Conditional variational autoencoders for probabilistic wind turbine blade fatigue estimation using Supervisory, Control, and Data Acquisition data

Journal Article

Author(s):
Mylonas, Charilaos; Abdallah, Imad ID; Chatzi, Eleni ID

WILEY

RESEARCH ARTICLE

Conditional variational autoencoders for probabilistic wind turbine blade fatigue estimation using Supervisory, Control, and Data Acquisition data

Charilaos Mylonas[ID] | Imad Abdallah[ID] | Eleni Chatzi[ID]

Chair of Structural Mechanics and Monitoring, Federal Institute of Technology Zürich, Zürich, Switzerland

Correspondence
Charilaos Mylonas, Chair of Structural Mechanics and Monitoring, Federal Institute of Technology Zürich, Zürich, Switzerland.
Email: mylonas.charilaos@gmail.com

Abstract

Wind turbine fatigue estimation is based on time-consuming Monte Carlo simulations for various wind conditions, followed by cycle-counting procedures and the application of engineering damage models. The outputs of the fatigue simulations are large in volume and of high dimensionality, as they typically consist of estimates on finite-element computational meshes. The strain and stress tensor time series, which are the primary quantities of interest when considering the problem of fatigue estimation, are dictated by complex vibration characteristics due to the coupled effect of aerodynamics, structural dynamics, geometrically non-linear mechanics, and control. A Variational Auto-Encoder (VAE) is trained in order to model the probability distribution of the accumulated fatigue on the root cross-section of a simulated wind turbine blade. The VAE is conditioned on historical data that correspond to coarse wind-field measurement statistics, such as mean hub-height wind speed, standard deviation of hub-height wind speed and shear exponent. In the absence of direct measurements of structural loads, the proposed technique finds applications in making long-term probabilistic deterioration predictions from historical Supervisory, Control, and Data Acquisition (SCADA) data, while capturing the inherent aleatoric uncertainty due to the incomplete information on strain time series of the wind turbine structure, when only SCADA data statistics are available.

KEYWORDS

blade root fatigue, conditional variational autoencoder, CVAE, deep generative models, high dimensional simulation outputs, SCADA, uncertainty quantification, wind turbine blade

1 | INTRODUCTION

Prediction of remaining life of wind turbines is of great interest to turbine owners and operators. Wind turbines are designed for an operational lifetime of at least 20 years.[1] As a consequence, a significant number of units may be reaching their end of fatigue design-life. Therefore, it is important to exploit historical records of influencing factors, such as records related to wind conditions, for robust estimates of the accumulated structural fatigue damage. Such a calculation may be in turn used for decisions on possible extension of the operation of an existing wind turbine structure.

Medium/high-fidelity estimates on accumulated fatigue from historical data, can be computed by reproducing the loading conditions of the turbine in simulations.[2] Such simulations, take into account the joint effect of structural dynamics, aerodynamics, multi-body dynamics, and turbine control (aero-servo-elastic simulations). Subsequently, cross-section load time series are post-processed in order to yield realistic stress and strain time series at the level of the cross-section materials.

Wind turbines typically contain Supervisory, Control, and Data Acquisition (SCADA) systems, which record relatively coarse summary statistics (mean and standard deviation) of quantities of interest and occurring faults. In order to obtain reliable estimates of the distribution of fatigue loads

from these historical data, several time-domain simulations need to be performed, in a way that reproduces the historical summary statistics and takes into account the statistics of the approximated quantities of interest. In practice, for wind loading time series, this is achieved by assuming a turbulence model and creating a random field that reproduces such statistics accordingly.[3,4] Due to the non-negligible computational expense of each full aero-servo-elastic simulation, as well as the creation of turbulent wind fields, it is practically infeasible to simulate the number of wind load time series that is sufficient for estimating a probability of accumulated damage equivalent loads on the turbine. Therefore, we must rely on probabilistic approximations for the quantities of interest, informed by the historical records of SCADA summary statistics. It should be noted that although historical SCADA streams do not contain wind profile measurements, wind profile may be available through nacelle-mounted Lidar or nearby stations.

A rational approximation, proposed by design standards,[5,6] is to *bin* different wind conditions and establish damage estimates through simulations. Typical influencing factors of variation include 10-min mean hub-height wind, 10-min wind standard deviation and wind shear, which are defined as the height-dependent wind velocity variation. Such an approach is not scalable when considering additional sources of uncertainty (more than 3), due to the so-called *curse of dimensionality*. A more efficient method, for sampling high-dimensional spaces, is pure random sampling (Monte Carlo sampling) or quasi Monte Carlo where low-discrepancy sequences[7] are used to create the discrete integration points.

In this work, we propose the construction of probabilistic models for the accumulated blade cross-section fatigue, employing large-scale simulations. Measurements representative of the distribution of the actual on-site conditions are employed, by inspecting the histograms of the marginal distributions of the inputs. Traditionally, due to the high computational requirements of estimating the accumulated fatigue on all material points of all cross-sections, fatigue is estimated only on critical points of the cross-section, such as the points of maximum stress variations due to bending. In this work we perform no such approximation, in order to capture potential variations on the spatial distribution of fatigue estimates in the cross-section. The raw outputs of the simulations consist of dynamic cross-sectional moments and forces on several cross-sections along the span of the considered blade. The time series of the moments and forces of the cross-sections are used to determine the strain and stress time series for the material points of all cross-sections. The time-series of the stress tensors on a finite element mesh of the blade root are post-processed by rainflow cycle-counting and by applying a simple fatigue damage accumulation model. The blade root cross-section damage equivalent load is analyzed. The analysis and methodology presented in this work naturally extend to any cross-section of the blades or to other mechanical fatigue estimates of different components of the wind turbine, which can be learned jointly employing the proposed methodology.

We treat the problem of estimating the output of fatigue damage equivalent load, as a high-dimensional statistics problem. A general-purpose probabilistic modeling technique that can cope with quantities of interest characterized by high dimensionality, such as the short-term accumulated fatigue damage on the whole cross-section considered herein, is proposed. Moreover, the proposed method can cope with high-dimensional conditioning inputs, even though this is not apparent from the case-study included herein. Namely, a Variational Autoencoder (VAE) is trained on the results of the short-term fatigue damage-equivalent load accumulation simulation. Furthermore, the VAE is *conditioned* (*Conditional Variational Autoencoder* (**CVAE**)) on coarse-grained parameters, which are typically available in SCADA systems. A fast to evaluate surrogate model for the fatigue accumulation on the cross-section fatigue is constructed. Moreover, instead of making simplifying parametric assumptions on the modes of variation in damage accumulation, such as the decomposition of fatigue in edge-wise and flap-wise, such modes of variation are learned directly in an unsupervised manner directly from the simulation outputs.

This paper is organized as follows: The paper begins with a short review on prior work on machine-learning enabled analysis of wind-energy related applications in Section 2 together with a short note on applications of the emerging deep generative models, such as the CVAE used in this work. In Section 3 the simulation setup is detailed, together with a short note on composite laminate fatigue damage modeling approaches for long-term (high-cycle) fatigue. In Section 4, for completeness, elementary notions of deep neural networks leading up to the Stochastic Gradient Variational Bayes (SGVB) algorithm and the VAE are presented. Finally, in Section 5 results are presented. We release the code and dataset for our case study.* The contributions of this work are three-fold:

- To the best of our knowledge, the problem of estimating the probability distribution of fatigue damage equivalent loads under the aleatoric uncertainty induced by coarse SCADA readings is for the first time addressed in this work.
- A state-of-the-art probabilistic machine learning technique (CVAE) is integrated into the composite blade fatigue analysis pipeline, allowing for extracting insights from large-scale Monte Carlo simulations
- The same technique is proposed as a viable alternative for the integration of long-term SCADA readings into fatigue remaining life estimates for existing wind turbines.

The CVAE, proposed in this work, combines ideas from unsupervised learning in order to exploit correlations in data of high-dimensionality in a data-driven manner and supervised learning, by jointly learning a feed-forward conditioning network. Moreover, the presented technique provides a natural means for quantifying irreducible (aleatoric) uncertainty in predictions, due to its derivation as a latent variable model trained with the so-called *Stochastic Gradient Variational Bayes* (**SGVB**) technique. Finally, the proposed technique offers the advantage of scaling well to high-dimensional problems, as it implicitly exploits the correlations of the quantities of interest, and at the same time scaling well to large datasets due to the data-parallel training enabled by stochastic gradient descent.

---

*Code and data will be available in https://www.github.com/mylonasc/fatigue-cvae upon publication.

2 | PRIOR RELATED WORK ON MACHINE LEARNING AND PROBABILISTIC TECHNIQUES FOR DAMAGE MONITORING AND REMAINING USEFUL LIFE PREDICTION

2.1 | Application of machine learning techniques in wind energy

In Cosack[8] simulation and measurement data were used for damage equivalent load estimation. Deterministic Artificial Neural Networks (ANNs) and regression were employed to that end. In Perez et al.[9] 5880 simulations with random shear, mean wind speed, turbulence intensity, air density, and up-flow were performed. The resulting 10-min damage equivalent loads (10-min DELs) were accumulated for each simulation for flap-wise bending of the blade, shaft torque, and tower-base bending moment. Deterministic feed-forward neural networks and linear regression were the tools exploited for this purpose. As inputs to the predictive algorithms, features were manually extracted from available SCADA summary statistics data, while considering dimensionality reduction techniques and feature selection procedures for reducing the dimensionality of the inputs. Finally, ANNs used in tandem with parametric models of wakes were proposed in Dimitrov and Natarajan[10] for site-dependent life estimation of wind turbines and in Müller et al.[11] single layer ANNs were used for damage equivalent load estimation.

In Murcia et al.,[12] non-linear regression with orthogonal basis functions was applied as a predictive model (*Polynomial Chaos Expansions* - PCE) for damage equivalent loads. In Mylonas et al.[13] PCE was employed together with linear dimensionality reduction, that is, principal components analysis (PCA) and non-negative matrix factorization, for the purpose of treating the high dimensionality of cross-section finite element analysis results. That work can be seen as an application of the "Dimensionality Reduction Surrogate Modeling" (DRSM) framework.[14] Similarly, Dimitrov et al.,[15] and Slot et al.[16] proposed regression with orthogonal polynomials (Legendre PCE) and Gaussian process regression[17,18] for the approximation of the blade root moments given SCADA information from the turbine. Toft et al.,[19] focusing on interpretability, used second order 2D polynomial fitting (response surface methodology) and bootstrapping to estimate uncertainty on a limited number of simulated cases of DEL computations.

In Dervilis et al.[20] the authors applied Autoencoders (also commonly referred to as "Auto-associators") with radial basis functions and PCA for demonstrating the utility of pattern recognition in damage detection on a wind turbine blade tested in the lab. In Garcke et al.,[21] the authors demonstrated how classical non-linear dimensionality reduction methods, such as Dynamic Time Warping[22] and diffusion maps[23] can be used to extract insights from simulated time-series of wind turbine responses. In Tang et al.,[24] the authors used a particular dimensionality reduction technique, namely, Orthogonal Neighborhood Preserving Embedding[25] for training a support vector machine classifier. In Khan et al.[26] deep Belief Networks[27] were used feature generation in wind turbine power forecasting applications.

