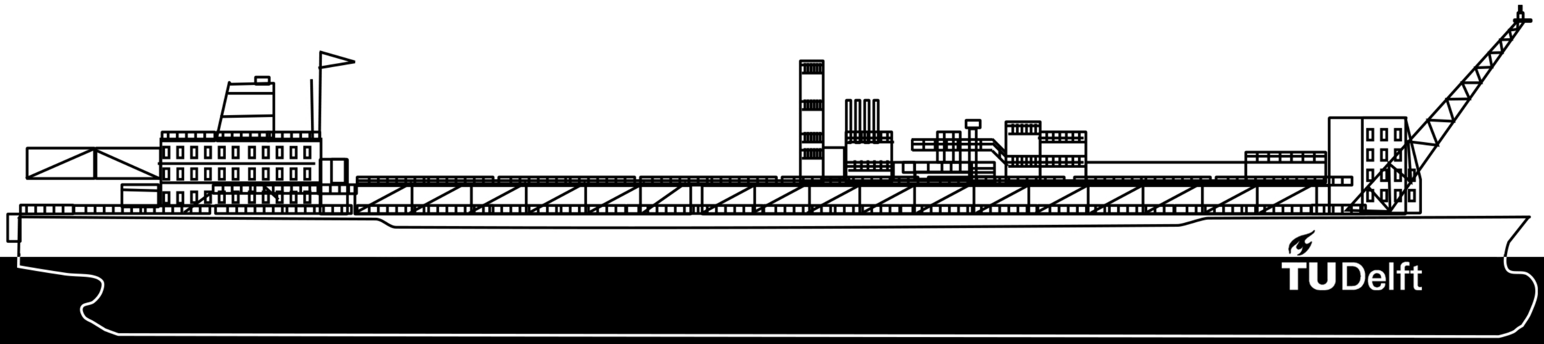
GIANNIS
THEODORIDIS



2021

Faculty of 3mE
Department of Maritime
and Transport Technology

FEASIBILITY STUDY OF THE DEVELOPMENT OF A DIGITAL TWIN FOR THE STRUCTURAL HEALTH MONITORING OF MARINE STRUCTURES USING BIG DATA.

Feasibility Study of the Development of a digital twin for the structural health monitoring of marine structures using big data

by

Giannis Theodoridis

to obtain the degree of Master of Science(MSc)
at the TU Delft University of Technology.

Student number: 5146038		
Project duration:	November 1, 2020 – June 21, 2021	
Thesis committee:	Dr. L. Pahlavan,	TU Delft, supervisor, chair
	Dr. C. L. Walters,	TU Delft
	Dr. H. C. Seyffert,	TU Delft
	Dr. D. Zarouchas,	TU Delft
	Ir. R. Hageman,	MARIN, daily supervisor

Abstract

Structural health monitoring of maritime and offshore structures can contribute to the reduction of the amount of uncertainty in assessment of the operational loads and the structural integrity. Such a survey can also reduce the large safety margins during the design and optimize the conservative inspection and maintenance schedules. One of the emerging variants of structural health monitoring are the so-called *digital twin* systems that not only allow the monitoring of the structures, but can also simulate the past or the future state of the structural integrity.

In the context of this thesis, the feasibility of a digital twin system for the structural health assessment of marine structures based on machine learning algorithms and utilization of big data is examined. The main research question is if the digital twin technology can be enhanced with machine learning to accurately monitor the structural health of marine structures in terms of loading and fatigue damage accumulation.

In order to answer this question four modules have been developed and tested. The first module uses artificial neural networks trained on operational data to predict the fatigue damage accumulation and the frequency that corresponds to the maximum stress power density. The error of the developed networks turned out to be negligible in the investigated cases, with standard deviation of less than 4% in the predictions. The peak frequency of the stress power density is predicted using a random forest regression algorithm. The optimized version of the algorithm has led to an accuracy of 93% with standard deviation of less than 1% in the predictions.

The second module is used to recalibrate the design response amplitude operators of the structure using the predictions of the first module or the operational data, if available. The design response amplitude operators are scaled and shifted in order to minimize the deviation between the spectral fatigue calculations and the predicted (or measured) data.The third module is a static/quasi-static load-reconstruction module. Using on-board strain measurements it is able to calibrate the loading of the structure based on a reformulation of the conventional finite element problem for static and quasi-static loading. A sensitivity analysis of the algorithm effectiveness have been performed using multiple load cases in which the reconstruction error turned out negligible.

The fourth and final module relates to structural reliability analysis. This module is based on the estimation of the Hansofer-Lind reliability index as a minimization problem. The employed optimization engine is a variation of the particle swarm optimization algorithm using chaotic system behaviour to iteratively calculate the user-defined parameters. The limit state equation is formulated in a way that it takes the uncertainties related to the fatigue damage accumulation prediction model and Miner's Rule into consideration. Uncertainties related to the material and the fabrication process are not taken into account.

The developed modules have been tested on structural details of a vessel monitored in the scope of Monitas Joint Industry Project (JIP). About a year's worth of data has been used to train the machine learning algorithms.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

AHMS	Advisory Hull Monitoring System
AI	Artificial Intelligence
AIM	Asset Integrity Management
AIS	Automatic Identification System
ANN	Artificial Neural Network
BIM	Building Information Modeling
CPSO	Chaotic Particle Swarm Optimization
DNV	Det Norske Veritas
DT	Digital Twin
FEA	Finite Element Analysis
FEM	Finite Element Method
FORM	First Order Reliability Method
FPSO	Floating Production Storage Offloading
FPU	Floating Production Unit
IMR	Inspection Maintenance Repair
JIP	Joint Industry Project
LBSG	Long Base Strain Gauge
LHS	Latin Hypercube Sampling
LMC	Load Magnitude Calibration
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
MSLE	Mean Squared Logarithmic Error
PCA	Principal Component Analysis
PDF	Probability Density Function
PSO	Particle Swarm Optimization
RAO	Response Amplitude Operator
RBI	Risk Based Inspection
ReLU	Rectified Linear Unit
RSM	Response Surface Modeling
SHM	Structural Health Monitoring
SVM	Support Vector Machine

Chapter 1

Introduction

In a world pushing for digitalization, the marine industry could be no exception. In the past few years, structural integrity assessment strategies have evolved to include hull monitoring systems and digital twin technologies. Such systems can deepen the understanding of how structures respond and provide valuable insights in terms of how fatigue is accumulated. New information that is available through such systems can be utilized for risk-based inspection purposes, maintenance and lifetime extension of the structures.

Figure 1.1: A field production unit in its operating environment

1.1. The Monitas Group

The Monitas Joint Industry Project (JIP) was initiated in 2006 supported by 18 participants including the major oil companies and classification societies. During the first phase of this project an Advisory Hull Monitoring System(AHMS) was developed. The Glas Dowr(Bluewater) Field Production Storage and Operation Unit (FPSO), during its operation offshore South Africa, was used as a test bed for the development of the AHMS [6]. The aim of AHMS was to monitor the fatigue integrity of FPSOs. The first phase of the project was concluded in 2013.

In 2014, the Monitas Group was founded aiming to further develop the added value of advisory monitoring systems for integrity management [7]. During this period AHMS has been installed on board the Glas Dowr FPSO (Bluewater), the USAN FPSO (ExxonMobil), the CLOV FPSO (TOTAL), the Ichthys FPSO (INPEX), the Moho Nord Field Production Unit (FPU) (TOTAL) and the Bonga FPSO (Shell) and the Aoka Mizu FPSO(Bluewater). The operation of the AHMS on board the above mentioned vessels has result in the acquisition of invaluable data that can provide valuable insight and deeper understanding of the fatigue damage evolution during the lifetime of marine structures. Part of this data will be utilized within this project for the development of a digital twin for structural health monitoring purposes. The main objectives of the Monitas Group are listed below:

- Prevention of unexpected failures resulting in loss of production
- Rational Inspection, Maintenance and Repair (IMR)
- Improved design of future units
- Justification of rational life-time extension decisions
- Assess operational conditions for future design
- Provide advice on operation

1.2. Advisory Hull Monitoring System

The significance of monitoring the fatigue accumulation in marine structures has already been implied. In order to access this problem the AHMS has been developed within the Monitas JIP. This system provides feedback to the operator about the fatigue accumulation rate and if it deviates from the expectation the system provides possible explanations about the deviation origin. Within the AHMS, both environmental conditions and strain measurements are utilized in order to access the accumulation of fatigue damage independently. Both fatigue damage calculations are compared to the design calculations in order to draw conclusions regarding the performance of the hull structure. The environmental data is acquired through a weather buoy. The response data is collected through a system of strain gauges and long base strain gauges located on the hull structure. The AHMS is custom made for each vessel on which it is installed. A typical layout, installed on board the Bonga FPSO, is illustrated in Figure 1.2. The information provided by the AHMS regards only the damage history of the unit [5]. In order to use the data collected by the AHMS for the determination of optimal inspection, repair and maintenance methods or to justify lifetime extension, a system that provides insight in future damage accumulation is required.

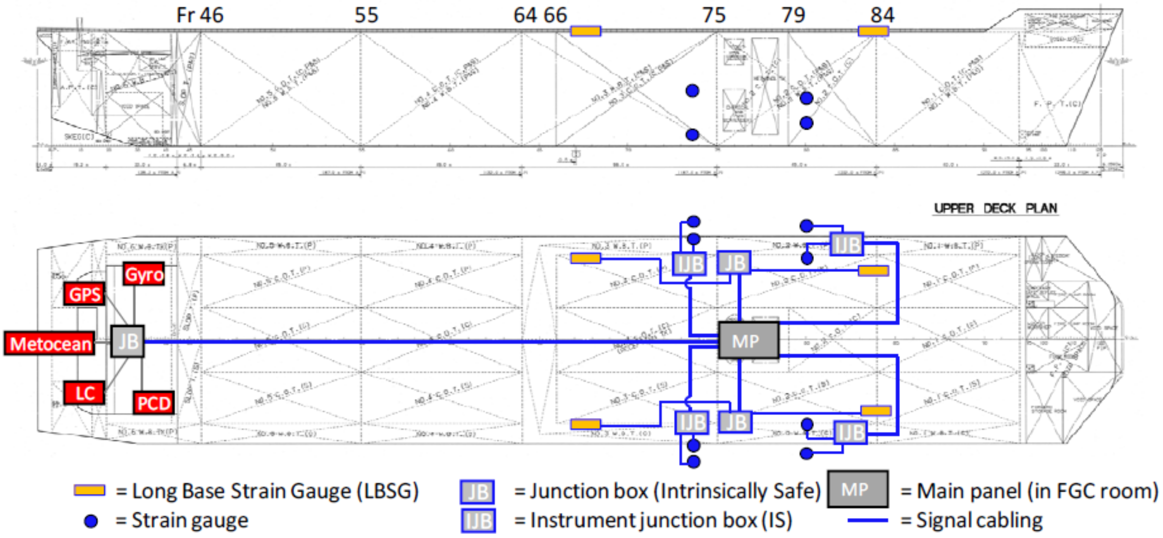


Figure 1.2: Advisory Hull Monitoring System, Bonga FPSO [1]

1.3. Problem Statement

In marine structures, the loading typically involves wave-induced loads, wind-induced loads, and inertia loads resulting from motions of the structure. Such loads can affect the structure in a global or a local level. The most typical cases of loading are global hull girder bending and local panel water pressure. Such loading comes with inherent uncertainty due to the stochastic nature of the sea state [1]. This inherent uncertainty affects the accuracy of the hydro-mechanic modeling of marine structures. The necessity for safe operation of marine structures combined with the above-mentioned uncertainty leads to the incorporation of large safety factors in the calculation of fatigue accumulation in marine structures [5].

In order for the safety margins required by classification societies to be achieved, marine structures are significantly over-designed in numerous occasions. This over-design results in excessive economical cost of the structures. In order to avoid excessive costs while ensuring safety of operations, the fatigue accumulation process should be carefully examined and translated to rational safety margins that do not lead to over-designed structures.

Under this perspective, there are a lot of benefits to be gained from the development of state-of-the-art structural health monitoring systems [6]. A digital twin built for structural health monitoring purposes seems as a highly promising solution since it provides such benefits while being in line with the push for digitalization.

1.4. Literature Review

Performing a state of the art literature review is a pivotal step in order to understand the relevant published research, assess the knowledge gap and come up with innovative suggestions. The scientific research fields that are considered in the context of this literature review are:

- Digital twin technology for Structural Health Monitoring (SHM) applications
- Fatigue analysis using Machine Learning (ML) algorithms
- Load reconstruction techniques that utilize in-situ measurements
- Structural reliability analysis

1.4.1. Structural Health Monitoring using Digital Twin Technology

In recent years substantial effort has been put towards the digitalization of the global fleet. A significant amount of this effort has been focused on the development of digital twins. One of the most promising areas where digital twin technologies can be applied is the health monitoring of structures. In order to understand the ways in which a digital twin can prove beneficial for structural health monitoring, an important first step is to review the definition of a digital twin.

1.4.1.1 What is a Digital Twin?

Several definitions of a digital twin have been proposed. Grieves and Vickers[8] define a digital twin as "a set of virtual information construct that fully describes a potential or actual physical manufactured product", while in the context of Digital Built Britain, a digital twin is defined as "a realistic digital representation of assets, processes or systems in the built or natural environment" [9].

A digital twin as it will be established in the scope of this project, is best described by the definition given by Lu et al [10]. "Living digital simulation models that are able to learn and update from multiple sources, and to represent and predict the current and future conditions of the physical counterparts correspondingly and timely". This definition is deemed to be the most accurate one and learning from multiple sources is the pillar of the digital twin system proposed to be developed in the scope of this project. Utilizing machine learning methods and combining weather and response data, the digital twin will aim to predict the fatigue damage accumulation of the structures under study as well as their operational response and load spectrums.

1.4.1.2 Digital Twin Applications for SHM

The benefits of virtual hull monitoring (Digital Twin) for SHM have been examined by Defence Research and Development Canada for the Canadian Department of National Defence [11]. Within the scope of this report, multiple developments in the fields of hull monitoring, digital twins, and risk based inspection were put under consideration in order to estimate the value of information. The utilization of various data in order to calculate fatigue damage accumulation is proposed. More specifically, according to the study, heading data, hindcast data, loading condition and ship speed can be combined in order to generate the operational profile of the ship. This operational profile can then be coupled with a hydro-mechanical analysis of the vessel in order to calculate the fatigue damage accumulation. It is also suggested that digital twin technologies should be updated with in-situ measurements to improve the accuracy. Finally it is concluded that a digital twin coupled with inspection and maintenance data can be used in order to optimize the maintenance practises. The above mentioned study focuses on fast displacement vessels. However, similar approaches as the ones suggested have been adopted in heavy applications in the marine and offshore industry.

Offshore Applications Numerous research studies have been focused on the development of intriguing digital twin applications for structural health monitoring of offshore structures. Tygesen et al. [12] have examined the possibility of developing a digital twin that can adapt to the changing behaviour of a marine structure throughout its lifetime. They called this application a "True Digital Twin". The concept of this true digital twin is to couple the measurements that are performed by a structural health monitoring system with the behaviour of the digital twin model. Such a coupling will allow the evaluation of the performance of the digital twin against actual measurements and the recalibration of the model in order to minimize errors. In this way it will be able to capture the actual behaviour of the structure under study. The development of the above-mentioned digital twin has been performed in three levels using a fixed offshore platform as a case study. In the first level, the behaviour of the structure was screened and then compared with the behaviour of an uncoupled finite element model. The comparison was based on the correlation of the measured and predicted dynamic structural response. The real modal parameters of the structure were determined using accelerometers and stochastic sub-space identification [13]. In the second level, the FE model was updated in order to mimic the behaviour of the actual structure. The updating was performed using Bayesian FE model updating [14] and utilizing the real modal parameters that were identified in the first level. Levels 1 and 2 made sure that the actual structural behaviour was captured. However, in order to develop a digital twin able to capture the fatigue performance of the structure the wave model should be calibrated as well. This was done in Level 3 using data collected form strain gauges and accelerometers. This data was used to calibrate the drag and inertia hydrodynamic parameters of the Morison equation [15].

Boutrot et al. proposed a methodology for the lifetime extension of aging offshore units based on 3D digital twin models [16]. The focus of proposed the methodology under study is put on the structure degradation mechanisms of fatigue and corrosion. The digital-twin used, is a 3D model of the monitored structure with attached metadata. This metadata can be design factors, Inspection Maintenance and Repair factors, Risk Based Inspection (RBI) factors and condition measurements. This metadata is frequently updated to ensure that the digital twin model corresponds to the state of the physical asset. The metadata are utilized in order to calculate the corrosion rate and remaining material, and also to create accurate FEM models of the structure. The above mentioned 3D digital twin is used as the cornerstone of an Asset Integrity Management (AIM) platform.

Marine Applications Many digital twin applications have also been developed by Det Norske Veritas (DNV) as indicated in reference [17]. One of them, called "Nerves of Steel", has been developed in order to limit unexpected failures on a ship hulls due to wave-induced damage. The digital twin model combines sensor data with wave loading and response models in order to calculated the risk of hull damage caused by waves. Furthermore, by constantly updating the hull model it is able to predict experienced fatigue loads, extreme loads, and seakeeping of the monitored vessel. It has already been applied on a range of container ships, FPSOs, semi-submersibles, flare towers, pipelines, and fixed platforms [18].

Another novel application of digital twin technology for SHM has been proposed by Hulkonnen et al. [19]. In this study, a digital twin was developed in order to evaluate the remaining fatigue lifetime of ship hull structures. The proposed digital twin uses big data regarding the position of the ship and the weather in order to evaluate the operational conditions of the ship. The position data is collected from Automatic Identification System (AIS) messages, while the weather data is collected from multiple sources. The collected data are used in order to recreate the wave spectrum corresponding to the sea area where the ship is operating. When this wave spectrum is combined with the design Response Amplitude Operators (RAO) of the ship that were calculated with the 3D panel method, spectral fatigue analysis can be performed leading to the estimation of the resulting damage.

Other Applications Digital twin technologies for SHM are not restricted to maritime applications. Extensive research is focused on the development of digital twins for structural applications from the aerospace, automotive, civil, and material science sectors. A near real-time structural health monitoring digital twin, has been developed by NASA for probabilistic damage prediction of aerospace structures [20]. The fact the the digital twin is operating at almost real time makes a physics-based simulations framework prohibitive due to the high computational cost related to it. The computational cost is even higher when uncertainty quantification needs to be integrated to the analysis. In order to bypass the problem of high computational times, a different approach was adopted. The analysis was posed as an inverse problem model based-diagnosis [21]) and then a probabilistic solution was formed using Bayesian statistics. The inverse problem in this case was the estimation of parameters that characterize the existence of damage in the structure using in-situ measurements (strain,vibrations etc.). The Bayesian solution is the likehood of damaged conditions based on the measured response. Because this probability distribution cannot be calculated analytically, it was generated using sampling algorithms based on the Markov Chain Monte Carlo algorithm [22] and optimized for low computational cost.

Krobot Z. et al. examined the use of a digital twin model to support the maintenance of military automotive vehicles [23]. In the scope of this study a digital twin was deployed to monitor the operational condition of the automatic transmission system of an LMV Iveco military vehicle. Using Matlab Simulink tool, an accurate model containing all the operational parts of the automatic transmission was generated. This model using as input the rotational speed of the primary shaft of the gearbox can estimate the rotational speed of the output shaft of the transmission. This was verified using operational data from three different vehicles. Coupling the developed digital twin with data loggers and a telemetry module allows the service crew to perform diagnostics and maintenance while having the vehicles actual operational profile available.

In the civil engineering sector, an innovative study was performed by Ye C. et al. about the feasibility of using digital twin technology to model the structural health of bridges [24]. The study is focused on real-time data management using Building Information Modeling (BIM), physics-based approaches, data-driven approaches and data-centering approaches [25] which are essentially a combination of physics-based and data-driven approaches. Using this analysis as a theoretical foundation, the framework for development of a digital twin model for SHM is proposed. The suggested framework is to use a Gaussian Process in order to minimize the errors of an FE model taking advantage of real-time data that is collected by monitoring systems placed on the bridge.

The feasibility of a rather radical use of digital twin technologies is put under study in references [26] and [27]. The first part of the study focuses on the prediction of fatigue lifetime before the accumulation of observable damage. In order to achieve this, fracture mechanics and material science concepts are put under consideration. More specifically, the predictive capability is enabled by a digital twin model that predicts the crack growth from initiation until failure using micro-structure based modelling on multiple scales. To achieve this, two modules are used in-line. The first module operates in the microstructurally small fatigue crack scale and explicitly incorporates the variability in microstructure. It generates multiple statistically equivalent microstructures, of the material under study. Using 3D FE models of the generated microstructures the scatter in crack growth curves is estimated and then used to evaluate probability density functions (PDFs) of parameters that are forward-propagated to the second module of the digital twin. The second module works in the microstructurally large fatigue scale and is based on linear fracture mechanics principles. This model simulates the fatigue crack growth using Paris's law taking into consideration the uncertainty in the crack growth rate that is propagated from the first module. The transition from the one module to the other happens at a predetermined crack length. Both models are calibrated using data collected from fatigue tests on specimens [28] of the material under study. The second part of the research is focused on a generalized method for the reduction of uncertainty using in-situ diagnostics and prognostics in a probabilistic fatigue lifetime prediction framework. The method suggested at the second part can only be utilized after size of the fatigue crack becomes observable.

1.4.2. Fatigue Analysis Using Machine Learning

The scientific interest for Machine Learning applications and Artificial Intelligence (AI) in general has skyrocketed in recent years. The reason behind the re-ignition of interest in this specific field of study is no other than the rapid increase in computational power, that enables the use of algorithms that were once considered as utopian. Nowadays, machine learning has conquered the industry and is at the heart of many high-tech products and applications. Under this perspective, maritime applications could be no exception.

1.4.2.1 Predicting Fatigue Damage with Machine Learning

Nowadays, machine learning has applications in most field of studies and fatigue analysis could be no exception. An abundance of applications can be found both in the marine and offshore energy industries. A wide range of applications has been examined in order to define the current state of research regarding machine learning for prediction of fatigue damage. Applications from other industries such as aerospace and civil engineering have been taken under consideration as well.

Offshore Applications Wong and Kim propose the use of an artificial neural network (ANN) in order to predict vortex-induced fatigue damage of top tensioned risers [29]. The database required for the training and testing of the ANN was built using simulations performed with Orcina's Orcaflex software. Six features were selected as input for the ANN. The features were sampled using Latin Hypercube Sampling (LHS) [30]. The ANN model consisted of one input layer, one hidden layer and one output layer. It was tuned with respect to the activation function, the optimization algorithm, the cost function, the mini-batch size, the number of neurons, and the learning rate. The predicted fatigue damage was very close to the damage value estimated using Orcaflex. However, low prediction error is expected in applications using a simulation-generated database, due to the consistency of the data.

Another interesting framework for fatigue analysis of steel catenary risers has been developed by Hegazi et al. [31]. In the scope of this framework, the use of an ANN is suggested in order to predict the transfer functions needed to translate the wave spectrum to a stress spectrum. The input of the model are features regarding both the sea state and the geometry of the risers. The ANN practically substitutes the RAO transfer functions that are commonly calculated through hydro-mechanical simulations [32]. The results of this new framework where compared against the results of numerical simulations resulting in errors less than 5%. The main benefit of the developed framework is once again a significant reduction in terms of computational time.

Taking a more industry-focused approach Engebretsen et al.[33] propose a methodology for the estimation of fatigue damage accumulation using field measurements and machine learning algorithms . In this approach, field measurements of operational and environmental data are used in order to train a machine learning model that predicts the heading and motions of the FPSO that is used as a case study. The performance of five different machine learning models was evaluated. The algorithms tested were:

- Extra Randomized Tree
- Random Forest
- Decision Tree
- Gradient Boosting
- K-Nearest Neighbours

After selecting the best performing ML algorithm, a fatigue counter is established by coupling the predicted heading and motions with a fatigue analysis software such as BFLEX that relates vessel motions to local stress at the flexible riser. Using this methodology, a live fatigue counter can be built that provides valuable insight in terms of fatigue lifetime consumption.

A hydrid model of an ANN and FEA has been developed for short-term wave-induced fatigue analysis of flexible pipes [34]. The main objective of this study was to reduce the high computational time required by time domain stochastic wave dynamic analyses but proposing a surrogate model. The model is called hybrid since it combines FEA with ML. It uses FEA and analytical calculations to create short time series of the structural response and motions. Then using an ANN, it predicts the rest of the response time series. Fatigue damage accumulation is calculated from the response time series using Rainflow Counting [35] and Miner's rule [36]. The ANN consists of three fully connected dense layers in total. Because of the dynamic nature of the problem, an auto-regressive neural network with exogenous inputs [37] was used in order to account for the memory effect of the system. A free-hanging flexible pipe connected to a spread-moored FPSO was used as a case study. The model was tested for 88 sea-states each one of them requiring 4000s of FEA simulation time, while the hybrid model needed about 500s to perform the same simulation. The authors suggest that there is more to be gained in longer simulations.

The same model was also successfully applied to steel catenary risers [38]. A steel catenary riser connected to a semisubmersible platform was used as a case study. The resulting error in terms of fatigue damage prediction was far below 1%. The hybrid model was tested on a 3-hour time series generated using FEA. The required computational time of the hybrid model was 20 times lower compared to the time required for the full FEA.

Due to the intended transition towards renewable energy, a lot of research has been focused on offshore wind turbines. Müller et al.[39] studied the fatigue damage assessment of floating offshore wind turbines using Response Surface Modeling (RSM) coupled with an ANN and LHS. The proposed method is a computationally inexpensive alternative to time-domain simulations. Initially, a probabilistic analysis was performed for the environmental parameters that affect damage accumulation of a floating offshore wind turbine (wave height, wind speed etc.). Then using LHS, samples of these environmental parameters were generated. Using simulation software, the damage equivalent loading for these samples was calculated. Afterwards, the ANN was trained to predict the damage equivalent loading using the dataset generated. Taking advantage of the low computational cost of the ANN, the response surface was created. It is noted that due to the uncertainty inherent to the ANN, the resulting response surface is slightly different between different runs of the method. By increasing the number of samples in the dataset this uncertainty is minimized. The algorithm was tested on a hypothetical semi-submersible concrete wind turbine that is a combination of the DTU10MW [40] reference turbine with the SWE triple-spar developed in the INNWIND EU project [41].

Another alternative to the time-domain approach for the fatigue analysis of offshore wind turbines is provided by Kim et al. [42]. In order to reduce the required computational time, a frequency-domain method is introduced. In the proposed method the wave fatigue damage is described through 'conventional fatigue analysis' because the wave-induced RAO is invariant with the wave height and wind speed if the transfer function is calculated based on linear wave theory. However, for calculation of the wind fatigue damage, a different approach is suggested. This reasoning behind this alternative proposal is that the wind-induced RAO is different for different mean wind speeds. In order to avoid time-costly simulations to obtain the RAO for each and every encountered wind speed, the use of ANN is proposed. The role of the ANN is to predict a correction factor that scales a reference wind-induced RAO for different wind speeds. For the WWHybrid system used as a case study, the number of required simulations was dropped from 41 to 7 due to the introduction of the ANN.

Marine Applications The application of machine learning for fatigue analysis applications is not restricted to the offshore energy industry. The use of a deep-learning multi-layer ANN for fatigue accumulation prediction of a tanker's cargo-hold is suggested by Lee et al. [43]. In this study, the residual fatigue life of the structure is defined as the difference between the calculated fatigue life using simulation software and the predicted fatigue life. In order to predict the fatigue life of the structure, a multi-layer ANN is utilized in order to predict the strain time series at the examined location. The ANN takes as an input the acceleration, the roll, and the pitch ship motion as well as the previous predicted strain value. The strain time series is then translated to damage using a cycle counting method. The model was trained using data generated from simulations. The study concludes that using this framework only short term forecasts (order of days) can be produced accurately.

With a rather fundamental approach, Christiansen et al. examined the possibility of reducing the training error of an ANN that predicts the tension forces on an anchor chain by using error an error function designed for this specific purpose [44]. In the scope of this study, an ANN is trained using different error functions based on the Minkowski-R error [45] in order to evaluate the effect of the selected error function on the performance of the model. The performance of the model is evaluated by comparing the predicted fatigue accumulation with the fatigue accumulation that is estimated using time-domain simulation software for the same sea states. The model used as a case study is based on a simplified version of the multi-layer ANN developed for the prediction of the dynamic response of mooring lines [46]. The study concludes that using weighted error functions deteriorate the performance of the model, while using higher order error functions only slightly increases the performance of the model.

Other Applications Novel applications of machine learning algorithms in the field of fatigue analysis can also be found in other industries. The performance of machine learning algorithms in the quantification of fatigue damage in structural details of slender coastal bridges is put under examination by Lu et al.[47]. The machine learning algorithms under examination are:

- Support Vector Machine
- Artificial Neural Network
- Random Forest
- Gradient Boosting

The input of the models are parameters related to the wind, wave, and vehicle loading of the bridge under study. The output is the resulting fatigue damage. The models are trained using data generated through FEA. As it is common with ML models trained on simulation data, all the algorithms achieve high accuracy. The best performing algorithm in terms of error and variability minimization for the application under study is the Gradient Boosting. Using the trained ML models, the fatigue damage PDF is also generated and then a Monte Carlo structural reliability analysis is performed.

1.4.3. Load Reconstruction using In-Situ Measurements

Calculation of fatigue accumulation in marine and offshore structures is subjected to many uncertainties. A major source of uncertainty is the hydro-mechanical modeling of the structure. Adopting wave load modelling parameters from codes and standards frequently leads to overly conservative predictions of fatigue lifetime. Subsequently, in order to achieve a high fidelity digital twin, in-situ measurements coupled with load reconstruction techniques should be used to calibrate the actual loading on the structures. The research regarding load reconstruction is divided in two large fields: static (and quasi-static) load reconstruction and dynamic load reconstruction.

1.4.3.1 Static and Quasi-Static Load Reconstruction

The methodologies that have been developed to reconstruct static and quasi-static loads are based on strain gauge measurements. The measurement data that is collected can be used to reconstruct the loads as long as the superposition principle holds. The precision of the reconstruction is dictated by both the number and the location of the strain gauges [48].

