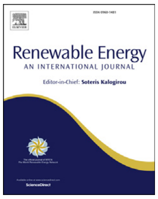
Renewable Energy 185 (2022) 932–944

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene



# Physics-informed deep learning model in wind turbine response prediction

Xuan Li [1], Wei Zhang [2, *]

*Dept. of Civil and Environmental Engineering, University of Connecticut, Storrs, Connecticut, 06269, USA*

A R T I C L E I N F O

*Article history:*
Received 29 January 2021
Received in revised form
27 September 2021
Accepted 13 December 2021
Available online 22 December 2021

*Keywords:*
Physics-informed deep learning model
Wind turbine
Recurrent neural network (RNN)
Long-short term memory (LSTM)
Structure linearization
Response prediction

A B S T R A C T

Subjected to strong cyclic wind and wave loads, wind turbines could experience severe fatigue damages and possibly fail to function normally due to accumulated damages at certain critical locations. Therefore, fatigue damage evaluation and prediction are essential and important to be conducted, which could involve massive numerical simulations and computational costs due to dynamic analyses of the wind turbines under various environmental conditions. To reduce the calculation cost related to the time-consuming dynamic analysis, sequence models such as the recurrent neural network (RNN) and the long-short term memory model (LSTM) originated from the deep learning topic are good and promising candidates to predict structural dynamic responses at multiple critical locations under different environmental scenarios. However, the training cost and prediction accuracy of these deep learning models might not be satisfiable since these models are purely data-driven and require significant amount of training data and a large number of training parameters. To reduce the computational cost and improve the prediction accuracy, a hybrid method that integrates the physical information of the underlying wind turbine system into the data-driven model is implemented in the present study as a computationally efficient simulation model. Structural properties and linearized representations of the wind turbine system are served as the physical constraints and applied in a recently proposed deep residual recurrent neural network (DR-RNN) to form as a physics-informed deep learning model. This physics-informed model is first applied to a frame structure with four degrees of freedom as a benchmark study to show the accuracy and efficiency of this model. The applicability of this physics-informed model to a complex wind turbine system is then investigated, and the performance of the developed physics-informed model on the structural response prediction is also compared with a regular data-driven model.

## 1. Introduction

Long-term performance assessments of wind turbines typically require a large number of dynamic simulations, which could be time-consuming or even prohibitive, especially when more and more varying influence factors are to be included in dynamic simulations [1,2]. Meanwhile, structural dynamic simulations typically require a long runtime, especially for the complex structures with many degrees of freedom and subjected to complicated external forces. Such a time-consuming process might work well for investigating structural performance under certain critical scenarios. However, the long runtime of performing numerical simulations could possibly prevent the numerical models from applications that require many simulations, such as sensitivity analysis, uncertainty analysis, and lifetime assessment [3]. Furthermore, the number of simulations required to cover the entire parameter space increases exponentially as the number of considered parameters increases. A reduced simulation resolution or a simplified physical process might need to be employed in order to decrease the runtime for each simulation and make a large number of simulations numerically tractable. To address the long runtime issue, many different types of surrogate models have been proposed, which have the potential of speeding up the structural evaluation without sacrificing accuracy and structural details [4].

[*] Corresponding author.
*E-mail address:* wzhang@uconn.edu (W. Zhang).

[1] Graduate Student, Dept. Of Civil and Environmental Engineering, University of Connecticut, Storrs, Connecticut 06269.

[2] Associate Professor, Dept. Of Civil and Environmental Engineering, University of Connecticut, Storrs, Connecticut 06269, wzhang@uconn.edu

https://doi.org/10.1016/j.renene.2021.12.058
0960-1481/© 2022 Elsevier Ltd. All rights reserved.

Surrogate models can be used to overcome the computational burden of multi-query tasks such as structural dynamic systems governed by large-scale partial differential equations or linear algebraic equations [5–7]. In general, surrogate models can be categorized into two groups. One is the data-driven approach, and the other is the projection-based method. The data-driven surrogate model, also known as the response surface and black box model, directly builds up the relationship between the complex model outputs and inputs based on a certain number of numerical simulations. Multiple concepts have been implemented to achieve the fitting process, such as Kriging models [8–10], polynomial chaos expansions [11], Gaussian processes [12,13], neural networks [14,15], radial basis functions [16,17], support vector machines [18], dynamic mode analysis [19], genetic programming [20,21] and Bayesian networks [22,23]. The projection-based method, also known as the reduced-order, reduced basis, and model reduction method, project the system governing equations onto a reduced dimension subspace characterized by a basis of orthonormal vectors. Similar to the data-driven surrogate model, the projection-based method also consists of several implementations, including proper orthogonal decomposition (POD) [24,25], Krylov subspace method [26,27], reduced basis method [28,29], dynamic mode decomposition [30], proper generalized decomposition [31,32], and Karhunen-Loeve expansion [33]. Both of these two general types of methods have their own merits and limitations. The data-driven surrogate model behaves like a black box. No knowledge and perception about the underlying system is required in the learning process, which makes this method convenient to use and apply for complex systems not fully understood. Nevertheless, the accuracy of the data-driven model is strongly dependent on the number of training data, and the trained surrogate model might only provide accurate results for the input conditions similar to those from the training data. The projection-based method, in contrast, could possibly provide accurate predictions for the input conditions that have never been seen in the model training process since the internal physical constraints are retained in this type of surrogate model. However, this method could also lose critical physical information of the underlying dynamic system. The information loss increases when the original space dimension is decreased significantly to achieve a low computational cost. Therefore, it is necessary to combine the aforementioned two approaches together to generate a more accurate and robust surrogate model for structural system dynamic response predictions. In this hybrid model, the physical constraints of the investigated dynamic system should be integrated into a pure data-driven approach in order to take full advantage of those two types of surrogate models.

One potential underlying data-driven model that can be applied in the new hybrid model to predict the structural system responses is the recurrent neural network (RNN), which is a class of artificial neural networks where connections between each neural node form a directed graph along a temporal sequence [34,35]. The RNN model is attractive and powerful in sequential result prediction and has been successfully applied to various modeling tasks related to time sequence processing, such as automatic speech recognition and language translation [36–38]. Furthermore, the RNN model is also transferred to the structural engineering domain and has successfully demonstrated its capability to emulate the dynamic system's evolution based on its sequential characteristics in many applications [39–41]. Choe et al. succesfully applied two different types of RNN model, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) neural networks, in structural damage detection of floating offshore wind turbine (FOWT) blades [57]. Agga et al. proposed two variants of LSTM models to effectively predict the power production of a self-consumption photovoltaics plant and an accurate result is achieved by these models [55]. Chen et al. used the RNN approach to assess sequential condition monitoring data of the wind turbine and the model performance has been verified by a case study using real-world wind turbine condition monitoring data [56]. Therefore, the RNN model could be a strong candidate for modeling the dynamic responses of structural systems, and it is capable of obtaining the dynamic response of any nonlinear dynamic system with substantial flexibility and high efficiency. Similar to other pure data-driven approaches, the RNN model, however, is still purely data-driven without any knowledge of the physical constraints and the governing dynamic systems. This might result in a larger number of training data and therefore increase the training cost in order to generate an accurate surrogate model. Besides the size of training data, the RNN model also has multiple hyperparameters that can be tuned during the training process and therefore make the computational cost of the training process further increase. Meanwhile, the accuracy of the trained RNN model strongly depends on the number of training samples, which are generated from the costly numerical simulations. To reduce the cost as well as to improve the model accuracy, a physics-informed deep learning model called deep residual recurrent neural network (DR-RNN), which integrates physical information into the regular RNN model, is implemented in this study to simulate the dynamics of the wind turbines. The key idea of this deep learning model is to embed the underlying physics of the structural dynamic system, e.g. the governing equations of structure dynamics, into the model during the training and predicting process. With the enforced physical constraints, the physics-informed deep learning model could achieve a good generalization capability for the simulated dynamic system, which is normally superior to that of the pure data-driven deep learning model. Furthermore, a reduction of training cost and improvement of model prediction accuracy could also be achieved in this physical constrained model. The most appealing feature of the physics-informed deep learning model is to bring scientific knowledge and deep learning model together such that the underlying physics of the dynamic system could be integrated into the purely data-driven model [42]. In other words, the physics-informed deep learning model introduces scientific constraint as an essential component in the learning process and further makes the data-driven model physically interpretable. Recently, the physics-informed deep leering model has been applied in several engineering applications to reduce the computational complexity associated with high-fidelity numerical simulations, and an accurate prediction is achieved in these studies [43,44].

In the present study, a physics-informed deep learning model with structural state properties integrated is introduced for the wind turbine's dynamic simulations. After the introduction section, the paper is organized as follows. At first, the standard recurrent neural network (RNN) model is reviewed, and then the deep residual recurrent neural network (DR-RNN) is introduced as the physics-informed learning model. After that, a simple frame structure with four degrees of freedom subjected to varying external forces is used to demonstrate the efficiency and prediction accuracy of the physics-informed deep learning model. The complete nonlinear aeroelastic equations of motion for the NREL 5-MW reference wind turbine (Jonkman et al., 2009) as shown in Fig. 1 and the corresponding linearized representation of the equations in the state space are then presented to serve as the physical constraints. The trained physical-informed model under arbitrary initial states, controls, and wind disturbances is used to show the applicability of the model in wind turbine response predictions. Meanwhile, the response prediction performance of the physics-informed model is also compared with the purely data-driven LSTM model to show

**Fig. 1.** NREL 5-MW reference wind turbine.

the advantage of integrating structural physical constraints. Finally, some brief discussions are delivered.

