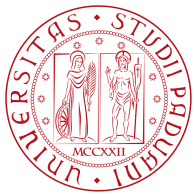


University of Padua

Department of Information Engineering

Master Degree in Computer Engineering

Deep learning for Fatigue Damage prediction in offshore structures

Supervisor
Prof. Emanuele Menegatti

Coadvisor
Prof. Stefano Ghidoni

Coadvisor
Dott. Ing. Nicolò Spiezia

Coadvisor
Dott. Ing. Alberto Forti

Student
Samuel Zanella

ACADEMIC YEAR 2022-2023

Degree date 17-04-2023

Index

Acknowledgments

I would like to express my gratitude to my advisors at M3E Dott. Ing. Nicolò Spiezia and Dott. Ing. Alberto Forti, for their feedback and guidance. Their help made these months of work on the thesis an extremely valuable period of professional and human growth.

I would like also to thank my advisors at the University of Padova, Prof. Stefano Ghidoni and Prof. Emanuele Menegatti. I am really grateful for having had the opportunity to find and pursue this work.

Finally, a special thank you goes to my family, closest friends and girlfriend. Their loving support and patience with me during this period has been one of the most important factors behind this achievement.

Abstract

This thesis investigates a predictive maintenance approach for estimating the Fatigue Damage in crucial components of offshore floating cables and pipes (umbilicals, riser, flowlines, etc.). The direct monitoring of these slender structures with sensors in critical sections is impractical and not cost efficient. Therefore, for the prediction of the Fatigue Damage, a so-called “Virtual Sensor” is implemented to forecast the mechanical loads in critical points, due to the motions of the floating part. Such Virtual Sensor consists of a Recurrent Neural Network trained on synthetic data from FEM simulations. The results suggest the potentials and limits of the approach in estimating the loads and transferring this information to the fatigue damage estimation.

1

Introduction

Different typologies of offshore platforms are currently deployed in the sea/ocean for energy production purposes. In particular, such structures can be aimed at the extraction and elaboration of fossil fuels or provide renewable energy from wind especially, but also from waves, tides, salinity etc. This thesis is taking into consideration floating platforms, and in particular it tackles the problem of estimating the remaining life of components crucial to these specific structures for predictive maintenance, with the goal of reducing costs by tuning the life of service without sacrificing safety.

The considered platforms are constrained to the seabed by means of some mooring chains, maintaining therefore some degrees of freedom for movement and rotation. In order to carry extracted resources / water or to transport the produced energy to the ground, some submarine pipes are installed, connecting the floating part with the seafloor. Such pipes represent crucial components that would cause serious environmental and economic damage if broken during usage, therefore it is important to schedule their replacement carefully according to their life

of service. The actual lifespan of the submarine pipes is variable but can be approximately estimated by assessing the Fatigue Damage on critical points, a metric that depends on the stress loads to which the component is subject in the considered critical nodes. Such stress loads are calculated using the time series of axial tension and bending moments in the considered points of the pipes. which could be obtained by applying physical sensors. However, direct subsea monitoring of submarine tubes of offshore platforms over the entire service life implies important costs. Furthermore, the feasibility of such monitoring depends on the locations where the sensors are required, and fatigue critical areas are typically not practical to install sensors, for example in the Touch Down Zone (TDZ). A lack of known response in service can result in qualitative inspection planning, conservative fatigue predictions and reduced asset utilization.

Problem and workflow

The problem of monitoring the critical points can be tackled by replacing them with so-called Virtual Sensors, that produce sensor data of the desired positions without being physically installed in such locations. A Virtual Sensor with these capabilities can be implemented through a data-driven algorithm, i.e. a Recurrent Neural Network (RNN) trained on synthetic data from numerical simulations. The input of such RNN consists of the time series of motions on the floating platform (and eventually other measures), while the output consists of the time series of axial tension and bending moments in the crucial points.

Feeding real data to virtual sensors allows to obtain the desired time series in the critical nodes of submarine pipes, that determines the stress loadings which can be used to compute an estimation of the Fatigue Damage using SN life estimation technique based on Miner's Rule. The workflow is described in Figure 1.1.

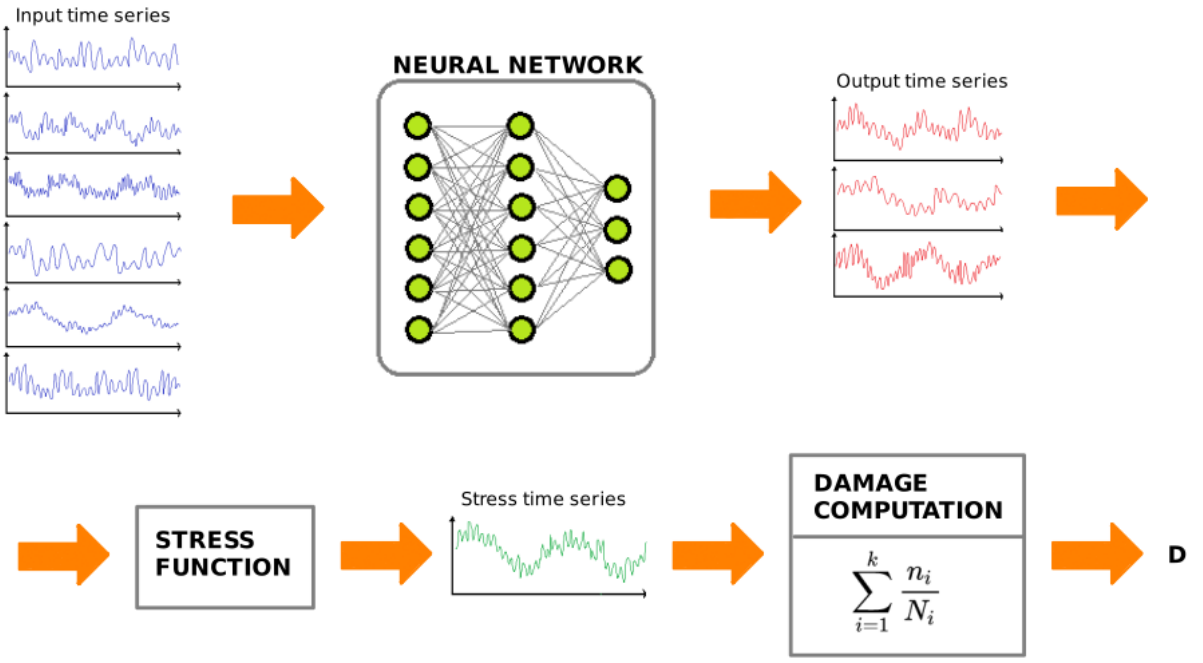


Figure 1.1: Workflow of Fatigue Damage estimation from time series forecasting.

Literature

The idea of using numerical simulations to train *Artificial Neural Network* (ANN) to forecast time series of relevant measures in crucial components of offshore structures has already been studied in the literature [1] [2] [3] [4]. For example, M. Yetkin and Y. Kim [1], deploy a *nonlinear autoregressive with external input* (NARX) model, based on a *Recurrent Neural Network* (RNN) trained on data from numerical simulations based on different sea conditions, to predict the top-tension of a mooring line of a *floating production storage and offloading* (FPSO) with the floater's motions in input. The authors explored the non-linearity of the system and its contribution to the response, extracting the transfer functions of different orders. S. Sundararaman [2] et al, provided a framework to estimate the fatigue damage on critical points of the riser of an offshore platform. The time series of the output data is learned by training ANN on Finite Element Analysis (FEA) data, specifically two approaches are compared: NARX and *Convolutional Neural Network* (CNN). An estimation of the fatigue damage is computed as the average

of the relative errors on distinct sea states simulations, weighted by the probabilities of such sea states. The NARX approach shows great results with relative errors on Fatigue Damage always under 5%. H. S. Gustad [3], applies RNN to forecast the bending moments in critical points of the riser and wellhead system of an offshore structure, training on data produced by two-dimensional numerical simulations. Architectures, hyperparameters, optimization methods and positioning / number of sensors in input is discussed, together with the ability of the model to generalize well on noisy data. C. Millevoi [4], used FEM synthetic data produced using the numerical simulator OrcaFlex to train and test the time series forecasting algorithm, in this case a Neural Network relying on *Long short term memory* (LSTM) cells in order to well capture long term memory dependencies. The data available was a set of 1 hour long simulations with different sea conditions. This work includes a sensitivity analysis to tune the parameters of the Neural Network and understand which perform better, and studies the impact on performances of the addition of time series from other points of the submarine tube (which are still easier to monitor than the critical point) improves performances. The described approach is shown to perform well considering metrics as Mean Squared Error (MSE) and R-squared (R2) for evaluation of the time series forecasting, when training on single simulations (fixed sea state), while the training on multiple simulations with similar sea conditions is attempted up to 7 simulations with promising results.

Structure

The thesis will be structured in the following chapters.

Chapter 2, *Theory*, illustrates the theory behind the main mathematical and scientific tools and concepts used in this work, in particular the Recurrent Neural Networks, the concept of Fatigue Damage and Rainflow Analysis for its computation.

Chapter 3, *Data Exploration*, presents and describes the data used and highlights the steps performed to extract some useful insights from it.

Chapter 4, *Time Series Forecasting*, deals with the deep learning techniques used for time series forecasting on external input and the evaluation of the results obtained in terms of R2 scores of the predictions.

Chapter 5, *Fatigue Damage Estimation*, investigates the the lack of correlation between the performances in terms of R2 scores of the predictions and the relative errors on Fatigue Damage and propose solutions.

Last, Chapter 6, *Conclusions*, summarizes the main contributions of this thesis on the considered problem, highlighting advantages and limits of the approach studied, and illustrates some possible directions for future research.

2

Theory

This Chapter provides an overview of the theoretical background behind the work of this thesis. In particular, the first Section introduces the major ideas behind the deep learning models exploited in this thesis to solve the problem of *forecasting* time series through *regression*. The basic notions behind feedforward neural networks and the mechanics of the training are assumed to be known: the concepts of forward propagation and backward-propagation to compute the gradient of the loss function and the methods to minimize the loss function such as Stochastic Gradient Descent (SGD) or Adam, and the knowledge of the main activation functions.

The second Section illustrates the Fatigue Damage metric, and how it is defined in our context. Next, the Section continues with the description of the SN life estimation based on the linear cumulative damage and the cycle counting methods applied to extract meaningful information from stress loading histories. Last, the evaluation metric used for the Fatigue Damage is explained.

2.1 Deep Learning

2.1.1 Recurrent Neural Network

A *time series* is an ordered sequence of points, which can be univariate or multivariate. A univariate time series is a sequence of scalars while a multivariate time series is a sequence of points in a vectorial space.

There are several examples of machine learning tasks that require dealing with generic sequential data in output, such as image captioning, speech synthesis, and music generation; in input, such as time series forecasting, video analysis, and musical information retrieval; or in both input and output, such as translating passages of text from one natural language to another, engaging in dialogue, or controlling a robot.

The main task of interest in thesis is *time series forecasting*, which consists of predicting the future points of a time series. This task can be viewed as a *regression* problem. In particular, the problem can be modelled as the iterative prediction of point at time t of the considered time series, knowing either the same time series until point $t-1$ (*AutoRegressive model (AR)*), either some other distinct but correlated time series until point $t-1$ (*EXogenous inputs model (EX)*), or both (*(Nonlinear) AutoRegressive with eXogenous inputs model (NARX)*).

A *regressive* model with the described purpose can be implemented with an ANN that receives in input the past sequence of points as a vector of fixed-size (window length) and process it through different layers to provide the final output. CNNs have been adopted in this scenario with good results, but in general Recurrent Neural Networks show the best performance [5].
Recurrent Neural Networks [6] [7], or RNNs, are Neural Networks designed precisely for processing sequential data. In a RNN, each member of the input is a function of the previous members of the output, and results from the application of the same update rule used on the previous output. In this way, a sequence can be fed to the network point by point, rather than entirely, applying always the same function to it. This formulation contains the idea of recurrency

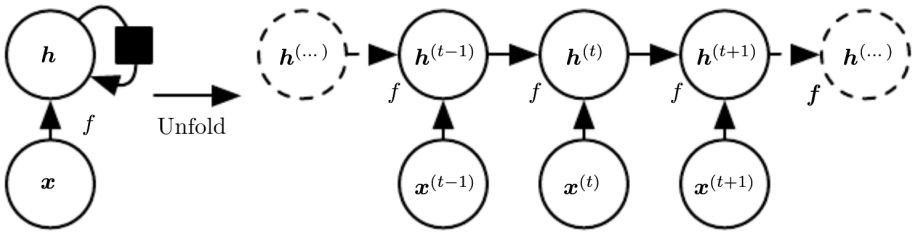


Figure 2.1: Computational graph of a simple RNN with only hidden states on the left. Associated unfolded computational graph on the right.

to which the network owes the name.

The main advantage of this approach comes from the *parameter sharing* across different positions in time, which provides benefits in terms of generalization capabilities and size of the network. This factor also allows for scalability to much longer sequences than possible by exploiting other not sequence based solutions, and sometimes the possibility to process sequences of lengths not seen during training.

A basic recurrent neural network can be expressed as a dynamical system driven by an external signal, as in the following equation:

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta) \tag{2.1}$$

where $h^{(t)}$ represents the hidden state of the system, $x^{(t)}$ the external signal, $f(\cdot)$ the recursive function, with parameters θ. Such equation can be expanded by recursively expressing the hidden state inside the function as a function of the previous hidden state. This operation, applied to the computational graph associated to the function, is called *unfolding*, and can be visualized in Figure 2.1.

In order to illustrate the equations that govern a RNN, let's consider a classic design for a Recurrent Neural Network, that maps an input sequence of **x** values to a corresponding sequence of output **o** values (see Figure 2.3). In this network, the computation of the output at time t is

given by:

$$h^{(t)} = tanh(b + Wh^{(t-1)} + Ux^{(t)}) \tag{2.2}$$

$$o^{(t)} = c + Vh^{(t)} \tag{2.3}$$

$$\hat{y}^{(t)} = \varphi(o^{(t)}) \tag{2.4}$$

where the parameters are the bias vectors b and c along with the weights matrices U, V and W, respectively, for input-to-hidden, hidden-to-output and hidden-to-hidden connections, and $tanh(\cdot)$ is the activation function of the hidden layer, while $\varphi(\cdot)$ represents the activation function of the output neuron.

Training the weights of a RNN requires the computation of the gradient of the loss function as for any normal feedforward neural network. For a Recurrent Neural Network, this means performing a forward propagation pass moving left to right through the unrolled computational graph illustrated in Figure 2.3, followed by a backward propagation pass moving left to right. The forward propagation graph is inherently sequential, thus each time step may be computed only after the previous one. This implies that the runtime for a sequence of length τ is $O(\tau)$ and cannot be reduced by parallelization. The memory cost is also $O(\tau)$, since states computed in the forward propagation must be stored until the backward pass. Therefore a RNN is quite expensive to train. The procedure described of applying the back-propagation to the unfolded computational graph is known as *Back-Propagation Through Time (BPTT)*.

Vanishing and exploding gradient

RNNs involve the composition of the same function multiple times, once per time step. In particular, such function composition resembles matrix multiplication. Under this intuition, we can think of the recurrence relation:

$$h^{(t)} = W^T h^{(t-1)} \tag{2.5}$$

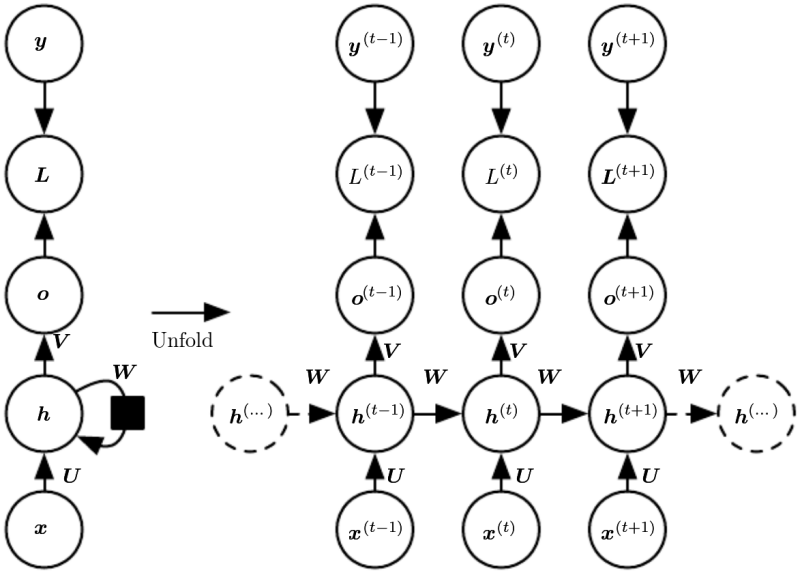


Figure 2.2: Computational graph of a RNN that maps an input sequence of $\mathbf{x}$ values to a corresponding sequence of output $\mathbf{o}$ values.

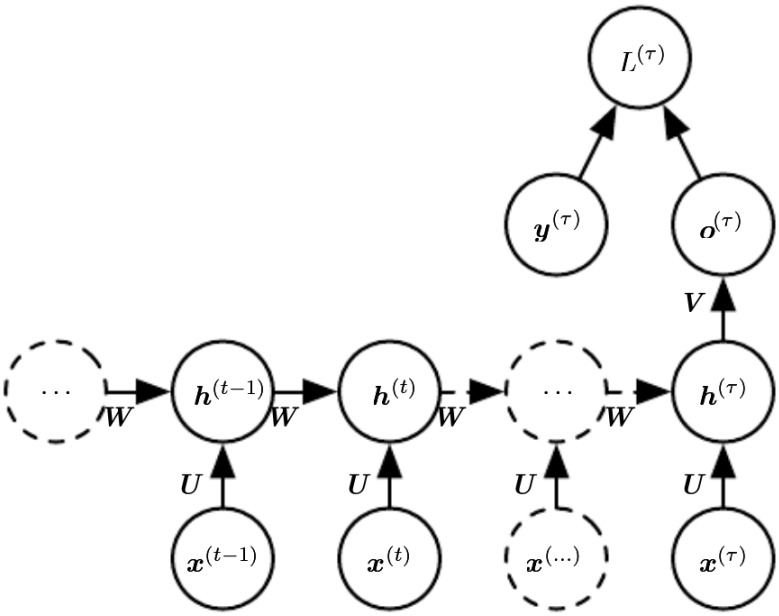


Figure 2.3: Unfolded computational graph of a RNN with one single output. This design can be adopted for example in regression.

as a very simple recurrent neural network lacking a nonlinear activation function and without inputs **x**. It can be argued that this recurrence relation describes the power method [6] and thus may be simplified to:

$$h^{(t)} = (W^T)^T h^{(0)} \tag{2.6}$$

and if W admits an eigendecomposition of the form

$$W = Q\Lambda Q^T \tag{2.7}$$

with orthogonal Q, the recurrence may be further simplified to:

$$h^{(t)} = Q^T \Lambda^t Q h^{(0)} \tag{2.8}$$

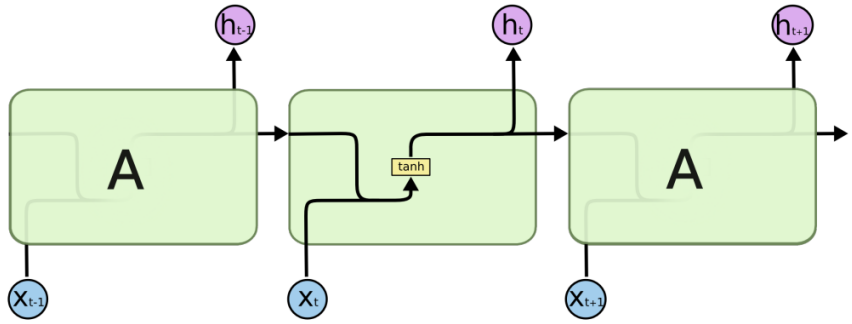
Raising the eigenvalues to the power of t, causes the decay to zero of those with magnitude less than one, and the explosion of those with magnitude greater than one. Therefore, any component of $h^{(0)}$ not aligned with the largest eigenvector will eventually be discarded.

This problem translates into the challenge of RNNs to learn long-term dependencies, since gradients propagated over many time steps tend to either vanish or explode.

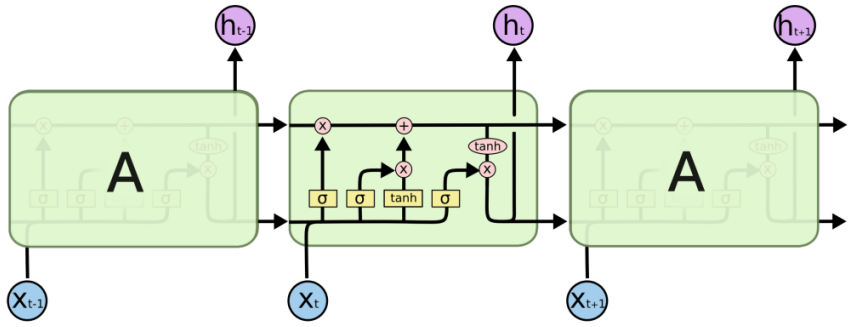
2.1.2 Long Short Term memory

Long Short Term Memory networks (LSTM) [7] were designed specifically to overcome the long-term dependency problem encountered by recurrent neural networks RNNs, due to the vanishing or exploding gradient problem.

LSTMs resemble standard Recurrent Neural Networks, with the difference that the ordinary recurrent node is replaced by a memory cell (see Figure 2.4), a composite unit, built from simpler nodes with the inclusion of multiplicative nodes, implementing gating mechanisms. The main advantage of LSTMs over vanilla RNNs, comes from the presence of gating of the hidden state,



(a) Vanilla RNN.



(b) LSTM.

Figure 2.4: Comparison between RNNs and LSTMs: notice that the difference relies in the module A, which is the recurrent node.

consisting in dedicated mechanisms establishing when a hidden state should be updated and also when it should be reset.

A detailed illustration of the functioning of an LSTM memory cell is provided in Figure 2.5. We can observe that a memory cell is equipped with an internal state and a number of multiplicative gates. Such gates determines whether (i) a given input should impact the internal state (the input gate), (ii) the internal state should be flushed to (the forget gate), and (iii) the internal state of a given neuron should be allowed to impact the cell's output (the output gate). First, input gate I_t, forget gate F_t and output gate O_t at current time step t, are fed with the input at the current time step X_t and the hidden state of the previous time step H_{t-1}, as illustrated in Figure 2.5a. Each gate consists of a fully connected layer with *sigmoid* activation function,

producing a value in the range (0,1). Mathematically, we obtain the following equations:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \tag{2.9}$$

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \tag{2.10}$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \tag{2.11}$$

where W_{xi}, W_{xf}, W_{xo} and W_{hi}, W_{hf}, W_{ho} are weight parameters and b_i, b_f, b_o are bias parameters.

Additionally, an input node $\tilde{C}_t$ with a *tanh* activation function (value range of (-1,1)) is provided with the same input data as the gates. This leads to the following equation at time step t:

$$\tilde{C}_t = tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c) \tag{2.12}$$

where W_{xc} and W_{hc} are weight parameters and b_c is a bias parameter. A quick illustration of the input node is shown in Fig. 2.5b.

