Data Augmentation of Sensor Time Series using Time-varying Autoregressive Processes

Douglas Baptista de Souza[1] and Bruno Paes Leao[2]

[1] *Siemens Advanta, 4800 North Point Pkwy, Alpharetta, GA, 30022, United States*
douglas.de-souza@siemens.com

[2] *Siemens Technology, Princeton, NJ, 08540, United States*
bruno.leao@siemens.com

ABSTRACT

This work presents a novel data-centric solution for fault diagnostics and failure prognostics consisting of a data-augmentation method which is well suited for non-stationary mutivariate time-series data. The method, based on time-varying autoregressive processes, can be employed to extract key information from a limited number of samples and generate new artificial samples in a way that benefits the development of diagnostics and prognostics solutions. The proposed approach is tested based on three real-world datasets associated with failure diagnostics problems using two types of machine learning methods. Results indicate the proposed method improves performance in all tested cases.

1. INTRODUCTION

Fault diagnostics and failure prognostics of equipment and systems, a.k.a. PHM, predictive maintenance, etc., have been the focus of active investigation and intense real world solution development during the recent decades. The motivation is clear, as avoiding the occurrence of failures or reducing their consequences in general translates in benefits such as reduced downtime, improved yield and safety. A large variety of methods have been proposed over the years, ranging from reliability methods based on population statistics to data-driven solutions employing cutting-edge machine learning methods, or physics-based approaches incorporating advanced failure mechanism models. However, it can be argued that the most important factors limiting the successful development and application of PHM solutions are not in the methods themselves. Those limitations are in general related to the availability of good quality historical data which can be used for the development and validation of such solutions (Biggio & Kastanis, 2020; Kim, Choi, & Kim, 2021). Failure events can be very rare but their impact can be so relevant that it is worth investing in development of related PHM solutions. Many times the failure events of interest may have happened a reasonable number of times in the past such that available data from those events would be enough for development of related PHM solutions. However, many times, relevant sensor data associated to the historical events has not been collected or, if collected, corresponding labels are not available, or if available they are not reliable or not detailed enough, e.g., defining the actual failure mode.

Such data limitations naturally affect more directly data-driven methods, but reliability-based and even physics-based methods may also be impacted. The former because reliability solutions are also based on historical data, despite simplifications such as the assumption of parametric statistical models or the combination of data from different but similar equipment. The latter is also impacted, although to a lesser extent, as physics-based diagnostics and prognostics solutions must also be validated based on sufficient real historical data before they can be deployed to the field. Whenever enough data is not available, despite the extensive literature related to PHM methods, the development and validation of diagnostics or prognostics solution may result in poor performance or may not even be feasible. In such cases, the possibilities for PHM solution development are limited to anomaly detection which are usually not as prescriptive or actionable as fault diagnostics and failure prognostics.

Considering the points above, data-centric approaches to PHM can be valuable in terms of addressing the relevant data-related challenges faced in real world both for diagnostics (Leao, Fradkin, Lan, & Wang, 2021) and prognostics (Garan, Tidriri, & Kovalenko, 2022). However, focusing on a data-centric PHM still represents a broad set of possibilities ranging from improved data collection and labeling to approaches for making the most out of available data. Among those possibilities, data augmentation methods have gained

ever increasing attention over the recent years also considering both diagnostics (Matei, Zhenirovskyy, de Kleer, & Feldman, 2018; Kwak & Lee, 2023) and prognostics (Kim, Kim, & Choi, 2020) applications. In particular, data augmentation methods have improved the performance of diagnostics approaches with limited data in different PHM use cases (Yang et al., 2023; D. Wang, Dong, Wang, & Tang, 2023; Li, Zhang, & Zhang, 2023; Jiang & Ge, 2021; Shen, Yao, Jiang, Yang, & Zeng, 2023; de Oliveira, Niemi, García-Ortiz, & Torres, 2023; Taghiyarrenani & Berenji, 2022).

This work proposes a novel method for augmentation of multivariate time-series data based on time-varying autoregressive (TVAR) models. The goal is to extract information from scarce data and use it to create additional samples in a way that can improve the quality of diagnostics and prognostics solutions. One characteristic which makes it especially suited for failure diagnostics and prognostics problems is the fact that it can directly deal with non-stationary time series.

The remaining of this paper is organized as follows: section 2 contains the technical background related to TVAR; in section 3 the proposed application of TVAR for data augmentation is presented; experiments and results are described and discussed in section 4; section 5 is the conclusion.

2. BACKGROUND ELEMENTS

2.1. Considerations About The Time Series Data

This paper addresses the problem of augmenting sensor time series from the viewpoint of time-series groups or classes. To this end, we consider that the data to be analyzed are divided into classes like "anomalous" and "normal". Furthermore, the following assumptions are considered for the time-series datasets to be augmented by the proposed method:

A1) Time series of the same dataset are sampled equally.
A2) Time series of the same class can be modeled by the same mutivariate, potentially non-stationary stochastic process.

The potential non-stationary behavior means that the properties of the stochastic process are allowed to vary over time. The proposed data augmentation derives from assumption A2), and consists in finding a suitable stochastic process to model the time series of a given class, then using it to create new time series as new realizations of the stochastic process.

2.2. Considerations About The Time Series Models

In this work, we use parametric time series models to represent the stochastic processes characterizing the signal classes. These models describe mathematically how the samples and stochastic moments of the time series vary over time time. Examples of well-known models are Autoregressive (AR), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA) (Deistler & Scherrer, 2022). Such models often require the data to be stationary, or the potential nonstationarity in the data to follow specific forms (e.g., trend and seasonality), so it can be extracted from the data through subtraction, differencing or other transformations (Box, Jenkins, Reinsel, & Ljung, 2015). However, real-world time series often show a variety of non-stationary behaviors that cannot be modeled by traditional approaches (de Souza, Chanussot, Favre, & Borgnat, 2012, 2014, 2018, 2019; de Souza, Chanussot, & Favre, 2014).

Unlike classical approaches, time-varying autoregressive (TVAR) models allow their parameters to change over time to better capture the nonstationarities in the data (Kay, 2008). This change is controlled by a set of TVAR basis functions, weights, and the model order. Based on how these quantities are chosen, different non-stationary behaviors can be modeled (Niedzwiecki, 2000). Owing to these powerful modeling capabilities, we choose to characterize the underlying stochastic process of the time series with a TVAR model, meaning that time series belonging to the same class will share the same TVAR expression (see A2). Different customizations to the TVAR basis functions and parameters have been proposed for modeling various types of non-stationary data (e.g., acoustic signals (Sodsri, 2003), electroencephalography (EEG) (Pachori & Sircar, 2008), and radar clutter data (Abramovich, Spencer, & Turley, 2007)). Here, we choose the customization proposed in (de Souza, Kuhn, & Seara, 2019) and use it as a startpoint to derive the data-augmentation procedure.

