

Different Artificial Neural-Network Architectures for Riser VIV Analysis

Maokun Ye* and Decheng Wan*†

Computational Marine Hydrodynamics Lab (CMHL), School of Naval Architecture, Ocean and Civil Engineering
Shanghai Jiao Tong University, Shanghai, China

Haichao Liu

Zhongnan Engineering Corporation, Ltd.
Changsha, China

Bowei Song

Ningbo Pilot Station, Ningbo Dagang Pilotage Co., Ltd.
Ningbo, China

Vortex-Induced Vibration (VIV) is a nonlinear fluid-structure interaction observed when a marine riser or pipeline is placed into sea currents. In the present study, we adopt two representative artificial neural network (ANN) architectures in machine learning, i.e., multi-layer perceptron and long short-term memory, to predict riser VIV. The training data is generated by a three-dimensional computational fluid dynamics simulation performed by using the finite-analytic Navier-Stokes code. Flow field quantities at selected spatial points and the response of the riser are recorded and the flow quantities fed into the ANNs to infer riser VIV. The performances of both ANN architectures are then evaluated and compared.

INTRODUCTION

Machine learning, especially deep learning (LeCun et al., 2015) that takes advantage of artificial neural networks (ANNs), has proved itself a powerful and effective approach in natural language processing (NLP) (Collobert et al., 2011), image processing (Sonka et al., 2013), object detection (Zhao et al., 2019), recommender systems (Cheng et al., 2016), weather forecasting (Bi et al., 2023), etc. To deal with modeling tasks with distinct intrinsic characteristics, different neural network (NN) structures have been designed to more effectively learn the different input features. In general, three fundamental types of ANNs exist, i.e., multi-layer perceptron (MLP; Gardner and Dorling, 1998), convolutional neural network (CNN; LeCun and Bengio, 1995), and recurrent neural network (RNN; Lipton et al., 2015). MLP, which features a stack of fully connected neuron layers, is the most fundamental structure of ANNs, and it can theoretically be applied to any machine-learning task. To effectively learn image or matrix-like data, a CNN was then designed in which the convolution operation was leveraged in the abstraction of input features. An RNN, on the other hand, was invented to account for the historical series of the input features, such that the information of the input series itself could be effectively extracted.

With the fast development of machine-learning methods in recent years, the capability of ANNs in ocean engineering applications has attracted increasing attention from researchers. Jin et al. (2018) designed a CNN architecture that can handle spatial-

temporal flow data and leveraged it to establish the mapping relationship between the pressure fluctuations on a cylinder and the velocity field around the cylinder. Fukami et al. (2019) leveraged the capability of a CNN in image processing to reconstruct high-fidelity flow fields from a low-resolution two-dimensional flow around a fixed cylinder, and it was shown that both laminar and turbulent flow can be accurately reconstructed. D'Agostino et al. (2022) assessed three different types of RNN structures in the time-series predictions of ship maneuvering and achieved a 20-second ahead prediction. Shi et al. (2023) leveraged the long short-term memory (LSTM) type of an RNN to predict the platform motions from mooring force, wave elevation, and the history of platform motions, and they demonstrated that the prediction accuracy from multi-variable inputs is more favorable than it is from single-variable input.

The objective of the current work is to perform a comparative study of the different ANN architectures in the predictions of riser VIV to provide a better idea of the performance of those structures. The aim of ANN implementation is to predict riser displacement at a certain time instance by using the information of the flow field quantities or the time history of the riser motion itself. This is not only of research interest but also of importance in engineering practice. The present work will contribute to a better comprehension of the performance of different ANN architectures in the prediction of riser VIV. The remainder of this paper is organized as follows. First, the methods used in the current study, including the computational fluid dynamics (CFD) approach adopted to generate the original VIV data and the ANN structures to be implemented in the machine-learning task, are introduced. Second, the generation of the VIV data of a marine riser using CFD is described in detail. Afterward, different ANNs are implemented to learn and predict the VIV motions of the marine riser. Further, the performances of different training strategies are compared and discussed. Last, conclusions are drawn from the results and discussions.

* ISOPE Member; † Corresponding author.

Received October 2, 2024; updated and further revised manuscript received by the editors December 5, 2024. The original version (prior to the final updated and revised manuscript) was presented at the Thirty-fourth International Ocean and Polar Engineering Conference (ISOPE-2024), Rhodes, Greece, June 16–21, 2024.

KEY WORDS: VIV, machine learning, neural networks, time series prediction.

METHODS

CFD Approach

Riser Motion Solver. A marine riser can be modeled as a top-tensioned beam with streamwise and crossflow motions (Huang et al., 2011; Ye and Chen, 2019). The partial differential equations which govern the lateral motions are as follows:

$$T \frac{d^2 y}{dx^2} + \frac{dy}{dx} \frac{dT}{dx} - \frac{d^2}{dx^2} \left(EI \frac{d^2 y}{dx^2} \right) + f_y = m \ddot{y} + D_S \dot{y} \quad (1)$$

$$T \frac{d^2 z}{dx^2} + \frac{dz}{dx} \frac{dT}{dx} - \frac{d^2}{dx^2} \left(EI \frac{d^2 z}{dx^2} \right) + f_z = m \ddot{z} + D_S \dot{z} \quad (2)$$

where T is the effective tension; x represents the pipe axial direction, which is vertical to the ground; y and z denote the in-line and cross-flow displacements of a riser, respectively; E and I are Young's modulus and the area moment of inertia of the riser; m and D_S denote the mass per unit length and damping coefficient of the riser, respectively; and f_y and f_z are the forces in corresponding directions.

Flow Solver. The CFD simulations in the current study are performed by using the finite-analytic Navier-Stokes code (FANS). The code solves the unsteady incompressible Navier-Stokes equations by using the finite-analytic method (Chen and Patel, 1989; Chen et al., 1990). FANS code solves the mean flow and turbulence quantities in arbitrary combinations of embedded, overset, or matched grids. In the simulations, the computational domain is first decomposed into smaller blocks to efficiently deal with calculations that involve complex configurations and flow conditions. Within each computational block, the finite-analytic numerical method is leveraged to solve the governing equations on a general curvilinear, body-fitted coordinate system. The pressure-velocity coupling is achieved through a hybrid PISO/SIMPLER algorithm developed by Pontaza et al. (2005). The code is well-validated and has been successfully applied to the investigations of complex flow problems (Huang and Chen, 2021; Chen and Chen, 2023; Ye et al., 2023).

ANN Structures

Multi-Layer Perceptron (MLP). An MLP network consists of an input layer, one or more hidden layers, and an output layer. The input layer receives the input data, which is then fed forward through the hidden layers to the output layer. The hidden layers perform a series of non-linear transformations on the input, allowing the network to model complex relationships between the input and output. An illustration of an MLP is shown in Fig. 1.