The framework that is used as a basis for later developments in the field was developed in 1991 by Masroor and Zachary [49]. In this analysis, the generation of a coefficient matrix is suggested in order to reconstruct the loads applied on the structure. This matrix is used to map measured strain values to corresponding applied load magnitudes. The coefficient matrix is generated through an experimental procedure, in which the loads and the corresponding deformation of the structure are measured. The introduction of error due to measurement inaccuracy and simplifications is also examined within this study.

Based on this framework, Wickham et al. [50] proposed a methodology to determine the optimum location and angular orientations for strain gauge placement. The methodology utilizes the k-exchange algorithm [51] in order to generate the D-optimal design that leads to the best reconstruction of the loads. The number of strain gauges was selected through an arbitrary threshold for maximum variance in the results. A finite element model of a C-spring was used as a case study.

Despite the fact that the framework for the reconstruction of static and quasi-static loads using a coefficient matrix was developed in 1991 recent applications of it can be found in current research. Riccioli F. [52] uses a retrofit of this framework for piezoelectric sensors in his research. Using a spectral elemental model [53] of the problem under study, he maps the measured voltage of piezoelectric sensors to forces through the sitffness matrix of the model and a matrix that maps measured voltage to displacements. He then uses Lagrange polynomials to translate the forces into an acting pressure. The location of the sensors in order to provide optimal results is also optimized using the Gauss-Lobatto-Legendre scheme [54].

1.4.3.2 Dynamic Load Reconstruction

The reconstruction of dynamic loads is a far more complicated field of study compared to the reconstruction of static and quasi-static loads. Chang et al. [55] developed a framework to reconstruct unknown time-dependent external forces based on the response of the structure. The inherent error in response measurement usually makes the reconstruction of forces in time-domain an ill-posed problem. In order to solve this problem, the use of the implicit Landweber method is proposed. The load identification and reconstruction was achieved by utilising the concept of response sensitivity and a genetic algorithm strategy. The effectiveness and robustness of the method was demonstrated by means of both numerical simulations and physical tests. The selection of proper convergence criteria was highly dependent on the level of noise in the data.

A different approach for the load reconstruction for advanced grid-stiffened composite plates is suggested by Zhang et al.[56]. The load reconstruction problem here is translated to an optimization problem. Initially a forward state-space model is generated using the governing equations of the structures response. Then the load reconstruction is performed by minimizing the error between the measured structural response and the forward model. This minimization is performed using an outer loop that searches for the position of the load and an inner loop reconstructing the load history. The accuracy and effectiveness of the method was verified

by comparison of the results to both FEM simulations and physical experiments.
Using a similar approach, Hashemi and Kargarnovin [57] tried to identify the impact force on a simply supported beam based on the measured response . The inverse problem is once again converted in a minimization problem between a numerical model of the structure and the measured response. The minimization problem is solved through the use of a genetic algorithm primarily formulated by Holland [58]. The effects of measurement noise and measurement location on the quality of the reconstruction are examined as well. The study concludes that the proposed approach is insensitive to the location of the measurements and adequately robust to measurement noise.
Another study suggests the use of an inverse FEM procedure to reconstruct the load and shape of a membrane using strain-measurements [59]. The inverse problem of structural displacement reconstruction is addressed through the application of a variational formulation [60]. With the displacement field available the pressure distribution on the membrane is reconstructed. The proposed method was verified using both the results of a direct simulation and experimental data.

1.4.4. Structural Reliability Analysis

Every aspect of offshore and marine engineering design contains a certain extent of uncertainty consideration. This uncertainty is related to environmental exposure, loading, material properties etc. The theoretical framework that can be utilized to consider these uncertainties and translate them to a comprehensive decision scheme is called reliability analysis [61]. As Melchers and Beck [62] define it structural reliability analysis is "the calculation and prediction of the probability of limit state violation for an engineered structural system at any stage during its life". Common limit states for structures are:

- Ultimate Strength Limit State
- Fatigue Strength Limit State
- Serviceability Limit State

Structural reliability methods are categorized in different levels according to their complexity. These levels are [63]:

- Level I: First Moments Methods
- Level II: Second Moment Methods
- Level III: Full Probabilistic Methods
- Level IV: Full Probabilistic Methods coupled with Cost Benefit Analysis

The computational complexity of the method increases as the level of the method increases.

1.4.4.1 Structural Reliability Analysis Applications

In the context of this literature study the focus was put on the more sophisticated methods of Level II and III as they provide the most accurate results in terms of probability of failure.
Cardoso et al. [64] examined a full probabilistic method that combines Monte Carlo simulation with an Artificial Neural Network. The proposed method uses the neural network in order to predict the value of the limit state function taking the random variables under study as input. The neural network is trained on a simulation-generated database. This way the computational cost of the Monte Carlo simulation is significantly reduced. The validity of the method is proved through structural examples. The study concludes that there are considerable benefits to be gained by the coupling of Monte Carlo Simulations and Neural Networks.
A similar approach was suggested by Budzik and Potrzeszcz-Sut [65]. Once again a neural network was used in order to generate explicit neural limit state functions that substitute the conventional implicit limit state functions. The structure under examination was a spatial truss susceptible to stability loss. The random variables were uncorrelated and the ANN was trained on simulation data. The Hansofer Lind method [66], which is a First Order Reliability Method (FORM), was coupled with the ANN to predict the probability of failure of the structure. In order to verify that FORM can produce accurate predictions, it was tested against a second order reliability method and a Monte Carlo simulation reliability analysis. The maximum relative prediction

error of these methods was 3.2%. Because of this low error the study concludes that the use of FORM can lead to accurate predictions.

Structural reliability analysis can also be transformed to an optimization problem. Such an approach is proposed by Elegbede [67]. In the scope of this study, reliabilty analysis was performed through the minimization of the Hansofer-Lindt reliability index. In order to apply the suggested methodology. the random variables had to be transformed to normal Gaussian variables. This can be done using the Rosenblatt [68], Nataf [69], or Plain Linear transformation. The minimization of the Hansofer-Lindt index was posed as a constrained optimization problem with the limit function being equal to zero as the acting constraint. The Particle Swarm Optimization (PSO) algorithm[70] was used as the optimization engine. This selection was based on the consideration that genetic algorithms are generally good in finding global optimums unlike gradient based algorithms. The proposed methodology was tested on a number of structural examples.

This methodology is pushed even further on the optimization front by Zhao et al.[71]. In the suggested methodology, the reliability analysis is converted once again to an optimization problem through the minimization of the reliability index [72]. The major difference now is that the PSO algorithm is coupled with a chaotic system to avoid the selection of initial parameters that can lead to an ill-posed optimization problem. This coupling is achieved by updating the PSO parameters using a system that exhibits chaotic dynamics. The chaotic system used in this study is a special set-up of the logistic equation [73]. The feasibility of the method is verified through numerical examples.

1.4.5. Literature Review Conclusions

The extensive on-going research is indicative of the high potential of using digital twin systems for structural health monitoring. The main advantage of digital twin systems is that, contrary to conventional health monitoring systems, they can also be used to simulate past or future states of the structure during which measurements are not available. Through the examined literature, it becomes clear that digital twin models can be based on different modeling techniques such as probabilistic models, FE models, analytical models etc. Most of the studies that were examined are based on a physics-based model using real data only to calibrate the model. It also becomes evident that a digital twin system should be able to recreate the behaviour of the actual structure utilizing in-situ measurements.

Looking through the recent developments of structural health monitoring ML applications, it becomes evident that ANNs are selected in most applications since they perform better in highly non-linear problems. After carefully examining the applications of machine learning algorithms for fatigue analysis purposes, one can conclude that the ML algorithms are used as a inexpensive computational substitute of some part of the fatigue analysis methodology applied. Moreover, the majority of these algorithms are trained on simulation data. This means that any error associated with the simulation model is propagated to the ANN. Furthermore, achieving accurate predictions of simulation results is something to be expected since the results are based on deterministic equations. Measurement data on the other hand contain measurement noise and usually deviate from applied engineering models. These facts raise a question whether an ANN trained on measurement data could predict fatigue accumulation in a consistent and accurate manner.

Taking into consideration the findings of this literature review, it can be concluded that dynamic load reconstruction is a vibrant field of research. However, the applications that are examined require a dense cloud of measurements in order to be accurate. If dense measurements are not available, a quasi-static load reconstruction approach should be considered.

In the field of structural reliability analysis, the utilization of ANNs in order to reduce the computational effort of reliability analysis methods is a common occurrence, especially in Monte Carlo simulations. However, when the probability of failure is low, Monte Carlo algorithms are not desirably efficient. In such cases FORM methods are preferred. When FORM methods are coupled with genetic algorithms they require lower computational time and their ability to locate global optimum points is also satisfactory.

1.5. Knowledge Gap

As it has been observed during the Literature Review various applications of digital twin technologies have been developed for structural health monitoring purposes. One generally common aspect in all the applications was a framework for fatigue accumulation estimation which utilizes physics-based models using in-situ measurements as means to calibrate these models.

Moreover, during the Literature Review, it became evident that there have been multiple attempts to utilize machine learning algorithms for prediction of fatigue damage accumulation. Almost the entirety of the examined applications were once again based on physics-based models using machine learning to substitute (small) parts of the fatigue analysis process. Moreover, all of the studied applications that were able to predict future damage accumulation have been trained using data-sets generated through numerical simulations. The combination of the above-mentioned observation leads to the following question:

Can the digital twin technology be enhanced with machine learning to accurately monitor the structural health of marine structures in terms of loading and fatigue damage accumulation?

In order to answer the above-mentioned question in a consistent and accurate manner it was broken down to several research questions. The research questions have been formulated in a way that they fill the knowledge gap while contributing to the general objectives of the Monitas Group.

- Is it feasible to develop a digital twin model based on machine learning algorithms?
- How accurately can machine learning algorithms predict fatigue damage accumulation trained on big data?
- Can a structural reliability analysis module be integrated into a digital twin model?
- How can the collected operational data be used to monitor operational conditions of marine structures?

The methodology described below consists a framework within which these question can be answered thus filling the corresponding knowledge gap.

1.6. Methodology

A brief overview of the proposed methodology is illustrated in Figure 1.3. The input of the digital twin model will be the wave energy spectrum of the environment where the structure is operating. This spectrum can either be measured using buoys or reconstructed through other measured data (hindcast etc.). Afterwards, the features required by the machine learning (ML) algorithms will be extracted from the wave spectrum. Two machine learning models will be utilized. One for the prediction of damage accumulation corresponding to the wave spectrum and the other one will predict the frequency corresponding to the maximum stress power density. Both models will be trained using data that is acquired by strain gauges installed on the structure. For the task of damage prediction, an Artificial Neural Network (ANN) will be used. This selection is based on the fact that ANNs perform better than other ML algorithms in highly non-linear problems. The performance of multiple ML algorithms will be tested in order to select the optimal model for the prediction of the peak stress power density frequency.

In order to assess the feasibility of an improved tool that can be utilized for the justification of lifetime extension a reliability analysis method will also be developed. Taking the examined literature into consideration a methodology based on a Chaotic Particle Swarm Optimization (CPSO) algorithm will be coupled with a FORM and an ANN that replaces the conventional limit function. The ANN that will be used is the one previously trained to predict fatigue damage accumulation. The selection of a FORM coupled with a CPSO algorithm is grounded in the fact that the algorithm should be efficient even for low failure probability. It is expected that this method will require low computational effort while providing accurate results.

Finally a framework that enables direct load reconstruction based on strain measurements will be developed. The aim of this framework is to provide better insight in the operational conditions of the structure and enable comparison with the design conditions in order to improve future designs. Because monitoring systems with dense measurement points are not common on-board marine structures an inverse FEM quasi-static approach based on the methodology developed by Masroor and Zachary [49] will be pursued.

1.7. Scope and Objectives

The methodology that is proposed consists an attempt to minimize the cost related to preventing unexpected fatigue failures through the development of a digital twin that accurately predicts fatigue accumulation in marine structures combining big data with machine learning techniques. This effort is motivated by the high uncertainty that is inherent in the fatigue calculations of marine structures as well as the high risk associated

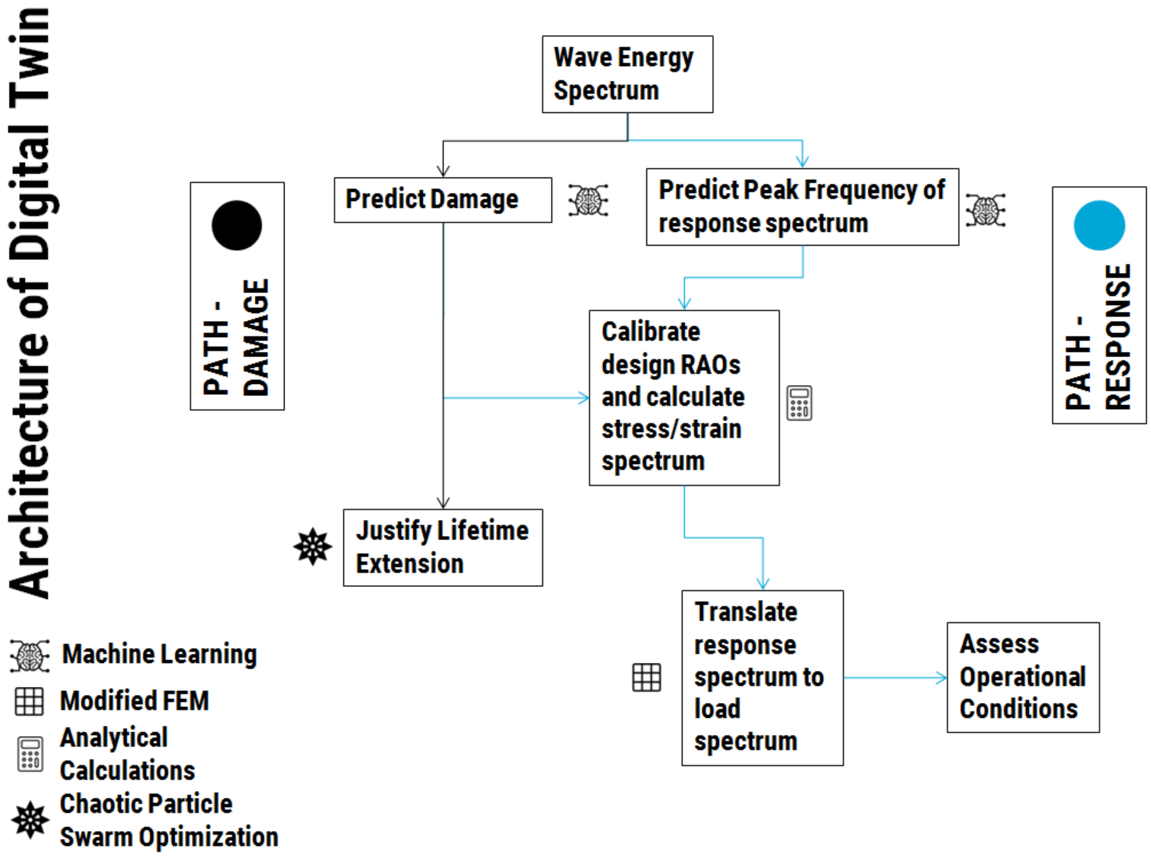


Figure 1.3: Architecture of Digital Twin

with on site repairs at the remote and potentially harsh environments where such structures are operating. By increasing the accuracy of fatigue damage accumulation predictions this study aims to contribute to the reduction of the operational expenses of marine structures. The main objectives of this project can be summarized in the following points:

- Evaluation of feasibility of development of a high fidelity digital twin model for the prediction of fatigue lifetime consumption of marine structures using operational data and machine learning algorithms
- Assessment of the operational condition by means of load reconstruction using in-situ measurements and finite element modelling
- Development of a method capable of probabilistic lifetime estimation of the structure utilizing available data and the developed fatigue damage accumulation prediction module
- Validation of the above mentioned modules using an operating Field Production Unit as a case study

1.8. Report Outline

In the initial parts of the report the theoretical basis of the methodology is set. In Chapter 2 the ML algorithms used in the context of this project and their operational principles are shortly described. Chapter 3 set the basis of the quasi-static load reconstruction module. It presents a reformulation of the classic FEM formulation that can be utilised for direct load reconstruction using strain measurements. Finally Chapter 4 is focused on the reliability analysis module. In its context, the CPSO algorithm is presented, as well as the approach of reliability analysis as an optimization problem.
The later parts of the report are focused on the feasibility study of the modules utilizing the theoretical basis that was set in the previous chapters. Chapter 5 describes the development procedure of the ML models responsible for the prediction of the fatigue damage accumulation and the frequency in which the stress

spectral density is maximum. The procedure is presented using data collected during the operation of an FPU operating in West African coast,in the context of Monitas JIP. Chapter 6 the algorithm responsible for quasi-static load reconstruction and a sensitivity study regarding the performance of the algorithm. Finally, Chapter 7 presents the framework in which a reliability analysis can be performed for a marine structure coupling the theory of Chapter 4 and the damage accumulation prediction model built in Chapter 5. The framework is applied on structural details of the West Africa FPU.
Finally, in Chapter 8 the conclusions of the research are presented as well as suggestions for future research.
The data acquisition, processing and storage process is presented in Appendix 1.

Chapter 2

Machine Learning Regression Algorithms

The scientific interest for Machine Learning applications and Artificial Intelligence in general has sky-rocketed in recent years. The reason behind the re-ignition of interest in this specific field of study is no other than the rapid increase in computational power, that enables the use of algorithms that were once considered as utopian. Nowadays, machine learning has conquered the industry and is at the heart of many high-tech products and applications. Under this perspective, maritime applications could be no exception.

2.1. What is Machine Learning?

The term machine learning was initially used by Samuel Arthur, a pioneer in computer gaming and artificial intelligence, in 1959 [74]. Some research was focused on machine learning during the 1960s and 70s but was later abandoned due to the computational complexity of the developed algorithms. The recent machine learning 'tsunami' has started in 2006 when Geoffrey Hinton published a method capable of training deep artificial neural networks [75]. Machine Learning is essentially a sub-space of AI. Machine Learning systems are rather trained and not explicitly programmed [76]. As defined by IBM "Machine Learning is a branch of artificial intelligence focused on building applications that learn from data and improve their accuracy over time without being programmed to do so" [77]. Machine Learning and especially Deep Learning algorithms are greatly benefited by the existence of large amounts of available data. They can train themselves to achieve results far superior to the ones of conventional data fitting algorithms. This high performance of machine learning when big data is available can prove highly beneficial to the development a high-fidelity digital twin model.

2.2. Artificial Neural Network

The Artificial Neural Network is one of the most widely used Machine Learning Algorithms. As its name implies it is based on biological learning systems that are composed of complicated networks of interconnected simple units.

These simple units in both biological and artificial neural networks are called neurons. In biological networks neurons are the cells responsible for reception,transmission and processing of signals. The functions they perform on a unit level are very simple. However, when they are combined they produce very powerful networks. The same goes for artificial neural networks. Each neuron receives one input(x_i) from every neuron that it is connected to and then goes on to calculate the weighted sum of these inputs using pre-assigned weights(w_i):

$$z = \sum_{i=1}^{n} w_i x_i \tag{2.1}$$

Then the weighted input summation is fed to an activation function that provides an output y. The operation of such neurons, also known as McCulloch-Pitts neurons [2], is illustrated in Figure 2.1. The selection of the application function is of utmost importance as it defines the computational properties of the neural network. The application function should have a monotonically increasing behaviour within a given range and have a constant value outside of this range. The most commonly used activation function, which is also used in the scope of this study, is the Rectified Linear Unit (ReLU) function [78]. The neurons of an artificial neural network are arranged into layers. The first layer is called the input layer, the intermediate ones hidden layer and the final one output layer. In most neural network applications dense-layers are used which means that all the neurons of one layer are connected to all the neurons of the previous and the next level. The neural networks with more than one hidden layer are called multi-layer networks. The architecture of a neural network with a single hidden layer can be seen in Figure 2.2

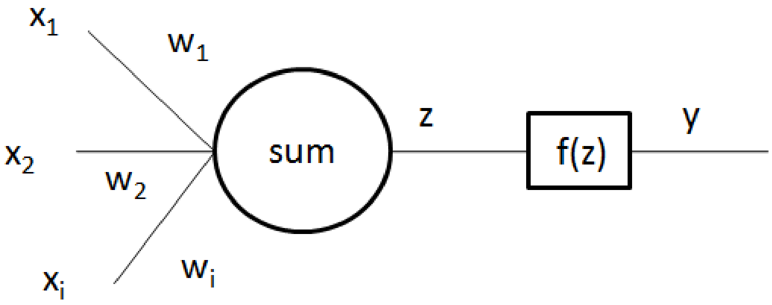


Figure 2.1: McCulloch-Pitts neuron[2]

The process of estimating the values of the weights that are connecting the neurons of a neural network is referred to as training the neural network. Artificial neural networks are trained using the back-propagation algorithm [79]. The algorithm consists from a forward and a backward step. In the forward step, one input is selected and its effect is propagated throughout the network until a response is produced. In the backward step the error between the produced and the desired response is calculated and then back-propagated to the start of the network adjusting the connection weights. The procedure is repeated until a stopping criterion is met. Different function can be used to calculate the error, with the most common of them being the mean squared error [44]:

$$E(t) = \frac{1}{2}\sum_{i=1}^{n}(y_i - v_i)^2 \tag{2.2}$$

The minimization of the error is achieved using gradient based optimization functions. A well trained network is able to provide accurate predictions for input which is not contained in the training set.

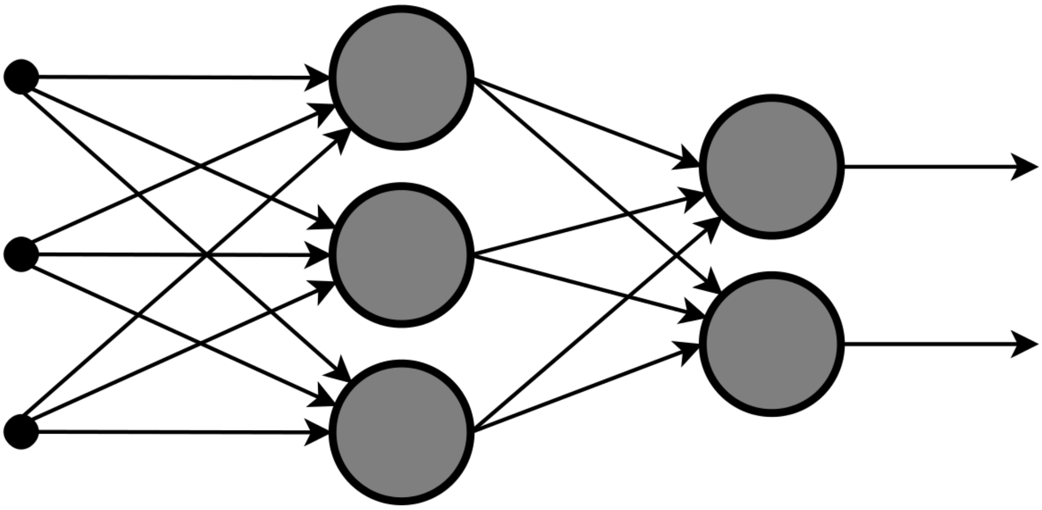


Figure 2.2: Neural Network Architecture

2.3. Support Vector Regression

Support vector machines (SVMs) are very powerful machine learning models. They are most commonly used for classification problems, but they can also be used for linear and non-linear regression problems as the one under examination. The regression problem is a generalization of the classification problem [80], in which the model instead of returning an output from a finite set of possible outputs, it returns a continuous-valued output.

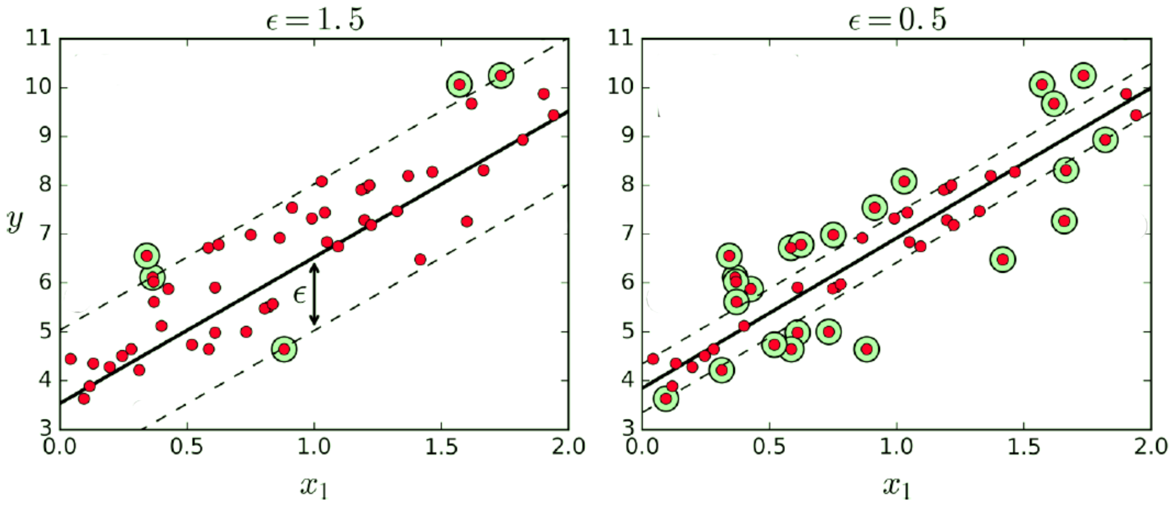


Figure 2.3: Linear support vector regression problem[3]

In order to solve binary classification problems support vector machines formulate them as convex optimization problems [81]. The optimization problem entails finding the maximum margin separating the hyperplane, while correctly classifying as many training points as possible. In a given dot product space a hyperplane can be described as:

$$[w] \cdot [x] + b = 0 \tag{2.3}$$

where [w] is the orthogonal vector to the hyperplane. The separating hyperplane can be estimated by minimizing:

$$\frac{1}{2}||[w]||^2 \tag{2.4}$$

Because of their sparse solution and their good generalization capability SVMs can be adapted to address regression problems. The generalization of support vector machines to solve regression problems is achieved through the introduction of an ϵ-insensitive region around the decision function [82]. Taking advantage of Vapnik's ϵ-insensitive approach, a flexible tube of minimal radius,called the ϵ-tube, is formed symmetrically around the calculated decision function as it can be observed in Figure 2.3. This means that when the error has absolute values lower than a pre-defined threshold it is ignored. In this manner, points located outside the tube limits are penalized, while those located within the tube do not receive penalty. The optimization problem is reformulated in order to estimate the ϵ-tube that best approximates the continuous function. The ϵ-tube function has the following shape [47]:

$$C \sum_{n-1}^{N} (\xi_n - \hat{\xi}_i) + \frac{1}{2}||[w]||^2 \tag{2.5}$$

Under the constraints:

$$t_n \geq y(x_n) - \epsilon - \hat{\xi}_n \tag{2.6}$$

$$t_n \leq y(x_n) + \epsilon + \xi_n \tag{2.7}$$

$$\xi_n \geq 0 \tag{2.8}$$

$$\hat{\xi}_n \geq 0 \tag{2.9}$$

Where ξ_n and $\hat{\xi_n}$ are slack parameters and C is a regularization parameter. Through the introduction of Lagrangian variables and the elimination of the corresponding variables the dual problem is given by maximizing:

$$L(a,\hat{a}) = -\frac{1}{2}\sum_{n-1}^{N}\sum_{m-1}^{N}(a_n-\hat{a_n})(a_m-\hat{a_m})k(x_n,x_m) - \epsilon\sum_{n-1}^{N}(a_n+\hat{a_n}) + \sum_{n-1}^{N}(a_n-\hat{a_n})t_n \tag{2.10}$$

Under the constraints:

$$0 \leq a_n \leq C \tag{2.11}$$

$$0 \leq \hat{a_n} \leq C \tag{2.12}$$

in which the kernel $k(x,x') = \phi(x)^T\phi(x')$ and $a_n, \hat{a_n}$ are Lagrange multipliers. The kernel function is used in order to transform the data to higher dimension space where it is easier for them to get separated. The most common kernel functions are:

- Polynomial function
- Gaussian function
- Radial base function[83]

Solving the dual problem the regression function can be obtained as follows:

$$y(x) = \sum_{n-1}^{N}(a_n-\hat{a_n})k(x,x_n) + b \tag{2.13}$$

2.4. Gradient Tree Boosting

Boosting or hypothesis boosting refers to ensemble methods that combine several weak models in a strong model. The idea is to train predictors sequentially and each predictor tries to improve the performance of the previous predictor [3].

One of the most used boosting methods is Gradient Tree Boosting. Gradient tree boosting is essentially a combination of Decision Trees [84], which are used as a base estimator, which the principles of hypothesis boosting. In gradient tree boosting each weak learner (decision tree) is trained on the error of the previous one. The mathematical formulation of the prediction of a gradient tree boosting algorithm is [85]:

$$\hat{y_i} = F_M(x_i) = \sum_{i=1}^{M} h_m(x_i) \tag{2.14}$$

where h_m are the decision trees which are utilized as base estimators. The constant M is the number of base estimator to be used and is defined by the user. By default, the initial model F_0 is selected as the constant that minimizes the error function.

The above stated formulation means that similar to other boosting methods the gradient tree boosting is built in a greedy fashion [86]:

$$F_m(x) = F_{m-1}(x) + h_m(x) \tag{2.15}$$

Each newly added tree h_m is trained to minimize a sum of losses given the previous ensemble model F_{m-1}:

$$h_m = argminL_m = argmin\sum_{i=1}^{n} l(y_i, F_{m-1}(x_i) + h(x_i)) \tag{2.16}$$

where $l(y_i, F(x_i))$ is defined be the selection of the error function. Using a first order Taylor based approximation l can be estimated as:

$$l(y_i, F_{m-1}(x_i) + h(x_i)) \approx l(y_i, F_{m-1}(x_i)) + h(x_i)\frac{\partial l(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} \tag{2.17}$$

The quantity $\frac{\partial l(y_i,F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}$ is easy to calculate for any F_{m-1} since the error function is differentiable. Combining equations 2.16 and 2.17 and removing the constant terms we get:

$$h_m = argmin\sum_{i=1}^{n} h(x_i)g_i \tag{2.18}$$

Where $g_i = \frac{\partial l(y_i,F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}$. In order to minimize 2.18 the decision tree is fitted to predict a value that is proportional to the negative gradient $-g_i$. The gradients are updated at each iteration. The iterative process of gradient tree boosting is illustrated in Figure 2.4.