2.3. TVAR Model

In (de Souza, Kuhn, & Seara, 2019), a TVAR process of first-order has been proposed to model non-stationary processes whose mean and covariance vary over time following an arbitrary functional form defined by the user. The non-stationary processes characterized by the TVAR model are M-dimensional vectors $\mathbf{x}(n)$ (multivariate time series) varying in time according to the following regressive expression:

$$\mathbf{x}(n+1) = a(n)\mathbf{x}(n) + b(n)\mathbf{v}(n) \tag{1}$$

with n as time variable, $\mathbf{v}(n) \in \mathbb{R}^M$ as the perturbation noise, and $a(n)$ and $b(n)$ as time-varying parameters given by the weighted sum of $f_0(n), ..., f_Q(n)$ scalar basis functions

$$a(n) = \sum_{q=0}^{Q} \alpha_q f_q(n) \quad \text{and} \quad b(n) = \sum_{q=0}^{Q} \beta_q f_q(n) \tag{2}$$

having $\alpha_0, ..., \alpha_Q$ and $\beta_0, ..., \beta_Q$ as weights. Also, two assumptions are considered for the TVAR model in Eq. (1):

A3) The initial value of the process $\mathbf{x}(0)$ is an arbitrary free parameter defined by the user.
A4) the noise vector $\mathrm{v}(n)$ is stationary with zero mean and

correlation matrix $\mathbf{\Phi} = \mathbb{E}\left[\mathrm{v}(n)\mathrm{v}^{\mathrm{T}}(n)\right]$. Also, it has uncorrelated samples for $n \neq m$, i.e., $\mathbb{E}\left[\mathrm{v}(n)\mathrm{v}^{\mathrm{T}}(m)\right] = \mathbf{0}$ if $n \neq m$.

Considering the assumptions above and Eq. (2), one can obtain the following general expressions for the mean vector and covariance matrix of $\mathbf{x}(n)$ in Eq. (1), respectively:

$$\mathbf{m}(n+1) = \prod_{\ell=0}^{n} a(\ell)\mathbf{x}(0) \tag{3}$$

and

$$\mathbf{C}(n+1) = \left\{ b^2(n) + \sum_{w=0}^{n-1} \left[\prod_{\ell=w+1}^{n} a^2(\ell) b^2(\ell) \right] \right\} \mathbf{\Phi} \tag{4}$$

whereas the non-stationary behavior of $\mathbf{x}(n)$ can be verified by the fact that Eqs. (3) and (4) are functions of n. The basis functions and constants in Eq. (2) are often customized so the TVAR model can describe different non-stationary evolutions. One customization for Eq. (2) proposed in (de Souza, Kuhn, & Seara, 2019) enables the mean and covariance in Eqs. (3) and (4) to converge at different rates to a predetermined functional form. This customization is obtained by making $Q = \gamma_0 = \beta_1 = 1$ and $\gamma_1 = \beta_0 = 0$ in Eq. (2), and choosing the following expressions for the TVAR basis functions:

$$f_0(n) = \frac{p(n+1)\mathrm{e}^{r_1^{n+1}}}{p(n)\mathrm{e}^{r_1^{n}}} \tag{5}$$

and

$$f_1(n) = \lambda p(n+1)\mathrm{e}^{r_1^{n+1}} \sqrt{\frac{(1-r_2^{n+1})^2}{\mathrm{e}^{2r_1^{n+1}}} - \frac{(1-r_2^{n})^2}{\mathrm{e}^{2r_1^{n}}}} \tag{6}$$

where $p(n)$ in an arbitrary functional form characterizing the time-varying behavior of the mean vector and covariance matrix. Here, $p(n)$ is called the TVAR interpolation function. Moreover, r_1, r_2, and λ are real constants meeting the following requirements: R1) $0 < \{r_1, r_2\} < 1$ and R2) $\lambda \neq 0$. By substituting the values of Q, α_0, α_1, β_0, β_1, and the expressions for $f_0(n)$ and $f_1(n)$ into Eqs. (2) and (1), we get

$$\begin{aligned} \mathbf{x}(n+1) = {} & \frac{p(n+1)\mathrm{e}^{r_1^{n+1}}}{p(n)\mathrm{e}^{r_1^{n}}}\mathbf{x}(n) + \lambda p(n+1)\mathrm{e}^{r_1^{n+1}} \\ & \times \sqrt{\frac{(1-r_2^{n+1})^2}{\mathrm{e}^{2r_1^{n+1}}} - \frac{(1-r_2^{n})^2}{\mathrm{e}^{2r_1^{n}}}}\mathbf{v}(n). \end{aligned} \tag{7}$$

By using the same substitutions in Eqs. (3) and (4), it can be shown that the following expressions for the mean vector and covariance matrix of Eq. (7) can be obtained:

$$\mathbf{m}(n+1) = \frac{\mathrm{e}^{-1}}{p(0)} p(n+1)\mathrm{e}^{r_1^{n+1}}\mathbf{x}(0) \tag{8}$$

and

$$\mathbf{C}(n+1) = \lambda^2 p^2(n+1)\left(1 - r_2^{n+1}\right)^2 \mathbf{\Phi}. \tag{9}$$

Since the value of $\mathbf{x}(0)$ in Eq. (8) is arbitrary (see A3), one can define it as $\mathbf{x}(0) = \gamma\tilde{\mathbf{x}}(0)$, where γ is an arbitrary real constant and $\tilde{\mathbf{x}}(0)$ is a basis vector. Then, Eq. (8) becomes

$$\mathbf{m}(n+1) = \gamma \frac{\mathrm{e}^{-1} p(n+1)}{p(0)} \mathrm{e}^{r_1^{n+1}} \tilde{\mathbf{x}}(0) \tag{10}$$

where γ is assumed to meet requirement R3) $\gamma \neq 0$. By looking at Eqs. (10) and (9), it can be seen that parameters γ and λ are simply constant gains, while r_1 and r_2 control the convergence of the mean and covariance towards their steady-state expressions (i.e., for large n) given by

$$\mathbf{m}_{\mathrm{ss}}(n+1) = \gamma \frac{\mathrm{e}^{-1} p(n+1)}{p(0)} \tilde{\mathbf{x}}(0) \tag{11}$$

and

$$\mathbf{C}_{\mathrm{ss}}(n+1) = \lambda^2 p^2(n+1)\mathbf{\Phi}. \tag{12}$$

Note that Eqs. (11) and (12) both depend on the TVAR interpolation function $p(n)$, which is the most important quantity to be found. Hence, in this paper, we focus on obtaining $p(n)$ and consider γ, λ, r_1, and r_2 as fine-tuning parameters. Details on how to calculate the TVAR interpolation function and the proposed data augmentation method are given next.