Figure 2 shows the details of the operations within an MLP node inside the hidden layers. Each node in the hidden layer performs a weighted sum of its inputs followed by an activation function, which activates the node by performing a nonlinear transformation as denoted by σ in Fig. 2. During training, the MLP network adjusts the weights to learn the intrinsic relations between the input and output data by using back-propagation, a technique for computing the gradient of the loss function with respect to the weights. This allows the network to learn the optimal set of weights that minimizes the error between the predicted output and the true output.

Unless otherwise specified, the activation function used in the current study is the tanh activation function, defined as

$$\sigma_{\tanh} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

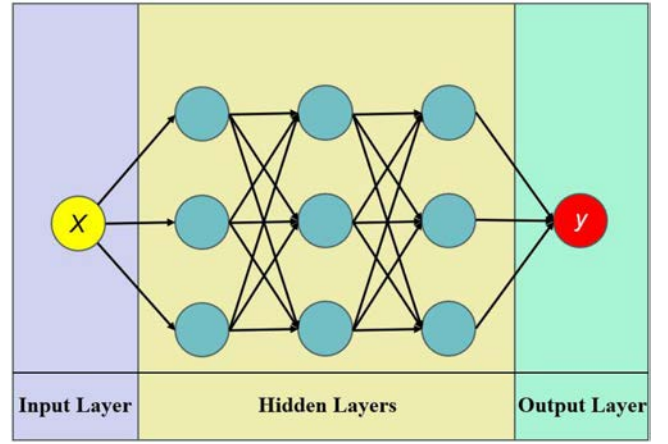


Fig. 1 A multi-layer perceptron (MLP)

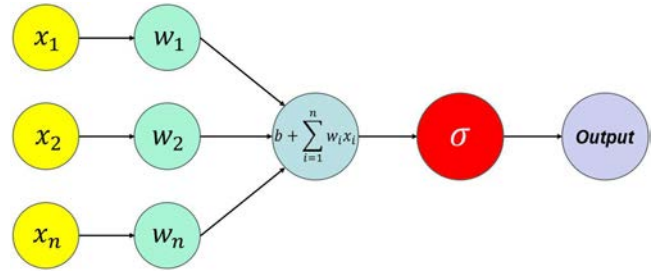


Fig. 2 An MLP node

Long Short-Term Memory (LSTM) Networks. An LSTM (Yu et al., 2019) network is a type of RNN that was specifically designed to address the issue of vanishing gradients in traditional RNNs. LSTMs are widely used in various fields of machine learning, including NLP, speech recognition, and time series analysis. The key idea behind LSTM is to introduce a memory cell that can retain information over long sequences, allowing it to capture long-term dependencies in the data. This memory cell is composed of three main components: an input gate, a forget gate, and an output gate. A schematic diagram for an LSTM network is shown in Fig. 3, and the according equations can be written as follows:

$$h_t = \sigma[\text{dot}(\text{state}_t, U_h) + \text{dot}(\text{input}_t, W_h) + \text{dot}(c_t, V_h) + b_h] \quad (4)$$

where U , V , and W are trainable coefficients, and

$$c_{t+1} = i_t * k_t + c_t * f_t \quad (5)$$

in which i_t , f_t , and k_t are calculated by

$$i_t = \sigma[\text{dot}(\text{state}_t, U_i) + \text{dot}(\text{input}_t, W_i) + b_i] \quad (6)$$

$$f_t = \sigma[\text{dot}(\text{state}_t, U_f) + \text{dot}(\text{input}_t, W_f) + b_f] \quad (7)$$

$$k_t = \sigma[\text{dot}(\text{state}_t, U_k) + \text{dot}(\text{input}_t, W_k) + b_k] \quad (8)$$

By introducing these gates, LSTM networks can selectively update and retrieve information over long sequences, making them capable of capturing both short-term and long-term dependencies in the data. The ability to remember or forget information over extended time spans makes LSTMs particularly effective in tasks involving sequential data. LSTMs have become a fundamental building block in many deep-learning models due to their ability to handle sequential information effectively.

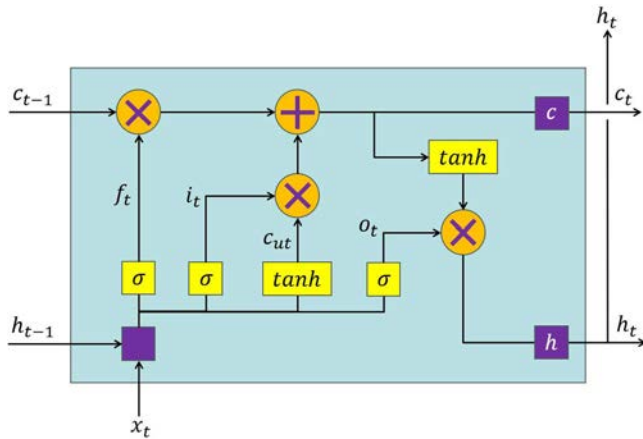


Fig. 3 An LSTM cell

VIV SIMULATION OF A MARINE RISER BY CFD

In the current study, CFD simulations are first carried out to generate the data to be used in the machine-learning process. In the simulations, the time histories of the lateral displacements, i.e., the motions in streamwise and crossflow directions, are collected along with the time histories of the flow quantities at selected spatial locations.

Test Case Description

In the simulations, a riser model with a length-to-diameter ratio (L/D) of 482.5 is used. The riser stands vertically in the fluid field with a linearly distributed tension. Both ends of the riser are pinned. The basic properties of the studied riser are listed in Table 1.

Computational Setup

A computational domain with $20D$ (riser diameter) in the streamwise direction and $10D$ in the crossflow direction is created, with the riser positioned at the center of the domain. Overset grids are then generated to discretize the computational domain. An overview and a detailed cross-section view of the grids are shown in Fig. 4. It can be seen that two overset-grid blocks are constructed including a near field block (red) and a background grid block (green). The near field block has 231,322 computational points in total with a dimension of 31 (in the axial direction) $\times 182$ (in the circumferential direction) $\times 41$ (in the radial direction), and the background block has a total of 629,331 grid points with a dimension of $31 \times 201 \times 101$. The time step size used in this paper is 0.01 dimensionless time, i.e., the flow travels $0.01D$ at every time step. To capture the unsteady vortex-shedding, the CFD simulations are performed under a Large Eddy Simulation (LES) framework. A uniform current of 0.42 m/s is specified at the inlet in the simulations.