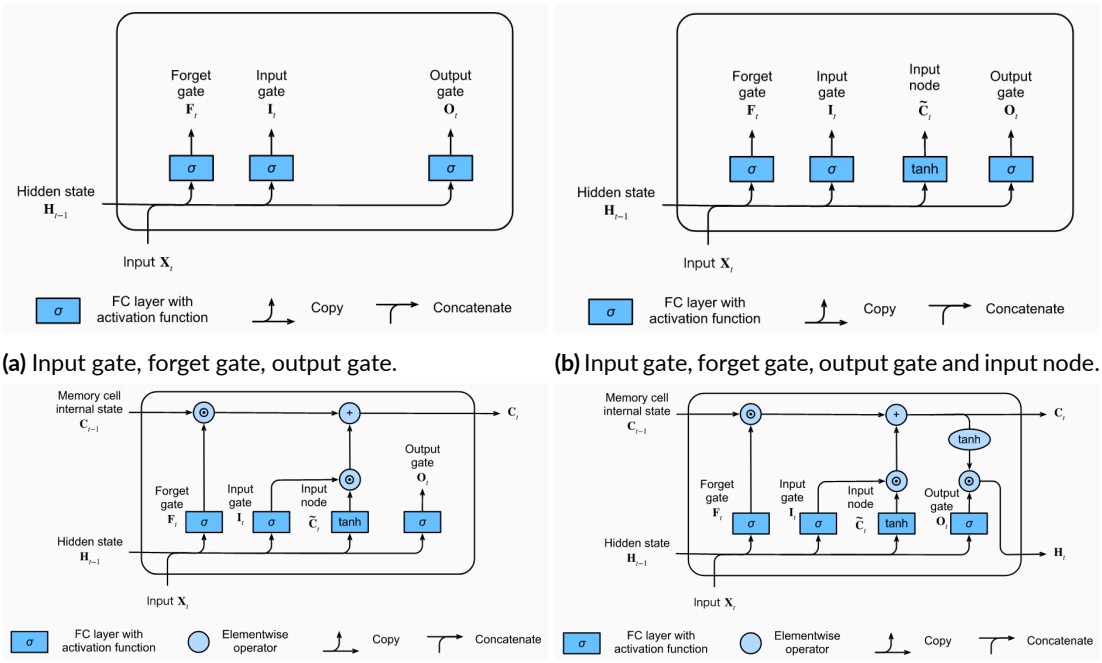
Intuitively, the input gate determines how much of the input node's value should be added to the current memory cell internal state. The forget gate determines whether to keep the current value of the memory or flush it. And the output gate determines whether the memory cell should influence the output at the current time step.

As previously mentioned, the input gate I_t governs how much we take new data into account via $\tilde{C}_t$. The forget gate F_t, instead, addresses how much of the old cell internal state C_{t-1} we retain. This is obtained with the following update equation:

$$C_t = F_t \circ C_{t-1} + I_t \circ \tilde{C}_t \tag{2.13}$$

where $\circ$ is the Hadamard (elementwise) product operator. Figure 2.5c summarizes the data flow until this point.

Note that forget gate set to 1 and input gate set to 0 keep the memory cell internal state C_{t-1}



(a) Input gate, forget gate, output gate.

(b) Input gate, forget gate, output gate and input node.

(c) Memory cell internal state.

(d) Entire LSTM node.

Figure 2.5: Illustration of an LSTM node.

constant forever. Input and forget gates play exactly the role of learning when to keep unchanged the memory cell internal state and when to perturb it in response to subsequent inputs. This design tackles the vanishing gradient problem, resulting in models that are much easier to train, especially when handling long sequences.

Last, the output of the memory cell is stored in the hidden state H_t: we first apply *tanh* to the memory cell internal state and then apply another point-wise multiplication, this time with the output gate. This ensures that the values of H_t are always in the interval (-1, 1):

$$H_t = O_t \circ tanh(C_t) \tag{2.14}$$

A value of the output gate close to 1 allows the memory cell internal state to impact the subsequent layers uninhibited, whereas output gate values close to 0 prevent the current memory from impacting other layers of the network at the current time step. In this way a memory cell

can accumulate information across many time steps without impacting the rest of the network (so long as the output gate takes values close to 0), and then suddenly impact the network at a subsequent time step as soon as the output gate flips from a value close to 0 to a value close to 1. The illustration of the overall data flow is provided in Figure 2.5d.

2.1.3 Sliding window approach

The regression task at hand consists in forecasting a value $y^{(t)}$ from an input sequence $x^{(0)},\ldots,x^{(t-1)}$. Considering a long time series of finite length T for input and target features x and y, we can therefore extract training samples of variable length t according to the time step of the considered target value $y^{(t)}$. However, the most common approach to train a Recurrent Neural Network for this purpose, consists of using a set of samples $([x^{(t-w)},\ldots,x^{(t-1)}],y^{(t)})$ for different time steps t, with fixed window length w, rather than using as inputs the whole past sequences $x^{(0)},\ldots,x^{(t-1)}$.

The first benefit of this approach is that handling data of fixed length for training is more practical than handling variable length data. On the other hand, part of the information contained in the input is lost by considering just a window of time steps. However, in practice this method proves to be effective provided that the chosen window length is sufficient to account for the long-term dependencies of the considered problem. Furthermore, using short time windows as inputs during training mitigates the problem of the high time complexity of BPTT presented in section 2.1.1.

2.1.4 Evaluation of time series forecasting

To assess the quality of the forecasting performed on a time series, we used the coefficient of determination.

The coefficient of determination, commonly referred to as R2 (“R squared”), is an evaluation metric defined as follow:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{N}(y_i-\bar{y})^2} \tag{2.15}$$

where y_i and $\hat{y}_i$ are the values at time i of the original and predicted time series respectively, $\bar{y}$ is the average of the values of the original time series y and N is the length of the time series. The fraction expresses the ratio between the variance in the error between original and predicted data (numerator) and the variation in the original data (denominator).

The metric assumes values in the range $(-\infty, 1]$, where $R2 = 1$ when the values $\hat{y}_i$ exactly match the values y_i. A model that always predict the mean $\bar{y}$, which can be considered a bad estimator, will have the variance of the error equal to the variance in the original data, leading to a R2 of 0. In the case in which the predicted values are worse than the original mean value, the coefficient of determination assumes a negative value. A model with $R2 \geq 0.5$ is a model in which more than half of the variance of the error is explained by the variance in the original data. According to these intuitions, the R2 score provides a quite informative measure of the quality of a single prediction.

2.2 Fatigue Damage

In material science, *fatigue* is the initiation and propagation of cracks in a material due to cyclic loading. Once initiated, a fatigue crack grows until failure occurs. The term fatigue is historically associated with metal components. However, most materials experience some sort of fatigue-related failure. Therefore, also the submarine pipes we are studying, composed of layers of distinct materials, are subject to some failures.

The American Society for Testing and Materials defines the *fatigue life*, N_f, as the number of stress cycles of a specified character that a specimen, i.e. a sample of the studied material or component, sustain before failure of a specified nature occurs [8]. An estimate of this value can be accessed by averaging the results of various fatigue tests, performed for simple cyclic stress loadings of constant amplitude. However, in the most common scenario of a variable amplitude loading history, how can we define a metric to express the damage due to fatigue to predict failure?

This Chapter covers briefly the theoretical background behind the definition of Fatigue Damage and the methods to compute it exploited in the current thesis. Specifically, we start by defining the stress function for our study case. We proceed with an introduction to cumulative damages theories, focusing on Miner's rule. Next, life estimation approach is presented and last we illustrate the Rainflow Counting.

2.2.1 Stress loading

The Fatigue Damage is extracted from the stress time history at a specific position, where the stress is a function of the axial tension and the bending moments. In this thesis, we consider a pipe consisting of a long tube composed of various concentric layers. A critical area in the pipe is therefore a circular section. For this reason, a critical node of the pipe is associated to different stress time histories, according to the considered points of the circular section. In order to account for this aspect, our stress function is parametric in the angle α, as visible in Figure 2.6, with the following formulation:

$$\sigma(\alpha) = \frac{Tw}{Area} + \frac{OD}{2} \cdot (Mx \cdot \sin\alpha - My \cdot \cos\alpha) \cdot \frac{1}{Inertia} \tag{2.16}$$

where Tw, Mx, My indicate respectively the axial tension and the bending moments at x-axis and y-axis, and $Area$, OD and $Inertia$ are fixed parameters characterizing the pipe.

2.2.2 Cumulative Damage

The term *cumulative damage*[9] refers to the fatigue effects of loading events other than constant amplitude cycles.

A widely known method to compute the cumulative damage is the *Miner's rule* or *Palmgren-Miner linear damage hypothesis*, which assumes that distinct loading events consume fractions of life that can be simply summed together to obtain a measure of the damage. In particular, if

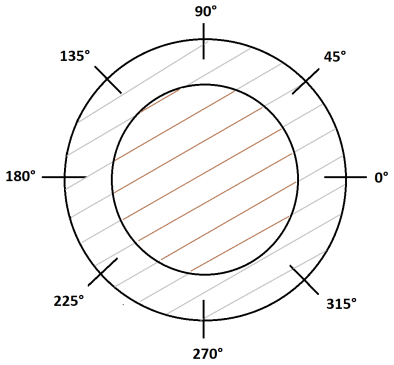


Figure 2.6: Section of the pipe.

there are k different stress magnitudes in a spectrum, $S_i (1 \leq i \leq k)$, the cumulative damage D is expressed as:

$$D = \sum_{i=1}^{k} \frac{n_i}{N_{fi}} \tag{2.17}$$

where n_i/N_{fi} expresses the fraction of life associated to the stress magnitude S_i, with n_i number of cycles extracted from the loading history and N_{fi} number of cycles before failure occurs (determined by fatigue tests). According to this rule, failure is expected to occur when:

$$D = \sum_{i=1}^{k} \frac{n_i}{N_{fi}} \geq C \tag{2.18}$$

where $C > 0$ is a fixed real constant. In the specific case of the submarine pipes considered, this formula is often modified to the average of the Fatigue Damages in distinct sea conditions, weighted over the probability of the sea state, the resulting expression is the following:

$$D = \sum_{j=1}^{N_{SS}} \sum_{i=1}^{k_j} \frac{n_i^j}{N_{fi}} \gamma_j \geq C \tag{2.19}$$

where the external sum iterates over the index j of the N_{SS} sea states accounted for, and γ_j is the probability of sea state j (with $\sum_{j=1}^{N_{SS}} \gamma_j = 1$); while k_j is the size of the stresses spectrum of sea

state j.

The assumption of linear damage is open to many objections. For example:

- sequence and interaction of load events may have great impact on life;
- the rate of damage accumulation may depend on the load amplitude;
- experimental evidence often indicates that the failure occurs for $\sum_{i=1}^{k} n_i/N_{fi} \neq C$ for a low-to-high or a high-to-low loading sequence.

For these reasons, other theories have been proposed in the literature, such as nonlinear damage theories. Nonetheless, linear damage rule is still the most widely used because of its simplicity and the better agreement with data from many different tests, compared to the other methods proposed.

The following Sections will cover how is the fatigue life of a component under various stress levels assessed and how can we decompose a variable amplitude loading time history into a set of simple cyclic loadings.

2.2.3 Life estimation using S-N approach

In order to access the fatigue life, the current thesis relies on the *S-N life estimation approach*, which allows the cumulative damage to be computed based on the SN curve of a given component / material.

The *stress-life (SN) curve* is a plot of the cyclic stress (S) against the cycles to failure (N), usually on a logarithmic scale. It is usually obtained performing several fatigue tests for distinct simple cyclic stress loadings. An example of an SN curve is provided in Figure 2.7.

The SN life estimation starts with a decomposition of the stress loading time-history into a set of number of cycles for distinct cyclic stresses of constant amplitude in a spectrum, performed using a cycle counting method such as Rainflow Counting. The information extracted, combined

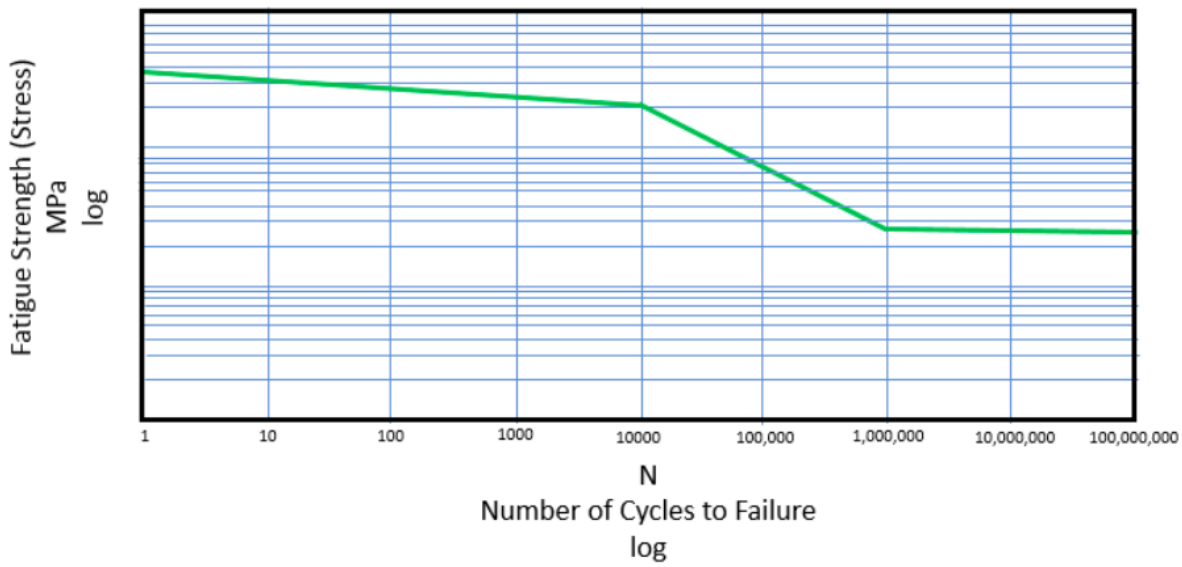


Figure 2.7: Example of an SN curve.

with the SN curve, provide the individual contributions to the cumulative damage of each stress cycle in the spectrum, that can be aggregated using with Miner's rule to compute the total Fatigue Damage.

To provide a complete overview, Section 2.2.4 explains how to perform cycle counting with the Rainflow method.

2.2.4 Rainflow Counting

In 1967, T. Endo proposed a method called Rainflow Counting [10] [11] to breakdown any load-time history into its constituent fatigue cycles. This method consists of four steps as documented in standards as ASTM E1049: hysteresis filtering, peak-valley filtering, discretization and Four Point Counting method.

The first three steps, illustrated in Figure 2.8, consist of a preprocessing of the load time-history before the cycle counting. Hysteresis filtering (see Figure 2.8a) removes very small cycles (with low impact on the damage) from the load time-history by defining a gate of a

specific amplitude, usually depending on the bin size that is discussed in the discretization step, and removing turning points smaller than such gate. Next, the time-history is reduced to a sequence of reversals in slope, i.e. points of local maximum or minimum, so-called peaks and valley (refer to Figure 2.8b). Last, the discretization or binning step consist in dividing the stress load axis into discrete values, referred to as "bins" (a bin is then a fixed amplitude stress range), and each point from the previous step is mapped to the closest bin (increasing the number of bins leads to higher accuracy).

After the preprocessing, a cycle counting method such as the Four Point Counting method is applied. This procedure aims at decomposing the peaks and valleys stress sequence into a finite number of stress cycles identified by amplitude and mean from a stress loading, and returning them in the form of a Rainflow matrix (of "Range-Mean" form in this case). The Rainflow matrix that we will consider here is of the form "From-to", a *nxn* matrix where *n* equals the number of bins, and each cell counts the number of complete cycles going from the stress level bin of the row index to the stress level bin of the column index. As we will see this matrix can be easily transformed in the form Range-Mean previously mentioned.

The Four Counting method can be summarized in the following steps:

1. Consider four consecutive points $S1, S2, S3, S4$ in the sequence of reversals.
2. Define inner stress range $SR_I := |S2 - S3|$
3. Define outer stress range $SR_O := |S1 - S4|$
4. If $SR_I \leq SR_O$ and the points $S2, S3$ are inside the interval defined by $S1$ and $S4$, a complete cycle is counted. In this case the Rainflow matrix is updated with +1 as in the example of Figure 2.9a, $S2$ and $S3$ are removed from the sequence of four points, and the first four points of the remaining sequence are evaluated again.
5. If $SR_I \geq SR_O$ and $S2, S3$ are not inside the interval defined by $S1$ and $S4$, there is not a complete cycle (Figure 2.9b). The next four points are evaluated.

This process continues until all identifiable cycles are removed, which generally leaves out some points, forming the so-called Residue of the Rainflow matrix (Figure 2.9d). The residue contains the largest unclosed cycles present in the time history. An illustration of the process is provided in Figure 2.9.

The Rainflow matrix is usually represented with colored cells according to a color index as in Figure 2.10 or as a three dimensional histogram. From the Rainflow matrix obtained and its Residue we can extract the desired information:

- Cycle amplitude. The range or amplitude of a cycle identified by row *From* and *To* To is the $|From - To|$.
- Cycle mean. The mean of the cycle is given by $(To + From)/2$
- Number of cycles is expressed in the content of the cells of the matrix, with some half cycles due to the residue.

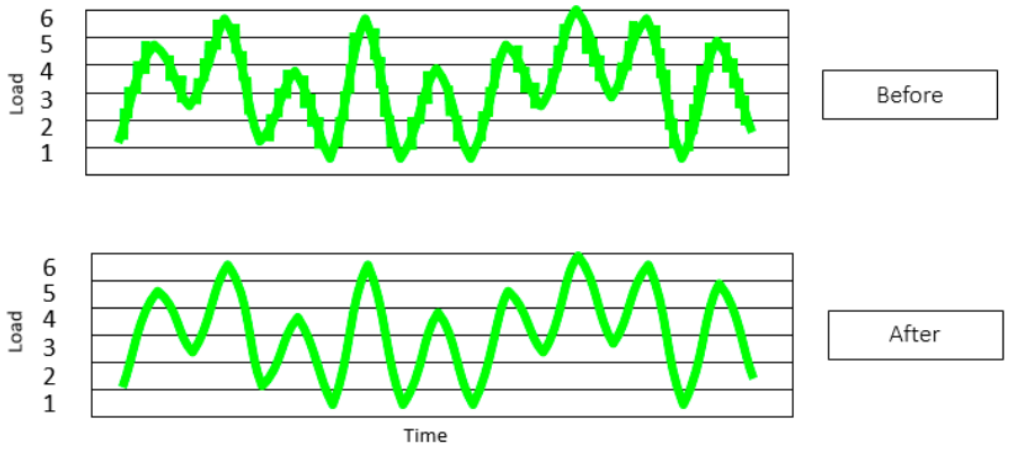
2.2.5 Evaluation of Fatigue Damage estimation

Following a common practice, we evaluate the quality of the Fatigue Damage computed on the forecasted stress time series, in terms of relative error with respect to the Fatigue Damage of the original time series, as follow:

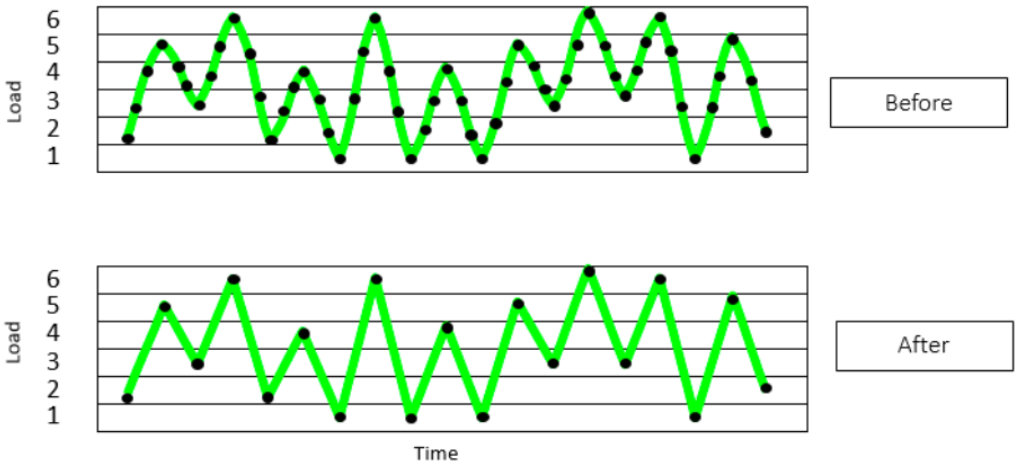
$$ERR(\%) = \frac{|FD_{true} - FD_{pred}|}{FD_{true}} \cdot 100 \tag{2.20}$$

where FD_{true} and FD_{pred} refer respectively to the true and predicted values of the Fatigue Damage. The value is expressed as a percentage. However, this metric varies inside the range $[0, \text{inf})$, since the Fatigue Damage is a positive real value. In particular, we obtain relative errors in the range $[0, 100]$ if the predicted *FD* is smaller than the original *FD*, and values in the range $[0, \text{inf})$

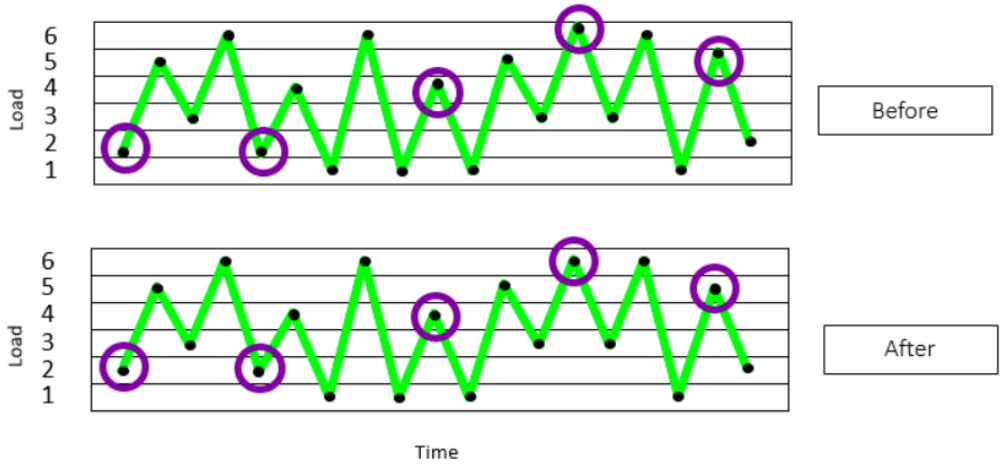
in the opposite case. For the same reason the metric is said to be not “symmetric”, which leads to a greater relative error if the predicted *FD* is bigger than the true *FD* (rather than predicted *FD* being smaller than true *FD*) while their absolute difference is the same.



(a) Hysteresis filtering.

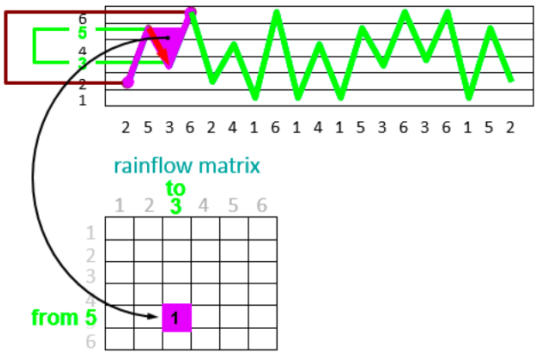


(b) Peak-valley filtering

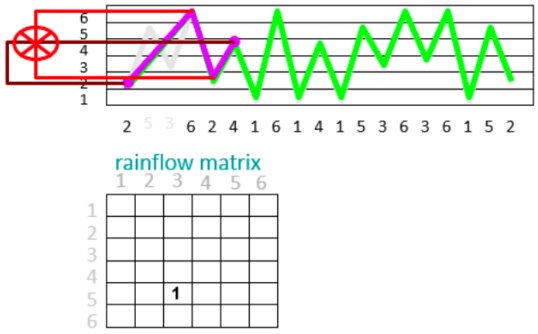


(c) Binning.