In contrast to the aforementioned previous work on neural network-based and regression based wind turbine DEL estimation, the novelty of the present work lies in the use of state-of-the-art stochastic, rather than deterministic deep neural networks. The proposed conditional VAE model, yields *samples* from a learned multi-dimensional probability distribution of damage-equivalent loads, rather than point estimates such as statistical moments (expectations or quantiles). Such samples, can be used in order to obtain long-term deterioration estimates on wind turbine blades.

2.1.1 | Application of deep generative models in science and engineering

The variational autoencoder belongs to the class of deep *generative models*. Generative models approximate the joint distribution $p(\mathbf{x}, \mathbf{z})$ of observed variables, denoted by $\mathbf{x}$ and un-observed ones, denoted by $\mathbf{z}$. The dependencies (or conditional dependencies) may be modeled explicitly as in Bayesian networks or identified automatically from variations in data. The utility of generative modeling in engineering lies in quantitative statistical reasoning about uncertain or inherently stochastic quantities of interest. Deep generative models in particular, are models where at least some dependencies are modeled by deep neural networks. There are several advantages to using deep neural networks, but most prominently they serve as highly flexible non-linear function approximators. One of the first successful deep latent variable models related to neural networks, was the Restricted Boltzmann Machine (RBM)[28] that contained discrete (Bernoulli) latent variables. Deep Belief Nets (DBNs) consist of layers of autoencoders or RBMs and they have found applications in a variety of machine learning tasks including speech-related tasks[29] and computer vision.[30] Modern deep generative models, in addition to VAEs, include Generative Adversarial Networks (GANs),[31] and flow-based generative models.[32,33] Certain autoregressive models[34] can be considered as special cases of flow-based models. VAEs, GANs and flow-based generative models, and combinations thereof, seem to have largely replaced deep belief networks based on RBMs due to a combination of more modeling flexibility and training efficiency.[35] Typical application domains of modern deep latent variable models are computer vision and speech[31,36-38] computational genomics and drug design.[39-41] In engineering applications, there are few works that take advantage of such models, and even fewer that explicitly take advantage of their probabilistic underpinnings. Zaidan et al.[42] seems to be one of the early attempts using variational Bayes in engineering with an application in a hierarchical Bayesian regression model for gas turbine engines. Wang et al.[43] applied conditional variational autoencoders as feature extractors for down-stream classification tasks, in the context of planetary gear fault classification. With similar motivations as ours, Zhixin et al.[44] applied VAEs and GANs for data-driven electric vehicle charging profile generation. Chen et al.[45] applied GANs for sampling modeling renewable energy scenarios. Finally, in Zhu et al.[46] *conditional glow*,[37] which is a particular type of conditional generative normalizing flow model, was successfully employed for computational fluid dynamics problems.

Yoon et al.[47] applied variational autoencoders for feature learning, with an application in semi-supervised remaining useful life prediction. The VAE is used as a means of learning features from the monitoring data. In Mylonas et al.[48] we have applied conditional variational autoencoders

n the problem of modeling the SCADA summary statistics data, on a simulated wind farm, with the purpose of establishing VAEs as a tool for condition monitoring.

3 | CASE STUDY DESCRIPTION: WIND TURBINE SIMULATION

3.1 | Simulation inputs and stochastic wind-field generation

In this work, the input dataset is constructed by post-processing a large number of LiDAR (light detection and ranging) measurements of summary statistics retrieved from a LiDAR corresponding to an offshore wind farm.[†]

The choice of using samples from real measurements allows for unbiased estimation of the relative effects of the different possible input quantities to the quantity of interest. This circumvents possible issues associated with the relative importance of the influencing factors in the models proposed in the literature and the design standards. Accurate statistical modeling of site-specific conditions is an important topic, which is not treated in this work.

Ten-minute wind speed statistics are available at 10 different heights that cover the height of the turbine rotor considered. This allows for producing realistic time-series of environmental conditions. The simulation inputs consist of samples of wind speed, turbulence intensity and wind-shear. The mean hub-height wind speed and turbulence intensity were linearly interpolated to the hub-height of the considered rotor. The shear effect is taken into account according to a standard power law.[3] The formula describing the mean horizontal wind distribution with respect to height reads

$$\bar{u}(z) = \bar{u}_{hub} \left(\frac{z}{z_{hub}} \right)^{\alpha} \tag{1}$$

where $\bar{u}(z)$ is the mean wind speed at height z, z_{hub} is the hub-height, α is the so-called shear *power law exponent* and u_{hub} is the mean wind speed at the hub-height. The parameter α was fitted by application of Equation (1) such that the linearly interpolated wind speed on the higher and lower part of the rotor is consistent with the LIDAR readings.

The inputs to the simulations utilized herein, consist of 2000 samples of wind speed, turbulence intensity, and wind-shear as shown in Figure 1. TurbSim using the Kaimal turbulence model is used for creating the spatiotemporal time-series used for the simulations.

3.2 | Blade material properties, structural model setup and aero-servo-elastic simulation

The focus of this study is on the fatigue damage equivalent load (DEL) estimation of the thick laminate mix found close to the root. The NREL 1.5MW wind turbine was used. The geometrically non-linear behavior of the blades is described in BeamDyn reference manual.[49] BeamDyn is a non-linear spectral finite element simulator integrated into the OpenFAST[50] framework. Fifth order spectral elements were chosen for the simulations since higher polynomial orders came on a much higher computational cost coming from time-step size constraints. The definition of the geometry and the composite layout for the blade corresponding to the 1.5-MW wind turbine was taken from.[51] The cross-section properties were derived by setting up a detailed 3D computational model in ANSYS-ACP and extracting section cuts to BECAS[52] which extracts effective stiffness properties for the cross-section. The OpenFAST[50] software was used for the aero-servo-elastic simulation of the wind turbine. The simulations and the subsequent post-processing of the raw outputs were performed in parallel on the ETH cluster "*Euler*".

3.3 | Fatigue damage equivalent load computation

Fatigue damage modeling of glass fiber reinforced composites, is an active research subject and no universally accepted strategy exists for remaining life prediction. Most of the work on fatigue of composites adopts techniques from the relatively well-established fatigue of metals and has focused on adapting static failure criteria for composites. Several adaptations of the metal fatigue models exist, in order to take into account various effects characteristic of composite fatigue. Typically cycle-counting procedures are applied. Cycle-counting remaining life-based models and linear accumulation rules are justifiably criticized due to their lack of consideration of load-cycle order—that is, loading with a number of low amplitude cycles followed by high amplitude cycles does not have the same damaging effect as the opposite and such effects cannot be taken into account with cycle-counting models.

Two tasks of direct interest in engineering are fatigue-resistant modeling and remaining fatigue life estimation. Structural design relies on empirical assumptions, or at best on case-specific phenomenological models, about how the different factors may influence the quantities of interest. Therefore, influencing factors for the quantities of interest are scaled according to design norms, often arbitrarily in order to provide conservative estimates on the fatigue remaining life of a structure. On the other hand, when considering the remaining-life prediction task, one has to consider the actual loading conditions that the structure has experienced and the actual material properties and geometry of the structure. In that setting, damage propagation models are not required to be accurate only for the worst loading case but should be valid on average for the encountered distribution of loadings. Although design norms offer guidelines for the conservative design of structures, there is limited research on workflows that allow for confident remaining life estimation, in particular for composite structures. Therefore, a conscious choice was made to

---

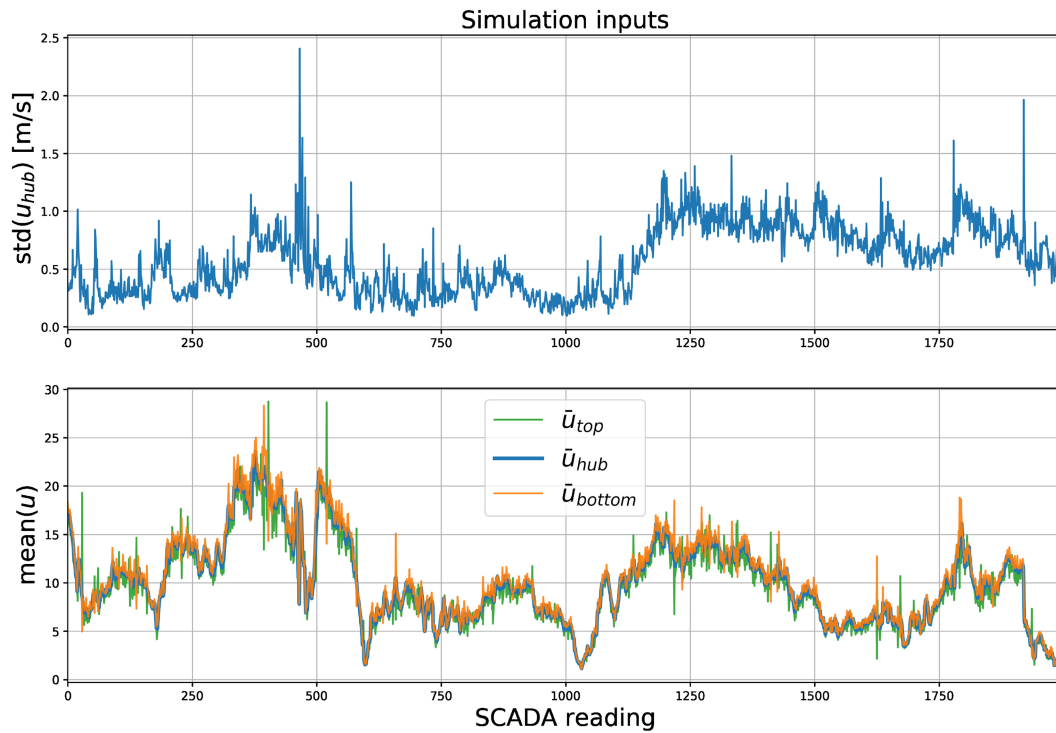[†]The data correspond to the Anholt wind farm.

FIGURE 1 Actual input variables for the simulations Top: standard deviation of the hub-height wind speed Bottom:Hub-height wind speed with the mean wind speed for the highest and lowest point of the rotor (vertical shear effect) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Properties of the composite materials of blade root

Material	E [GPa]	σ_c [MPa]	σ_t [MPa]
A260	27.1	−284	466
CDB 340	24.2	−478	667

use a simplified damage model, since the higher computational cost of more complicated models, is not warranted by their fidelity and generality, especially in the high-cycle fatigue regime, which is of interest in our application. The cross-section time-series resultants are read into BECAS for strain analysis at the material level. The resultants are returned from BeamDyn on a local coordinate system of the blade, which coincides with the system considered during the analysis with BECAS. For the estimation of cycles to failure N_{f,σ_a} under cyclic loading of amplitude σ_a with mean stress zero, the following simple power law is assumed[51]

$$N_{f,\sigma_a} = \left(\frac{\sigma_t - \sigma_c}{2\sigma_a} \right)^m \tag{2}$$

with σ_t the tensile strength of the material, σ_c the compressive strength of the material and m the so-called *Wöhler exponent*. For both composite materials comprising the root cross-section $m = 13$ is adopted. The tensile and compressive strengths of glass fiber reinforced composites are different, with the tensile strength being higher in absolute value. The tensile and compressive strengths are given in Table 1 for the root cross-section materials.