Parameter	Unit	Value
Riser diameter	m	0.3
Riser length	m	144.45
Bending stiffness	$N \cdot m^2$	6.85E6
Top pretension	N	1.84E5
Bottom pretension	N	1.75E5

Table 1 Riser properties

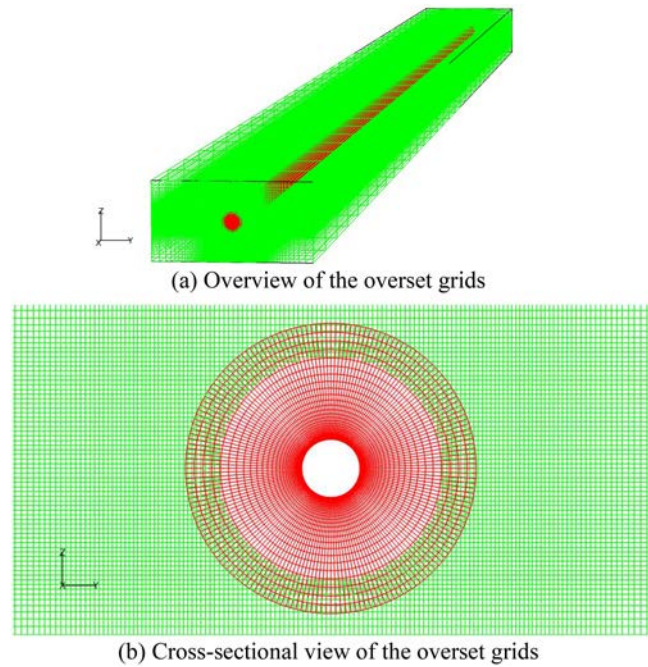


Fig. 4 Computational grids

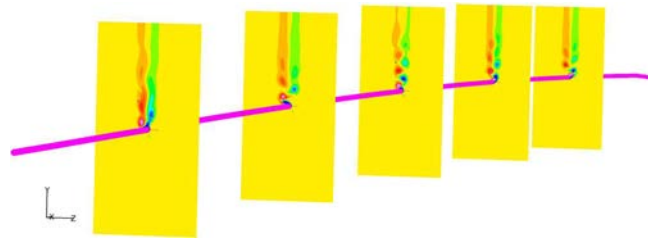


Fig. 5 Vortex shedding along vertical direction of the riser

CFD Results

The CFD results are shown briefly below to give a general idea of the flow field. It is worth emphasizing here that the CFD results in the current study only serve as the training dataset in the machine-learning processes; therefore, detailed discussions of the CFD results will be omitted for the sake of brevity. More details of the FANS simulations of riser VIV can be found in previous papers (Huang et al., 2011; Ye and Chen, 2019). The vortex shedding along the vertical direction of the riser is shown in Fig. 5.

It should be noted that although three-dimensional CFD calculations are performed, only one selected slice of the complete domain is used in the machine-learning process. By doing this, we arbitrarily assume that the lateral motions, i.e., streamwise and crossflow, can be predicted by the flow information or the lateral motions of the riser solely at the same vertical location, ignoring the effects of the information at neighboring vertical locations. This simplification is well enough for the scope of the current study in which the goal is to compare the performance of different ANN structures. The vortex generation and shedding of the riser at a selected vertical slice, i.e., $1/4$ riser length from the top end, is shown in Fig. 6.

PREDICTION OF VIV BY ANNs

As mentioned earlier, only one slice of the computational domain is used in the machine learning processes. Figure 7 illus-

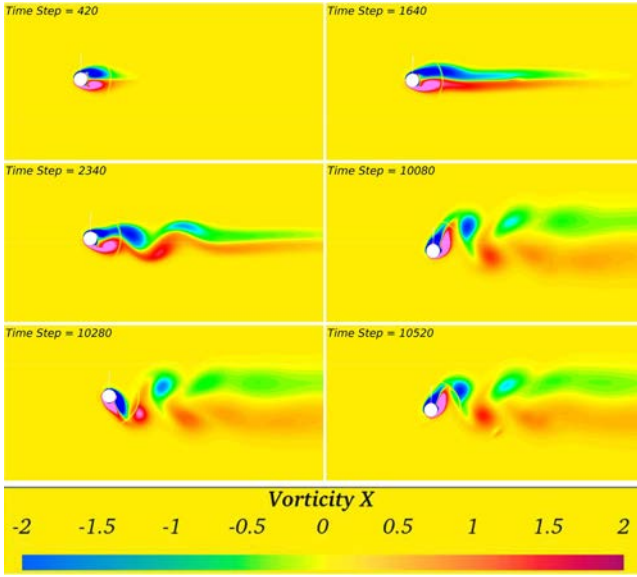


Fig. 6 Vortex shedding of the riser at a given vertical location

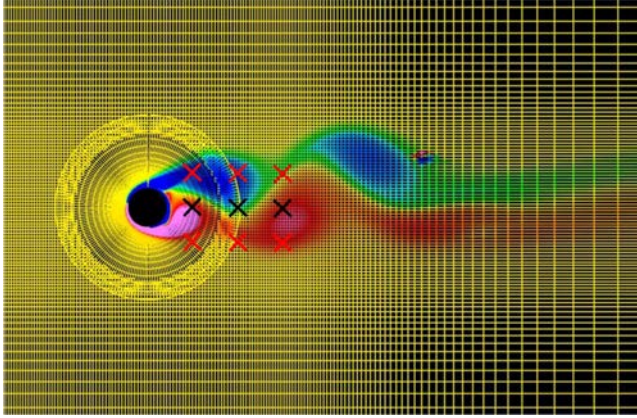


Fig. 7 Selected spatial monitoring points

trates the monitoring points at which the flow quantities, specifically x -vorticities, are recorded during the CFD simulations. Three columns of points are used. The first column of points is positioned at $1.2D$ downstream of the riser, and the points are spaced $1D$ from each other; the second and third columns are respectively $1.2D$ and $2.4D$ further downstream of the first column.

A search of optimal hyperparameters used in the training processes, e.g., the batch sizes, number of epochs, activation functions, and number of cells used in each of the hidden layers, are performed according to their performances and training efficiency, i.e., cost of each ANN structure. Then, we use hyperparameters as similar as possible for each training of the different ANNs. Unless otherwise mentioned, a three-layer structure with 128 cells in each layer is defined for the hidden layers of the ANNs in this work. The complete dataset consists of 30,000 time instances (time steps), and the ANNs are trained by utilizing the first 75% of the time series data, while the last 25% is reserved for testing purposes. The activation function is set to be “tanh” for the MLP networks and to “tanh” and “sigmoid” for the LSTM networks. The “Adam” algorithm is adopted as the optimizer with a learning rate of 0.001 to train the ANNs, and the mean-square error (MSE) is selected as the loss function to be minimized in the training processes. For each training, 500 epochs are performed.

All training is performed by using the machine-learning package, Keras, with TensorFlow as its backend.