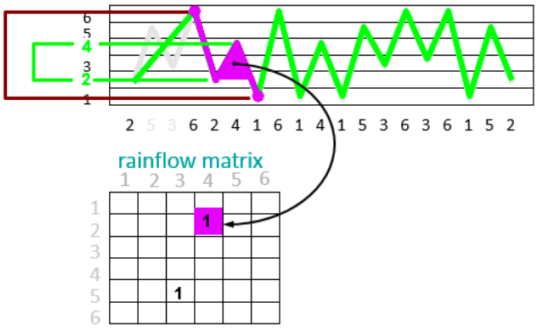
Figure 2.8: Preprocessing of a time-history in Rainflow Counting.



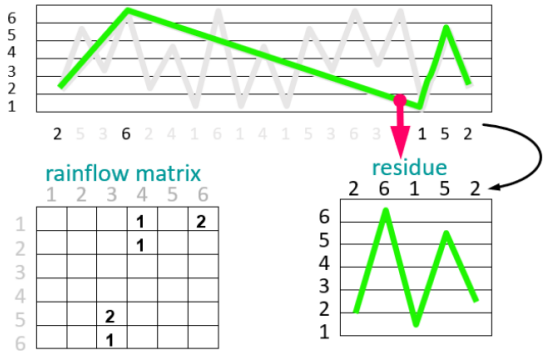
(a) Complete cycle identified: Rainflow matrix updated, inner points removed and the new first four consecutive points are evaluated.



(b) Incomplete cycle: evaluate the next four consecutive points.



(c) New complete cycle found.



(d) Final output of the four point counting method: Rainflow Matrix and Residue

Figure 2.9: Illustration of the four point counting method: in green the remaining sequence of points and in purple the currently considered four consecutive points at each iteration.

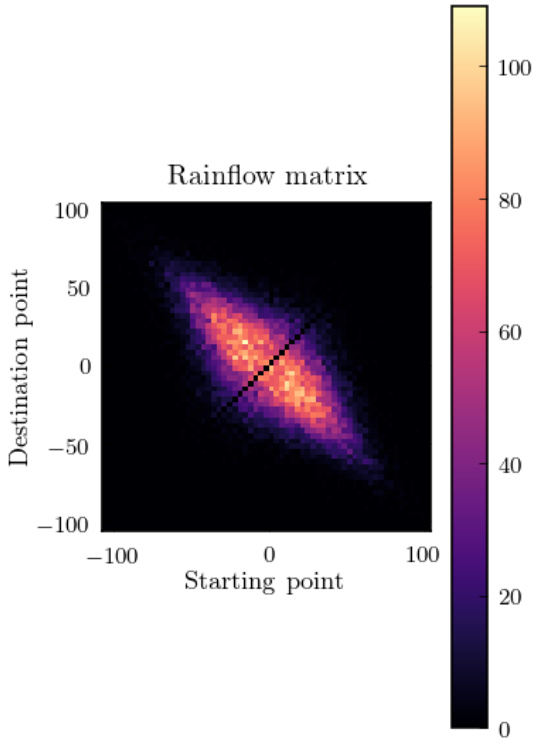


Figure 2.10: Rainflow matrix.

3

Data Exploration

The synthetic data used for this thesis is obtained by running FEM simulations using Orcaflex *. The data produced by the simulations consists of multivariate time series with a sampling period of 0.25 s, and a total length of 1 hour. A small time window of one simulation is shown in Figure 3.1.

The columns represent different measures considered associated to distinct points of the floating platform that is object of study. In particular, Figure 3.2 illustrates the setting floating platform - pipe and highlights the name of the critical nodes that we will refer to from now on.

The variables associated to the columns are grouped by nodes, where each node can be of interest for either physical monitoring (measures easy to monitor, i.e. motions, rotations, accelerations), either virtual sensing (measures difficult to monitor, i.e. axial tension and bending moments). The variables associated to physical monitoring are meant to be used as input to our

*Orcaflex is a package for dynamic analysis software of offshore marine systems. It is used for research analysis and engineering in works involving oil & gas, wet renewables, oceanographic, seismic, defence or aquaculture. The software is widely applied for riser systems, moorings, pipelines, cable-stayed structures and others

Time	FWT_X	FWT_Y	FWT_Z	FWT_RX	FWT_RY	FWT_RZ	OW_Azi	OW_Dec	OW_Gam	OW_T	...
0 days 00:00:00	-3.173447	-1.640042	-17.928789	1.385504	0.100327	-0.049783	78.589296	167.759284	2.647144	3.669909e+06	...
0 days 00:00:00.250000	-3.165171	-1.669152	-17.917704	1.398555	0.099311	-0.045621	78.659912	167.727084	2.712256	3.662808e+06	...
0 days 00:00:00.500000	-3.156104	-1.700998	-17.907665	1.403964	0.096139	-0.042368	78.727638	167.694979	2.774634	3.655396e+06	...
0 days 00:00:00.750000	-3.146301	-1.735278	-17.899103	1.401620	0.090795	-0.039998	78.790780	167.663574	2.832578	3.647845e+06	...
0 days 00:00:01	-3.135838	-1.771663	-17.892464	1.391457	0.083308	-0.038661	78.848500	167.633639	2.885390	3.640280e+06	...

Figure 3.1: Small window of time of the simulation performed in sea state 1.

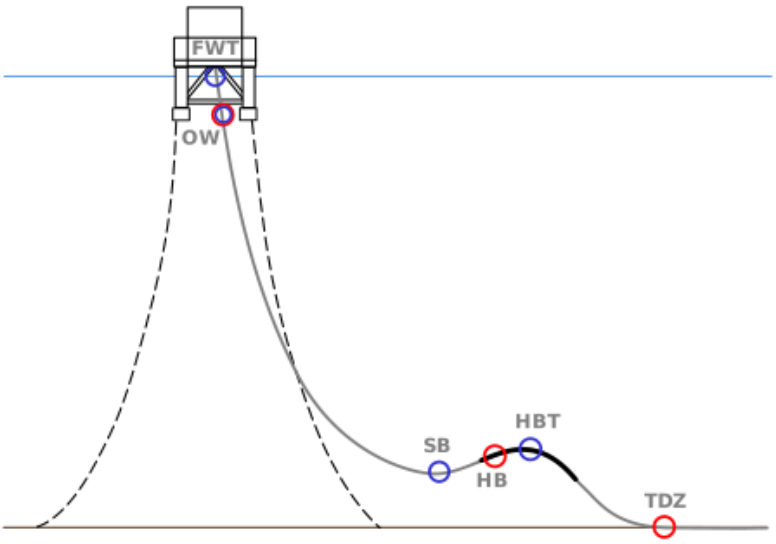


Figure 3.2: Representation of the floating wind tower system. The critical points are: FWT (node attached to the floating wind tower), OW (offshore weld node), SB (sag-bend node), HB (hog-bend node), HBT (hog-bend top node) an TDZ (touch down zone node). These points are colored to distinguish between input nodes (blue circles), output nodes (red circles), or both.

Virtual Sensor, that will output the remaining variables.

The former set of variables is associated to nodes *FWT, OW, SB, HBT*, while the latter to nodes *OW, HB, TDZ*. The variables *FWT_X, FWT_Y, FWT_Z, FWT_RX, FWT_RY, FWT_RZ* express the motions (linear and angular) in the 3 axes of the top side of the pipe, directly attached to the floating wind tower. *OW_Azi, OW_Dec, OW_Gam* contains the time histories of azimuth, declination and gamma at offshore weld node. *SB_accX, SB_accY, SB_accZ, HBT_accX, HBT_accY, HBT_accZ* refer to acceleration at sag bend and hog bend nodes. The remaining columns contain histories of tensions at bending moments in the output nodes. In this thesis, the point of the pipe of main interest is the Touch Down Zone (TDZ), where the axial tension and bending moments in x and y axis are collected through time in the variables *TDZ_T, TDZ_Mx, TDZ_My*.

In total, 808 simulations were available, one for each sea state of interest for our specific case of study. Each sea state is characterized by a probability of occurrence and a specific set of parameters defining the sea conditions to simulate through Orcaflex. Such parameters are listed here for an intuitive understanding: roll damping (% of critical damping), wave height -Hs- (m), wave period -Tp- (s), wave gamma, wave direction (deg), associated wind speed (m/s), wind direction (deg), associated current speed (m/s), current direction (deg).

A selection among all the available sea states is visible together with their respective parameters in the Appendix A in Table A.1. The distributions of sea states parameters across all simulations is shown in Figure 3.3.

It is worth noting that by ordering the sea states in terms of probability and considering the first ones, the cumulated probability grows pretty fast compared to the number of simulations (see Figure 3.4), i.e. a small subset of all the available sea states actually provides an important fraction of the sea conditions occurring during a year. Indeed, a cumulated probability of 80% can be obtained with just the 8.8% of the simulations, a probability of 95% is given by 21.7% of simulations , and 99% of sea state scenarios accounts for the 38.5% of simulations. However, sea scenarios with an extremely low probability of occurring, can be associated to extreme weather conditions, causing much higher stress and thus Fatigue Damage to the pipes.

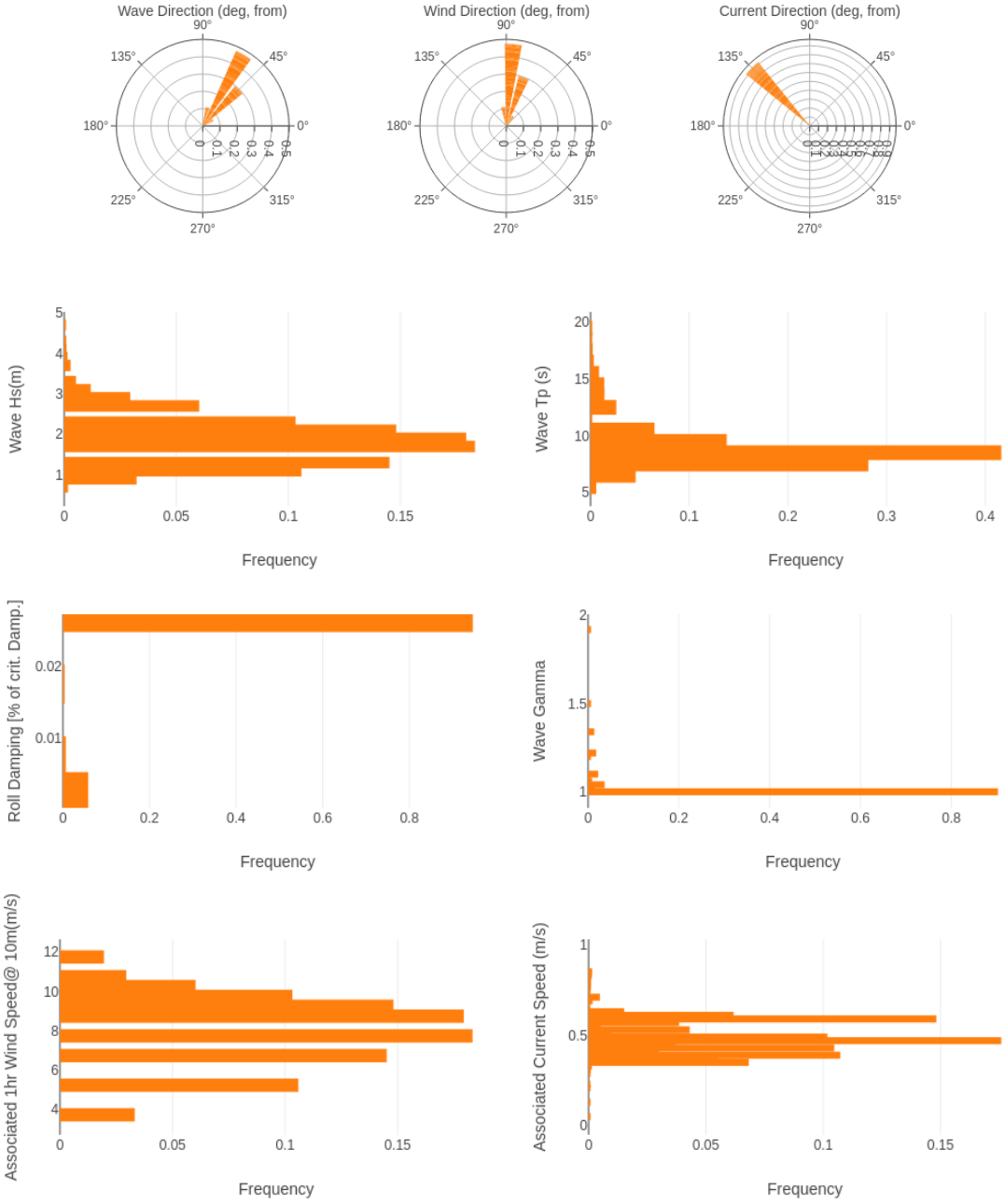


Figure 3.3: Probability histograms showing the distribution of values of each sea state parameter, considering the sea states probabilities. From the top row to the bottom row: wave, wind and current directions, wave height and period, roll damping and wave gamma, wind and current speed.

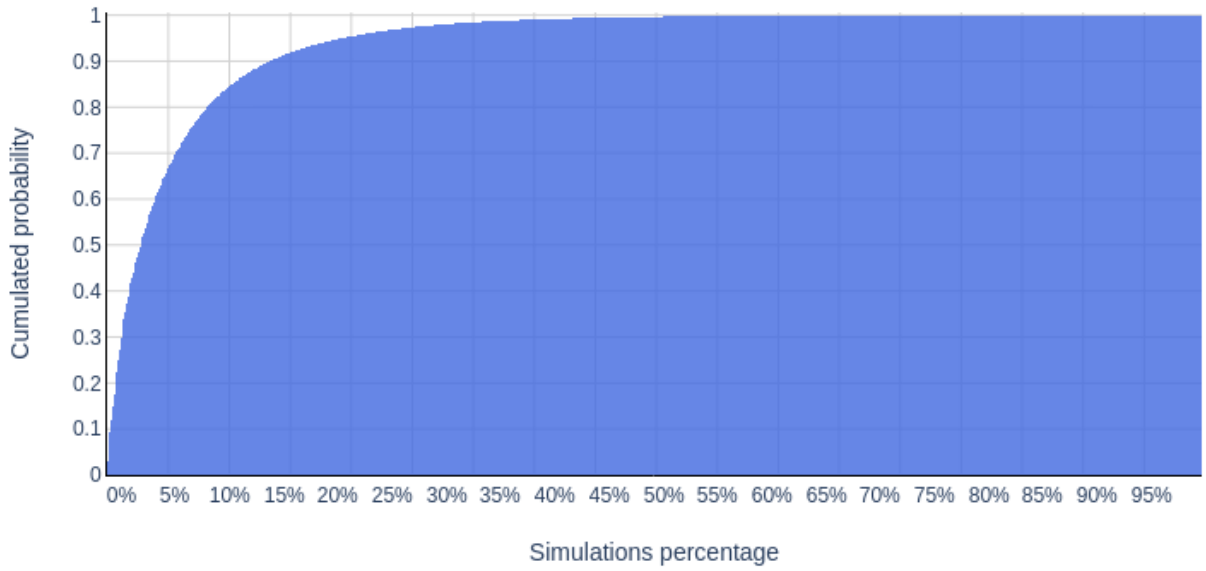


Figure 3.4: Cumulative distribution of sea states' probability by ordering sea states by decreasing probability.

Due to the synthetic nature of data, the preprocessing did not require any noise filtering, handling missed data points or detection of anomalies. The only required action involved cutting the first seconds of the time series, associated to data still unstable and thus not reliable.

3.1 Time series visualization

The first step of data exploration has been to plot the time series of some relevant variables in input and output. For this purpose, the simulation associated to the sea conditions with the highest probability has been selected. The motions and rotations of the floating platform, due to their practicality in physical monitoring, have been the main focus of the thesis in terms of input variables. Whereas the output variables subject of study are axial tension and bending moments of the pipe at the TDZ, representing the point of which we want to assess the Fatigue Damage. The time series of such variables are plot in Figure 3.53.63.73.8, where we take in consideration a typical sea state scenario, i.e. 488 (see parameters in Table 3.1).

Load Case	*Roll Damping (%)*	*Wave Hs (m)*	*Wave Tp (s)*	*Wave Gamma*	*Wave Direction (deg)*	*Associated Wind Speed (m/s)*	*Wind Direction (deg)*	*Associated Current Speed (m/s)*	*Current Direction (deg)*	*Probability*
488	0.025	1.875	8.5	1.0	60.0	8.53	85.0	0.49	135.0	0.031

Table 3.1: Parameters of simulation based on load case 488.

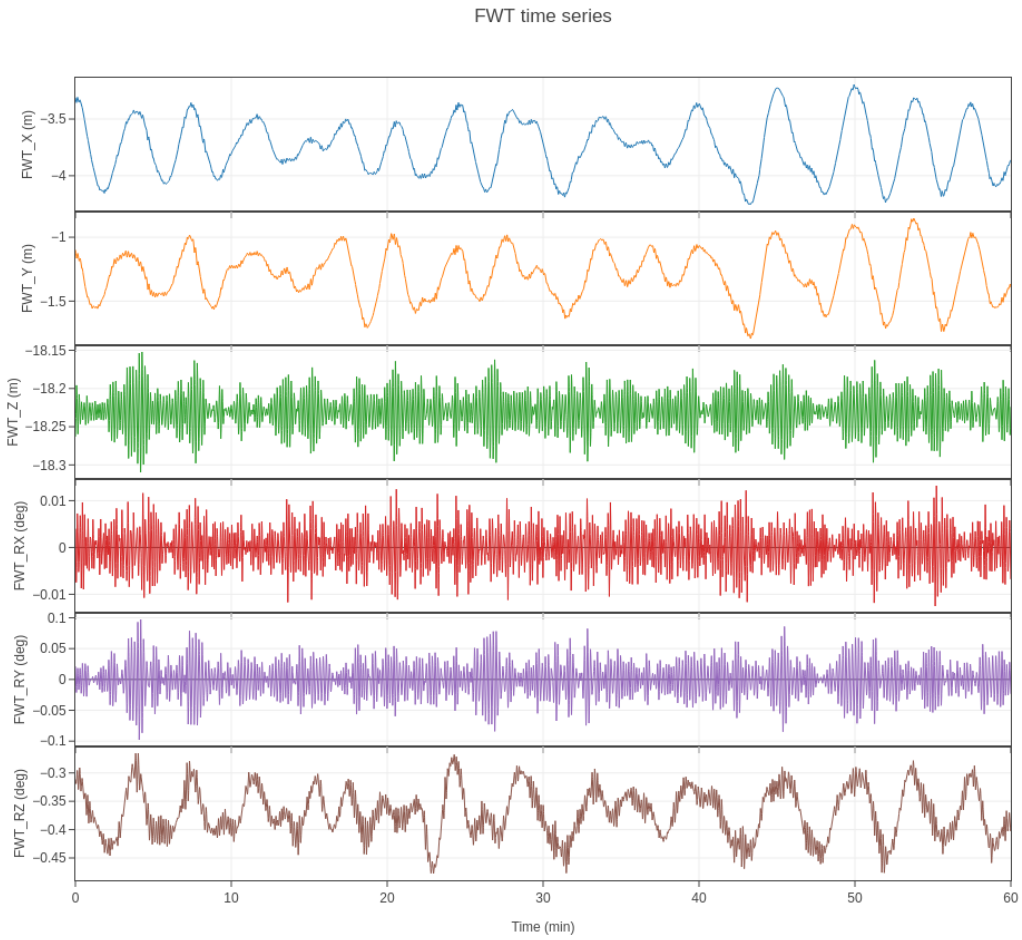


Figure 3.5: Plots of the input time series at FWT point of simulation 488.

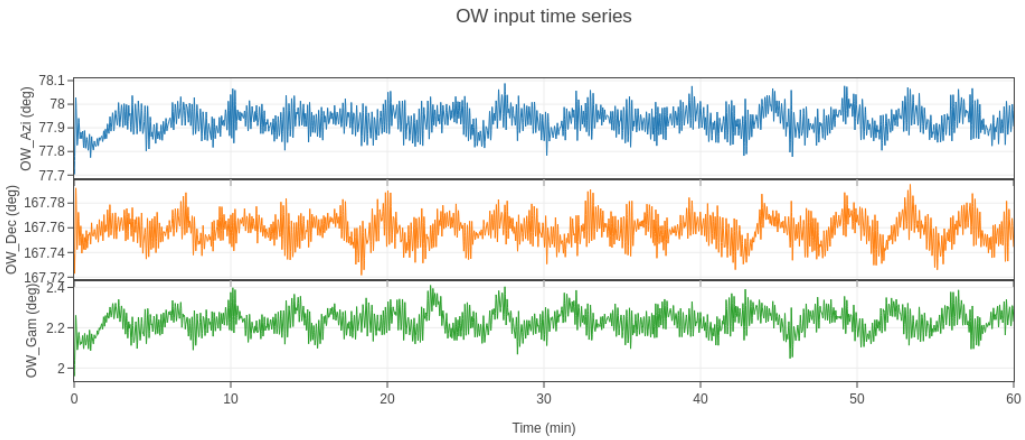


Figure 3.6: Plots of the input time series at OW point of simulation 488.

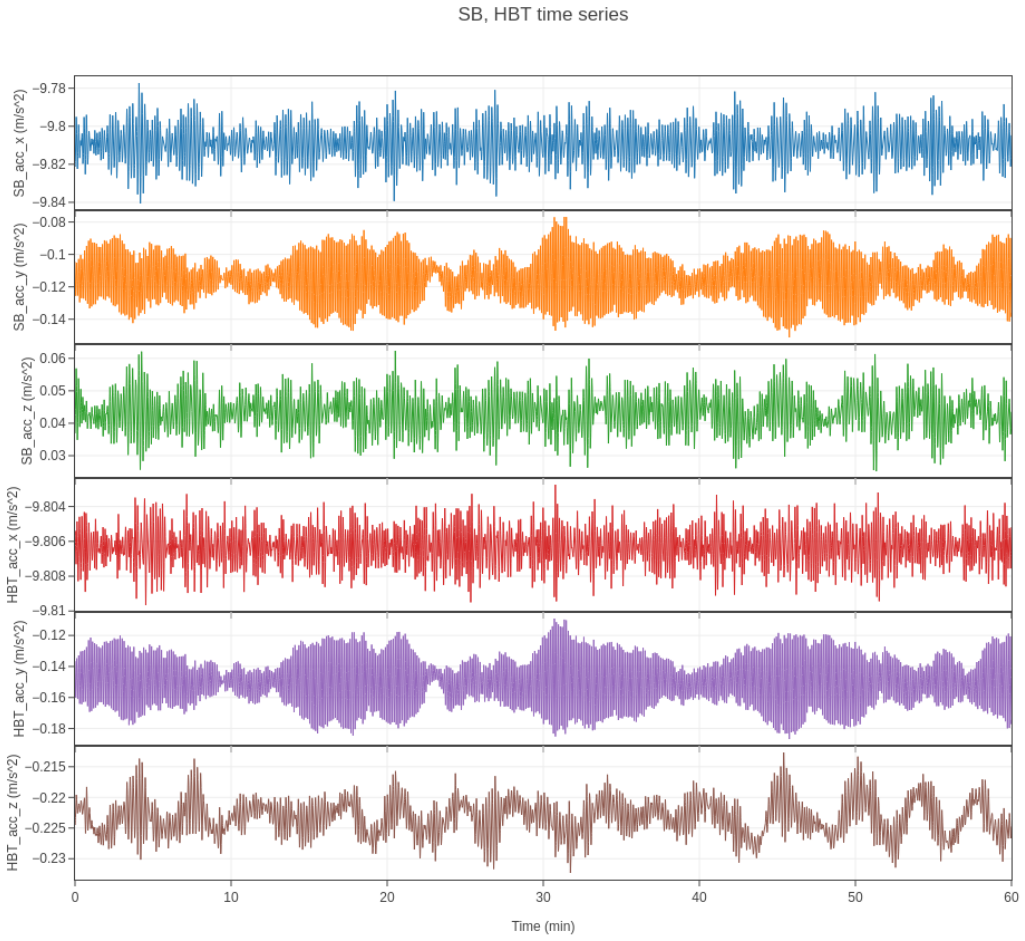


Figure 3.7: Plots of the input time series at SB and HBT points of simulation 488.