The mean-stress effect is taken into account through the so-called *Shifted Goodman* relation,[6,53] which amounts to Equation (3) for the cycles to failure in fatigue, given constant mean and constant amplitude loading.

$$N_{f,\sigma_a,\sigma_\mu} = \left(\frac{\sigma_t + |\sigma_c| - 2\sigma_\mu - \sigma_t + |\sigma_c|}{2\sigma_a} \right)^m \tag{3}$$

where σ_μ is the mean stress of the applied cycle. This relation is appealing due to its simplicity and due to the fact that it represents the different behavior of composites in compression and tension fatigue. Nevertheless, more complicated damage accumulation models are straight-forward to accommodate in the proposed framework as the methodology proposed is by no means constrained by the complexity of the model. A modeling assumption is made that the elastic properties do not exhibit large variations during the bulk of the lifetime of the blades. This is based on experimental observations at the coupon[54] and at the component level,[55,56] that in practice a maximum of 4% stiffness change is expected to occur from the 10% to the 90% of the total life of the component. Nevertheless, large stiffness changes indeed occur in composites under a high-cycle fatigue regime, and they are associated with the beginning of the cyclic loading and with reaching fatigue failure. Stiffness changes in composites depend on the main orientation of the reinforcing fibers with respect to the main orientation of the applied load. Large and progressive stiffness changes are observed when the loading is not aligned with the main strength and stiffness direction of the composite
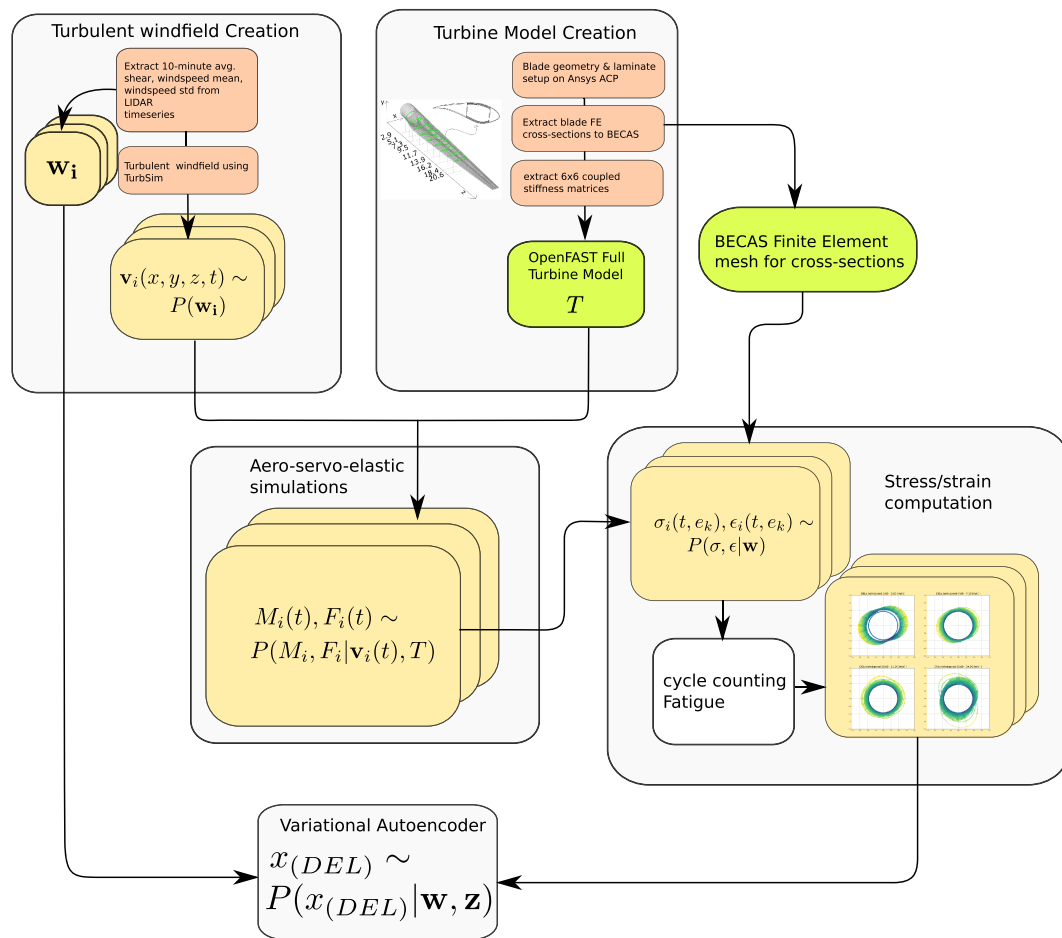
FIGURE 2 Flowchart of the simulation and analysis setup. Fatigue estimates at the material level are extracted corresponding to the wind field inputs. A CVAE is trained in order to be able to sample from the conditional probability distribution of fatigue loads, conditioned on coarse wind-field input parameters [Colour figure can be viewed at wileyonlinelibrary.com]

reinforcement or when large compressive stresses develop.[54] It is hereby assumed that the blade root does not have such obvious design flaws that are easy to prevent, simply by assigning the main portion of the reinforcing fibers to be aligned with the longitudinal dimension of the blade. This provides an effective load transfer mechanism for the cyclic tensile and compressive loads that are mainly due to blade bending. Moreover, simply by making the root cross-section adequately thick, the inevitable compressive stresses can be effectively transferred.

It is noted that, in contrast to fatigue loading associated with a high number of loading cycles (smaller mechanical loads), in low-cycle fatigue (higher mechanical loads) regardless of the underlying failure mechanism, such as matrix cracking coupled with fast progressing delaminations, composite stiffness changes are more pronounced. This failure mode is expected to occur relatively soon after deployment or detected during full-scale fatigue experiments in the lab for certification purposes. Therefore, it is less relevant to lifetime prediction for wind turbine blades, and therefore, this failure regime is not considered in the present study. A multitude of alternative formulations for mean-stress correction exist, which require a much larger volume of experimental data for their application.[57] The *Palgrem-Miner* linear damage accumulation assumption is therefore adopted. In order to perform short-term damage accumulation for each individual simulation run, for varying amplitude and mean stress, the rainflow counting procedure[58] is applied. A flowchart of the simulation and analysis process is shown in Figure 2.

4 | METHODOLOGICAL PART

4.1 | Background

4.1.1 | Dimensionality reduction

Oftentimes, the data of interest are characterized by strong correlations. Such correlations may allow for lower-dimensional representations that facilitate visualization and interpretation of such data. In addition to facilitating interpretation, lower dimensional representations are often useful for downstream tasks such as regression and classification.

Techniques that attempt to address such problems fall under the umbrella term *unsupervised learning*. Classical techniques for this task, such as PCA,[59,60] rely on learning linear projections as transformations of the data to the lower dimensional space. For many real-world datasets,
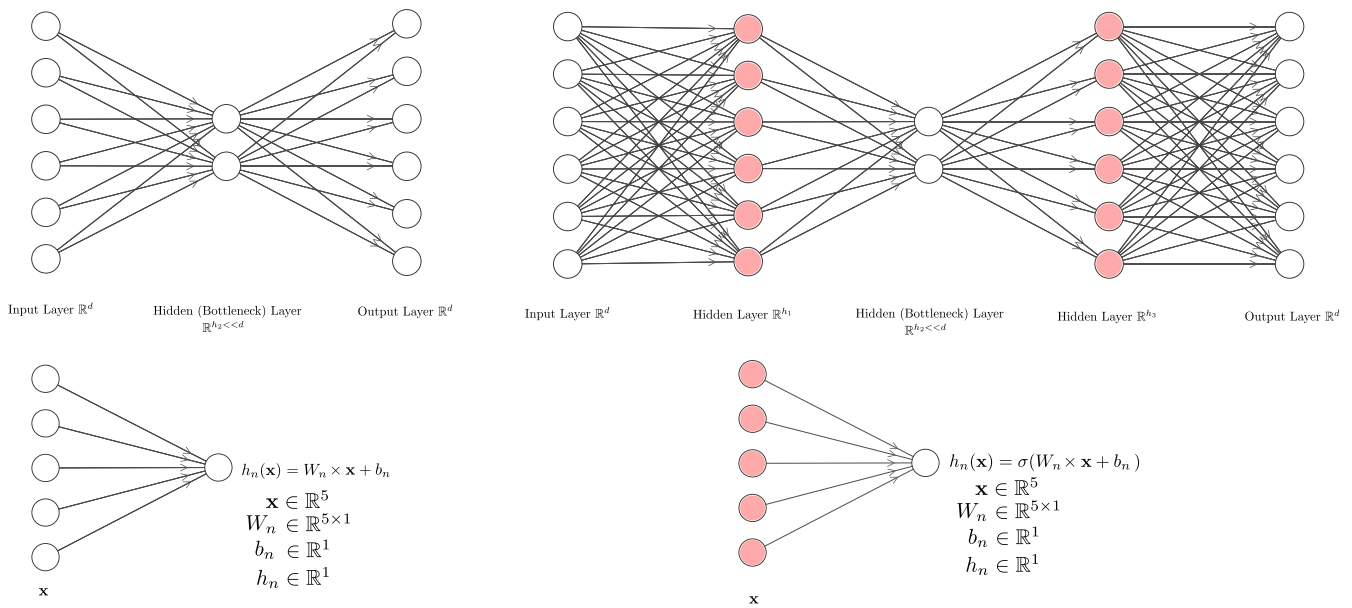
FIGURE 3 Top left: A linear autoencoder that is equivalent to any model based on sums of linear transformations such as PCA and factor analysis. Top right: Structure of an autoencoder with a 1-layer encoder and a one-layer decoder. Bottom left: the operation that represents a single *linear* neuron that takes as input $\mathbf{x} \in \mathbb{R}^5$ and returns a scalar output. Bottom right: A neuron with non-linearity [Colour figure can be viewed at wileyonlinelibrary.com]

linear transformations are not suitable for a reduced representation, in a way that facilitates analysis or interpretation. Therefore, non-linear unsupervised learning techniques have been invented such as kernel PCA,[61] locally linear embedding,[62] isomap,[63] and t-SNE.[64] In this work, a non-linear dimensionality reduction method, namely the neural network-based autoencoder[65] is of special interest.

An autoencoder consists of two parts. The *encoder* function $f(\cdot; \boldsymbol{\phi})$ parametrized by $\boldsymbol{\phi}$, *transforms* the high dimensional input data to a lower dimensional parameter space. The *decoder* $g(\cdot; \theta)$, parametrized by θ, is tasked with retrieving the high dimensional data back from this lower dimensional parameter space. The general form of an autoencoder is

$$\arg\max_{\phi, \theta} L(g(f(\mathbf{x}; \boldsymbol{\phi}); \theta), \mathbf{x})$$

$$f(x; \boldsymbol{\phi}) \; : \; \mathbb{R}^d \to \mathbb{R}^h$$

$$g(y; \theta) \; : \; \mathbb{R}^h \to \mathbb{R}^d \text{ with } d >> h$$

where L is a *reconstruction loss*, $\boldsymbol{\phi}$ are parameters of the encoder and θ are parameters of the decoder function $f(y; \theta) \; : \; \mathbb{R}^h \to \mathbb{R}^d$ with $d >> h$. PCA can be considered as a special case of an autoencoder, where the encoder is simply an orthogonal matrix containing the eigenvectors of the covariance matrix of the data, the decoder is the inverse, whereas the loss is the mean-squared error of the reconstruction. A comparison between he computational diagram of PCA/factor analysis and an autoencoder is shown in Figure 3.

