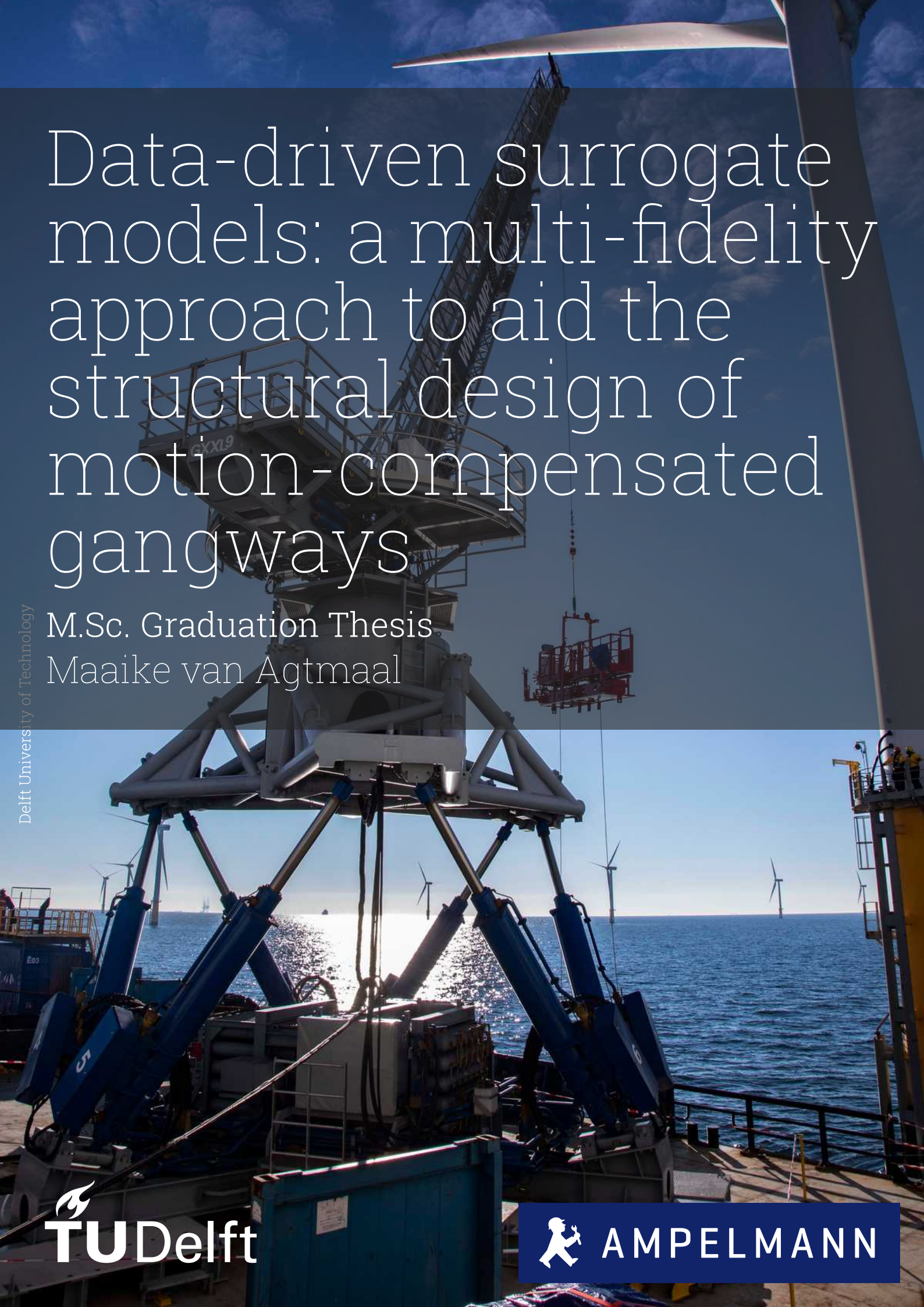
Data-driven surrogate models: a multi-fidelity approach to aid the structural design of motion-compensated gangways

M.Sc. Graduation Thesis

Maaike van Agtmaal

Delft University of Technology



TU Delft

AMPELMANN

Thesis for the degree of MSc in Marine Technology in the specialization of Ship Design

Data-driven surrogate models: a multi-fidelity approach to aid the structural design of motion-compensated gangways

by

Maaike van Agtmaal

Performed at

Ampelmann Operations B.V.

This thesis (**MT.24/25.046M**) is classified as confidential in accordance with the general conditions for projects performed by the TUDelft.

To be defended publicly on July 18th 2025

Company supervisors

Responsible supervisor: Ir. V. Sfetsios

Thesis exam committee

Student number:	4819438	
Project duration:	September, 2024 – July, 2025	
Thesis committee:	Dr. A. Coraddu,	TU Delft, supervisor
	Dr. ir. P. de Vos,	TU Delft
	Ir. V. Sfetsios	Ampelmann Operations B.V.

Author details

Author:	Maaike van Agtmaal
Student number:	4819438

Cover: Motion-compensated gangway (Ampelmann GXXL). This image belongs to Ampelmann Operations B.V.

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

In this document, a Master's thesis for the track Ship Design of the master Marine Technology is reported. The title of the research states: Data-driven surrogate models: a multi-fidelity approach to aid the structural design of motion-compensated gangways. This research has been performed at Delft University of Technology in collaboration with Ampelmann Operations B.V., which provided the case study. This report describes the path followed in the research of using data-driven surrogate models for the structural design of complex offshore mechanical systems, such as a walk-to-work system as designed by Ampelmann.

I would like to express gratitude to my supervisor, Dr. Andrea Coraddu, for his efforts and interest in this work, and his academic guidance towards worthy and valuable research. I would also like to thank Ir. Vasileios Sfetsios for being my day-to-day supervisor from Ampelmann, and for his honesty regarding the feasibility of my ambitions for this work. Moreover, I would like to thank Dr. Elsy Saloumi for providing very useful insights from her experience as a structural concept engineer. These insights frequently put my research in a different and more useful perspective, enabling me to steer the project in the right direction. Furthermore, I would like to thank Ir. Rene Hoogendoorn for insights from his experience and for providing the opportunity to perform this research in collaboration with Ampelmann, as he came up with the application and started the project. My gratitude also goes towards other colleagues at Ampelmann, who have been of great help in understanding the case study. Finally, I would like to thank my friends and family for their continuous support and encouragement throughout this journey.

Maaike van Agtmaal
Delft, July 2025

Summary

This thesis investigates the application of data-driven surrogate models within a multi-fidelity framework to aid and accelerate the structural design of motion-compensated offshore gangways. The motivation was found in the literature review, where gaps were found regarding suitability for higher-dimensional problems and multi-type input, sampling techniques, suitability for data-scarcity and suitability for application on complex mechanical systems.

The literature review starts by establishing the foundations of surrogate modelling, including its evolution from response surface methods and Kriging, to more advanced data-driven techniques such as support vector regression and neural networks. A distinction is made between physics-based, data-driven, and hybrid models, highlighting trade-offs in terms of complexity, interpretability, and data requirements. Challenges such as generalizability, extrapolation, and the curse of dimensionality are addressed, alongside the importance of design of experiments (DoE) in constructing effective models. Classical and modern DoE methods are discussed, including Latin hypercube sampling and space-filling strategies, as well as adaptive sampling techniques. Recent trends in the field include hybrid models that integrate physical laws into machine learning algorithms, and multi-fidelity frameworks that fuse low- and high-fidelity data to reduce computational cost. These advances are framed as promising solutions to limitations in data efficiency and model applicability to real-world engineering problems.

The case study centers on an offshore gangway structure designed by Ampelmann Operations B.V., a company specializing in motion-compensated systems for transferring personnel and cargo to offshore platforms. The gangway operates under diverse loading conditions such as live loads, environmental forces, and cargo transfers, resulting in thousands of unique load combinations. These are categorized according to certification standards and reduced to 4016 discrete cases for model input. Structural data from an existing, fully designed system forms the basis for both model training and evaluation. A detailed description of the gangway, including its main components and loading scenarios, is provided to contextualize the design space.

The methodology involves several key stages. First, the data of the case study is described in detail. This involves defining a numerical representation of structural severity to act as the model output, condensing global responses such as stress, deflection, and buckling into a single ranking metric for each load case. A low-fidelity analytical model is implemented in Python to provide fast approximations of structural response, extended to include stress and buckling calculations. A high-fidelity finite element model is provided by the partner company, which is simplified to ensure numerical stability. As an alternative, a simple truss model is used for better suitability for beam model analysis in FEM. As a preliminary sampling, random sampling is proposed for the design of experiment. Then, an intelligent sampling technique based on Euclidean distances is proposed to improve data efficiency by selecting high-fidelity samples in underrepresented regions. After data sampling, a machine learning framework is created to support data-driven surrogate modelling. Three model types are implemented: Ridge regression, KRR and XGBoost. These were implemented after pre-processing the data, and a random grid search is implemented for hyperparameter tuning, combined with cross-validation. The selected performance metrics are the REP, MAPE, PPMCC and the R-squared score. A multi-fidelity framework is introduced to combine predictions from low- and high-fidelity sources using the comprehensive approach.