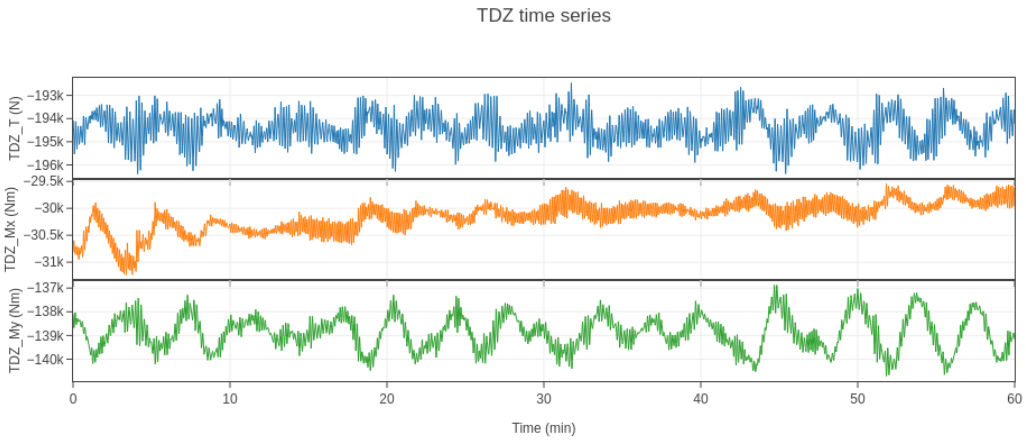


Figure 3.8: Plots of the output time series at TDZ point of simulation 488.

In order to build an exogenous input model for time series forecasting, one important step is to try to evaluate the existence of correlation between the external input variables and the target output variables, and more specifically between the value of an input time series in the present and the whole past sequence of an output time series. However assessing the correlation between variables is not an easy task.

Pearson correlation coefficient can be used to investigate the existence of a linear trend between two distinct variables. The values of this coefficient range between 1 and -1, with 1 indicating perfect linear relationship with positive slope, -1 indicating perfect linear relationship with negative slope, 0 indicating no linear association between the variable. The results of the computation of this coefficient for all input and output variables of simulation 488 are shown in Figure 3.9. The coefficients captures only linear correlations between two single variables, but some nonlinear relations may also be present. Furthermore, in some cases a correlation function may exists between a group of variables and another group of variables; the existence of such function cannot be easily assessed to the knowledge of the author.

	FWT_X	FWT_Y	FWT_Z	FWT_RX	FWT_RY	FWT_RZ	OW_Azi	OW_Dec	OW_Gam	SB_accX	SB_accY	SB_accZ	HBT_accX	HBT_accY	HBT_accZ
OW_T	0.00	0.03	-0.96	0.30	0.87	0.06	-0.11	-0.04	-0.06	-0.96	-0.02	-0.90	0.22	-0.02	-0.60
OW_Mx	0.05	0.04	-0.16	0.07	0.30	0.16	0.37	0.33	0.25	-0.20	-0.16	-0.15	0.08	-0.15	-0.05
OW_My	0.07	0.08	-0.10	0.25	0.05	0.05	-0.37	-0.21	-0.36	-0.07	0.12	-0.14	0.16	0.12	-0.05
HB_Mx	-0.05	-0.05	-0.03	0.01	0.03	-0.12	-0.07	-0.08	0.04	-0.05	0.95	-0.02	-0.21	0.97	-0.03
HB_T	-0.21	-0.19	-0.88	0.12	0.79	-0.10	-0.25	-0.25	-0.07	-0.93	0.01	-0.83	0.21	-0.00	-0.79
HB_My	0.10	0.13	-0.32	0.62	0.30	0.07	0.18	0.43	0.14	-0.11	0.06	-0.48	0.15	0.05	0.24
TDZ_T	-0.59	-0.57	-0.62	-0.03	0.54	-0.42	-0.33	-0.39	0.07	-0.69	0.07	-0.66	0.13	0.03	-0.95
TDZ_Mx	-0.39	-0.34	0.01	-0.03	0.02	-0.28	0.01	-0.09	0.21	-0.02	-0.29	-0.09	0.06	-0.32	-0.29
TDZ_My	0.86	0.92	-0.17	0.20	0.14	0.67	0.36	0.48	-0.15	-0.11	-0.11	0.02	0.04	-0.06	0.57

Figure 3.9: Heatmap of Pearson correlation values between input variables (columns) and output variables (rows) for simulation 488.

3.2 Extracting features

The previous Section provides some insights useful for all simulations, by focusing in studying the data related to a single simulation. However how much similar are the various simulations between each other? An important phase of the data exploration performed consisted in studying the differences "caused" in simulations data by the distinct sea conditions. In order to analyze the differences across the consistent amount of simulations available we decided to extract some global features potentially capable to capture the characteristics of each single simulation.

This Chapter will consider in particular two types of features: the sea state parameters and some statistical descriptors (e.g. mean and standard deviation) of the univariate time series associated to each variable in a simulation. The sea state parameters are *initially* provided to Orcaflex to describe the environment conditions in which a simulation has to be run. They are *a priori* constraints of the simulations, providing important global characteristics. On the other hand, the statistical descriptors are extracted *a posteriori* from the single time series associated to each variable, representing the characteristics that the data produced in given sea conditions manifested in the simulation.

We first focused on sea state parameters, considering the scatter plots of some relevant parameters against the others to visualize possible correlation patterns in the data. By visually inspecting the several plots in Figure 3.10, we can visually notice some interesting trends in the

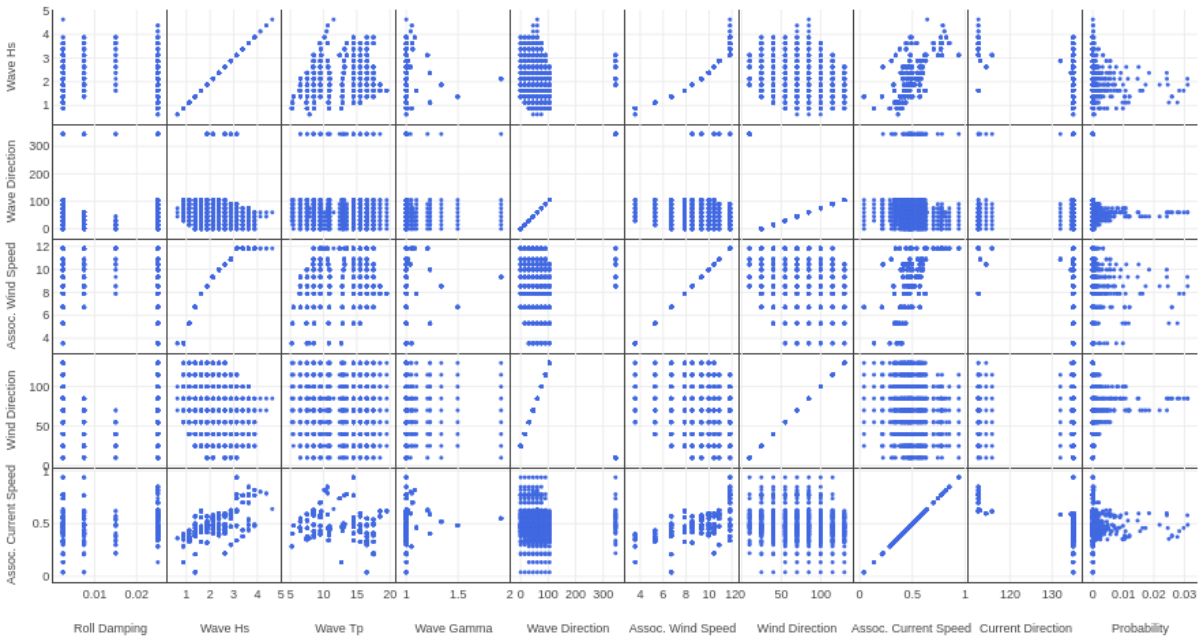


Figure 3.10: Scatter plots between some relevant parameters of simulations to highlight possible correlations.

following graphics:

1. wave height / wind speed: the points are mainly distributed on a curve showing a possible quadratic relationship, except for the last ones;

2. wave direction / wind direction: these plots show a clear linear trend; (the apparent outlier is due to the circularity of the wave direction axe, that cannot be captured with this type of plot);

3. wave height / current speed: the points seems to be scattered around a linear trend.

These plots have to be interpreted according to the fact that initial parameters of the simulations have been chosen artificially in order to describe the most common scenarios of the specific case study and of the phenomena involved. From their inspection it is possible to see that in our scenarios wave height grows with wind speed, and wave and wind direction are correlated as well. Current speed shows a less evident correlation with wave height, probably related to more superficial currents.

Next, a set of global features has been extracted from the time series of each simulation and compared to provide some intuitions. The choice of the features converged first to the mean (AVG), standard deviation (STD) and energy (E), as main descriptors of the time series in the time domain. Additionally, the computation of the Fourier Transforms of the time series allowed to include other features in the frequency domain such as the frequency with the highest peak in amplitude (FPeak) and the relative amplitude, the crest factor (FCrest) and the spectral energy density (SED), which is the energy of the Fourier Transform). Peaks in time domain could not represent really good indicators since in a long time series (1 hour with 0.25s sampling period), they could be subject to noise and there could be several. Dealing with long time series, shape-based metrics have been excluded as well since more fitted to short time windows (see the literature review of time series clustering from S. Aghabozorgi et al. [12] for a wider overview on assessing time series similarity or dissimilarity).

In total, 6 features have been computed for each of the 24 variables in a simulation's data, leading to a flattened feature vector of length 144: 90 features for input variables and 54 features for output variables. The high dimensionality of the feature vector makes it unfeasible to plot a 90*54 correlation matrix composed of scatter plots or even just a heatmap of Pearson coefficient to find linear trends. Therefore we first focused the analysis to a subset of variables, and started by selecting rows and columns of the Pearson coefficients heatmap with at least one cell with absolute value above a fixed threshold. This choice highlights stronger linear trends in data.

In this phase of the analysis we focused on the Touch Down Zone node. High Pearson's coefficients (above 0.85 in absolute value) between TDZ variables' features and input variables' features are shown in Figures 3.11, 3.12, 3.13,3.14. In particular, considering each input node:

1. at position FWT (Figure 3.11), the displacement on Z axis (in terms of STD, E -energy- and SED) shows high correlation with the tension at TDZ (in terms of STD and SED), whereas the bending moments at TDZ (STD and SED) have high correlation with both displacement Z (AVG, STD, SED) and rotation X (STD, E, SED);

	FWT_Z_AVG	FWT_Z_STD	FWT_Z_E	FWT_Z_SED	FWT_RX_STD	FWT_RX_E	FWT_RX_SED
TDZ_T_STD	0.56	0.95	0.90	0.93	0.60	0.57	0.57
TDZ_T_SED	0.49	0.85	0.95	0.94	0.49	0.50	0.50
TDZ_Mx_STD	0.72	0.87	0.69	0.81	0.74	0.73	0.73
TDZ_My_STD	0.84	0.89	0.64	0.80	0.88	0.84	0.84
TDZ_My_SED	0.88	0.83	0.67	0.84	0.85	0.90	0.90

Figure 3.11

2. at position OW (Figure 3.12), the declination (STD and SED) has high linear correlation coefficients with all the tension and bending moments (STD and SED) of TDZ. Moreover AVG and E of declination are strongly correlated with AVG and E of bending moment at y-axes;

	OW_Dec_AVG	OW_Dec_STD	OW_Dec_E	OW_Dec_SED
TDZ_T_STD	-0.77	0.95	-0.77	0.94
TDZ_T_SED	-0.68	0.84	-0.68	0.94
TDZ_Mx_STD	-0.68	0.92	-0.68	0.86
TDZ_Mx_SED	-0.59	0.84	-0.59	0.88
TDZ_My_AVG	0.93	-0.52	0.93	-0.47
TDZ_My_STD	-0.50	0.91	-0.50	0.83
TDZ_My_E	-0.94	0.53	-0.94	0.48

Figure 3.12

3. at position SB (Figure 3.13), notice in particular the coefficients for STD and SED of acceleration along x and z axis against STD and SED of tension and bending moments at TDZ. The axial tension at TDZ has more features with high correlation coefficients, especially considering the acceleration of SB on z axis;

	SB_accX_STD	SB_accX_SED	SB_accZ_AVG	SB_accZ_STD	SB_accZ_E	SB_accZ_SED
TDZ_T_AVG	0.36	0.34	-0.94	0.41	0.21	0.38
TDZ_T_STD	0.98	0.97	-0.62	0.99	0.78	0.95
TDZ_T_E	-0.34	-0.32	0.93	-0.40	-0.18	-0.37
TDZ_T_SED	0.88	0.97	-0.55	0.91	0.88	0.98
TDZ_Mx_STD	0.88	0.81	-0.58	0.84	0.56	0.75
TDZ_My_STD	0.86	0.78	-0.45	0.80	0.47	0.70

Figure 3.13

4. at position HBT (Figure 3.14), in general the same observations of SB are valid. Notice in particular the perfect (with 2 digits approximation) correlation coefficients of AVG and E of tension at TDZ against acceleration on z axis at HBT.

	HBT_accX_STD	HBT_accX_SED	HBT_accZ_AVG	HBT_accZ_STD	HBT_accZ_E	HBT_accZ_FCrest	HBT_accZ_SED
TDZ_T_AVG	0.21	0.16	-1.00	0.28	0.99	-0.10	0.22
TDZ_T_STD	0.84	0.80	-0.43	0.93	0.52	-0.07	0.90
TDZ_T_E	-0.21	-0.15	1.00	-0.27	-0.99	0.10	-0.21
TDZ_T_SED	0.75	0.79	-0.38	0.83	0.48	-0.12	0.90
TDZ_Mx_STD	0.88	0.81	-0.43	0.91	0.51	0.17	0.84
TDZ_Mx_SED	0.82	0.85	-0.35	0.85	0.44	0.11	0.88
TDZ_My_STD	0.97	0.90	-0.36	0.95	0.43	0.32	0.87
TDZ_My_FCrest	0.30	0.21	-0.01	0.19	0.00	0.88	0.10
TDZ_My_SED	0.93	0.97	-0.27	0.90	0.35	0.25	0.92

Figure 3.14

To sum up, the standard deviation and spectral energy density, descriptors respectively of the variations in a signal and of the energy of its frequencies amplitudes, are the output features showing the highest Pearson correlation coefficients across variables.

Some other general observations that we can extract from these coefficients is that the axial tension at TDZ has stronger linear correlation than the bending moments to other input variables (with the bending moment on x-axis being the least correlated), and the input variables associated to measures at z and x axis, together with the declination at OW, are the most relevant input

variables. Furthermore, not surprisingly, the correlations of TDZ measures to input variables appears stronger as the considered nodes get closer to the TDZ.

Next, we tried to understand which sea state parameters have a direct impact on the output variables by searching for correlations between such parameters and the previously extracted features of output variables. In order to investigate these correlations, Pearson's coefficients and scatter plots have been exploited as previously done to focus the attention on some parameters by setting a threshold on correlation values of 0.7. We can see the resulting heatmap and scatter plots in Figures 3.15, 3.16, 3.17.

	Wave Hs(m)	Wave Tp (s)	Associated Wind Speed (m/s)	Associated Current Speed (m/s)	Current Direction (deg)
OW_T_AVG	0.35	-0.06	0.23	0.23	-0.78
OW_T_E	0.46	0.04	0.33	0.26	-0.73
OW_T_FPeak	-0.22	-0.85	-0.26	0.09	-0.01
OW_Mx_AVG	-0.10	0.72	-0.07	-0.57	0.55
OW_Mx_STD	0.78	0.61	0.76	0.38	-0.33
OW_Mx_E	0.84	0.24	0.76	0.61	-0.57
OW_Mx_FPeak	-0.16	-0.79	-0.17	0.12	-0.03
OW_Mx_SED	0.77	0.50	0.70	0.39	-0.33
OW_My_FPeak	-0.25	-0.92	-0.27	0.05	-0.02
OW_My_FCrest	0.11	0.80	0.14	-0.19	0.11
HB_T_AVG	0.59	-0.07	0.50	0.58	-0.76
HB_T_E	0.63	0.01	0.53	0.56	-0.71
HB_Mx_AVG	0.61	-0.03	0.54	0.90	-0.81
HB_Mx_E	0.57	0.01	0.48	0.80	-0.75
HB_My_AVG	-0.55	0.13	-0.56	-0.78	0.27
HB_My_E	-0.51	0.18	-0.52	-0.79	0.27
HB_My_FPeak	-0.24	-0.84	-0.27	0.08	-0.02
TDZ_T_AVG	0.39	-0.03	0.27	0.25	-0.79
TDZ_T_E	-0.38	0.03	-0.26	-0.23	0.78
TDZ_My_AVG	-0.62	0.09	-0.63	-0.83	0.37
TDZ_My_E	0.62	-0.08	0.62	0.83	-0.36

Figure 3.15: Heatmap of Pearson's correlation coefficients of output variables' features with sea state parameters (only rows and columns where at least one cell has absolute value above 0.85).

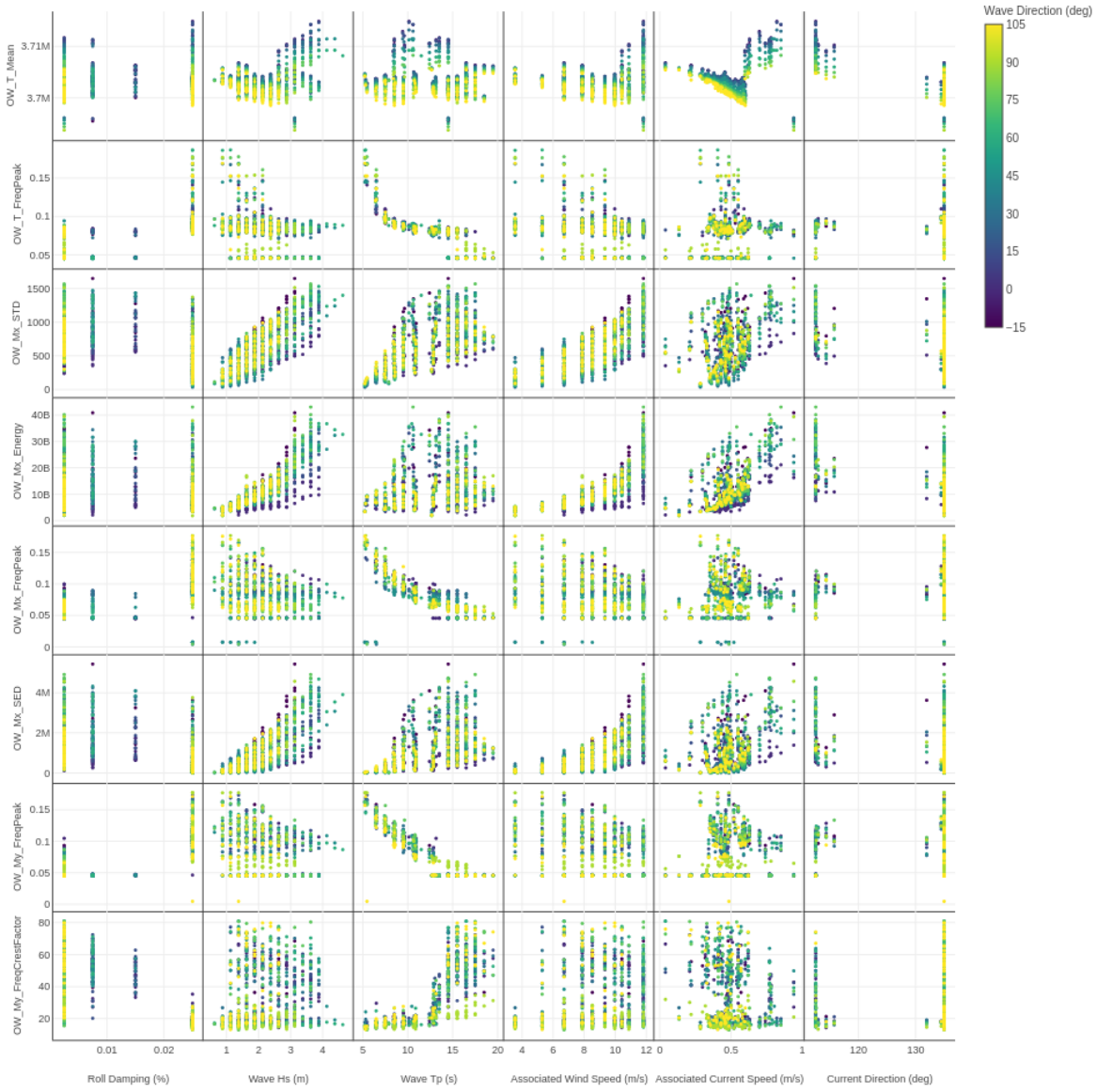


Figure 3.16: Scatter plots of OW variables' features against sea state parameters. Scatter points are colored according to the wave direction of the corresponding simulation.

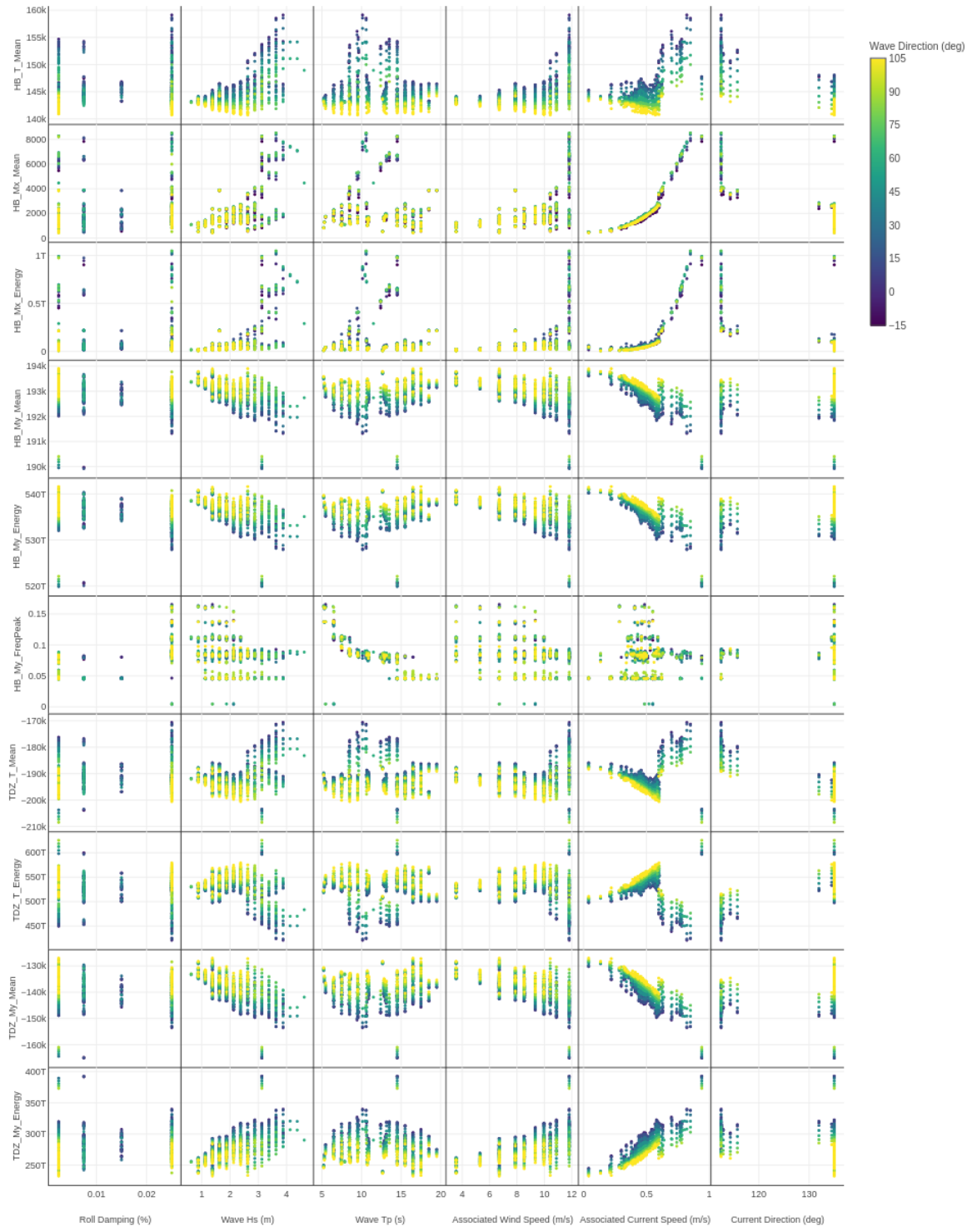


Figure 3.17: Scatter plots of HB and TDZ variables' features against sea state parameters. Scatter points are colored according to the wave direction of the corresponding simulation.