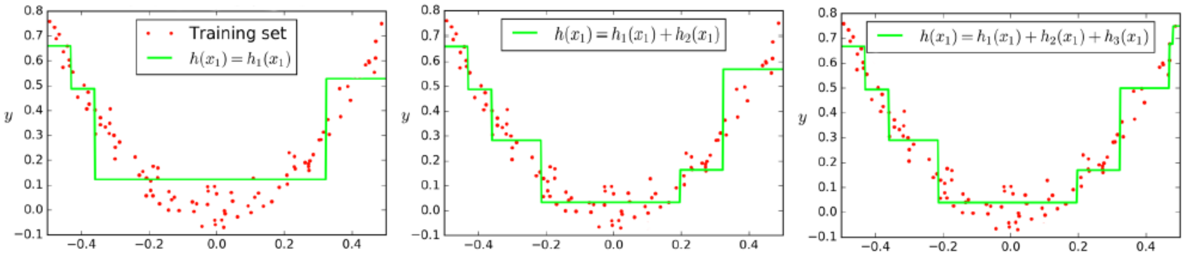


Figure 2.4: Gradient Boosting[3]

2.5. Random Forest

Decision tree algorithms tend to overfit complicated problems. In order to address this problem the random forest algorithm has been developed [87]. The random forest algorithm is an ensemble model of decision trees, in which each tree is trained on a random set of variables. Following the law of large number the algorithm always converges, which solves the overfitting problem [88]. The method of training base estimators on subspaces of the feature space is an efficient method of reducing bias in the model.
In original implementation of the random forest model each tree has a vote for the most popular class of the input X. However, in the implementation used in the scope of this project the classifiers are combined by averaging their probabilistic prediction, instead of letting each classifier vote for a single class [85].The mathematical formulation of the output of the ensemble model is [47]:

$$H(x) = argmax \sum_{i=1}^{k} I(h_i(X,\Theta) = Y) \tag{2.19}$$

in which θ are random feature vectors, h_i are decision trees, Y is the output variable of each tree and I is the characteristic function. The maximum generalization error of the random forest depends on the correlation of the trees and their strength [88]. The concept of the algorithm is illustrated in Figure 2.5.

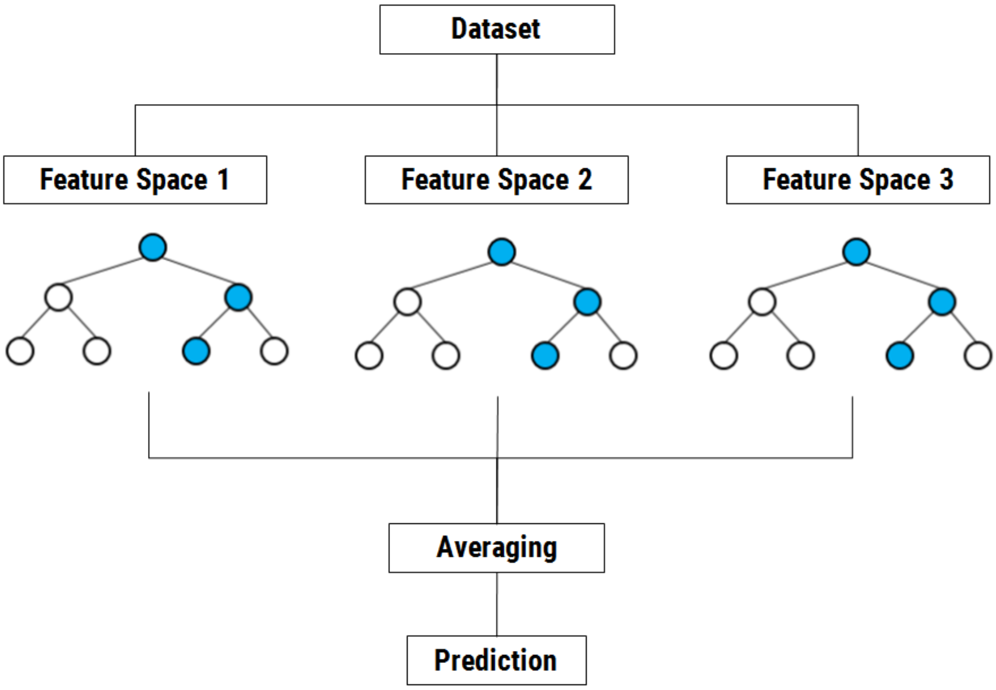


Figure 2.5: Random Forest

Chapter 3

Finite Element Method Reformulation

The present chapter deals with the formulation of the inverse Finite Element Method approach for in-situ assessment of the operational load. The stiffness matrices of the structures studied in this project have been collected as an output of Abaqus. The reformulation of the problem as well as its solution have been implemented in Python. The results have been verified using Abaqus.

3.1. Finite Element Method

The finite element method is by far the most widely used method in computational mechanics. It allows obtaining approximate solutions for boundary values problems in engineering. The procedure allows the continuum to be discretized in a number of elements with known behaviour and then it formulates the global problem by assembling all the elemental equations [89]. It involves three major steps which are no other than, pre-processing, solution and post-processing. During pre-processing the equilibrium equation is formulated [90]. The steps for the formulation of the equilibrium equation are the following ones:

- Discretize domain into elements.
- Assume approximate functions as behaviour of the element.
- Set-up equilibrium equation in element level.
- Assemble the local equations to form the global equilibrium equation.
- Apply boundary conditions and loads.

In the case of a static or quasi-static problem the resulting equilibrium equation is:

$$[K][u] = [F] \tag{3.1}$$

where K is the global stiffness matrix of the structure, u is the displacement vector and F is the load vector.

3.2. Load Magnitude Calibration FEM Reformulation

As it becomes evident in order to solve the classical formulation of the equilibrium equation of FEM, the properties of the structure, the boundary conditions and the external loading need to be known. However, in many cases the loading of the structure is not known and its measurement is too costly or complicated. Under certain circumstances, with a proper reformulation of the problem, the theoretical background behind FEM can be used to estimate the magnitude of the load. The prerequisite circumstances are the following two:

- The "shape" of the expected load is known.
- Strain measurements can be performed on the structure.

Saying the shape of the expected load needs to be known, it is meant that one needs to know the location and the direction of the applied load as well as its intensity compared to a maximum value. The reason for these prerequisites will be explained in the following sections.

3.2.1. Reformulation of the load vector

In the classical FEM approach the load vector is a vector containing a number of cells equal to the degrees of freedom of the discretized structure. The value of each cell of the load vector corresponds to the magnitude of the load applied in the corresponding direction of the cell in the global system. This means that the load vector has the following format:

$$[F] = \left[f_{1_x}, f_{1_y}, f_{1_z}, f_{1_{xr}}, f_{1_{yr}}, f_{1_{zr}} l ..., f_{i_x}, f_{i_y}, f_{i_z}, f_{i_{xr}}, f_{i_{yr}}, f_{i_{zr}}\right]' \tag{3.2}$$

where the value f_i is the magnitude of the load.
This can be re-written as:

$$[F] = a[b] \tag{3.3}$$

where a is the absolute maximum value of the load vector. This means that $[b]$ will be a vector whose values are in the range [-1,1]. The information that vector $[b]$ contains, are essentially the ones needed as prerequisites. Namely the nodes where load is applied, the direction of the load and its comparative magnitude. This vector $[b]$ will be referred to as the shape vector from now on. The absolute maximum value of the load vector a will be named calibration factor. The description of more complicated load cases can be achieved through the decomposition of the load vector to a linear combination of shape vectors and calibration factors. In this case the load vector can be written as:

$$[F] = \sum_{i=1}^{n} a_i [b]_i \tag{3.4}$$

In this case, each shape vector b_i and calibration factor a_i refer to a local part of the loading. This means that the comparative magnitude between the different parts of the loading is no longer of importance, since a different calibration factor is assigned to their shape vector.

3.2.2. Translation of strain measurements to nodal equations

The most common way of measuring strain is the use of strain gauges. If it is assumed that each end of a strain gauge is attached to a node then what the strain gauge does is measure the deformation of the line that is connecting the two nodes. By the definition of strain [91] the following equation holds:

$$\epsilon = \frac{\Delta l}{l} = \frac{u_2 - u_1}{l} \tag{3.5}$$

where ϵ is the deformation l is the distance between the nodes and u_i is the displacement of each node on the axis on which the calculated strain value is referred to. Equation 3.5 can also be written as:

$$[0, ..., 0, -1, ..., 1, 0, ..., 0][\mathbf{u}] = \epsilon * l \tag{3.6}$$

where u is the displacement vector.

3.2.3. Reformulation of the equilibrium equation

Inserting equation 3.3 to the FEM equilibrium equation, it now becomes:

$$[K][u] = \sum_{i=1}^{n} a_i [b]_i \tag{3.7}$$

Under the assumption that the shape vector is known and the calibration factor is unknown, it means that one more free variable is added to the system. The system is now under-determined, which means that an infinity of non-trivial solutions exists [92].

The equation 3.6 that describes the strain measurement can used as a constraint. The addition of this extra equation in the system will make the rank of the system equal to the number of the free variables, thus resulting in a unique solution. After inserting the equation and rearranging the system can be re-written as:

$$[A][x] = [y] \tag{3.8}$$

where $[A]$ is the stiffness matrix $[K]$ horizontally concatenated with the shape vector $[b]$ and vertically concatenated with equation 3.6, $[x]$ is the displacement vector $[u]$ vertically concatenated with calibration factor

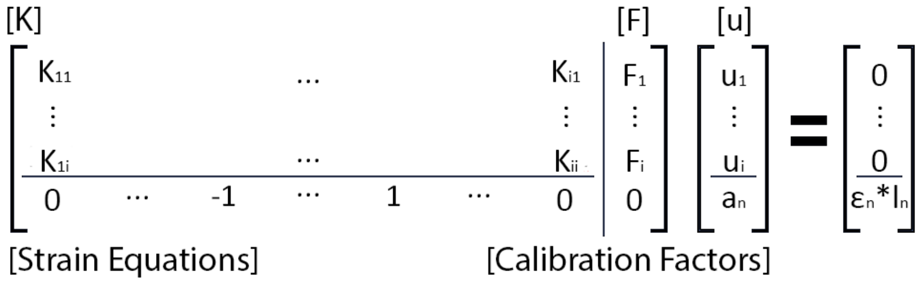


Figure 3.1: The shape of concatenated matrices

a and $[y]$ only contains zeros with the exception of the last cell where the value $\epsilon * l$ is assigned. The shape of the concatenated matrices is shown in Figure 3.1. As it becomes evident, the number of calibration factors that can be calculated is equal to the number of the strain measurements available. The solution of the system described by the equation 3.8 will result in the estimation of the nodal displacements and the calibration factors.

Chapter 4

Structural Reliability Analysis using Chaotic Particle Swarm Optimization

In the context of time invariant structural reliability analysis the design variables x on which there is a certain degree of uncertainty are described using a random vector $[X]$. Afterwards, the limit state function $G(X)$ is generated that defines the limit between the survival and the failure domain. The limit state function is defined to have the following properties:

- If $G(X) < 0$ the structure is in the failure domain
- If $G(X) = 0$ the structure is in a limit state
- If $G(X) > 0$ the structure is the survival domain

The value of the limit state function $Z = G(X)$ is called safety margin of the structure. According to the definition of the limit state function is the safety margin is less or equal to zero the structure fails. Taking this into account the failure domain is defined as $D = [X \epsilon R^n, G(X) \leq 0]$. The probability of failure of the structure can be calculated as:

$$P_f = \int_D f_X(x)dx \tag{4.1}$$

Where $f_X(x)$ is the joint probability density function of the design variables. Over the last decades several approaches have been developed to solve Equation 4.1. The main of these approaches are [4]:

- Direct integration over the failure domain. Such methods have very high accuracy. However, in order to implement them the joint probability density function needs to be known.
- Monte Carlo simulations. Such methods are accurate and easy to implement for all kinds of applications. However, in order to accurately estimate low values of probability of failure many samples are required to achieve convergence thus making them computationally expensive.
- First or Second order reliability methods using the response surface method or reliability index. The accuracy of such methods depends on the formulation of the limit state function. For low values of the probability of failure they are computationally inexpensive compared to their alternatives.
- Discrete approximation. Such methods are used in complex systems. Their accuracy depends on the quality of the approximation.

Taking the strengths and weaknesses of these approaches into consideration a FORM was selected to be used in the context of this study. This choice is motivated by the fact that low values of probability of failure are expected to be encountered during the lifetime of marine structures. The framework that will be developed in the scope of this research consists of the minimization of the Hansofer-Lind reliability index using an implementation of the Chaotic Particle Swarm Optimization algorithm.

4.1. Hansofer-Lind Reliability Index

The Hansofer-Lind reliability index [93] is defined in a Gaussian standard space (Λ) where the design variables are uncorrelated. The limit state function defines a n-th dimensional limit surface(Γ) in the design space (Ψ). This limit surface can be translated to the Gaussian space Λ to be $\Sigma = T(\Gamma)$. Then the Hansofer-Lind index (β_{HL}) is defined as the minimum distance between the origin of space Λ and a point that evolves on the limit

surface Σ. A visualization of the definition of Hansofer-Lind index is given in Figure 4.1.
The matrix formulation of the Hansofer-Lind index for correlated design variables is [94]:

$$\beta_{hl} = min\sqrt{([X]-[\mu])^T[C]^{-1}([X]-[\mu])} \tag{4.2}$$

where $[\mu]$ is the vector of the mean values of the design variables and C is the covariance matrix. This can also be re-written as [95]:

$$\beta_{hl} = min\sqrt{(\frac{x_i-\mu_i}{\sigma_i})^T[R]^{-1}(\frac{x_i-\mu_i}{\sigma_i})} \tag{4.3}$$

where R is the correlation matrix.
If the design variables in equation 4.3 have a Gaussian distribution the it can be written as:

$$\beta_{hl} = min\sqrt{[u]^T[R]^{-1}[u]} \tag{4.4}$$

in which $[u]$ is the vector of design variables in the standard Gaussian space. For uncorrelated variables this becomes [4]:

$$\beta_{hl} = min\sum_{i=1}^{n} u_i^2 \tag{4.5}$$

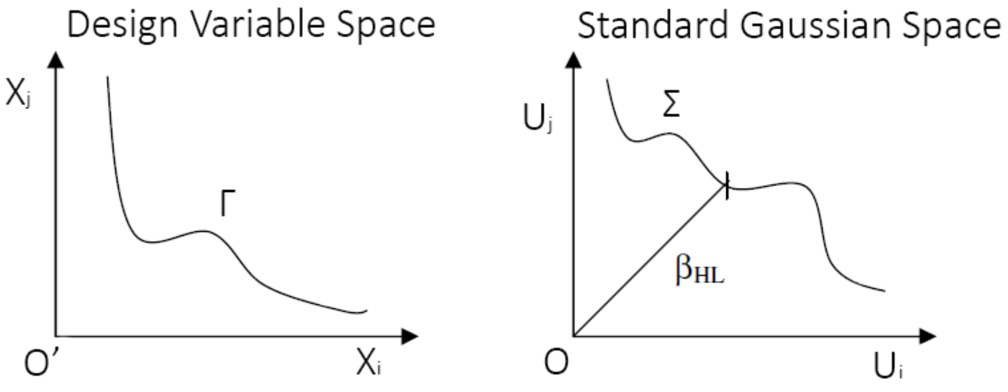


Figure 4.1: Hansofer Lindt Index Visualization [4]

4.2. Structural Reliability Analysis as an Optimization Problem

As it becomes evident from the definition of the Hansofer-Lind reliability index the calculation of the reliability index is essentially a constrained minimization problem. The value of the reliability index needs to be minimized under the constraint that the limit state function needs to be equal to zero. Taking into account the matrix formulation of the Hansofer-Lind reliability index the mathematical formulation of the minimization problem is:

$$min\sqrt{(\frac{x_i-\mu_i}{\sigma_i})^T[R]^{-1}(\frac{x_i-\mu_i}{\sigma_i})} \tag{4.6}$$

Under the constraint:

$$G(X) = 0 \tag{4.7}$$

Using the penalty method to apply constraints objective function of the problem can be reformulated as:

$$min[\sqrt{(\frac{x_i-\mu_i}{\sigma_i})^T[R]^{-1}(\frac{x_i-\mu_i}{\sigma_i})} + M * G(X)] \tag{4.8}$$

where M is the penalty coefficient. The value of the penalty coefficient should be properly selected to avoid ill-posing of the minimization problem. Solving the constrained minimization problem stated by Equation 4.8 the Hansofer-Lind reliability index can be estimated. Then the probability of failure can be calculated as:

$$P_f = 1 - \Phi(\beta_{HL}) \tag{4.9}$$

in which Φ is the cumulative distribution function of the standard Gaussian distribution.

4.3. Chaotic Particle Swarm Optimization

The Chaotic Particle Swarm Optimization algorithm is variation of the Particle Swarm Optimization genetic algorithm introduced by Kennedy and Eberhart [96]. The method is based on simulating the social behaviour of flocks of birds.

In the classic PSO implementation an initial population is generated randomly in the search space Ω. Each particle of the initial population is assigned an initial velocity equal to zero. The size of the population depends on the trade-off between performance and computation time. Larger populations result in a more consistent performance of the algorithm. However, they are related to higher computation times.

At each iteration of the algorithm all the particles are evaluated with respect to the value of the objective function that they result into. Afterwards, their velocities and positions are updated accordingly. The velocity of the particles is updated according to the following equation:

$$v_{ij}^{h+1} = w * v_{ij}^{h} + c_1 * r_1 * (Pbest_{ij}^{h} - x_{ij}^{h}) + c_2 * r_2 * (Gbest^{h} - x_{ij}^{h}) \tag{4.10}$$

in which i is the range of the population and j is the dimensions of the search space. $Pbest$ is the best performing location of each particle and $Gbest$ is the best performing location of all the particles. Moreover, w is a deterministic parameter called an inertia parameter, c_1 and c_2 are deterministic parameters and r_1 and r_1 are samples of a uniform random variable. The velocity updating procedure consists of two main contributors:

- The own experience of each particle taken into consideration through the term $(Pbest_{ij}^{h} - x_{ij}^{h})$
- The experience of the whole population taken into consideration through the term $(Gbest^{h} - x_{ij}^{h})$

After the update of the particles' velocities, their positions are updated according to the equation:

$$x_{ij}^{h+1} = x_{ij}^{h} + v_{ij}^{h+1} \tag{4.11}$$

This process is repeated until a termination criterion is fulfilled. An flowchart of the PSO algorithm is presented in Figure 4.2.

The selection of the parameters w, r_1 and r_2 is regarded to be crucial for the performance and robustness of the algorithm [71]. The inertia parameter controls the effect of the previous velocity on the updated velocity. A larger value of the inertia parameter makes the PSO algorithm look for the global optimum in a larger area. On the negative side, larger values of the inertia parameter reduce the precision of the algorithm. The selection of w is a trade-off between exploration and exploitation. What is more, fixed values of the r_1 and r_2 parameters cannot always ensure ergodicity within the search space.

To avoid being trapped in local optima the Chaotic Particle Swarm Optimization algorithm expands the conventional implemation of the PSO by utilising a chaotic system for the selection of the above mentioned parameters. Chaos as Zhao et al. definite it is "a characteristic of a non-linear system involving a bounded unstable dynamic behaviour that exhibits sensitive dependence on the initial conditions and includes infinite unstable periodic motions" [71]. The utilization of such a system to determine the parameters of a PSO algorithm enhances the search behaviour of the algorithm.

The chaotic system that has been selected is the logistic equation:

$$x_{n+1} = \mu * x_n * (1 - x_n) \tag{4.12}$$

$$0 \leq x_0 \leq 1 \tag{4.13}$$

Despite its deterministic behaviour the logistic equation displays chaotic dynamics when $\mu \neq 4$ and $x_0 \neq 0, 0.25, 0.5, 0.75, 1$. The selection of the logistic equation as the chaotic system for this implementation is motivated by its computational simplicity and the fact the output space of the equation matches the definition space of the user-defined parameters of the PSO algorithm.

Taking that into account the equations that control the behaviour of the PSO parameters in the CPSO implementation are:

$$w_{n+1} = \mu * w_n * (1 - w_n) \tag{4.14}$$

$$0 \leq w_0 \leq 1 \tag{4.15}$$

and

$$r_{i(n+1)} = \mu * r_{i(n)} * (1 - r_{i(n)}) \tag{4.16}$$

$$0 \leq r_{i(0)} \leq 1 \tag{4.17}$$

where $i = 1,2$ and $w(0), r_{i(0)} \neq 0, 0.25, 0.5, 0.75, 1$.

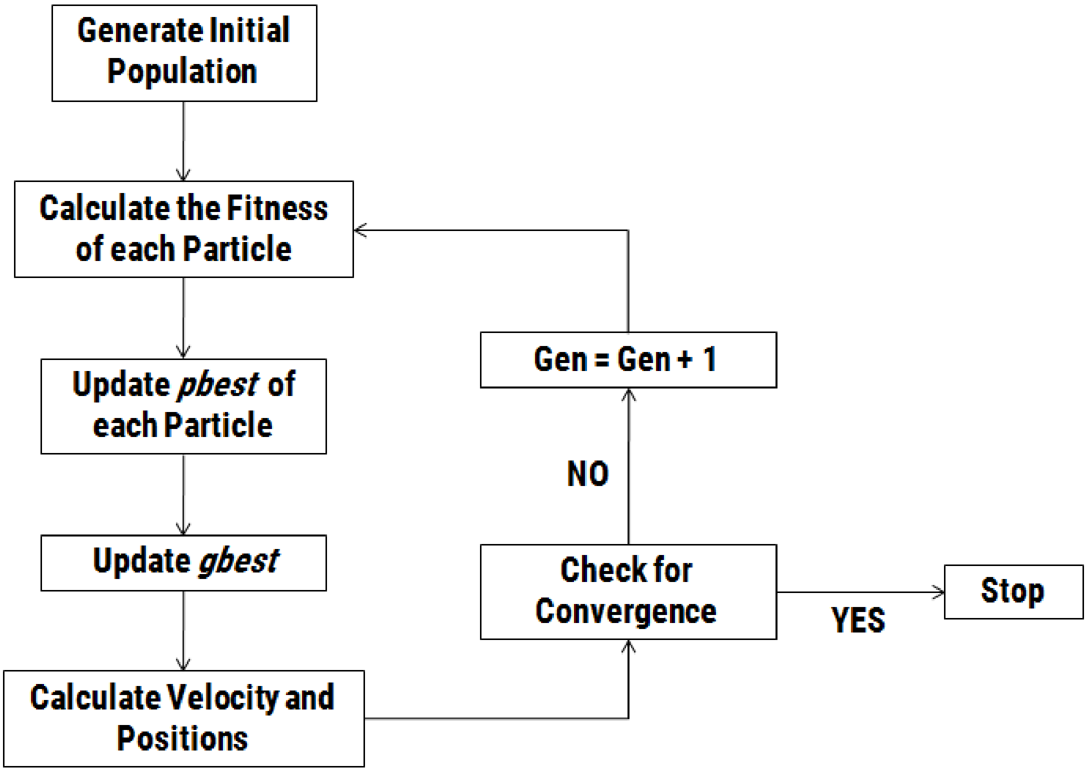


Figure 4.2: Particle Swarm Optimization algorithm flowchart

Chapter 5

Damage and Response Prediction

A considerable amount of sea state and response data has been acquired within the Monitas Group. In the scope of this project a part of this data, regarding the West Africa unit, has been utilized in order to create a digital twin of certain structures of the unit as a case study. Specifically, two machine learning models have been developed and trained using the available data. The first of these models is used to predict the fatigue damage accumulated during a half-hour sea state receiving as input the wave energy spectrum of the sea state. The second one aims to predict the frequency at which the peak of the stress power density spectrum is encountered. The predicted damage and frequency are then used in order to scale and shift the design RAOs which can afterwards be used to calculate the response of the structure.
It is strongly advised to read Appendix 1 before proceeding with this Chapter.

5.1. Data Filtering

The above mentioned data was acquired using a weather buoy as well as an AHMS similar to the one installed on-board the Bonga FPSO [1]. The data was stored in half-hour samples. The weather buoy was acquiring the wave energy spectrum of the sea state, while the AHMS was recording time-series of strain at multiple locations. The data from the weather buoy and the AHMS was acquired and saved independently. As a result, data was not always available for both the sea state and the structure response, because either the buoy or the sensors were occasionally out of service. Furthermore, sometimes the collected data was corrupted due to malfunction of the data acquisition system. The above stated facts make it clear that the data had to be filtered and then matched in order to end up with a usable dataset.
After careful examination of the data samples in which one of the strain gauges was possibly malfunctioning, it was derived that the recorded strain was fluctuating around a mean with an infinitesimal amplitude as it can be seen in Figure 5.2. The amplitude of the time-series when the strain gauge is functioning normally is illustrated in Figure 5.1 for comparison. This infinitesimal amplitude served as an indication that when this time-series of strain was translated to fatigue damage, utilizing the rainflow counting algorithm and the S-N curve of the structure, it would result in unrealistically small values of fatigue damage. Taking advantage of this fact, the corrupted sensor data was removed by calculating the corresponding fatigue damage of each data sample and applying a high-pass filter. More specifically, data samples corresponding to a fatigue damage value lower than 10^{-10} were flagged as corrupted and removed. This value was selected after inspection of the available data.

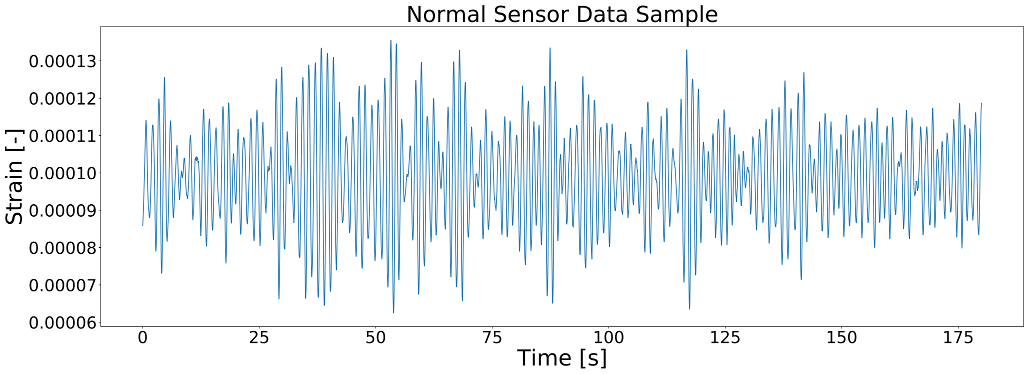


Figure 5.1: Strain time-series when strain gauge is functioning properly

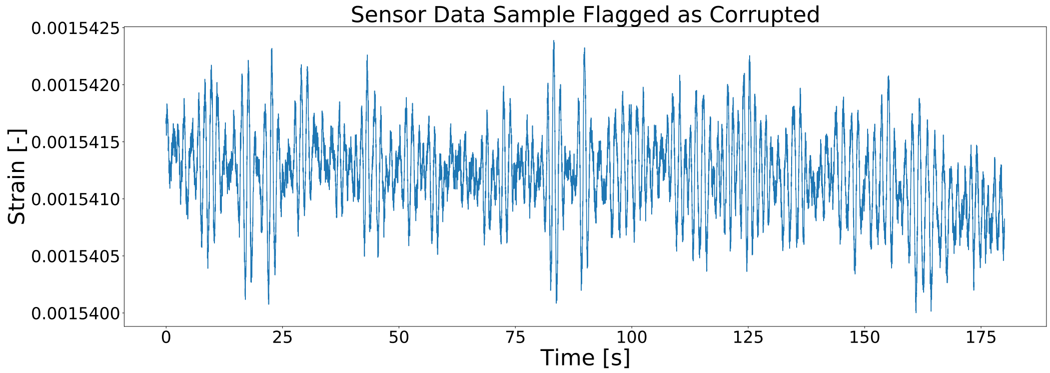


Figure 5.2: Strain time-series that has been flagged as corrupted

After the corrupted sensor data was removed, the remaining sensor and buoy data were matched, leaving only the samples when both response and sea-state information were available in the dataset.

Due to the lack of a straightforward way to filter the buoy data, the filtering was achieved by comparison to the sensor data. More specifically, the accumulated fatigue damage of each sample was calculated using spectral fatigue analysis and cycle counting methods for the buoy and the sensor data correspondingly. If the buoy data was acquired without being corrupted the calculated fatigue damage was expected to be comparable to the fatigue damage that was calculated using the corresponding sensor data. Some deviation was expected due to the inherent inaccuracy of the design RAOs. Based on this assumption, all the buoy data samples, of which the calculated fatigue damage was less than 10% of the fatigue damage calculated using the sensor data, were flagged as corrupted. Different threshold values were tested with only minor effect on the filtering process. Only an one sided margin was used. This decision was driven by the fact that when the weather buoy was on-board the vessel for maintenance it could only measure the part of the wave energy that was transferred to the vessel. This assumption was verified by the data, since there were no fatigue damage estimations in a higher order of magnitude than the measured fatigue damage.This filtering process resulted to the final form of the dataset, in which both the corrupted buoy and sensor data have been removed.

5.2. Damage Prediction

With the final form of the dataset available, the machine learning model responsible for the fatigue damage prediction could be developed. The model selected for this application is an ANN, since the fatigue damage accumulation is a highly non-linear problem. The first step in the process of developing an artificial neural network is the selection and extraction of the features. It is followed selection of the network's architecture, the processing of the feature and label data and finally the training and evaluation of the model.

5.2.1. Feature Selection for Damage Prediction

In engineering applications it is fairly common for quantities involved in the analytical calculations to be selected as features. This happens because their involvement in the analytical calculations serves as a strong indicator that they are closely related to the quantity that needs to be predicted.