MLP-1: Mapping from Flow Information to Riser Motions

First, the general-purpose of an ANN structure, i.e., an MLP, is adopted in the prediction of VIV motions of the riser, and the results obtained by an MLP will also serve as a benchmark for comparison. In the training of MLP networks, the temporal dependence of different time instances is omitted, meaning that each of the 30,000 time steps is an individual snapshot that can be viewed as a separate input. Under this framework, the MLP networks are used to establish a mapping relation between the x -vorticity of the nine monitoring points and the lateral displacements of the riser at the selected vertical location by using the information at one time step exclusively. This mapping relation can be described by the following equation:

$$MLP(flow^t) = riser^t \quad (9)$$

Figure 8 shows the change of the loss functions in the training process of the MLP networks for the streamwise (y) and crossflow (z) motions. It can be seen that the loss function drops quickly during the first 100 epochs but starts to show an asymptotic behavior with oscillations. This observation implies that the MLP net-

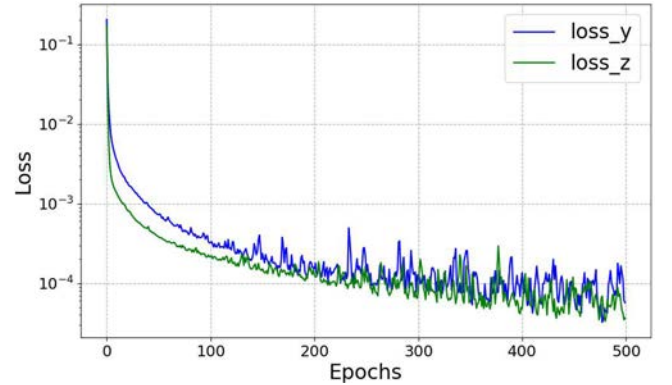


Fig. 8 Training history of MLP-1

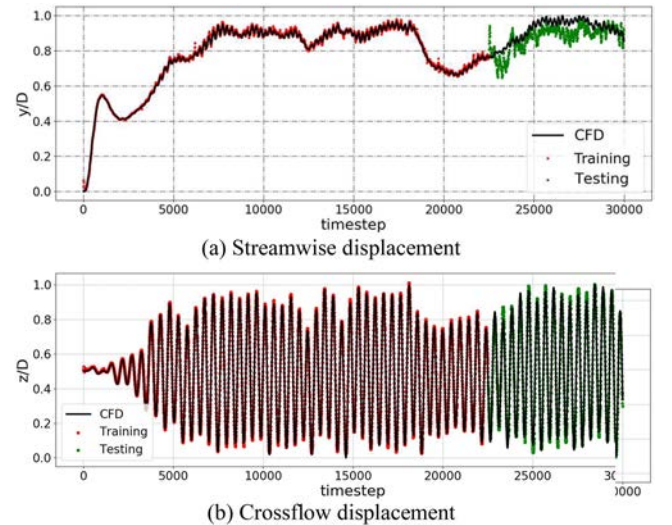


Fig. 9 Predictions of the riser VIV motion by MLP-1

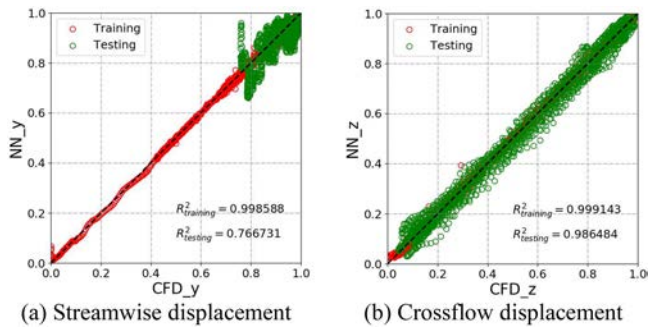


Fig. 10 Accuracy of NN-predicted (MLP-1) displacements in streamwise and crossflow directions

work has approached a potential optimal set of weights while still having some extent of difference in different training batches.

Afterward, the trained MLP networks are used to predict the streamwise and crossflow displacements of the riser, as shown respectively in Figs. 9a and 9b. The red points are the NN-predicted riser displacements on the training dataset. It can be observed that the prediction by the NNs on the training dataset matches the CFD result well in general; however, large discrepancies in the prediction at some time steps can still be identified in Fig. 9a. For prediction of the trained NNs on the testing dataset, as shown in green in the two figures, we see that the prediction of the crossflow displacement is much better than the prediction of streamwise displacement. However, this is expected since the crossflow displacement of the riser shows a mainly sinusoidal pattern with a mean value around 0, while the streamwise displacement is much more complicated. Moreover, the streamwise displacement of the riser can be seen as a superposition of small-scale vibrations and a large-scale motion of the riser, making it challenging for the NN to be effectively trained.

Despite the aforementioned difficulties and some noticeable spikes, as shown in Fig. 9a, the NN-predicted streamwise displacement of the riser is acceptably reasonable. The large-scale displacement trend as well as the small-scale vibrations have been captured successfully, although discrepancies between the prediction and the CFD data can still be identified.

Figures 10a and 10b respectively show the overall accuracy for prediction of the trained MLP networks for the streamwise and crossflow displacements of the riser. It can be seen that both of the MLP networks reach high accuracies on the training dataset with an R^2 score of nearly 1. For performance on the test dataset, the crossflow prediction maintains high accuracy while the accuracy of the predicted streamwise displacement is relatively lower.

MLP-2: Inclusion of Implicit Gradient Information

In the present study, we attempt to improve the performance and interpretability of the NNs by including more expert knowledge rather than increasing the complexity of the NNs as reported in most of the related studies.

Based on the results of MLP-1, improvements of the NNs are proposed by adding more information to either their output or input while keeping their structures unchanged.

The first attempt is to add gradient information of the label into the output. Specifically, instead of outputting the riser motion only at one instance in each sample, we simultaneously output the riser motion at three consecutive time instances. In this way, we include the riser information of two neighboring temporal points, i.e., the time derivatives of the central point are taken into account implicitly.

