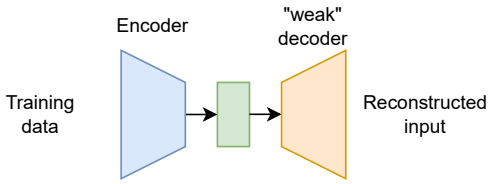


Figure 4: A simplified figure of an AutoEncoder.

emulate this, only a subset of the faulty data is used, that is the vibration of 12 full rotations of the shaft.

Table 2: Training data availability at various loads and speeds.

Load/Speed	Frequency					
	10 Hz	12 Hz	14 Hz	16 Hz	18 Hz	20 Hz
0	✓	×	×	✓	×	×
20	×	×	×	×	✓	×
40	✓	×	×	✓	✓	×

In order to ensure equally large datasets for faulty and healthy data, the pseudo-anomaly scheme described in [14] is used. Lastly, the data is standardized, as SVMs are sensitive to the scale of input features.

The process of creating data for SVM is shown in Figure 5. Now, the AE combined with SVM is used to classify whether the tested data set is faulty (Class 1), or healthy (Class 0), as shown in Figure 3.

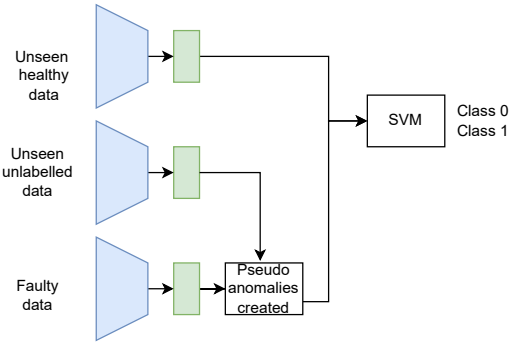


Figure 5: A graphical overview of how the datasets are used to train the SVM based on the features from the pre-trained AE and the pseudo anomaly scheme in [14].

3.3. Results

Two different experiments are carried out where, the SVM is trained without and with the pseudo anomalies. The encoded space for the test data with the corresponding cluster tendencies are illustrated in Figure 6.

The data used for the experiments contain different operating conditions, both in a healthy and faulty states. Class 0 is defined as healthy data, where Class 1 is defined as outliers. In Figure 7 the confusion matrix for both experiments are depicted. The classification performance was analyzed by considering True Positive Rate (TPR) (sensitivity), False Positive Rate (FPR), True Negative Rate (TNR) (specificity), and False Negative Rate (FNR). The detailed results are summarized in Table 3.

The inclusion of pseudo-anomalies during training demonstrably enhances the model's ability to detect outliers, as reflected in the quantitative performance metrics. Table 3 shows a notable increase in true

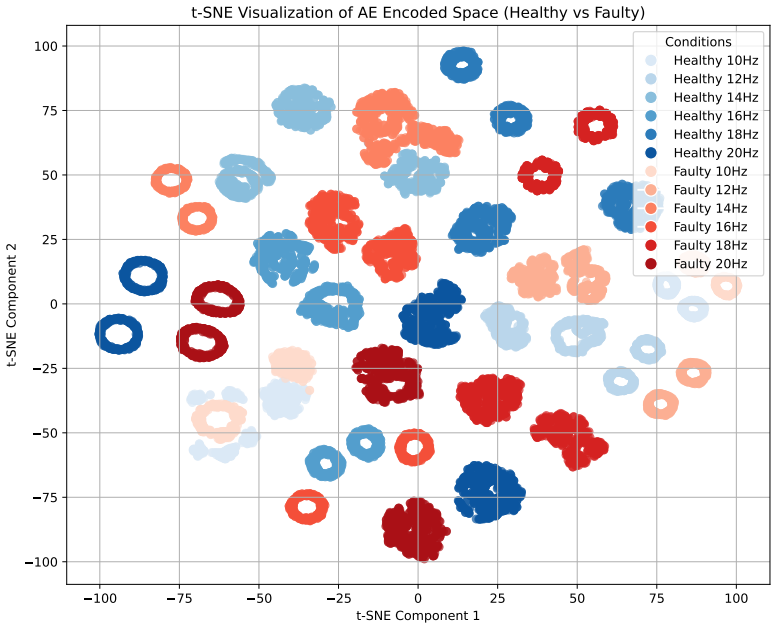


Figure 6: A 2 dimensional plot of the encoded space obtained on test data, with the pre-trained AE. The plot is reduced to 2 dimensions using t-SNE.