For the application at hand, the features were extracted from the wave energy spectrum data. The features were carefully selected in order to capture the information contained in the wave energy spectrum. The information that were deemed as crucial were the location of the peak energy as well as the magnitude of the energy content and its spread around the peak. In order to capture the above mentioned information, the following features were selected in order to be examined:

- Peak energy frequency
- Peak energy direction
- Peak energy magnitude
- Direction range
- Frequency range

- Zero-up crossing period
- Mean wave period
- Significant wave height

The peak energy frequency, peak energy direction and peak energy magnitude represent the coordinates of the maximum energy point in the (frequency, direction,energy) space. The direction range is defined as the range of wave headings in which the energy content is at least 10% of the maximum energy:

$$R_\theta = \theta_{max_{10\%}} - \theta_{min_{10\%}} \tag{5.1}$$

The frequency range is defined likewise as the range of wave frequencies in which the energy content is at least 10% of the maximum energy:

$$R_\omega = \omega_{max_{10\%}} - \omega_{min_{10\%}} \tag{5.2}$$

The zero-up crossing frequency is calculated using the following formula where m_0 is they zero order spectral moment and m_2 is the second order spectral moment:

$$T_0 = 2\pi\sqrt{\frac{m_0}{m_2}} \tag{5.3}$$

The spectral moments can be calculated using the following equation:

$$m_i = \int \omega^i S_\sigma(\omega) d\omega \tag{5.4}$$

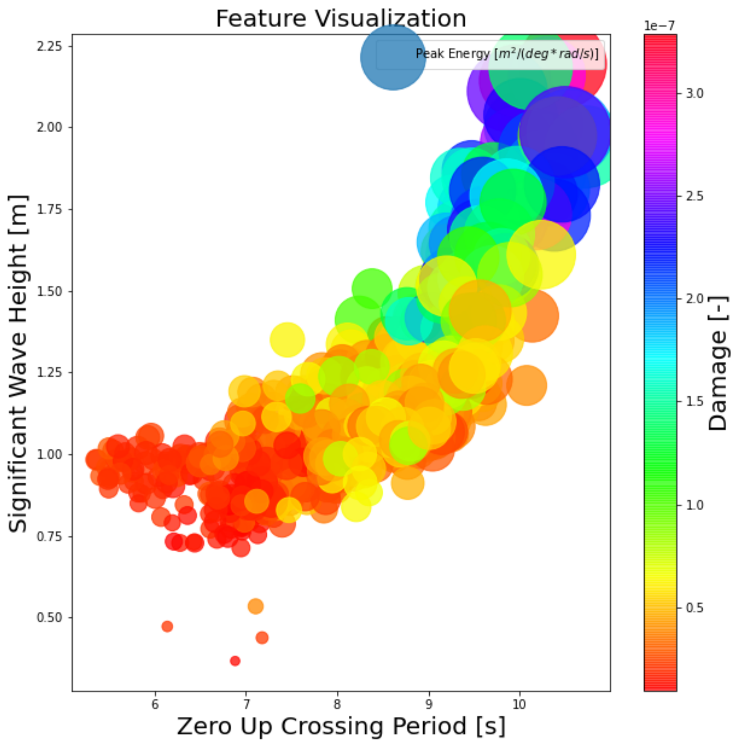
Where i is the order of the moment, S_σ is the stress spectrum and ω is the frequency. The equation used to calculate the mean wave period is:

$$T_m = 2\pi\frac{m_0}{m_1} \tag{5.5}$$

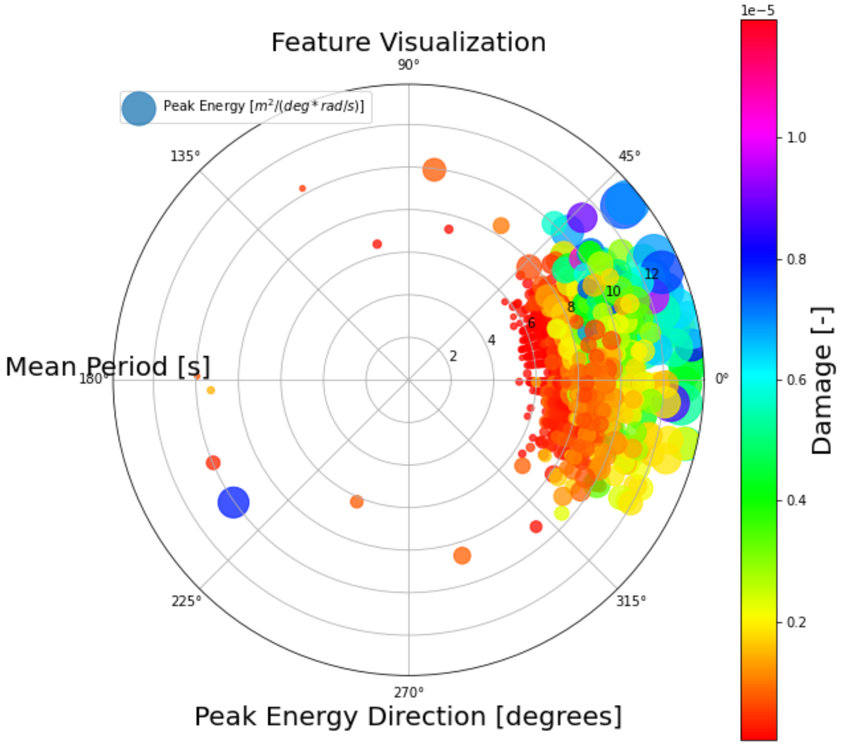
where m_1 is the first order spectral moment.
The significant wave height has been calculated using the zero order spectral moment:

$$H_s = 4\sqrt{m_0} \tag{5.6}$$

Subsequently to their extraction, the features where visualized. The purpose of the features' visualization is the optical assessment of data clusters and underlying trends in order to verify their relevance to the quantity that needs to be predicted.
In figure 5.3a the underlying relationship between significant wave height (H_s), zero-up crossing period (T_0), peak energy and fatigue damage was examined. The zero-up crossing period was assigned to x-axis, the significant wave height to y-axis, the peak energy level was indicated by the area of each scatter point and the colour was related to the fatigue damage. The data points illustrated correspond to one month of filtered data. As it becomes evident there is a clear trend in the illustration. Sea-states resulting to low fatigue damage accumulation are related to low significant wave height and zero up crossing period values and low peak energy content. The more H_s, T_0 and maximum energy increase the more the resulting fatigue damage accumulation increases. This observation is in line with wave and fatigue analysis theory since significant wave height serves as an indicator of the wave's energy content and waves with a higher energy content usually result in higher deformations and thus higher fatigue damage.
As it becomes clear in Figure 5.3b, fatigue damage accumulation also increases as mean period increases. This is also in line with what was expected from theory. A higher mean period means a higher wavelength and larger waves result in higher fatigue damage accumulation in structures stressed in a global loading state. For structures stressed in a local manner, such as a side shell structure, lower periods (higher frequency) result in more stress cycles which might result in more overall damage accumulation. This phenomenon has been observed for side shell sensors in the dataset under study during the winter months when the weather is calmer. One can also conclude that the direction of peak energy in most sea-states is spread in a 90°range around the heading of the vessel. Furthermore, it can be noticed that sea-states with a peak energy direction from 0° to 45° tend to have higher energy content and result in more damage than sea-states spread from 0° to $-45°$.



(a) (H_s, T_0, Peak Energy) Feature Visualization, July 2017, Side Shell Sensor



(b) (θ_{peak}, T_m, Peak Energy) Feature Visualization, September 2018, Deck Sensor

Figure 5.4 serves the purpose of depicting the relationship between direction range, frequency range peak energy and fatigue damage accumulation. It was concluded that the more spread wave energy spectrums both in terms of frequency and direction are related to lower wave energy peaks. An underlying relationship between the peak energy and the fatigue damage seems to be existent. However, this relationship is not entirely straightforward by means of visual inspection.

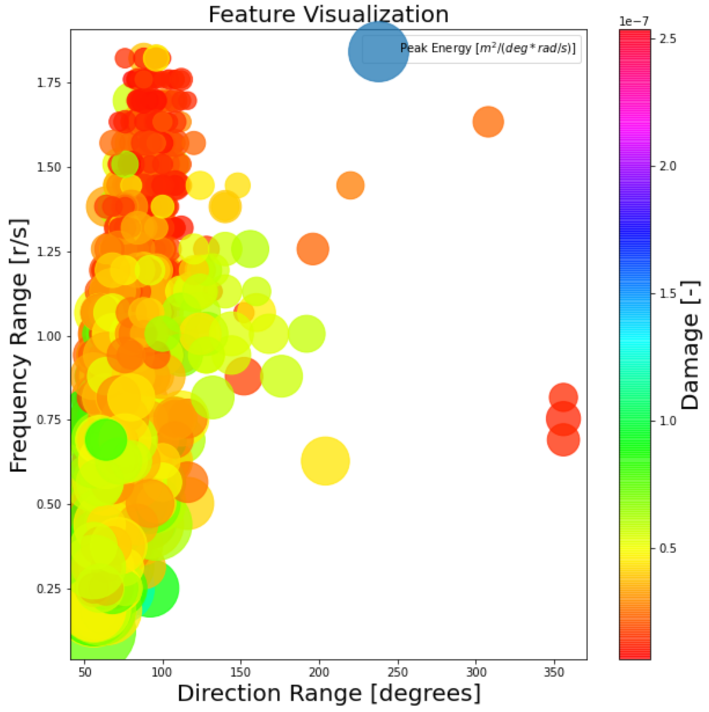
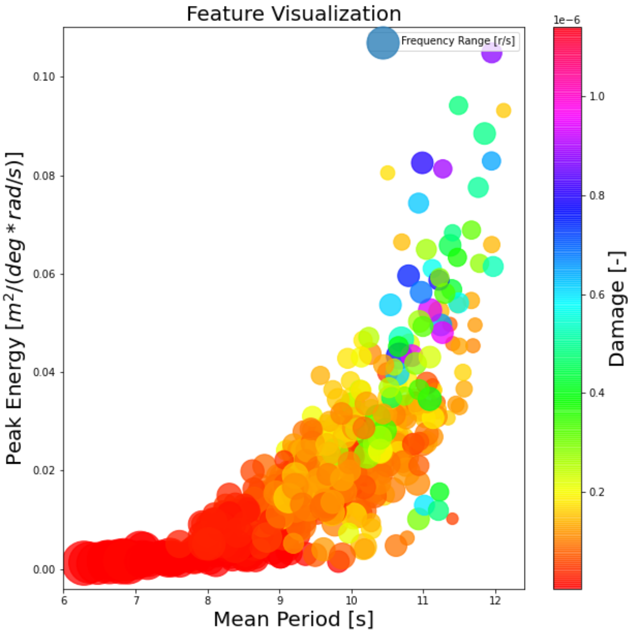


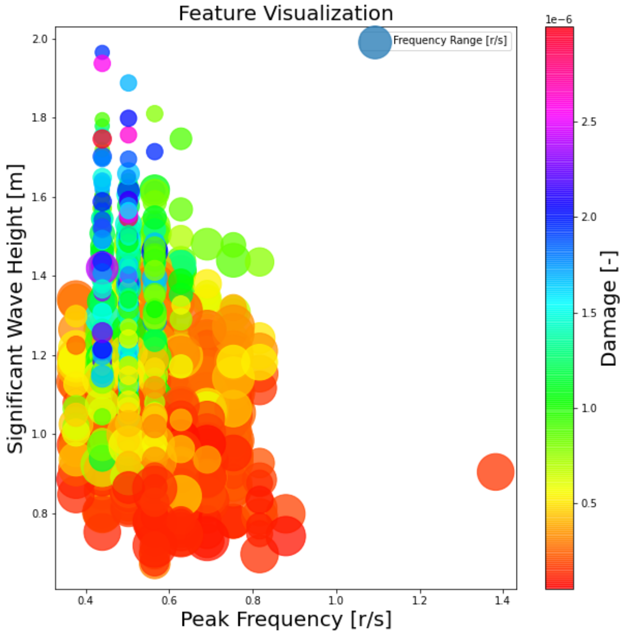
Figure 5.4: (R_θ , R_ω, Peak Energy) Feature Visualization, November 2018, Side Shell Sensor

When the connection between peak energy content, mean wave period, frequency range and damage was put under consideration, interesting conclusions were drawn. First of all, it is clear, through Figure 5.5a, that sea-states related to low fatigue accumulations formed a cluster at low mean wave periods and low peak energy levels. As the mean wave period and peak energy values increased, the resulting fatigue damage increased as well. However, for mean wave period values larger than $10s$ this trend seems to become more complicated. This is stated because high and mid damage sea states seem to become rather mixed for high peak energy contents and mean wave periods. It seems that the fatigue damage accumulation reaches a peak for sea-states with a mean wave period about $10.5s$ and then drops again. It is possible that this happens due a lower energy transfer from the waves to the vessel in these higher wave periods. The complexity of such an underlying connection between the features and the label serves as a strong indication that a properly selected and tuned machine learning regression algorithm can outperform a heuristic regression model.

Finally, the connection between significant wave height, peak frequency and frequency range was put under consideration. As it is indicated in Figure 5.5b the relationship between these values is much clearer. High damage sea-states, seem to be connected with high significant wave heights and low frequency spreading. It was also observed that such states have low peak frequencies.



(a) (T_m, Peak Energy, Rω) Feature Visualization, August 2018, Side Shell Sensor



(b) (Peak Frequency, H_s, Rω) Feature Visualization, October 2018, Deck Sensor

Besides visual inspection, which is a qualitative method of feature assessment, quantitative methods can be utilized as well. One of them is the calculation of the population Pearson correlation coefficient between the features and the label, which in this cases is the damage accumulation. Population Pearson correlation coefficient is indicative of the linear correlation of the features and the fatigue damage accumulation. However, in the case of fatigue damage accumulation strongly non-linear relations are observed as well. Under

this perspective the correlation coefficient can serve as a mere guideline and not as an absolute measure of feature evaluation.
The population Pearson correlation coefficient was calculated using the following formula [97]:

$$\rho_{X_i Y} = \frac{cov(X_i, Y)}{\sigma_X \sigma_Y} \tag{5.7}$$

Where features are denoted with X_i and the label values with Y. σ_X and σ_Y are correspondingly the standard deviation of the feature and the label. The function cov refers to the covariance of the feature and the label. It was calculated as it follows [98]:

$$cov(X, Y) = E[(X - \mu_X)(Y - \mu_Y)] \tag{5.8}$$

In equation 5.8 E stands for the expectation of the value and μ_X and μ_Y for the mean of the feature and the label. The results obtained from the calculation of the population Pearson correlation coefficient are presented in Table 5.1. LBSG3 and LBSG4 are long base strain gauges placed on the deck of the vessel while LSG10, LSG11 and LSG11 are strain gauges located on a side shell structure.

Table 5.1: Pearson correlation coefficient of features and labels

	LBSG3 Damage	LBSG4 Damage	LSG10 Damage	LSG11 Damage	LSG12 Damage
Peak Energy	0.744	0.748	0.751	0.708	0.740
Peak Frequency	-0.264	-0.278	-0.286	-0.255	-0.274
Peak Direction	-0.006	0.014	0.043	0.017	0.033
Frequency Range	-0.403	-0.409	-0.422	-0.367	-0.405
Direction Range	-0.145	-0.148	-0.153	-0.116	-0.140
Significant Wave Height	0.781	0.770	0.790	0.768	0.791
Zero Up Crossing Period	0.595	0.603	0.605	0.535	0.584
Mean Wave Period	0.583	0.595	0.599	0.527	0.575

As it is becomes obvious, the Peak Energy and the Significant Wave Height are the strongest features, in terms of linear correlation, since they have a Pearson correlation coefficient value larger than 0.7 with all the labels, which implies a very strong correlation. On the other hand, Peak Direction and Direction Range have Pearson correlation coefficients close to zero, which indicates that they are not linearly related to the fatigue damage accumulation. Taking into account the weak correlation of the peak energy direction and the direction range to the fatigue damage and the fact that no clear connection is indicated in the feature visualizations, peak energy direction and direction range were removed from the feature vector. This means that the final form of the feature vector, after the feature assessment, is:

$$[PeakEnergy, \omega_{peak}, R_\omega, H_s, T_0, T_s] \tag{5.9}$$

After the final selection of the features they were extracted from the available data thus building the final form of the dataset. The full dataset was then split to a training set, which will be used for the training of the network, and a test set on which the performance of the final model will be evaluated. It is important that the ANN has never seen the test set contents in order for the evaluation results to be accurate. The train set consists of 80% of the available data and the test set of the remaining 20%. The train-test split ratio was selected according to common practice.

5.2.2. Network Architecture

It is a widespread opinion that building neural networks is more of an art than it is a science [76]. Deciding the final architecture of a network is an intuitive process and it is based on the developer's experience and instinct.

Reviewing literature regarding similar applications is crucial in order to define an initial design point for the network's architecture. After examining several application in the context of the literature study that has been performed in the scope of this project, it has been decided that a multi-layer network would be optimal to result in accurate fatigue damage accumulation predictions. Due to the large quantity of available data a larger number of neurons could be properly trained with the back-propagation algorithm.

In order to be able to monitor the performance of the network during the architecture design phase 20% of the training dataset was set aside as a validation set. The performance of the network can now be evaluated on the validation set, thus leaving the test set uncompromised. The initial design approach was a neural network with two dense fully connected layers each one of them consisting of 100 neurons. Afterwards, the number of neurons in the layers was iteratively increased until it reached its final value of 512 neurons in each layer. After that point no significant increase in the networks performance was observed. However, it was noticed that the network was performing significantly better in the training set than in the validation set. This is a strong indication that the network is overfitting the data.

In order to address this problem a dropout layer was introduced [99]. Dropout layers are a common tool to prevent overfitting in deep learning applications. Their function is to set the output weights of the weakest neurons to zero, thus killing the connection of these neurons with subsequent layers. The percentage of the neurons to be killed is predefined by the user. In the application under study a 20% dropout layer was selected. The dropout ratio was decided through an iterative trial and error procedure. After the introduction, of the dropout layer the performance of the neural network on the train set and the validation set converged. The final architecture of the network is presented in Figure 5.6.

The activation function used in all the layers, except the output layer, is the ReLU. The ReLU activation function has a definition space from minus infinity to infinity and is partially linear from minus infinity to zero and from zero to infinity. This partial linearity means that the calculation of the function gradient is computationally inexpensive and that using ReLU gives the ability to the network to approximate non-linear problems. The output layer has a linear activation function in order to be able to return non-bounded continuous values.

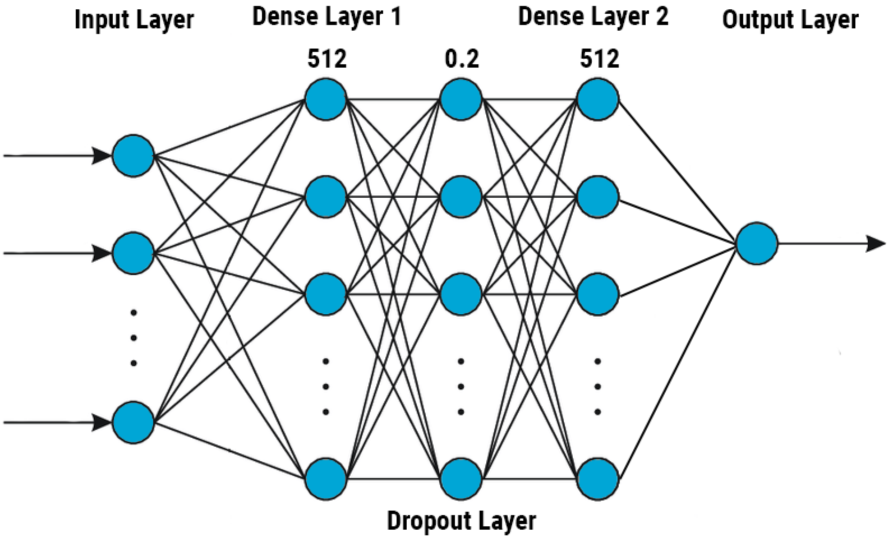


Figure 5.6: Final Neural Network Architecture

5.2.3. Data Processing Pipeline

Before training a neural network on a dataset the features and labels need to be processed beforehand. This happens because they have different ranges and magnitudes which disturbs the function of the neural network. In order for the neural network to function efficiently the data should be regularized to have a range from [-1,1]. In the application under examination a different processing pipeline has been selected for the features and the labels. The features are initially regularized using the MinMax scaling algorithm [100]. The mathematical formulation of this regularization algorithm is the one below:

$$X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.10}$$

$$X_{scaled} = X_{std} * (F_{max} - F_{min}) + F_{min} \tag{5.11}$$

In which X is the space of the feature that is being scaled and F is the whole feature space. The application of this transformation results in a transformed feature space with a range [0,1]. This regularization algorithm

was selected on the grounds that it results in the best performance of the neural network.
After the MinMax scaling, a Principal Component Analysis (PCA) [101] was performed on the transformed features. The purpose of the PCA was neither the dimensionality reduction nor the improvement of the performance of the ANN. It was applied in order to transform the features to the principal components of the feature space which removes the linear correlation between them. Having uncorrelated data is needed for sampling will be performed in Chapter 7. PCA is an orthogonal linear transformation that transforms the data to a new coordinate system in which variance is successively increased between principal components. Mathematically PCA is defined as a set of weights w_k that maps each row x_i of an X space (in this case the feature space) to a new space of principal component scores t_i:

$$t_i = x_i * w_k \tag{5.12}$$

In order for the variance to be maximized the first weight is calculated as shown below:

$$[w_1] = argmax\sum(t_1)_i^2 = argmax\sum(x_i w)^2 \tag{5.13}$$

Writing this in matrix form gives:

$$[w_1] = argmax||[X][w]||^2 = argmax([w]^T[X]^T[X][w]) \tag{5.14}$$

The quantity to be maximised can be recognised as a Rayleigh quotient [102]. A standard result for a positive semi-definite matrix such as $[X]^T[X]$ is that the quotient's maximum value is the largest eigenvalue of the matrix, which is true when w is the corresponding eigenvector.
The kth principal component can be calculated as:

$$[\hat{X}_k] = [X] - \sum_{s=1}^{k-1}[X][w_s][w_s]^T \tag{5.15}$$

Then the weight vector that results in the maximum variance can be calculated as:

$$[w_k] = argmax||[\hat{X}_k][w]||^2 = argmax\frac{[w]^T[\hat{X}_k]^T[\hat{X}_k][w]}{[w]^T[w]} \tag{5.16}$$

This returns the remaining eigenvectors of $[X]^T[X]$, with the maximum values for the quantity in brackets given by their corresponding eigenvalues. Thus the weight vectors are eigenvectors of $[X]^T[X]$. The application of the PCA after the MinMax scaling results in features in the range of [-1,1], due to the change of the coordinate system performed by the PCA.
As it has already been mentioned, the data processing pipeline is different for the labels(damage accumulation). In this case the values are scaled using the maximum absolute value of the labels:

$$X_{scaled} = \frac{X}{|X_{max}|} \tag{5.17}$$

This scaling algorithm normally results is a range [-1,1]. However, in the case of damage accumulation, because there are no negative damage values, the resulting range is [0,1].

5.2.4. Model Training and Evaluation

With the features and labels of the application extracted and processed the training of the neural network is now feasible. In order to achieve optimum performance of the models the training procedure was optimized.The optimization was performed using the data of LBSG3 and the resulting hyperparameters were applied to all the models. This was done under the assumption that the datasets are highly similar and thus the models would display similar behaviour. The most influential aspects of a model's training procedure are:

- The Error Function
- The Optimizer
- The Learning Rate
- The Batch Size

The error function is of utmost importance, since it dictates the way in which the training model penalizes deviations between the prediction and the label during the training phase. The available error functions, in Keras deep API where the model has been developed, for regression problems are[103]:

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- Mean Absolute Percentage Error (MAPE)
- Mean Squared Logarithmic Error (MSLE)
- Cosine Similarity
- Huber
- LogCosh

After studying relevant literature and the documentation of each error function it has been decided that the Mean Squared Error function should be used in the scope of this research. The motivation behind this decision is that MSE severely penalizes large errors, while it is more tolerant with low error values. This leads to a consistent model that avoids large errors in its prediction.

Having selected the error function the rest of the aspects could now be optimized. They were optimized in a sequential manner. It is possible that such an approach will not lead to a global optimum. However, it is preferred since it is much cheaper from a computational perspective and less time consuming than a global optimization approach that takes all the aspects into consideration simultaneously. Furthermore in order to reduce the required time for the training of the network an early stopping approach was selected. This means that if the performance of the model is not significantly improving after a pre-defined number of epochs(training iterations) the training procedure is terminated. Early stopping was coupled with a learning rate reduction strategy to break possible plateaus in the training procedure.

The first aspect to be optimized is the Optimizer. The ptimizer is nothing more than the optimization algorithm responsible for the weight estimation during the training of the network. Three different algorithms where taken into consideration:

- Adam
- Adamax
- Nadam

Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The other two optimizers are variations of the Adam algorithm [104]. Adamax is a variant of Adam based on the infinity norm. Nadam is essentially Adam with the addition of Nesterov momentum [105].

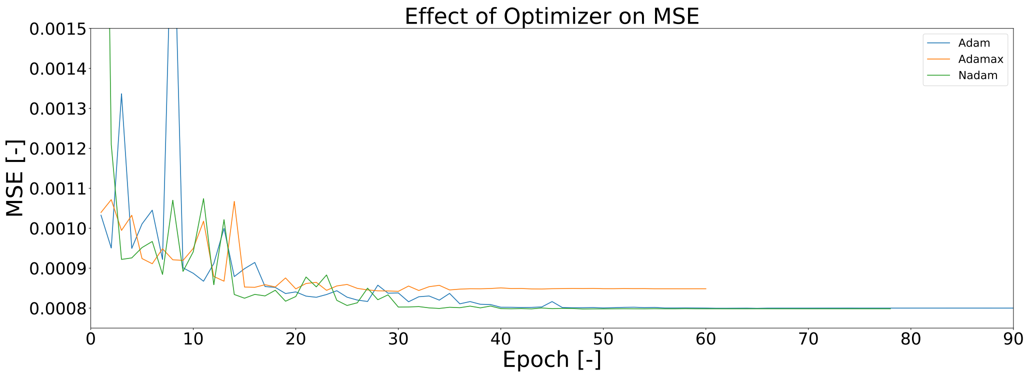


Figure 5.7: The effect of optimizer on the performance of the neural network

The selection of the optimizers to be tested was done after examining the available documentation [106]. The neural network has been trained using all three optimizers and its performance was evaluated on the validation set. The results are depicted in Figure 5.7. As it can be observed the performance of the model is almost identical for the Adam and Nadam optimizers. However, the implementation of Adam is considerably faster, thus leading to lower training time despite the larger number of epochs needed to achieve convergence. On these grounds Adam has been selected as the optimizer of the artificial neural network for this application.
After selecting the optimizer the learning rate of the algorithm was optimized. The learning rate is a hyperparameter that controls how much the model should change with response to the estimated error in each epoch. The best way to determine the optimal learning rate is through a sensitivity analysis. The examined range of learning rates for the neural network under study was [0.001,0.00005]. The of the range selection was performed taking into account that the default learning rate of the Adam optimizer is 0.001 [107].

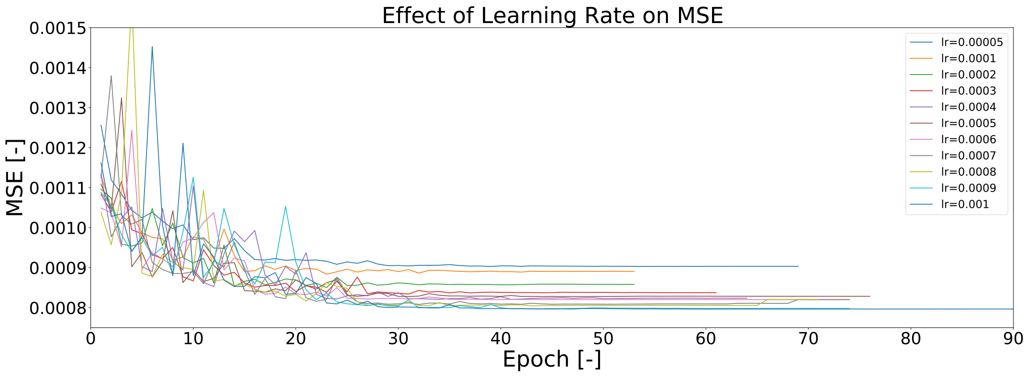


Figure 5.8: The effect of learning rate on the performance of the neural network

As it can be witnessed in Figure 5.8 the neural network performs better for higher learning rates. The MSE converges to its minimum value for a learning range equal to 0.0009.
The last hyperparameter of the neural network to be optimized is the Batch Size. Batch size is a term that refers to the number of training samples used in one training iteration. The batch size is usually selected to be a power of two. This is done in order to optimize the performance of GPU where the training of the neural network takes place. In the scope of the neural network under study the values of 32,64,128 and 256 where tested as possible batch size options. As it is illustrated in Figure 5.9 the optimal performance of the model is achieved for a batch size of 64.