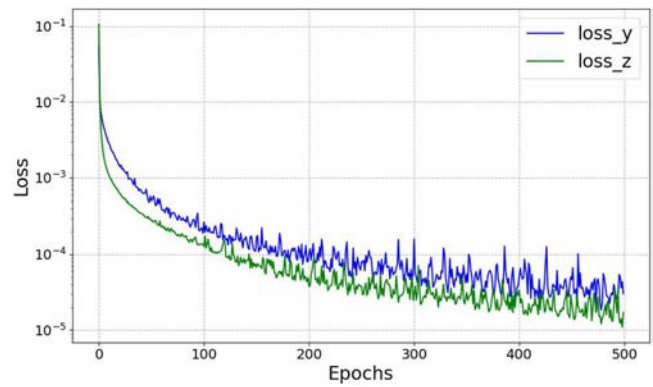


Fig. 11 Training history of MLP-2

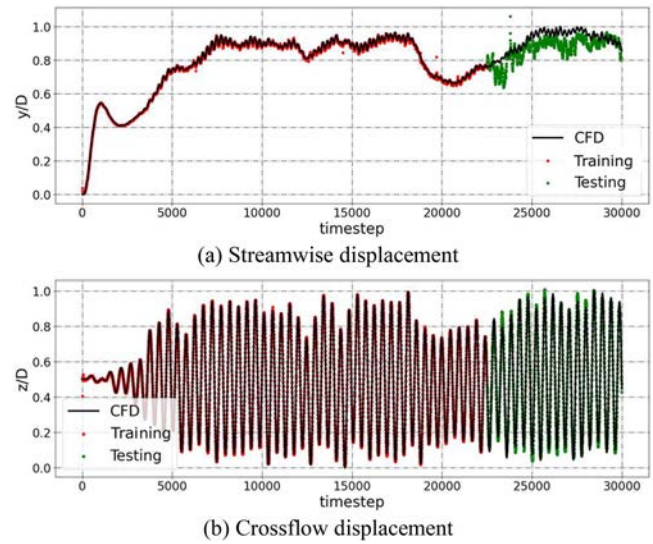


Fig. 12 Predictions of the riser VIV motion by MLP-2

For illustration purposes, the mapping relation from the input to the output of MLP-2 can be written as Eq. 10:

$$MLP(flow^t) = riser^{t-1:t:t+1} \quad (10)$$

and the training history of MLP-2 is shown in Fig. 11. It can be observed that the losses for both y and z displacements are lower than those in MLP-1. This may imply that a better mapping relation was found by the NNs after adding the gradient information to the output.

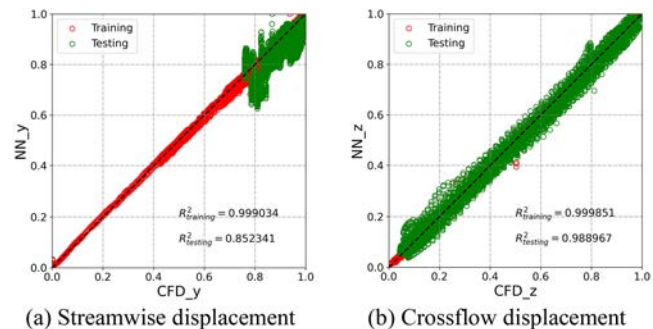


Fig. 13 Accuracy of NN-predicted (MLP-2) displacements in streamwise and crossflow directions

After the training is completed, the obtained NN is used to predict the riser motion by feeding into required inputs. Comparing with MLP-1, it can be seen that fewer spikes in the streamwise displacement are found, and the overall prediction accuracy of the streamwise displacement is significantly raised from 0.767 to 0.852. However, the improvement is not obvious in general, and prediction in the testing dataset is challenging.

Figure 13 shows the overall accuracy for prediction of the trained MLP networks. A significant improvement in the prediction of streamwise displacement is observed. Owing to the inherent periodicity of crossflow motion, which enhances its predictability, MLP-2 attains prediction accuracy commensurate with MLP-1.

MLP-3: Further Inclusion of Explicit Time Labels

Next, based on the results of MLP-2, we attempt to further improve the performance of the NNs by including the time labels in the input features. Specifically, for each of the samples, the mapping relation can now be written as the following:

$$MLP(flow^t, t) = riser^{t-1:t+1} \quad (11)$$

The training histories of MLP-3 are shown in Fig. 14. It can be seen clearly that the training of the crossflow displacement is able to reach a significantly lower loss level. This is understandable due to the strong periodic behavior in the crossflow vibration of the riser.

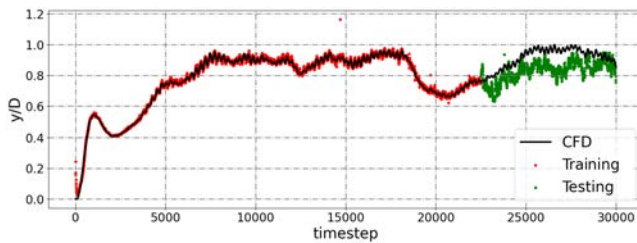
Figures 15 and 16 show the NN-predicted riser motion and the prediction accuracy, respectively. In general, a similar performance is found compared with MLP-2.

LSTM-1: A Pure Time-series Analysis

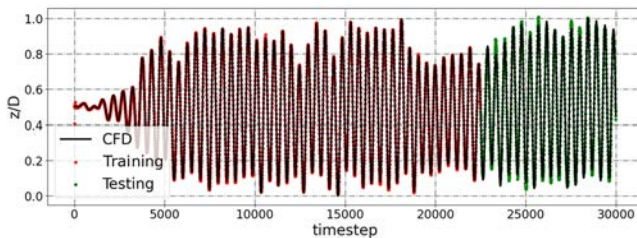
Instead of treating the current task as a mapping from flow-field variables to riser motion, the same task can also be treated as a time-series problem. Specifically, we predict the riser motion by using the time history of the motion.

First, in LSTM-1, we apply a pure time-series analysis by using the information exclusively from the motion history of the riser itself without any information from the flow field, which can be written as

$$LSTM(riser^{history}) = riser^t \quad (12)$$



(a) Streamwise displacement



(b) Crossflow displacement

Fig. 14 Predictions of the riser VIV motion by MLP-3

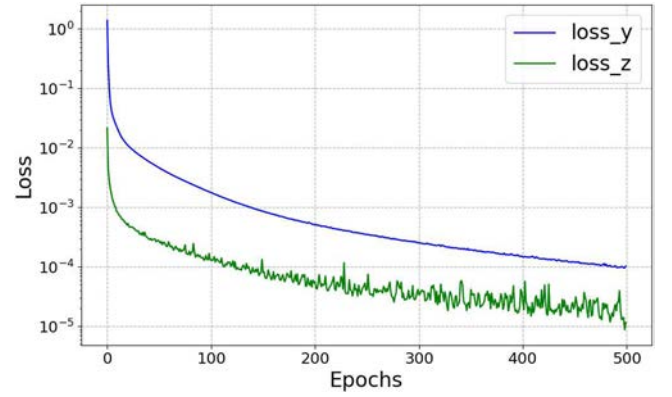


Fig. 15 Training history of MLP-3

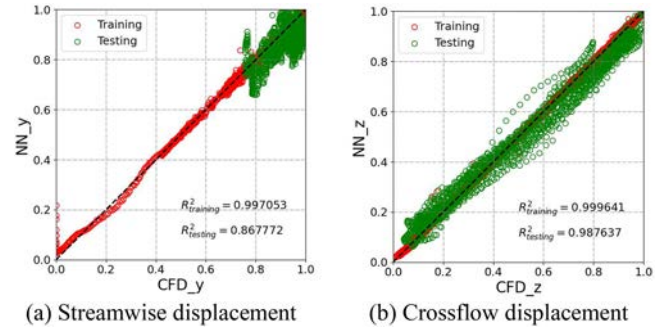


Fig. 16 Accuracy of NN-predicted (MLP-3) displacements in streamwise and crossflow directions

More specifically, the time sequence of the riser displacements within the time steps of 1 to 50 is fed into the LSTM network to infer the riser displacements at the 51st time step. For the sake of brevity, the above description is referred to as 1–50 to predict 51; 2–51 to predict 52; 3–52 to predict 53, etc. We expect that this strategy of training the NN to predict riser VIV motions would be extremely challenging, if not impossible, due to the fact that the VIV motion of the riser is a highly nonlinear, fluid-structure interaction (FSI) problem, and the determination of the riser displacements acquires information from the flow field as well.