positive rate—from 59% without pseudo-anomalies to 89% with them indicating a substantial reduction in undetected faults. The total accuracy improves similarly, from 73% to 89%, while precision rises modestly from 84% to 88%, suggesting that the classifier becomes not only more sensitive but also more reliable in its predictions. Importantly, these gains come with only a marginal increase in the false positive rate, from 11% to 12%, highlighting that the improved sensitivity does not come at the cost of mistakenly flagging healthy data as anomalous.

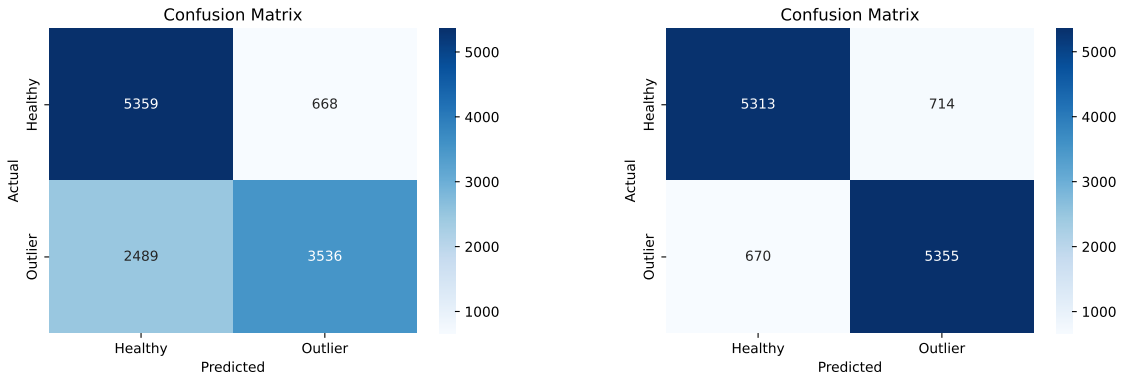
Table 3: Classification Performance Metrics

Metric	without pseudo-anomalies	with pseudo-anomalies
True Positive Rate (Sensitivity)	59%	89%
False Positive Rate	11%	12%
True Negative Rate (Specificity)	89%	88%
False Negative Rate	41%	11%
Total accuracy	73%	89%
Total precision	84%	88%

The confusion matrices illustrated in Figure 7 further reinforce these observations. Without pseudo-anomalies, the model fails to identify a large portion of true outliers, misclassifying approximately 2,489 faulty samples as healthy, which is depicted in Figure 7a. In contrast, when pseudo-anomalies are introduced, the misclassifications number drops dramatically to 670, as shown in Figure 7b, confirming the effectiveness of this strategy in reducing false negatives.

Finally, the t-SNE visualization of the AEs encoded feature space, depicted in Figure 6 illustrates a clear separation between the healthy and the faulty operating conditions across the different operation frequencies. This structural distinction in the latent space suggests that the model has learned meaningful features capable of capturing the underlying variation between classes.

Collectively, this demonstrate that augmenting the training set with pseudo-anomalies significantly improves the model's ability to generalize to previously unseen faults, enhancing both sensitivity and overall



(a) Without pseudo-anomalies

(b) With pseudo-anomalies

Figure 7: Confusion matrices for both the SVM trained without (a) and with (b) the pseudo-anomaly injections

robustness without materially compromising specificity.