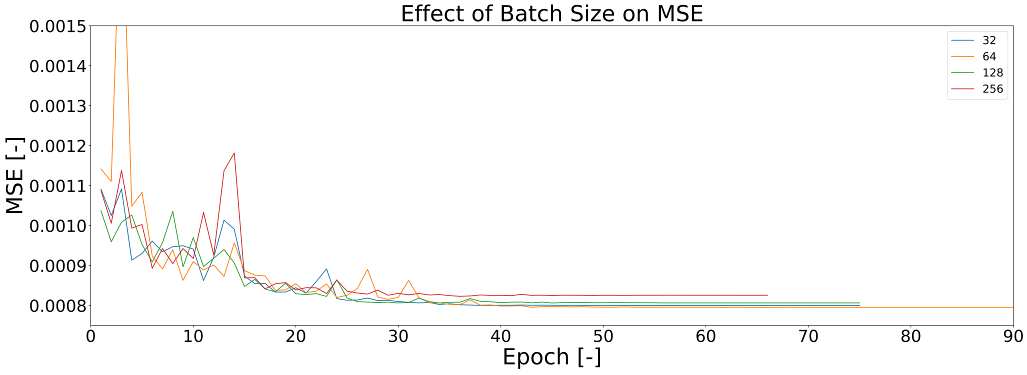


Figure 5.9: The effect of batch size on the performance of the neural network

The final configuration of the hyperparameters of the neural network is:

- Error Function: Mean Squared Error
- Optimizer : Adam
- Learning Rate : 0.0009
- Batch Size : 64

In order to determine the performance of the final configuration the neural network the model was iteratively trained and evaluated on random train-test set splits for a total of 10 iterations on each sensor location. The accuracy of the model was evaluated in terms of how accurately the fatigue damage accumulation is predicted on a moving weekly basis. The accuracy is defined as:

$$Accuracy = \frac{Damage_{predicted}}{Damage_{true}} * 100 \tag{5.18}$$

The selection of a moving weekly basis was based on the fact that using a moving week about 2000 accuracy results are generated for each iteration. Otherwise only about 300 accuracy results are generated. A large set of accuracy results is required in order to obtain a converged mean value and standard deviation. The moving weekly basis was selected instead of more iterations was preferred because of the high amount of time required for each iteration(about 15 minutes for 10 iterations) and the process would have to be repeated for a total of 5 sensor locations. The accuracy results were fitted with normal distributions with the following parameters:

- LBSG3 Mean: 100.3% Standard Deviation: 3.12%
- LBSG4 Mean: 100.7% Standard Deviation: 3.92%
- LSG10 Mean: 99.3% Standard Deviation: 3.5%
- LSG11 Mean: 99.9% Standard Deviation: 4.2%
- LSG12 Mean: 100.9% Standard Deviation: 3.4%

The fitting of the model accuracy results for each sensor location is presented in Figures 5.10, 5.11, 5.12, 5.13 and 5.14. These results serve as proof that the proposed artificial neural networks can accurately and consistently predict the fatigue damage accumulation for all the sensor locations on a weekly basis.

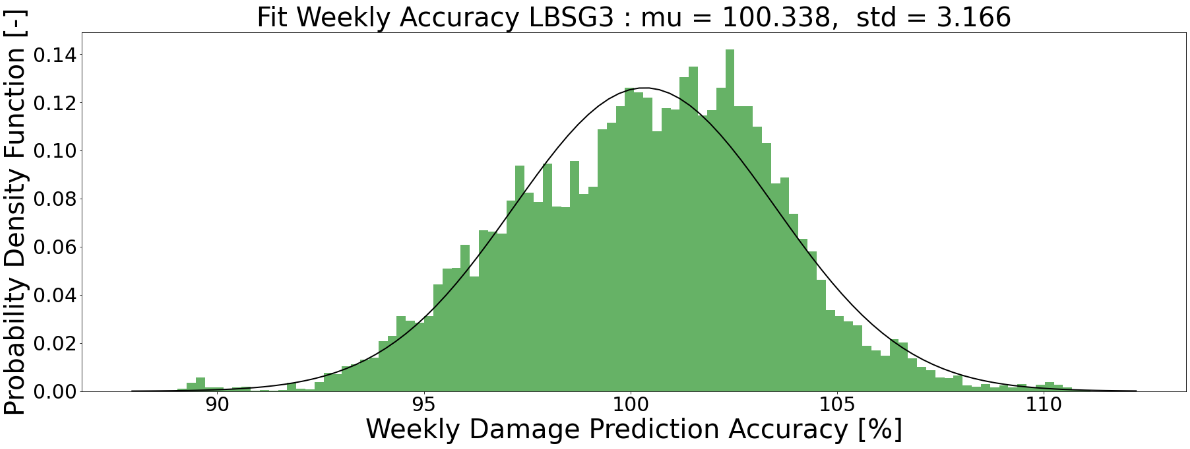


Figure 5.10: Model accuracy of LBSG3 fitted with a normal distribution

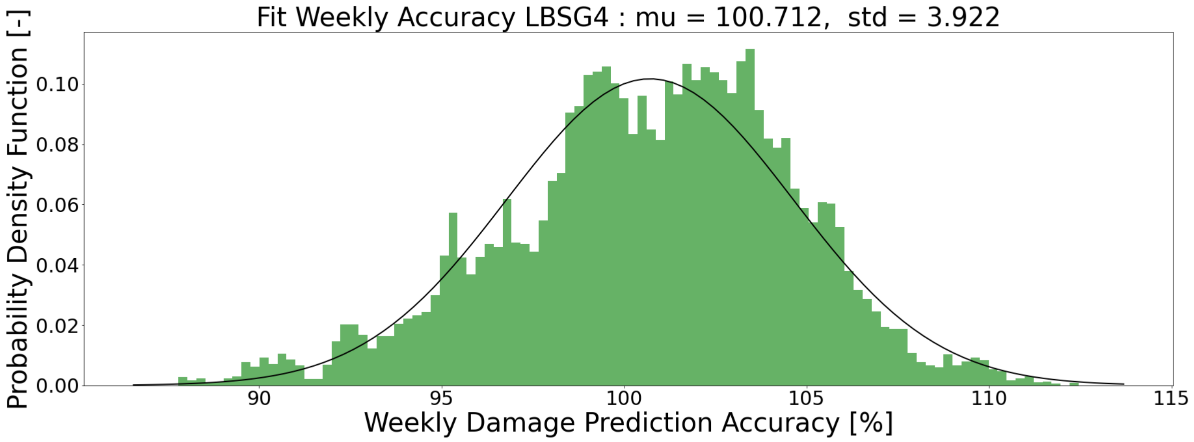


Figure 5.11: Model accuracy of LBSG4 fitted with a normal distribution

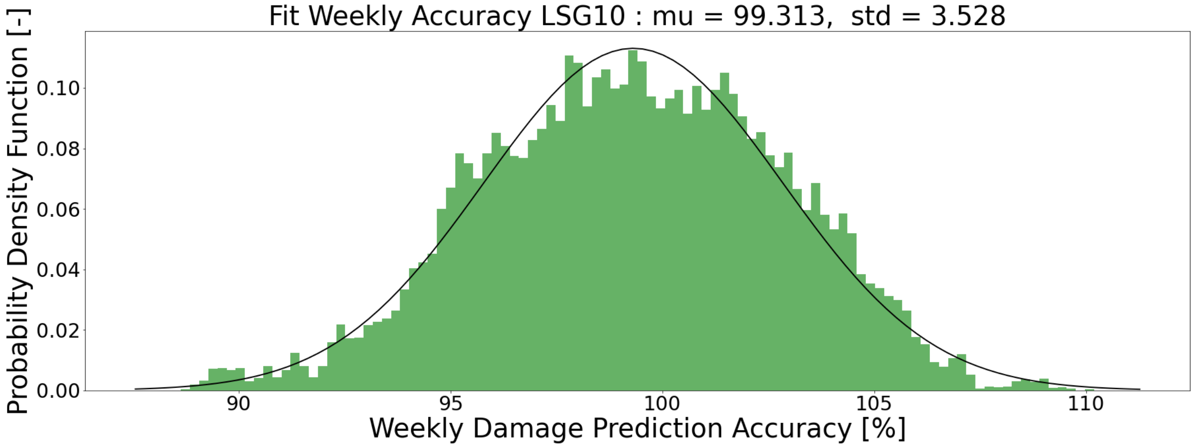


Figure 5.12: Model accuracy of LSG10 fitted with a normal distribution

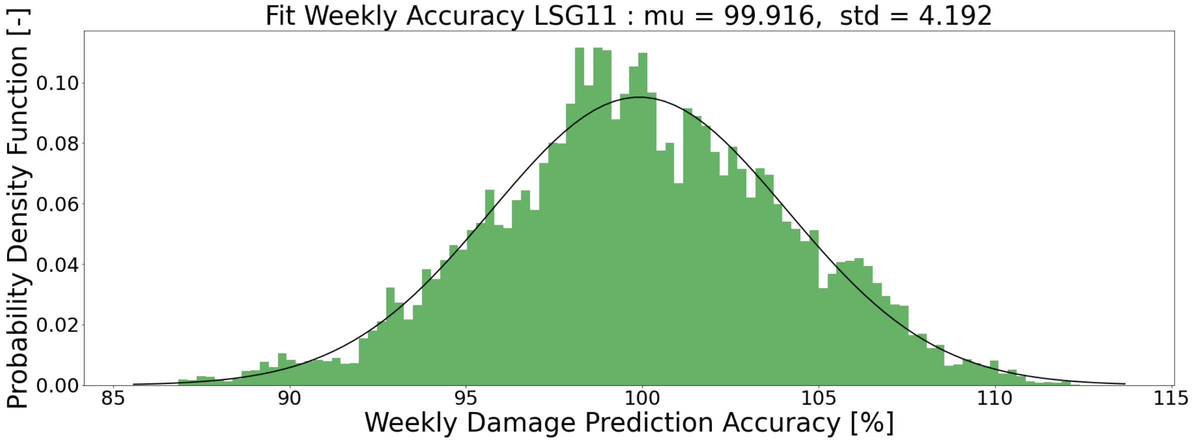


Figure 5.13: Model accuracy of LSG11 fitted with a normal distribution

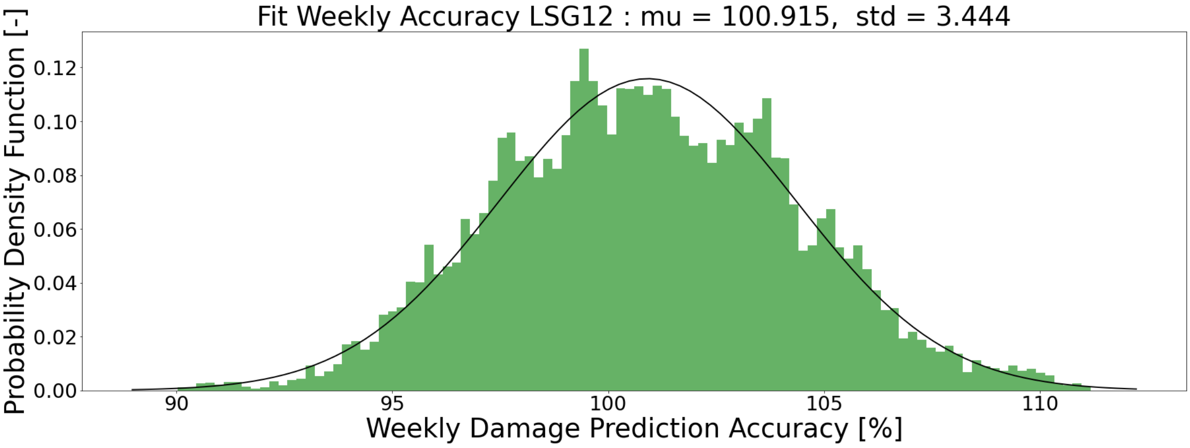


Figure 5.14: Model accuracy of LSG12 fitted with a normal distribution

The performance of the developed neural networks over the whole test set is also displayed in Figures 5.15, 5.16, 5.17, 5.18 and 5.19,. As it can be observed the predicted fatigue damage accumulation is almost identical with the measured one, with all the models only slightly over- or under-predicting. Moreover, the performance of the models serves as a verification of the assumption that they would have a similar behaviour.

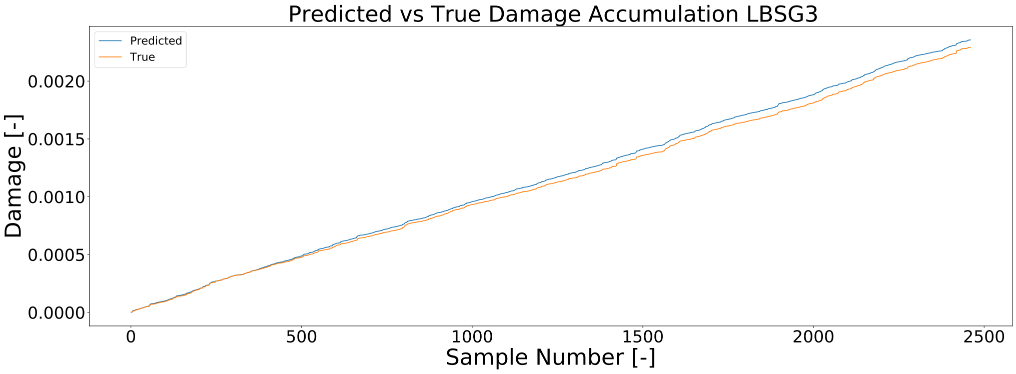


Figure 5.15: True vs Predicted fatigue damage accumulation at LBSG3

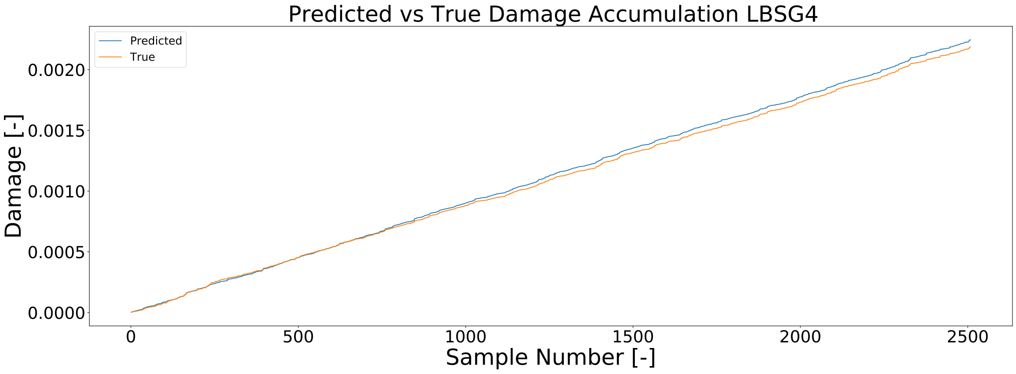


Figure 5.16: True vs Predicted fatigue damage accumulation at LBSG4

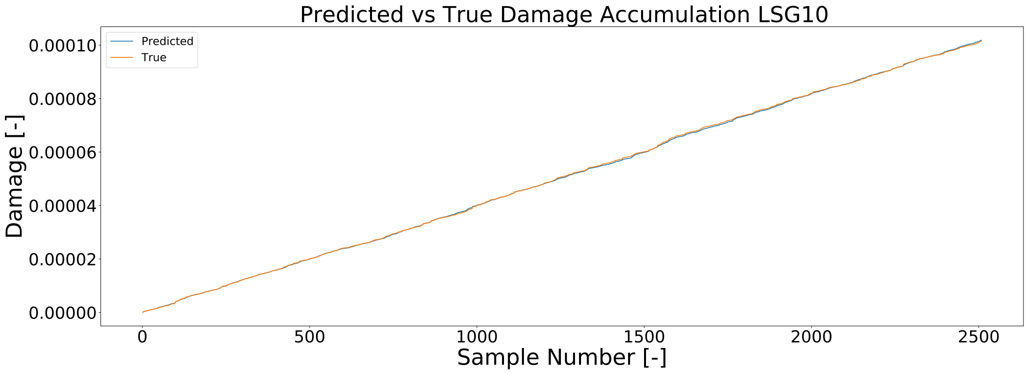


Figure 5.17: True vs Predicted fatigue damage accumulation at LSG10

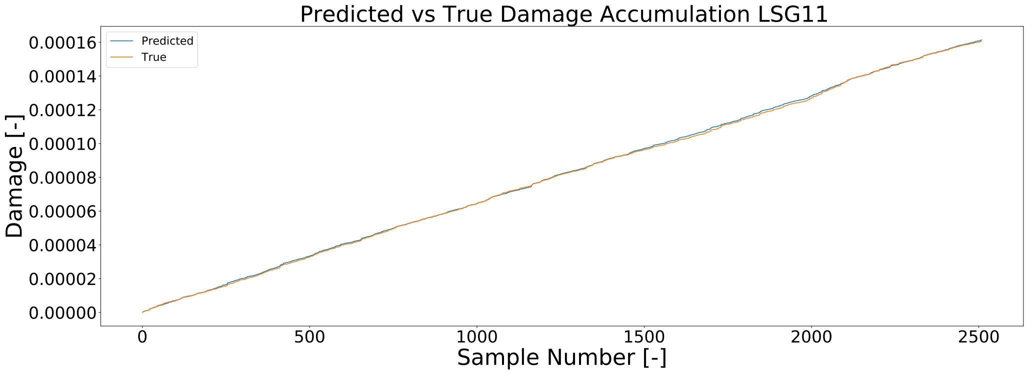


Figure 5.18: True vs Predicted fatigue damage accumulation at LSG11

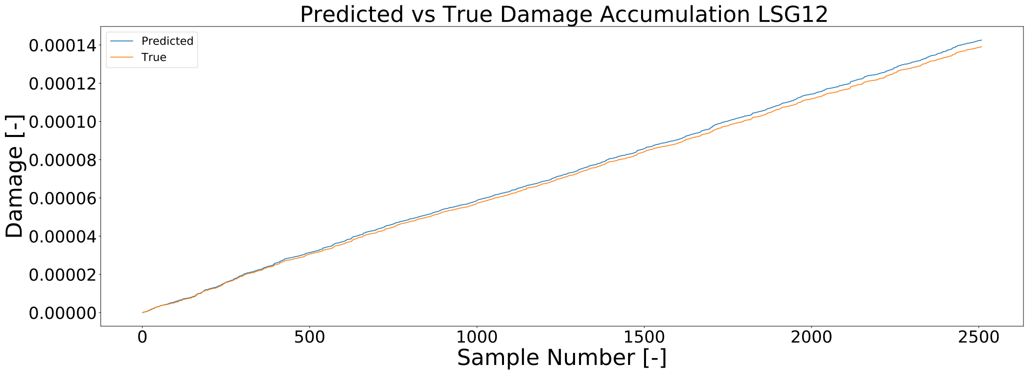


Figure 5.19: True vs Predicted fatigue damage accumulation at LSG12

5.3. Response Prediction

After developing the machine learning models that predict the fatigue damage accumulation at the structural details, the models that predict the frequency, in which the peak stress power density is located at, were developed. The extraction of these frequencies, in order to be used as label values, was performed using the strain times-series that were recorded and a Python package that translates them to power density spectrums that was provided by MARIN. The extraction was performed on the filtered dataset.

5.3.1. Feature Selection for Response Peak Frequency Prediction

The features considered for the fatigue damage predicting application provide information regarding the wave spectrum that were deemed relevant for the frequency prediction problem as well. In order to select the features for this application the same features were once again taken into consideration:

- Peak energy frequency
- Peak energy direction
- Peak energy magnitude
- Direction range
- Frequency range
- Zero-up crossing period
- Mean wave period
- Significant wave height

Initially, the features were visualized in order to inspect the formation of clusters or any underlying trends. As it can be observed in Figure 5.20, the significant wave height seems to have a weak relationship with the frequency where the maximum stress power density is observed. Higher wave heights are related to lower maximum stress power density frequencies. This was expected, since higher wave heights are usually related to sea states with lower frequency contents. From the same illustration one can also conclude that for sea states where the maximum wave energy is located at a higher frequency the maximum stress power density is also located at a higher frequency. The data in the illustration appear in vertical lines. This happens because the peak energy frequency is not a continuous variable due to the way that data is stored during the data acquisition procedure.

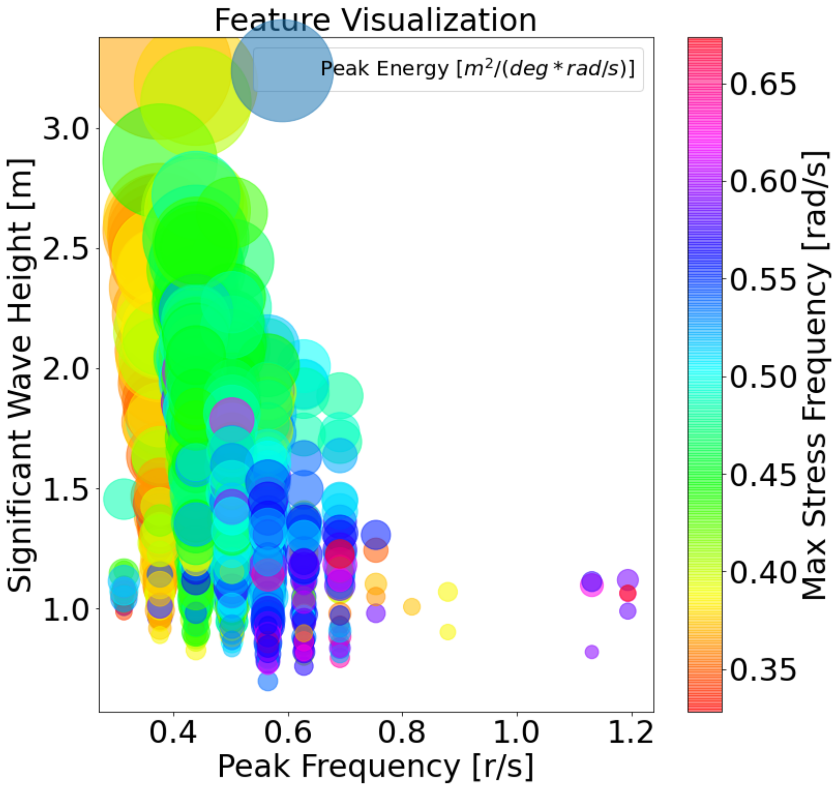


Figure 5.20: (Peak Frequency, H_s, E) Feature Visualization, April 2018, Side Shell Sensor

The inverse proportional relationship of the significant wave height and the maximum stress power density frequency can also be verified in Figure 5.21 . An inverse proportional relationship between the zero up crossing period and the maximum stress power density frequency also seems to appear. However, this relationship is not entirely clear since the low and high maximum stress power density frequency data clusters overlap to a certain extent. What becomes more clear is that sea state with lower maximum wave energy content result in higher maximum stress frequencies.

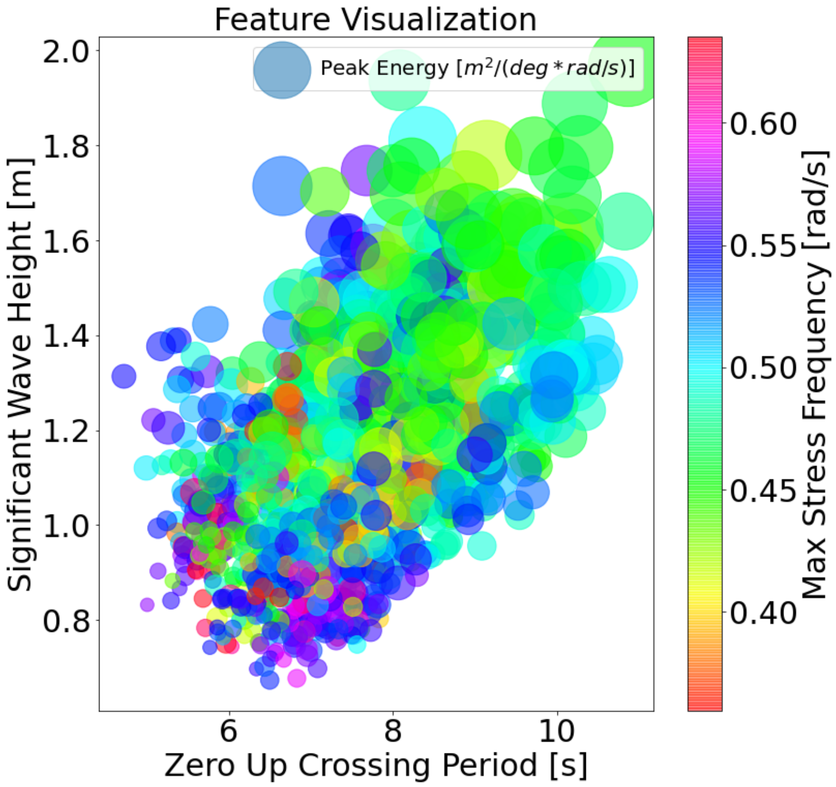


Figure 5.21: (T_0, H_s, E) Feature Visualization, December 2018, Deck Sensor

After the visual inspection of the features the linear correlation between the maximum stress power density frequency and the features was examined. This was once again done through the calculation of the population Pearson correlation coefficient. As it is displayed in Table 5.2 the linear correlation of the features is weaker than for the damage prediction application. Once again all the features that do not display an absolute correlation value larger than 0.2 for all the sensor locations were dropped. This means that the Peak Direction, Frequency Range and Direction Range features were dropped. The final feature vector has the following form:

$$[PeakEnergy, \omega_{peak}, H_s, T_0, T_s] \tag{5.19}$$

After the final selection of the features they were extracted from the available data thus building the final form of the dataset. The full dataset was then split to a training set and a test set with a ratio of 80-20 according to common practice. The test set was seen by the model only during the final evaluation.

Table 5.2: Pearson correlation coefficient of features and labels

	LBSG3 Peak Stress Frequency	LBSG4 Peak Stress Frequency	LSG10 Peak Stress Frequency	LSG11 Peak Stress Frequency	LSG12 Peak Stress Frequency
Peak Energy	-0.276	-0.269	-0.313	-0.319	-0.320
Peak Frequency	0.399	0.377	0.434	0.441	0.441
Peak Direction	-0.199	-0.179	-0.173	-0.173	-0.177
Frequency Range	0.143	0.127	0.196	0.208	0.210
Direction Range	0.046	0.041	0.069	0.076	0.077
Significant Wave Height	-0.242	-0.241	-0.286	-0.294	-0.293
Zero Up Crossing Period	-0.286	-0.273	-0.334	-0.343	-0.345
Mean Wave Period	-0.350	-0.336	-0.397	-0.406	-0.408

5.3.2. Model Selection for Response Peak Frequency Prediction

In order to select the appropriate model for the prediction of the frequency that corresponds to the maximum stress power density three different models were tested:

- Random Forest
- Support Vector Regression
- Gradient Tree Boosting

The values of the hyperparameters of these algorithms were initialized based on values used in similar applications. Then through trial and error these parameters where slightly altered to bring the performance of the algorithms closer to the optimum.
The hyperparameters of the Random Forest were set to be:

- Number of Estimators:100
- Bagging: Enabled
- Number of Features: Auto

Where the number of estimators refers to the number of decision trees and the number of features to the number of features used for the training of each decision tree.Bagging is a meta-algorithm designed to improve the performance and the stability of machine learning methods that are based on decision trees[108].
The hyperparameters of the Support Vector regression were selected to be:

- Kernel Function:Radial Base Function
- C:20
- ϵ:1e-2

The mathematical meaning of each one of these hyperparameters was explained in Chapter 2.
For the Gradient Tree Boosting model the following selection of hyperparameters was made:

- Number of Estimators:100
- Learning Rate: 0.1

The Number of Estimators refers to the number of decision trees that are utilized in order to minimize the residual error. Learning Rate determines affects the correction rate in the error function in each stage of the algorithm. The contribution of each tree is shrunk by the learning rate. Learning rates below 1.0 are used in order to avoid overfitting of the results.
Using the above mentioned hyperparameters for each one the models, their performance was evaluated using cross validation. This means that each model was tested and evaluated multiple times using a different training and validation set each time. The statistics of the performance metric were exported afterwards. The

Table 5.3: Cross Validation Results

	Random Forest	Support Vector Machine	Gradient Tree Boosting
Mean RMSE	0.066	0.072	0.066
RMSE STD	0.002	0.0024	0.0022

selected metric for this application is the root of the mean squared error. The number of training and evaluation iterations for this application, taking into account the required computational time, was selected to be 10. Since cross-validation is a time demanding process the comparison was done using only LBSG3 data. It was assumed that the models perform in a similar way in all sensor locations. The results of the cross validation are presented in Table 5.3.

As it becomes evident from the results of the cross validation Random Forest and Gradient Tree Boosting were equally accurate in their predictions. However, the lower standard deviation of the Random Forest implies that it is more consistent in its performance. Taking this into account Random Forest was selected as the prevailing model for this application.

5.3.3. Fine-tuning and Results

As it has already been stated, the selection of the hyperparameters was performed based on similar applications and it was then improved through a trial and error procedure. In order to obtain the best possible results after the selection of Random Forest as the best of the three algorithms, its hyperparameters were now fine-tuned using a grid search approach. This means that several different values and combination of values of hyperparameters were tested and then the best performing model was selected. The hyperparameter values that were tested are:

- Number of Estimators:[10, 50,100, 200, 400, 600]
- Bagging: [Enabled, Disabled]
- Number of Features: [Auto, Sqrt, Log2]

After testing all the possible combinations of the above mentioned values the best performing set up was:

- Number of Estimators:600
- Bagging: Enabled
- Number of Features:Sqrt

The grid search has once again been performed only taking LBSG3 into account. The performance of the model is displayed in Table 5.4. The accuracy has been defined as:

$$accuracy = \frac{f_{predicted}}{f_{true}} * 100 \tag{5.20}$$

As it can be observed the fine-tuned model performs slightly better than the set-up used during the comparison of the algorithms. The high accuracy and the low mean squared error of the model are indicative of its high predictive capacity. Moreover, the low standard deviation of the mean squared error proves the consistency of the models high accuracy.

Table 5.4: Performance of the fine-tuned Random Forest model

Accuracy	93.15%
Mean RMSE	0.065
RMSE Std	0.0018

The performance of the fine-tuned model is visualized in Figures 5.22, 5.23, 5.24, 5.25 and 5.26. As it can be observed most of the instances are concentrated in close proximity to the 45°line. These illustrations indicate the accurate and consistent behaviour of the final model. They also confirm the assumption that the model performs similarly in all the sensor locations.