Observe in particular the strong connection between the frequency peak of bending moment at y-axis and the wave period at OW (refer to Figure 3.15 and 3.16), and between the mean of bending moment at x-axis and the current speed at HB (refer to Figure 3.15 and 3.17). In general, from Figure 3.15 we can notice that for superficial positions (OW) the Pearson's correlation coefficients are higher (in absolute value) with wave and wind parameters, related to the surface of the sea, whereas for deeper positions (HB and TDZ), the correlation coefficients are higher for current parameters, related to phenomena happening under the surface of the sea. This could also explain the lower number of features showing high correlations if considering deeper positions: the current direction and speed are not directly captured by any time histories, thus they can have an impact on tension and bending moments at deeper positions that is not related to e.g. the motions of the FWT.

Last, notice also that by coloring the scatter points according to the wave direction, some clearer trends in data become visible in some plots, suggesting that wave direction may have an important impact in clustering sea states' characteristics of output time series.

3.3 Fatigue Damage computation

A conclusive part of the data exploration involved the Fatigue Damage. In the previous Chapter we defined the stress function employed in our study case in Equation 2.16. In particular, our pipe is characterized by the parameters listed in Table 3.2.

Area	$\sim$0.0372
ID	0.24
Inertia	$\sim$0.000378

Table 3.2: Fixed pipe parameters.

Applying the given numerical values to the fixed parameters, we obtain an approximated function 3.1, where the axial tension is multiplied by a factor one order of magnitude smaller than the factor multiplying the bending moments, indicating that the bending moments have a

higher impact on the stress of a section.

$$\sigma(\alpha) = 27 \cdot Tw + 428 \cdot (Mx \cdot \sin\alpha - My \cdot \cos\alpha) \tag{3.1}$$

Since our stress function 2.16 is parametric in the angle, we define the fatigue damage in a critical area of the pipe as the maximum fatigue damage across the damages of the stresses at various angles of the circular section.

Interestingly, the data shows that for TDZ and HB, the highest damage occurs at 0° or 180° in the vast majority of the simulations, while the same is not valid for OW (see Table 3.3).

Angle of max. Fatigue Damage	*Count of load cases*		
	OW	*HB*	*TDZ*
0°	209	574	369
45°	102	6	6
90°	279	4	1
135°	60	3	1
180°	103	185	429
225°	29	17	2
270°	25	19	0
315°	1	0	0

Table 3.3: Table that shows the number of sea state scenarios for which the Fatigue Damage is maximum at a given angle, for the critical points. The values of Fatigue Damage are computed using the entire simulations.

Focusing on TDZ, we can also observe the interesting relation between the standard deviation of the bending moment in the y-axis and the fatigue damage of the stress at angle 0° computed for a simulation (see Figure 3.18). This relation appears to be exponential, as it becomes more evident by considering logarithmic axes (see Figure 3.19). This results can be explained if we consider that the fatigue damage depends on the amplitude of the stress ranges inside the given stress time histories, and that the standard deviation of such time histories also represent an indicator of the amplitude of stress ranges. Notice that we are here considering the bending moment time histories and not the stress time history, however they are likely to be very similar (up to scaling) according to Equation 3.1.

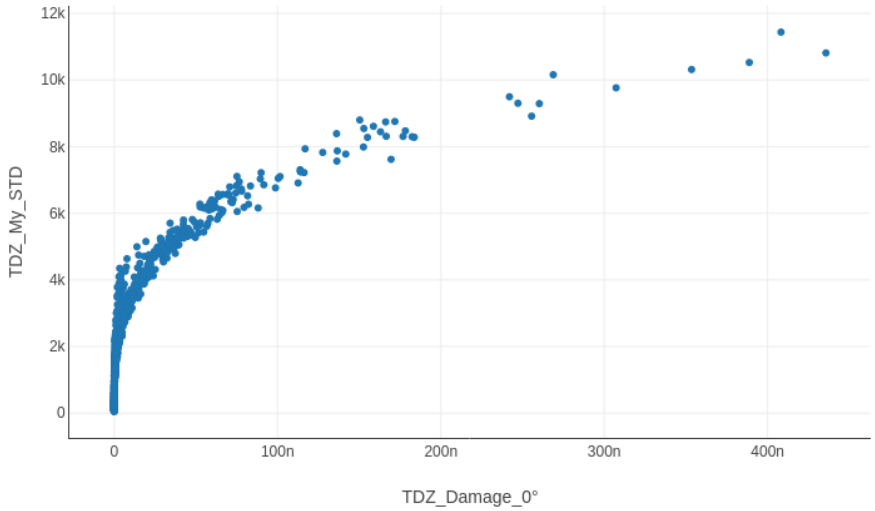


Figure 3.18: Scatter plots of the standard deviation of the bending moment time histories VS the fatigue damage at angle 0° of the stress time histories.

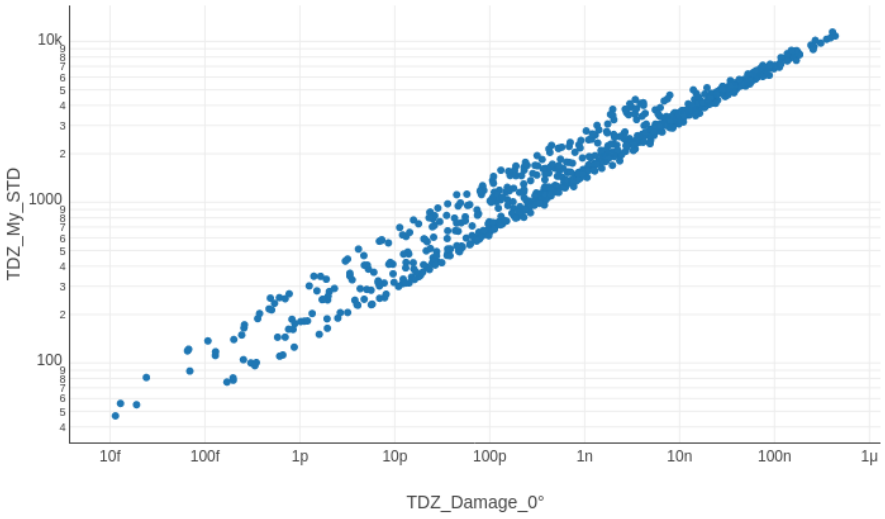


Figure 3.19: Scatter plots of the standard deviation of the bending moment time histories VS the fatigue damage at angle 0° of the stress time histories, on logarithmic axes.

4

Time Series Forecasting

The work of time series forecasting of this thesis relied on LSTM Neural Networks, which proved to perform well for the task [5]. M3E already fine tuned LSTM Neural Networks using data from a single simulation for different combinations of input and output variables.

The strategy adopted for feeding data to the network during training and test is the Sliding Window approach, illustrated in Chapter 2. In particular, the NN is based on a many-to-one strategy, receiving in input a window of length w, that contains values of input variables from time t to $t + w$, and producing in output a single value of the target variables at time $t + w + 1$.

We explored the time series forecasting problem applied to different sea state scenarios. In order to reduce the size of the problem, we focused in particular on a subset of all simulations and on a single node (TDZ) for the output variables.

The data available consisted of a set of 808 1-hour-long multivariate time series, each associated to a specific sea state. Data are not necessarily homogeneous between different simulation, as we investigated in Chapter 3 at Section 3.2, since they are impacted by the differences in the

sea states. Consequently, a NN trained on a specific sea state is able to generalize to another sea state data differently depending on how much the differences in the sea conditions impact the considered time series.

The Neural Networks presented in this Chapter have been implemented in PyTorch and trained on a Nvidia GTX 1060 3GB GPU.

The Chapter starts by analysing the results of training with a given architecture on a single simulation, providing a direction (better defining the task/problem) to extend the study to multiple simulations. The following Section discusses the results of trainings performed using the same architecture and hyperparameters across different simulations. Last, a bigger network is trained on the previously chosen set of simulations to compare results.

4.1 Definition of the learning task

First, we performed some trainings oriented at identifying the specific problem we want to tackle in terms of input and output variables. In this phase we trained a Neural Network just on a single simulation (488), using input samples coming from a sliding window of 40 time steps, for a total of 10s, and different combinations for input and output variables (focusing always on TDZ node). For this purpose, we implemented a LSTM Neural Network with a Dense layer in output. Details about the architecture and the hyperparameters are summarized in Table 4.1.

Number of LSTM layers	1
LSTM Layer Size	32
Number of Dense layers	1
Dense layer size	32
Batch size	32
Optimizer	Adam
Learning rate	0.005
Epsilon	10^{-7}

Table 4.1: Architecture, hyperparameters and optimizer details.

4.1.1 Data preparation

The simulation's data has been split into subsets of 70% (10080 time steps equivalent to 42 minutes) for training, 15% for validation (2160 time steps equivalent to 9 minutes) and 15% for testing (2160 time steps equivalent to 9 minutes). The split has been performed on the original ordered samples extracted from the time series, without previous shuffling, allowing for a later visual comparison between true time series and predicted time series. On the other hand, we applied data shuffling just to samples used for training, in order to improve the model's ability to generalize.

As a common preprocessing phase, each input and output variables training data have been normalized with a MinMax approach, scaling it in the range [0,1]. Scalers are stored to apply the same preprocessing to input data fed to the trained network and to denormalize (post-processing) the produced output data.

4.1.2 Results

The Neural Network defined by Table 4.1 have been trained with inputs node FWT or all the available input nodes, and evaluated on the test set (using data from sea state 488).

The two NNs configurations lead to different sizes: the first network is composed of 5.1k parameters while the second network counts 6.3k parameters.

The results are provided in Table 4.2.

	Input FWT node		*Input all nodes*	
	R2	*ERR*	*R2*	*ERR*
TDZ_T	0.927		0.990	
TDZ_Mx	−8.756		−4.157	
TDZ_My	0.958		0.958	
Stress_0°	0.959	9.02%	0.959	6.34%

Table 4.2: Results of the NNs trained on sea state 488 with output variables T, Mx and My at TDZ position.

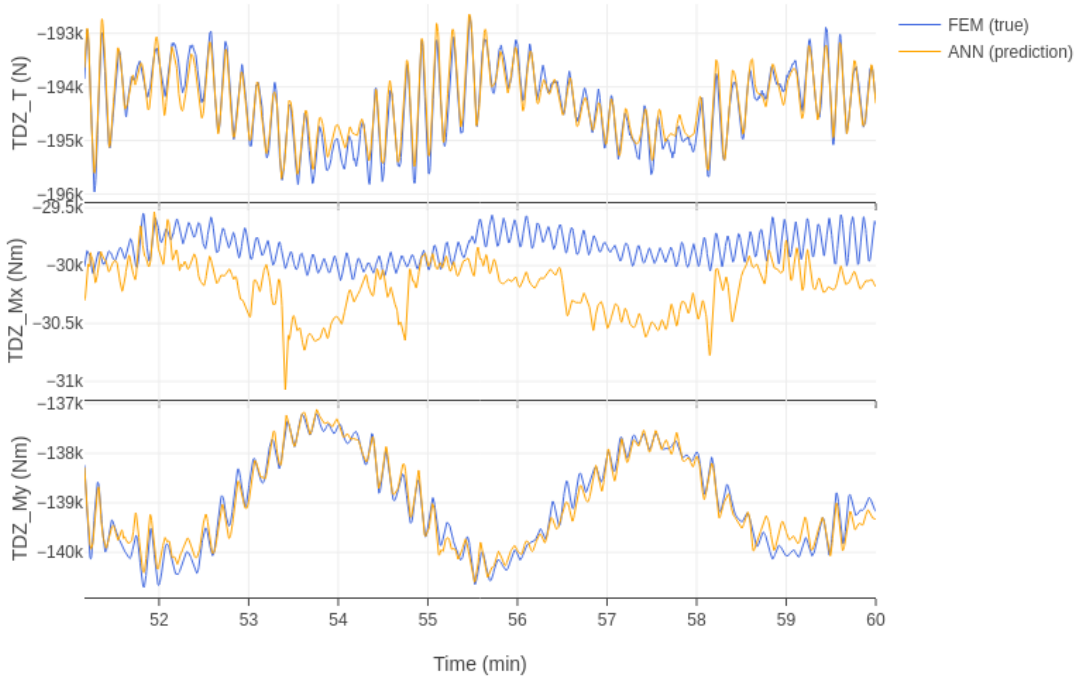
Notice that the performances in terms of R2 are acceptable (R2 scores usually around 0.95 or above) for the axial tension and the bending moment at y-axis, but the same does not hold for the bending moment at x-axis. This is particularly visible if we consider the plots of the predictions on test data (Figure 4.1). These results suggest that the bending moment at x-axis is not sufficiently correlated with the input data in order to allow for a successful learning. However, this is not impacting the accuracy on the stress time series, since the stress at angle 0° (providing the highest damage among all angles) depends only on axial tension and bending moment at y-axis.

This consideration led us to focus on output variables T and My, which are sufficient to compute the stress at angle 0° and 180°. Such choice does not represent an important limitation since, at TDZ, the stress at angle 0° and 180° are the highest among all angles for the vast majority of simulations, as shown in Chapter 3 at Section 3.3. The new results are provided in Table 4.3. Comparing Table 4.2 and 4.3, we can observe in the latter an increased quality of

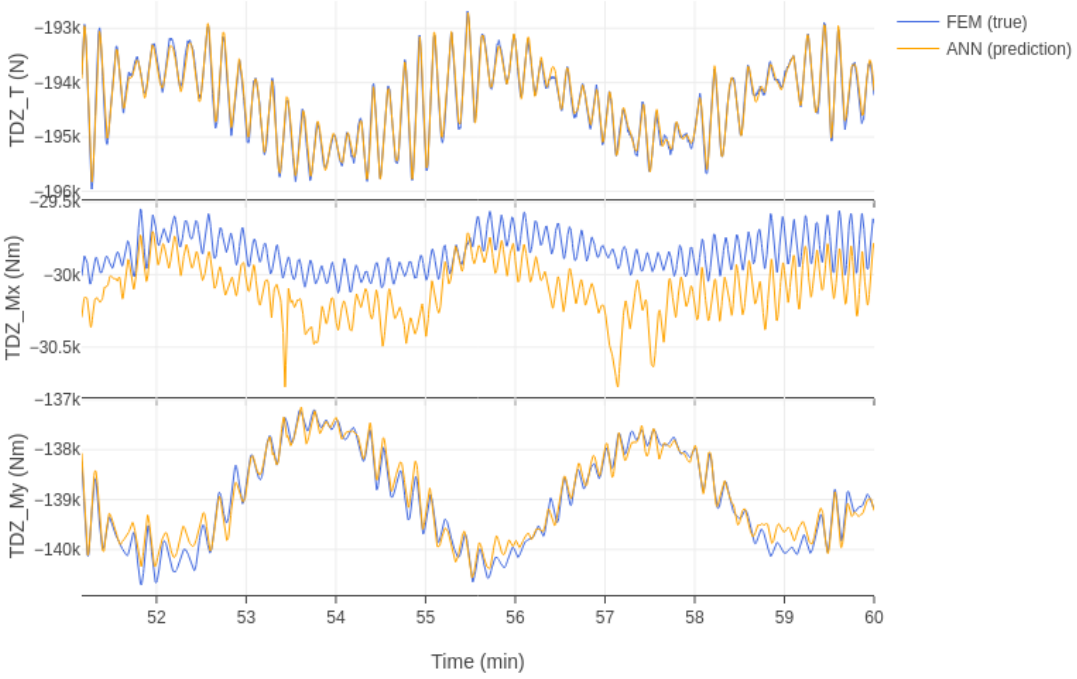
	Input FWT node		*Input all nodes*	
	R2	*ERR*	*R2*	*ERR*
TDZ_T	0.973		0.995	
TDZ_My	0.963		0.975	
Stress_0°	0.964	3.81%	0.976	11.9 %

Table 4.3: Results of the NNs trained on sea state 488 with output variables T and My at TDZ position.

prediction, in terms of R2 scores, for both T and My. We can explain this results with the fact



(a) Training performed with input variables from FWT node.



(b) Training performed with all input variables available.

Figure 4.1: Visualization of original against predicted output time series for two different NN configurations. Trainings are performed in both cases on data from sea state 488 with output variables T and My at TDZ position.

that the high loss due to Mx forced the optimization procedure to focus on fitting the model to the prediction of Mx much more than on the prediction of T and My. Observe also that in the Table 4.3, considering all input nodes shows more visible benefits in terms of R2, with around 2% higher R2 scores for T and 1% higher R2 scores for My. In both Tables, the R2 scores on the stress are very close to the R2 scores on the bending moment at y-axis, which can be explained considering the higher weight of bending moments than tension in the stress function (refer to Equation 3.1) and the fact that, in this simulation, the magnitude of the values of tension and bending moment at y-axis is comparable (see Figure 4.1).

Last, the relative error on the fatigue damage associated to the stress time series at angle 0° does not seem to be correlated to the R2 scores obtained on the stress time series.

In the next Section, we will extend the analysis to a subset of simulations and evaluate the obtained results.

4.2 Multiple sea states scenario

In order to extend the forecasting to multiple sea state scenarios, we followed two approaches: the first approach involved performing several trainings of the NN validated in the previous Section, one for each given simulation; the second approach involved training a bigger NN on the whole set of simulations. The factors determining the success of the two strategies are how much the simulations are heterogeneous and which generalization capabilities (among different sea states) a single model can achieve.

Notice that, in a real application, the first approach has some major limitations. First, such strategy requires to know which sea state is present in a given moment in order to choose the appropriate NN to perform the time series forecasting: additional sensors or a sea state classification algorithm are required for this purpose, introducing another layer of errors. A related problem is the choice of the frequency or policy to adopt to update the sea state and consequently the NN in use: a frequent update would reduce the time in which a potentially not appropriate

NN is in use (provoking unrealiable results), but it would also increase the number of switches from one NN to another, and the consequent number of eventual discontinuities in the predicted time histories. The latter problem, as we will see in the following Chapter, could have an important negative impact on the final Fatigue Damage estimation. These problem are overcome with the second approach, that is designed to be able to generalize over different sea state scenarios. For this reason, our main focus will be to be able to obtain results with the second approach that are comparable or better to the results of the first approach.

For robustness of the results, the Neural Networks have been trained on 10 different seeds: this allowed often to exclude the hypothesis that a specific result was due to the random initialization of weights of the network or the shuffing of data.

4.2.1 Selection of a subset of simulations

Performing our tests on all the available simulations allows to obtain the most extensive results, but is expensive in terms of time and computational resources required. For this reason we decided to focus on a subset of simulations representative of the whole dataset.

Our choice consisted in taking all the simulations with the most recurrent wave height value, i.e. 1.875 m. This subset contains 99 of the 808 considered sea states. The distribution of sea state parameters and Fatigue Damages for the considered subset of simulations in comparison to the whole dataset, is shown in Figure 4.2.

Notice that a consequence of not considering all scenarios is the smaller size of the problem that the NN has to solve when trained simultaneously on all simulations. Furthermore, the selected simulations could lack the ability to represent all scenarios' data, making the learning task less relevant and eventually impossible (depending on how "sparse" the selected training dataset is).

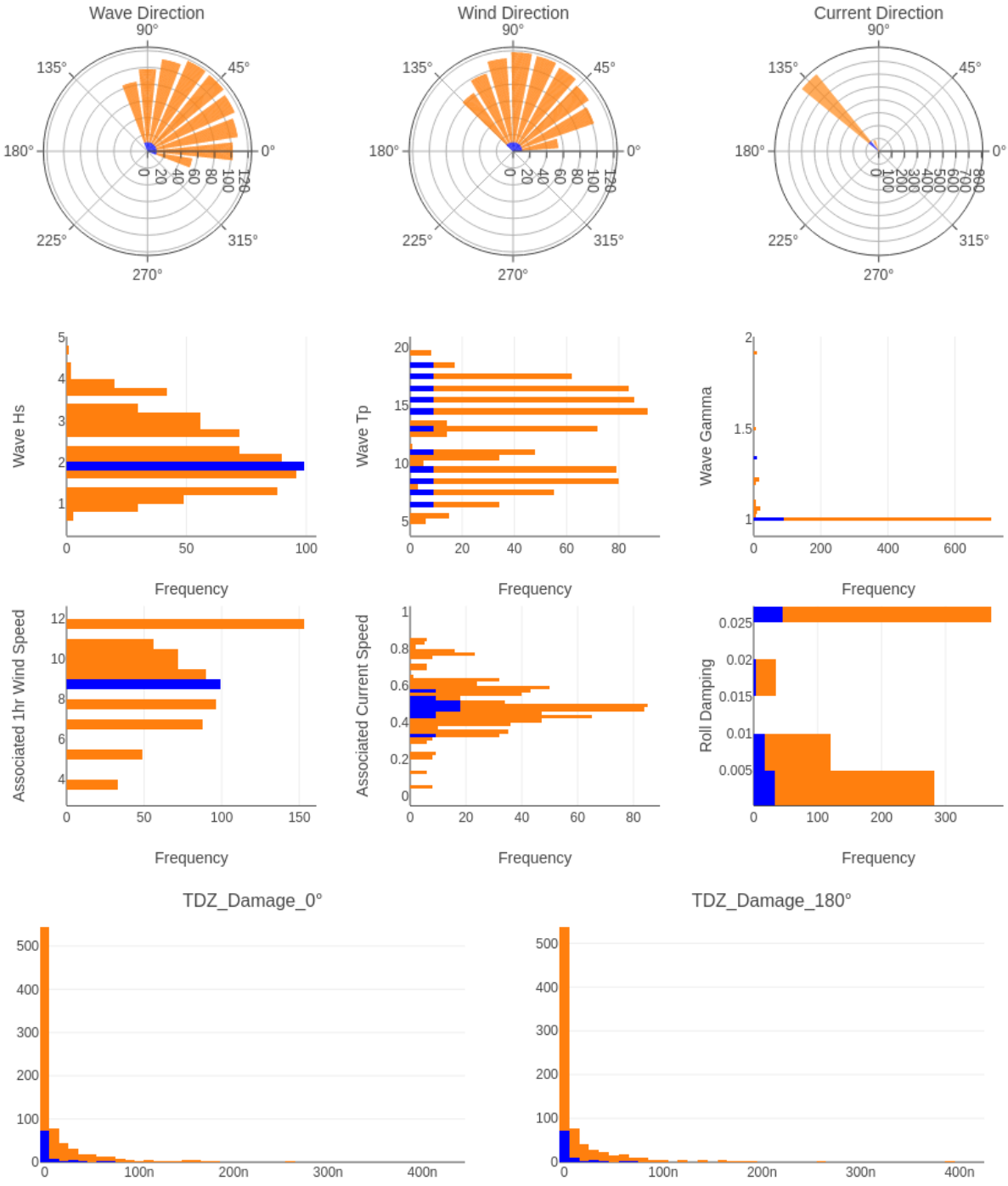


Figure 4.2: Distribution of sea state parameters and Fatigue Damages at TDZ (computed for the entire simulations): in orange the histogram bars of all sea states, while in red the histogram bars of the selection of sea states.