The results show that modelling issues in the high-fidelity model lead to data that is not physically plausible. Among the surrogate types tested, Ridge regression is inadequate for complex, mixed-type input features. KRR performs moderately, and XGBoost demonstrates strong performance particularly in scenarios with limited high-fidelity data. The intelligent sampling strategy does not improve the performance compared to random sampling. The multi-fidelity model yields marginal improvements for KRR on the complex model, but it does not achieve better performance then a single-fidelity XGBoost model. The simple truss model showed an improvement of XGBoost performance when data is scarce,

which confirms the hypothesis. As the simple truss model can be extended to cover the full range of load cases, this could justify the time required for implementing the multi-fidelity framework. However, in the current state the benefit of multi-fidelity is marginal. Regarding the complex dataset with modelling issues in the source, it is still useful that this research found a well-predicting surrogate in XGBoost, even though multi-fidelity might not be the answer for that dataset due to badly correlated data. The predicting surrogate, proven to have well-predictive capabilities even though the data represents modelling issues, can point engineers to all load cases that result in exceeding members. Therefore, the surrogate could aid engineers by acting as a pointer to load cases that require engineering judgment. Also, the predictive capabilities of the surrogate enable surrogate-based optimization. Both findings can increase efficiency of the structural design of the gangway.

The study concludes that surrogate modelling, when carefully implemented, can improve the efficiency of structural design evaluations for offshore gangways. XGBoost appeared to be best suited for this case involving complex, moderate-dimensional input spaces. Multi-fidelity modelling is viable but sensitive to data quality. Future work should focus on extending the framework to surrogate-based optimization and exploring adaptability to different geometries.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
LF	Low-fidelity
HF	High-fidelity
MF	Multi-fidelity
ML	Machine learning
DoE	Design of experiment
RSM	Response surface method
OVAT	One variable at a time
CDoE	Classical design of experiments
MDoE	Modern design of experiments
DACE	Design and analysis of computer experiments
SRS	Simple random sampling
LHS	Latin hypercube sampling
FEA	Finite-element analysis
CFD	Computational fluid dynamics
MARS	Multivariate adaptive regression splines
RBF	Radial basis function
MLS	Moving least squares
SVR	Support vector regression
SVM	Support vector machines
NN	Neural network
RF	Random forests
GTB	Gradient tree boosting
XGBoost	Extreme gradient tree boosting
LightGBM	Light gradient boosting machine
CatBoost	Categorical boosting
AdaBoost	Adaptive boosting
HPC	High performance computing
GNN	Graph neural network
SHM	Structural health monitoring
FEMU	Finite element model updating
FEM	Finite element model
SAO	Surrogate-assisted optimization
PIML	Physics-informed machine learning
LSTM	Long short-term memory
PINN	Physics-informed neural network
PKeDA	Prior-knowledge-embedded data-driven approach
ANN	Artificial neural network
ROM	Reduced-order modelling
DOF	Degrees of freedom
T-Boom	Telescoping boom
RoS	Ranking of severity
RAO	Response amplitude operator
GPR	Gaussian process regression
KRR	Kernel Ridge regression
MAPE	Mean absolute error percentage

Abbreviation	Definition
REP	Relative error percentage
PPMCC	Pearson product-moment correlation coefficient
MAE	Mean absolute error
(R)MSE	(Root) Mean squared error
CV	Cross-validation

Symbols

Symbol	Definition	Unit
E	Young's modulus	[GPa]
L_{beam}	Length single beam	[mm]
L	Gangway length	[mm]
L_{mb}	Length main boom	[mm]
L_{tb}	Length T-boom	[mm]
I	Area moment of inertia	[mm^4]
I_{mb}	Inertia main boom	[mm^4]
I_{tb}	Inertia T-boom	[mm^4]
A	Cross-sectional area	[mm^2]
A_{mb}	Cross-sectional area main boom	[mm^2]
A_{tb}	Cross-sectional area T-boom	[mm^2]
M	Bending moment	[kNm]
M_{tipp}	Maximum tipping moment of the gangway	[kNm]
F_{eq}	Equivalent force	[kN]
q_{eq}	Equivalent distributed load	[kN/m]
$\hat{y}$	Predicted target variable	[-]
K	Effective length factor	[-]
P_{cr}	Critical buckling load	[kN]
ρ	Scaling factor	[-]
σ	Stress	[MPa]
σ_y	Yield strength	[MPa]
σ_{max}	Maximum stress in gangway	[MPa]
σ_{cr}	Critical buckling strength	[MPa]
ν	Deflection	[mm]
μ	Input-dependent scaling factor	[-]

1 Introduction

Interest in data-driven models has grown exponentially in the past years, mainly due to better availability of resources to store data. The research and innovation opportunities with this new data-driven mindset have been explored extensively in the past decade, and the full potential has not nearly been reached. Literature on the subject is vast and more papers are being published every year. Moreover, interest reaches beyond the scientific community. Since data can be used to optimize complex workflows in engineering practices, several industries have started to show interest in machine-learning developments as well.

This research is performed in collaboration with Ampelmann Operations B.V., which is an example of an industry with an interest in machine-learning developments. It was suspected that data-driven models could aid and accelerate the design process of their gangway structure as described in Chapter 3. Offshore mechanical systems are subject to a large variety of loads, which are often able to assume different directions and magnitudes. Designing a complex structure that can survive all possible load combinations is an inefficient process due to the large amount of cases that have to be analyzed with a detailed and slow finite-element analysis. By predicting which loading scenarios are most governing for the structure, time would be saved by refraining from running all load cases through computationally expensive structural calculation software. On top of that, a successful surrogate could link input describing loading scenarios to a certain ranking of the severity for the structural integrity, which makes the framework suited for optimization. This could further increase the design efficiency.

This case study is a valuable addition to the academic developments in the world of data-driven surrogate modeling. Chapter 2 gives a detailed review of the current literature in this field and shows that the field of surrogate-based design optimization is largely in development. Gaps can be found concerning the surrogate model itself, but also concerning the method for selecting appropriate input variables. These areas are of great importance for successful, effective and useful implementation of a surrogate modelling framework. In short, the main challenges are centered around model flexibility and data efficiency, both in a quantitative and computational manner. Surrogate models should provide predictions as accurately as possible, with the least possible amount of data in the shortest time possible. This extends the limitations to scalability, extrapolation capability and generalizability as well, as retraining the model every time for slightly different cases is not considered time-efficient. Also, the time required to implement the model should not approach the time savings achieved with the model, so model simplicity should also be considered. Furthermore, in order to use the model for real-world engineering applications, it should provide physically plausible solutions with preferably known uncertainties, such that it can be used for design optimization.

It has been proven that a multi-fidelity approach can be beneficial when dealing with computationally expensive data, as described in Section 2.2.3. Since there is a low-fidelity model available, and it is less time-consuming to create low-fidelity data than high-fidelity data, it is proposed to implement a multi-fidelity framework. With that, the proposed research will be led by the research question below.

How can data-driven surrogate modelling techniques, embedded in a multi-fidelity framework, aid and accelerate the structural design process when applied to a complex mechanical system with complex moderate-dimensional input?

The main research question is broad and will be approached by tackling the following sub-questions:

1. Which machine learning technique is best suited for the complex nature of design variables?
2. Which surrogate modelling technique can provide accurate predictions when data is scarce?
3. How can intelligent sampling improve the model performance when data is scarce?
4. How can a multi-fidelity framework effectively leverage low-fidelity data to multi-fidelity output, such that a higher accuracy is reached with less data?

In order to answer these questions, three different machine-learning techniques will be trained on the complex data. Then, a subset of the data is used for training to evaluate performance when data is scarce. These will show which model is best-suited for the case in this research. Then, it is evaluated whether this performance improves when using an intelligent sampling technique. Finally, it is evaluated whether the multi-fidelity approach improves the required percentage of sampled data in high-fidelity for accurate predictions.

The remainder of this report is structured as follows. Chapter 2 gives the detailed literature review that led to the found gaps described above, providing the academic relevance to this research. Then, the case study is described in detail in Chapter 3. Chapter 4 follows with the methodology to tackle the research questions. It elaborates on the data generation, machine-learning framework, multi-fidelity framework and the proposed intelligent sampling technique. This chapter is followed by the results of the methodology in Chapter 5, which also discusses the findings in detail. Finally, the conclusion is stated in Chapter 6, along with recommendations for future work.

2

Literature review

This chapter contains the literature review performed at the start of this research to identify research gaps and to set up a plan of approach towards academically relevant research. Section 2.1 gives the historical background on surrogate modelling by elaborating on data sampling techniques, physics-based surrogates, data-driven surrogates and the limitations. Section 2.2 follows with the State of the Art, presenting techniques to overcome the common limitations of surrogate models. Solutions are presented regarding data sampling techniques, hybrid modelling and multi-fidelity modelling.

2.1. Background

This research contains a case study of a complex mechanical system for offshore operation, for which a model will be used to predict the structural response of the mechanical system. The structural response is usually calculated with a computationally expensive finite-element simulation, but the goal is to provide a quicker, more intelligent prediction of the structural response, such that the algorithm can learn the relations between design variables and structural integrity. With this insight, the design process of the structure could become easier and more efficient. These types of models are commonly referred to as *surrogate models*.

The concept of surrogate modeling arose when scientists needed quicker solutions when traditional computer simulations became too time-consuming. It originates from classical Design of Experiment theory, where a polynomial response surface method (RSM) was used to describe the relation between the input variables and the output [50, 70]. With the newly found relation, the response could be interpolated and predicted for all parameters inside the design space. With the advances in computer modeling, the idea of treating computer simulations as experiments came up, and statistical methods such as Kriging were developed to approximate expensive computer simulations with a model [71].