## 2. Physics-informed deep learning model

### 2.1. Standard recurrent neural network model

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. Unlike the regular feedforward neural networks with only one input vector, RNN has one additional temporal dimension allowing it to receive the input vectors at different time steps. Meanwhile, each node in the RNN can maintain an internal state, serving as memory, to remember information from previous time steps and process the input vector at the current time step together with its internal state to generate the output vectors. With these advanced features, RNN exhibits strong potential in sequential data processing and has been proven to perform extremely well on temporal data. The basic RNN architecture, as well as the detailed operations for a single time step of an RNN cell, are shown in Fig. 2. Meanwhile, the hidden state vector and the output vector at each time step, as shown in Fig. 2 can be evaluated using the following equations:

$$\boldsymbol{a}^{<t>} = \tanh\left(\boldsymbol{W}_{ax}\boldsymbol{x}^{<t>} + \boldsymbol{W}_{aa}\boldsymbol{a}^{<t-1>} + \boldsymbol{b}_a\right) \tag{1}$$

$$\boldsymbol{y}^{<t>} = f_a\left(\boldsymbol{W}_{ya}\boldsymbol{a}^{<t>} + \boldsymbol{b}_y\right) \tag{2}$$

where $\boldsymbol{x}^{<t>}$ is the input vector at time $t$; $\boldsymbol{a}^{<t>}$ is the hidden state vector at time $t$; $\boldsymbol{y}^{<t>}$ is the output vector at time $t$; $\boldsymbol{W}_{ax}$, $\boldsymbol{W}_{aa}$, and $\boldsymbol{W}_{ay}$ are the weight matrices in RNN model for the input vector, hidden state vector, and output vector, respectively; Similarly, $\boldsymbol{b}_a$ and $\boldsymbol{b}_y$ are the bias vectors in RNN model for the hidden state vector and output vector; and $f_a$ is a specified activation function to be used in the RNN model. It is noteworthy that all weight and bias parameters in the RNN model are the same for each RNN cell, which means these model parameters are shared by each cell rather than modeled independently, and thus substantially reduce the number of training parameters and make the RNN feasible for long time sequence modeling. As shown in Eq. (1), the hidden state at the current time step depends on its corresponding values from the previous time step and the input at the current time step. Therefore, the hidden state in the RNN model is usually thought of as a memory of the system and contains certain useful information from the past. A variety of activation functions can be selected to generate the outputs of the system. In this study, the linear activation function is used, which is suitable for predicting real values, e.g., the dynamic responses of the structural system in this study. Similar to other neural networks, RNN model parameters $\boldsymbol{\theta} = [\boldsymbol{W}_{ax}, \boldsymbol{W}_{aa}, \boldsymbol{W}_{ay}, \boldsymbol{b}_a, \boldsymbol{b}_y]$ are trained iteratively using the backpropagation technique on the training data set [45]. During the training process, the loss function needs to be minimized over time in order to improve the RNN model performance, and the loss function used in this study is defined as:

$$L(\boldsymbol{\theta}) = \frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}\left(\boldsymbol{y}_{pred}^{(n)<t>} - \boldsymbol{y}_{true}^{(n)<t>}\right)^2 \tag{3}$$

Where $N$ is the total number of training samples in the training dataset; $T$ is the number of time steps for each training sample; $\boldsymbol{y}_{pred}^{(n)<t>}$ and $\boldsymbol{y}_{true}^{(n)<t>}$ are the predicted output from the RNN model and the observed output, respectively, for the *n-th* training sample at the *t-th* time step. The weight parameters are updated based on the partial derivative of the above loss function with respect to the current weight in each iteration of training. When the absolute value of the gradient of the selected activation function is less than one, the calculated derivatives at the front RNN cells could be vanishingly small based on the chain rule, which could significantly prevent the weight from changing its value. Therefore, the training process for the whole neural network, especially for the front RNN cells, could be very slow and even completely stopped from further training. Similar to deep neural networks with multiple hidden layers, this kind of problem is so-called gradient vanishing, which deteriorates the performance of the basic RNN network with a large number of RNN cells to simulate a longer time sequence [46].

### 2.2. Long short term memory model

Long Short Term Memory (LSTM) model is a special type of RNN that is capable of learning long-term dependencies and, therefore, could avoid the gradient vanishing problem that occurred in the RNN for a long sequence of data [47]. Similar to the RNN, the LSTM also has a chain-like structure with repeating modules of the neural network. However, instead of having a single activation in each cell like the RNN, the LSTM has four different activations interacting in a very special way to control the information flow. Furthermore, there is an additional cell state being tracked and updated at every time step by the LSTM besides the hidden state. A detailed LSTM cell structure is shown in Fig. 3 as an example and the parameters occurred in this figure could be evaluated using the following expressions:

$$\boldsymbol{\Gamma}_f^{<t>} = \sigma\left(\boldsymbol{W}_f\left[\boldsymbol{a}^{<t-1>}, \boldsymbol{x}^{<t>}\right] + \boldsymbol{b}_f\right) \tag{4}$$

$$\boldsymbol{\Gamma}_u^{<t>} = \sigma\left(\boldsymbol{W}_u\left[\boldsymbol{a}^{<t-1>}, \boldsymbol{x}^{<t>}\right] + \boldsymbol{b}_u\right) \tag{5}$$

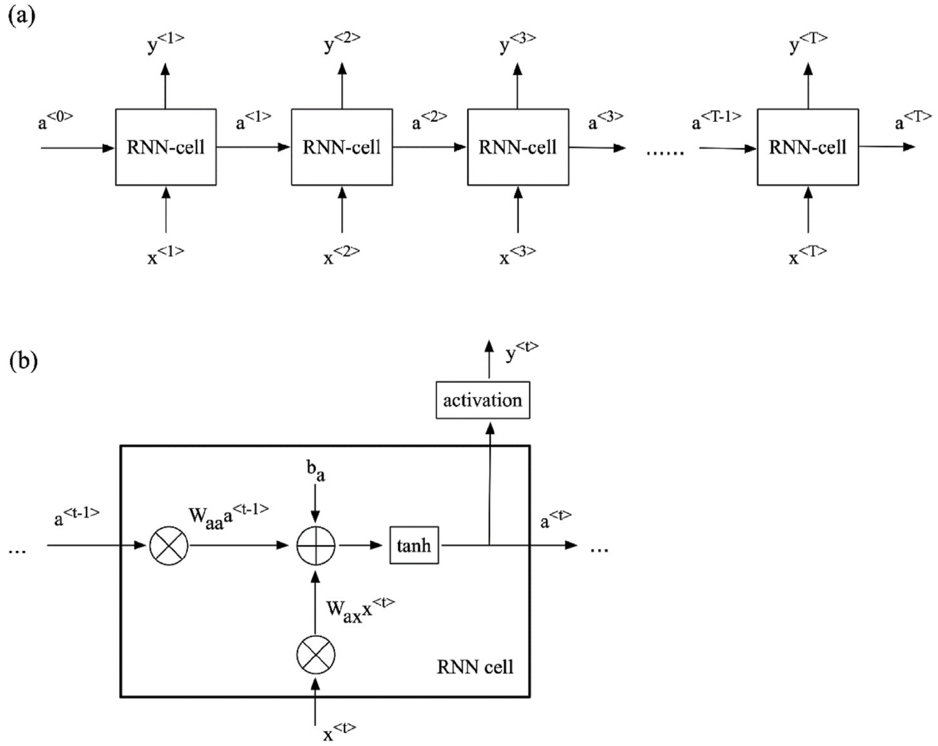


**Fig. 2.** Architecture of recurrent neural network (RNN): (a) basic RNN model and (b) basic RNN cell.

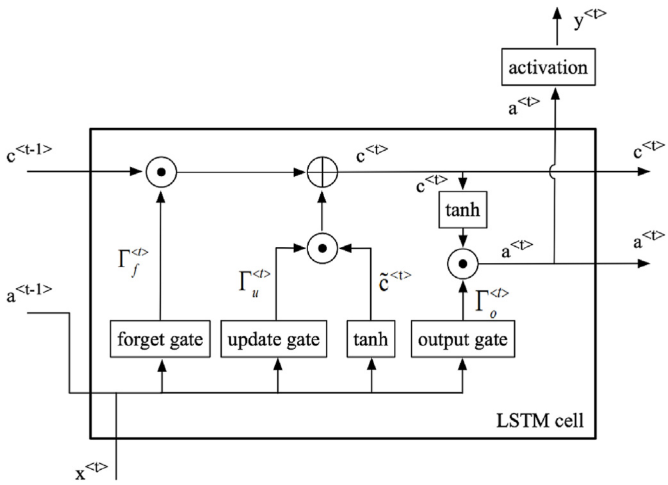


**Fig. 3.** LSTM cell structure.

$$\boldsymbol{\Gamma}_o^{<t>} = \sigma\left(\boldsymbol{W}_o\left[\boldsymbol{a}^{<t-1>}, \boldsymbol{x}^{<t>}\right] + \boldsymbol{b}_o\right) \tag{6}$$

$$\tilde{\boldsymbol{c}}^{<t>} = \tanh\left(\boldsymbol{W}_c\left[\boldsymbol{a}^{<t-1>}, \boldsymbol{x}^{<t>}\right] + \boldsymbol{b}_c\right) \tag{7}$$