4.2.2 Model validation

For the different training approaches mentioned above, we adopted the same method to split data into training, validation and test sets. In particular, each simulation data have been split into 70% for training, 15% for validation and the remaining 15% for test, following the method used in Section 4.1.1; note that for the second strategy the shuffling of training data has been performed among all the samples of all simulations. The main reason that motivated this choice was to provide a common test dataset for each simulation for the evaluation comparison among the two training strategies, while simultaneously being the most natural choice for the first training strategy, in which a Neural Network was optimized for each distinct simulation. Additionally, this choice also provides the second strategy with a dataset balanced over all simulations in training, validation and test phases.

The validation procedure has been applied mainly to the second strategy, since the first strategy relies on the previous results from M3E (tuning each NN to each specific sea state is not feasible nor potentially relevant for performances).

4.2.3 A set on Neural Networks: one for each sea state

This approach provides results tuned on each state regardless of how much the data is heterogeneous among different sea states inside our selection. The trainings have been performed using either just the time series from FWT sensor, either all input time series. In this Chapter of the thesis, we are going to focus in particular on the quality of the prediction evaluated on the test data in terms of R2 scores of axial tension T and bending moment My. Additionally, we also consider the evaluation metrics computed on the stress time series. The relative error on fatigue damage is briefly discussed, but a deeper analysis on this metric is left for the next Chapter. The overall results are summarized by taking the mean and standard deviation across all simulations of the median values of the metrics across the different seeds, as shown in Table 4.4. Considering the R-squared metric, notice the high quality of the prediction of the axial tension compared

to the bending moments, and the much more important standard deviation from the mean value. Furthermore, notice the pretty high value of the mean of the relative error and its high standard deviation: note that for our problem we consider acceptable the performances when the relative error is smaller than 10%.

	Mean	*STD*
R2_T	0.965	0.037
R2_My	0.892	0.116
R2_S0	0.890	0.116
R2_S180	0.893	0.115
ERR_0°	14.5 %	13.0 %
ERR_180°	14.1 %	12.0 %

Table 4.4: Results of training.

In order to perform a more detailed analysis, for each considered metric, a synthetic representation of the results is provided by plotting, for each simulation, the distribution of the metric values across the different seeds used for training.

The results of the trainings performed using input variables associated to FWT node only are shown in Figure 4.3. Notice that here we omitted to show the R2 scores of the stress time series at angle 180°, since they are strictly related to the scores of the stress time series at angle 0°. Such relation is due to the formula 3.1, and is further proved by plots 4.4d and 4.4e.

Observe that the axial tension is almost always predicted very well, with scores above 0.9. The forecasts are less accurate on the bending moment on y-axis, with several sea states scores between 0.8 and 0.9, and some below 0.8, justifying the high variation on the mean value that we can see in Table 4.4. This discrepancy in the prediction accuracy between the two variables may indicate that the bending moment requires more information (time series from other input nodes) to be effectively learnt.

Another important consideration is that the relative error on the fatigue damage is pretty high and it is not related to the quality of the prediction of stress in terms of R2 scores (see Figure 4.4b). This problem is quite relevant given our final goal, so as previously mentioned it will be

specifically addressed in the following Chapter.

Figure 4.3: Evaluation of the performances of the NNs on each sea state. For each sea state scenario a boxplot synthesizes the distribution of the results obtained setting up the NN with distinct seeds.

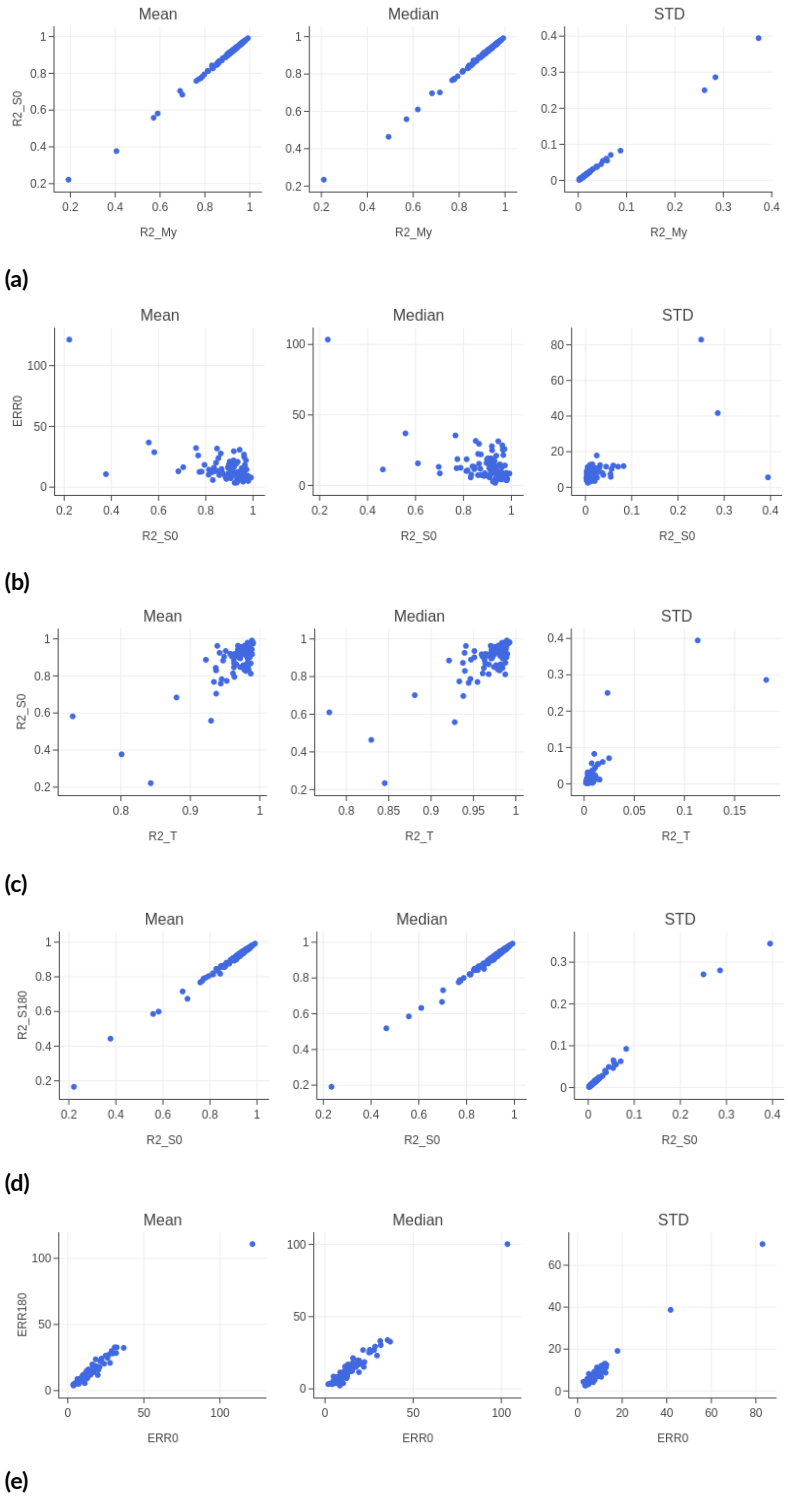


Figure 4.4: Scatter plots of mean, median and standard deviation across seeds of the given evaluation metric, for each simulation.

The same trainings have later been performed using data from all input nodes. This configuration shows a relevant improvement compared to the previous one, as it is visible from Table 4.5 and more in detail in Figure 4.5. Observe that the addition of other input variables allows to remove all the "outliers" in the results, i.e. the sea states that the model using just FWT node as input was not able to successfully learn, especially in terms of bending moment (e.g 74, 77, 270, 591, 686, 691). This suggests that, for some sea states, the input variables excluding the FWT node are crucial. Furthermore, the relative error on Fatigue Damage, while still pretty high, has much smaller mean and standard deviation. Consider however that the improvement is mainly due to absence of strong outliers in this run of trainings, and therefore is not well representative of the performances excluding the outliers (see the relative error on simulation 690 for the first run of trainings).

	Mean	*STD*
R2_T	0.992	0.004
R2_My	0.963	0.030
R2_S0	0.963	0.030
R2_S180	0.963	0.030
ERR_0°	9.1%	4.3%
ERR_180°	8.8%	4.1%

Table 4.5: Results of training.

4.2.4 A single Neural Network for all sea states

This approach is aimed at creating a model capable of generalization across different sea states. For this purpose we adopted a bigger NN, with a total of 24.3k parameters. The architecture differs from the one in the previous paragraph mainly due to increased number of LSTM layers. We decided to use all the available input variables, due to the improvements in performance shown in the previous Section. We tuned batch size and Adam learning rate values and increased the number of epochs in order to adapt to the much bigger dataset in use, leading to 7767 training

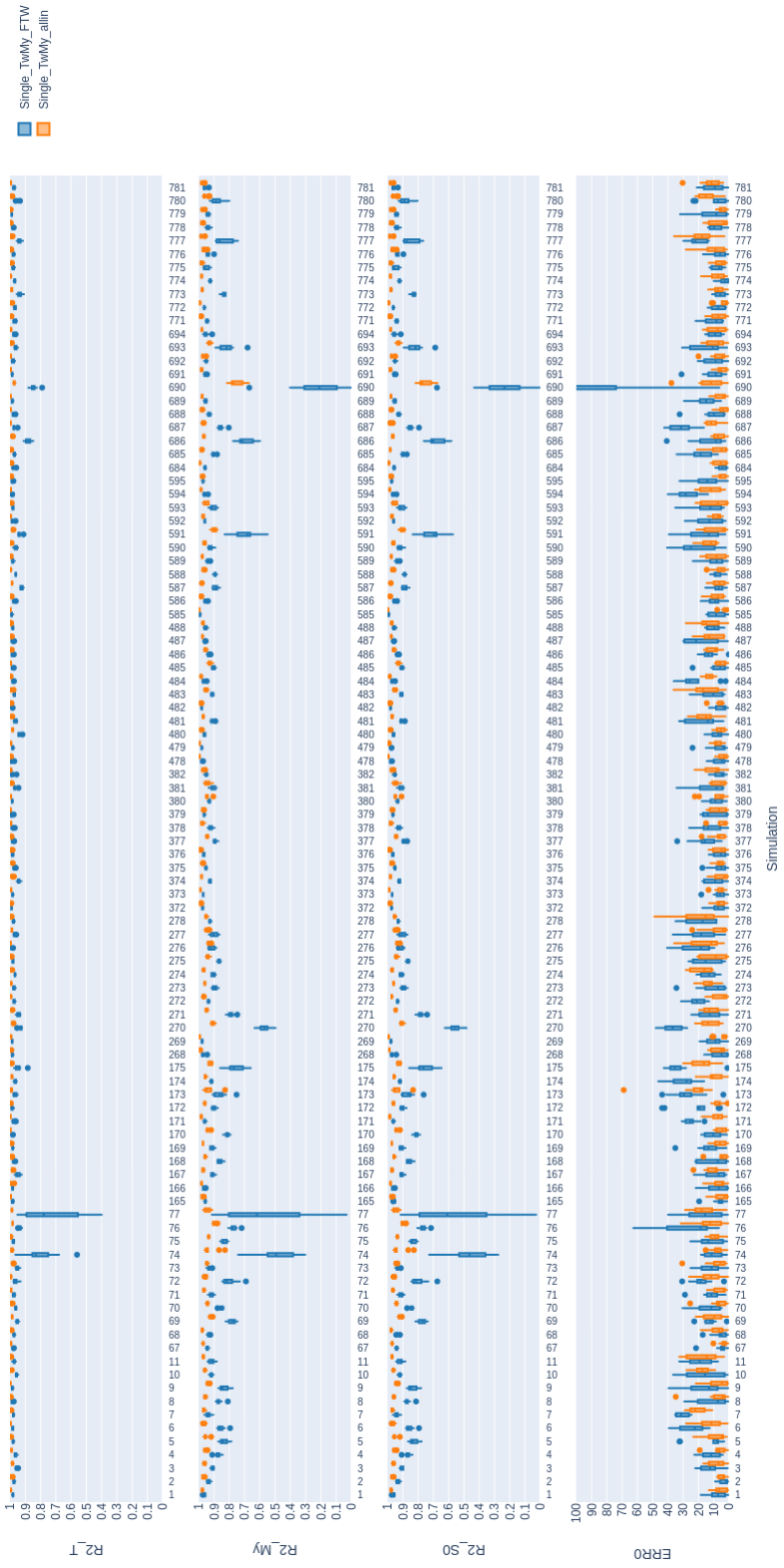


Figure 4.5: Evaluation of the performances of the NNs on each sea state. For each sea state scenario a boxplot synthesizes the distribution of the results obtained setting up the NN with distinct seeds.

batches and 1641 batches for validation and test. We also applied dropout to the LSTM layers for regularization. These choices allowed to achieve more stable train-validation curves (see Figure 4.6). A summary of the hyperparameters and training parameters is provided in Table 4.6.

Number of LSTM layers	3
LSTM layer Size	32
Number of Dense Layers	2
Dense Layer Size	32
Batch Size	128
Optimizer	Adam
Learning Rate	0.001
Epsilon	10^{-7}

Table 4.6: Architecture, hyperparameters and optimizer details.

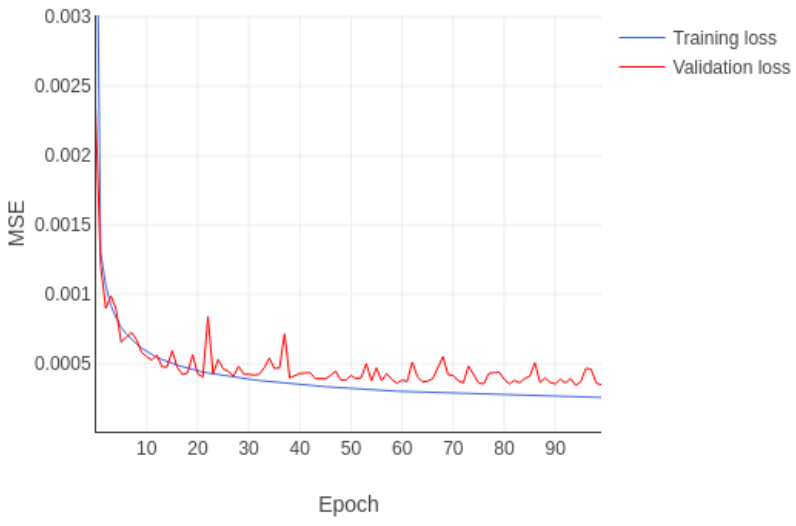


Figure 4.6: Train-validation curves on seed 1234.

Another relevant modification involved the loss function used for training. We are aware of the greater weight of the bending moment with respect to the axial tension in determining the stress function (on which we compute the Fatigue Damage), from Equation 3.1. Moreover, in the previous Section we observed that the axial tension is also easier to learn than the bending moment. For these reasons, we chose to modify the loss function in order to weight more the learning of the bending moment, as follow:

$$loss(y, \hat{y}) = MSE(y_T, \hat{y}_T) + 5 \cdot MSE(y_{My}, \hat{y}_{My}) \tag{4.1}$$

where $y = (y_T, y_{My})$ and the factor 5 accounts for the difference in weight between axial tension and bending moments in Equation 3.1, which is of the order of 10, and proves to perform well in practice (the gap between the R2 scores of axial tension and bending moment is much reduced).

In Figure 4.7 we present the performances evaluation of the model, while Table 4.7 summarizes the mean values across different seeds of the evaluation metrics, averaged through all simulations. Notice the presence of some strong outliers with really low performances in terms of R2 scores, in particular for simulations 591, 690, 692, 693, 694, 777, 778, 779, 780 and 781. We can actually observe that all these simulations (see Figure 4.8) have in common a low standard deviation in the bending moment of the y-axis (again, the main actor in the stress at angle 0° and 180°). In terms of relative error on Fatigue Damage, the considerations relative to the

	Mean	*STD*
R2_T	0.972	0.016
R2_My	0.936	0.018
R2_S0	0.937	0.016
R2_S180	0.935	0.020
ERR_0°	10.1 %	2.9%
ERR_180°	10.5 %	2.3%

Table 4.7: Results of training.

previous Section are still valid: the error is quite high and does not appear correlated to the R2

score of predicted stress time history.

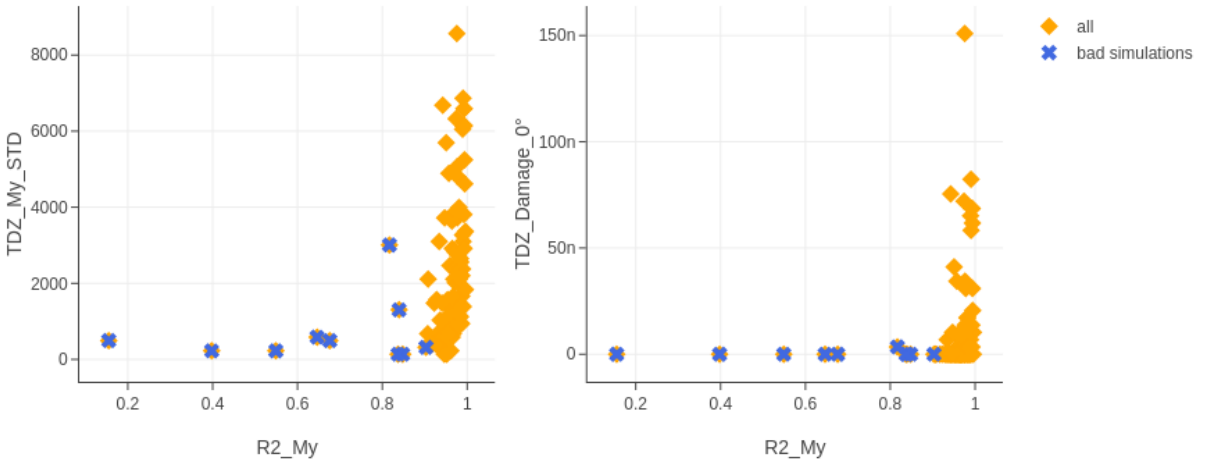


Figure 4.8: Scatter plot of R2 on My against STD of My and Damage at 0° (the considered values are the median across all seeds). In blue are represented the simulations 591, 690, 692, 693, 694, 777, 778, 779, 780 and 781, which have low R2 scores, in orange all the remaining simulations of the subset.

4.2.5 Performances comparison

To conclude we want to provide a comparison in performances between the approach using a set of NNs and the single NN approach.

The LSTM is highly sensitive to the choice of the seed, problem that becomes even more relevant when the network and the dataset grow in size. For this reason, we chose here the networks initialized with the seed that resulted in the best performances in terms of average R2 of stress across sea states on validation data (to avoid overfitting on test). These results are visible in Figure 4.9. Furthermore, Table 4.8 provides the average scores on the given seeds of the considered models.

While the first approach provide better results, in a real scenario it would suffer some major limitations widely discussed before: first, the requirement of an algorithm or set of sensors capable of detecting the sea state scenario and choosing accordingly which NN to use for the forecasting (introduce additional uncertainty and possibility of error) and second the discontinuities in the

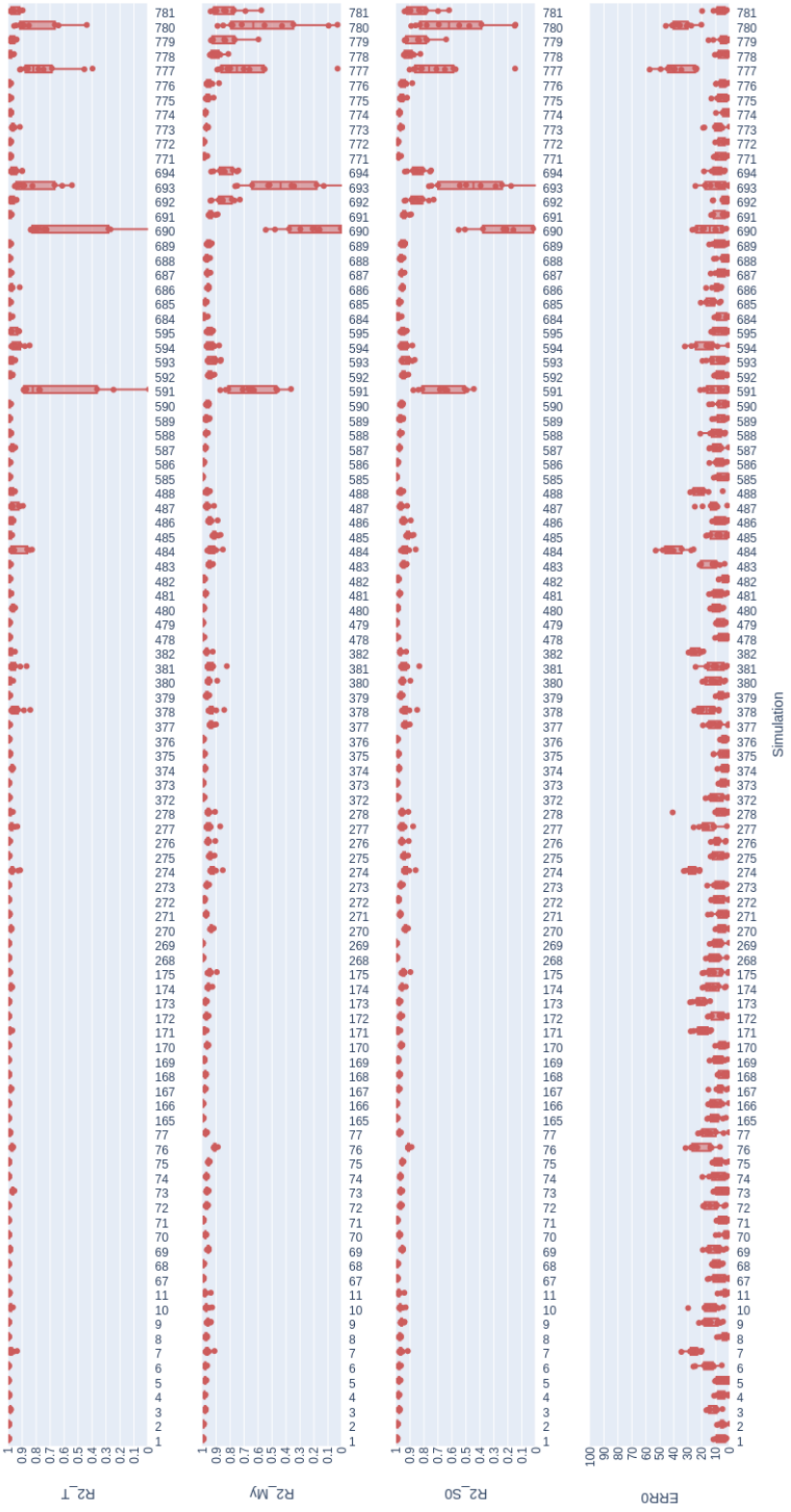


Figure 4.7: Evaluation of the performances of the NN on each sea state. For each sea state scenario a boxplot synthesizes the distribution of the results obtained setting up the NN with distinct seeds. Notice that, for visibility purposes, the y-axis of the R2 scores is bounded in [0,1], but few outliers actually go down to -0,5.

predicted time series due to the swtiches between NN. This arguments in our opinion favor the single NN over the set of NNs in spite of the acceptable decrease in quality of forecasting in terms of average R2 score across simulations.