An attractive property of autoencoders is that the mapping from the higher-dimensional space to the lower-dimensional space is learned jointly with the inverse mapping. This often allows for interpretation of the lower-dimensional representation through mapping-back to physical space and interpolating between the lower-dimensional representations. Alternative non-linear dimensionality reduction techniques, such as kernel PCA, need an additional training procedure for learning the inverse mapping from the lower dimensional representation to the original physical space, termed "the pre-image problem."[66] For the functions f and g, typically deep neural networks are used. The Variational Autoencoder (VAE), described in Section 4.2, can be conceived as an autoencoder where not only the lower-dimensional representation is learned, but at the same time, a probability distribution is learned over that lower-dimensional representation. For completeness, a brief introduction to deep neural networks is provided in the following.

4.1.2 | Deep neural networks

Deep neural networks (DNNs) are powerful function approximators, loosely inspired by simplified models of biological neurons. In recent years, DNNs have been shown to produce spectacular results on a large palette of particularly challenging tasks. A *k*-layer deep neural network, mathematically, is a composition of multi-dimensional, non-linear functions. Explicitly, one writes

$$f_{NN}(\mathbf{x}) = h_k \circ h_{k-1} \circ \ldots h_1(\mathbf{x})$$

$$h_i(\mathbf{y}) = \sigma_i(\mathbf{W}_i \mathbf{y} + b_i)$$

where $f{\circ}g(\cdot) = f(g(\cdot))$ is the composition operator and $\sigma_i(\cdot)$ is a non-linear function applied component-wise (i.e. $\sigma_i(\{x_1, x_2, \ldots, x_n\}) = \{\sigma_i(x_1), \sigma_i(x_2), \ldots \sigma_i(x_n)\}$). This function is referred to as the activation function may be different for different layers of the network. Typical choices for activation functions are the sigmoid function $Sigmoid(z) = \frac{1}{e^{-z}+1}$, the hyperbolic tangent $tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ and the rectified linear function $ReLU(z) = max(0, z)$. The matrix $\mathbf{W}_i$ and the vector b_i are refered to as the *weight* and *bias* of layer i. Most often, a differentiable loss is defined, such as the mean-squared loss for regression problems, and the partial derivatives of all the parameters of all the layers are computed with respect to the loss function with respect to the loss function. Deep neural networks are trained with the so-called *back-propagation* algorithm,[67] in order to compute gradients for parameters of all layers. Back-propagation is the numerical application of the chain rule of differentiation.

The main driving factors of the recent successes are better regularization,[68] training heuristics[69,70] and several architectural innovations.[71-73] Moreover, the increase in computational power and the availability of software frameworks focused on DNNs.[74-76] have greatly facilitated adoption from the industry and academia.

4.2 | Continuous latent variable modeling with variational autoencoders

In *latent variable modeling*, instead of attempting to compute the probability distribution $p(\mathbf{x})$ of our quantity of interest directly, we assume there is a process where observations of $\mathbf{x}$ are generated from. The observations of this process are dictated by a random variable $\mathbf{z}$. A joint distribution over $\mathbf{x}$ and $\mathbf{z}$ is assumed, parametrized with unknown hyper-parameters θ

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$$

and the probability of a particular observation $\mathbf{x}^{(i)}$ becomes

$$p_\theta(\mathbf{x}^{(i)}) = \int p_\theta(\mathbf{x}^{(i)}|\mathbf{z})p_\theta(\mathbf{z})dz$$

The variable $\mathbf{z}$ is the co-called latent variable.[‡] The maximum likelihood optimization goal, in order to find the set of parameters θ, assuming that all observations $\mathbf{x}^{(i)}$ are independent and identically distributed, is

$$\theta^* = \arg\max_\theta \prod_{i=1}^N p(\mathbf{x}^{(i)})$$

$$= \arg\max_\theta \prod_{i=1}^N \int p_\theta(\mathbf{x}^{(i)}|\mathbf{z})p_\theta(\mathbf{z})dz$$

The integral above is often computationally and analytically intractable to evaluate. Next, a parametric distribution q_ϕ over variables $\mathbf{z}$ is assumed. This distribution is differentiable with respect to the parameters ϕ and it is conditional on $\mathbf{x}$ which are the observed variables. This distribution is denoted by $q_\phi(\mathbf{z}|\mathbf{x})$ and it is referred to as the *approximate posterior* distribution of $\mathbf{z}$. The log marginal likelihood of $\mathbf{x}$, when we marginalize over $\mathbf{z}$, for any distribution over $\mathbf{z}$, reads

$$\log p(\mathbf{x}) = \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x})dz$$

$$= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z}|\mathbf{x})p_\theta(\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} \right) dz$$

$$= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z}, \mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} \right) dz$$

$$= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z}, \mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})} \right) dz$$

$$= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z}, \mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} \right) - q_\phi(\mathbf{z}|\mathbf{x}) \log \left(\frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})} \right) dz$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log(p_\theta(\mathbf{z}, \mathbf{x})) - \log(q_\phi(\mathbf{z}|\mathbf{x}))] + D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))$$

The quantity $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))$ is the *Kullback-Leibler* divergence between the approximate and the true posterior distribution. This quantity is non-negative. Moreover, $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) = 0$ implies $q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$. The KL divergence indeed is typically used as a measure[§] of how similar two distributions are. Due to the non-negativity of D_{KL}, the following inequality holds

---

[‡] In our concrete example, the latent variables can be thought of a parametrization of how the fatigue damage equivalent loads are distributed over the cross section. The premises of this paper, is that we do not fix a-priori such parametrizations, but we learn them directly from data.
[§] note that the KL divergence is not symmetric. i.e. $D_{KL}(q||p) \neq D_{KL}(p||q)$.

$$\log p(\mathbf{x}) \geq \log p(\mathbf{x}) - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))$$

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log(p_\theta(\mathbf{z}, \mathbf{x})) - \log(q_\phi(\mathbf{z}|\mathbf{x}))]$$

$$= \mathcal{L}(\phi, \theta; \mathbf{x}) \tag{4}$$

The quantity $\mathcal{L}(\phi, \theta; \mathbf{x}^{(i)})$ in Equation (4) is the so-called *evidence lower bound* or ELBO. Due to the non-negativity of D_{KL}, by optimizing ϕ, θ so as to maximize $\mathcal{L}(\phi, \theta, \mathbf{x})$, hyper-parameters ϕ and θ are simultaneously tuned in a way that they approximately maximize $\log p(\mathbf{x})$. Moreover, the same objective implicitly forces the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ to better approximate the true posterior. This effect can be observed by the need for the non-negative D_{KL} term to reduce, in order for the ELBO to become larger. The following is focused toward a scalable and fairly general technique for maximizing the ELBO that is trainable with stochastic gradient methods. For a comprehensive review focusing on scalable and general techniques, the reader is referred to Zhang et al.[77] In order to optimize ϕ and θ, which are typically the weights and biases of a neural network, we need to estimate the gradients of the ELBO. The main contribution of Kingma et al.[36] and Rezende et al.[78] was to propose a straight-forward low variance estimate for the gradient of the ELBO, through the so-called *re-parameterization trick*. For particular choices of approximate posteriors $q_\phi(\mathbf{z}|\mathbf{x})$ as a multivariate diagonal Gaussian, with mean and variance depending on functions $\mu_\phi(\cdot), \sigma_\phi(\cdot)$ differentiable w.r.t. ϕ, we can re-parameterize samples from the distribution as

$$\varepsilon^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{z}^{(i,l)} \sim q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{x}^{(i)}), \sigma_\phi(\mathbf{x}^{(i)}) \cdot \mathbf{I})$$

$$= \mu_\phi(\mathbf{x}^{(i)}) + \varepsilon^{(l)} \odot \sigma_\phi(\mathbf{x}^{(i)})$$

where $\odot$ denotes component-wise multiplication. Through this simple transformation expectations are evaluated using samples from a standard multivariate Gaussian, while at the same time the derivative $\nabla_\phi \mathcal{L}(\phi, \theta, \mathbf{x})$ is evaluated deterministically and separately for each sample. The ELBO can be re-arranged as

$$\mathcal{L}(\phi, \theta, \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log(p_\theta(\mathbf{x}|\mathbf{z})) + \log(p_\theta(\mathbf{z})) - \log(q_\phi(\mathbf{z}|\mathbf{x}))] \tag{5}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log(p_\theta(\mathbf{x}|\mathbf{z})) + \log\left(\frac{p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right)\right] \tag{6}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log(p_\theta(\mathbf{x}|\mathbf{z}))\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\left(\frac{p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right)\right] \tag{7}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log(p_\theta(\mathbf{x}|\mathbf{z}))\right] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) \tag{8}$$

In this form, with the special choice of a spherical unit Gaussian prior for $p_\theta(\mathbf{z})$ and a Gaussian re-parameterized posterior for $q_\phi(\mathbf{z}|\mathbf{x})$, the D_{KL} term can be computed analytically. It is noted that this is not the only option available for obtaining an analytically computable D_{KL} term, as noted also in the original papers. The first term plays the role of a reconstruction loss. The actual estimator for the ELBO reads

$$\mathcal{L}(\phi, \theta, \mathbf{x}) \simeq \frac{1}{L}\sum_{l=1}^{L} log(p_\theta(\mathbf{x}^{(i)}|z^{(i,l)})) - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))$$

The differentiable estimator used in the stochastic gradient updates, given a single input data-point $\mathbf{x}^{(i)}$ and L samples of the latent space per data-point $\mathbf{z}^{(i,l)}$, when using a Gaussian $q_\phi(\mathbf{z}|\mathbf{x})$ with diagonal covariance, and Gaussian $p_\theta(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as also presented in the original paper[36],

$$\mathcal{L}(\phi, \theta, \mathbf{x}^{(i)}) \simeq \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log(\sigma_\phi^{(j)}(\mathbf{x}^{(i)}))^2 - (\mu_\phi^{(j)}(\mathbf{x}^{(i)}))^2 - (\sigma_\phi^{(j)}(\mathbf{x}^{(i)}))^2\right) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{x}^{(i)}|z^{(i,l)}) \tag{9}$$

where J is the dimension of the Gaussian latent space and $\mathbf{z}^{(i,l)} = \mu_\phi(\mathbf{x}^{(i)}) + \sigma_\phi(\mathbf{x}^{(i)}) \odot \varepsilon^{(l)} \in \mathbb{R}^J$. In principle, even a single point from the latent space may be sampled ($L = 1$). Alternative methods available for training such models, provide estimates of the gradient with higher variance and converge slower or to worse local minima.[36,78]

4.2.1 | Conditional VAE

In the use-case presented herein, it is required to efficiently sample from a distribution *conditional* on SCADA-like data. In real settings, estimates of the mean hub-height wind speed and standard deviation are assumed available from nacelle anemometers. Wind shear profile can be made inferred through near-by LiDAR installations, nacelle-mounted LiDAR or post-processing of certain loads measurements as suggested in Namura.[79]

With a straight-forward variation on the VAE computational graph [80], this is easy to achieve. Namely, with $\mathbf{w}$ denoting the conditioning variables (in our specific example the SCADA data), the $q_\phi(\mathbf{z}|\mathbf{x})$ is replaced with $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{w})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ with $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{w})$. The computational graph of the conditional VAE (CVAE) is presented in Figure 4. The conditioning of the sub-networks on the SCADA variables is achieved by simply