$$\boldsymbol{c}^{<t>} = \boldsymbol{\Gamma}_f^{<t>} \circ \boldsymbol{c}^{<t-1>} + \boldsymbol{\Gamma}_u^{<t>} \circ \tilde{\boldsymbol{c}}^{<t>} \tag{8}$$

$$\boldsymbol{a}^{<t>} = \boldsymbol{\Gamma}_o^{<t>} \circ \tanh\left(\boldsymbol{c}^{<t>}\right) \tag{9}$$

where $\boldsymbol{\Gamma}_f^{<t>}$ is the forget gate to decide the extent of getting rid of the previously stored memory; $\boldsymbol{\Gamma}_u^{<t>}$ is the update gate to decide how much new information should be integrated into the new state; $\boldsymbol{\Gamma}_o^{<t>}$ is the output gate to decide which outputs are to be used; $\boldsymbol{W}_f$, $\boldsymbol{W}_u$, and $\boldsymbol{W}_o$ are the weight matrices and $\boldsymbol{b}_f$, $\boldsymbol{b}_u$, and $\boldsymbol{b}_o$ are the bias vectors of the three gates, respectively; $\boldsymbol{W}_c$ and $\boldsymbol{b}_c$ are the weight matrix and bias vector for the calculation of the candidate cell state; $\sigma$ is the sigmoid activation function; $\tilde{\boldsymbol{c}}^{<t>}$ is the new candidate of the cell state; $\boldsymbol{x}^{<t>}$ is the input vector at time $t$; $\boldsymbol{a}^{<t>}$ is the hidden state vector at time $t$; $\boldsymbol{c}^{<t>}$ is the cell state vector at time $t$; $\boldsymbol{y}^{<t>}$ is the output vector at time $t$; and the operator ∘ denotes the element-wise multiplication. As can be seen from Eq. (8), the new cell state is determined based on both the previous cell state and the new candidate cell state that are both controlled, namely filtered, by certain gates. Therefore, the important information from previous and current time steps that should be maintained during the training process could be determined by the LSTM model without continuously losing from gradient vanishing. In addition, the output can also be controlled by the model itself to decide which part of the cell state should be part of the output.

### 2.3. Physics-informed deep learning model

For dynamic systems, the underlying physics is typically described using the governing equations in which the relationships of the system's motions, external forces, constraints, etc., are established. To solve a complex dynamic system, the governing equations are discretized into certain degrees of freedom, and numerical schemes are usually applied to advance the simulation step by step [48,49]. The discretized governing equations for a general nonlinear dynamic system could be expressed as:

$$\frac{d\boldsymbol{y}}{dt} = \boldsymbol{A}\boldsymbol{y} + \boldsymbol{f}(\boldsymbol{y}) \tag{10}$$

where $\boldsymbol{y}$ is the state vector of the dynamic system with its size depending on the total number of degrees of freedom we discretized; $\boldsymbol{A}$ is a matrix representing the linear part of the governing equations, whereas $\boldsymbol{f}$ is a vector representing the nonlinear component of the governing equations. Many time integration schemes are available in the literature to solve the above discretized governing equations. For a demonstration purpose, the above equations could be expressed in the following form using the implicit Euler method.

$$\boldsymbol{y}_{t+1} = \boldsymbol{y}_t + \Delta t \times \boldsymbol{A}\boldsymbol{y}_{t+1} + \Delta t \times \boldsymbol{f}(\boldsymbol{y}_{t+1}) \tag{11}$$

where $\Delta t$ is the predefined time step size. To make the simulation step forward accurately, the residual vector defined in the following equation should be minimized at any given time step by selecting the approximate state vector at the next time step.

$$\boldsymbol{r}_{t+1} = \boldsymbol{y}_{t+1} - \boldsymbol{y}_t - \Delta t \times \boldsymbol{A}\boldsymbol{y}_{t+1} - \Delta t \times \boldsymbol{f}(\boldsymbol{y}_{t+1}) \tag{12}$$

This residual function could be implemented to solve the state vector at the next time step based on the state vector at the current time step and the dynamic system characteristics. Basic RNN could be applied to obtain the dynamic system responses. However, only training parameters and hyperparameters which are determined based on our insights and practices exist in the basic RNN model without any underlying physical information from the dynamic system. To incorporate the physical information into the neural networks, a physics-informed deep residual recurrent neural network (DR-RNN) is implemented where a series of network layers are stacked in each RNN cell to iteratively minimize the residual vector defined in the above equation. The DR-RNN architecture is shown in Fig. 4 and the expressions for the intermediate output vector of each stacked network layer are defined as:

$$\boldsymbol{y}_k^{<t+1>} = \boldsymbol{y}_{k-1}^{<t+1>} - \boldsymbol{W} \circ \tanh\left(\boldsymbol{U}\boldsymbol{r}_k^{<t+1>}\right) \qquad \text{for } k = 1 \tag{13}$$

$$\boldsymbol{y}_k^{<t+1>} = \boldsymbol{y}_{k-1}^{<t+1>} - \frac{\eta_k}{\sqrt{G_k + \varepsilon}}\boldsymbol{r}_k^{<t+1>} \qquad \text{for } k > 1 \tag{14}$$

where $k$ represents the stacked layer number as shown in the above figure; $\boldsymbol{W}$ and $\boldsymbol{U}$ are the weight matrices, and $\eta_k$ is the bias that should be trained in the DR-RNN model; $\varepsilon$ is a smoothing term to avoid division by zero; $\boldsymbol{r}_k^{<t+1>}$ is the residual vector defined by Eq. (12) for the *k-th* stacked layer at the time $t+1$, which is evaluated based on the out vector from the previous stacked layer; and $G_k$ is an exponentially decaying squared norm of the residual vector defined as:

$$G_k = \gamma \left\| \boldsymbol{r}_k^{<t+1>} \right\|^2 + \zeta G_{k-1} \tag{15}$$

where $\gamma$ and $\zeta$ are the fractional factors, which also are hyperparameters that can be adjusted for the DR-RNN model. In this study, the commonly used values of 0.1 and 0.9 are assigned to these two parameters, respectively [50]. In the DR-RNN model, the output of the last stacked layer is treated as the prediction of the output vector at that time step. With the integration of the residual function, the RNN model no longer performs like a purely data-driven model. Instead, the physical constraints of the underlying structural system could be utilized during both the training and predicting process and, therefore, could significantly reduce the training cost and improve the prediction accuracy, especially for certain corner cases that are not covered during the training phase. In this study, the dynamic responses of a frame structure and a wind turbine are predicted through the DR-RNN model with their state-space equations serving as the physical constraint. Therefore, after the model training process, the DR-RNN model is capable of minimizing the residuals consecutively by the stacked layers, which is similar to performing implicit numerical integrations on the structural dynamic equations to obtain system responses at each time step. To be specific, the input vector at each time step is corresponding to the external force exerted on the structural system at that time step. This input vector, together with the output response vector from the previous time step, will serve as the combined input for the first stacked layer of the next time step.

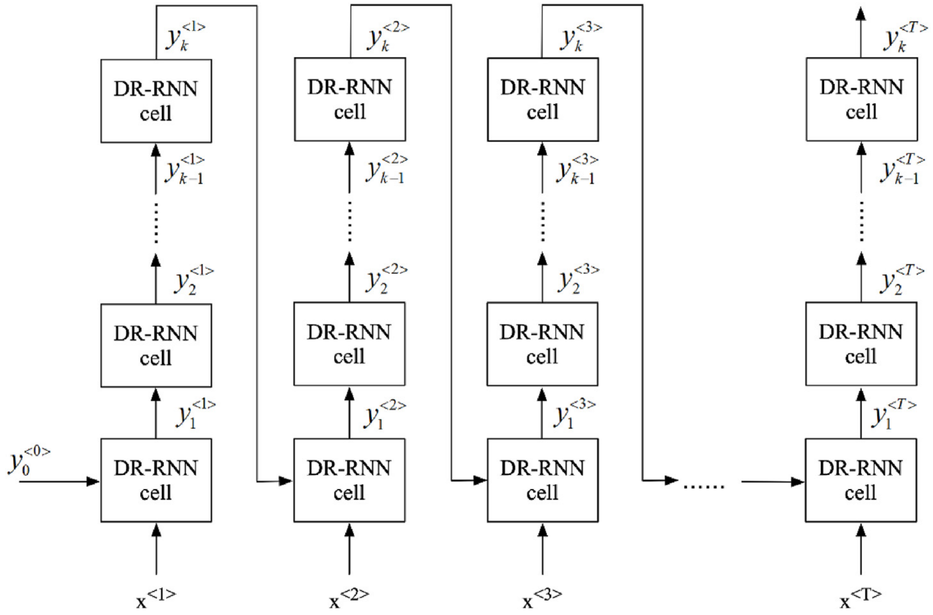


**Fig. 4.** Architecture of deep residual recurrent neural network (DR-RNN).

## 3. Case study on simple frame structure

### 3.1. Frame structure model

A benchmark study on a simple frame structure with four DOFs subjected to given external loads is firstly performed to evaluate and verify the performance of the physics-informed deep learning model in structural dynamic response predictions. The frame structure is shown in Fig. 5 and the structural properties shown in Fig. 5 are defined as k = 60 kips/in, c = 0.5 kip-sec/in, and w = 300 kips. The dynamic forces acting on the frame structure are described by Fig. 6, and the general equations of motion for this frame structure can be written as:

$$\boldsymbol{M}\ddot{\boldsymbol{y}}+\boldsymbol{C}\dot{\boldsymbol{y}}+\boldsymbol{K}\boldsymbol{y}=\boldsymbol{f} \tag{16}$$