3. DATA AUGMENTATION WITH TVAR

3.1. Overview of the Proposed Method

Here, we use the TVAR model to represent the underlying stochastic process of the time series belonging to a given class (see A2). We consider that such a stochastic process can be described by empirical statistics computed from the data. Thus, fitting the TVAR model amounts to finding parameters and interpolation functions that make the expressions for the first- and second-order moments of the TVAR process match the empirical statistics calculated for the data[1]. Having found suitable TVAR parameters and interpolation function, we plug them back in the TVAR process formula (see (7)) to create randomized augmented data for a given signal class.

3.2. TVAR Sub-Models for the Mean and Covariance

As $p(n)$ appears both in the mean and in the covariance expressions (see Eqs. (10) and (9)), we cannot resort to one TVAR model (based on a single $p(n)$) to describe time series in which mean and covariance change in distinct forms over time. To account for these cases, we propose to use two TVAR sub-models to characterize the underlying stochastic process of the signal class, one for the mean ($\mathrm{TVAR_m}$) and another one for the covariance ($\mathrm{TVAR_c}$). We let each sub-model

[1]Thus, this case differs from the usual time series model fitting paradigm, where model parameters are adjusted to match single time series.

to have its own interpolated expressions $p_{\mathrm{m}}(n)$ and $p_{\mathrm{c}}(n)$, as well as parameters $\{\gamma_{\mathrm{m}}, \lambda_{\mathrm{m}}, r_{1_{\mathrm{m}}}, r_{2_{\mathrm{m}}}\}$ and $\{\gamma_{\mathrm{c}}, \lambda_{\mathrm{c}}, r_{1_{\mathrm{c}}}, r_{2_{\mathrm{c}}}\}$, which should be initialized with the sub-models and meet R1) to R3). These constants can be fine-tuned with Eqs.(10) and (9) to match the non-stationary statistics of the data (See Section 3.3.1). We consider that the $\mathrm{TVAR_m}$ and $\mathrm{TVAR_c}$ sub-models capture the first- and second-order dynamics of the time series of a given class, respectively. The final augmented signals are obtained by using the $p_{\mathrm{m}}(n)$ and $p_{\mathrm{c}}(n)$ and parameters for $\mathrm{TVAR_m}$ and $\mathrm{TVAR_c}$ in (7), which will lead to Eqs. (13) and (14), respectively, both of which are used to generated the augmented (new) synthetic time series.

$$\mathbf{x}_{\mathrm{m}}(n+1) = \frac{p_{\mathrm{m}}(n+1)\mathrm{e}^{r_{1_{\mathrm{m}}}^{n+1}}}{p_{\mathrm{m}}(n)\mathrm{e}^{r_{1_{\mathrm{m}}}^{n}}}\mathbf{x}_{\mathrm{m}}(n) + \lambda_{\mathrm{m}} p_{\mathrm{m}}(n+1)\mathrm{e}^{r_{1_{\mathrm{m}}}^{n+1}} \times \sqrt{\frac{(1-r_{2_{\mathrm{m}}}^{n+1})^2}{\mathrm{e}^{2r_{1_{\mathrm{m}}}^{n+1}}} - \frac{(1-r_{2_{\mathrm{m}}}^{n})^2}{\mathrm{e}^{2r_{1_{\mathrm{m}}}^{n}}}}\mathbf{v}(n). \tag{13}$$

$$\mathbf{x}_{\mathrm{c}}(n+1) = \frac{p_{\mathrm{c}}(n+1)\mathrm{e}^{r_{1_{\mathrm{c}}}^{n+1}}}{p_{\mathrm{c}}(n)\mathrm{e}^{r_{1_{\mathrm{c}}}^{n}}}\mathbf{x}_{\mathrm{c}}(n) + \lambda_{\mathrm{c}} p_{\mathrm{c}}(n+1)\mathrm{e}^{r_{1_{\mathrm{c}}}^{n+1}} \times \sqrt{\frac{(1-r_{2_{\mathrm{c}}}^{n+1})^2}{\mathrm{e}^{2r_{1_{\mathrm{c}}}^{n+1}}} - \frac{(1-r_{2_{\mathrm{c}}}^{n})^2}{\mathrm{e}^{2r_{1_{\mathrm{c}}}^{n}}}}\mathbf{v}(n). \tag{14}$$

The $\mathrm{TVAR_m}$ and $\mathrm{TVAR_c}$ processes in Eqs. (13) and (14) will more likely represent behaviors captured by the mean and covariance, respectively. The augmented signals have length N and are computed L times, with N and L being free parameters of the method. The mean vector and covariance matrix of $\mathbf{x}_{\mathrm{m}}(n)$ are shown in Eqs. (15) and (16), while Eqs. (17) and (18) show the same quantities for $\mathbf{x}_{\mathrm{c}}(n)$, respectively.

$$\mathbf{m}_{\mathrm{m}}(n+1) = \gamma_{\mathrm{m}}\frac{\mathrm{e}^{-1}p_{\mathrm{m}}(n+1)}{p_{\mathrm{m}}(0)}\mathrm{e}^{r_{1_{\mathrm{m}}}^{n+1}}\tilde{\mathbf{x}}_{\mathrm{m}}(0) \tag{15}$$

$$\mathbf{C}_{\mathrm{m}}(n+1) = \lambda_{\mathrm{m}}^2 p_{\mathrm{m}}^2(n+1)\left(1-r_{2_{\mathrm{m}}}^{n+1}\right)^2\mathbf{\Phi}. \tag{16}$$

$$\mathbf{m}_{\mathrm{c}}(n+1) = \gamma_{\mathrm{c}}\frac{\mathrm{e}^{-1}p_{\mathrm{c}}(n+1)}{p_{\mathrm{c}}(0)}\mathrm{e}^{r_{1_{\mathrm{c}}}^{n+1}}\tilde{\mathbf{x}}_{\mathrm{c}}(0) \tag{17}$$

$$\mathbf{C}_{\mathrm{c}}(n+1) = \lambda_{\mathrm{c}}^2 p_{\mathrm{c}}^2(n+1)\left(1-r_{2_{\mathrm{c}}}^{n+1}\right)^2\mathbf{\Phi}. \tag{18}$$

Since we search analytical expressions for the augmented time series (like Eq. (7)), finding the TVAR interpolating functions stands for interpolating regression expressions for $p_{\mathrm{m}}(n)$ and $p_{\mathrm{c}}(n)$ over the empirical statistics calculated from the data. Next, we address the calculation of the empirical statistics from the data and the TVAR interpolation functions.