FIGURE 4 The computational graph of a conditional VAE during training (top) and evaluation (bottom). The deterministic functions $f_\phi(\mathbf{w}, \mathbf{x})$ and $g_\theta(\mathbf{w}; \mathbf{z})$ are deep neural networks. Note the separation of the outputs of f_ϕ for μ_ϕ and σ_ϕ [Colour figure can be viewed at wileyonlinelibrary.com]
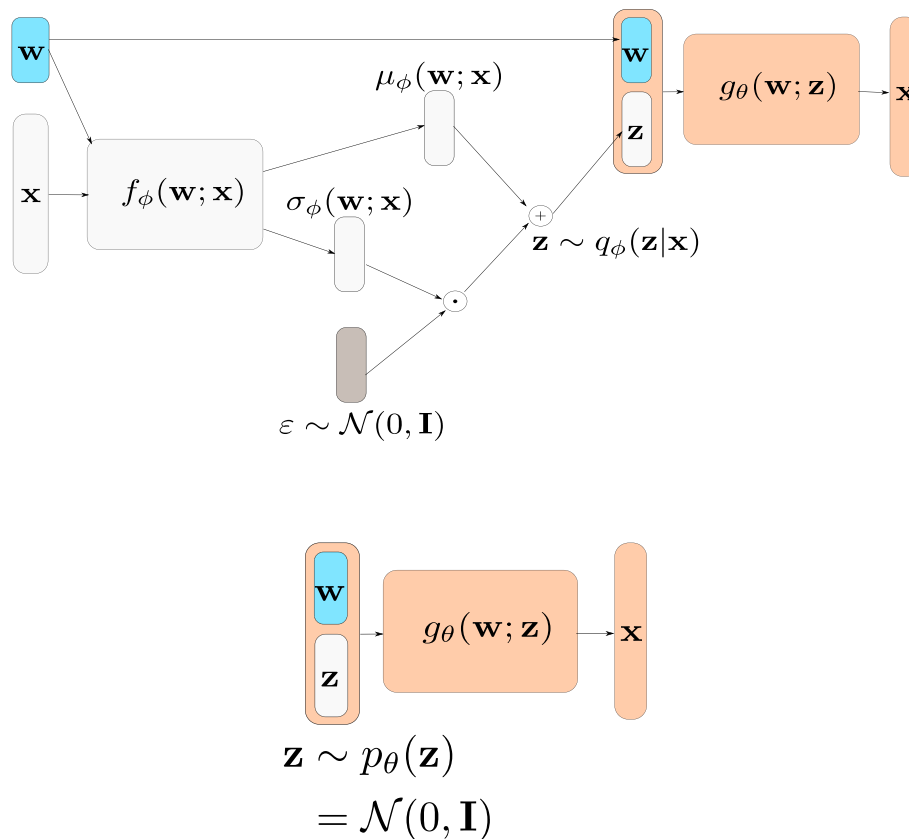
appending the SCADA variables to the inputs of the sub-networks. Instead of explicitly defining how the conditioning will happen, the function is learned internally from the decoder network.¶

During training the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ approaches the prior p_θ. In order to sample from the trained autoencoder, we simply discard the recognition network denoted with gray nodes in Figure 4 and replace sampling $\mathbf{z}$ from the approximate posterior with sampling from $\mathcal{N}(0, \mathbf{I})$. Clearly, the quality of this approximation depends strongly on how low is the $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))$ term in the loss, which quantifies how well the approximate posterior approximates the assumed prior.

5 | RESULTS

5.1 | Fatigue simulation results

The fatigue damage in the cross-section of the root is a stochastic variable. This stochasticity is attributed mainly to the characteristics of the created stochastic wind-field, and on a second level to the complex interaction of aero-dynamics, pitch control, and the geometrically non-linear response of the turbine blades. The variation in material parameters or the constructed geometry of the blade are other important sources of potential variation of remaining life estimates.[81] In this work, we do not treat such sources of uncertainty. For the purpose of uncertainty propagation for the accumulating damage equivalent loads, it is of interest to model the probability distribution of short term accumulating fatigue loads with respect to the variables that are registered in SCADA data. In Figure 5 the 10-min damage equivalent load results are presented for varying wind speed ranges. The orientation of the accumulating fatigue is varying from predominantly flap-wise in low wind speeds to predominantly edge-wise for high wind speeds. This is attributed to blade-pitch control. More specifically, in lower wind speeds blades are oriented for maximum power extraction and therefore higher lift, whereas for wind speeds corresponding to rated output power and above, the blades are pitched in a manner that yields lower lift, in order to maintain constant rotor speed. Due to the different pitch angles in different operational regimes, the stress distribution changes along the cross-section which is mainly attributed to blade bending under self-weight. Although there is a clear trend due to mean hub-height wind speed, perhaps unsurprisingly, there is also a clear trend for spatially irregular fatigue accumulation distributions over the cross-section for the results corresponding to higher wind speeds. This is interpreted, as an effect due to the larger 10-min variations of wind speed (turbulence), which in turn cause larger variations in rotor speed.

¶It is not strictly necessary to include the conditioning variables on the encoder network input.
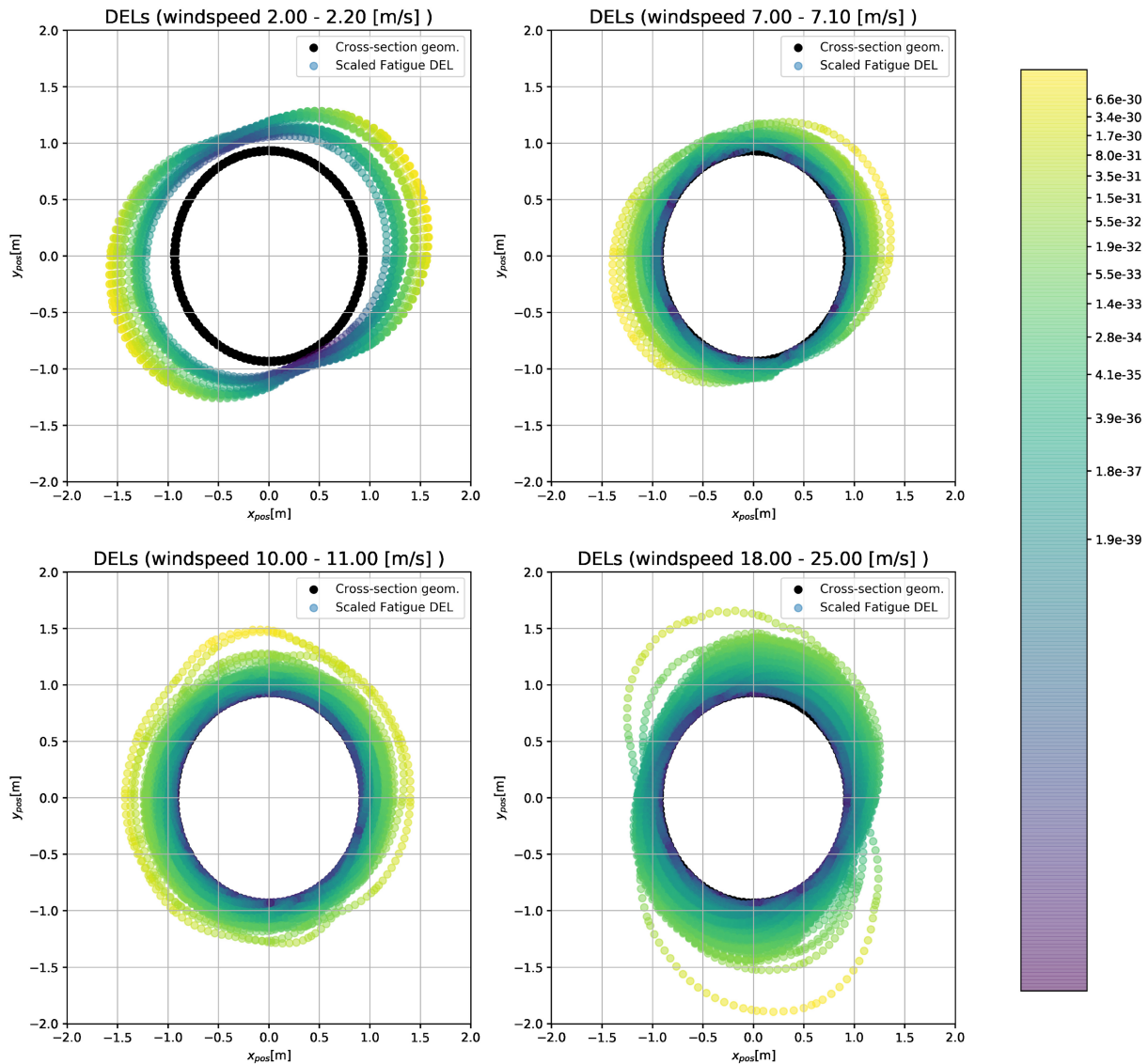
FIGURE 5 Variation of fatigue estimates on the root cross-section for different hub-height wind speed ranges. For effective visualization of the spatial variation of short-term fatigue damage equivalent loads (DELs), a polar coordinate system with the magnitude of the short-term damage equivalent load denoted by color and the distance coordinate, whereas the angle variable is computed by the coordinates of the cross-section geometry. The cross-section geometry is more easily discernible in the top left plot with black color. Note that color conveys the same information as the radial distance from the center of the plot which is the 10-min accumulated fatigue. There is a pronounced difference in the angular distribution of mean and variance of DELs among different ranges of wind speed [Colour figure can be viewed at wileyonlinelibrary.com]
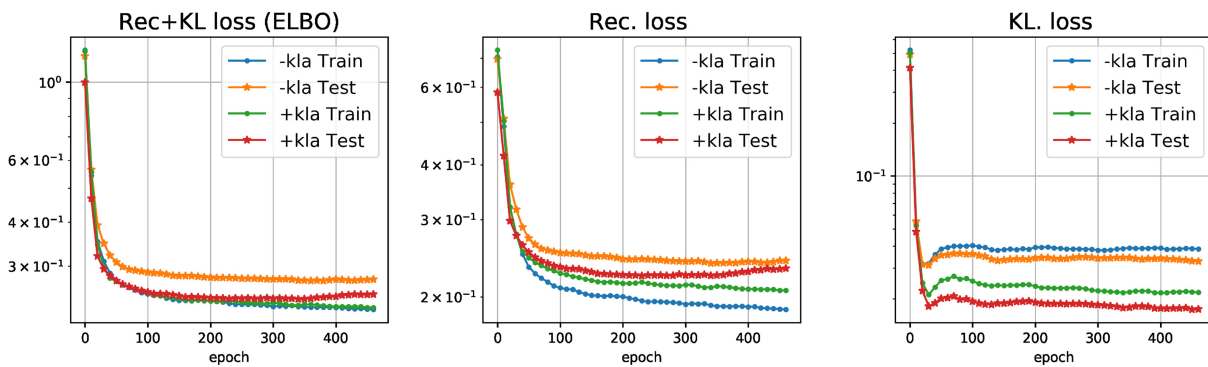

FIGURE 6 Train/Test 80%/20% split for model selection and investigation on training heuristics. The terms "test" and "validation" set are used interchangeably. Number of training steps for a particular architecture, learning rate and batch size are selected by detecting in which epoch the validation set error starts increasing. In the plots the effect of KL annealing is demonstrated. **+kla** with periodic KL-annealing, **-kla** without KL annealing. It is observed that the network trained with KL annealing achieves a much lower KL divergence and has better validation-set performance [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2　Reconstruction (mean squared error) and D_{KL} losses for different network architecture hyper-parameters