Where M is the mass matrix; C is the damping matrix; K is the stiffness matrix; f is the excitation force vector; and y is the structural displacement vector depicted in Fig. 5. These equations of motion will server as the physical constraint in the physics-informed deep learning model mentioned in the last section. Instead of directly applying these equations of motion to the physics-informed learning model, they should be first transformed into the state space in which the representation of the structural system is in the first-order and then server as physical constraints in Eq. (12) to reduce the training error. The equations of motion in the state-space form can be represented as:

$$\dot{\boldsymbol{y}}_S=\boldsymbol{A}\boldsymbol{y}_S+\boldsymbol{f}_S \tag{17}$$

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C}\end{bmatrix} \tag{18}$$

$$\boldsymbol{y}_S=\begin{bmatrix}\boldsymbol{y}\\ \dot{\boldsymbol{y}}\end{bmatrix} \tag{19}$$

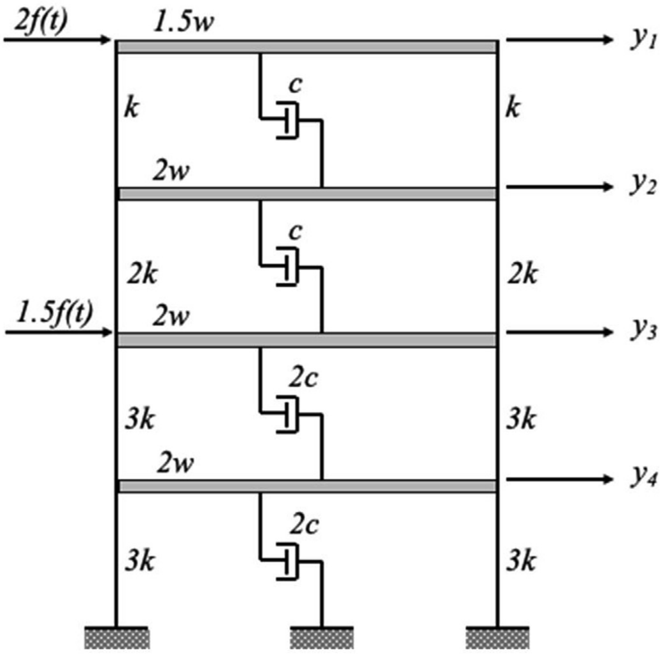


**Fig. 5.** Four DOFs frame structure subjected to dynamic loadings.

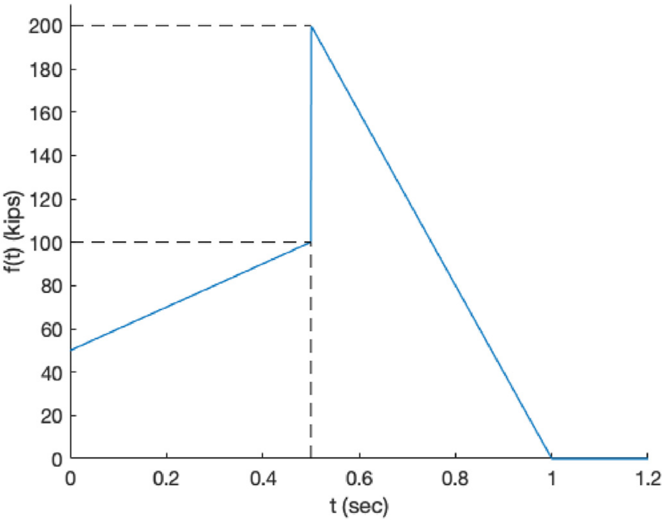


**Fig. 6.** Time-varying force acting on the frame structure.

$$\boldsymbol{f}_S=\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{M}^{-1}\boldsymbol{f}\end{bmatrix} \tag{20}$$

where ***A*** is the state matrix; ***ys*** is the state vector; ***fs*** is the state force vector; ***0*** is a matrix or a vector of zeros for Eq. (18) and Eq. (20), respectively; and ***I*** is the identity matrix.

### 3.2. Assessment of physics-informed deep learning model

After obtaining the state-space equations of the investigated structural system, the physics-informed deep learning model can be trained with the state-space equations integrated in each DR-RNN cell and with arbitrary initial conditions and the given time-varying external forces as the model inputs as shown in Fig. 4. The outputs of the trained model are the horizontal displacement of the four rigid frame girders at each time step. To train the model, training data from real numerical simulations are required in order to calculate the loss function in Eq. (3), as well as to update the training parameters from partial derivatives calculation in the backpropagation process. In the present study, the training data is obtained from solving the equations of motion for the frame structure, i.e., Eq. (16), by using the Wilson-$\theta$ method. The initial displacements for those four frame girders are generated randomly from a uniform distribution from −0.3 to 0.3 inches, and the duration of numerical simulation is 5 s. The physics-informed deep learning model is established using the Keras framework running on top of the TensorFlow [51] and is trained using the Adam optimizer [52] on 500 samples. Multiple choices of the number of stacked layers in the DR-RNN model are tried, and the results indicate that the DR-RNN model with three stacked layers has the best performance with a training error of 0.023 and a test error of 0.031 based on the mean square error.

To demonstrate the response prediction performance of the trained DR-RNN model, two prediction cases are considered in the present study. In the first case, the initial conditions for all degrees of freedom of the frame structure are set to zero, which represents the case enveloped by the range of the training data but not exactly trained during the model training process. In the second case, the initial conditions are randomly selected as 0.4283 inches for the 1st DOF, −0.5144 inch for the 2nd DOF, −0.5075 inch for the 3rd DOF, and 0.2391 inches for the 4th DOF to represent the case outside the training space. The first case is to demonstrate the prediction

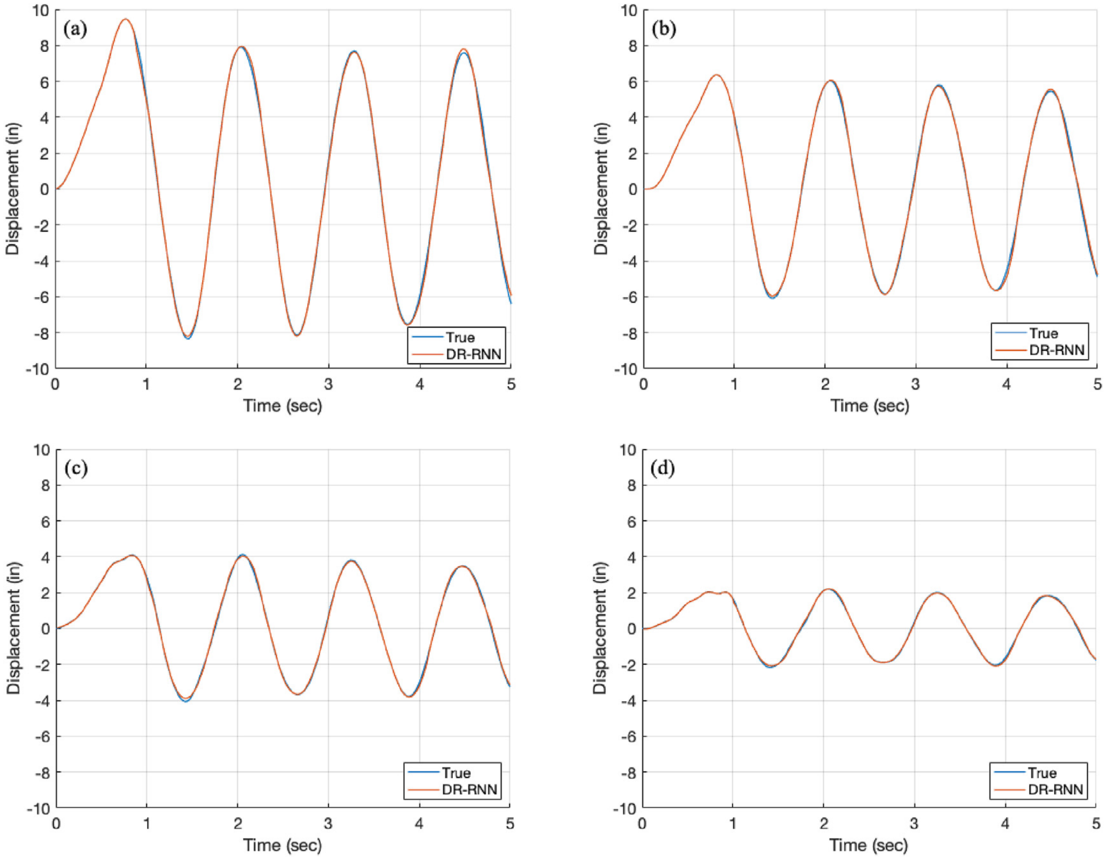


**Fig. 7.** Structural response prediction based on trained DR-RNN model for case 1: (a) 1st DOF, (b) 2nd DOF, (c) 3rd DOF, and (d) 4th DOF.

accuracy of the trained model in the original training space, whereas the second case is to show the robustness of the trained model when the input is never seen by the trained model and is far away from the training space. The prediction responses for all four degrees of freedom of the frame structure in the first case are shown in Fig. 7.