To prepare the training dataset, the first 75% of the time histories of the streamwise and crossflow displacements are broken down into multiple sub-sequences, each consisting of (a memory of) 50 consecutive time steps, and the target of prediction is the next coming time step. Although not presented here in detail, a hyper-parameter search of the memory size is performed, and the final memory size of 50 is a compromise between accuracy and training cost. The training history of LSTM-1 is shown in Fig. 17.

The trained LSTM-1 networks are then used to predict the streamwise and crossflow displacements of the riser, as shown in Figs. 18a and 18b, respectively. Different from the previous NNs, in the current time series prediction, the predictions on the testing dataset are generated by the LSTM network one time step by one time step. For example, we use the first 22,500 time steps to form the training dataset, the 22,501st time step is generated by using time steps from 22,450 to 22,500. Then, the NN-predicted 22,501st time step is used to generate the 22,502nd time step; the NN-predicted 22,502nd time step is then added to the time series to generate the 22,503rd time step, and so on.

As can be observed in Figs. 18a and 18b, in terms of the training dataset, either prediction of the streamwise or the cross-

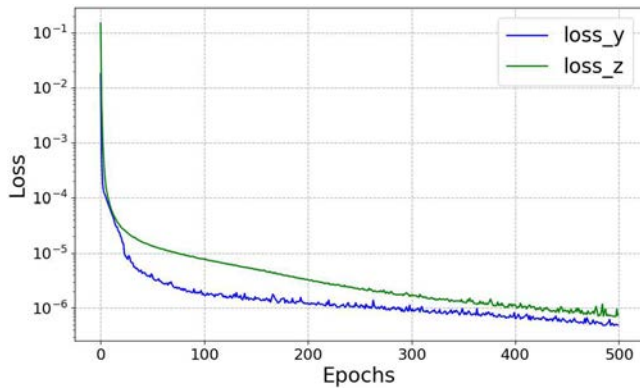


Fig. 17 Training history of LSTM-1

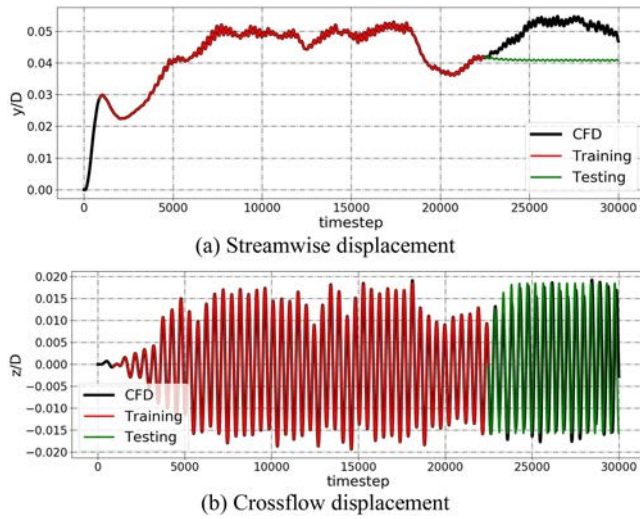


Fig. 18 Predictions of the riser VIV motion by LSTM-1

flow displacement matches the CFD data well. However, for the testing data range, obvious discrepancies are observed, especially for the streamwise prediction. As mentioned earlier, the streamwise motion of the riser at a given vertical location can be seen as a superposition of large-scale motions and small-scale vibrations. It seems that from Fig. 18a, only the small-scale vibrations of the riser in the streamwise direction are captured (partially) while the large-scale motion of the riser is missing entirely. For the crossflow prediction, a much better performance of the LSTM network can be seen. However, closer inspection reveals that the amplitudes of the crossflow vibration are not, or at least not obviously, changing with time. This argument is more prominent when compared to the previous NN predictions, as shown in Fig. 9a, Fig. 12a, and Fig. 15a.

Furthermore, a phase difference can be identified in the predicted crossflow displacement. As shown in Fig. 19a, the green circles are the predicted crossflow displacement with lighter colors representing later time steps. It can be seen that the prediction accuracy is high at the beginning and decreases over time.

LSTM-2: Mapping from Flow Information to Riser Motion by Using Flow Histories

Another LSTM network specifically designed for time-series predictions is implemented for the same task to account for the sequential information, i.e., memory, of the flow field. Specifically, by not only using the data to train the networks at the current time step exclusively, we also use the 50 consecutive time

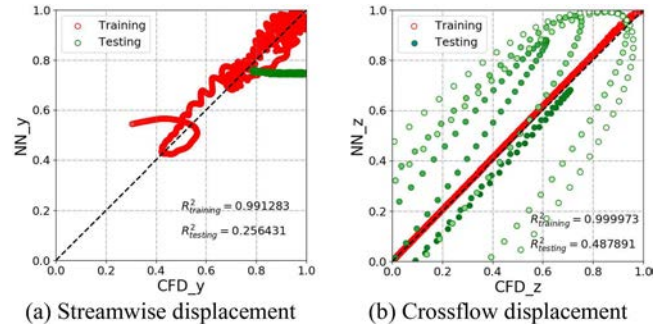


Fig. 19 Accuracy of NN-predicted (LSTM-1) displacements in streamwise and crossflow directions

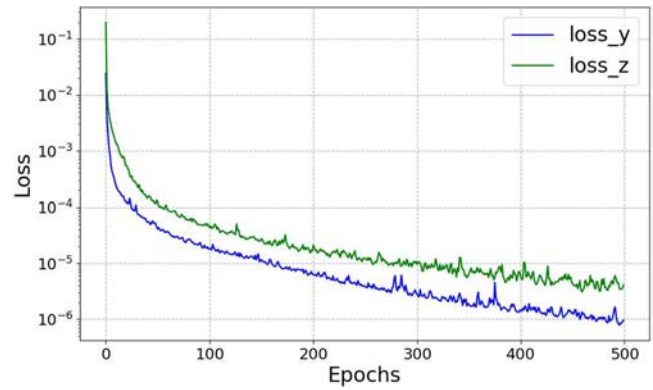


Fig. 20 Training history of LSTM-2

steps of the flow data at the nine monitoring points to infer the lateral displacements of the riser at the 50th time step. That is to say, besides the flow data at the current time step, the memories, i.e., the time histories, of those spatial points are also leveraged in the training processes.

$$LSTM(flow^{history}) = riser^t \quad (13)$$

Figure 20 shows the histories of the loss functions in the training of the NNs. It can be observed that the loss functions reached a lower level and remained smoother compared to those of the MLP networks.