Enc/Dec width	Enc/Dec depth	latent size	activation	$D_{KL}(q_\phi(z|x)||p_\theta(z|x))$	$\frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{x}_i)^2$	
300	1	5	tanh	2.780e−03	2.603e−01	
			relu	2.939e−04	2.372e−01	*
			leaky_relu	5.296e−04	2.576e−01	
		15	tanh	2.685e−03	2.557e−01	
			relu	6.369e−04	2.483e−01	
			leaky_relu	7.883e−04	2.566e−01	
300	2	5	tanh	8.513e−05	2.522e−01	
			relu	7.232e−05	2.257e−01	*
			leaky_relu	1.492e−04	2.648e−01	
		15	tanh	2.828e−03	2.870e−01	
			relu	1.263e−04	2.448e−01	
			leaky_relu	2.190e−04	2.566e−01	
100	1	5	tanh	8.927e−04	2.645e−01	
			relu	1.076e−03	2.602e−01	
			leaky_relu	1.167e−03	2.636e−01	
		15	tanh	2.376e−03	2.727e−01	
			relu	8.520e−04	2.599e−01	
			leaky_relu	2.438e−04	2.540e−01	*
100	2	5	tanh	4.754e−04	2.609e−01	
			relu	2.618e−04	2.556e−01	
			leaky_relu	2.091e−04	2.590e−01	
		15	tanh	2.100e−03	2.592e−01	
			relu	3.068e−04	2.580e−01	
			leaky_relu	1.879e−04	2.498e−01	*
50	1	5	tanh	8.365e−04	2.647e−01	
			relu	4.117e−04	2.609e−01	
			leaky_relu	1.552e−04	2.597e−01	*
		15	tanh	1.658e−03	2.674e−01	
			relu	6.398e−04	2.629e−01	
			leaky_relu	2.498e−04	2.646e−01	
50	2	5	tanh	6.376e−04	2.626e−01	
			relu	2.797e−04	2.591e−01	*
			leaky_relu	1.753e−04	2.594e−01	
		15	tanh	1.631e−03	2.694e−01	
			relu	2.677e−04	2.635e−01	
			leaky_relu	1.797e−04	2.611e−01	

Note: In practice, in addition to selecting models with low reconstruction and KL-loss it was found important to manually inspect the generated samples of the trained models.

5.2 | Results on CVAE

5.3 | Model selection and optimization heuristics

For both g_θ and f_ϕ feed-forward neural-networks were used. Both *ReLU* and *tanh* activations showed similar performance. Regularization with drop-out was used between each dense layer. Dropout is a simple technique for preventing neural networks from overfitting by randomly setting to zero random subsets of layer outputs during training.[68] The Adam[69] optimizer was used. The learning rate was linearly incremented over 100 epochs from 0 to 10^{-3}. Moreover, as proposed in β-VAE[82] it is possible to consider the D_{KL} term of the ELBO as a varying regularization term. The modified ELBO objective becomes

$$\mathcal{L}_\beta(\phi, \theta, x) = \mathbb{E}_{q_\phi(z|x)}[\log(p_\theta(x|z))] - \beta D_{KL}(q_\phi(z|x)||p_\theta(z))$$

and the parameter β may be varied in order to control the rate/distortion properties of the VAE.[83] It was found that periodically varying the term from 0.5 to 1.5 among training steps is beneficial for keeping a balance between the reconstruction and training loss as demonstrated in Figure 6. The network is trained with a batch size of 100 and the period of the KL-annealing schedule was set to 500 training steps.

　Finally, 10 samples from the latent space were drawn for every data-point and for every forward pass in the network (the parameter corresponding to L in Equation (9)) which was found to marginally improve training. Networks with layer widths of 50, 100, and 300 were trained, and with depths ranging from one to three non-linear layers on the encoder and decoder. Wider deep networks (more than one layer for encoder and decoder) were found to achieve smaller validation set performance and over-fit in few epochs. This phenomenon may be alleviated with more regularization and may be less pronounced when more data are available. Relatively high dropout rate (30%) was found to partially alleviate

	Weights	Bias	Type	Activation	Dropout rate
Encoder	(276 × 300)	x1 ✓	Dense	ReLU	30%
	(300 × 300)	x1 ✓	Dense	ReLU	30%
	(300 × 30)	x1 ✓	Dense	Linear	None
Decoder	(18 × 300)	x1 ✓	Dense	ReLU	30%
	(300 × 300)	x1 ✓	Dense	ReLU	30%
	(300 × 273)	x1 ✓	Dense	Linear	None

TABLE 3 Layers of the CVAE network with $\mathbf{z} \in \mathbb{R}^{15}$, $\mathbf{w} \in \mathbb{R}^3$ and $\mathbf{x} \in \mathbb{R}^{273}$

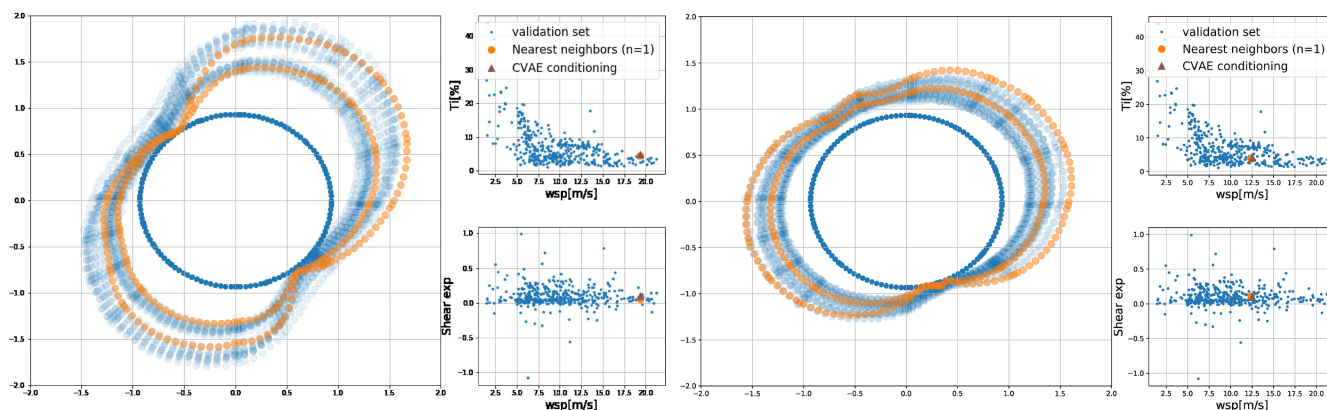Abbreviation: CVAE, Conditional Variational Autoencoder.

FIGURE 7 Samples from the Conditional Variational Autoencoder (CVAE) and single nearest neighbor according to the conditioning parameters from the test-set and the conditioning parameters plotted against each other. On the damage equivalent load (DEL) plot the orange points denote the nearest neighbor whereas the blue points denote 10 samples for the CVAE for a single conditioning parameter. Color information was discarded in order to effectively overlay the plots of the nearest-neighbor on the CVAE estimates [Colour figure can be viewed at wileyonlinelibrary.com]

this problem. Finally a network with 300-unit wide layers, and with two layers both in the encoder and decoder was selected. Smaller and deeper networks were found to produce comparable results. Numerical experiments were performed with 15-dimensional and five-dimensional latent variables and showed no significant performance differences. The losses of the different models are summarized in Table 2

A detailed summary of the layers for the selected network is given in Table 3. This architecture corresponds to the fifth row of Table 2. Note that the output of the encoder is simply split into two sets of output variables. One set of variables is $\mu_\phi(\mathbf{x})$ and $\sigma_\phi(\mathbf{x})$. Also note that due to the conditioning of both networks with the three SCADA-like variables chosen (shear, wind, turbulence), the input of the encoder is three variables larger than the quantity of interest (damage equivalent loads at the cross-section) and the input to the decoder is correspondingly three variables larger.

5.4 | Inspection of CVAE results on test-set

The model is expected to learn a distribution over possible fatigue damage equivalent loads. In order to investigate the effectiveness of the CVAE in capturing the variations in the data, unseen samples from the validation set are overlaid to samples from the CVAE. In order to demonstrate the effect of different SCADA parameters, the validation set samples are selected as the nearest neighbors according to the Euclidean distance from the conditioning SCADA variables of the CVAE, plotted in blue. For easier inspection, the conditioning variables are plotted alongside the DEL estimates.

Ten samples from the trained CVAE are drawn for each case. The shape of the fatigue accumulation over the cross-section is indeed captured as demonstrated in Figure 7. In Figure 8 (left) the effectiveness of the CVAE in capturing unseen examples for scarce regions of the parameter space is examined. The CVAE predicts a relatively uniform distribution along the cross-section of the blade root, and indeed the unseen example has properties similar to the ones predicted. Finally in Figure 8 (right), for a region of the SCADA parameters where enough DEL estimates exist in the test set, 10 nearest-neighbors are plotted over 10 samples from a CVAE. It is observed that the CVAE captures the spatial scatter associated with this region of the parameter space. It is therefore demonstrated that the CVAE captures effectively, and in an automated manner the conditional heteroscedasticity and the intricate spatial distribution of fatigue for a variety of operating conditions.