As shown in Fig. 7, the response predictions for all four degrees of freedom based on the trained DR-RNN model match very well with the true responses obtained from the direct numerical integration method (Wilson-$\theta$ method). Meanwhile, it is noteworthy that the DR-RNN model is trained using a time step of 0.01 s which is larger than the time step of 0.001 s used in the direct integration method to expedite the training speed. The results indicate that using a larger time step in the DR-RNN model than that of the input data will not deteriorate the DR-RNN model performance on structural response predictions. This good prediction performance can be attributed to the integration of the physical information, i.e., the structural equations of motion, in the learning model to make the model to learn its training parameters based on physical constraints instead of purely performing curve fitting with a large amount of training data. To show the robustness of the trained DR-RNN model, the prediction responses for all four degrees of freedom of the frame structure in the second case are shown in Fig. 8. As shown in Fig. 8, response predictions obtained from the trained DR-RNN model still have a good match with the true responses even for the case with initial conditions deviating far away from the training space. Comparing with traditional data-driven models, which usually have poor predictions for new data offset from the initial training data, the superiority of the physics-informed DR-RNN model is distinct by integrating structural dynamic constraints into the learning and predicting process.

## 4. Response prediction on wind turbine

### 4.1. Wind turbine structural model

The application of the physics-informed DR-RNN model has reflected its advantages in structural response predictions, as shown in the last section. In this section, this physics-informed DR-RNN model will be further applied to a more complex structure, i.e., the NREL 5-MW reference wind turbine (Jonkman et al., 2009), subjected to wind loads to invest the model performance on a larger scale structural system. The complete nonlinear aeroelastic equations of motion of the wind turbines can be expressed as follows:

$$\boldsymbol{M}(\boldsymbol{q},\boldsymbol{u},t)\ddot{\boldsymbol{q}}+\boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},\boldsymbol{u},\boldsymbol{u}_d,t)=\boldsymbol{0} \tag{21}$$

where $\boldsymbol{M}$ is the wind turbine mass matrix; $\boldsymbol{f}$ is the nonlinear forcing function vector; $\boldsymbol{q}$ is the vector of the selected structural DOF displacements; $\dot{\boldsymbol{q}}$ is the vector of the corresponding structural DOF velocities; $\ddot{\boldsymbol{q}}$ is the vector of the corresponding structural DOF accelerations; $\boldsymbol{u}$ is the vector of wind turbine control inputs; $\boldsymbol{u}_d$ is the vector of wind input disturbances; and $t$ is time. These nonlinear equations of motion should be firstly linearized in order to be applied as physical constraints in the physics-informed DR-RNN model. The linearization could be achieved by using small perturbation theory, in which each of the structural system variables can be expressed in the following form about their respective operating point values.

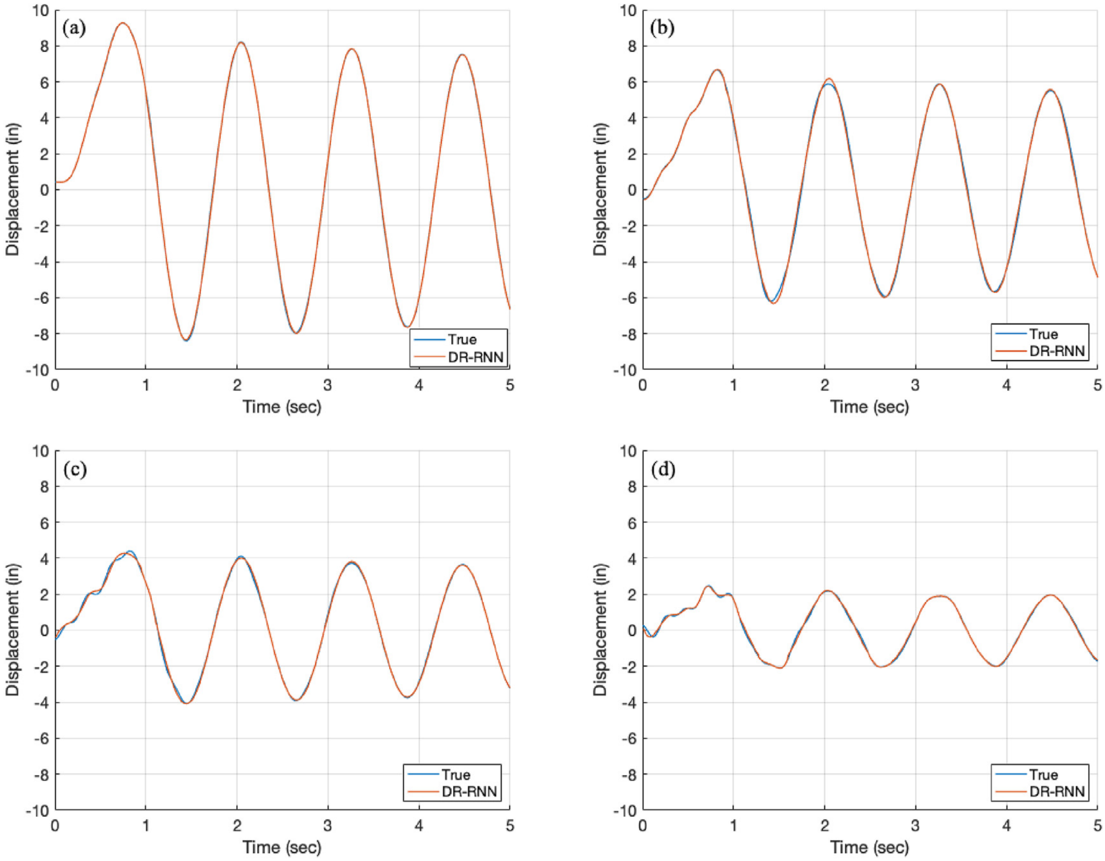


**Fig. 8.** Structural response prediction based on trained DR-RNN model for case 2: (a) 1st DOF, (b) 2nd DOF, (c) 3rd DOF, and (d) 4th DOF.

$$\begin{aligned} \boldsymbol{q} &= \boldsymbol{q}_{op} + \Delta\boldsymbol{q} \\ \dot{\boldsymbol{q}} &= \dot{\boldsymbol{q}}_{op} + \Delta\dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} &= \ddot{\boldsymbol{q}}_{op} + \Delta\ddot{\boldsymbol{q}} \\ \boldsymbol{u} &= \boldsymbol{u}_{op} + \Delta\boldsymbol{u} \\ \boldsymbol{u}_d &= \boldsymbol{u}_{dop} + \Delta\boldsymbol{u}_d \end{aligned} \tag{22}$$

Substituting these expressions into the nonlinear form of equations of motion and ignoring high-order terms in the Taylor series expansion will lead to the second-order linearized form of the structural system equations expressed below:

$$\boldsymbol{M}\Delta\ddot{\boldsymbol{q}} + \boldsymbol{C}\Delta\dot{\boldsymbol{q}} + \boldsymbol{K}\Delta\boldsymbol{q} = \boldsymbol{F}\Delta\boldsymbol{u} + \boldsymbol{F}_d\Delta\boldsymbol{u}_d \tag{23}$$

$$\boldsymbol{M} = \boldsymbol{M}|_{op} \tag{24}$$

$$\boldsymbol{C} = \left.\frac{\partial \boldsymbol{f}}{\partial \dot{\boldsymbol{q}}}\right|_{op} \tag{25}$$

$$\boldsymbol{K} = \left.\left[\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{q}}\ddot{\boldsymbol{q}} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\right]\right|_{op} \tag{26}$$

$$\boldsymbol{F} = -\left.\left[\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{u}}\ddot{\boldsymbol{q}} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right]\right|_{op} \tag{27}$$

$$\boldsymbol{F}_d = -\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}_d}\right|_{op} \tag{28}$$

where $\boldsymbol{M}$ is the mass matrix; $\boldsymbol{C}$ is the damping matrix; $\boldsymbol{K}$ is the stiffness matrix; $\boldsymbol{F}$ is the control input matrix; $\boldsymbol{F}_d$ is the wind input disturbance matrix; and $|_{op}$ notation is used to indicate that the partial derivatives are calculated at the specified operating point, and the steady-state periodic operating point is used in the present study. Meanwhile, the output measurements besides the structural DOF displacements, such as internal forces and moments at certain locations of the wind turbine, can also be expressed in the linearization form, and the corresponding second-order linearization is written as follows:

$$\boldsymbol{y} = \boldsymbol{O}_v\Delta\dot{\boldsymbol{q}} + \boldsymbol{O}_d\Delta\boldsymbol{q} + \boldsymbol{D}\Delta\boldsymbol{u} + \boldsymbol{D}_d\Delta\boldsymbol{u}_d \tag{29}$$

where $\boldsymbol{y}$ is the desired output measurements; $\boldsymbol{O}_v$ is the velocity output matrix; $\boldsymbol{O}_d$ is the displacement output matrix; $\boldsymbol{D}$ is the control input transmission matrix; and $\boldsymbol{D}_d$ is the wind input disturbance transmission matrix.