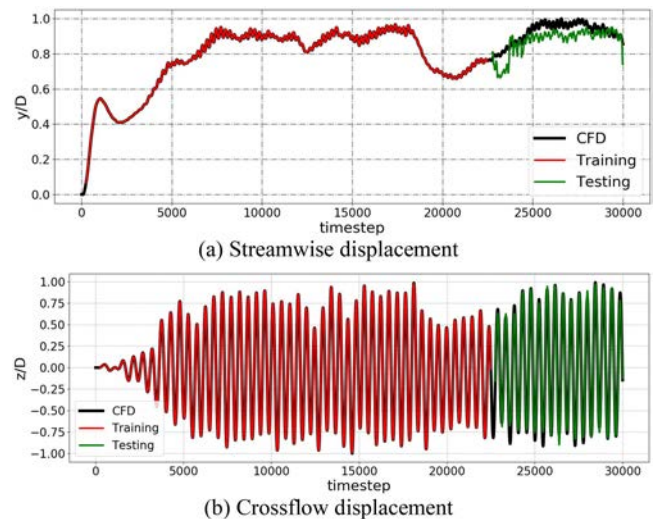


Fig. 21 Predictions of the riser VIV motion by LSTM-2

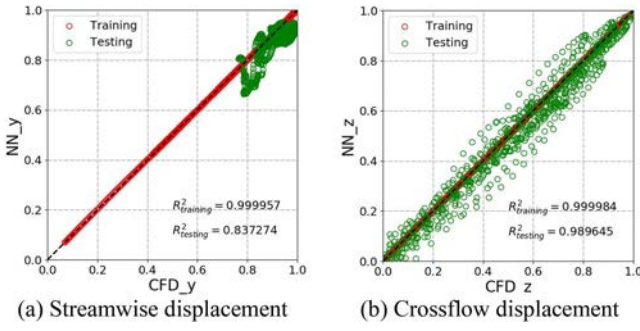


Fig. 22 Accuracy of NN-predicted (LSTM-2) displacements in streamwise and crossflow directions

The trained LSTM networks are then used to predict the streamwise and crossflow displacements of the riser, as shown in Fig. 21a and Fig. 21b, respectively. Again, the overall performance of the network trained for the crossflow motion is better than one trained for streamwise motion. Furthermore, by comparing Fig. 9a with Fig. 21a, obvious improvements in the prediction are identified: 1) there are no abrupt spikes in Fig. 21a as presented in Fig. 9a; 2) the overall amplitude of the small-scale oscillations predicted by the LSTM network is smaller than that predicted by the MLP network, which matches the original CFD data better. For the crossflow displacement, again, the network reproduces the vibration well, as discussed earlier.

Figures 22a and 22b show the overall prediction accuracy of the trained LSTM networks for the streamwise and crossflow displacements of the riser, respectively. Both LSTM networks reached high accuracies on the training dataset with an R^2 score of almost 1. For the performance on the test datasets, the accuracy of the streamwise displacement is relatively lower than that of the crossflow, but a non-negligible improvement of the R^2 score is achieved compared with the one obtained from the MLP network, i.e., from 0.767 to 0.837.

COMPARISON OF NEURAL NETWORK STRUCTURES

Figure 23 shows the comparison of R^2 scores of different NN predictions. It can be seen that for all five NNs, the predictions of crossflow motion of the riser reaches higher accuracies than

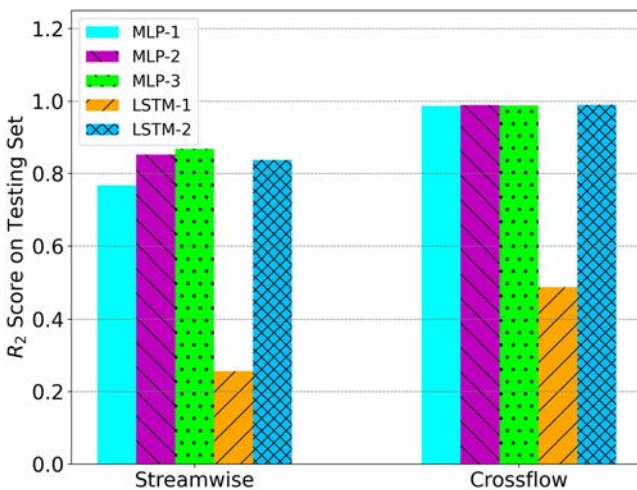


Fig. 23 Comparison of R^2 scores of different NN predictions on the testing dataset

the predictions of the streamwise motion using the same type of NN. It can also be observed that MLP-3 reaches the highest prediction accuracy in the streamwise displacement by including both the label gradients and time labels. In addition, the exclusion of flow field information, labeled as LSTM-1, negatively impacts the accuracy of the prediction, demonstrating that the prediction of riser VIV can hardly be achieved by performing a pure time series prediction of the riser motion. This is expected due to the physical sophistication of riser VIV, as discussed earlier.

CONCLUSIONS

In this study, a total of five different NN architectures or training strategies have been applied to the predictions of riser VIV motions, i.e., the streamwise and the crossflow displacements. The original VIV data used in the machine-learning training processes have been generated by CFD simulations using overset grids under an LES framework. The time histories of the x -vorticity at selected points in the flow field as well as the time histories of the riser lateral displacements have been recorded for the following training processes. The five different training strategies have then been adopted to train the NNs to predict the riser VIV motions at the current time step as follows:

- MLP-1: to infer the riser motions by using exclusively the flow field information at the current time instance
- MLP-2: to add implicit gradient information to MLP-1 by simultaneously outputting riser displacements at three consecutive time instances
- MLP-3: to add explicit time labels to the input of MLP-2
- LSTM-1: to infer the riser motions by using exclusively the time sequences of the riser displacements
- LSTM-2: to infer the riser motions by using time histories of flow information

According to the results, the following conclusions can be drawn:

- Predictions of the crossflow displacement of the riser generally reach higher accuracy than the streamwise displacement.
- Inclusion of the label gradients, i.e., MLP-2 and MLP-3, significantly improves prediction performance of the NNs.
- Predictions of riser VIV using NNs can barely be, if at all, achieved by performing a pure time series prediction of the riser motion itself due to the intrinsic nature of riser VIV problems.