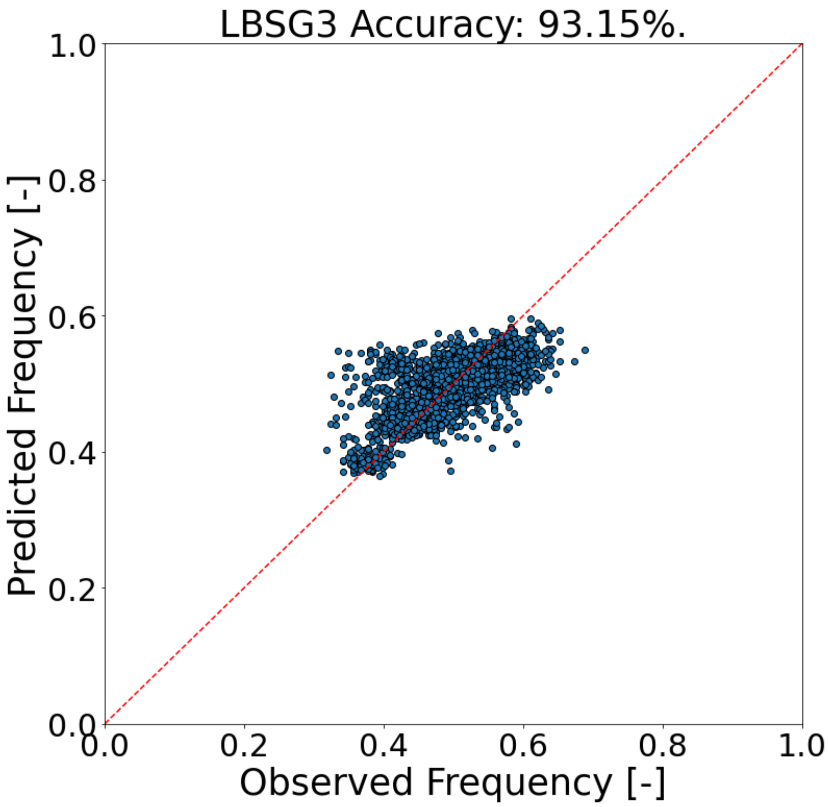


Figure 5.22: True vs Predicted LBSG3

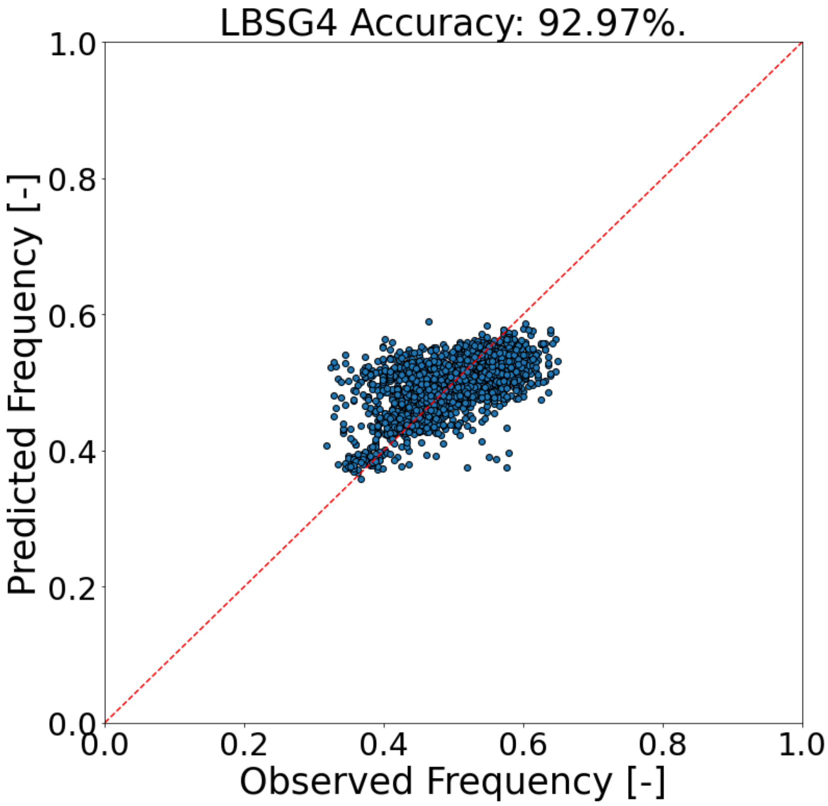


Figure 5.23: True vs Predicted LBSG4

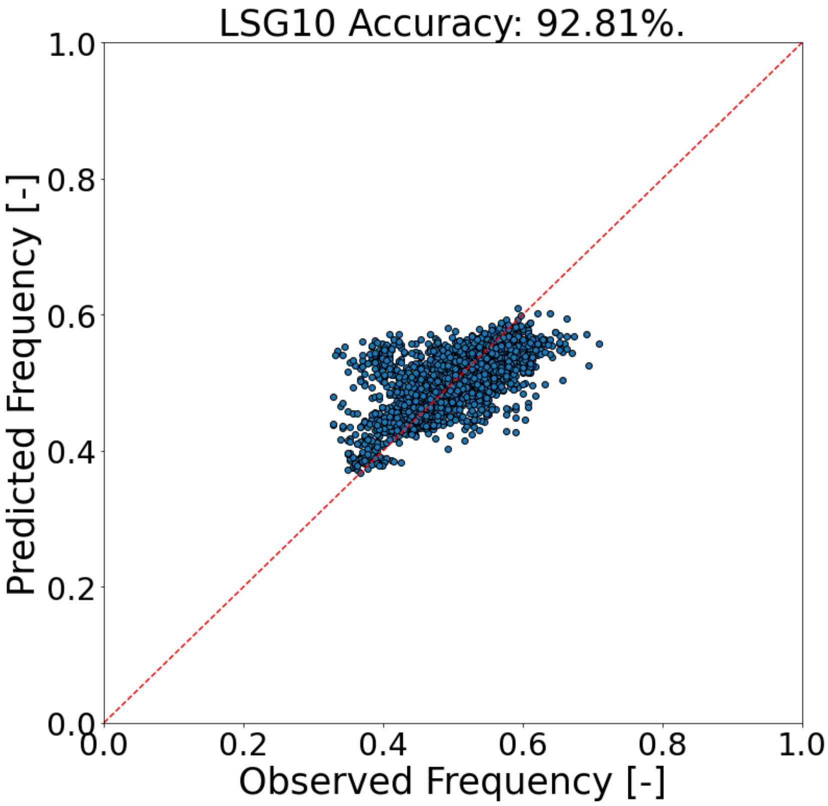


Figure 5.24: True vs Predicted LSG10

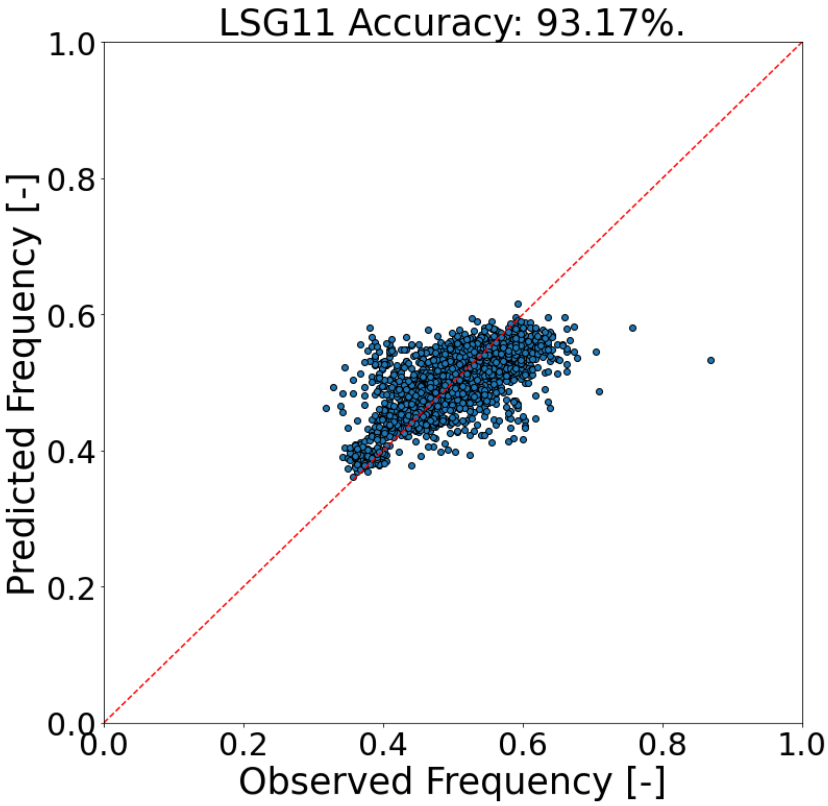


Figure 5.25: True vs Predicted LSG11

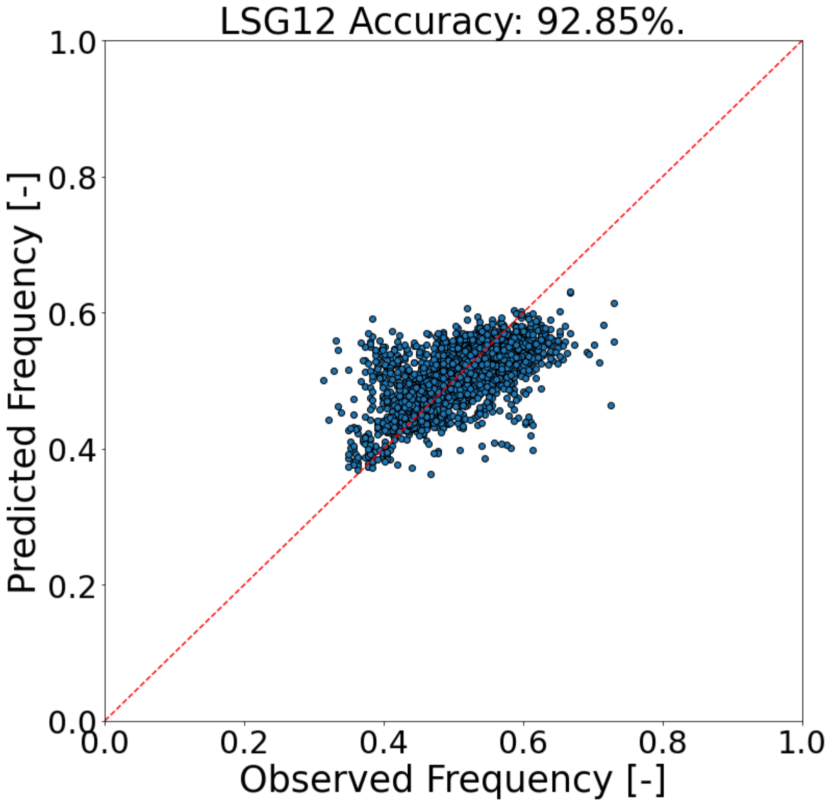


Figure 5.26: True vs Predicted LSG12

5.4. RAO Calibration

As it has already been mentioned the hydro-mechanical modeling of marine structures is related to high uncertainty. As a result the estimated RAOs of a structure are not always accurate. In order to address this uncertainty, an RAO calibration framework has been developed. The developed RAO calibration process can be performed either using actual measurements of using the predictions of the models that have been developed in the previous sections. In order to calibrate the RAOs two parameters have been used:

- α a correction factor that shifts the frequency of the RAOs so that the sum of squared errors between the measured and the calculated frequency of the maximum stress power density is minimized
- β a scaling parameter that calibrates the magnitude of the RAOs so that the fatigue damage calculated through spectral fatigue analysis matches the predicted or measured fatigue damage

The calculation of the above-mentioned parameters is done using Scipy's minimize function from the optimization toolbox [109]. More specifically, Powell's method was used to contrain the estimated parameters to realistic values.

A converge study has been performed the results of which are illustrated in Table 5.5. As it becomes evident

Table 5.5: Convergence Study results of RAO calibration

	LBSG3		**LBSG4**		**LSG10**		**LSG11**		**LSG12**	
Nr. Of Samples	alpha	beta	alpha	beta	alpha	beta	alpha	beta	alpha	beta
10	-0.108	1.008	-1.000	9.125	-0.1685	1.0742	-0.041	0.579	-0.038	0.870
50	-0.004	0.848	-0.020	0.906	-0.1647	1.1695	0.042	0.562	-0.179	0.946
100	-0.003	0.840	-0.109	1.153	-0.1315	1.0710	0.043	0.551	-0.138	0.904
200	-0.141	1.137	-0.110	1.097	-0.0826	0.9834	0.016	0.534	-0.005	0.812
400	-0.081	0.982	-0.066	0.954	-0.1342	1.0082	0.019	0.538	0.000	0.821

convergence has not been achieved in all sensor locations for a sample size up to 400 (200 hours). In order to achieve convergence for a similar process implemented by MARIN 3 months worth of sample (4320 samples) were required [5]. The computational time required by the RAO calibration procedure did not allow for a convergence study up to such a number of samples to be performed.
The results of the RAO calibration procedure for the RAO of LSG10 using $\alpha = -0.1315$ and $\beta = 1.071$ are illustrated in Figure 5.27.

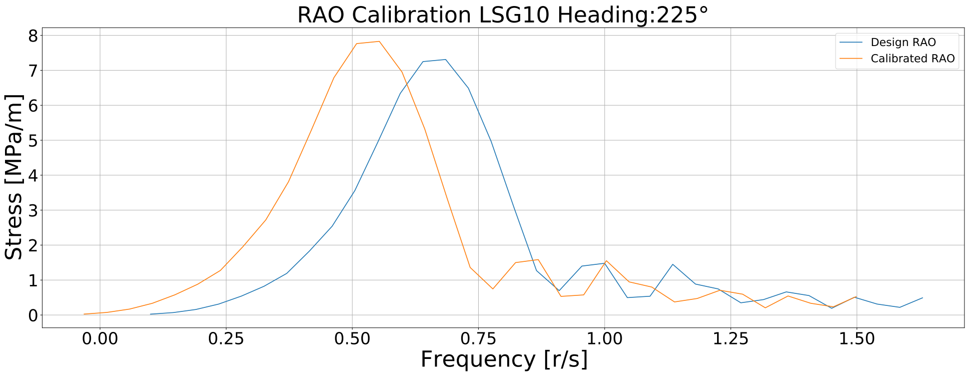


Figure 5.27: Design vs Calibrated RAO

Chapter 6

Load Magnitude Calibration Algorithm

Calculation of fatigue accumulation in marine and offshore structures is subjected to many uncertainties. A major source of uncertainty is the hydro-mechanic modeling of the structure [5]. Adopting wave load modelling parameters from codes and standards frequently leads to overly conservative predictions of fatigue lifetime. Subsequently, in order to achieve a high fidelity digital twin, in-situ measurements should be used to calibrate the actual loading on the structures [12].

In the context of this research an algorithm has been developed that allows the calibration of the load magnitude on a structure using strain measurements. The algorithm is based on a reformulation of the Finite Element Method as it has been described in Chapter 3 and the advisory hull monitoring system installed on the unit used as a case study, which is highly similar to the one installed on Bonga FPSO [1].

6.1. Description of the Algorithm

As it has already been stated in Chapter 3 by reformulating the classic FEM problem it is possible to end up with an algorithm that not only calculates the displacement vector but also calibrates the load. The required input for such an algorithm is:

- The stiffness matrix $[K]$ of the structure.
- The boundary conditions.
- Strain measurements somewhere on the structure.
- A shape vector that describes the application points, the direction and the comparative magnitude of the load with respect to a maximum.

In the scope of this project, the stiffness matrix with the boundary conditions applied using the penalty method [110] is exported from Abaqus. The strain measurements are acquired through the advisory hull monitoring system and the shape vectors have been generated so as to model the major load components of global hull bending and local water pressure [111]. The output of the algorithm is:

- The displacement vector, containing the displacement of each node.
- The calibration factor a.

All the calculations have been implemented in Python. Sparse data format has been used in order to decrease memory usage and computation times. The solution of the linear system is performed using LU decomposition.

6.2. Verification of the Results

With the theoretical basis and the description of the algorithm in place, its functionality should now be verified. In order to do so, a simulation has been run using FEM with a known applied force. Then magnitude of the force was regarded as an unknown, and the problem was simulated using the Load Magnitude Calibration (LMC) algorithm. The strain data the were required, were obtained form the post-processing of the initial FEM simulation. To avoid unnecessary complexity, the simple case of a fixed plate in uni-axial tension was simulated for verification purposes.

More specifically, a plate with dimensions $2000mmx2000mm$ and a thickness of and a thickness of $1mm$ has been selected. The material of the plate was set to be steel with a Young's Modulus of $210000MPa$. The discretizetion was performed using S4 shell elements [112] with an element length of $40mm$. The plate was

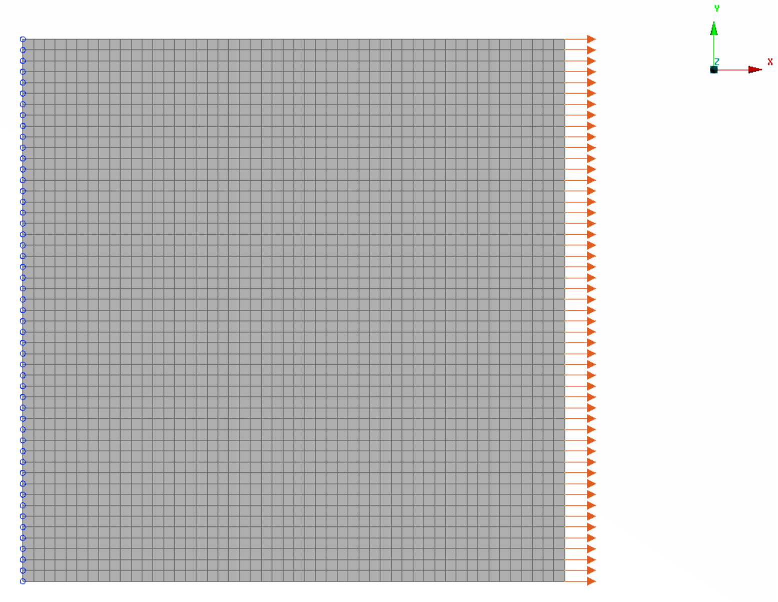


Figure 6.1: Verification Model

fixed in the left end and a uniform tension load of $2040N$ was applied on the other end. The strain gauge was modelled using a truss T3D2 [112] element which is a constant strain element. The area of the truss element was set to be $1mm^2$ and its stiffness $1000MPa$. They were selected to be so in order to be indicative of the low stiffness of the strain gauge. The verification model is displayed in picture 6.1.
The total error in the load after the calibration was calculated as it is shown below:

$$error = \frac{\sum_{i=1}^{n} f_i - \sum_{i=1}^{n} a * b_i}{\sum_{i=1}^{n} f_i} \tag{6.1}$$

where f_i are the components of the initial load vector, b_i are the components of the shape vector and a is the calibration factor. The calculated error was far below 0.01%. The calculated nodal displacements were also identical with the ones of the FEM simulation.

6.3. Sensitivity Analysis

In the previous section the ability of the LMC algorithm, to provide the results it was expected to, has been established.Subsequently, a sensitivity analysis was performed in order to monitor the behaviour of the algorithm in more complex structures and loads. The performance of the algorithm in the following cases was tested:

- The introduction of error in the measurement. A uniform tension load on the free edge was applied in this case.
- The ability to find the average of a varying load using multiple measurements. The varying load that was used had the shape of uniform tension load blocks with different magnitudes. The load was applied on the free edge.
- The effect of the measurement location on the load calibration. A load identical to the varying load of the previous case was applied.
- The reconstruction of a load with a quadratic shape using multiple measurements. The quadratic load was applied on the free edge of the plate.
- The effect of multiaxiality. The examined load case was uniform tension loads of different magnitudes applied on the free edges of the plate.

- The effect of more complicated structures. A uniform pressure was applied on the structure as a load case.

Examining the above mentioned cases provided better insight regarding the effectiveness and the robustness of the algorithm.

6.3.1. Error in the Measurement

In all the experiments a certain degree of error is present in the acquired measurements. Because of this, it was of utmost importance to monitor how the LMC algorithm responds to error introduced in the measurements. Such an analysis enables the estimation of the error in the results, under the condition that a valid estimation of the measurement error is in place.

For the above mentioned analysis, the example used for the verification of the functionality of the algorithm was used once again. However, the strain data was distorted by the introduction of an error. The error was introduced using the following formula:

$$\epsilon_{distorted} = \epsilon_{measured} * f_e \tag{6.2}$$

where f_e is a scalar in the range [1,2]

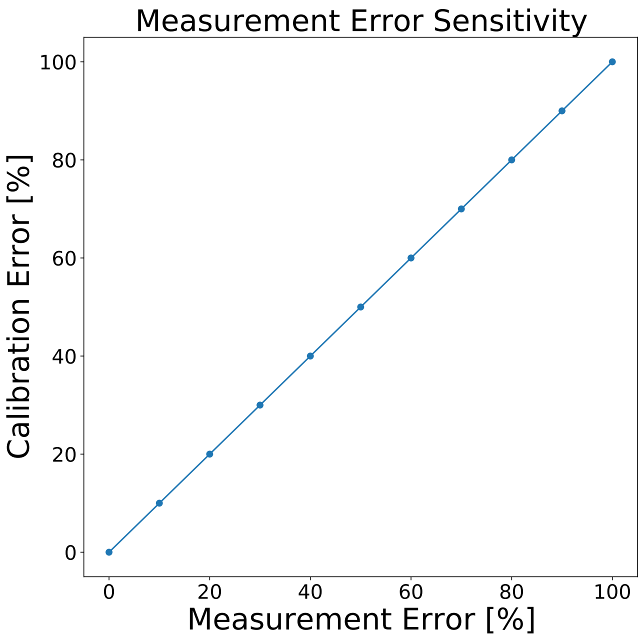


Figure 6.2: Sensitivity of calibration to introduced error

The analysis was performed for a number of different error levels and led to the conclusion that the error in the calibration is equal to the error introduced in the measurement. The results of the analysis are shown in Figure 6.2.

6.3.2. Averaging a Varying Load

The majority of the loads encountered in real-life applications are not uniform in magnitude. However, if their amplitude is low they are frequently modelled as such for simplification of the load case. For this reason, the ability of the algorithm to calculate the uniform equivalent of a varying load was examined. In order to test the performance of the algorithm a new experiment was set in place. Once again an identical plate fixed on the one side was used. In terms of loading, as load of varying magnitude was used in the FEM analysis as it indicated in Figure 6.3.

In the Load Magnitude Calibration algorithm the load was modelled with the shape vector of a uniform load. Many strain measurements were taken along the width of the plate. The algorithm was run multiple times us-

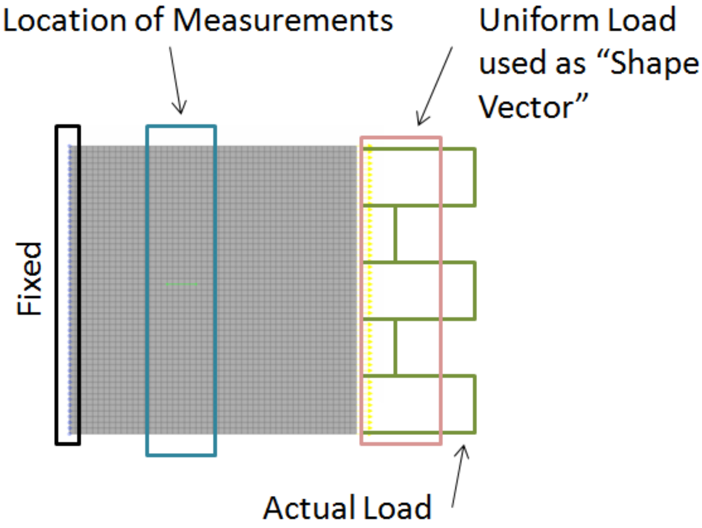


Figure 6.3: Model used to examine uniform load approximation capability

ing each one of these measurements. The final calibration factor was estimated as the mean of the calibration factors that have been calculated.

$$a = \frac{\sum_{i}^{n} a_i}{n} \tag{6.3}$$

The error in the total load magnitude was estimated using Equation 6.1 and the mean calibration factor.As it can be seen in Figure 6.4, the introduction of a second measurement point leads to a reduction of the error value. The introduction of any further points of measurement makes the error value to fluctuate. The low error values that are encountered for two or more measurement points serve as a strong indication that the Load Magnitude Calibration algorithm has the ability to approximate varying load with an equivalent uniform load. This averaging capability was expected due to the Saint Venant's principle [113]. The interpretation of Saint Venant's principle indicates that the exact distribution of a load is not important far away from the loaded region, as long as the resultants of the load are correct.

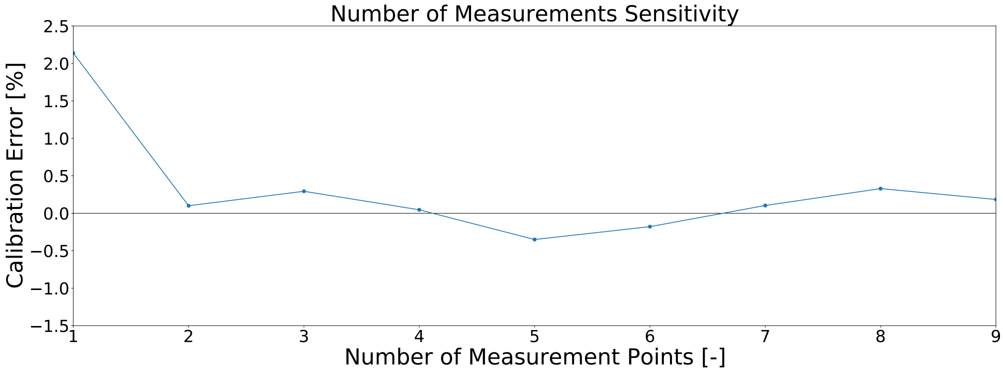


Figure 6.4: Error in total load magnitude

6.3.3. Effect of Measurement Location

Another aspect of the algorithm worth investigating, is the degree in which the location of the measurements is affecting the results. In order to examine this effect, a FEM model similar to the one used in the previous example was used. The only difference is that now strain gauges where placed to another location as well. The exact model configuration is depicted in Figure 6.5.

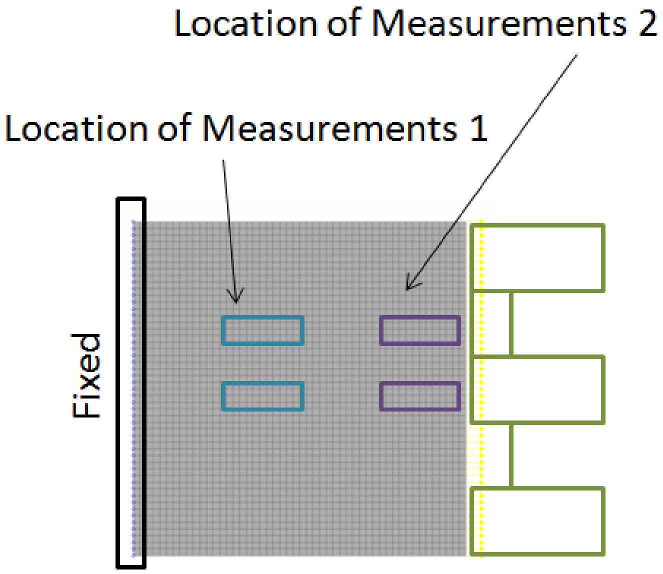


Figure 6.5: Model used to examine the effect of strain gauge location

The major difference with respect to the previous case, was that now the load was modelled in its actual shape using two shape vectors. This means that both measurements of each location would be used simultaneously. The Load Magnitude Calibration algorithm was executed for each location of measurements generating the results that are shown in Table 6.1. The formula used to calculate the error is:

$$error = \frac{\sum_{i=1}^{n} f_i - \sum_{j=1}^{m} \sum_{i=1}^{n} a_i * b_{ij}}{\sum_{i=1}^{n} f_i} \tag{6.4}$$

Table 6.1: Calibration error with respect to measurement location

	Location 1	Location 2
Error in Low Magnitude Load Block	8.1%	-6.34%
Error in High Magnitude Load Block	-39.41%	-2.96%
Error in Total Load Magnitude	-7.35%	-5.24%

This was expected since the loading at Location 1 has been almost completely redistributed according to the Saint Venant's principle[113] meaning that any local effects from the load blocks are almost completely lost. This loss of information leads to the large error values that are witnessed. When Location 2 is examined, where the strain field is still highly affected for the different load blocks in a local manner, the calibration algorithm exhibits far superior results.

6.3.4. Reconstruction of a Load with Quadratic Shape

Loads with a more complicated shape are a common occurrence in engineering applications as well. Subsequently, it is essential to monitor the effectiveness of the algorithm in such loads. In order to do so a tension load of quadratic shape was applied at the edge of the plate as it is illustrated in Figure 6.6. In the Load Magnitude Calibration algorithm the load was modelled using a number of measurement points each one of them corresponding to the calibration of a uniform load block. The measurement points were placed close to the loaded edge to ensure optimal efficiency of the calibration. The efficiency of the algorithm was measured using the error in the estimation of total load magnitude as an indicator. Equation 6.4 was used to calculate the error. It has to be noted that the use of more measurement points enhances the shape representation of the load since more load blocks are used, thus approximating the quadratic shape better.

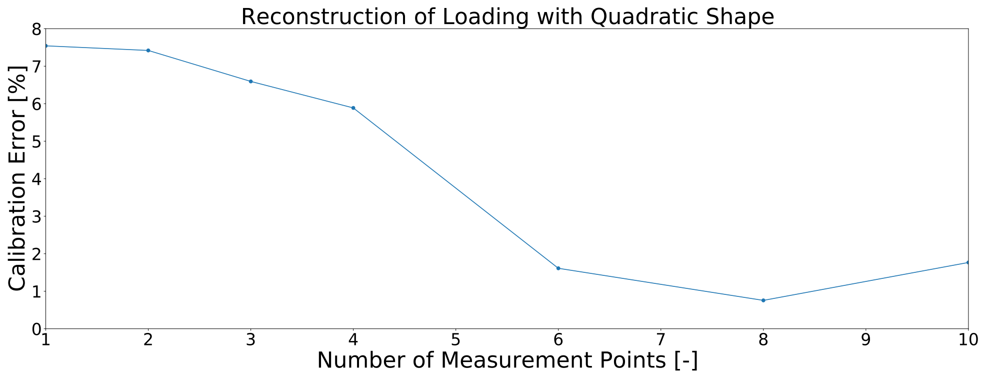


Figure 6.6: Model used to examine the reconstruction of a quadratic load

As it can be witnessed in Figure 6.7, when six or more measurement points are used the calibration error drops below 2% after that point the addition of more measurement points, only enhances the representation of the load in terms of shape representation. The error in the calibration of the total load magnitude starts oscillating. This analysis indicates that using a coarse system of measurement points in combination with the LMC algorithm can lead to an adequate load reconstruction both in terms of magnitude and shape.