Similar to the frame structure described in the last section, the second-order linearized equations of motion expressed above should be transformed into the first-order representation in the state space in order to be applied to the physics-informed DR-RNN model. The state vector $\boldsymbol{x}$ and state derivative vector $\dot{\boldsymbol{x}}$ of the linearized structural system can be defined as:

$$\boldsymbol{x} = \begin{Bmatrix} \Delta\boldsymbol{q} \\ \Delta\dot{\boldsymbol{q}} \end{Bmatrix} \tag{30}$$

$$\dot{\boldsymbol{x}} = \begin{Bmatrix} \Delta\dot{\boldsymbol{q}} \\ \Delta\ddot{\boldsymbol{q}} \end{Bmatrix} \tag{31}$$

And therefore, the first-order state-space representation of the structural equations of motions can be expressed as follows:

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\Delta\boldsymbol{u} + \boldsymbol{B}_d\Delta\boldsymbol{u}_d \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\Delta\boldsymbol{u} + \boldsymbol{D}_d\Delta\boldsymbol{u}_d\end{aligned} \tag{32}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \tag{33}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{F} \end{bmatrix} \tag{34}$$

$$\boldsymbol{B}_d = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{F}_d \end{bmatrix} \tag{35}$$

$$\boldsymbol{C} = [\,\boldsymbol{O}_d \quad \boldsymbol{O}_v\,] \tag{36}$$

where $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{0}$ is a matrix of zeros. The sizes of the above matrices and vectors depend on the number of structural degrees of freedom enabled as well as the number of control inputs, wind input disturbances, and output measurements selected. In the present study, the enabled degrees of freedom include first and second flapwise blade mode DOF and first edgewise blade mode DOF for three turbine blades, drivetrain and generator DOF, nacelle yaw DOF, first and second fore-aft tower bending mode DOF, and first and second side-to-side tower bending mode DOF. The collective blade pitch angle is selected as the control inputs. The horizontal wind speed and the wind propagation direction are selected as the wind input disturbances. Furthermore, the desired output measurements consist of six force components at the tower base section, as shown in Fig. 9.

### 4.2. Applicability of physics-informed deep learning model on wind turbine

The derived first-order state-space equations of the wind turbine can be directly integrated into the physics-informed DR-RNN model to serve as the physical constraint during the model training and prediction process. The training data are generated from numerical simulations using the multiphysics code OpenFAST (Jonkman and Buhl Jr 2005), and the duration of the simulation is 10 s. The initial displacements of the tower bending mode DOFs are sampled from a zero-mean Gaussian distribution with a standard deviation of 0.1 m. The wind speed is sampled from a uniform distribution from 8 to 20 m/s, and the wind direction is sampled from a uniform distribution from −30 to 30°. The collective blade pitch angle is sampled from a uniform distribution from 0 to 15°. A total of 500 samples are used to train the physics-informed DR-RNN model based on the Keras framework with the TensorFlow as backend and the Adam optimizer. Same as the DR-RNN model established for the frame structure, three stacked layers are also used in the DR-RNN model for wind turbine response predictions.

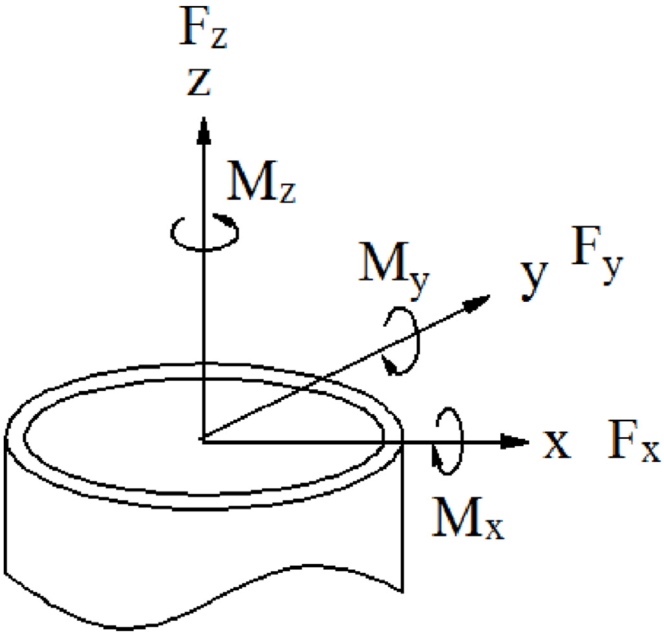


**Fig. 9.** Output measurements of six force components.

The response prediction of the trained DR-RNN model for a test case with zero initial tower bending mode displacements, 12 m/s wind speed, 20° wind propagation direction, and 10° collective blade pitch angle is shown in this section to demonstrate the performance of the physics-informed DR-RNN model for the wind turbine. The response prediction results for the six desired force and moment measurements at the tower base section are shown in Fig. 10.

As shown in Fig. 10, the response predictions for six force components at the tower base section based on the trained physics-informed DR-RNN model match well with the true responses in general. The trained DR-RNN model has a very good prediction performance in the range where the force component does not have a very high-frequency oscillation. However, the trained DR-RNN model cannot accurately capture the high-frequency responses and can only capture the underlying low-frequency responses. The bad performance of the model when predicting high-frequency response oscillations might be due to the physical constraint, which is the wind turbine state-space equations, integrated in the physics-informed DR-RNN model. In the present study, only the structural state-space equations linearized at the steady-state periodic operating point of the wind turbine are used as the physical constraint in the physics-informed DR-RNN model. According to the perturbation theory, the linearized equations of motion are only valid and accurate under small oscillations about the linearization point, which is the wind turbine steady-state periodic operation point in this study. When structural states of the wind turbine deviate too far from this selected operation point, the linearized structural model might not be a suitable candidate as the physical constraint in the physics-informed DR-RNN model. Furthermore, the linearized structural state-space equations are different at different rotor azimuth angles, and typically the azimuth-averaged state-space equations computed from all available periodic system state equations at different rotor azimuth angles are sufficient in the wind turbine control system analysis. However, only integrating linearized structural state-space equations at one operation point might not be the case for the structural response predictions, since the structural response at each time step might be very sensitive to the selected linearized state-space equations. As shown in Fig. 9b and d, the force and moment within the rotor plane have a high-frequency oscillation in the initial transient phase as the rotor gets started to rotate, and the trained DR-RNN model has a lousy prediction performance in this phase. Therefore, a method that can combine all linearized system state-space equations at different wind turbine operation points into the physics-informed DR-RNN model and automatically determine the most suitable state-space equations to be used at each time step needs to be developed in the future work.

The superiority of the physics-informed DR-RNN model on the long-term structural response prediction is also well demonstrated in Fig. 10. The shaded area is corresponding to the selected time duration of 10 s for the training data used to train the DR-RNN model. In the test case, instead of only predicting the structural

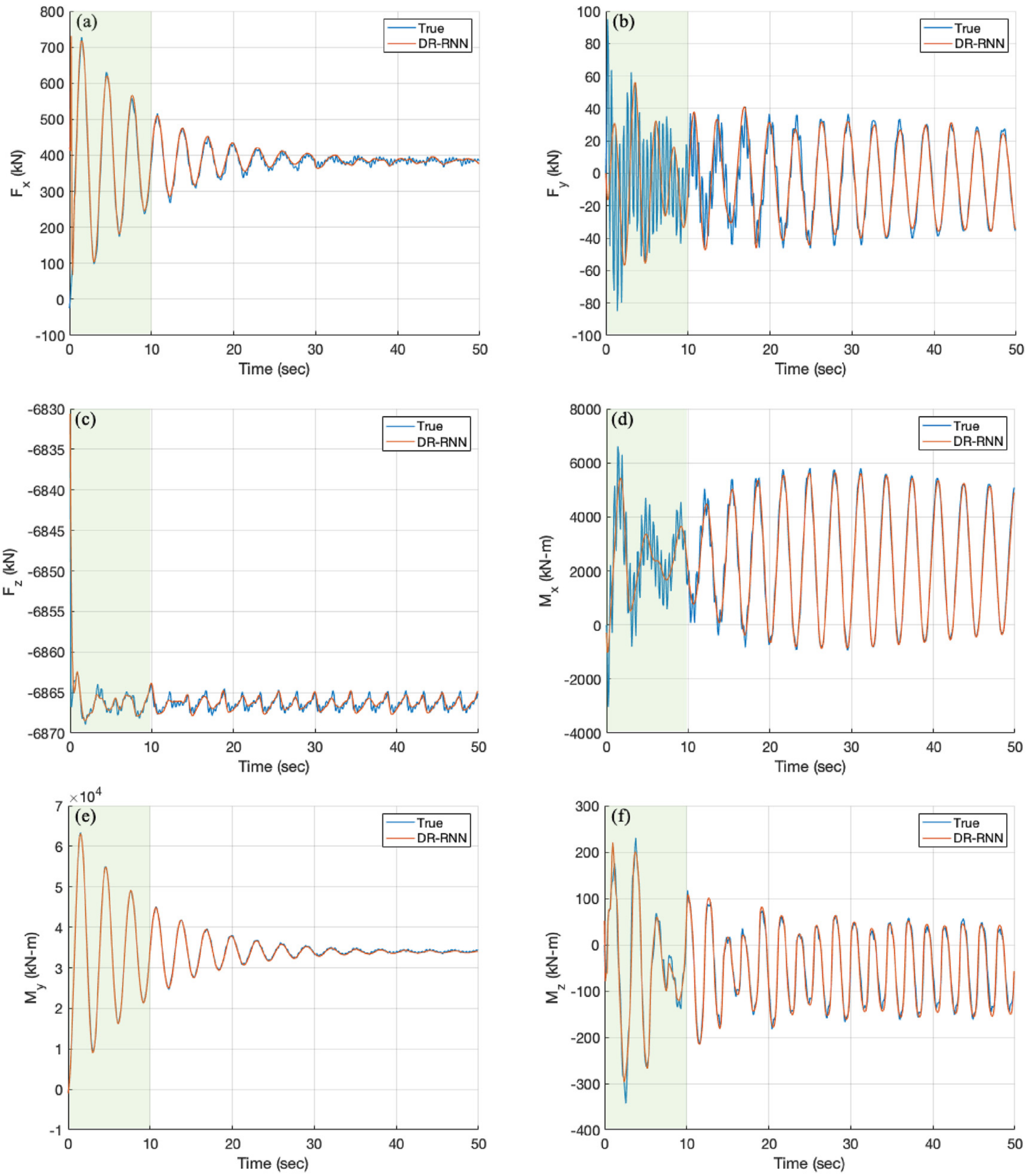


**Fig. 10.** Response predictions of the trained DR-RNN model for forces and moments at the tower base section: (a) force along x-axis, (b) force along y-axis, (c) force along z-axis, (d) moment about x-axis, (e) moment about y-axis, (f) moment about z-axis. Response in the shaded area is corresponding to the selected time duration in the original trained model.