5.5 | Uncertainty propagation with the CVAE for cross-section fatigue

In this section, the trained CVAE is used in order to efficiently yield estimates of fatigue for the cross-section, employing only the SCADA data. The decoder network (inference network) of the VAE was conditioned during training with the inputs to the simulations. It is possible to perform sampling conditioned on the simulation inputs to yield short-term fatigue damage equivalent loads for the entire root cross-section, as
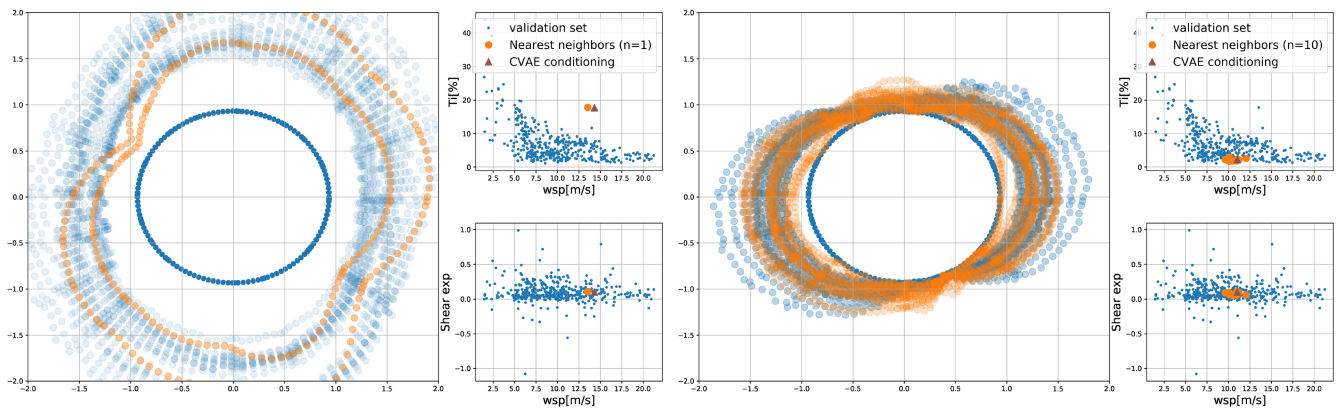
FIGURE 8 Left: Sample from the Conditional Variational Autoencoder (CVAE) for a data-point in a region of the parameter space where the training data are scarce. Right: Demonstration of the CVAE capturing the heteroscedasticity conditioned on the Supervisory, Control, and Data Acquisition (SCADA) variables [Colour figure can be viewed at wileyonlinelibrary.com]
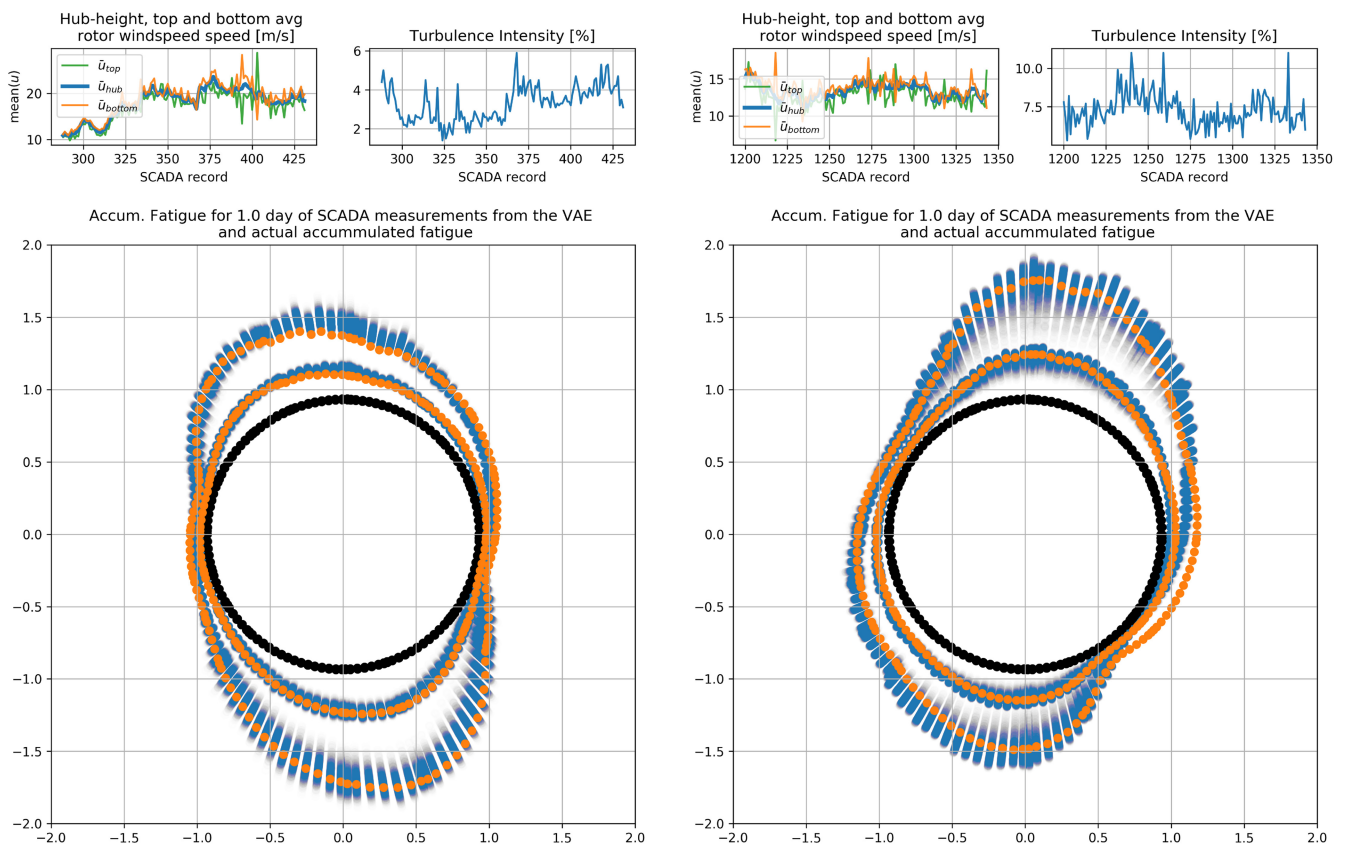

FIGURE 9 Accumulated 1-day fatigue estimates & uncertainty on estimation from the trained Variational Auto-Encoder (VAE). The simulated training data are depicted in orange. Conditionally sampling repeatedly from the VAE for the same Supervisory, Control, and Data Acquisition (SCADA) data sequence reveals the learned uncertainty on the accumulated fatigue DEL estimates from the SCADA data. The values are non-linearly scaled ($\bar{x} \rightarrow x^{0.01}$) in order to make spatial correlations more pronounced. The two distinct lines occur due to the difference in fatigue resistance of the A260 and CDB340 composites, that constitute the materials of the cross-section [Colour figure can be viewed at wileyonlinelibrary.com]

the CVAE learns internally the correlations of the fatigue estimates. The sampling is performed using the assumed prior distribution, which is a 15-dimensional spherical Gaussian, rather than the data-dependent approximate posterior.

In Figure 9, 1000 distinct *accumulated* fatigue estimates from the autoencoder for a set of 144 consecutive SCADA records, corresponding to one day, are shown. The computed fatigue estimate from the actual training data is overlaid (orange curve). Due to the differences in their stiffness and compressive and tensile strengths, the two materials of the cross-section accumulate fatigue at a different rate. This is the reason for the two distinct regions of fatigue estimates. Note that a single CVAE yields the estimates for all the points on the FE mesh without any explicit modeling for the spatial correlations or the different fatigue behavior of the two different composite materials. It is observed that the accumulated fatigue estimates from the CVAE are highly heteroscedastic and feature a spatial dependency.

The scatter in the CVAE estimates is due to the inherent aleatoric uncertainty of the problem of estimating fatigue on a cross-section using only 10-min SCADA readings. Recall that every estimate from the trained CVAE is stochastic, and is distributed in a manner that best approximates the stochastic simulation outputs as learned from the training data, and *not* a point estimate. Therefore, for every independent sample of the latent space **z** for the same SCADA reading we yield a different estimate of damage equivalent load. What is visually demonstrated in Figure 9, by the scatter in the CVAE samples, is how uncertain we may be about a daily fatigue estimate when only 10-min SCADA data are available. On the other extreme, when considering the entire load history for the day, one may yield a deterministic estimate, which is represented by the orange points overlaid on the CVAE fatigue estimates.

6 | CONCLUSIONS AND FUTURE WORK

In this work the task of quantifying the aleatory uncertainty of short-term damage equivalent loads, when only 10-min SCADA summary statistics are available was addressed. The short-term damage equivalent loads were accumulated and analyzed on the material level for the root cross-section of a blade corresponding to a wind turbine with 1.5-MW rated power. The damage accumulation patterns were found to be interpretable in terms of the mechanical loads expected in different loading conditions, and in particular, the varying blade-pitch with mean hub-height wind speed and the increased variability associated with larger turbulence.

Moreover, a novel, easy to implement, state-of-the-art technique for approximating in a principled manner stochastic quantities of interest was employed. It is argued that VAEs and generative models in general are a promising tool for diverse overarching tasks related to structural health monitoring and engineering under uncertainty. Latent variable models (LVMs), such as VAEs, offer the possibility of modeling in an automated manner the effects of unobserved variables, as the name directly implies. Such effects are typically treated as noise in regression settings, where models fit for the most probable or the expected value under a noise model (which prevents the direct comparison of LVMs with regression models). The regression approach discards important information for various engineering problems, such as the modeling of long-term deterioration. Latent variables can, for instance, represent effects that are not captured in the modeling of a physical process or the acquired data, as it is the case in the present work. In the offered case study, we demonstrated this idea on short-term fatigue damage equivalent loads for wind turbines. Other wind-energy related problems where CVAEs may find application are, for instance, wind farm level SCADA data modeling[48] or turbulence modeling[84,85] where finer turbulence scales are resolved in a data-driven manner in place of the mostly empirical and mathematically simplified standard classical approaches.

One advantage of deep generative models, stemming from the fact that they rely on neural networks for approximation, is that they may be applied in datasets that are large in volume and contain variables of high-dimensionality. The scalability of deep neural networks stems from the data-parallel nature of stochastic gradient descent which greatly facilitates distributed training and the high performance in high dimensional settings is empirically supported by the recent successes of deep neural networks.

During numerical experiments it was observed that training required several heuristics for achieving at the same time a low reconstruction error and a low $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))$, amounting to high ELBO. Current research in Variational Autoencoders, focuses in the incorporation of more flexible posteriors through the use of invertible transformations,[86-88] theoretical insights and proposals for more robust training procedures[83] and the more robust optimization losses.[89]

Peer Review

The peer review history for this article is available at https://publons.com/publon/10.1002/we.2621.

ORCID

Charilaos Mylonas [iD] https://orcid.org/0000-0003-2108-2247

Imad Abdallah [iD] https://orcid.org/0000-0001-8678-0965

Eleni Chatzi [iD] https://orcid.org/0000-0002-6870-240X

REFERENCES

1. Ziegler L, Gonzalez E, Rubert T, Smolka U, Melero JJ. Lifetime extension of onshore wind turbines: a review covering germany, spain, denmark, and the uk. *Renew Sustain Energy Rev*. 2018;82:1261-1271.
2. Muskulus M. Simplified rotor load models and fatigue damage estimates for offshore wind turbines. *Philos Trans Royal Soc A: Math Phys Eng Scie*. 2015;373(2035):20140347.

3. Kelley ND, Jonkman BJ. Overview of the turbsim stochastic inflow turbulence simulator, Golden, CO (United States), National Renewable Energy Lab.(NREL); 2005.

4. Mann J. Wind field simulation. *Probabilistic Eng Mech*. 1998;13(4):269-282.

5. Committee IE. Wind turbines IEC standard; 2005.

6. DNV GL. As: Dnvgl-st-0376–rotor blades for wind turbines. https://rules.dnvgl.com/docs/pdf/DNVGL/ST/2015-12/DNVGL-ST-0376.pdf; 2015.

7. Caflisch RE. Monte Carlo and quasi-Monte Carlo methods. *Acta Numer*. 1998;7:1-49.

8. Cosack N. Fatigue load monitoring with standard wind turbine signals; 2010.

9. Pérez-Campuzano D, Gómez de las Heras-Carbonell E, Gallego-Castillo C, Cuerva A. Modelling damage equivalent loads in wind turbines from general operational signals: exploration of relevant input selection methods using aeroelastic simulations. *Wind Energy*. 2018;21(6):441-459.

10. Dimitrov N, Natarajan A. *From SCADA to Lifetime Assessment and Performance Optimization: How to Use Models and Machine Learning to Extract Useful Insights From Limited Data*. Bilbao, Spain: IOP Publishing; 2019: 12032.

11. Müller K, Dazer M, Cheng PW. Damage assessment of floating offshore wind turbines using response surface modeling. *Energy Procedia*. 2017;137:119-133.

12. Murcia JP, Réthoré P-E, Dimitrov N, Natarajan A, Sørensen JD, Graf P, Kim T. Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates. *Renew Energy*. 2018;119:910-922.

13. Mylonas C, Abdallah I, Chatzi E. Surrogate modelling for fatigue damage of wind-turbine blades using polynomial chaos expansions and non-negative matrix factorization. 39th IABSE Symposium in Vancouver 2017: Engineering the Future. International Association for Bridge and Structural Engineering (IABSE); 2017:801-808.

14. Lataniotis C, Marelli S, Sudret B. Extending classical surrogate modelling to ultrahigh dimensional problems through supervised dimensionality reduction: a data-driven approach. arXiv preprint arXiv:1812.06309; 2018.