	Set of NNs		*Single NN*	
	Mean	*STD*	*Mean*	*STD*
R2_T	0.992	0.005	0.980	0.045
R2_My	0.965	0.028	0.957	0.069
R2_S0	0.965	0.028	0.958	0.068
R2_S180	0.965	0.028	0.957	0.069
ERR_0	8.1%	7.6%	7.1%	7.6%
ERR_180	8.0%	6.5%	7.7%	8.8%

Table 4.8: Comparison of the single NN approach and the set of NNs approach in terms of average evaluation metrics across simulation (on the best performing seeds).

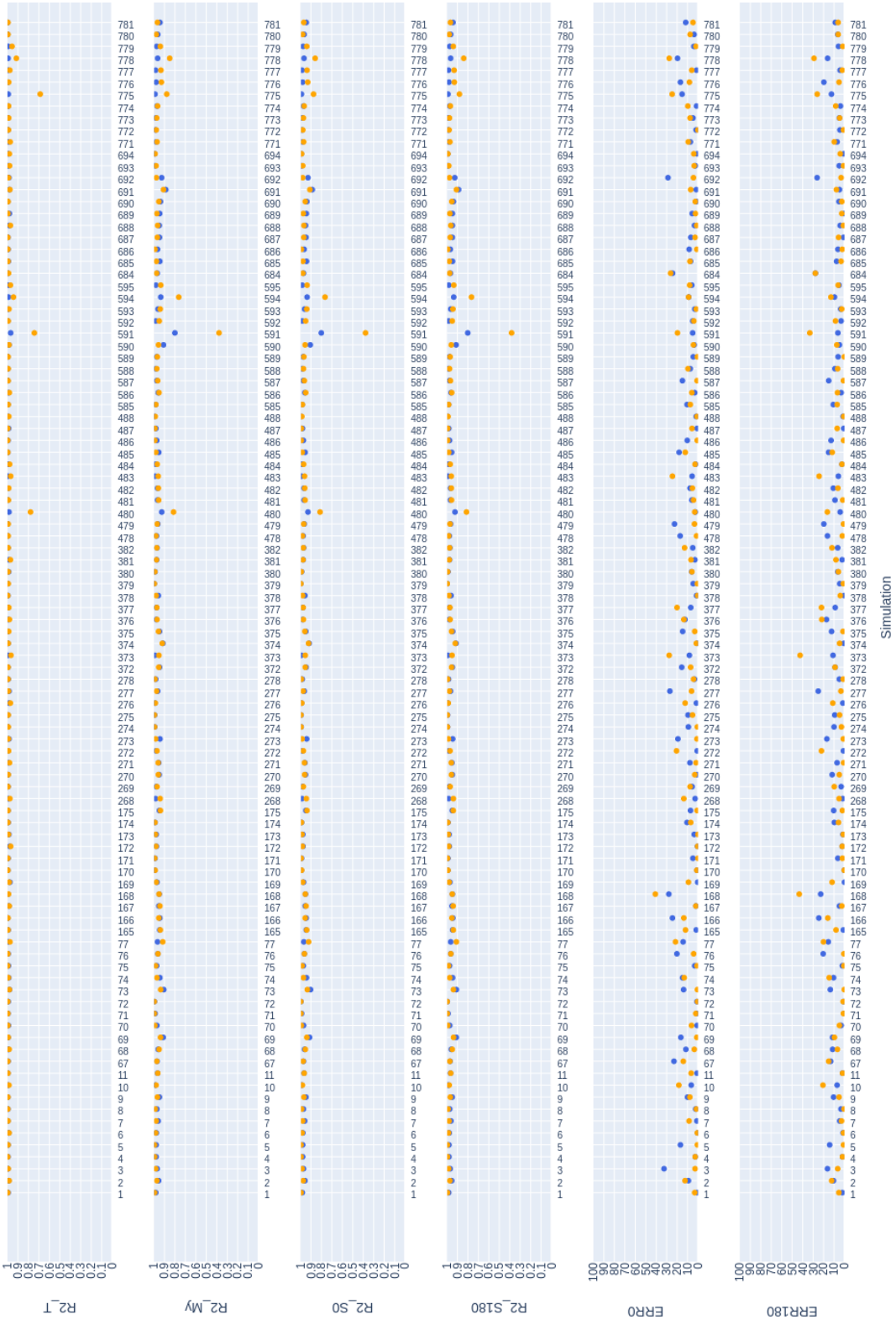


Figure 4.9: Comparison of the performances of the best model for the set of NNs (in blue) and for the single NN for multi scenario (in orange), which are given by seed 4567 and 3456 respectively.

5

Fatigue Damage Estimation

The results of the previous Chapter show that the relative error on the fatigue damage forecasting is in general quite high and unrelated to the R2 scores. This Chapter investigates the reasons behind this problem. All the computations that will be presented relatively to the Fatigue Damage, rely on the python package *fatpack**.

We will start by investigating the reasons behind the high relative errors on Fatigue Damage when computed locally on the test dataset of a single simulation, and next we will discuss a "global" evaluation of Fatigue Damage (accounting for test dataset of all simulations together) in terms of relative error.

*https://github.com/Gunnstein/fatpack

5.1 Local Fatigue Damage

In this section we analyze the problem of the relative error on Fatigue Damage for a single sea state. For this purpose, we consider the predictions on test performed in Section 4.2.3 by training on all input variables available, and we select one sea state for which we obtain, for stress at angle 0°, high R2 scores in the stress but an important relative error on the Fatigue Damage. A sea state leading to such performances is 8, trained on seed 5678, which results in a R2 score of stress of 0.951 and a relative error on Fatigue Damage of 34.9%. First, we can verify the quality of the time series forecasting by considering the plot of the true vs predicted stress time history, visible in Figure 5.1.

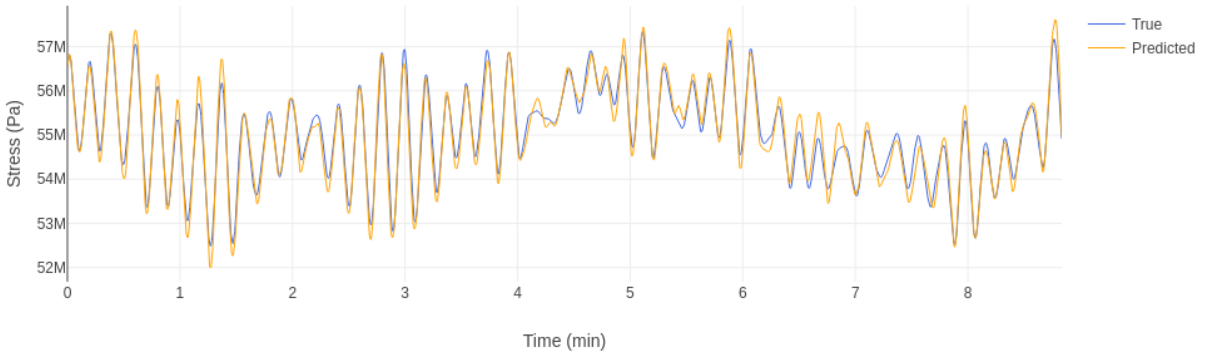
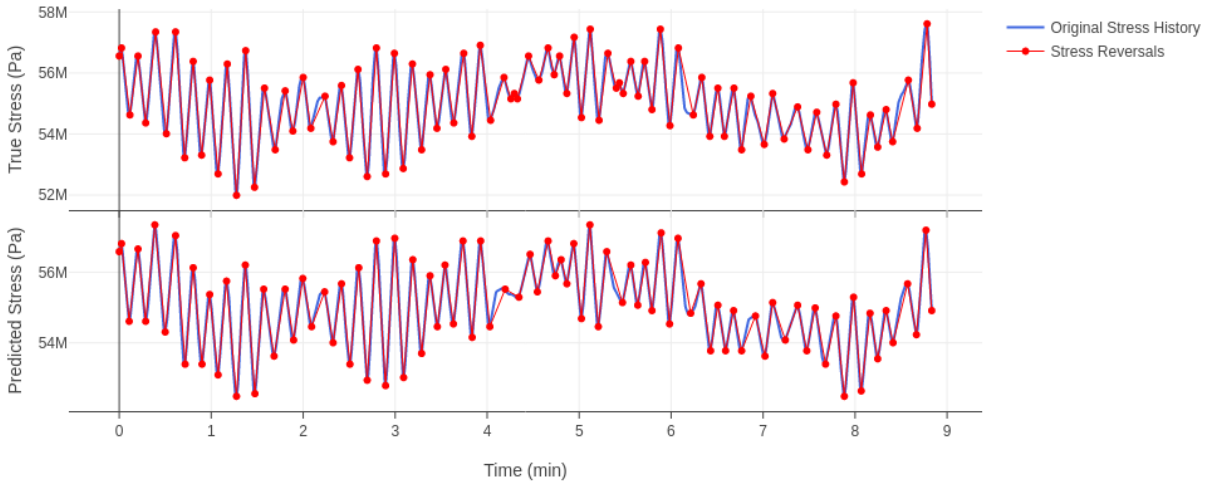


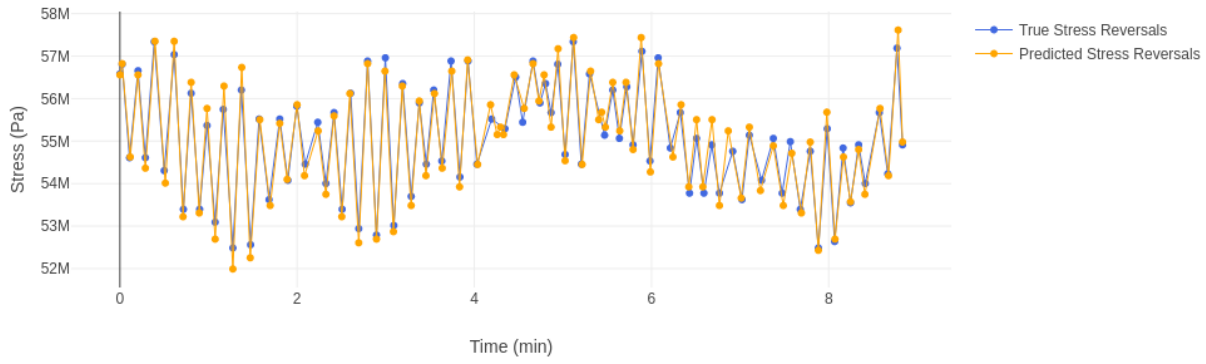
Figure 5.1: Plot of true and predicted stress time series on test data of simulation 8, with training seed 5678.

Next, as seen in Section 2.2.4, the stress time series are preprocessed leading to a sequence of peaks and valleys, which are again compared between true and predicted in Figure 5.2.

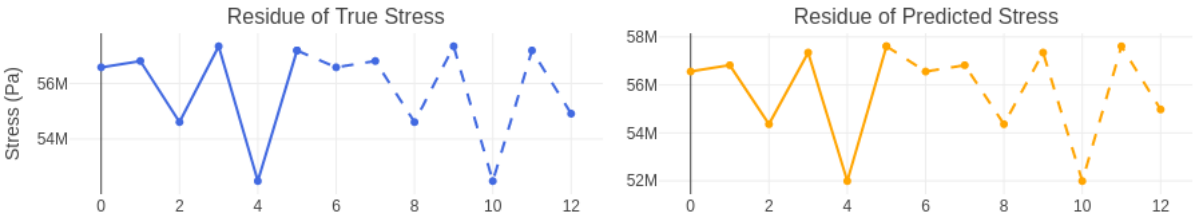
Now, the main insights come from comparing how well the distribution of the stress ranges extracted with Rainflow Counting Algorithm from the original time series is reproduced from the forecasted time series. In particular, we want to compare the distributions of the damage per stress range between true and predicted time series. This comparisons are shown in Figure 5.3b. We can argue that for a short stress time series (such as the 9 minute long time series that belong to our test and validation set), the distribution of the extracted stress ranges is a poor approximation of the “true” distribution, where we consider the “true” distribution as the distribution of stress



(a) Preprocessing leading to sequences of peaks and valleys for true and predicted stress time series.



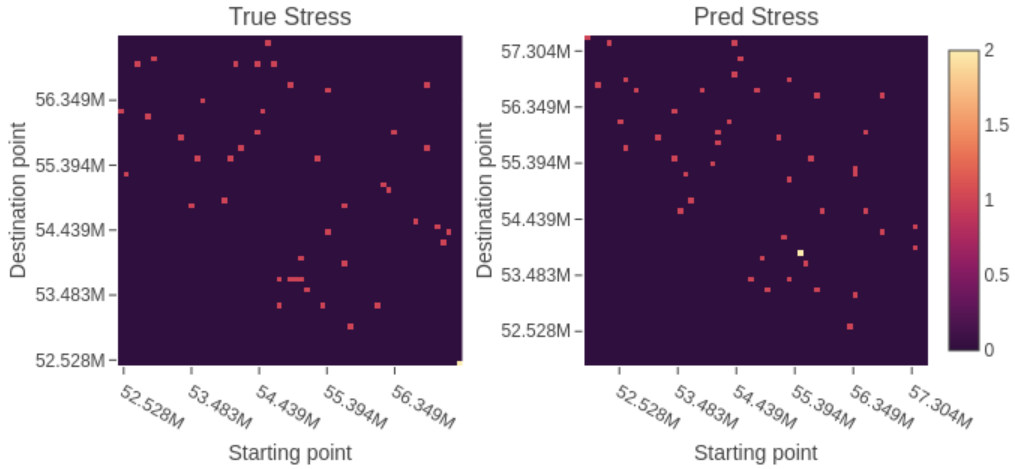
(b) Comparison between true and predicted peak-valley sequences.



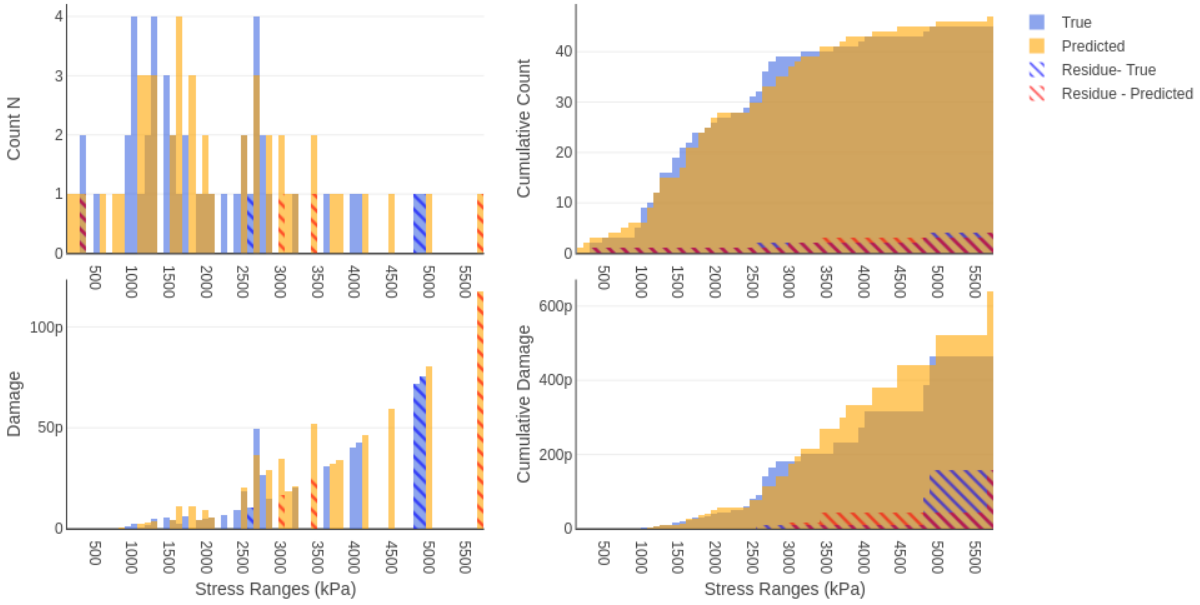
(c) Comparison between the residues of true and predicted peak-valley sequences.

Figure 5.2: Visualization of the preprocessing steps of rainflow counting for simulation 8 trained on seed 5678.

ranges of an hypothetical infinitely long time series. Thus, a longer time series is statistically likely to produce a better approximation of the "true" stress ranges distribution (see Figure 5.3a).



(a) Comparison of Rainflow matrices of true and predicted stress time series.



(b) Comparison between true and predicted distribution of stress ranges and relative damage.

Figure 5.3: Visualization of Rainflow matrices and of the distribution of stresses ranges and relative damage for the stress of simulation 8 trained on seed 5678.

Moreover, notice that the cumulative damage is dominated by few medium and high stress amplitude cycles, while the low stress amplitude cycles impact slightly the cumulative damage

regardless of their numerosity, in accordance to the exponential decrease of the fatigue (number of cycles until break) for higher stress ranges (see Figure 5.4). For example, observe in Figure 5.3b the cumulative count of stress cycles under 3000 kPa compared to the total count, and the cumulative damage of stress cycles under 3000 kPa compared to the total cumulative damage.

Last, observe the presence of some stress cycles, including some in the medium and high amplitude ranges, which are associated to the residue and have an important impact on the total cumulative damage. From Section 2.2.4, we recall that the residue is the sequence of peaks and valleys remaining at the end of the cycle counting algorithm, that is not producing complete cycles. By design, *fatpack* handles incomplete cycles by making them complete; this is obtained by concatenating an identical residue to the original one and computing its cycles 5.2c. However, this solution leads to an overestimation of the real Fatigue Damage. Such overestimation can become more important if the Damage is computed for a relatively short time series, since the length of the residue is likely to be a higher fraction of the total length of the peaks-valleys sequence.

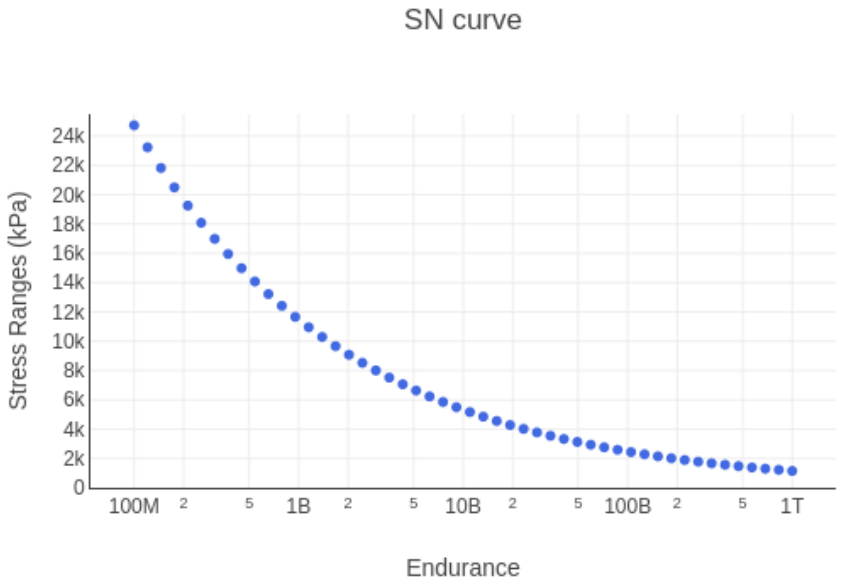


Figure 5.4: SN curve of reference for our case of study, zoomed to highlight the stress ranges detected in the various stress time histories present in the test set.

From all these observations we can conclude that the relative error on the Fatigue Damage

computed locally on the test data of a single simulation is not a reliable metric, and this is mostly due to the insufficient length of the time series, which leads to a poor approximation of the true distribution of stress ranges, further exacerbated by the more relevant fraction of the peaks-valleys sequence (the residue), that is required to be repeated to complete all cycles.

5.2 Global Fatigue Damage

According to the observations of previous Section, the relative error on Fatigue Damage computed locally on single test stress time histories is not a reliable metric. This Section investigates how to assess the capability of our models of estimating the Fatigue Damage. In particular, we focus on trying to estimate the Fatigue Damage globally on all our test data, composed of a set of single test stress time histories of 9 minutes.

Summing the individual Fatigue Damage computed over all the test time histories is an option, but it does not solve the problem of the high impact of the residue in short time series.

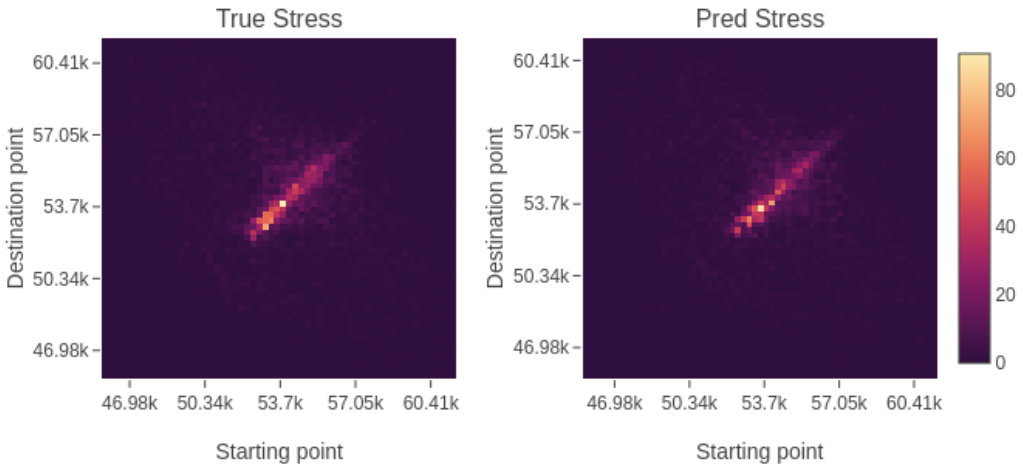
Therefore, the adopted solution consists in simply concatenating all the test time series and computing the Fatigue Damage of the resulting time series. This solution has the advantage of providing a common preprocessing phase (binning in particular) for all the sea states. Moreover it solves the problem of the residue by having a single residue resulting from a much longer time series, and thus representing a much smaller fraction of the peaks-valleys sequence. The disadvantage of this solution consists in the introduction of artificial discontinuities in the concatenated time series between each two distinct adjacent time series. This artificial discontinuities create stress ranges that count for cycles contributing to the final Fatigue Damage. However, in our case, we computed an upper bound[†] to the contribution of the Fatigue Damage caused by these discontinuities to the total Fatigue Damage, and such contribution consists of just 0.88% of the total Fatigue Damage (considering the true time histories), making it not so relevant. Note that in less fortunate cases, the contribution of discontinuities on the Fatigue Damage can still be

[†]The upper bound has been computed counting one complete cycle for each stress range identified by the amplitude of a discontinuity.