3.3. Interpolation Functions and Empirical Statistics

3.3.1. Computing the Empirical Statistics

The interpolation functions $p_{\mathrm{m}}(n)$ and $p_{\mathrm{c}}(n)$ are fitted to the empirical first- and second-order statistics computed from the data. As we assume the time series to be equally-sampled (see A1), we propose to compute the empirical mean vector and covariance matrix of the signals by means of ensemble averaging (Manolakis, Ingle, & Kogon, 2005). More precisely, let us define the collection of equally-sampled, multivariate time series $\mathbf{Y}_j^{(g)}$ belonging to a given class or group g as follows:

$$\mathcal{C}_g = \{\mathbf{Y}_j^{(g)}\}_{j=1,\ldots,J} \tag{19}$$

where the time series $\mathbf{Y}_j^{(g)}$ have length N and dimension M

$$\mathbf{Y}_j^{(g)} = [\mathbf{y}_j^{(g)}(0), ..., \mathbf{y}_j^{(g)}(N-1)]^{\mathrm{T}} \text{ with } \mathbf{y}_j^{(g)}(n) \in \mathbb{R}^M. \tag{20}$$

Based on $\mathcal{C}_g$, the empirical mean vector and covariance matrix at time n can be computed via ensemble averaging as

$$\overline{\mathbf{m}}^{(g)}(n) = \frac{1}{J}\sum_{j=1}^{J}\mathbf{y}_j^{(g)}(n) \tag{21}$$

and

$$\overline{\mathbf{C}}^{(g)}(n) = \frac{1}{J}\sum_{j=1}^{J}\left[\mathbf{y}_j^{(g)}(n) - \overline{\mathbf{m}}^{(g)}(n)\right]\left[\mathbf{y}_j^{(g)}(n) - \overline{\mathbf{m}}^{(g)}(n)\right]^{\mathrm{T}}. \tag{22}$$

3.3.2. Finding the Interpolation Functions

For the sake of simplicity, to present the interpolation method, we consider the time series datasets to be univariate (i.e., we make $M = 1$ in Eq. (20)). Also, we drop the superscript (g) characterizing the group or class of the time series. By doing so, Eqs. (21) and (22) become the univariate sequences $\overline{m}(n)$ and $\overline{c}(n)$, respectively. The results of this section can be extended to the multivariate case without loss of generality[2].

As explained in Section 3.2, sub-models $\mathrm{TVAR_m}$ and $\mathrm{TVAR_c}$ characterize the first- and second-order dynamics of the data separately. Therefore, $p_{\mathrm{m}}(n)$ is interpolated solely to $\overline{m}(n)$, and $p_{\mathrm{c}}(n)$ to $\overline{c}(n)$. Ahead, we address the calculation of the interpolation functions $p_{\mathrm{m}}(n)$ and $p_{\mathrm{c}}(n)$ by considering one approach based on sinusoidal regression via Discrete Fourier Transform (DFT), which allows modeling a wide range of sensor time series. However, other curve fitting techniques could be considered, like splines (de Boor, 2001) and other

[2]To assign the m^{th} element of the multivariate arrays to their corresponding scalar parameters and interpolation functions, one can use diagonal matrices to replace the TVAR scalar quantities (e.g., $\mathrm{diag}([p_{\mathrm{m},0}(n), ..., p_{\mathrm{m,M}}(n)])$ for $p_{\mathrm{m}}(n)$), where $\mathrm{diag}(\cdot)$ is the diagonal operator.

forms of linear (Montgomery, Peck, & Vining, 2006) and nonlinear regressions (Bates & Watts, 2007), as long as they return an analytical expression for the fitted curves.

3.3.3. Sinusoidal Regression

Many PHM time series have an oscillatory nature (e.g., vibration and electrical signals), which can be better characterized by a sinusoidal model. A simple sinusoidal regression for $p_{\mathrm{m}}(n)$ and $p_{\mathrm{c}}(n)$ can be obtained by computing a DFT-based decomposition of $\overline{m}(n)$ and $\overline{c}(n)$. Ahead, we show the steps to compute the regression expression only for $p_{\mathrm{m}}(n)$, as the results for $p_{\mathrm{c}}(n)$ will be the same (the only difference being replacing $\overline{m}(n)$ by $\overline{c}(n)$ in the calculations). Let the DFT of $\overline{m}(n)$ at discrete frequency $\omega_k = 2\pi k$ be

$$F_m(\omega_k) = \sum_{n=0}^{N-1} \overline{m}(n) \mathrm{e}^{-i\omega_k n/N}. \quad (23)$$

By expressing $F_m(\omega_k)$ in terms of its magnitude $|F_m(\omega_k)|$ and phase $\phi_m(\omega_k)$, such that $F_m(\omega_k) = |F_m(\omega_k)|\mathrm{e}^{i\phi_m(\omega_k)}$, and by computing Eq. (23) for $\omega_0, ..., \omega_{N-1}$ (i.e., for all the N available time points), we can build the vector

$$\mathbf{f}_m = [|F_m(\omega_0)|\mathrm{e}^{i\phi_m(\omega_0)}, ..., |F_m(\omega_{N-1})|\mathrm{e}^{i\phi_m(\omega_{N-1})}] \quad (24)$$

with the magnitude and phase spectra of $\overline{m}(n)$ for frequency values $\omega_0, ..., \omega_{N-1}$. Note that we can reconstruct $\overline{m}(n)$ from Eq. (24) by computing the inverse DFT, i.e.,

$$\overline{m}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{f}_m(k) \mathrm{e}^{i\omega_k n/N}. \quad (25)$$

Let us sort the vector $\mathbf{f}_m$ in descending order according to the values of magnitude, obtaining $\mathbf{f}'_m$ as sorted vector

$$\mathbf{f}'_m = [|F_m(\omega'_0)|\mathrm{e}^{i\phi_m(\omega'_0)}, ..., |F_m(\omega'_{N-1})|\mathrm{e}^{i\phi_m(\omega'_{N-1})}] \quad (26)$$

where ω'_k is the sorted frequency variable, so that largest and smallest values of magnitude in $\mathbf{f}'_m$ are at ω'_0 and ω'_{N-1}, respectively. If we recompute Eq. (25) by using the values of the sorted vector in Eq. (26) up to the P^{th} element, we get

$$\overline{m}'(n) = \frac{1}{N} \sum_{k=0}^{P-1} |F_m(\omega'_k)|\mathrm{e}^{i\phi_m(\omega'_k)} \mathrm{e}^{i\omega'_k n/N} \quad (27)$$

as an approximation of $\overline{m}(n)$ with the P most significant frequencies of the spectrum (here, described by the P largest values of magnitude). By expressing (27) in terms of cosines and sines and taking the real part of the resulting expression, the final sinusoidal regression formula for $p_{\mathrm{m}}(n)$ is obtained

$$p_{\mathrm{m}}(n) = \frac{1}{N} \sum_{k=0}^{P-1} |F_m(\omega'_k)| \cos[\omega'_k n/N + \phi_m(\omega'_k)]. \quad (28)$$