15. Dimitrov NK, Kelly MC, Vignaroli A, Berg J. From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases. *Wind Energy Science*. 2018;3(2):767-790.

16. Slot RM, Sørensen JD, Sudret B, Svenningsen L, Thøgersen ML. Surrogate model uncertainty in wind turbine reliability assessment. *Renew Energy*. 2020;151:1150-1162.

17. Rasmussen CE, Nickisch H. Gaussian processes for machine learning (gpml) toolbox. *J Mach Learn Res*. 2010;11(Nov):3011-3015.

18. Lataniotis C, Marelli S, Sudret B. Uqlab user manual—Kriging (Gaussian process modelling). Report UQLab-V0; 2017.

19. Toft HS, Svenningsen L, Moser W, Sørensen JD, Thøgersen ML. Assessment of wind turbine structural integrity using response surface methodology. *Eng Struct*. 2016;106:471-483.

20. Dervilis N, Choi M, Taylor SG, Barthorpe RJ, Park G, Farrar CR, Worden K. On damage diagnosis for a wind turbine blade using pattern recognition. *J Sound Vib*. 2014;333(6):1833-1850.

21. Garcke J, Iza-Teran R, Marks M, Pathare M, Schollbach D, Stettner M. Dimensionality reduction for the analysis of time series data from wind turbines. In: Griebel M, Schüller A, Schweitzer MA, eds. *Scientific Computing and Algorithms in Industrial Simulations: Projects and Products of Fraunhofer SCAI*. Cham: Springer; 2017:317-339. https://doi.org/10.1007/978-3-319-62458-7_16

22. Itakura F. Minimum prediction residual principle applied to speech recognition. *IEEE Trans Acoust, Speech, and Signal Process*. 1975;23(1):67-72.

23. Coifman RR, Lafon S, Lee AB, Maggioni M, Nadler B, Warner F, Zucker SW. Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. *Proc Natl Acad Sci*. 2005;102(21):7426-7431.

24. Tang B, Song T, Li F, Deng L. Fault diagnosis for a wind turbine transmission system based on manifold learning and Shannon wavelet support vector machine. *Renew Energy*. 2014;62:1-9.

25. Liu X, Yin J, Feng Z, Dong J, Wang L. Orthogonal neighborhood preserving embedding for face recognition. *2007 IEEE Int Confer Image Process*; 2007:I-133.

26. Khan A, Zameer A, Jamal T, Raza A. Deep belief networks based feature generation and regression for predicting wind power. arXiv preprint arXiv:1807.11682; 2018.

27. Hinton GE. Deep belief networks. *Scholarpedia*. 2009;4(5):5947.

28. Hinton GE. Boltzmann machine. *Scholarpedia*. 2007;2(5):1668. revision #91076.

29. Mohamed A, Dahl GE, Hinton G. Acoustic modeling using deep belief networks. *IEEE Trans Audio, Speech, and language Process*. 2011;20(1):14-22.

30. Lee H, Grosse R, Ranganath R, Ng AY. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal, Quebec, Canada: Association for Computing Machinery; 2009:609-616.

31. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*. Curran Associates, Inc; 2014:27;2672-2680. https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494ac97b1afccf3-Paper.pdf

32. Dinh L, Sohl-Dickstein J, Bengio S. Density estimation using real NVP. arXiv preprint arXiv:1605.08803; 2016.

33. Dinh L, Krueger D, Bengio Y. NICE: non-linear independent components estimation. arXiv preprint arXiv:1410.8516; 2014.

34. Huang C-W, Krueger D, Lacoste A, Courville A. Neural autoregressive flows. arXiv preprint arXiv:1804.00779; 2018.

35. Goodfellow I, Bengio Y, Courville A. *Deep Learning*: MIT press; 2016;693-695.

36. Kingma DP, Welling M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114; 2013.

37. Kingma DP, Dhariwal P. Glow: generative flow with invertible 1x1 convolutions. *Advances in Neural Information Processing Systems*. 57 Morehouse Lane Red Hook NY, United States: Curran Associates Inc.; 2018:10215-10224.

38. Prenger R, Valle R, Catanzaro B. Waveglow: a flow-based generative network for speech synthesis. *Icassp 2019-2019 IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP)*. IEEE; 2019:3617-3621. https://ieeexplore.ieee.org/xpl/conhome/8671773/proceeding

39. Kadurin A, Aliper A, Kazennov A, Mamoshina P, Vanhaelen Q, Khrabrov K, Zhavoronkov A. The cornucopia of meaningful leads: applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*. 2017;8(7):10883.

40. Sanchez-Lengeling B, Aspuru-Guzik A. Inverse molecular design using machine learning: generative models for matter engineering. *Science*. 2018;361(6400):360-365.

41. Way GP, Greene CS. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *Pac Symp Biocomput*. 2018;23:80–91.

42. Zaidan MA, Mills AR, Harrison RF, Fleming PJ. Gas turbine engine prognostics using Bayesian hierarchical models: a variational approach. *Mech Syst Signal Process*. 2016;70:120-140.

43. Wang Y, Jin Q, Sun G, Sun C. Planetary gearbox fault feature learning using conditional variational neural networks under noise environment. *Knowledge-Based Syst*. 2019;163:438-449.

44. Pan Z, Wang J, Liao W, et al. Data-driven ev load profiles generation using a variational auto-encoder. *Energies*. 2019;12(5):849.

45. Chen Y, Wang Y, Kirschen D, Zhang B. Model-free renewable scenario generation using generative adversarial networks. *IEEE Trans Power Syst*. 2018;33(3):3265-3275.

46. Zhu Y, Zabaras N, Koutsourelakis P-S, Perdikaris P. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J Comput Phys*. 2019;394:56-81.

47. Yoon AS, Lee T, Lim Y, et al. Semi-supervised learning with deep generative models for asset failure prediction. arXiv preprint arXiv:1709.00845; 2017.

48. Mylonas C, Abdallah I, Chatzi EN. Deep unsupervised learning for condition monitoring and prediction of high dimensional data with application. *Model Validation and Uncertainty Quantification*; 3:189.

49. Wang Q, Sprague MA, Jonkman J, Johnson N, Jonkman B. Beamdyn: a high-fidelity wind turbine blade solver in the fast modular framework. *Wind Energy*. 2017;20(8):1439-1462.

50. Openfast. https://github.com/openfast; 2018.

51. Griffin DA. Windpact turbine design scaling studies technical area 1-composite blades for 80-to 120-meter rotor. Tech. Rep. NREL/SR-500-29492, Colorado, USA, National Renewable Energy Laboratory; 2001.

52. Blasques JP, Lazarov B. Becas v2, a cross-section analysis tool for anisotropic and inhomogeneous beam sections of arbitrary geometry. RisøTechnicalReportRisø-R 1785, TechnicalU-niversity of Denmark; 2012.

53. Nijssen RPL. Fatigue life prediction and strength degradation of wind turbine rotor blade composites. Contractor Report SAND2006-7810P, Albuquerque, NM, Sandia National Laboratories; 2006.

54. Cadavid MO, Al-Khudairi O, Hadavinia H, Goodwin D, Liaghat GH. Experimental studies of stiffness degradation and dissipated energy in glass fibre reinforced polymer composite under fatigue loading. *Polymers and Polymer Composites*. 2017;25(6):435-446.

55. Larwood S, Musial W. Comprehensive testing of nedwind 12-meter wind turbine blades at NREL. 51; 2000.

56. Chen X. Structural degradation of a large composite wind turbine blade in a full-scale fatigue test; 2017.

57. Vassilopoulos AP, Keller T. *Fatigue of Fiber-Reinforced Composites*: Springer Science & Business Media; 2011.

58. Nieslony A. Rain flow counting algorithm. File Exchange-MATLAB Central. http://www.mathworks.com/matlabcentral/leexchange/3026-rainflow-counting-algorithm

59. Pearson K. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. 1901;2(11):559-572.

60. Jolliffe I. *Principal Component Analysis*: Springer; 2011.

61. Schölkopf B, Smola A, Müller K-R. Kernel principal component analysis. *International Conference on Artificial Neural Networks*; 1997:583-588.

62. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. *science*. 2000;290(5500):2323-2326.

63. Tenenbaum JB, De Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. *science*. 2000;290(5500):2319-2323.

64. Maaten L, Hinton G. Visualizing data using t-sne. *J Mach Learn Res*. 2008;9(Nov):2579-2605.

65. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science*. 2006;313(5786):504-507.

66. Kwok JT-Y, Tsang IW-H. The pre-image problem in kernel methods. *IEEE Trans Neural Netw*. 2004;15(6):1517-1525.

67. Rumelhart DE, Hinton GE, Williams RJ, et al. Learning representations by back-propagating errors. *Cogn Model*. 1988;5(3):1.

68. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *The J Mach Learn Res*. 2014;15(1):1929-1958.

69. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980; 2014.

70. Tieleman T, Hinton G. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Netw Mach Learn*. 2012;4(2):26-31.

71. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV, USA: IEEE; 2016:770-778.

72. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780.

73. LeCun Y, Bottou L, Bengio Y, Haffner P, et al. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278-2324.

74. Abadi M, Barham P, Chen J, et al. Tensorflow: a system for large-scale machine learning. In: 12th {*USENIX*} symposium on operating systems design and implementation ({*OSDI*} 16); 2016:265-283.

75. Paszke A, Gross S, Chintala S, Chanan G, et al. Automatic differentiation in pytorch; 2017.

76. Theano Development Team. Theano: a Python framework for fast computation of mathematical expressions. arXiv e-prints, abs/1605.02688; 2016.

77. Zhang C, Butepage J, Kjellstrom H, Mandt S. Advances in variational inference. *IEEE Trans Pattern Anal Mach Intell*. 2018.

78. Rezende DJ, Mohamed S, Wierstra D. Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082; 2014.

79. Namura N. Wind shear estimation model using load measurement of wind turbine tower and surrogate model. *Wind Energy*. 2020;23(2):327-339.

80. Sohn K, Lee H, Yan X. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*; 2015:3483-3491.

81. Veldkamp HF. Chances in wind energy: a probalistic approach to wind turbine fatigue design; 2006.

82. Higgins I, Matthey L, Pal A, et al. beta-vae: learning basic visual concepts with a constrained variational framework. *ICLR*. 2017;2(5):6.

83. Alemi AA, Poole B, Fischer I, Dillon JV, Saurous RA, Murphy K. Fixing a broken elbo. arXiv preprint arXiv:1711.00464; 2017.

84. Geneva N, Zabaras N. Multi-fidelity generative deep learning turbulent flows. arXiv preprint arXiv:2006.04731; 2020.

85. Maulik R, Fukami K, Ramachandra N, Fukagata K, Taira K. Probabilistic neural networks for fluid flow surrogate modeling and data recovery. *Phys Review Fluids*. 2020;5(10):104401.

86. Tomczak JM, Welling M. Improving variational auto-encoders using householder flow. arXiv preprint arXiv:1611.09630; 2016.

87. Kingma DP, Salimans T, Jozefowicz R, Chen X, Sutskever I, Welling M. Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*; 2016:4743-4751.

88. Berg R, Hasenclever L, Tomczak JM, Welling M. Sylvester normalizing flows for variational inference. arXiv preprint arXiv:1803.05649; 2018.

89. Rezende DJ, Viola F. Taming vaes. arXiv preprint arXiv:1810.00597; 2018.