responses for the first 10 s, the trained model is extended to predict the structural responses for a much longer duration. A good match between the model prediction and the true response could be observed in Fig. 10 for the remaining 40 s, which indicates that the physics-informed DR-RNN model has strong capability and potential in long-term response predictions even though it is trained on a dataset with shorter duration. This advantageous characteristic of the physics-informed DR-RNN model can be attributed to the integration of the structural dynamic constraints into the deep learning model.

To further demonstrate the advantage of integrating structural dynamic constraints in the deep learning model, a purely data-driven model based on the long short term memory (LSTM) network is also trained using the same training data, optimizer, and hyperparameters as the physics-informed DR-RNN model. The dimension of the hidden layer in the LSTM model is set to the dimension of the state variables in the linearized structural state-space equations in order to achieve a similar amount of model training parameters to the DR-RNN model. It is noteworthy that the training time for the LSTM model is very close to that for the physics-informed DR-RNN model when using the same training dataset, hyperparameters, and optimizer, so adding physical constraints to the data-driven model doesn't affect the model performance in terms of the training cost. The trained LSTM model is also used to predict the force and moment responses at the tower base section for a longer duration of 20 sections than the original training duration of 10 s, and its performance comparison with the DR-RNN model is shown in Fig. 11. The LSTM model could generally well capture the structural responses within the original training duration but does not match well with the true responses compared with the DR-RNN model. Better performance of physics-informed model are also found in other studies for solving partial differential equations even with a

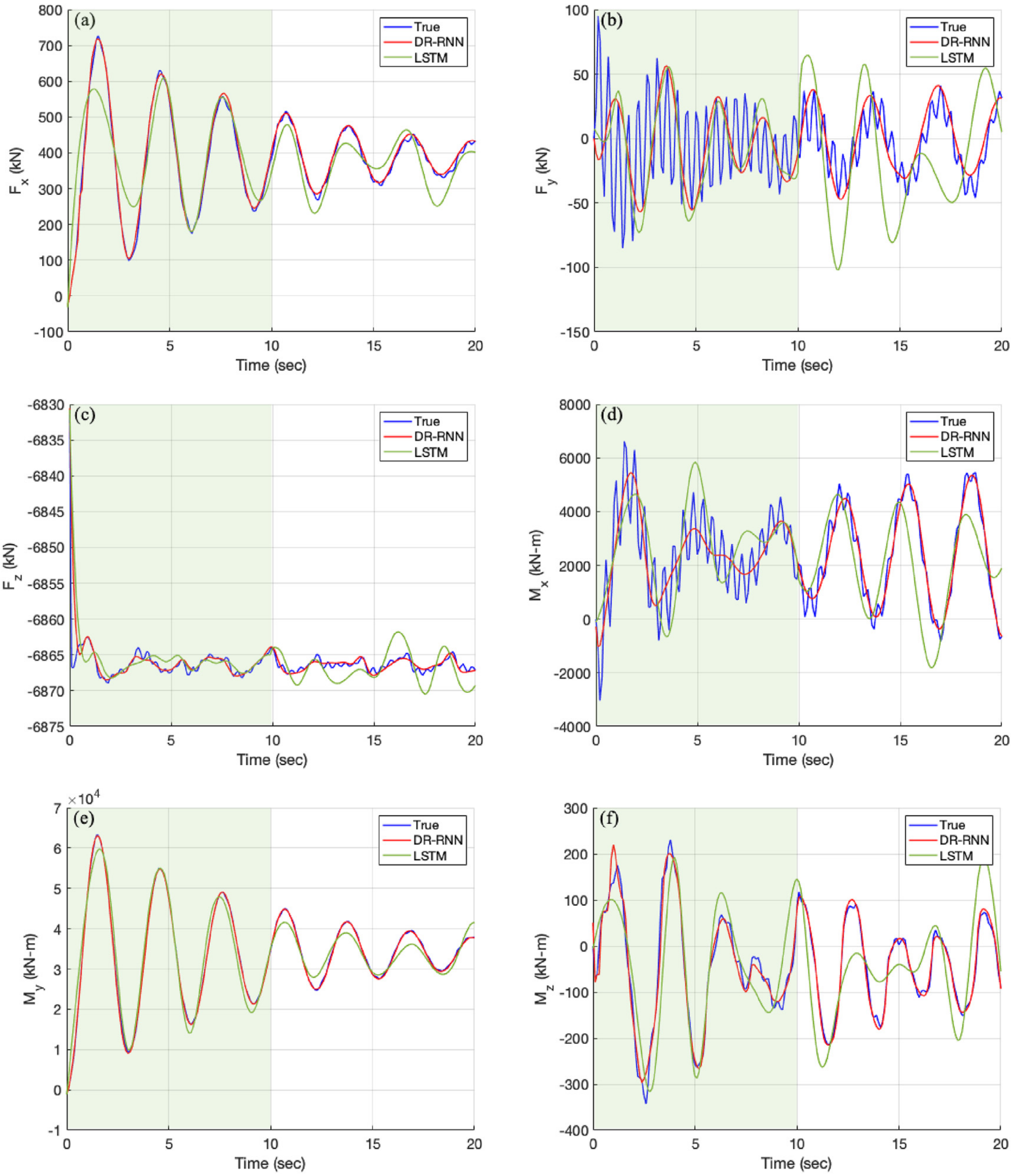


**Fig. 11.** Comparison of response predictions between LSTM model and DR-RNN model for forces and moments at the tower base section: (a) force along x-axis, (b) force along y-axis, (c) force along z-axis, (d) moment about x-axis, (e) moment about y-axis, (f) moment about z-axis. Response in the shaded area is corresponding to the selected time duration in the original trained model.

small training dataset which can be attributed to encoding structured information into a learning algorithm resulting in amplifying the information content of the data that the model sees [53,54]. When predicting the structural responses out of the original training duration, the LSTM model performance becomes much worse, and even cannot capture the general trend in some responses. This phenomenon is typically observed in purely data-driven sequence models as the curve-fitting is only performed on the training data with a specified duration, and only this duration is focused. Therefore, integrating structural dynamic constraints into the underlying deep learning model can not only improve the response prediction accuracy in the original training duration, but also allow the model to be extended to make response predictions for a longer duration with high accuracy.

## 5. Conclusions

Lifetime damage assessment of wind turbines usually requires a large number of numerical simulations, and regular purely data-driven models also need a significant amount of training data for better prediction accuracy. To reduce the computational cost and improve the prediction accuracy, a hybrid method integrating the physical constraints of the underlying structural system into the data-driven model is implemented in the present study as a computationally efficient learning model for structural response predictions. The physical constraints are selected as the linearized structural state-space equations to facilitate the learning process. The performance of the physics-informed deep learning model is firstly evaluated on a simple frame structure. The results indicate that this model has a good prediction accuracy and is very robust in structural response predictions regardless of whether the case is in the original training space or not. Meanwhile, a larger time step relative to that used in the direct numerical integration scheme could also be adopted in the physics-informed deep learning model without sacrificing the prediction accuracy. With the linearization of the aeroelastic equations of motion of the wind turbine in the state-space form, the physics-informed deep learning model is applied to predict the force and moment responses at the tower base section. It is noteworthy that the physics-informed deep learning model adopted in the present study is applied to a strong non-linear structural systems with both different wind input conditions and control parameters considered. A good match between the predicted responses and the true responses could be observed. Meanwhile, the physics-informed deep learning model also demonstrates its strong capability in long-term response prediction even if it is trained in a short duration. Comparing with the purely data-driven LSTM model, the advantage of integrating structural dynamic constraints into the deep learning model is conspicuous. However, the trained physics-inform deep learning model does not perform well on the high-frequency oscillating response prediction and can only capture the general trend of the response. One potential improvement is to linearize the wind turbine at multiple operation points and use all of these dynamic state-space equations in the physics-informed deep learning model. Therefore, a method that can combine all linearized system state-space equations at different wind turbine operation points into the physics-informed deep learning model and automatically determine the most suitable state-space equation to be used at each time step is deserved to be developed in future studies.