It is also important to note that while statistical techniques could be used to improve the accuracy of streamwise displacement predictions, such as ignoring the initial stage or manually amplifying vibration amplitudes to simplify neural network training, we found that these adjustments cannot be made without first analyzing the data curves. Therefore, to provide a clearer and more straightforward evaluation of the capabilities of different NN architectures and training strategies, we chose not to employ such statistical tricks in this study. Further studies are also recommended to account for the three-dimensional flow information and to integrate flow and riser time sequences, i.e., to integrate spatial and temporal information of the flow field and the riser, in the training of NNs.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (52131102) and the Research and Application Demonstration Project of Key Technologies for Safeguarding of Container vessels in Ningbo Zhoushan Port Based on Intelligent Navigation (Grant No. ZJHG-FW-2024-27), to which the authors are most grateful.

REFERENCES

- Bi, K, et al. (2023). "Accurate Medium-Range Global Weather Forecasting with 3D Neural Networks," *Nature*, 619(7970), 533–538. <https://doi.org/10.1038/s41586-023-06185-3>.
- Chen, HC, and Chen, CR (2023). "CFD Simulation of a Container Ship in Random Waves Using a Coupled Level-Set and Volume of Fluid Method," *J Hydrodyn*, 35(2), 222–231. <https://doi.org/10.1007/s42241-023-0019-x>.
- Chen, HC, and Patel, VC (1989). "The Flow Around Wing-body Junctions," *Proc 4th Symp Numer Phys Aspects Aerodyn Flows*, Long Beach, CA, USA.
- Chen, HC, Patel, VC, and Ju, S (1990). "Solutions of Reynolds-averaged Navier-Stokes Equations for Three-Dimensional Incompressible Flows," *J Comput Phys*, 88(2), 305–336. [https://doi.org/10.1016/0021-9991\(90\)90182-Z](https://doi.org/10.1016/0021-9991(90)90182-Z).
- Cheng, HT, et al. (2016). "Wide and Deep Learning for Recommender Systems," *Proc 1st Workshop Deep Learn Recommender Syst*, Boston, MA, USA, 7–10. <https://doi.org/10.1145/2988450.2988454>.
- Collobert, R, et al. (2011). "Natural Language Processing (Almost) from Scratch," *J Mach Learn Res*, 12, 2493–2537. <https://doi.org/10.48550/arXiv.1103.0398>.
- D'Agostino, D, Serani, A, Stern, F, and Diez, M (2022). "Time-series Forecasting for Ships Maneuvering in Waves via Recurrent-type Neural Networks," *J Ocean Eng Mar Energy*, 8(4), 479–487. <https://doi.org/10.1007/s40722-022-00255-w>.
- Fukami, K, Fukagata, K, and Taira, K (2019). "Super-resolution Reconstruction of Turbulent Flows with Machine Learning," *J Fluid Mech*, 870, 106–120. <https://doi.org/10.1017/jfm.2019.238>.
- Gardner, MW, and Dorling, SR (1998). "Artificial Neural Networks (The Multilayer Perceptron) - A Review of Applications in the Atmospheric Sciences," *Atmos Environ*, 32(14–15), 2627–2636. [https://doi.org/10.1016/S1352-2310\(97\)00447-0](https://doi.org/10.1016/S1352-2310(97)00447-0).
- Huang, H, and Chen, H-C (2021). "Coupled CFD-FEM Simulation for The Wave-induced Motion of a CALM Buoy with Waves Modeled by a Level-Set Approach," *Appl Ocean Res*, 110, 102584. <https://doi.org/10.1016/j.apor.2021.102584>.
- Huang, K, Chen, H-C, and Chen, C-R (2011). "Numerical Scheme for Riser Motion Calculation During 3-D VIV Simulation," *J Fluid Struct*, 27(7), 947–961. <https://doi.org/10.1016/j.jfluidstructs.2011.06.010>.
- Jin, X, Cheng, P, Chen, WL, and Li, H (2018). "Prediction Model of Velocity Field Around Circular Cylinder over Various Reynolds Numbers by Fusion Convolutional Neural Networks Based on Pressure on the Cylinder," *Phys Fluids*, 30(4). <https://doi.org/10.1063/1.5024595>.
- LeCun, Y, and Bengio, Y (1995). "Convolutional Networks for Images, Speech, and Time Series," in: *The Handbook of Brain Theory and Neural Networks*, MA Arbib (Ed), 3361(10), 1995. <https://dl.acm.org/doi/10.5555/303568.303704>.
- LeCun, Y, Bengio, Y, and Hinton, G (2015). "Deep Learning," *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>.
- Lipton, ZC, Berkowitz, J, and Elkan, C (2015). "A Critical Review of Recurrent Neural Networks for Sequence Learning," Arxiv Preprint. <https://doi.org/10.48550/arxiv.1506.00019>.
- Pontaza JP, Chen, HC, and Reddy, JN (2005). "A Local Analytic-based Discretization Procedure for the Numerical Solution of Incompressible Flows," *Int J Numer Methods Fluids*, 49(6), 657–699. <https://doi.org/10.1002/fld.1005>.
- Raissi, M, Wang, Z, Triantafyllou, MS, and Karniadakis, GE (2019). "Deep Learning of Vortex-induced Vibrations," *J Fluid Mech*, 861, 119–137. <https://doi.org/10.1017/jfm.2018.872>.
- Shi, W, et al. (2023). "Short-term Motion Prediction of Floating Offshore Wind Turbine Based on Multi-Input LSTM Neural Network," *Ocean Eng*, 280(May), 114558. <https://doi.org/10.1016/j.oceaneng.2023.114558>.
- Sonka, M, Hlavac, V, and Boyle, R (1993). *Image Processing, Analysis and Machine Vision*, Springer. <https://link.springer.com/book/10.1007/978-1-4899-3216-7>.
- Ye, M, and Chen, H-C (2019). "VIV Simulation of Riser-Conductor Systems Including Nonlinear Soil-Structure Interactions," *Ocean Syst Eng*, 9(3), 241–259. <https://doi.org/10.12989/ose.2019.9.3.241>.
- Ye, M-K, Wang, N-N, Chen, H-C, and Wan, DC (2023). "CFD Simulations Targeting the Performance of the NTNU BT1 Wind Turbine Using Overset Grids," *J Hydrodyn*, 35(5), 954–962. <https://doi.org/10.1007/s42241-023-0065-4>.
- Yu, Y, Si, X, Hu, C, and Zhang, J (2019). "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Comput*, 31(7), 1235–1270. https://doi.org/10.1162/neco_a_01199.
- Zhao, Z-Q, Zheng, P, Xu, S-T, and Wu, X (2019). "Object Detection with Deep Learning: A Review," *IEEE Trans Neural Networks Learn Syst*, 30(11), 3212–3232. <http://doi.org/10.1109/TNNLS.2018.2876865>.