Finally, if we repeat the steps above for $\overline{c}(n)$, with $|F_c(\omega'_k)|$ being its sorted magnitude, we get the expression for $p_{\mathrm{c}}(n)$

$$p_{\mathrm{c}}(n) = \frac{1}{N} \sum_{k=0}^{P-1} |F_c(\omega'_k)| \cos[\omega'_k n/N + \phi_c(\omega'_k)]. \quad (29)$$

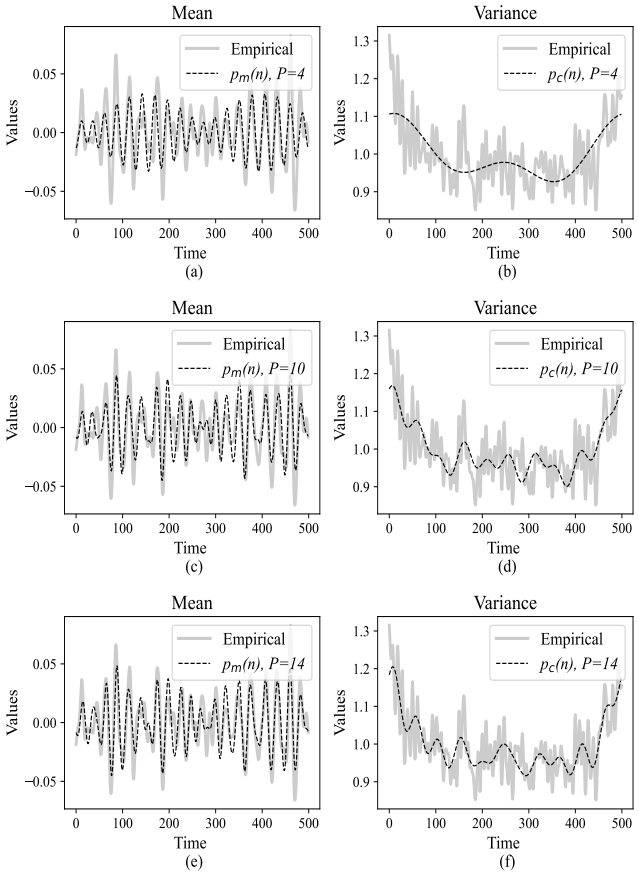

Figure 1. Example of sinusoidal interpolation $p_{\mathrm{m}}(n)$ and $p_{\mathrm{c}}(n)$ of the empirical mean and variance for the Ford A dataset. Interpolation orders considered are (a) and (b) $P = 4$, (c) and (d) $P = 10$, and (e) and (f) $P = 14$.

Figure 1 illustrate the application of the sinusoidal interpolation to the empirical mean and variance curves compute for the time series belonging to the Ford A dataset (see Section 4.1 for details on this dataset). Three different orders have been considered for $p_{\mathrm{m}}(n)$ and $p_{\mathrm{c}}(n)$, namely $P = 4$ (Figure 1 (a)), $P = 10$ (Figure 1 (b)), and $P = 14$ (Figure 1 (c)). Notice that the sinousidal regression can fit reasonably well the empirical statistic curves, especially for higher orders.

The proposed data augmentation procedure is given in Algorithm 1, which can be repeated for all class the user wants to augment. In the next section, we discuss the experiments to evaluate the data augmentation in anomaly detection settings.

Input: Collection $\mathcal{C}_g = \{\mathbf{Y}_j^{(g)}\}_{j=1,\ldots,J}$ of time series of a given class g to augment (see Eq. (19))

Output: i) Collection $\mathcal{C}_{g,\text{aug}}$ of augmented time series for class g. ii) Expressions for $\mathbf{x}_\text{m}(n+1)$ and $\mathbf{x}_\text{c}(n+1)$ (see Eqs. (13) and (14))

Step 1: Initialize TVAR_c and TVAR_m parameters $\{\gamma_\text{m}, \lambda_\text{m}, r_{1_\text{m}}, r_{2_\text{m}}\}$ and $\{\gamma_\text{c}, \lambda_\text{c}, r_{1_\text{c}}, r_{2_\text{c}}\}$ (see R1 to R3), and perturbation noise correlation matrix $\mathbf{\Phi}$ (see A4).

Initialize time series length N and number of augmented samples to create L;
Initialize curve interpolation order P;

Step 2: For $\mathcal{C}_g$, compute empirical statistics $\overline{\mathbf{m}}^{(g)}(n)$ and $\overline{\mathbf{C}}^{(g)}(n)$ using Eqs. (21) and (22);

Store $\overline{\mathbf{m}}^{(g)}(n)$ and $\overline{\mathbf{C}}^{(g)}(n)$;

Step 3: Compute P^{th} interpolation of $\overline{\mathbf{m}}^{(g)}(n)$ and $\overline{\mathbf{C}}^{(g)}(n)$, obtain one pair of scalar functions $p_\text{m}(n)$ and $p_\text{c}(n)$ per element of the array (See Section 3.3);

Store pair(s) of $p_\text{m}(n)$ and $p_\text{c}(n)$;

Step 4: Create $\mathbf{x}_\text{m}(n+1)$ and $\mathbf{x}_\text{c}(n+1)$ functions by using $p_\text{m}(n)$, γ_m λ_m, r_{1_m}, r_{2_m} and $p_\text{c}(n)$, γ_c λ_c, r_{1_c}, r_{2_c} in (13) and (14);

Store expressions for $\mathbf{x}_\text{m}(n+1)$ and $\mathbf{x}_\text{c}(n+1)$;

Step 5: Nested loop

for $l = 1$ **to** L **do**
  for $n = 0$ **to** $N-1$ **do**
    Compute $\mathbf{x}_\text{m}(n+1)$ and $\mathbf{x}_\text{c}(n+1)$;
  end
  Append $\mathbf{x}_\text{m}(n+1)$ and $\mathbf{x}_\text{c}(n+1)$ to $\mathcal{C}_{g,\text{aug}}$;
end

Return: i) $\mathcal{C}_{g,\text{aug}}$, ii) $\mathbf{x}_\text{m}(n+1)$ and $\mathbf{x}_\text{c}(n+1)$;

Algorithm 1: Proposed data augmentation procedure.

4. EXPERIMENTAL STUDY

In this section, we evaluate the ability of the proposed data augmentation method to improve the classification performance of typical ML methods in anomaly detection settings. We have tested two ML model architectures, three public sensor time series datasets, and we have compared the proposed technique against a competing data augmentation approach. All the simulations have been carried out in Python. More details are given ahead about the experiment design choices.

4.1. Datasets

We have tested three public datasets of univariate time series that that can be considered for anomaly detection studies. These are the collection of bearing vibration signals released by Case Western Reserve University (CWRU dataset) (Case School of Engineering Bearing Data Center, 2023), the set of measurements from piezeoelectric (PZT) sensors from the 2019 Prognostics Health Management Data Challenge (PHMDC2019 dataset) (He et al., 2013; Peng et al., 2015), and the Ford engine noise data (Ford A dataset) (Chen et al., n.d.).