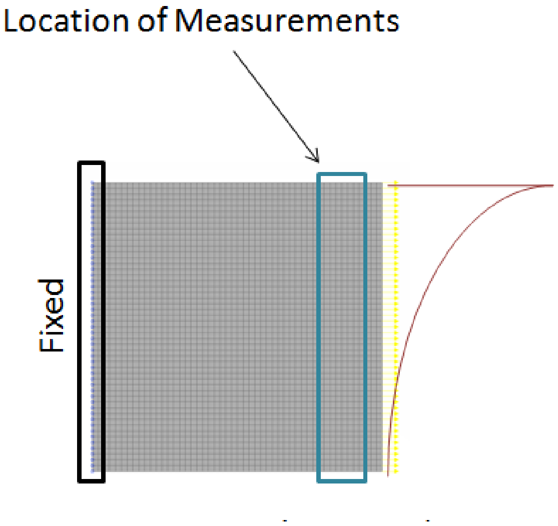


Figure 6.7: Total load magnitude error of the reconstruction of a quadratic load

6.3.5. Effect of Multiaxiality on Load Reconstruction

Multiaxiality is a common state of loading in engineering applications and thus it was deemed crucial to study the performance of the algorithm when the structure experiences multiaxial loading. The model that was selected in order to monitor this aspect of the algorithm's sensitivity is once again a plate, now loaded in both directions with uniform loads as it is shown in Figure 6.8.

The analysis was performed in two parts. In the first part, the load along the width axis was not taken into account in the calibration algorithm and only the load along the length axis was calibrated. As it was expected this lead into an error in the calibration. The magnitude of this error is depicted in Figure 6.9 as a function of the magnitude of the load along the width axis. It becomes evident that relationship between the error and the load magnitude is linear. One can also conclude that for weak multiaxiality conditions the value of the error is not significant and thus the secondary load can be disregarded in the calibration algorithm without ending up with a considerable calibration error in the main load.

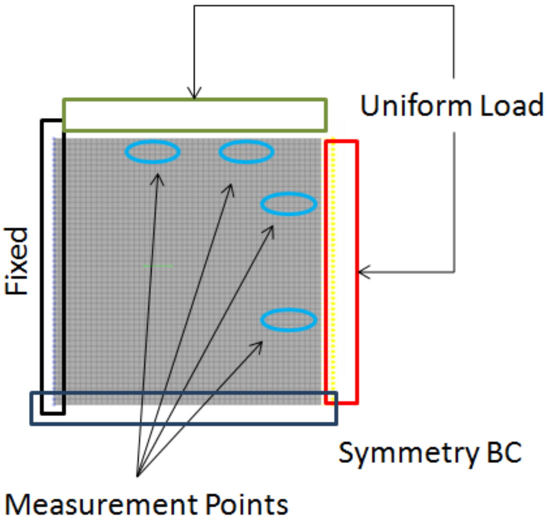


Figure 6.8: Multiaxial Loading Model

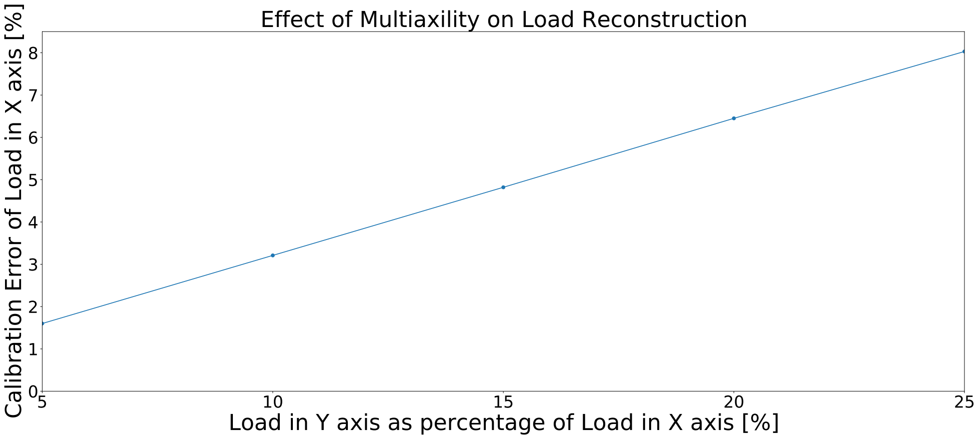


Figure 6.9: Calibration error with respect to the level of multiaxiality

However, multiaxiality is not always weak and thus disposable. To account for this, the second part of the analysis was performed. In this part two measurement points were used simultaneously in order to calibrate both loads. The measurement points where aligned both along the width and the length axis. As it becomes evident by the results presented in Table 6.2, both loads can be accurately calibrated. Furthermore, the orientation of the strain gauges has effects of lesser importance on the results.

Table 6.2: Calibration error in a multiaxial load state

	Load X Error[%]	Load Y Error[%]
2 Points in X-axis	0.007615490	0.0712379
2 Points in Y-axis	0.000466037	0.000430532
1 Point in each axis	0.000111665	0.000102558

6.3.6. Load Reconstruction on a Stiffened Panel

So far the algorithm was only tested on plates. In order to obtain a complete understanding of the behaviour of the algorithm, it should be tested on more complicated structures as well. A stiffened panel has been selected as the structure on which the algorithm would be tested, since it is one of the most common structures in marine applications. The stiffened panel was fixed along its circumference and subjected to a uniform pressure. The geometry of the structure and the boundary conditions applied are illustrated in Figure 6.10.

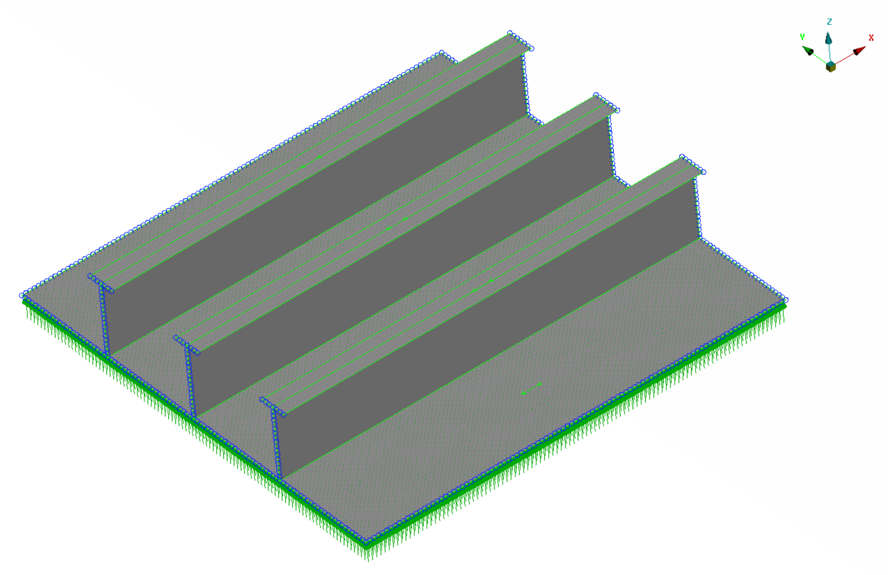


Figure 6.10: Stiffened Panel Model

One strain gauge was placed on each stiffener and bay of the panel. The load was reconstructed using one measurement at a time. The error was calculated using equation 6.1. As the contents of Table 6.3 show the error is almost zero for all the measurement points. This is a strong indication that the algorithm can operate effectively on stiffened panel structures subjected to a similar loadcase.

Table 6.3: Calibration error for a stiffened panel structure

	Error[%]
Sensor 1	0
Sensor 2	0
Sensor 3	3E-06
Sensor 4	1E-06
Sensor 5	0
Sensor 6	8E-06
Sensor 7	1.7E-05

Chapter 7

Structural Reliability Analysis Framework

The neural network models that have been developed in Chapter 5 provide the means to accurately predict the evolution of fatigue damage accumulations in structural details of the FPU that is used as a case study. Taking things one step further, these models will be utilized to develop a structural reliability analysis framework based on the approach that is described in Chapter 4.

7.1. Limit State Function

The first step in the development of a structural reliability analysis framework is the definition of the limit state function. The limit state function $G(X)$ consists of two parts:

- The resistance of the structure $R(X)$
- The loading of the structure $S(X)$

The mathematical formulation of the limit state function is:

$$G(X) = R(X) - S(X) \tag{7.1}$$

In the application under consideration the resistance of the structure is the maximum fatigue damage accumulation and the loading of the structure is the current fatigue damage accumulation of the structure.
As it has been shown in Chapter 5 the developed neural networks can accurately predict the fatigue damage accumulation of the structural details on a weekly basis. However, if the structural reliability analysis needs to be performed on a horizon of several years two major problems arise:

- How will the sea state during those years be forecasted?
- The expected computational time for the predictions is very high.

In order to bypass these problems a different approach has been selected. More specifically the fatigue damage accumulation will be modeled through a design variable that will be named "Typical Weekly Fatigue Damage" (D_{weekly}). The concept behind the definition of this random variable is based on the fact that in Miner's rule(linear damage accumulation) the order in which individual stress cycles are applied does not affect the fatigue damage accumulation. According to this a year of data in which the features(the features are extracted from half hour samples) in every week follow the yearly feature distribution of an actual year of data, and an actual year of data with the features in the order they were acquired, will lead to exactly the same fatigue damage accumulation. The fatigue damage accumulation of a week during which the features follow their yearly distribution is the definition of the design variable "Typical Weekly Fatigue Damage". Taking the above mentioned facts and definitions into account, the "loading of the structure" component of the limit state function for the application under study can be written as:

$$S(X) = D_{total} \approx NumberOfWeeks * D_{weekly} \tag{7.2}$$

It has to be noted that the accuracy of this equation is reduced to a certain extent when the number of weeks under consideration do not account for an integer number of years since the typical weekly fatigue damage disregards seasonal effects. What is more, this formulation of the loading equation does not take the accuracy of the neural network under consideration. In order to account for the accuracy of the models it can be reformulated as:

$$S(X) = NumberOfWeeks * \frac{D_{weekly}}{M_{acc}} \tag{7.3}$$

where the design variable M_{acc} is the accuracy of the model.
According to Miner's rule a structure fails when the fatigue damage accumulation is equal to $1.o$. However, while Miner's rule works reasonably well on the average it has a considerable amount of uncertainty[114]. One the most commonly used models to account for this uncertainty has been developed by Wirsching [115]. According to this model the damage at failure($D_{failure}$ can be described as a design variable with a lognormal distribution that has a mean of 1.0 and a standard deviation of 0.3. Using Wirsching's model the resistance component of the limit function can be written as:

$$R(X) = D_{failure} \tag{7.4}$$

This lead to the final form of the limit state function:

$$G(X) = R(X) - S(X) = D_{failure} - NumberOfWeeks * \frac{D_{weekly}}{M_{acc}} \tag{7.5}$$

Where the design variables are $D_{failure}$, D_{weekly} and M_{acc} and the number of weeks the time horizon of the reliability analysis.

7.2. Typical Weekly Fatigue Damage PDF Estimation

The probability distribution of the failure damage is defined by Wirsching's model and the one of the accuracy of the neural networks has been estimated during the evaluation of their performance in Chapter 5. However, the probability density function of D_{weekly} has been established yet. In order to determine the PDF of D_{weekly} the following steps were followed:

- The distributions best fitting the principal components of the features were determined.
- A large dataset of weeks following these distributions was generated.
- The fatigue damage accumulation of each typical week was predicted using the neural networks.
- The weekly fatigue damage accumulation value that were predicted were fitted in order to determine the PDF.

In order to select the distribution that best fits each one of the principal components of the features the following distributions were tested:

- Normal
- Lognormal
- Weibull
- Double Weibull
- Exponential
- Exponential Normal
- Exponential Weibull
- Exponential Power
- Generalized Normal
- Generalized Exponential
- Generalized Extreme
- Power Law
- Power Lognormal
- Power Normal
- Gamma

The fitting performance of the distribution was evaluated through the summation of squared errors. The prevailing distributions for each principal component are listed below:

- Principal Component 1 : Exponential Power | Mean = -0.82 Std = 1.13 b = 2.12
- Principal Component 2 : Exponential Normal | Mean = -0.08 Std = 0.09 K = 0.84
- Principal Component 3 : Exponential Normal | Mean = -0.07 Std = 0.05 K = 1.23
- Principal Component 4 : Exponential Normal | Mean = -0.04 Std = 0.06 K = 0.70
- Principal Component 5 : Double Weibull | Mean = 0.00 Std = 0.02 c = 1.08
- Principal Component 6 : Exponential Weibull | Mean = -0.16 Std = 0.15 a = 3.27 c = 8.03

The PDFs of all the principal components are illustrated in Figures 7.1, 7.2, 7.3, 7.4, 7.5 and 7.6 .

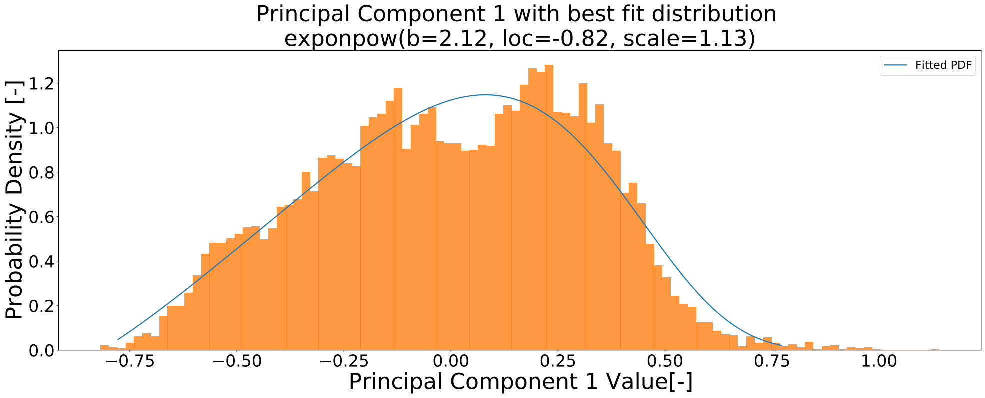


Figure 7.1: Principal Component 1 Fitted with PDF

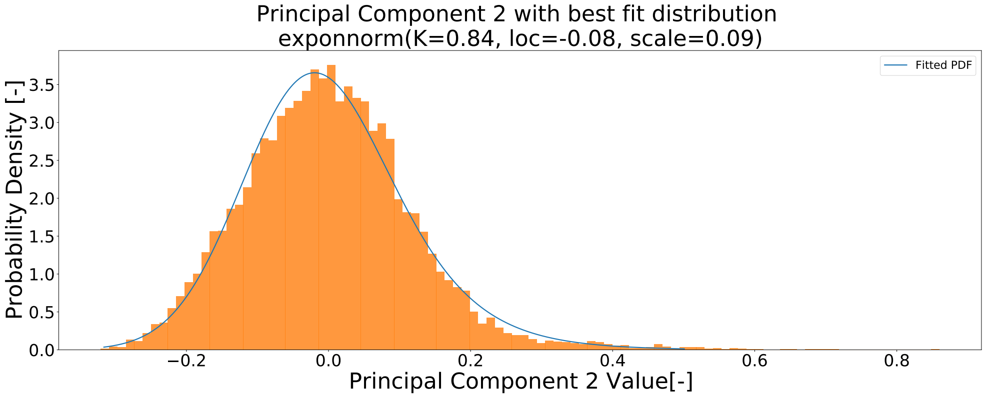


Figure 7.2: Principal Component 2 Fitted with PDF

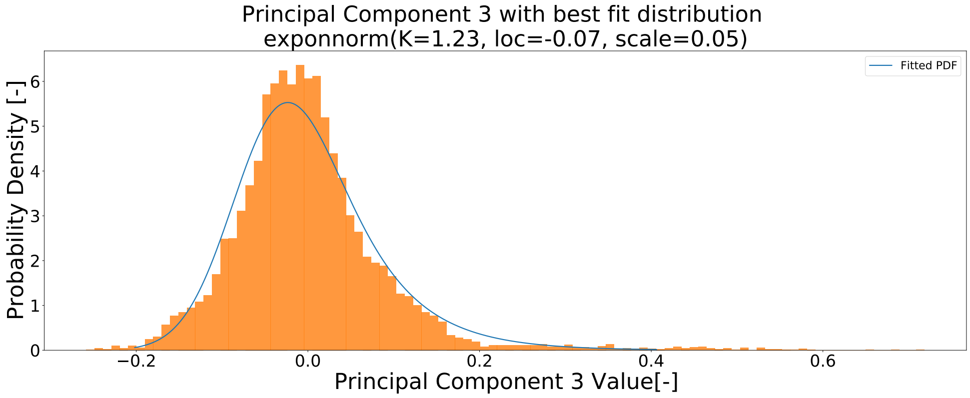


Figure 7.3: Principal Component 3 Fitted with PDF

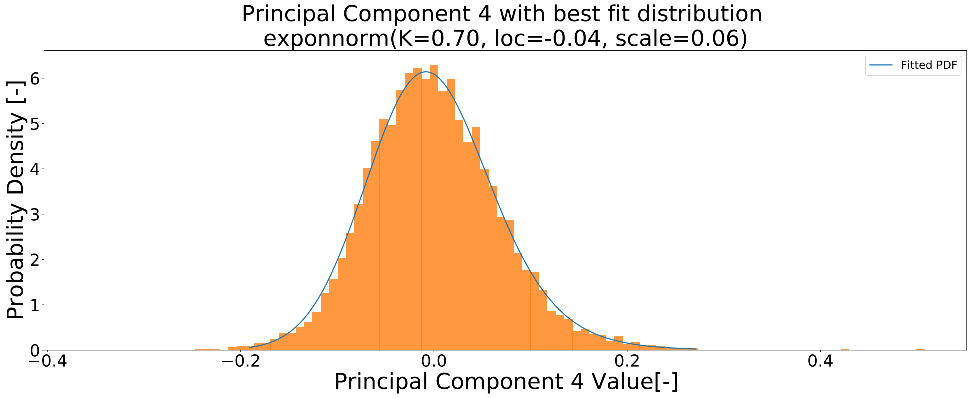


Figure 7.4: Principal Component 4 Fitted with PDF

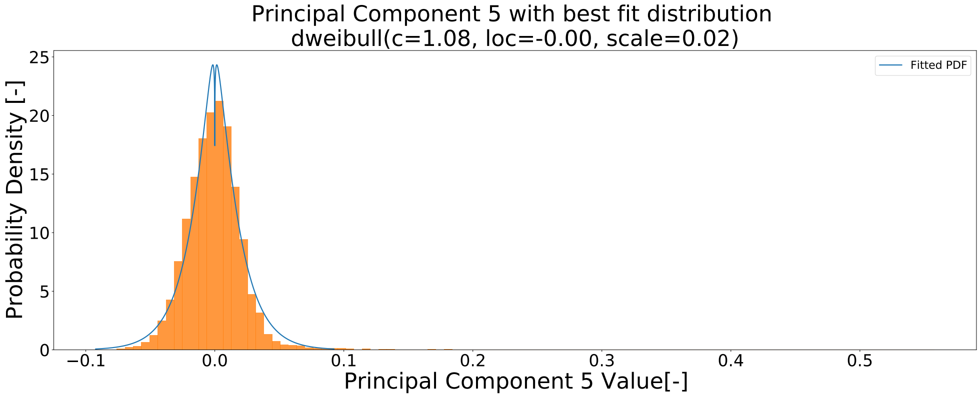


Figure 7.5: Principal Component 5 Fitted with PDF

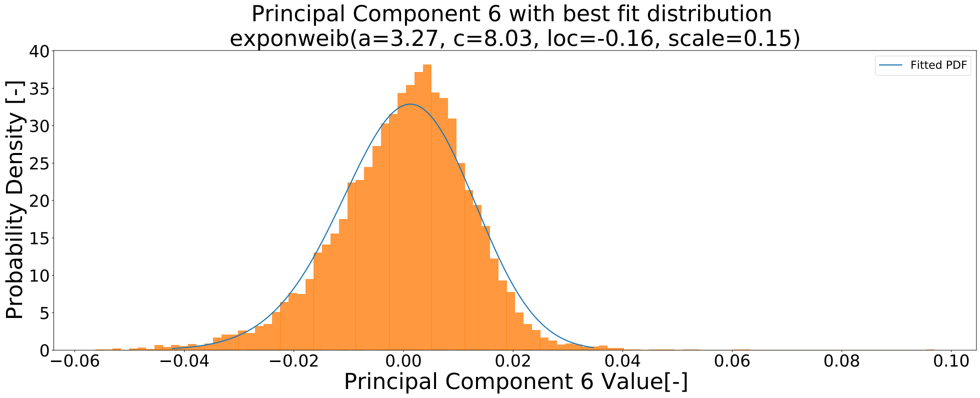


Figure 7.6: Principal Component 6 Fitted with PDF

Since the principal components are uncorrelated by definition typical weeks can be sampled using the distributions of the principal components. Afterwards the damage of the typical weeks was calculated using the neural networks developed in Chapter 5. In order to achieve convergence in the parameters of the typical weekly fatigue damage distribution, 10000 typical weeks were generated. Because the computational time demanded for this procedure is significant it was only performed for one deck sensor(LBSG3) and one side shell sensor(LSG12). The PDF of the typical weekly fatigue damage of these sensors is displayed in Figures 7.7 and 7.8 . The data was fitted using a normal distribution.

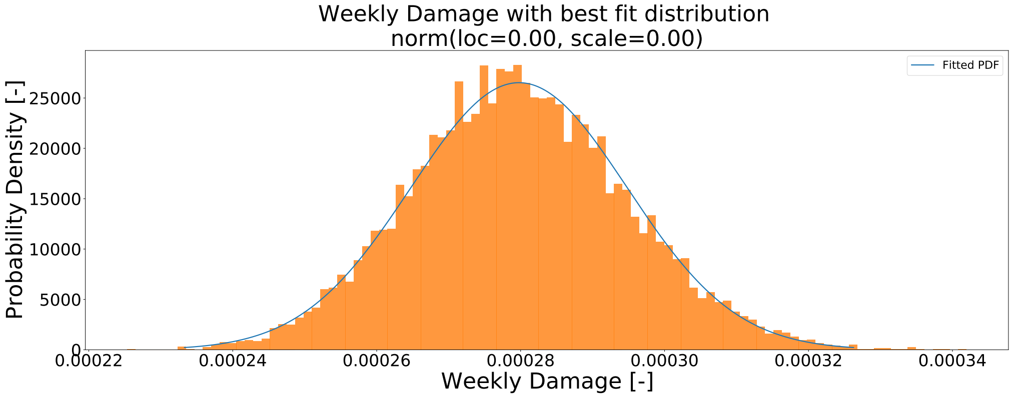


Figure 7.7: Typical Weekly Fatigue Damage PDF LBSG3

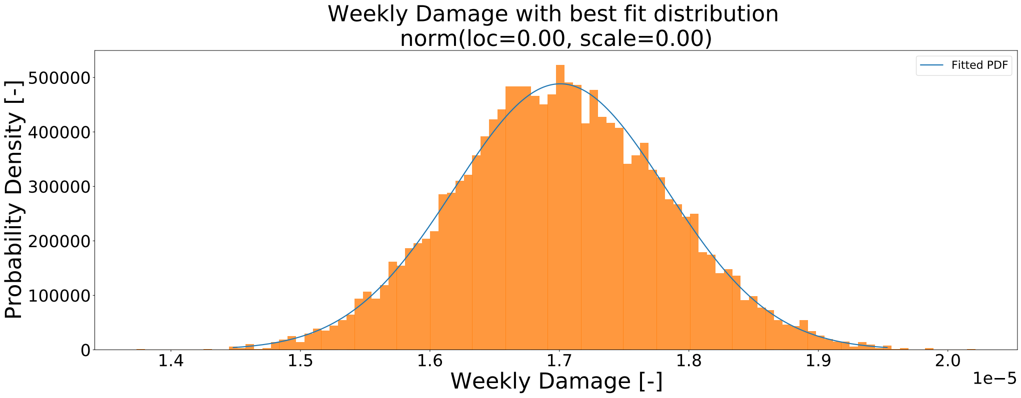


Figure 7.8: Typical Weekly Fatigue Damage PDF LSG12

7.3. Calculation of the Reliability Index and the Probability of Failure

With the final form of the limit state function and the probability distributions of all the design variables in place the reliability index and the probability of failure of the details could now be calculated using the methodology that is described in Chapter 4.
The values of the parameters required by the CPSO algorithm has been selected as:

- w = 0.7
- $c_1 = 1.5$
- $c_2 = 1.2$
- $r_1 = 0.1$
- $r_2 = 0.15$
- M = 1000

The performance of the CPSO algorithm has been compared with the performance of Monte Carlo for a time horizon of 10 years and the resulting error for both sensors is less than 1%. The estimated reliability index and probability of failure of the critical structural detail in proximity to LBSG3 for a time horizon from 1 to 25 years is displayed in Figure 7.9. The estimated reliability index and probability of failure of the critical structural detail in proximity to LSG12 for a time horizon from 10 to 600 years is displayed in Figure 7.10. The side shell structures have large safety margins since the failure of a side shell structure can compromise the integrity of the whole asset. This fact explains the large number of years over which the probability of failure of the side shell structure remains low. It has to be noted that the current reliability analysis does not take into consideration the following parameters:

- Material Degradation effects
- Seasonal Effects
- Uncertainties related to the fatigue resistance curve

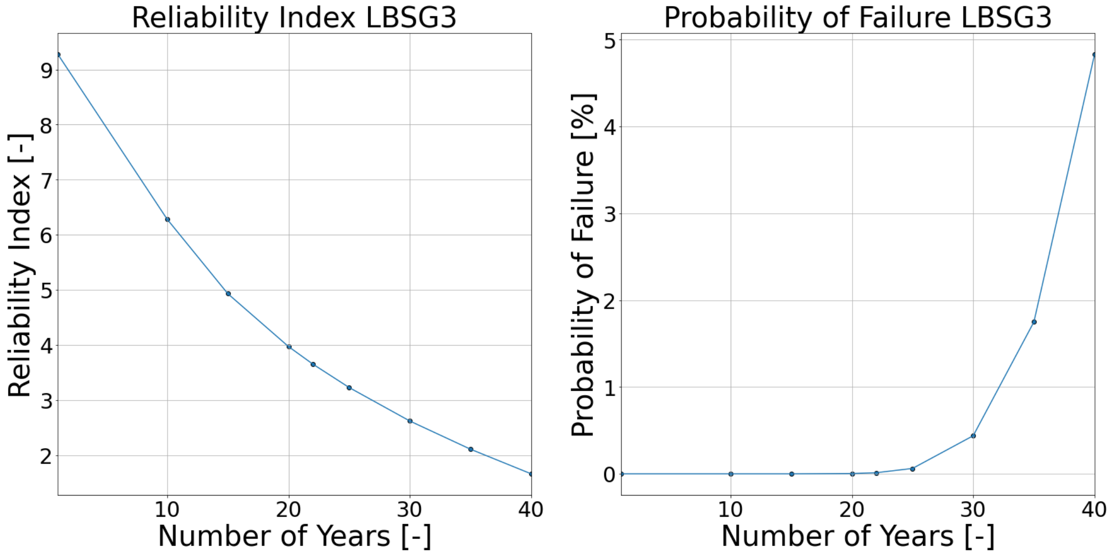


Figure 7.9: Reliability Index and Probability of Failure LBSG3

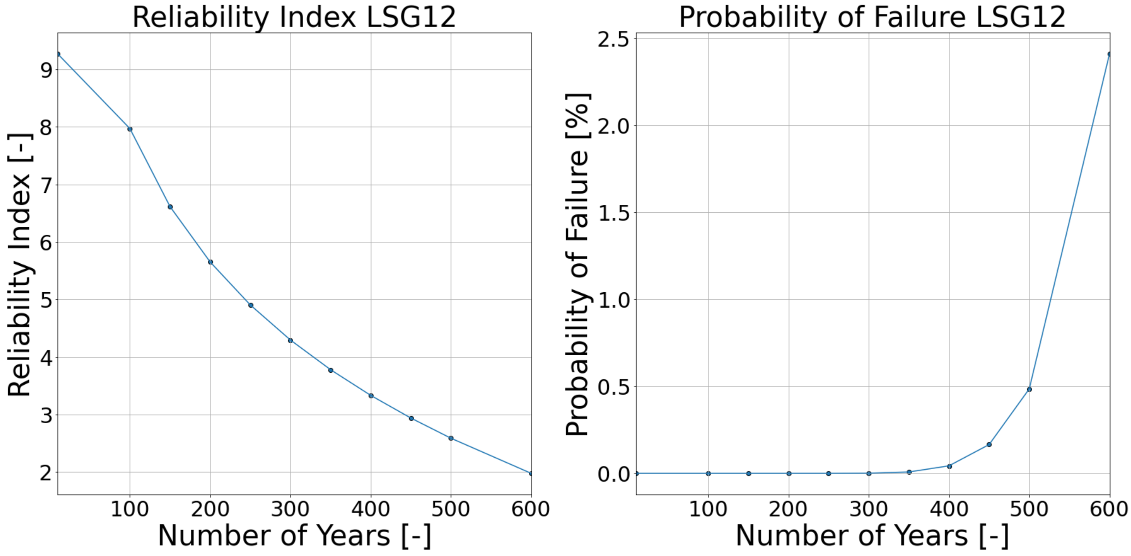


Figure 7.10: Reliability Index and Probability of Failure LSG12

Chapter 8

Conclusions and Future Research Recommendations

8.1. Conclusions

In order to assess to what extent the main research question:

> *Can the digital twin technology be enhanced with machine learning to accurately monitor the structural health of marine structures in terms of loading and fatigue damage accumulation?*

has been answered a sort summary is given on the conclusions with regard to the research sub-questions.