Therefore, a surrogate model can be described as a *model of a model* and is sometimes also referred to as a metamodel [52, 71, 47, 14]. Queipo et. al [59] describe surrogate modeling: "Surrogate modeling can be seen as a non-linear inverse problem for which one aims to determine a continuous function (f) of a set of design variables from a limited amount of available data ($\mathbf{f}$)". In other words, it is a technique to find relations between input and output variables, which allows for quick predictions replacing complex and expensive simulations. The concept is widely applied to engineering problems, and it keeps developing at a fast pace. Especially with the increasing attention for and knowledge of machine learning techniques, scientist are rapidly exploring new possibilities of surrogate modeling.

There are three parts in surrogate model construction. First the initial sampling data must be extracted from the design space. Then a surrogate modeling technique must be chosen to correctly represent the mathematical model, and afterwards the model must be validated with a fitting model. Negrin et al. [52] state that an offline structural design optimization project consists of the following steps:

1. Problem formulation
2. Select initial sampling points

3. Creation of a surrogate model
4. Optimization process using the surrogate model

Step one is described in the previous section. More information on steps two and three can be found in the remainder of this section. The aim is that the end result of this work can be extended to surrogate-based optimization. However, the actual optimization, step 4, is considered out of scope, and so is adapting the designs based on the optimization results, which would be the next step in a surrogate-based optimization project.

2.1.1. Design of Experiments

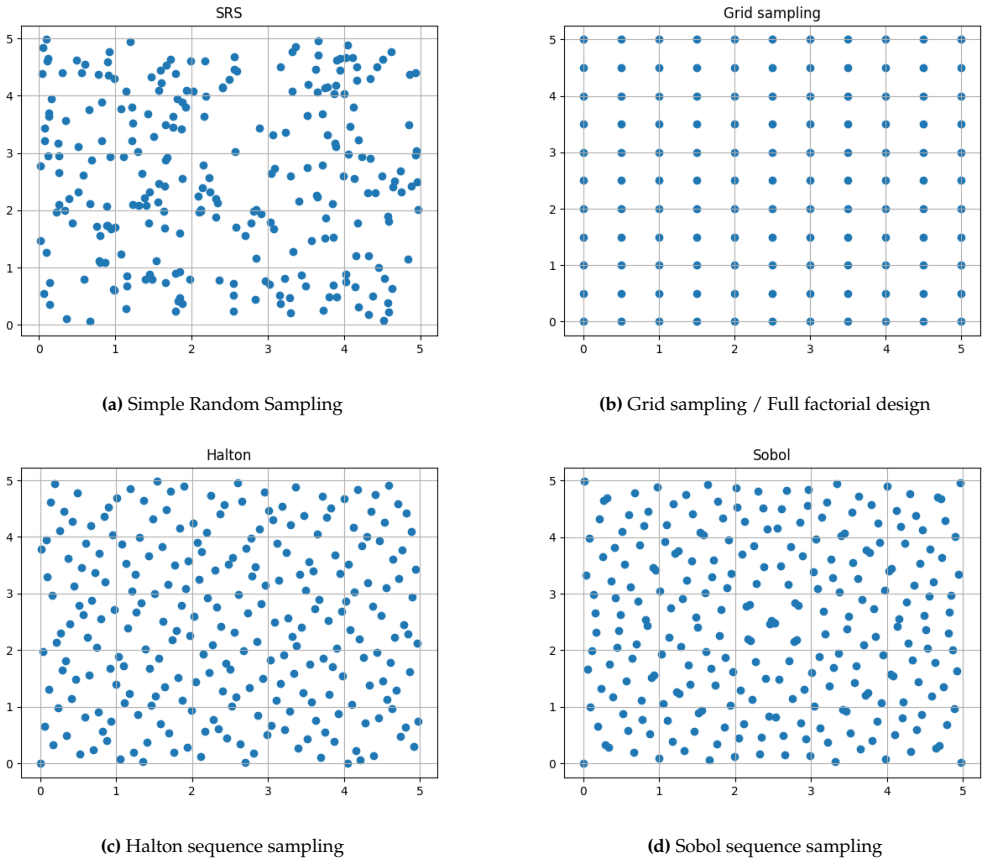
Sampling the data is an important step in the process of surrogate modelling and can have a great influence on the model performance. The sampled data is usually referred to as Design of Experiment. In [50], an *experiment* is defined as follows: a test or series of runs in which purposeful changes are made to the input variables of a process or system so that we may observe and identify the reasons for changes that may be observed in the output response. The design of such an experiment represents the method of sampling those input variables. Therefore, the term *Design of Experiment (DoE)* is widely used to describe the sampling method for the input variables of a model. This section describes common methods for data sampling.

Traditionally, researchers used to vary one variable at a time (OVAT), while keeping all other variables the same. Another method was based on trial-and-error. However, as experiments became more advanced and the amount of input dimensions became larger and larger, this method turned out to be quite inefficient, and sometimes it led to large unrepresented areas in the parameter space. Additionally, relationships between variables were not captured well. The DoE literature [50] elaborates on multiple sampling methods to better sample the parameter space. These sampling methods can be divided into two categories: *Classical Design of Experiments* (CDoE) and *Modern Design of Experiments* (MDoE). The classical methods include the Full/Fractional factorial design, Central-Composite design (Box-Wilson design), the Box-Behnken design, optimal designs and Orthogonal array design (Taguchi Methods). This category was intended for physical experiments, where an inevitable random measurement error plays an important role. Due to this assumption, the spread of these samples is mainly focused on the boundaries of the parameters, with limited focus on intermediate values. Methods that are space-filling, such as the full factorial design, usually becomes too expensive, since the number of samples grows exponentially with the number of dimensions.

This is where MDoE comes into place. The theory of MDoE, which is sometimes also referred to as Design and Analysis of Computer Experiments (DACE), applies the DoE theory to computer experiments. They attempt to balance space-filling properties with a low number of samples, to represent the design space as efficient and accurate as possible. Rather than focusing on the boundaries, these methods provide a better spread of the samples to make sure that the entire parameter space is represented in the DoE. They are developed for deterministic computer experiments, where the same input will lead to the same output if repeated. This category is quite extensive and can further be organized as follows.

Firstly, there are *random designs*. These include Simple Random Sampling and Monte Carlo methods. These DoE risk clustering and could lead to unrepresented areas as dimensions get higher.

An improvement to these designs are *quasi-random designs*. Sampling methods of this category attempt to keep randomness while minimizing the discrepancy of the DoE. Common methods in this category include Quasi-Monte Carlo designs, Sobol sequence, Faure sequence, Hammersley and the Halton sequence and Niederreiter sequence. Figure 2.1 shows the differences of these sampling methods compared to random sampling and full-factorial design.



(a) Simple Random Sampling

(b) Grid sampling / Full factorial design

(c) Halton sequence sampling

(d) Sobol sequence sampling

Figure 2.1: Visualization of four different sampling methods

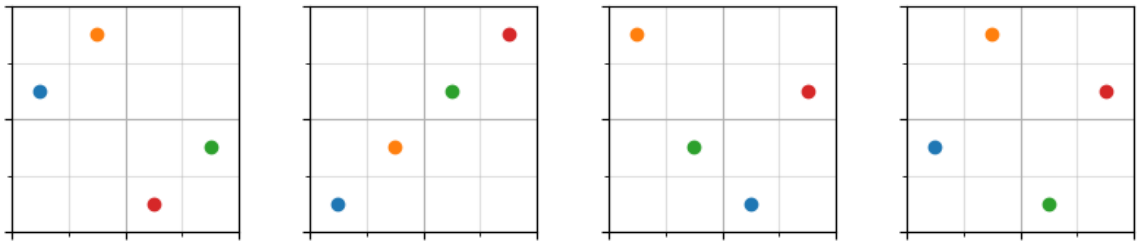
Then, there are *projections-based designs*, which focus on the space-filling property of the sample set. Many new improved implementations of orthogonality in designs due to their suitability for regression problems and space-filling property.

Also, there is the category of *uniform designs*. These are mathematical designs aiming to sample uniformly across the design space, either based on statistical measures such as discrepancy and entropy, or based on Euclidian distance with the maximin or minimax distance criteria.

There are DoE that are not part of the previous categories. *Latin Hypercube Sampling* (LHS) is the most important one. It is widely used for surrogate-based optimization and was initially developed for uncertainty quantification. The main idea behind LHS is that samples are randomly distributed with the requirement that all columns and rows are represented with an equal amount of samples (see Figure 2.2a). However, After initial development, it was found that sometimes this method does not offer full coverage of the design space, or draws samples in undesired patterns such as in 2.2b.

Since most DoE have drawback for high-dimensional parameter spaces, efforts have been made to improve the existing designs. Most of these improvements are on LHS, since it provides flexibility from its random characteristic, while incorporating good coverage. Therefore, multiple versions of LHS were created by combining them with other criteria such as orthogonal arrays, maximin, minimax, symmetry or Centroidal Voronoi Tessellations. Examples of orthogonal-based LHS and symmetric LHS can be found in Figures 2.2c and 2.2d respectively.

More information on these methods can be found in [78, 71, 50], which all propose LHS as the conventional method.