The PHMDC2019 dataset contains Lamb waves from fatigue experiments on aluminum lap joints. For more details on the experiment, please check (He et al., 2013; Peng et al., 2015). The selected Lamb wave signals have been measured for eight specimens (specimen T1 to T8). For each specimen, different loads have been applied to the testing material, and a pair of signals from two sensors have been recorded (each signal with 4000 samples). Then, the crack lengths have been measured for each specimen, applied load, and pair of signals. Here, we have considered that a given signal sample corresponds to a "damaged" sample (Class 1) if the observed crack length exceeds 4 mm. Otherwise, the sample is considered as "normal" (Class 0). We have selected specimens T3 to T8 to build the train data and the remaining specimens to build the test. By doing so, we could guarantee that both train and test partitions could have data from Class 1. The sizes of the train and test splits for this dataset are shown in Table 1. Note that PHMDC2019 is the smallest dataset considered in this experiment. Using such a dataset allows the performance evaluation of the proposed method in small-data settings.

The CWRU dataset contains vibration signals collected from a drive-end bearing of an electrical motor operating under different loads. The selected vibration signals have been sampled at 12 kHz for motor speed of 1797 RPM (Case School of Engineering Bearing Data Center, 2023). The chosen signals are from a normal bearing (Class 0), a bearing with a crack in the inner race (Class 1), and another one with a crack in the ball (Class 2). The cracks of the faulty bearings have 0.007 inches. We have created signal snippets of 0.1 second from the vibration data of the bearings. To simulate challenging real-world scenarios, training samples from the faulty classes have been under-sampled to make the train data imbalanced. The number of samples obtained for the different partitions (i.e., train and test) and classes are shown in Table 1.

The Ford A dataset is a collection of signals for anomaly detection. The data has been originally proposed as a part of the Ford Classification Challenge, which appeared in competition program of the 2008 edition of the IEEE World Congress on Computational Intelligence (IEEE WCCI, 2008), and currently is made available by the UCR Time Series Classification Archive (Chen et al., n.d.). The Ford A dataset consists of noise signals collected from sensors installed on automotive engines that are either normal (Class 0), or present failure symptoms (Class 1). The numbers of signal samples available for each class and data partition are shown in Table 1. The signals have 500 points and are index time series (i.e., no information was provided about the sampling frequency).

Table 1. For the train and test splits, number of samples per dataset and per class used in the experiment of Section 4.

	PHMDC2019		Ford A		CWRU		
Split	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	Class 2
Train	50	22	1846	1755	1016	50	51
Test	12	8	681	639	1016	510	505

4.2. Machine Learning Models

The augmentation methods have been evaluated in the context of anomaly detection via time series classification by two machine learning models, namely a convolutional neural network (CNN) and a random forest (RF). The CNN architecture has been originally proposed in (Z. Wang, Yan, & Oates, 2017) and made available in (Fawaz, 2020) as a Tensorflow/Keras tutorial on neural networks for time series classification. The model hyperparameters (number of filters, batch and kernel sizes, etc.) have been determined with the KerasTuner module (Keras, 2022), and the resulting architecture has been trained for 20 epochs. More details on the model structure and its hyperparameters can be found in (Fawaz, 2020).

The RF model is the standard random forest implementation offered by Python scikit-learn library (Buitinck et al., 2013). The chosen RF architecture has 50 estimators and max depth of 5. The RF model is fitted to a tabular feature representation extracted from the time series data with the Python package tsfresh (Christ, Braun, Neuffer, & Kempa-Liehr, 2018). This module allows for extracting typical features from time series via a systematic framework. Further information on the available features and calculations implemented by tsfresh can be obtained in (Christ, Braun, Neuffer, & Kempa-Liehr, 2023).

4.3. Data-Augmentation Procedures

In this study, we have compared the classification performance obtained by using the proposed augmentation method against an alternative approach available in the literature.

The competing data augmentation approach is given by the Python library TSAug (Arundo Analytics, 2023), which offers a suite of typical augmentation transformations for time series. For the simulations, we have chosen some of the default transformation effects selected by the TSAug authors in (Arundo Analytics, 2023) to exemplify the capabilities of the library. These are i) addition of white Gaussian noise (WGN) to the time series with varying values of standard deviation (here, set to vary from $\pm 10\%$ of the standard deviation estimated from the data), ii) random drop of $\%10$ of the data points followed by filling with zeros, iii) random drift of a sequence of data points up and down, and iv) reduce of the temporal resolution (sampling) of some sequence of data points. For iii) and iv), we have considered a sequence of up to five data points. These transformations simulate random fluctuations and artifacts that commonly take place when processing batches of sensor data. Given an input time series, the TSAug library randomly applies transformations i) to iv) to create a new synthetic time series with the random artifacts.

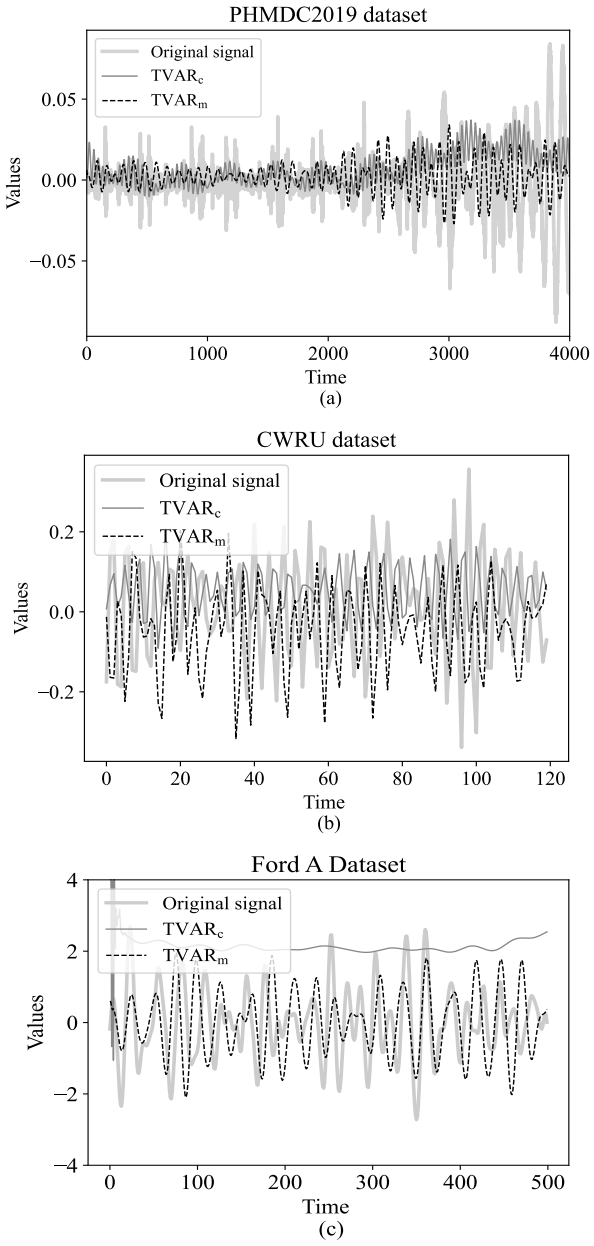

Figure 2. Examples of original and augmented time series for the datasets evaluated in this study. The augmented signals are generated by the fitted TVAR_m and TVAR_c sub-models with sinusoidal interpolation of order $P = 14$. (a) PHMDC2019, (b) CWRU, and (c) Ford A datasets.