## CRediT authorship contribution statement

**Xuan Li:** Conceptualization, Methodology, Data curation, Writing – original draft, Visualization, Investigation, Software, Validation. **Wei Zhang:** Conceptualization, Supervision, Methodology, Writing – review & editing, Resources.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] X. Li, W. Zhang, Long-term assessment of a floating offshore wind turbine under environmental conditions with multivariate dependence structures, Renew. Energy 147 (2020a) 764–775.
[2] X. Li, W. Zhang, Long-term fatigue damage assessment for a floating offshore wind turbine under realistic environmental conditions, Renew. Energy 159 (2020b) 570–584.
[3] S. Koziel, A. Bekasiewicz, Surrogate-Based Modeling and Optimization, " *Multi-Objective Design of Antennas Using Surrogate Models*, 2017, pp. 67–100.
[4] S. Razavi, B.A. Tolson, D.H. Burn, Review of surrogate modeling in water resources, Water Resour. Res. 48 (7) (2012).
[5] H. Bazargan, M. Christie, A.H. Elsheikh, M. Ahmadi, Surrogate accelerated sampling of reservoir models with complex structures using sparse polynomial chaos expansion, Adv. Water Resour. 86 (2015) 385–399.
[6] M. Frangos, Y. Marzouk, K. Willcox, B. van Bloemen Waanders, Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems, Large-Scale Inverse Probl. Quantification Uncertain. (2010) 123–149.
[7] J.X. Wang, J.L. Wu, H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, Phys. Rev. Fluids 2 (3) (2017).
[8] V. Dubourg, B. Sudret, J.M. Bourinet, Reliability-based design optimization using kriging surrogates and subset simulation, Struct. Multidiscip. Optim. 44 (5) (2011) 673–690.
[9] Z. Hu, S. Mahadevan, A single-loop kriging surrogate modeling for time-dependent reliability analysis, J. Mech. Design Trans. ASME 138 (6) (2016).
[10] X. Li, W. Zhang, Probabilistic Fatigue Evaluation of Floating Wind Turbine Using Combination of Surrogate Model and Copula Model, " *AIAA Scitech 2019 Forum*, 2019.
[11] P. Kersaudy, B. Sudret, N. Varsier, O. Picon, J. Wiart, A new surrogate modeling technique combining Kriging and polynomial chaos expansions - application to uncertainty analysis in computational dosimetry, J. Comput. Phys. 286 (2015) 103–117.
[12] F.A. Diazdelao, S. Adhikari, Structural dynamic analysis using Gaussian process emulators, Eng. Comput. 27 (5) (2010) 580–605.
[13] E. Solak, R. Murray-Smith, W.E. Leithead, D.J. Leith, C.E. Rasmussen, Derivative observations in Gaussian process models of dynamic systems, Adv. Neural Inf. Process. Syst. (2003).
[14] A.A. Elshafey, M.R. Haddara, H. Marzouk, Damage detection in offshore structures using neural networks, Mar. Struct. 23 (1) (2010) 131–145.
[15] R. Kalra, M.C. Deo, R. Kumar, V.K. Agarwal, Artificial neural network to translate offshore satellite wave data to coastal locations, Ocean Eng. 32 (16) (2005) 1917–1932.
[16] A. Beckert, H. Wendland, Multivariate interpolation for fluid-structure-interaction problems using radial basis functions, Aero. Sci. Technol. 5 (2) (2001) 125–134.
[17] J. Zhang, S. Chowdhury, A. Messac, L. Castillo, J. Lebron, Response surface based cost model for onshore wind farms using extended radial basis functions, in: Proceedings of the ASME Design Engineering Technical Conference, 1, PARTS A AND B), 2010, pp. 729–744.
[18] J. Mahjoobi, E. Adeli Mosabbeb, Prediction of significant wave height using regressive support vector machines, Ocean Eng. 36 (5) (2009) 339–347.
[19] P.C. Young, M. Ratto, Statistical emulation of large linear dynamic models, Technometrics 53 (1) (2011) 29–43.
[20] A.R. Kambekar, M.C. Deo, Wave prediction using genetic programming and model trees, J. Coast Res. 279 (2012) 43–50.
[21] A. Nazari, P. Rajeev, J.G. Sanjayan, Modelling of upheaval buckling of offshore pipeline buried in clay soil using genetic programming, Eng. Struct. 101 (2015) 306–317.
[22] M. Afenyo, F. Khan, B. Veitch, M. Yang, Arctic shipping accident scenario analysis using Bayesian Network approach, Ocean Eng. 133 (2017) 224–230.
[23] F. Dondelinger, S. Lèbre, D. Husmeier, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, Mach. Learn. 90 (2) (2013) 191–230.
[24] K.C. Hall, J.P. Thomas, E.H. Dowell, Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows, AIAA J. 38 (10) (2000) 1853–1862.
[25] G. Kerschen, J.C. Golinval, A.F. Vakakis, L.A. Bergman, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, Nonlinear Dynam. 41 (1–3) (2005) 147–169.
[26] T. Ando, E. Chow, Y. Saad, J. Skolnick, Krylov subspace methods for computing hydrodynamic interactions in Brownian dynamics simulations, J. Chem. Phys. 137 (6) (2012).
[27] Z. Bai, Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems, Appl. Numer. Math. 43 (1–2) (2002) 9–44.
[28] Y. Efendiev, J. Galvis, F. Thomines, A systematic coarse-scale model reduction technique for parameter-dependent flows in highly heterogeneous media and its applications, Multiscale Model. Simul. 10 (4) (2012) 1317–1343.
[29] C. Lieberman, K. Willcox, O. Ghattas, Parameter and state model reduction for large-scale statistical inverse problems, SIAM J. Sci. Comput. 32 (5) (2010) 2523–2542.
[30] M. Ghommem, V.M. Calo, Y. Efendiev, Mode decomposition methods for flows in high-contrast porous media. A global approach, J. Comput. Phys. 257 (PA) (2014) 400–413.
[31] P. Ladevèze, L. Chamoin, On the verification of model reduction methods based on the proper generalized decomposition, Comput. Methods Appl. Mech. Eng. 200 (23–24) (2011) 2032–2047.
[32] A. Nouy, A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations, Comput. Methods Appl. Mech. Eng. 199 (23–24) (2010) 1603–1626.
[33] K.K. Phoon, H.W. Huang, S.T. Quek, Simulation of strongly non-Gaussian processes using Karhunen-Loeve expansion, Probabilist. Eng. Mech. 20 (2) (2005) 188–198.
[34] M. Hermans, B. Schrauwen, Training and analyzing deep recurrent neural networks, Adv. Neural Inf. Process. Syst. 1 (2013) 1–9.
[35] G. Hinton, L. Deng, D. Yu, G. Dahl, A.R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[36] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate." *arXiv preprint arXiv:1409.0473*., 2014.
[37] A. Graves, A.R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc. (2013) 6645–6649.
[38] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, S. Khudanpur, Recurrent neural network based language model, Interspeech (September) (2010) 1045–1048.
[39] C.A.L. Bailer-Jones, D.J.C. MacKay, P.J. Withers, A recurrent neural network for modelling dynamical systems, Netw. Comput. Neural Syst. 9 (4) (1998) 531–547.

[40] Z. Wang, D. Xiao, F. Fang, R. Govindan, C.C. Pain, Y. Guo, Model identification of reduced order fluid dynamics systems using deep learning, Int. J. Numer. Methods Fluid. 86 (4) (2018) 255–268.

[41] H.G. Zimmermann, C. Tietz, R. Grothmann, "Forecasting with Recurrent Neural Networks: 12 Tricks." *Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics)*, 7700, LECTU, 2012, pp. 687–707.

[42] J.N. Kani, A.H. Elsheikh, DR-RNN: a deep residual recurrent neural network for model reduction, arXiv preprint arXiv:1709.00939 (2017).

[43] J. Nagoor Kani, A.H. Elsheikh, Reduced-order modeling of subsurface multiphase flow models using deep residual recurrent neural networks, Transport Porous Media 126 (3) (2019) 713–741.

[44] H. Xiao, J.L. Wu, J.X. Wang, R. Sun, C.J. Roy, Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: a data-driven, physics-informed Bayesian approach, J. Comput. Phys. 324 (2016) 115–136.

[45] R. Hecht-Nielsen, Theory of the Backpropagation Neural Network, 1989, pp. 593–605.

[46] R. Pascanu, M. Tomas, B. Yoshua, On the difficulty of training recurrent neural networks, J. Mach. Learn. Res. 8 (2) (2002) 157–166.

[47] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[48] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.

[49] A.H. Nayfeh, P.F. Pai, Linear and Nonlinear Structural Mechanics, John Wiley & Sons, 2008.

[50] T. Tieleman, G.E. Hinton, N. Srivastava, K. Swersky, Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude, COURSERA: Neural Networks Mach. Learn. 4 (2012) 26–31.

[51] A. Géron, Hands-on machine learning with Scikit-learn, Keras, and TensorFlow concepts, tools, and techniques to build intelligent systems, O'Reilly (2019).

[52] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015.

[53] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[54] Z. Mao, A.D. Jagtap, G.E. Karniadakis, Physics-informed neural networks for high-speed flows, Comput. Methods Appl. Mech. Eng. 360 (2020) 112789.

[55] A. Agga, A. Abbou, M. Labbadi, Y. El Houm, Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models, Renew. Energy 177 (2021) 101–112.

[56] H. Chen, H. Liu, X. Chu, Q. Liu, D. Xue, Anomaly detection and critical SCADA parameters identification for wind turbines based on LSTM-AE neural network, Renew. Energy 172 (2021) 829–840.

[57] D.E. Choe, H.C. Kim, M.H. Kim, Sequence-based modeling of deep learning with LSTM and GRU networks for structural damage detection of floating offshore wind turbine blades, Renew. Energy 174 (2021) 218–235.