(a) Default LHS **(b)** Problem with LHS **(c)** Orthogonal-Array based LHS **(d)** Symmetric LHS

Figure 2.2: Visualization of four different Latin Hypercube sampling methods

2.1.2. Surrogate modelling

Choosing a suitable model is quite a challenging task. Like George E. P. Box famously stated: *"All models are wrong, but some are useful"* [8]. In other words, one cannot expect that the model will be 100% correct, because all models have their own assumptions and simplifications that make them deviate from real-world responses. However, if those assumptions are good enough and the simplifications are well-understood, models can become a useful tool for engineers.

Within the field of surrogate modeling, there are three different categories of models that can act as a surrogate: pure physics-based models, pure data-driven models and hybrid models. Each of these model types has their own advantages and limitations. Hybrid models are a current trend and will be discussed later in this chapter.

Physics-based models

When surrogate models follow fundamental engineering principles, they fall under the category *physics-based* models. These models approximate real-world phenomena by applying mathematical functions to the situation, which usually contain simplifying assumptions. Poor assumptions lead to poor representation of reality. These types of models are widely used as surrogate models to approximate structural response and have been validated with experimental data.

In the field of structural engineering, the most important physics-based model is Finite Element Analysis (FEA), which is a numerical method that allows engineers to evaluate the local structural response in a detailed manner. This is achieved by splitting the structure into a finite number of small elements. Then, for each of those individual elements, the structural properties shear, strain and more are calculated by implementing fundamental equations of structural analysis. Although there are still assumptions present in the model, it is a widely used model for estimating structural response. When implemented correctly, the small elements can capture all relevant local details of the structure.

FEA is considered physics-based because it is based on fundamental engineering principles of structural analysis. For steal beams, the calculation is a discretized version of the Euler-Bernoulli beam theory, also referred to as *classical beam theory*. This method is widely used to describe global beam bending phenomena. Resulting deflection, stress, shear force and moments have standardized equations for different beam configurations, based on fundamental principles such as Hooke's law, a fundamental principle that relates stress to strain by adding Young's modulus E in the equation as described in [31]. The beam equations are simplified linear elastic theory and assume infinitely rigid cross-sections. Also, the cross-sections remain planar after deformation and their axis will always remain normal to the axis of the beam with curvature. Due to the linear elastic assumption, it only holds for small deformations with stresses below the yield strength.

Since the computational burden is quite high for FEA, it will most likely not be used as a surrogate. Rather, surrogate models are developed to replace the FEA ([38, 4, 32, 58]).

Although FEA models are most relevant for this paper, the field of fluid mechanics has similar problems that can give valuable insights as well. A popular modeling method for fluids is *Computational Fluid Dynamics* (CFD), and some fundamental physical principles are the Navier-Stokes equation and the Bernoulli principle. These are widely used aerodynamics for modeling air, as well as in the marine, civil

and offshore industry for modeling water. Surrogate modeling has been applied to these fields as well [3, 43].

Data-driven models

When fundamental principles of engineering fail to describe relations between input and response variables or when these relations are simply not well-understood, data-driven models be a suitable solution. Rather than applying physical laws to the data, the models search for the underlying relation by learning from a large amount of data. A function is fitted to the data and by minimizing the discrepancy, a model is constructed that can predict the response variable. The remainder of this section describes the most frequently used model types, for which more information can be found in [20], [78].

Classic polynomial regression, also referred to as the *Response Surface Method (RSM)* is the traditional method of surrogate modeling and still widely used due to its simplicity [20]. It is extensively explored in the past. The main essence is that the method attempts to fit any order - predominantly first or second - polynomials to the data, while optimizing the error with the least squares method. Then, the influences of design variables in the parameter space can be identified and used for optimization [61]. However, this approach is not suited for highly non-linear problems, high-dimensional or multimodal data. It may also require too much data or become unstable if the problem is complex [20], [78] and [61].

Kriging is a popular parametric modeling method, but quite complex to implement. Queipo et. al described it as follows: "The Kriging method in its basic formulation estimates the value of a function (response) at some unsampled location as the sum of two components: the linear model (e.g., polynomial trend) and a systematic departure representing low (large scale) and high frequency (small scale) variation components, respectively" [59]. It assumes that the departure components are correlated, and the method is very flexible in capturing these approximations. However, the model interpolates the sample data and is therefore very sensitive to noise, and due to its complexity it can be quite time-consuming to construct the model, especially in high-dimensional spaces [35].

An alternative method for Kriging was developed called Multivariate Adaptive Regression Splines (MARS) [21]. It is non-parametric, so there are no assumptions about the underlying function. It uses the method of *regression splines*, where the parameter space is cut into smaller pieces which are fitted individually with basis functions, and connected continuously at the knots. MARS selects these knots adaptively with a forward pass and backward pruning. It is flexible, it can capture non-linear relations and is significantly less time-consuming to implement than Kriging, but is very sensitive to the sample size. When data is scarce, it might be better to select a different method [35]. MARS has also shown overfitting issues and cannot handle gaps in datasets very well [78].

Radial basis function (RBF) is a popular replacement for RSM [20]. Rather than assuming some underlying polynomial function, it fits the data by applying linear combinations of Euclidean distances to points in the parameter space. It was developed for multivariate scattered data interpolation and proved to be better suited for high-dimensional problems with a non-linear nature. Although it is widely acknowledged that there is no single model that works best for every situation - *no-free-lunch theorem* [76] - RBF proved to perform best in the comparative study from Jin et. al [35]. The comparison included the traditional methods RSM, RBF, Kriging and MARS and the performance evaluation was based on accuracy and robustness.

Moving least squares (MLS) is a non-parametric form of polynomial regression for surfaces with the capability of capturing non-linear responses, introduced by Lancaster and Salkauskas [40]. Rather than fitting the entire dataset in a global manner, it fits polynomial functions locally between intermediate data points. It is a widely applied method for estimating response surfaces and is well-known for its smooth predictions. This smoothness makes the model suited for noisy data, but may also lead to inaccuracies when the true response function is too abrupt in its changes. Also, it can be too computationally demanding in high-dimensional spaces and can suffer numerical instabilities.

Support vector regression (SVR) is another non-parametric regression technique and originates from the widely used classification theory of *Support Vector Machines* (SVM). The main concept is characterized by the introduction of a tolerance. Small deviations from the true value will not be penalized. This makes the model robust and less prone to overfitting [68]. SVR works especially well for prediction in

high-dimensional noisy data. However, Forrester and Keane [20] also state that SVR is not suited for use inside an optimization loop due to the large training time of the model.

Another popular non-parametric method for modeling highly-nonlinear patterns in high-dimensional spaces is the theory of neural networks (NN). It is a very wide concept with numerous different types. A summary of the fundamentals can be found in [66]. It was initially designed by biologists to simulate the working of the brain when computer technology was still underdeveloped, but it gained a significant amount of attention in the nineties. Then, the computational force was strong enough to prove its potential. Its potential was also found in the field of structural optimization [54]. These types of surrogate models are well-known for their accuracy and ability to handle complex high-dimensonal data. However, a major drawback of neural networks is the required sample size for a well-trained model. Also, these models are not interpretable and the results are not necessarily physically plausible.

The final frequently used form of data-driven models are tree-based ensemble methods. The simplest form of tree-based models is the single Decision Tree. This interpretable model starts at the root node, and with conditional statements, leaf nodes can be reached that provide the prediction of the tree. It can be used for both classification and regression problems. The simplicity of the model implies good interpretability. However, the models are very prone to overfitting and are not robust. Ensemble-learning methods have emerged to mitigate these issues. In these methods, an ensemble of decision trees is implemented, which improves the predictive performance. Popular examples are Random Forests (RF) and Gradient Tree Boosting (GTB). The latter was introduced by Friedmann [22] and comes in multiple forms: XGBoost, LightGBM, CatBoost and AdaBoost are most frequently used, but XGBoost is often considered the best performing tree-based model when balancing speed and accuracy [12, 30]. A comparative review of gradient boosting trees can be found in [7]. Despite the great performance of tree-based models, it should be noted that this type of models have issues with extrapolation.

2.1.3. Limitations

The traditional methods for surrogate modelling have their limitations, both on concerning the sampling method and the model type itself.

The method of data sampling can heavily influence the accuracy of the trained model [25], so it is crucial to select the appropriate sampling method. However, this is not a straightforward process. The suitability of a sampling method depends on the type and dimension of input parameters, and the choice of surrogate model also has an influence on which DoE will perform the best. [61] confirmed the difficulties of choosing an appropriate DoE and proposed further research on a reliable method of generating a validated database of input and output parameters that are both easy to generate and suitable for developing surrogate models. Viana [70] states that traditional DoE is especially limited when the design space has more than 10 dimensions, due to the curse of dimensionality. Besides the limitations of dimensionality, it should also be noted that not all types of data sampling are suitable for both discrete and continuous parameters.

Each traditional model type has their own benefits and drawbacks as well. For example, physics-based models have the drawback that their assumptions can be quite rigorous, leading to inaccuracies, and more accurate approaches are usually quite time-consuming. But in [79], it was found that physics-based models perform better than data-driven models when data becomes scarce. Data-scarcity is the most important drawback for pure data-driven models when they need to be applied to real-world engineering problems, because usually there is not that much qualitative data available.