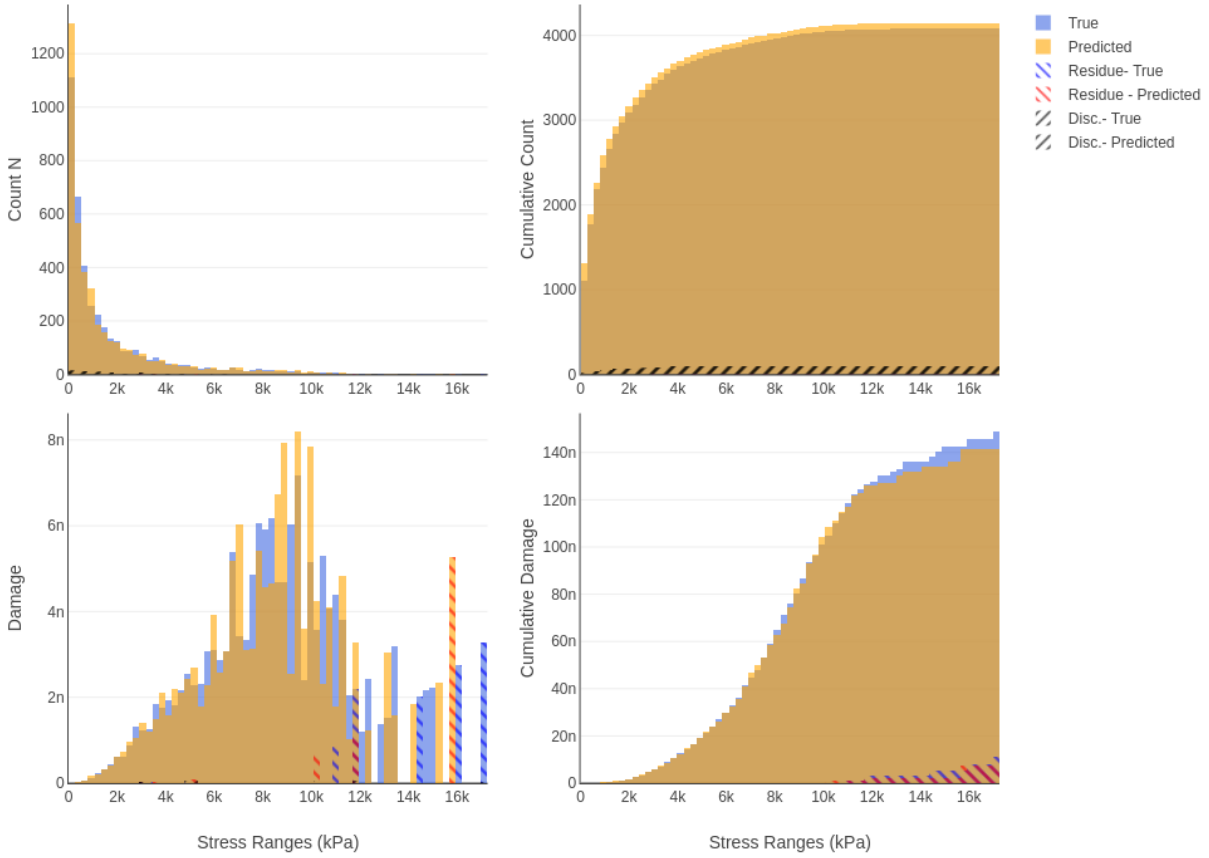
lowered by reordering the time histories before their concatenation in order to obtain lower stress ranges in the discontinuities. Such order can be found just by random trials or more optimally by means of genetic algorithms.

In particular, in Figure 5.5b, we can observe all that has been just explained. Notice how the residue has now a reduced impact on the overall Fatigue Damage, and the residues Fatigue Damage of true and predicted time series are matching pretty well. Moreover, the impact of discontinuities is not even visible in the plots of the cumulative damage.

Estimating the Fatigue Damage globally by concatenating the whole test data, the models reach good level of performances, with relative errors under 13% (see Figure 5.6). Notice once again that the higher performances of the models consisting of sets of NNs is actually limited in reality by other challenges discussed in the previous Chapter. In particular, the best single-NN model that we chose on the previous Chapter according to the average R2 scores across simulations, provides a relative error in the global fatigue damage forecasting of 2.31% at angle 0° and 3.15% at angle 180°.



(a) Comparison of Rainflow matrices of true and predicted stress time series.



(b) Comparison between true and predicted distribution of stress ranges and relative damage.

Figure 5.5: Visualization of Rainflow matrices and of the distribution of stresses ranges and relative damage for predictions of trainings with seed 8901.

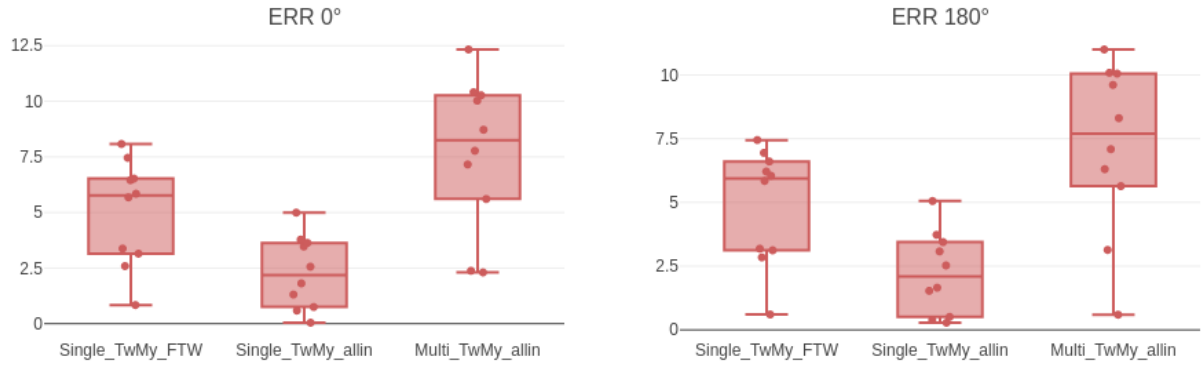


Figure 5.6: Boxplots expressing the distribution of the global relative error over distinct seeds of the various models trained.

6

Conclusions

This thesis focused on the problem of Fatigue Damage estimation of critical nodes in the pipe of offshore structures. The approach studied consisted of performing time series forecasting of the variables used to define the stress in order to compute the Fatigue Damage on the predicted stress time histories.

First, we restricted the domain of the problem to a better manageable study case. For this purpose, we observed that only two angles (0° and 180°) of the section of the pipe accumulate the maximum Fatigue Damage in almost all sea state scenario at the critical node at TDZ, according to our simulations. This considerations directed the efforts of time series forecasting to only the axial tension and the bending moment at y-axis. Moreover, further analysis of our context showed that the bending moment at y-axis impacts the stress much more than the axial tension. Next, to expand the forecasting capabilities into a multi sea state scenario, we selected a representative subset of 98 simulations over the total of 808 simulations.

In order to perform time series forecasting of axial tension and bending moment at y-axis, from

which to extract the stress at 0° and 180°, we relied on LSTM Neural Network. We obtained pretty good results by using as inputs only variables at FWT node. However, including all the input variables allowed for a major improvement in performances. In particular, on the best seed we obtained an average R2 score of 0.97 on stress at angle 0°* (with a STD of 0.02), by training one NN for each sea state of our subset. Unfortunately, this approach involves some relevant limitations in a real application, first due to the choice of the NN to use for forecasting (how to predict the current sea state and how often to update such prediction in order to switch to the corresponding sea state NN), and second due to the discontinuities in the forecasted time series resulting from switching between NNs. For these reasons we attempted to train a single model capable of generalization across multiple sea states, by training on the collective data of all the subset of simulations. Nonetheless, we achieved some really satisfying results: an average R2 score of 0.96 on the stress at angle 0°, with STD of 0.04 (on the best seed). The NN generalized really well to most of the sea states selected, with few outliers determining the high STD on the R2 score. Fortunately, such outliers were actually associated to "calm" sea states, since their bending moment time histories showed really low STD and, accordingly, the Fatigue Damage of the stress time histories was really low.

Throughout all the study we tried to not detach the intermediate goal of optimizing the time series forecasting of axial tension and bending moment, from the final goal of Fatigue Damage estimation of stress. For this purpose, we dived into the reasons behind the high relative errors obtained on the Fatigue Damage computed on the stress time histories, contradicting the good performances of the predictions in terms of R2 scores. We found out that the relative error on Fatigue Damage computed for short stress time histories of 9 minutes is not a reliable metric. On the other hand, testing the models' ability to predict the Fatigue Damage on longer time series, obtained by concatenating test data of all the considered simulations (for a total of 14h 42m), lead to much more reliable and low values of relative error on Fatigue Damage. The relative error on the "global" Fatigue Damage obtained in this way, under the single-NN approach, remains

*All the results obtained at angle 0° are very similar to the results obtained at angle 180°.

under 13% for all models trained on 10 different seeds, and stays around 3% (for stress at 0° and 180°) for the best of these models (the model with the higher R2 scores).

According to our results, estimating the Fatigue Damage relying on time series forecasting based on single NN model trained on several sea state is a promising approach. We concluded also that the quality of the final Fatigue Damage estimation specifically depends on how well the distribution of the medium and high amplitude stress cycles extracted with Rainflow analysis matches the distribution of the original stress time histories.

Directions of future research

In the future, the work on this topic could expand by attempting to obtain the same or better results in terms of R2 scores on time series forecasting and relative error on global Fatigue Damage when including all sea states simulations. For this purpose, other state of the art deep learning techniques can be employed, such as Temporal Fusion Transformers (TFT), with the potential advantage of improved performances and higher interpretability of results (for example, which input variables impact more the learning of the output). It is also necessary to extend the study to the other angles of the considered node (TDZ) and to other critical points (OW and HB). To conclude, a critical limit in this study consisted in the length of the available simulations: according to our findings, using longer test time histories (which obviously require longer simulations) could allow to obtain more accurate relative errors on Fatigue Damage globally (on the concatenation of all test data) but also locally (on single simulations).

A

Appendix

Table A.1: Parameters of the subset of simulations on which trainings have been performed.

Load Case	*Roll Damping (%)*	*Wave Hs (m)*	*Wave Tp (s)*	*Wave Gamma*	*Wave Direction (deg)*	*Associated Wind Speed (m/s)*	*Wind Direction (deg)*	*Associated Current Speed (m/s)*	*Current Direction (deg)*	*Probability*
1	0.015	1.875	18.5	1.00	345	8.53	10	0.57	135	1.120e-9
2	0.0075	1.875	17.5	1.00	345	8.53	10	0.50	135	7.278e-9
3	0.0075	1.875	16.5	1.00	345	8.53	10	0.32	135	1.512e-8
4	0.0075	1.875	15.5	1.00	345	8.53	10	0.42	135	5.375e-8
5	0.0025	1.875	14.5	1.00	345	8.53	10	0.51	135	8.398e-8
6	0.0025	1.875	12.8	1.00	345	8.53	10	0.47	135	4.401e-7
7	0.025	1.875	6.5	1.34	345	8.53	10	0.52	135	5.554e-7
8	0.025	1.875	10.9	1.00	345	8.53	10	0.46	135	8.863e-7

Table A.1: Parameters of the subset of simulations on which trainings have been performed.

Load Case	*Roll Damping (%)*	*Wave Hs (m)*	*Wave Tp (s)*	*Wave Gamma*	*Wave Direction (deg)*	*Associated Wind Speed (m/s)*	*Wind Direction (deg)*	*Associated Current Speed (m/s)*	*Current Direction (deg)*	*Probability*
9	0.025	1.875	9.5	1.00	345	8.53	10	0.48	135	1.203e-6
10	0.025	1.875	7.5	1.00	345	8.53	10	0.46	135	2.981e-6
11	0.025	1.875	8.5	1.00	345	8.53	10	0.49	135	3.773e-6
67	0.015	1.875	18.5	1.00	0	8.53	25	0.57	135	3.583e-8
68	0.0075	1.875	17.5	1.00	0	8.53	25	0.50	135	2.329e-7
69	0.0075	1.875	16.5	1.00	0	8.53	25	0.32	135	4.837e-7
70	0.0075	1.875	15.5	1.00	0	8.53	25	0.42	135	1.720e-6
71	0.0025	1.875	14.5	1.00	0	8.53	25	0.51	135	2.687e-6
72	0.0025	1.875	12.8	1.00	0	8.53	25	0.47	135	1.408e-5
73	0.025	1.875	6.5	1.34	0	8.53	25	0.52	135	1.777e-5
74	0.025	1.875	10.9	1.00	0	8.53	25	0.46	135	2.836e-5
75	0.025	1.875	9.5	1.00	0	8.53	25	0.48	135	3.848e-5
76	0.025	1.875	7.5	1.00	0	8.53	25	0.46	135	9.540e-5
77	0.025	1.875	8.5	1.00	0	8.53	25	0.49	135	1.207e-4
165	0.015	1.875	18.5	1.00	15	8.53	40	0.57	135	4.210e-7
166	0.0075	1.875	17.5	1.00	15	8.53	40	0.50	135	2.737e-6
167	0.0075	1.875	16.5	1.00	15	8.53	40	0.32	135	5.684e-6
168	0.0075	1.875	15.5	1.00	15	8.53	40	0.42	135	2.021e-5
169	0.0025	1.875	14.5	1.00	15	8.53	40	0.51	135	3.158e-5
170	0.0025	1.875	12.8	1.00	15	8.53	40	0.47	135	1.655e-4
171	0.025	1.875	6.5	1.34	15	8.53	40	0.52	135	2.088e-4
172	0.025	1.875	10.9	1.00	15	8.53	40	0.46	135	3.332e-4
173	0.025	1.875	9.5	1.00	15	8.53	40	0.48	135	4.522e-4
174	0.025	1.875	7.5	1.00	15	8.53	40	0.46	135	1.121e-3

Table A.1: Parameters of the subset of simulations on which trainings have been performed.

Load Case	*Roll Damping (%)*	*Wave Hs (m)*	*Wave Tp (s)*	*Wave Gamma*	*Wave Direction (deg)*	*Associated Wind Speed (m/s)*	*Wind Direction (deg)*	*Associated Current Speed (m/s)*	*Current Direction (deg)*	*Probability*
175	0.025	1.875	8.5	1.00	15	8.53	40	0.49	135	1.419e-3
268	0.0075	1.875	18.5	1.00	30	8.53	55	0.57	135	1.469e-6
269	0.0075	1.875	17.5	1.00	30	8.53	55	0.50	135	9.549e-6
270	0.0075	1.875	16.5	1.00	30	8.53	55	0.32	135	1.983e-5
271	0.0025	1.875	15.5	1.00	30	8.53	55	0.42	135	7.052e-5
272	0.0025	1.875	14.5	1.00	30	8.53	55	0.51	135	1.102e-4
273	0.0025	1.875	12.8	1.00	30	8.53	55	0.47	135	5.774e-4
274	0.025	1.875	6.5	1.34	30	8.53	55	0.52	135	7.287e-4
275	0.025	1.875	10.9	1.00	30	8.53	55	0.46	135	1.163e-3
276	0.025	1.875	9.5	1.00	30	8.53	55	0.48	135	1.578e-3
277	0.025	1.875	7.5	1.00	30	8.53	55	0.46	135	3.912e-3
278	0.025	1.875	8.5	1.00	30	8.53	55	0.49	135	4.950e-3
372	0.0075	1.875	18.5	1.00	45	8.53	70	0.57	135	6.486e-6
373	0.0075	1.875	17.5	1.00	45	8.53	70	0.50	135	4.216e-5
374	0.0075	1.875	16.5	1.00	45	8.53	70	0.32	135	8.756e-5
375	0.0025	1.875	15.5	1.00	45	8.53	70	0.42	135	3.113e-4
376	0.0025	1.875	14.5	1.00	45	8.53	70	0.51	135	4.864e-4
377	0.0025	1.875	12.8	1.00	45	8.53	70	0.47	135	2.549e-3
378	0.025	1.875	6.5	1.34	45	8.53	70	0.52	135	3.217e-3
379	0.025	1.875	10.9	1.00	45	8.53	70	0.46	135	5.133e-3
380	0.025	1.875	9.5	1.00	45	8.53	70	0.48	135	6.966e-3
381	0.025	1.875	7.5	1.00	45	8.53	70	0.46	135	1.727e-2
382	0.025	1.875	8.5	1.00	45	8.53	70	0.49	135	2.185e-2
478	0.0075	1.875	18.5	1.00	60	8.53	85	0.57	135	9.209e-6

Table A.1: Parameters of the subset of simulations on which trainings have been performed.

Load Case	*Roll Damping (%)*	*Wave Hs (m)*	*Wave Tp (s)*	*Wave Gamma*	*Wave Direction (deg)*	*Associated Wind Speed (m/s)*	*Wind Direction (deg)*	*Associated Current Speed (m/s)*	*Current Direction (deg)*	*Probability*
479	0.0075	1.875	17.5	1.00	60	8.53	85	0.50	135	5.986e-5
480	0.0025	1.875	16.5	1.00	60	8.53	85	0.32	135	1.243e-4
481	0.0025	1.875	15.5	1.00	60	8.53	85	0.42	135	4.420e-4
482	0.0025	1.875	14.5	1.00	60	8.53	85	0.51	135	6.907e-4
483	0.0025	1.875	12.8	1.00	60	8.53	85	0.47	135	3.619e-3
484	0.025	1.875	6.5	1.34	60	8.53	85	0.52	135	4.568e-3
485	0.025	1.875	10.9	1.00	60	8.53	85	0.46	135	7.289e-3
486	0.025	1.875	9.5	1.00	60	8.53	85	0.48	135	9.890e-3
487	0.025	1.875	7.5	1.00	60	8.53	85	0.46	135	2.452e-2
488	0.025	1.875	8.5	1.00	60	8.53	85	0.49	135	3.103e-2
585	0.0025	1.875	18.5	1.00	75	8.53	100	0.57	135	1.883e-6
586	0.0025	1.875	17.5	1.00	75	8.53	100	0.50	135	1.224e-5
587	0.0025	1.875	16.5	1.00	75	8.53	100	0.32	135	2.543e-5
588	0.0025	1.875	15.5	1.00	75	8.53	100	0.42	135	9.040e-5
589	0.0025	1.875	14.5	1.00	75	8.53	100	0.51	135	1.413e-4
590	0.0025	1.875	12.8	1.00	75	8.53	100	0.47	135	7.402e-4
591	0.025	1.875	6.5	1.34	75	8.53	100	0.52	135	9.342e-4
592	0.025	1.875	10.9	1.00	75	8.53	100	0.46	135	1.491e-3
593	0.025	1.875	9.5	1.00	75	8.53	100	0.48	135	2.023e-3
594	0.025	1.875	7.5	1.00	75	8.53	100	0.46	135	5.015e-3
595	0.025	1.875	8.5	1.00	75	8.53	100	0.49	135	6.346e-3
684	0.0025	1.875	18.5	1.00	90	8.53	115	0.57	135	4.658e-7
685	0.0025	1.875	17.5	1.00	90	8.53	115	0.50	135	3.028e-6
686	0.0025	1.875	16.5	1.00	90	8.53	115	0.32	135	6.289e-6

Table A.1: Parameters of the subset of simulations on which trainings have been performed.

Load Case	*Roll Damping (%)*	*Wave Hs (m)*	*Wave Tp (s)*	*Wave Gamma*	*Wave Direction (deg)*	*Associated Wind Speed (m/s)*	*Wind Direction (deg)*	*Associated Current Speed (m/s)*	*Current Direction (deg)*	*Probability*
687	0.0025	1.875	15.5	1.00	90	8.53	115	0.42	135	2.236e-5
688	0.0025	1.875	14.5	1.00	90	8.53	115	0.51	135	3.494e-5
689	0.025	1.875	12.8	1.00	90	8.53	115	0.47	135	1.831e-4
690	0.025	1.875	6.5	1.34	90	8.53	115	0.52	135	2.310e-4
691	0.025	1.875	10.9	1.00	90	8.53	115	0.46	135	3.687e-4
692	0.025	1.875	9.5	1.00	90	8.53	115	0.48	135	5.003e-4
693	0.025	1.875	7.5	1.00	90	8.53	115	0.46	135	1.240e-3
694	0.025	1.875	8.5	1.00	90	8.53	115	0.49	135	1.570e-3
771	0.0025	1.875	18.5	1.00	105	8.53	130	0.57	135	2.911e-8
772	0.0025	1.875	17.5	1.00	105	8.53	130	0.50	135	1.892e-7
773	0.0025	1.875	16.5	1.00	105	8.53	130	0.32	135	3.930e-7
774	0.0025	1.875	15.5	1.00	105	8.53	130	0.42	135	1.397e-6
775	0.0025	1.875	14.5	1.00	105	8.53	130	0.51	135	2.184e-6
776	0.0025	1.875	12.8	1.00	105	8.53	130	0.47	135	1.144e-5
777	0.025	1.875	6.5	1.34	105	8.53	130	0.52	135	1.444e-5
778	0.025	1.875	10.9	1.00	105	8.53	130	0.46	135	2.304e-5
779	0.025	1.875	9.5	1.00	105	8.53	130	0.48	135	3.127e-5
780	0.025	1.875	7.5	1.00	105	8.53	130	0.46	135	7.752e-5
781	0.025	1.875	8.5	1.00	105	8.53	130	0.49	135	9.810e-5

Listing of acronyms

ANN . . . . . Artificial Neural Network

AR . . . . . AutoRegressive model

AVG . . . . . Average

BPTT . . . . . Back-Propagation Through Time

CNN . . . . . Convolutional Neural Network

E . . . . . Energy

EX . . . . . EXogenous input model

FEA . . . . . Finite Element Analysis

FEM . . . . . Finite Element Model

FPSO . . . . . Floating Production Storage and Offloading

FWT . . . . . Floating Wind Tower

HB . . . . . Hog-Bend

HBT . . . . . Hog-Bend Top

LSTM . . . . . Long Short Term Memory

MSE . . . . . Mean Squared Error

Mx . . . . . Bending Moment at x-axis

My . . . . . Bending Moment at y-axis

NARX . . . . . Nonlinear AutoRegressive with eXternal input model

OW . . . . . Offshore Weld

R2 . . . . . R-squared coefficient

RNN . . . . . Recurrent Neural Network

SB . . . . . Sag-Bend

SED . . . . . Spectral Energy Density

SGD . . . . . Stochastic Gradient Descent

STD . . . . . Standard Deviation

T . . . . . Axial Tension

Tw . . . . . Axial Tension

TDZ . . . . . Touch Down Zone

References

[1] M. Yetkin and Y. Kim, "Time series prediction of mooring line top tension by the narx and volterra model," *Applied Ocean Research*, vol. 88, pp. 170–186, 2019.

[2] S. Sundararaman, R. Thethi, and M. Hill, "Data driven virtual sensors for riser prognostic integrity management," in *Offshore Technology Conference*. OnePetro, 2018.

[3] H. S. Gustad, "Using artificial neural networks for predicting bending moments of riser structures," Master's thesis, NTNU, 2019.

[4] C. Millevoi, "Reti neurali ricorrenti: un modello matematico per la previsione del danno a fatica."

[5] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International Journal of Neural Systems*, vol. 31, no. 03, p. 2130001, 2021.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] "Dive into deep learning," https://d2l.ai, accessed: 2023-01-05.

[8] R. I. Stephens, A. Fatemi, R. R. Stephens, and H. O. Fuchs, *Metal fatigue in engineering*. John Wiley & Sons, 2000.

[9] L. Yang and A. Fatemi, "Cumulative fatigue damage mechanisms and quantifying parameters: a literature review," *Journal of testing and evaluation*, vol. 26, no. 2, pp. 89–100, 1998.

[10] M. Matsuishi and T. Endo, "Fatigue of metals subjected to varying stress," *Japan Society of Mechanical Engineers, Fukuoka, Japan*, vol. 68, no. 2, pp. 37–40, 1968.

[11] "Rainflow counting," https://community.sw.siemens.com/s/article/rainflow-counting, accessed: 2023-01-05.

[12] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–a decade review," *Information systems*, vol. 53, pp. 16–38, 2015.