The proposed augmentation (TvarAug) has been computed

Table 2. Parameter of the TVAR_m and TVAR_c employed to created the synthetic signal used in the experimental study.

	TVAR_m parameters				TVAR_c parameters			
Dataset	r_{1_m}	r_{2_m}	γ_m	λ_m	r_{1_c}	r_{2_c}	γ_c	λ_c
PHMDC2019	0.5	0.5	0.01	0.1	0.5	0.5	0.01	0.1
Ford A	0.5	0.5	0.01	1	0.5	0.5	0.01	1
CWRU	0.5	0.5	0.1	1	0.5	0.5	0.1	1

for $P = 4, 6, 8, 10, 12, 14$ (see Eqs. (28) and (29)). For each interpolated curve, we have created the TVAR_m and TVAR_c expressions (see Eqs. (13) and (14)) by considering the sets of parameter values[3] given in Table 2. The noise of the TVAR expressions has been set up as a WGN(0,1) process. Examples of the augmented time series obtained by using these parameters values for TVAR_m and TVAR_c sub-models are shown in Figure 2. Notice that, overall, the augmented data can capture relevant patterns of the dynamics of the original time series, while introducing the level of stochasticity necessary for generating new synthetic signal segments.

4.4. Test Results

The TVAR augmentation (TvarAug) and the TSAug approach have been employed to create new synthetic faulty samples belonging to the training partitions of the three considered datasets (see Section 4.1). These signal samples have been augmented considering the following augmentation fractions: 0.1, 0.25, 0.5 and 1. The CNN and RF models have been trained in the original and augmented train data partitions, and tested considering the available test data. The evaluation in the test data has been performed by computing the accuracy score of classifying the time series samples into the considered Faulty/Normal categories. For the TvarAug method, the average test accuracy has been computed for the different values of P considered. The obtained accuracy values are shown in Table 3, with the best results for each case highlighted in bold. The use of TvarAug has improved the classification when compared to the original data (NoAug) in all cases except the CWRU using RF where the original data already yields very good performance. As expected, the application of data augmentation has improved the overall detection performances of the ML models in all cases which are more challenging. The proposed method outperforms the competing approach for all considered datasets and augmentation fractions, which evidences its superior performance.

5. CONCLUSION

This paper presented a novel method for data-augmentation which can be effectively employed for fault diagnostics or failure prognostics applications as it can adequately be used for generation of artificial samples of multivariate non-stationary time-series data. The method leverages time-varying autoregressive models for this purpose and artificial samples are generated based on mean and covariance estimates obtained from the available real samples. The method was successfully tested based on failure diagnostics applications. Three publicly available real world datasets and two different machine learning methods were employed in the experiments and the proposed methodology provided improved results in almost all cases.

As the observed diagnostics results are promising, the proposed method could be further tested in more complex PHM tasks. Future work will include the application of the methodology for failure prognostics problems, as for this kind of application the scarcity of data is in general even more limiting compared to diagnostics.

[3]These have been chosen by searching for parameters that could synthesize data approximately in the same range of variation of the original datasets.

Table 3. Results as average values of test accuracy computed over different values of P, where "NoAug" stands for no data augmentation (original dataset), "TSAug" is the augmentation given by the TSAug library, and "TvarAug" is the proposed method. Results are shown for different datasets, augmentation fractions and machine learning models.

		RF			CNN		
Dataset	Frac. Aug	NoAug	TSAug	TvarAug	NoAug	TSAug	TvarAug
PHM-DC 2019	0.10	0.618	0.658	**0.775**	0.462	0.491	**0.600**
	0.25	0.618	0.650	**0.791**	0.462	0.483	**0.608**
	0.50	0.618	0.633	**0.783**	0.462	0.466	**0.600**
	1.00	0.618	0.658	**0.800**	0.462	0.525	**0.700**
Ford A	0.10	0.951	0.959	**0.963**	0.840	0.744	**0.917**
	0.25	0.951	0.961	**0.963**	0.840	0.754	**0.924**
	0.50	0.951	0.939	**0.960**	0.840	0.790	**0.876**
	1.00	0.951	0.948	**0.966**	0.840	0.791	**0.897**
CWRU	0.10	**0.997**	0.996	**0.997**	0.876	0.815	**0.951**
	0.25	0.997	0.995	**0.998**	0.876	0.814	**0.915**
	0.50	**0.997**	0.995	**0.997**	0.876	0.833	**0.887**
	1.00	**0.997**	0.993	0.996	0.876	0.913	**0.982**

REFERENCES

Abramovich, Y. I., Spencer, N. K., & Turley, M. D. E. (2007). Time-varying autoregressive (TVAR) models for multiple radar observations. *IEEE Trans. Signal Process.*, *55*(4), 1298-1311.

Arundo Analytics. (2023). *TSAug, A Python module for time series augmentation.* Retrieved from `https://tsaug.readthedocs.io/en/stable/` (Accessed: May 10, 2023)

Bates, D., & Watts, D. (2007). *Nonlinear regression analysis and its applications* (2nd ed.). Hoboken, NJ: Wiley.

Biggio, L., & Kastanis, I. (2020). Prognostics and health management of industrial assets: Current progress and road ahead. *Front. Artif. Intell.*, *3*, 1-24.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th ed.). Hoboken, NJ: Wiley.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD workshop: Languages for data mining and machine learning* (pp. 108–122). Prague, Czech Republic.

Case School of Engineering Bearing Data Center. (2023). *Case Western Reserve University (CWRU) Motor Bearing Dataset.* Retrieved from `https://engineering.case.edu/bearingdatacenter` (Accessed: May 10, 2023)

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., & Batista, G. (n.d.). *The UCR Time Series Classification Archive.* Retrieved from `www.cs.ucr.edu/~eamonn/time_series_data/` (Accessed: May 10, 2023)

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a Python package). *Neurocomputing*, *307*, 72-77.