In [2], a critical literature review is presented on the computational cost of surrogate models in comparison to model accuracy and problem size. They provide a framework for selecting an efficient surrogate modeling process based on the model requirements. The suitability of the models is highly dependent on the amount of input parameters and the desired speed of the simulation. Therefore, six classes were defined to separate the problem types. The separation criteria were speed of simulation (below or above 5 hours) and amount of parameters (less than 10, more than 100 or in between). For each of these classes, it is relatively easy to choose a model if either accuracy, problem size or computational cost are not prioritized. However, if all three characteristics are evenly important, it can be difficult to find a suitable model. Following this review, it appears that Kriging might be a suitable choice for the

case in this project, as the number of variables is between 10 and 100, assuming that model accuracy is the most important model characteristic for this project. With this amount of parameters, RSM loses computational efficiency and SVM might not be as accurate. It must be noted that this review does not take other modeling techniques into account such as neural networks or random forests.

Most surrogate models require the same amount of input variables for every run, which makes the model quite stiff. In [75], this issue has been addressed. They used graph-based models in combination with transfer learning to predict displacement fields given a structure's geometry, supports, and loads as inputs. The forces were acting on the geometry directly without the need for parametric design features. These models perform well when data is scarce and are more flexible to small input changes. But once again, the study only includes analysis on neural networks.

In [61], it is stated that the focus in the field of surrogate modelling for civil structures is on Kriging, Gaussian process, and Neural Networks. They propose to focus on uncertainty quantification in further research, which can be achieved by applying Bayesian regression. This surrogate bases both the model and the parameters on a statistical distribution, providing information on the uncertainty. Also, they addressed that surrogate modeling in structural engineering is challenged by the need for High Performance Computing (HPC). In [73], a review paper in the category risk analysis for civil structures, the need for more interpretability in the models is also stated as a significant research gap.

Generalizability is also an issue for most surrogate models applied to structural engineering. This limitation is confirmed in [30], which extends to extrapolation capabilities. This directly follows from generalizability issues. Generalizability is described as the performance to unseen data, and extrapolation further refines this definition with performance to unseen data outside the design space. Due to the similarity in definition, these terms are often used interchangeably. In [30], the model can only work for one geometry and has to be retrained when the geometry changes, even with minimal structural changes such as a small increase of length.The lack of extrapolation capabilities is a major drawback of most surrogate models. Attempts are made to address the lack of generalizability for data-driven surrogate models. Nourbakhsh et. al [53] attempted to tackle this issue by proposing a surrogate model that could generalize on different versions of a 3D truss structure based on a parametrization of the geometry. This study shows in a coherent step-by-step manner how the model was set up, and why certain choices where made regarding data sampling, surrogate model choice and the fitting method. Since their model would be trained on thousands of data points from output of a FEA software, they only included neural networks in their study. Other models would fail to handle such large output spaces. However, this study had its limitations. Only one load case was evaluated, and some parameters such as wind directionality were ignored to prevent issues with the *curse of dimensionality*. Also, this study is limited to problems with high data availability. For problems with scarce data, generalizability remains an open issue. These limitations extend to extrapolation capabilities. Li et. al [42] found that Graph Neural Networks (GNN) also provide good extrapolation properties, but this method requires a huge amount of data to function properly.

With the limitations described above, the applicability to real-world engineering problems remains a challenge. For complex mechanical engineering problems in the industry, the limitations in this section are the following. Most techniques are either 1) only suited for low-dimensional problems, 2) cannot handle multiple types of input variables, 3) cannot be used for multiple geometries due to extrapolation issues or 4) require a huge dataset to be effective. Other issues of existing models are lack of 5) interpretability, 6) physical plausibility and 7) suitability for multi-response output. Improvements often resulted in complex models that either 8) have not been applied to complex mechanical systems yet or 9) are so time-consuming to implement that the benefit of using the surrogate model is minimal. Following the issue that the model should be beneficial, it should also 10) be possible to extend the model to a surrogate-based optimization framework.

2.2. State of the art

Attempts are made to overcome the typical limitations of surrogate models, as described in the previous section. There is an enormous amount of applications for surrogate modelling in structural engineering, with many review papers of the current state-of-the-art. For this research, data-driven techniques combined with structural engineering are the most relevant. Negrin et. al [52] present practical

recommendations for the use of surrogate modelling techniques in structural engineering problems by elaborating the state-of-the-art methods. They found that the literature on this subject are mainly presenting novel meta-modelling techniques, tested with simple structures, and they recommend testing the use of these surrogates on real-life structural optimization problems with more complex structures. In [48], a state-of-the-art review can be found on the current use of machine learning in structural engineering. Multiple areas were found in which machine learning models are used, such as tools for engineers to explore complex design spaces, identify patterns, and predict solutions beyond human capability. These goals can be realized by correctly implementing data-driven surrogate models. Regarding surrogate modelling techniques, a helpful paper by Kudela [38] stated that in general, there are four main categories in which surrogate modelling is used in structural engineering: prediction, uncertainty quantification, sensitivity analysis and Surrogate-Assisted Optimization (SAO).

Then, there were helpful papers tackling issues in case studies. Especially papers in offshore engineering [14, 32] and steel truss structures [6, 30, 53], [55, 75] are very relevant. However, also applications in aerospace engineering [46, 78, 3, 41], [4] or civil engineering [61, 73, 26, 12] give relevant insights in the possibilities of surrogate modelling techniques. For the field of civil engineering, the applications are mainly in the fields of Structural Health Monitoring (SHM), reliability assessment, Finite Element Model Updating (FEMU), and disaster risk analysis. Regarding the field of Finite Element Analysis (FEA), [38] presented an overview of advances of surrogate models for FEA models. They divide the usage of surrogates for FEM in the following categories: prediction, sensitivity analysis, uncertainty quantification and Surrogate-Assisted Optimization (SAO). These papers pointed out some of the current limitations and solutions, which are elaborated in this section.

The previous section described limitations of traditional surrogate modelling approaches, both regarding the method of data sampling and the model type that is fitted to the data. This section discusses the current trends to overcome these limitations.

2.2.1. Data sampling

The previously discussed sampling methods are so-called 'one-shot' sampling methods, where the entire parameter space is defined directly, but these do not perform well for higher dimensions. As a proposed solution, Viana et. al [70] recommend sequential sampling methods in combination with dimensionality and search space reduction for parameter spaces with a dimension of 10 or higher. Sequential sampling is also referred to as Bayesian optimization [74] or adaptive sampling [23] in other literature. Viana et. al [70] elaborate on sequential sampling. In these methods, an active learning function is used to select new sample points, which will be sequentially added to the design space [15]. Figure 2.3 visualizes adaptive sampling, where one can see that the new sampling points are based on the underlying response function to pinpoint the local optima. This can be useful if computations are expensive, because this method provides information on the interesting areas (areas with large gradients) of the design space when data is scarce.

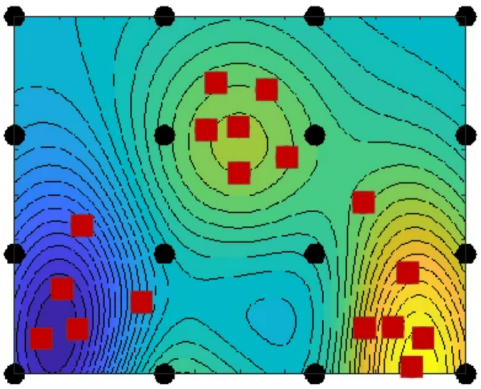


Figure 2.3: Adaptive sampling design (initial samples in black dots, sequentially added samples in red squares)[23]

However, implementing a well-performing Bayesian optimization framework remains a challenge for high-dimensional spaces, as stated in [74], where Wang et. al provide a state-of-the-art review in

Bayesian optimization techniques. Most techniques are based on a Gaussian process, but there have been some studies on other methods, such as random forests [33] and ensemble methods [27]. Wang et. al [74] divide the world of Bayesian optimization in nine categories, two of which are high-dimensional optimization and combinatorial optimization, where the parameters types can differ (e.g. discrete or continuous variables). Since these two categories are challenging by themselves, the combination high-dimensionality and multiple variable types makes the Bayesian optimization implementation increasingly complex. Tackling these issues remains an open challenge in the literature.

Although LHS is stated as the most conventional data sampling method in the field of surrogate modelling, Kamath [36] found that it might be better to use different methods in some cases. It can be computationally quite expensive to generate samples due to the optimization step, and they state that random sampling might be better suited when a balance between data exploration and exploitation is required. The review elaborates on intelligent sampling techniques.

2.2.2. Hybrid models

One current trend to overcome the common issues of data-driven surrogates is the development of hybrid models, sometimes in combination with adaptive sampling methods. Previous sections described models solely based on physics or pure data-driven. However, one can also combine these categories in a single surrogate model to construct a hybrid surrogate. Hybrid models, while combining best of both worlds, are usually quite complex and might be difficult to implement. They usually lack proof that they can work for complex structures as well. However, they are a promising solution to overcome the typical drawbacks of pure physics-based or pure data-driven models.

An important field of hybrid modelling is called Physics-Informed Machine Learning (PIML), which is a recent trend in surrogate modelling that gains a lot of attention. [65] shows the state-of-the-art from 2021 of prediction techniques in the building engineering field. They addressed that machine learning for prediction faces challenges when data is scarce, when a sampling method needs to be selected and when interpretability of the model is required. For data scarcity, they proposed the inclusion of physics in the model as a possible solution (PIML). Transfer learning could be another solution. The sampling method is mentioned due to its effect on the quality of the data and therefore on the quality of the trained surrogate model. They address the need for a systematic approach of data sampling. And finally, they pointed out the need for interpretability of the surrogate models. There are multiple proposed solutions to tackle this problem, such as a sensitivity analysis to understand the marginal effect of each feature on the response variable. Including physics into the loss function could be another method to improve model explainability and interpretability.