4. CONCLUSIONS

In this paper we have explored an application of ML methodology to detect faults in a drivetrain simulator under changing load and rotating speeds. For this purpose, an AE was trained solely on healthy reference data to extract fault-sensitive features from vibration signals and a pseudo-anomaly generation scheme was used to balance the healthy data and the faulty data before training the SVM. The results illustrate that the application of the pseudo-anomaly generation has a positive impact both on detectability of faults and the reduction in the false alarm rate. What's more, the achieved fault detection rates surpass the current benchmark results showing the applicative potential of the proposed methodology.

ACKNOWLEDGMENTS

Innovation Fund Denmark is hereby acknowledged for funding this research under the BeWind project, grant 3148-00015A.

A warm acknowledgment is dedicated to Andriana Georgantopoulou for performing and sharing the experimental measurements, and providing insights to the data.

REFERENCES

[1] Jiayu Chen, Cuiying Lin, Di Peng, and Hongjuan Ge. Fault diagnosis of rotating machinery: A review and bibliometric analysis. *Ieee Access*, 8:224985–225003, 2020.

[2] Qinghua Zhang, Lennart Ljung, and Rik Pintelon. On local lti model coherence for lpv interpolation. *IEEE Transactions on Automatic Control*, 65(8):3671–3676, 2020.

[3] N. Dervilis, K. Worden, and E.J. Cross. On robust regression analysis as a means of exploring environmental and operational conditions for shm data. *Journal of Sound and Vibration*, 347:279–296, 2015. ISSN 0022-460X.

[4] Callum Roberts, Luis David Avendaño-Valencia, and David García Cava. Robust mitigation of eovs using multivariate nonlinear regression within a vibration-based shm methodology. *Mechanical Systems and Signal Processing*, 208:111028, 2024. ISSN 0888-3270.

[5] Luis David Avendaño-Valencia, Eleni N. Chatzi, and Dmitri Tcherniak. Gaussian process mod-

els for mitigation of operational variability in the structural health monitoring of wind turbines. *Mechanical Systems and Signal Processing*, 142:106686, 2020. ISSN 0888-3270.

[6] Yaguo Lei and Ming J Zuo. Gear crack level identification based on weighted k nearest neighbor classification algorithm. *Mechanical Systems and Signal Processing*, 23(5):1535–1547, 2009.

[7] Dong Wang. K-nearest neighbors based methods for identification of different gear crack levels under different motor speeds and loads: Revisited. *Mechanical Systems and Signal Processing*, 70: 201–208, 2016.

[8] Haichen Shi, Keith Worden, and Elizabeth J. Cross. A regime-switching cointegration approach for removing environmental and operational variations in structural health monitoring. *Mechanical Systems and Signal Processing*, 103:381–397, 2018. ISSN 0888-3270.

[9] Szymon Gres, Michael Döhler, Palle Andersen, and Laurent Mevel. Subspace-based mahalanobis damage detection robust to changes in excitation covariance. *Structural Control and Health Monitoring*, 28(8):e2760, 2021.

[10] Michael Döhler and Laurent Mevel. Subspace-based fault detection robust to changes in the noise covariances. *Automatica*, 49(9):2734–2743, 2013. ISSN 0005-1098.

[11] Ming Yang and Viliam Makis. Arx model-based gearbox fault detection and localization under varying load conditions. *Journal of Sound and Vibration*, 329(24):5209–5221, 2010.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[13] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA, 2006. ISBN 978-0-387-31073-2.

[14] Hao Dong, Gaëtan Frusque, Yue Zhao, Eleni Chatzi, and Olga Fink. Nng-mix: Improving semi-supervised anomaly detection with pseudo-anomaly generation, 2024. URL `https://arxiv.org/abs/2311.11961`.

[15] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. URL `https://arxiv.org/abs/1710.09412`.

[16] Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017. URL `https://arxiv.org/abs/1708.04552`.

[17] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019. URL `https://arxiv.org/abs/1905.04899`.

[18] Andriana Georgantopoulou. Time series analysis for condition monitoring under operational variability, 2023.