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2023). *tsfresh - A Python package.* Retrieved from `https://tsfresh.readthedocs.io/` (Accessed: May 10, 2023)

de Boor, C. (2001). *A practical guide to splines* (revised ed.). New York, NY: Springer.

Deistler, M., & Scherrer, W. (2022). *Time series models* (1st ed.). Vienna, Austria: Springer.

de Oliveira, F. A. C., Niemi, A., García-Ortiz, A., & Torres, F. S. (2023). Partial camera obstruction detection using single value image metrics and data augmentation. In *International Conference on System Reliability and Safety (ICSRS 2023)* (p. 292-299). Venice, Italy.

de Souza, D. B., Chanussot, J., & Favre, A.-C. (2014). On selecting relevant intrinsic mode functions in empirical mode decomposition: An energy-based approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)* (p. 325-329). Florence, Italy.

de Souza, D. B., Chanussot, J., Favre, A.-C., & Borgnat, P. (2012). A modified time-frequency method for testing wide-sense stationarity. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)* (p. 3409-3412). Kyoto, Japan.

de Souza, D. B., Chanussot, J., Favre, A.-C., & Borgnat, P. (2014). A new nonparametric method for testing stationarity based on trend analysis in the time marginal distribution. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)* (p. 320-324). Florence, Italy.

de Souza, D. B., Chanussot, J., Favre, A.-C., & Borgnat, P. (2018). A nonparametric test for slowly-varying nonstationarities. *Signal Process.*, *143*, 241-252.

de Souza, D. B., Chanussot, J., Favre, A.-C., & Borgnat, P. (2019). An improved stationarity test based on surrogates. *IEEE Signal Process. Lett.*, *26*(10), 1431-1435.

de Souza, D. B., Kuhn, E. V., & Seara, R. (2019). A time-varying autoregressive model for characterizing nonstationary processes. *IEEE Signal Process. Lett.*, *26*(1), 134-138.

Fawaz, H. I. (2020). *Timeseries classification from scratch.* Retrieved from `https://keras.io/examples/timeseries/timeseries_classification_from_scratch/` (Accessed: May 10, 2023)

Garan, M., Tidriri, K., & Kovalenko, I. (2022). A data-centric machine learning methodology: Application on predictive maintenance of wind turbines. *Energies*, *15*(3), 1-21.

He, J., Guan, X., Peng, T., Liu, Y., Saxena, A., Celaya, J., & Goebel, K. (2013). A multi-feature integration method for fatigue crack detection and crack length estimation in riveted lap joints using lamb waves. *Smart Mater. Struct.*, *22*(10), 105007.

IEEE WCCI. (2008). *IEEE World Congress on Computational Intelligence.* Hong Kong. Retrieved from `https://ieeexplore.ieee.org/document/4762304`

Jiang, X., & Ge, Z. (2021). Data Augmentation Classifier for Imbalanced Fault Classification. *IEEE Trans. Autom. Sci.*, *18*(3), 1206-1217.

Kay, S. (2008). A new nonstationarity detector. *IEEE Trans. Signal Process.*, *56*(4), 1440-1451.

Keras. (2022). *KerasTuner.* Retrieved from `https://keras.io/keras_tuner/` (Accessed: May 10, 2023)

Kim, S., Choi, J.-H., & Kim, N. H. (2021). Challenges and opportunities of system-level prognostics. *Sensors*, *21*(22), 1-25.

Kim, S., Kim, N. H., & Choi, J.-H. (2020). Prediction of remaining useful life by data augmentation technique based on dynamic time warping. *Mech. Syst. Signal Process.*, *136*, 106486.

Kwak, M., & Lee, J. (2023). Diagnosis-based domain-adaptive design using designable data augmentation and bayesian transfer learning: Target design estimation and validation. *Appl. Soft Comput.*, *143*, 110459.

Leao, B. P., Fradkin, D., Lan, T., & Wang, J. (2021). Unleashing the power of industrial big data through scalable manual labeling. In *NeurIPS Data-Centric AI Workshop* (p. 1-5).

Li, H., Zhang, Z., & Zhang, C. (2023). Data augmentation via variational mode reconstruction and its application

in few-shot fault diagnosis of rolling bearings. *Measurement*, *217*, 113062.

Manolakis, D. G., Ingle, V. K., & Kogon, S. M. (2005). Random variables, vectors, and sequences. In *Statistical and adaptive signal processing: Spectral estimation, signal modeling, adaptive filtering and array processing* (p. 75-147). London, UK: Artech House.

Matei, I., Zhenirovskyy, M., de Kleer, J., & Feldman, A. (2018). Classification-based diagnosis using synthetic data from uncertain models. In *Annual Conference of the Prognostics and Health Management Society (PHM 2018)* (p. 1-8). Philadelphia, PA.

Montgomery, D., Peck, E., & Vining, G. G. (2006). *Introduction to linear regression analysis* (5th ed.). Hoboken, NJ: Wiley.

Niedzwiecki, M. (2000). *Identification of time-varying processes* (1st ed.). New York, NY: Wiley.

Pachori, R. B., & Sircar, P. (2008). EEG signal analysis using FB expansion and second-order linear TVAR process. *Signal Process.*, *88*(2), 415-420.

Peng, T., He, J., Xiang, Y., Liu, Y., Saxena, A., Celaya, J., & Goebel, K. (2015). Probabilistic fatigue damage prognosis of lap joint using bayesian updating. *J. Intell. Mater. Syst. Struct.*, *26*(8), 965-979.

Shen, B., Yao, L., Jiang, X., Yang, Z., & Zeng, J. (2023). Time series data augmentation classifier for industrial process imbalanced fault diagnosis. In *IEEE Data Driven Control and Learning Systems Conference (DDCLS 2023)* (p. 1392-1397). Xiangtan, China.

Sodsri, C. (2003). *Time-varying autoregressive modelling for nonstationary acoustic signal and its frequency analysis* (Unpublished doctoral dissertation). Pennsylvania State University.

Taghiyarrenani, Z., & Berenji, A. (2022). Noise-robust representation for fault identification with limited data via data augmentation. In *European Conference of the Prognostics and Health Management Society (PHME 2022)* (p. 473-479). Turin, Italy.

Wang, D., Dong, Y., Wang, H., & Tang, G. (2023). Limited Fault Data Augmentation With Compressed Sensing for Bearing Fault Diagnosis. *IEEE Sens. J.*, *23*(13), 14499-14511.

Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *International Joint Conference on Neural Networks (IJCNN 2017)* (p. 1578-1585). Anchorage, AK.

Yang, A., Lu, C., Yu, W., Hu, J., Nakanishi, Y., & Wu, M. (2023). Data Augmentation Considering Distribution Discrepancy for Fault Diagnosis of Drilling Process With Limited Samples. *IEEE Trans. Ind. Electron.*, *70*(11), 11774-11783.