For example, [67] proposed a physics-informed version of the recurrent neural network type Long Short-Term Memory (LSTM). The aim of the model is to improve predictions on the structural response of buildings subject to wind loading. They combined traditional wind equations with data from sensors on the structure to obtain a physics-informed surrogate model. Their solution mainly proved to be beneficial for computational efficiency and predictive value beyond the limits of traditional wind estimating methods. Only a few sensors were needed to acquire sufficient data, and therefore the physics-informed surrogate proved to be efficient and accurate while there is a relatively low amount of data available. Another example can be found in [29], where Hao et. al compared physics-informed versions of SVR, RF and XGBoost surrogates to improve fatigue-life prediction of a notch in complex structures of aerospace engineering subject to complex loading conditions. They enriched their analysis by a sensitivity analysis and uncertainty quantification and proved that embedding physics into the surrogate can lead to more accurate, data-efficient, fast and well-generalizing models for fatigue prediction, and some are suited for uncertainty quantification as well. Also for structures under different loading scenarios, PIML has been explored for neural networks (PINN). In [77], Xu et. al enrich a neural network with a physics-informed loss function to an inverse engineering problem: load prediction based on displacement field data from FEA. They based the model on transfer learning to make it scalable to multiple geometries, and compared it to the self-adaptive PINN as introduced in [49]. The transfer learning-based approach resulted in faster convergence while remaining evenly accurate. These papers are just examples, but numerous papers have been appearing in the past couple of years on the subject of PIML. Zhu et. al [82] elaborate on the state-of-the-art in PIML applied to structural integrity predictions.

Including physics into the model leads to better interpretability, and less data is required for achieving

the same accuracy. It can also enhance model adaptability to cases with slightly different design parameters. Transfer learning is another method for improving adaptability, which makes the model suited for slightly different use cases and enhances flexibility. Another common method to improve data efficiency is the implementation of adaptive sampling, which lowers the required sample size for accurate results. Although numerous implementations exist that combine existing models with physics, transfer learning and/or adaptive sampling, there is a lack of research in practical applications of those models. Also, most of these improvements are centered around neural networks and lack representation for other black-box models.

Other forms of hybrid modelling exist as well. The main concept is that the advantages of multiple surrogate types are combined and used into a single surrogate model. Chen et. al [12] proposed a new method to improve extrapolation performance and developed a Prior-Knowledge-embedded Data-driven Approach (PKeDA) embedded in an ANN. They compared their PKeDA-ANN approach with XGBoost, since that model has proved to perform best on similar case studies, and the extrapolation properties significantly improved. However, only a proof of concept has been provided, and there is a need for a study on its suitability for practical applications. Numerous other hybrid models have been deployed [27][44][70]. However, it should be noted that hybrid approaches increase model complexity, making them more difficult to implement.

2.2.3. Multi-fidelity modelling

As stated in the previous section, one of the issues of most surrogates is the data availability. Sometimes there is not enough computational budget to acquire the sample size that is required for the model to function well. This issue arises when data is generated with complex computations such as FEA. Multi-fidelity modelling can be a suited solution to overcome this issue.

This concept started when computational models became more and more complex due to advanced in computer technology, but the computational burden became too large for efficient use of the models [9, 1, 37]. The main concept is described by using models of lower complexity - *low-fidelity* (LF) models - to explore the entire design space, and using the model of higher complexity and accuracy, the *high-fidelity* (HF) model, to refine the interesting areas of the design space. The resulting surrogate is then referred to as a *multi-fidelity* (MF) model. The concept assumes that the LF model can capture the overall trend of the HF data [61].

MF models are characterized by their method of fusing the multilevel data to one desired output space. The common approach is based on scaling functions, which can be divided into three categories [56]:

1. Multiplicative scaling
2. Additive scaling
3. Comprehensive scaling

Originally, multiplicative scaling was introduced by Haftka [28] for mapping LF to MF data. Mathematically, it is formulated as follows:

$$\hat{y}_{MF}(\mathbf{x}) = \mu(\mathbf{x})\hat{y}_{LF}(\mathbf{x}) \tag{2.1}$$

Here $\hat{y}_{MF}$ and $\hat{y}_{MF}$ represent the predicted response of the multi-fidelity and low-fidelity surrogates, respectively. However, this method had issues with zero divisions. This was solved by the introduction of additive scaling, where the discrepancy δ is added to the LF response:

$$\hat{y}_{MF}(\mathbf{x}) = \hat{y}_{LF}(\mathbf{x}) + \hat{\delta}(\mathbf{x}) \tag{2.2}$$

Combining both worlds gives a comprehensive scaling function, and proved to be a great improvement on the performance [19]. Then, the LF response is also multiplied with a constant scale factor ρ:

$$\hat{y}_{MF}(\mathbf{x}) = \rho\hat{y}_{LF}(\mathbf{x}) + \hat{\delta}(\mathbf{x}) \tag{2.3}$$

Most current multi-fidelity methods use the comprehensive approach. The traditional approach for modelling two different levels of fidelity is called *Co-Kriging* [37]. This extended version of the Kriging method can effectively reduce computational time while remaining accurate, given that the LF data and HF data are strongly correlated and hierarchical. It assumes Gaussian distributed data and correlations

and can be quite time-consuming to implement. In recent years, other data-driven surrogates have been implemented in a multi-fidelity framework. Examples are combinations with linear regression [81], ML [72], SVR [63], RBF [18, 10, 64], ensemble methods [80] and neural networks [45].

It should be noted that the use of data-driven techniques inside the multi-fidelity framework transfers the limitations of that data-driven technique as well. The combination with linear regression will struggle with non-linearities, when implementing SVR the surrogate will be quite time-consuming, combining with RBF will show poor scalability, etc. Besides, most models will struggle with extrapolation. And, especially if it is a black-box model, there is a need for interpretability and physical plausibility.

Multi-fidelity approaches have shown to be able to accurately predict responses, but it should be noted that implementing a multi-fidelity framework also takes time. It should always be evaluated whether the method will indeed provide time savings for the project [19]. A recent study on a new multi-fidelity framework also pointed out this issue [39]. Although their method performed best, the use of optimization and validation techniques used for hyperparameter tuning was quite time-consuming and raised the need for faster multi-fidelity approaches while maintaining accuracy.

Finally, it should be noted that this is not the only existing method for reducing the computational burden. *Reduced-Order modelling* (ROM) is another approach, which simplifies the simulation by simulating with fewer dimensions. More information on this approach can be found in [62] and [78]. It was decided not to include this theory in this review because it is required that all dimensions are included in the model, and there are low-fidelity models already available.

To conclude, this research proposes to combine challenging gaps into one project by applying state-of-the art surrogate modelling techniques to a real-world complex mechanical problem as described in Chapter 3. Data-driven models will be researched that can provide accurate, physically plausible predictions with the complexity of real-world applications when data is scarce. A successful approach that generalizes well would be applicable to more fields of engineering as well. This benefits the industry, as there is a need for proven technologies of surrogate modelling for complex cases, rather than just new methodologies that work for simpler cases. This would benefit the entire offshore industry due to the complexity of environmental loads, as well as civil structures such as bridges. It would be especially beneficial in the field of aerospace engineering, which is well-known for its complex cases.

3

Case Study

This research was performed in collaboration with Ampelmann Operations B.V., which provided a case study for this research. This chapter describes this case study. Section 3.1 gives the motivation for this case study, with a problem statement regarding the structural design of their systems. Section 3.2 describes the system of the case in detail, followed by the description of the possible loads acting on the structure in Section 3.3. This chapter finishes with a description of the model input for this research, the load combinations, in Section 3.4

3.1. Background and motivation

The partner company, Ampelmann Operations B.V., specializes in providing safe solutions for offshore transfer operations. These operations can revolve around the transfer of either crew or cargo to an offshore platform, or both, by means of motion-compensated gangways, sometimes also referred to as Walk-to-Work systems. These systems function as a bridge for crew to walk to the offshore platform, and can sometimes be used as a crane as well.

Since the beginning of Ampelmann, multiple types of motion-compensated gangways have been designed. They started out with the A-type, which could compensate for vessel motions in all six degrees of freedom (DOF) by using the Hexapod concept (6DOF systems). After that, more gangway types were developed. Some types used a different motion compensating system, when compensating heave, roll and pitch was deemed to be sufficient (3DOF systems). The types also differ in transfer type: some were just for people transfer, others just for cargo, and some could be used for both operation types. Apart from that, the types can be differentiated in the range of the system and capacity of the cranes. More information on the company and their systems can be found on their website [5].

Offshore gangways are subject to multiple load scenarios during operation. They might be connected to an offshore platform while keeping the tip in position, or they might be swinging a cargo mass above sea towards the aiming point. Apart from these operational differences, there are also environmental loads that can vary substantially, such as wind and waves. Including variations in direction of environmental forces, system position, emergency cases, there can be up to thousands of load cases, depending on the system's complexity. The structure cannot fail during any of these load scenarios, so the design has to be based on all of them. However, some are more governing than others. But due to the complex nature of the system, it can be difficult to identify the governing loads cases. Some load case parameters are discrete, others are continuous, and the structural response is described by multiple factors. Therefore, the project has a complex mixed-type moderate-dimensional design space, making it difficult to model. Currently, the industry uses their expertise and engineering judgment to reduce this number of load cases, but it remains a time-consuming process. If the governing load cases had been known beforehand, the design could be based on fewer load cases, requiring less time for expensive computations such as a Finite Element Analysis (FEA). This project proposes to provide a solution for this inefficient design process by applying a data-driven surrogate model to the structural design of an Ampelmann gangway.