8.1.1. Feasibility of Development of a Digital Twin Model based on Machine Learning Algorithms

The model that has been proposed in the context of this graduation project is based on two machine learning models. The proposed model is able to learn and update through multiple sources as it is trained using data from multiple strain gauges and a weather buoy. Furthermore, it is able to accurately represent the past, present and future behaviour of its physical counterpart in terms of fatigue damage accumulation. Taking into account the definition of a digital twin model that has been given in Chapter 1 it can be concluded that the model proposed in the scope of this project has all the required attributes to qualify as a digital twin model. Under this perspective, the development of a digital twin model based on machine learning algorithms has been proven to be feasible.

The main advantage of the proposed digital twin system compared to a conventional hull monitoring system is that it can be used to calculate the fatigue damage accumulation of past and future states during which no strain measurements are available. This means that using this digital twin system someone can estimate the fatigue damage accumulation of a structure retrofitted with a hull monitoring system utilizing the collected operational data in an optimized manner. Furthermore, the digital twin system can be used to monitor the structure's fatigue damage accumulation in expected future sea states and take preventive action in case of a failure scenario.

8.1.2. Accuracy of Fatigue Damage Accumulation Prediction

As it has been shown in Chapter 5 the fatigue damage prediction accuracy achieved by the machine learning regression model is very high. In all the locations taken under consideration the achieved accuracy is about 100% with a standard deviation less than 4%. A quantitative comparison of this newly developed fatigue damage prediction framework with the corresponding framework already developed within the Monitas Group cannot be performed since the predictions for the vessel used as a case study are not available. However, taking into account the results published by Hageman et al. [5], a qualitative comparison can be performed. The results for a deck location and a side-shell location are illustrated in Figures 8.1 and 8.2 correspondingly. Before drawing any conclusions, it has to be noted the that these results are for a different unit a have a different time horizon than the results presented in Chapter 5 since the number of samples in the test set corresponds to a time horizon of about two months. Even with these significant differences taken into consideration, a qualitative comparison of the figures serves as a strong indication that a lot may be gained from the adoption of neural networks for the prediction of fatigue damage accumulation.

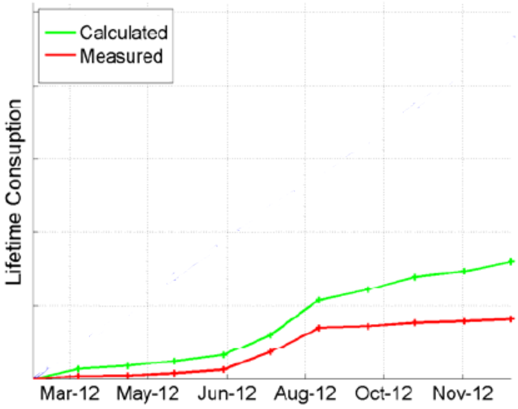


Figure 8.1: Prediction of fatigue damage accumulation for a deck location using the Monitas framework [5]

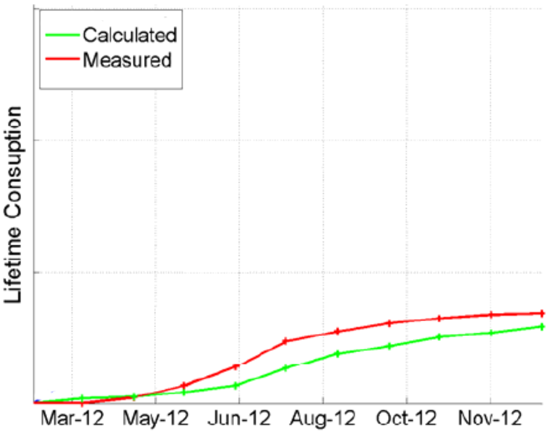


Figure 8.2: Prediction of fatigue damage accumulation for a side-shell location using the Monitas framework [5]

8.1.3. Integration of a Structural Reliability Analysis Module

Utilizing the fatigue damage accumulation prediction module and based on Miner's Rule assumption that the order of individual stress cycles does not affect fatigue accumulation the distribution of the design variable called "Typical Weekly Fatigue Damage" was obtained. Combining this design variable with a variable that describes the accuracy of the prediction module and Wirsching's model the limit state function of the structural details under study was formulated. Then using the Hansofer Lind reliability index coupled with a CPSO optimization algorithm a structural reliability analysis module was developed.

The developed structural reliability analysis module integrates the uncertainty of the prediction module as well as the uncertainty related to Miner's Rule. However, it does not take into consideration the uncertainty related to the fatigue resistance curve and to environmental effects. Integrating the uncertainty related to the fatigue resistance curve is expected to decrease the predicted probability of failure since a fatigue resistance curve with a 97.7% safety margin was used in the damage calculation. The integration of material degradation effects is expected to increase the probability of failure due to material degradation over the years. However, this increase is expected to be only marginal due to protection measures that restrict the material degradation.

This module can be used in order to generate a rational inspection, maintenance and repair schedule for the structure. Integrating more uncertainty sources in the module will lead to a schedule that contains smaller safety margins.

8.1.4. Utilization of Operational Data to Monitor Operational Conditions

In the context of this graduation project, a novel algorithm for load reconstruction using strain measurements has been proposed based on a reformulation of the static/quasi-static FEM problem. Utilizing this algorithm

one can gain insight into the operational conditions of the structure using the strain measurements of the hull monitoring system. The main benefit of the algorithm is that it can operate efficiently, as it has been shown by the sensitivity analysis, even for sparse measurements (in terms of sensor location). The main restrictions of the algorithm is that it is only accurate for static and quasi-static loading and that its accuracy is depended on the accuracy of the FEM model of the structure. The significance of this restriction largely depends on the operational conditions of the vessel. When slamming is not a common occurrence the significance of this restriction is limited.
The insights obtained through the use of this load-reconstruction algorithm can be utilized in order to improve the hydro-mechanical modeling of future designs.

8.2. Future Research Recommendations

In the scope of this study the feasibility of development of a digital twin for the structural health monitoring of marine structures has been positively evaluated. Taking into account the research that has been performed within this graduation project, the following recommendations are made in order to translate the proposed digital twin model concept into a fully operational high fidelity digital twin model.

8.2.1. On-line Training Scheme

In this feasibility study the machine learning models where trained in an off-line manner. This means that there was no live interaction between the AHMS, the weather buoy and the machine learning models. This "lag" in the acquisition of new data means that the digital twin is not operating at its maximum efficiency. This problem can be solved through the development of an on-line training scheme for the machine learning models. A framework should be developed within which the newly acquired data will be transferred to the machine learning models the moment they are acquired. In this way the digital twin model can always stay up-to-date.

8.2.2. Training on a Multi-Year Dataset

In the context of this study a year's worth of data was used to train the machine learning models. This means that any possible effect of the structure's degradation over the years on the rate of fatigue damage accumulation has not been taken into account. Moreover, changes in the local climate of the vessel's operational area may substantially affect the performance of the models. In order to take such effects into account the use of a multi-year dataset is required for the training of the machine learning algorithms. Moreover, in order for such effects to be taken into consideration new features may need to be engineered as well.

8.2.3. Consideration of Material and Fabrication Sources of Uncertainty

The developed structural reliability analysis framework is only considering uncertainty related to the models that are used. This means that any uncertainty related to the material and the fabrication process is disregarded. In order to improve the proposed framework these sources of uncertainty should be considered in future research.

8.2.4. Improved Advisory Hull Monitoring System

The Advisory Hull Monitoring System concept currently used in the scope of the Monitas JIP provides useful insight in the fatigue damage accumulation process. However, the sparse sensor placement does not allow the implementation of a dynamic load reconstruction module. As part of future research an improved version of the AHMS is proposed that has densely placed sensors in certain side shell areas near the waterline. In this way additional insights could be gained regarding the dynamic phenomena taking place by means of dynamic load reconstruction.

Bibliography

[1] P. Aalberts, A. Ibekwe, R. Hageman, G. Oguntola, J. Izuchukwu, K. Sorensen, V. Nadathur, and R. Vliet, "Advisory hull monitoring system for the bonga fpso," 01 2019.

[2] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, Dec 1943.

[3] A. Geron, *Hands-on machine learning with Scikit-Learn and TensorFlow*. Sebastopol, CA: O'Reilly Media, 2017.

[4] C. Elegbede, "Structural reliability assessment based on particle swarm optimization," *Structural Safety*, vol. 27, pp. 171–186, 04 2005.

[5] R. Hageman, P. Aalberts, M. Shaik, and H. van den Boom, "Development of an advisory hull fatigue monitoring system," *Transactions - Society of Naval Architects and Marine Engineers*, vol. 121, pp. 22–56, 01 2015.

[6] P. Aalberts, J. Cammen, and M. Kaminski, "The monitas system for the glas dowr fpso," 05 2010.

[7] D. L'Hostis, J. van der Cammen, R. Hageman, and P. Aalberts, "Overview of the monitas ii project," Jun 2013.

[8] M. Grieves and J. Vickers, *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*, pp. 85–113. 08 2017.

[9] CDBB, "The gemini principles," 2018.

[10] Q. Lu, A. K. Parlikad, P. Woodall, X. Xie, Z. Liang, E. Konstantinou, J. Heaton, and J. Schooling, "Developing a dynamic digital twin at building and city levels: A case study of the west cambridge campus," *Journal of Management in Engineering*, vol. 36, 10 2019.

[11] I. Thompson, *Digital Twinning of Ship Structural Fatigue:State of the art review and strategic research agenda*. Defence Research and Development Canada, July 2019.

[12] U. Tygesen, M. Jepsen, J. Vestermark, N. Dollerup, and A. Pedersen, "The true digital twin concept for fatigue re-assessment of marine structures," 06 2018.

[13] K. Tatsis, "Structural health assessment through vibration monitoring on fpsos," 2016.

[14] I. Boulkaibet, T. Marwala, L. Mthembu, M. Friswell, and S. Adhikari, "Sampling techniques in bayesian finite element model updating," *CoRR*, vol. abs/1110.3382, 10 2011.

[15] B. Sumer and J. Fredsoe, *Hydrodynamics Around Cylindrical Structures*, vol. 12. 01 2006.

[16] J. Boutrot, Y. Giorgiutti, F. Rezende, and S. Barras, "Reliable and accurate determination of life extension for offshore units," 05 2017.

[17] K. B. Ludvigsen, "Digital Twins for Blue Denmark," tech. rep., Det Norske Veritas(DNV), 2018.

[18] G. Storhaug, "Digital twins and sensor monitoring," *Maritime Impact*, 01 2019.

[19] T. Hulkkonen, T. Manderbacka, and K. Sugimoto, "Digital twin for monitoring remaining fatigue life of critical hull structures," 03 2019.

[20] J. Warner, M. Zubair, and D. Ranjan, "Near real time damage diagnosis using surrogate modeling and high performance computing," 01 2017.

[21] R. J. Barthorpe, "On model- and data-based approaches to structural health monitoring." 2010.

[22] D. Gamerman and H. Lopes, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition.* Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis, 2006.

[23] Z. Krobot, T. Túró, and V. Neumann, "Using vehicle data in virtual model for maintenance system support," in *2017 International Conference on Military Technologies (ICMT)*, pp. 171–174, 2017.

[24] C. Ye, L. Butler, B. Calka, M. Iangurazov, Q. Lu, A. Gregory, M. Girolami, and C. Middleton, "A digital twin of bridges for structural health monitoring," 09 2019.

[25] D.-H. Lau, N. Adams, M. Girolami, L. Butler, and M. Elshafie, "The role of statistics in data-centric engineering," *Statistics Probability Letters*, vol. 136, 02 2018.

[26] S. R. Yeratapally, P. E. Leser, J. D. Hochhalter, W. P. Leser, and T. J. Ruggles, "A digital twin feasibility study (part i): Non-deterministic predictions of fatigue life in aluminum alloy 7075-t651 using a microstructure-based multi-scale model," *Engineering Fracture Mechanics*, vol. 228, p. 106888, 2020.

[27] P. E. Leser, J. E. Warner, W. P. Leser, G. F. Bomarito, J. A. Newman, and J. D. Hochhalter, "A digital twin feasibility study (part ii): Non-deterministic predictions of fatigue life using in-situ diagnostics and prognostics," *Engineering Fracture Mechanics*, vol. 229, p. 106903, 2020.

[28] R. Bartels and R. Scott, "Computed and experimental flutter/lco onset for the boeing truss-braced wing wind tunnel model," 06 2014.

[29] D. Kim and E. Wong, "A simplified method to predict fatigue damage of ttr subjected to short-term viv using artificial neural network," *Advances in Engineering Software*, vol. 126, pp. 100–109, 11 2018.

[30] J. Morio and M. Balesdent, *Estimation of Rare Event Probabilities in Complex Aerospace and Other Systems.* 11 2015.

[31] R. Hejazi, A. Grime, M. Randolph, and M. Efthymiou, "An ann-based framework for rapid spectral fatigue analysis of steel catenary risers," 03 2018.

[32] DNV, "Dnvgl-rp-f204 riser fatigue recommended practise," 2010.

[33] R. Engebretsen, C. Nilsen-Aas, J. Muren, L. J. Hondebrink, and B. Kajolli, "Flexible Riser Fatigue Counter Powered by Digital Techniques," vol. Day 2 Tue, May 05, 2020 of *OTC Offshore Technology Conference*, 05 2020. D022S002R013.

[34] V. Chaves, L. Sagrilo, V. Silva, and M. Vignoles, "Artificial neural networks applied to flexible pipes fatigue calculations," p. V05BT04A022, 05 2015.

[35] ASTM, "Standard practices for cycle counting in fatigue analysis," 2017.

[36] A. Almar-Naess, *Fatigue Handbook: Offshore Steel Structures.* Tapir, 1985.

[37] A. C. de Pina, A. A. de Pina, C. H. Albrecht, B. S. Leite Pires de Lima, and B. P. Jacob, "Ann-based surrogate models for the analysis of mooring lines and risers," *Applied Ocean Research*, vol. 41, pp. 76–86, 2013.

[38] J. P. R. Cortina, F. J. M. de Sousa, and L. V. S. Sagrilo, "Neural networks applied to the wave-induced fatigue analysis of steel risers," *Mathematical Problems in Engineering*, vol. 2018, p. 2719682, Jul 2018.

[39] K. Müller, M. Dazer, and P. Cheng, "Damage assessment of floating offshore wind turbines using response surface modeling," *Energy Procedia*, vol. 137, pp. 119–133, 2017.

[40] C. Bak, F. Zahle, R. Bitsche, T. Kim, A. Yde, L. Henriksen, M. Hansen, J. Blasques, M. Gaunaa, and A. Natarajan, "The dtu 10-mw reference wind turbine," 2013. Danish Wind Power Research 2013 ; Conference date: 27-05-2013 Through 28-05-2013.

[41] F. Sande, W. Yu, D. Mattha, J. Azcona, M. X., E. Grela, S. Voutsinas, and A. Natarajan, "D4.33: Innovative concepts for floating structures," 2014.

[42] H.-j. Kim, B.-S. Jang, C.-K. Park, and Y. H. Bae, "Fatigue analysis of floating wind turbine support structure applying modified stress transfer function by artificial neural network," *Ocean Engineering*, vol. 149, pp. 113–126, 02 2018.

[43] S. G. Lee, Y. Yang, W. Chang, and J. Sohn, “Predicting the residual fatigue life of a cargo hull tank using a deep-learning technique,” *Journal of International Maritime Safety, Environmental Affairs, and Shipping*, vol. 4, pp. 84 – 92, 2020.

[44] N. Christiansen, P. Voie, O. Winther, and J. Høgsberg, “Comparison of neural network error measures for simulation of slender marine structures,” *Journal of Applied Mathematics*, vol. 2014, pp. 1–11, 03 2014.

[45] S. J. Hanson and D. J. Burr, “Minkowski-r back-propagation: Learning in connectionist models with non-euclidian error signals,” in *Proceedings of the 1987 International Conference on Neural Information Processing Systems*, NIPS’87, (Cambridge, MA, USA), p. 348–357, MIT Press, 1987.

[46] N. Christiansen, P. Voie, J. Høgsberg, and N. Sødahl, “Efficient mooring line fatigue analysis using a hybrid method time domain simulation scheme,” vol. 1, 06 2013.

[47] Q. Lu, J. Zhu, and W. Zhang, “Quantification of fatigue damage for structural details in slender coastal bridges using machine learning-based methods,” *Journal of Bridge Engineering*, vol. 25, p. 04020033, 07 2020.

[48] D. K. Gupta, “Inverse methods for load identification augmented by optimal sensor placement and model order reduction,” 2013.

[49] S. A. Masroor and L. Zachary, “Designing an all-purpose force transducer,” *Experimental Mechanics*, vol. 31, pp. 33–35, 1991.

[50] M. Wickham, D. Riley, and C. Nachtsheim, “Integrating optimal experimental design into the design of a multi-axis load transducer,” *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, vol. 117, pp. 400–405, January 1995.

[51] M. E. Johnson and C. J. Nachtsheim, “Some guidelines for constructing exact d-optimal designs on convex design spaces,” *Technometrics*, vol. 25, no. 3, pp. 271–277, 1983. Full publication date: Aug., 1983.

[52] F. Riccioli, “Development of a retrofit layer with an embedded array of piezoelectric sensors,” 2020.

[53] A. T. Patera, “A spectral element method for fluid dynamics: Laminar flow in a channel expansion,” *Journal of Computational Physics*, vol. 54, no. 3, pp. 468–488, 1984.

[54] S. V. Parter, “On the legendre–gauss–lobatto points and weights,” *Journal of Scientific Computing*, vol. 14, pp. 347–355, Dec 1999.

[55] X. Chang, Y. Yan, and Y. Wu, “Study on solving the ill-posed problem of force load reconstruction,” *Journal of Sound and Vibration*, vol. 440, pp. 186–201, 2019.

[56] B. Zhang, J. Zhang, Z. Wu, and S. Du, “A load reconstruction model for advanced grid-stiffened composite plates,” *Composite Structures*, vol. 82, no. 4, pp. 600–608, 2008.

[57] R. Hashemi and M. Kargarnovin, “Vibration base identification of impact force using genetic algorithm,” vol. 1, 01 2007.

[58] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press, 1975. second edition, 1992.

[59] M. Alioli, P. Masarati, M. Morandini, T. Carpenter, N. Osterberg, and R. Albertani, “Membrane shape and load reconstruction from measurements using inverse finite element analysis,” *AIAA Journal*, vol. 55, pp. 297–308, 2017.

[60] M. Alioli, P. Masarati, M. Morandini, T. Carpenter, and R. Albertani, “Nonlinear membrane inverse finite element model for pliant wings,” 01 2015.

[61] E. Bastidas-Arteaga and A.-H. Soubra, *Reliability Analysis Methods.* 01 2014.

[62] R. Melchers and A. Beck, *Structural Reliability Analysis and Prediction.* Wiley, 2018.

[63] O. Hughes, D. Béghin, and J. Paik, *Ship Structural Analysis and Design.* Knovel Library, Society of Naval Architects and Marine Engineers, 2010.

[64] J. Cardoso, J. Almeida, J. Dias, and P. Coelho, "Structural reliability analysis using monte carlo simulation and neural networks," *Advances in Engineering Software*, vol. 39, pp. 505–513, 06 2008.

[65] A. Dudzik and B. Potrzeszcz-Sut, "The structural reliability analysis using explicit neural state functions," *MATEC Web of Conferences*, vol. 262, p. 10002, 01 2019.

[66] Y. Bai and W.-L. Jin, "Chapter 31 - basics of structural reliability," in *Marine Structural Design (Second Edition)* (Y. Bai and W.-L. Jin, eds.), pp. 581–602, Oxford: Butterworth-Heinemann, second edition ed., 2016.

[67] C. Elegbede, "Structural reliability assessment based on particle swarm optimization," *Structural Safety*, vol. 27, pp. 171–186, 04 2005.

[68] O. Ditlevsen and H. O. Madsen, *Structural Reliability Methods.* 01 1996.

[69] R. Lebrun and A. Dutfoy, "A generalization of the nataf transformation to distribution with copula," *Probabilistic Engineering Mechanics*, vol. 24, pp. 172–178, 04 2009.

[70] J. Kennedy, *Particle Swarm Optimization*, pp. 760–766. Boston, MA: Springer US, 2010.

[71] H. Zhao, Z. Ru, X. Chang, and S. Li, "Reliability analysis using chaotic particle swarm optimization," *Quality and Reliability Engineering International*, vol. 31, 07 2014.

[72] B. Low and W. Tang, "Efficient spreadsheet algorithm for first-order reliability method," *Journal of Engineering Mechanics-asce - J ENG MECH-ASCE*, vol. 133, 12 2007.

[73] R. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 26, p. 457, 07 1976.

[74] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM JOURNAL OF RESEARCH AND DEVELOPMENT*, pp. 71–105, 1959.

[75] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, pp. 1527–54, 08 2006.

[76] F. Chollet, *Deep Learning with Python.* Manning, 2017.

[77] C. E. IBM, "What is machine learning?," July 2020.

[78] B. Jason, "A gentle introduction to the rectified linear unit," 2019.

[79] Rumelhart, D. E., J. Mcclelland, and J. L., *Parallel distributed processing: explorations in the microstructure of cognition. Volume 1. Foundations.* 01 1986.

[80] M. Awad and R. Khanna, *Support Vector Machines for Classification*, pp. 39–66. Berkeley, CA: Apress, 2015.

[81] V. N. Vapnik, *Statistical Learning Theory.* Wiley-Interscience, 1998.

[82] M. Awad and R. Khanna, *Support Vector Regression*, pp. 67–80. Berkeley, CA: Apress, 2015.

[83] B. Schölkopf and A. J. Smola, *Support Vector Machines*, pp. 187–188. 2001.

[84] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, pp. 1–37, Jan 2008.

[85] Scikit-Learn-Developers, "Ensemble methods," 2021.

[86] J. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, 11 2000.

[87] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 vol.1, 1995.

[88] L. Breiman, “Machine learning, volume 45, number 1 - springerlink,” *Machine Learning*, vol. 45, pp. 5–32, 10 2001.

[89] K. K. Pradhan and S. Chakraverty, “Chapter four - finite element method,” in *Computational Structural Mechanics* (K. K. Pradhan and S. Chakraverty, eds.), pp. 25 – 28, Academic Press, 2019.

[90] G. Liu and S. Quek, “Chapter 3 - fundamentals for finite element method,” in *The Finite Element Method (Second Edition)* (G. Liu and S. Quek, eds.), pp. 43 – 79, Oxford: Butterworth-Heinemann, second edition ed., 2014.

[91] E. Strømmen, *Structural Mechanics The Theory of Structural Mechanics for Civil, Structural and Mechanical Engineers: The Theory of Structural Mechanics for Civil, Structural and Mechanical Engineers*. 01 2020.

[92] B. Datta, *Numerical Linear Algebra and Applications, Second Edition*. Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2010.

[93] H. AM and N. Lind, “An exact and invariant first order reliability format,” *Journal of Engineering Mechanics*, vol. 100, 01 1974.

[94] O. Ditlevsen, *Uncertainty Modeling with Applications to Multidimensional Civil Engineering Systems*. Advanced book program, McGraw-Hill International Book Company, 1981.

[95] B. Low and W. Tang, “Efficient spreadsheet algorithm for first-order reliability method,” *Journal of Engineering Mechanics-asce - J ENG MECH-ASCE*, vol. 133, 12 2007.

[96] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948 vol.4, 1995.

[97] R. Statistics, “Basic concepts of correlation,” 2020.

[98] K. Park, *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer International Publishing, 2017.

[99] Keras-Developers, “Dropout layer,” 2021.

[100] Scikit-Developers, “Minmax scaler,” 2021.

[101] I. Jolliffe and J. Cadima, “Principal component analysis: A review and recent developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, p. 20150202, 04 2016.

[102] R. A. Horn and C. R. Johnson, *Canonical Forms for Similarity and Triangular Factorizations*, p. 163–224. Cambridge University Press, 2 ed., 2012.

[103] Keras-Developers, “Regression losses,” 2021.

[104] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.

[105] T. Dozat, “Incorporating nesterov momentum into adam,” 2016.

[106] Keras-Developers, “Optimizers,” 2021.

[107] Keras-Developers, “Adam,” 2021.

[108] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, pp. 123–140, Aug 1996.

[109] S. Developers, “Minimize function documentation,” 2021.

[110] I. Babuska, "The finite element method with penalty," *Mathematics of Computation*, vol. 27, pp. 221–228, 1973.

[111] A. DNVGL, "Dnvgl-rp-c206 fatigue methodology of offshore ships," 2007.

[112] M. Smith, *ABAQUS/Standard User's Manual, Version 6.9.* United States: Dassault Systèmes Simulia Corp, 2009.

[113] A. Saint-Venant, *Memoire sur la torsion des prismes: avec des considérations sur leur fléxion ainsi que sur l'équilibre intérieur des solides élastiques en général, et des formules pratiques pour le calcul de leur résistance à divers efforts s'exerçant simultanément.* S.l., 1853.

[114] A. Mansour, P. Wirsching, M. Luckett, and A. Plumpton, "Assessment of reliability of existing ship structures," 1997.

[115] P. H. Wirsching, "Fatigue reliability for offshore structures," *Journal of Structural Engineering*, vol. 110, no. 10, pp. 2340–2356, 1984.

[116] Fugro, "Seawatch metaocean buoys," 2021.

[117] A. DNVGL, "Dnvgl-cg-0130 wave loads," 2018.

Chapter A

Data Acquisition

As it has already been mentioned the proposed digital twin framework will be evaluated using data collected form an FPU in the scope of the Monitas JIP. The data is collected through the sensors of an AHMS that is installed on-board the vessel and a weather buoy that is located in the area of operation of the vessel. In this appendix the location of the sensors whose data will be utilized in the scope of this research as well as the data processing and storage procedure. It is important to be noted that none of the work presented in this chapter has been performed in the context of this graduation project. It is merely presented to improve the understanding of the case study.

A.1. Location of the Sensors

Five sensors are utilized to capture the response of the vessel in the scope of this study. More specifically two Long Base Strain Gauges(LBSG) located on the deck of the vessel and three conventional strain gauges(LSG) located on the side-shell of the vessel. All the five sensors are located on the same frame of the vessel. The exact position of the LBSGs can be seen in Figure A.1 while the location of the LSGs in Figure A.2. The sensors will be referred to as:

- LBSG3
- LBSG4
- LSG10
- LSG11
- LSG12

The strains recorded by the sensors are stored in h5 files each one of which contains a half hour time-series.Since the sensors are located at non-critical points, the response captured by them has been translated to the response of the closest fatigue critical points, which in this occasion are welded joints, with the use of stress concentration factors.

A.2. Weather Buoy

The wave energy spectrum related to each sea-state is captured using Fugro's SEAWATCH Wavescan Buoy [116]. The buoy is able to capture the wave period with the accuracy of 2% and the major heading of the waves with an accuracy of 0.3°. As it can be seen in Figure A.3 the buoy has a discus shape. Because of its shape only the major heading of the waves is captured. The energy is then spread in all the directions using a direction function [117]:

$$D(\beta) = \frac{\Gamma(1+\frac{n}{2})}{\sqrt{\pi}\Gamma(\frac{1}{2}+\frac{n}{2})} * cos^n(\beta - \beta_p) \tag{A.1}$$

Where β is the relative spreading around the main direction and β_p is the main direction. The parameter n is set to be equal to 8 since the sea state in the operating area is swell driven and for swells n>7 is proposed [117]. The resulting wave energy spectrum is stored in h5 files, each one of them corresponding to a half hour sea state.

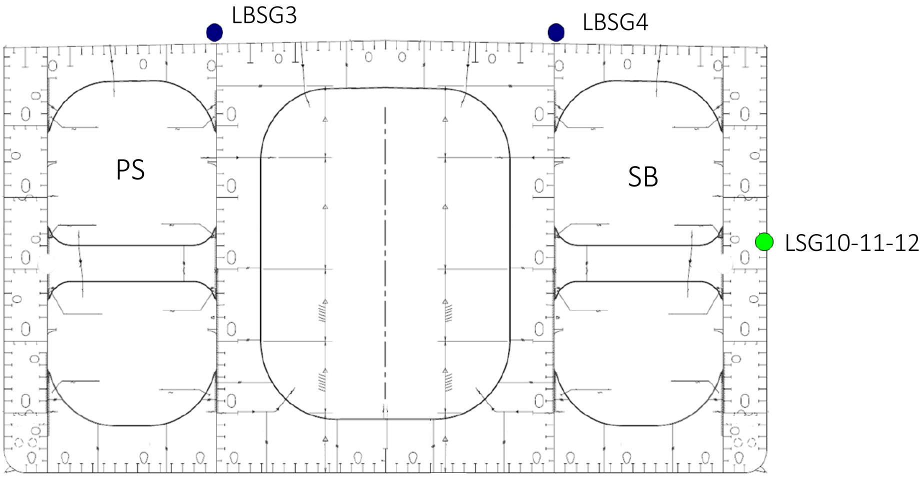


Figure A.1: Location of the LBSGs

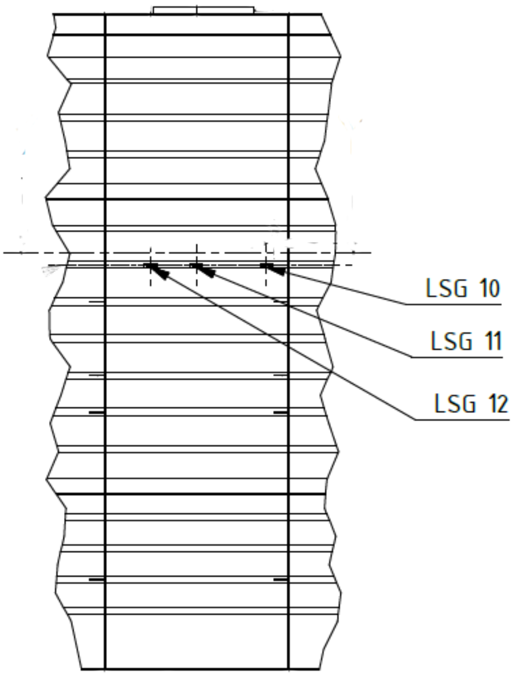


Figure A.2: Location of the LSGs

Figure A.3: SEAWATCH Wavescan Buoy