3.2. System Description

The aim of this research is to accelerate the design phase of an offshore gangway. The company provided the structural properties of a system that has already been fully designed. Although the company would benefit if this research ends with a predictive model for gangways without this detailed knowledge available, a trained model on a fully designed system could function as a comparative source. By usiing an existing design, it is easier to setup and train the model. It is expected that the predictions for this specific gangway type can be extracted to the structural design of different gangway types. This section describes the W-type and its operational profile in detail.

From this point in this research, the terms *gangway* and *system* will refer to the gangway/system of the W-type motion compensated walk-to-work system. This system consists of three large parts: the tower, the cursor and the gangway. The general arrangement is visualized in Figure 3.1. The gangway can be adjusted in height by moving the cursor vertically along the tower. Once set in height, people can access the gangway by using the lift inside the tower. Then, using a slewing ring, luffing cylinders and a telescoping system, the system can compensate the ship's motions such that the tip of the gangway can be hold in place in the global axis system.

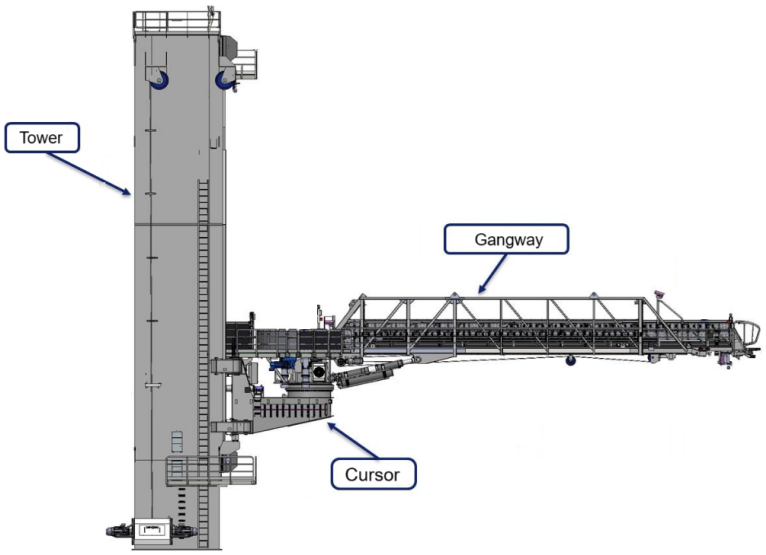


Figure 3.1: General sketch of the Ampelmann W-type

This research will focus on the structural design of the gangway, which consists of three main parts: the main boom, the telescoping boom (T-boom) and the tip. People access the gangway from the lift in the tower over the balcony to the transfer deck on the cursor. It can luff upwards and downwards with luffing cylinders and slew sideways with the slewing ring. Telescoping wheels and cables allow the T-boom to slide forwards and backwards, changing the length of the total gangway. A detailed visualization can be found in Figure 3.2.

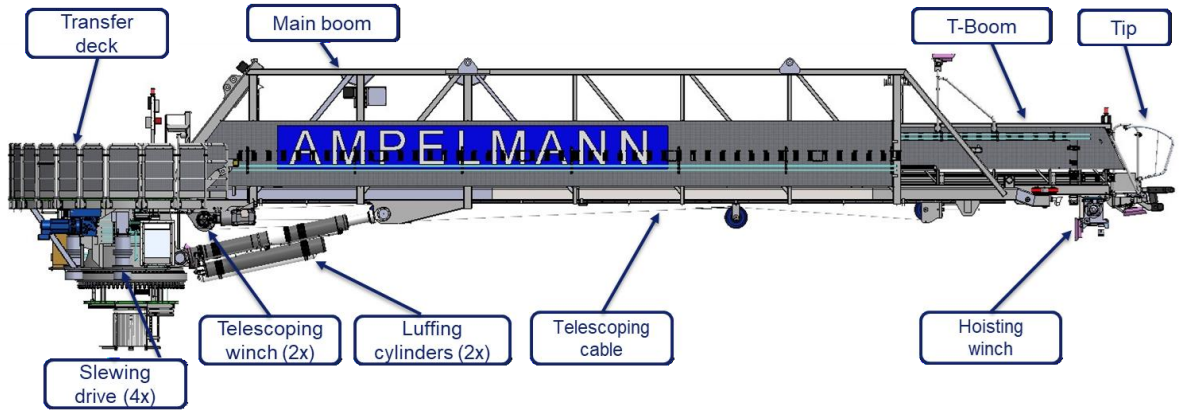


Figure 3.2: Gangway components

There are three different operation types. The first type is people transfer. During this operation, the gangway is maneuvered towards the offshore platform and aimed at the connection point. The tip is then connected such that the crew can safely transfer over the gangway to the offshore platform. The system is also able to safely transfer trolleys over the gangway.

The second type is cargo transfer. With the hook and winch attached to the end of the T-boom, the system is enabled to hoist cargo load. Therefore, the system is also suited for safe transport of cargo from ship to offshore platform, where the system operates as a motion-compensated crane.

And lastly, there is the stowed condition, during which the system is resting on deck. This is an important part of the operational profile as it occurs during transfer to the location of the project as well as waiting for suited weather for the operation. The structure should be able to withstand storms in this condition.

3.3. Loading types

Along the operating profile as described above, the system is subject to many different loading scenarios. These scenarios all follow from different combinations of environmental forces, operation type and system orientation. It is key that the structural integrity of the system is never compromised during either of the combinations. This section describes the parameters of these loading scenarios.

Generally, the loads acting on an offshore gangway can be described with seven categories [16]:

- Forces due to ship inclination
- Dead loads
- Environmental effects
- Dynamic forces due to crane movements
- Loads on access ways, platforms
- Live loads
- Load swing forces

Ship inclination affects the orientation of the local ship-bound axis system with respect to the global earth-bound axis system. Therefore, the loads due to gravity are affected for different configurations of vessel heel and trim. However, for this research it was decided to exclude heel and trim from the analysis. It was expected that the effect will be captured by varying the vessel accelerations.

Dead loads are defined by the geometry of the system and has a large effect of the loading profile. However, since this research does not extend to multiple geometries, these loads will not be varied.

However, the gangway has a range of luffing angles, slewing angles and telescoping strokes in which it can be positioned. Therefore the combination of masses and centers of gravity can change per scenario.

The environment cannot be excluded from the loading scenarios. Wind and waves have a large contribution to the loading profile. For wind, the speed will be varied and the angle is always set at the worst case scenario: perpendicular to the gangway. The effect of waves is transferred into vessel accelerations, which were derived from RAO analysis by Ampelmann.

The offshore gangway can luff up or down, slew sideways or telescope the T-boom forwards or backwards. These result in system accelerations and should be varied. However, for the final analysis this was not taken into account due to implementation problems in the FEM software.

The tip of the gangway can be connected to an offshore platform. This connection is realized by pushing against the platform and creating a frictional force as such, both horizontally and vertically. Sometimes, the gangway is pushed below a structure, pushing the gangway upwards to maintain the connection. These loads fall under the category of loads on access ways and platforms. Another example is the load of the gangway support structure in stowed condition, which can either be positioned under the main boom or the telescoping boom. To aid the support in stormy weather, an additional passive luffing force can be added, which can especially be key when placing the support under the telescoping boom. And finally, when moving the gangway with the height adjustable cursor, a vertical acceleration is induced which also falls under this category.

The category live loads describes all loads that can move. People on the gangway, a moving trolley and cargo on the hoisting system fall under this category. It is important to include the dynamic factor as well for loads in this category. This factor was provided and already extensively investigated by Ampelmann and will not be described in this research.

Moving cargo will result in some sideways deflection of the hoisting cable, resulting in swing forces. These are defined by the cable's offlead and sidelead angles, as well as the cargo mass and the cable length. The cable length is assumed to be constant in this research.

3.4. Load combinations

The Ampelmann systems are all certified with either Lloyd's register ([60]) or DNV ([17],[16]). These companies provide codes that the Ampelmann system needs to comply with in order to be certified for safe operation. Both state that there are different categories of load cases that need to be assessed.

- Group I: Normal operation - no wind
- Group II: Normal operation - wind
- Group III: Emergency conditions

These groups need to be assessed for all operational positions, including the stowed position when the system is in rest. Within these groups, engineers decide what possible operational scenarios the system can encounter. The system can be oriented in a certain range, vessel accelerations might have their limit, there might be wind playing a role, etc. For example, one cannot transfer people in the same range of luffing angles as one could transfer cargo due to the high inclination of the gangway. Therefore, for each category of load cases, a range is defined for each load parameter. This resulted in tables of parameter ranges for different load case categories, which was the starting point for this research. These tables can be found in Appendix C.

In theory, the table of load cases indicates that there are millions of load combinations defined for the system. If all possible combinations of all parameters must be checked in a FEM software, this is a painfully inefficient process. However, the structural integrity must be guaranteed. Currently, engineering judgment is used to reduce the amount of load scenarios, but it is still far from efficient. For this research, some load types were not included as described in previous section 3.3, which reduced the total number of load cases to 4016. The final model input can be found in Figures D.3 and D.4 in Appendix D.

4

Methodology

The previous chapters described the research questions, academic background on data-driven surrogate models and a real-world complex case to approach with a surrogate modelling technique. This chapter describes the path followed to obtain answers for the research questions as stated in the introduction, by elaborating the methodology of applying a multi-fidelity data-driven surrogate model to the complex case. A visualization of the workflow can be found in Figure 4.1. The hypothesis states that this multi-fidelity approach requires less high-fidelity data to reach similar accuracy compared to a model that would just be trained on high-fidelity data only.

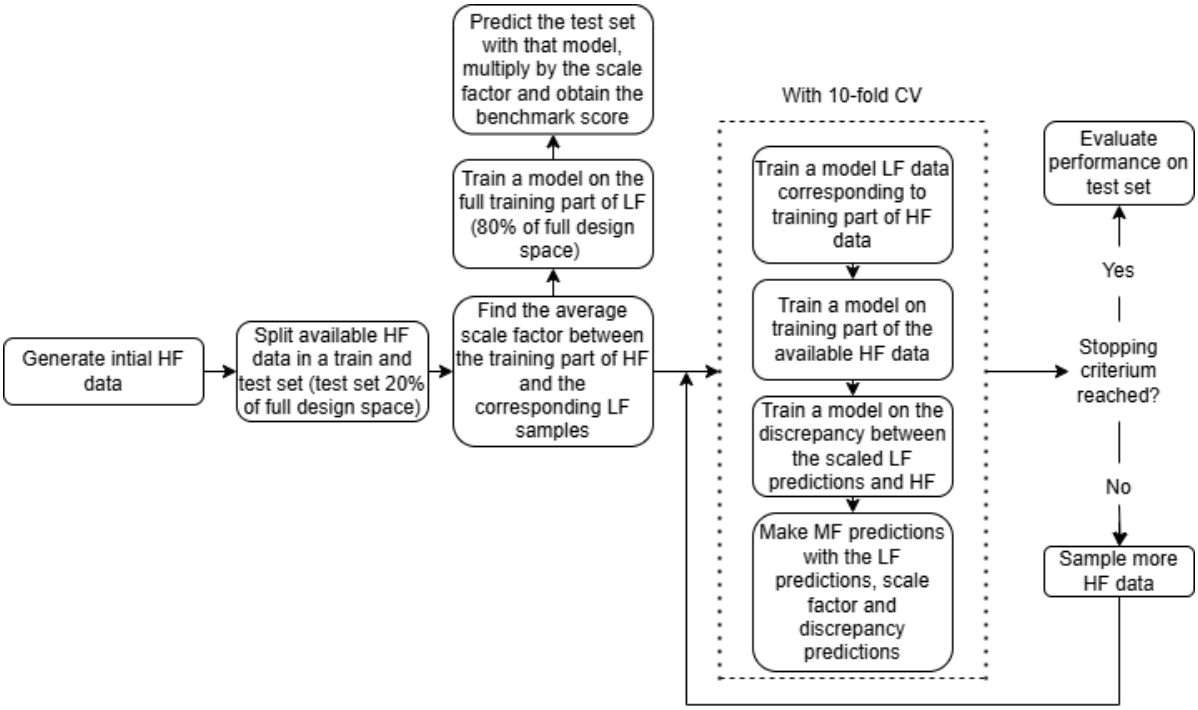


Figure 4.1: Workflow of the multi-fidelity framework

This chapter elaborates on all steps required to implement this framework. Section 4.1 starts by explaining the process of data sampling of both fidelities. The desired output of the surrogate is given, followed by a detailed description of the models that generated the low- and high-fidelity data. The proposed sampling method can also be found in this section, together with a proposed alternative. Then, Section 4.2 follows by elaborating all aspects of implementing the data-driven framework, including model

selection, data preprocessing, hyperparameter tuning, performance evaluation and implementation in a multi-fidelity framework.

4.1. Data sampling

To start off this research, the functionality of the surrogate model should be clarified. The input for the surrogate model has already been defined by the load combinations in Section 3.4. The output is the severity per load case, but there lacks a numerical representation for this ranking of severity. Therefore, the research starts with defining a numerical value for the model output in Section 4.1.1. The partner company, Ampelmann Operations B.V., provided diverse methods of approximating the structural integrity of their gangway. For fast approximations, they provided a python model that could calculate resulting forces and moments given an excel sheet with all load parameters. For detailed evaluation of the structure, they provided diverse FEM models in the software RFEM. It was decided to use the python model as a low-fidelity approximation, and the FEM model as a high-fidelity data source. However, both models need to be extended to be able to generate the low- and high-fidelity data for this specific research, where they represent the severity of the load cases. Therefore, Section 4.1.2 elaborates on the extension of the Python model and Section 4.1.3 discusses the transformation of the FEM model to a useful high-fidelity model. The proposed sampling method of the data is described in Section 4.1.4.

4.1.1. Define model output

The main objective for aiding the design process of the complex mechanical system is to predict the severity of each of the loading scenarios described in Section 3.4, such that the design can be based on the most governing scenarios in an early stage. Therefore, it is required to assign some value to each load case, representing the severity for structural integrity. A higher value represents a more governing load case.

However, 'structural integrity' is not limited to one simple number. For the analysis of a structure, there are resulting forces, moments, stresses, buckling and deflection. It gets even more complex when including local effects, such as results of a finite-element analysis (FEA). The output of the FEA software is huge. A typical finite element model of a steel structure consists of nodes, elements and boundary conditions. These describe the geometry and the support of the structure. Then, there are loads acting on these elements. The FEA software generates a mesh, cutting the structure in a finite number of small elements. For each of these elements, the software calculates the structural properties of that element. For the case of this research, where a steel structure with only beam members in FEM is considered, this output translates to stresses, strain, deflection, and buckling. This raises the question: how can the output of all the elements be combined into one value representing the structural integrity of the full structure?

There are other researches solving similar issues. Vasilopoulos et al. [69] researched a method to define which structural property governs the structural response of tall buildings. They used different geometries, and investigated the moment and shear induced by live loads, dead load and environmental loads. They proposed a method with comparing ratios to determine, for each combination of geometry and combination of loads, whether the shear or moment was governing. However, they did not state the method of extracting the base moment and shear from FEA. Also, their objective is to determine which specific structural property - moment or shear - governs the given response, rather than combining all structural properties into one response value to determine which load combinations result in the most severe response.

Capuano and Rimoli [11] used machine learning techniques to develop a faster alternative for the finite element method, as they pointed out the computational intensity and singularity issues of the traditional method. However, rather than training the machine learning model on the full structure, they trained every single element on the load inputs, which resulted in a model that could predict the structural response of each particular element in the structure. They refer to these trained elements as *smart finite elements*, and together they are capable of predicting the structural integrity of the full structure. However, the computational time to train all element might become an issue for large and complex finite element models, and the geometry of the structure must be fully known when using the model.

In conclusion, although there are papers solving similar issues, no methods were found in literature that are applicable to this case. Therefore, a new method is proposed. The structural response consists of three output parameters for each element: stress, deflection and buckling. Each of these parameters has an allowable limit that cannot be exceeded. The ratio of occurred response over the limit is calculated for every member for stress, deflection and buckling, and the highest value will be reported as the severity of the load case.

For stress, this limit is defined as the yield strength. After the yield strength, the steel will continue in the range of plastic deformation and cannot return to its original state. Therefore, the ranking of severity due to stress (RoS_σ) is defined as the maximum of the ratios of occurred stress ($\sigma[i]$) over the yield strength (σ_y) of all N beam members:

$$RoS_\sigma = \frac{\sigma_{max}}{\sigma_y} \tag{4.1}$$

The same principle can be applied to deflection. This limit originates from the design requirements of the system and cannot exceed 1/100th gangway length. However, this requirement acts on the total deflection (v) of the gangway. Therefore, the assessment point of deflection should only be in a node in the tip and there is no need for checking all members.

$$RoS_v = \frac{v}{v_{lim}} \tag{4.2}$$

Finally, the principle can be applied to buckling. The maximum compressive strength on all steel members is saved and divided by the critical buckling strength of this member (σ_{cr}). If this ratio is higher or equal to one, it means that the member will buckle.

$$RoS_{buck} = \frac{\sigma_{max}}{\sigma_{cr}} \tag{4.3}$$

Then, for a ranking of the severity regarding the full structural response, the maximum value of the three rankings is used:

$$RoS = max(RoS_\sigma, RoS_\delta, RoS_{buck}) \tag{4.4}$$

However, further research showed that it did not make sense to include deflection in the severity definition. First of all, it appeared to be an operational limit rather than a structural limit. It originates from comfort of people walking over the bridge, as certification companies stated it was too frightening to walk over a bridge that would deflect more than one-hundredth of the gangway length. Another issue with including deflection comes from modelling issues. Conversations with an experienced gangway engineer at the partner company led to the conclusion that both global approximations and FEM models are not accurate enough to capture all factors that affect the total deflection. It was found that they would always do a real life test to assess the deflection and it would not have any meaning to include this in the model for this research. It was decided to remove deflection from the analysis, and the ranking of severity will be defined as follows:

$$RoS = max(RoS_\sigma, RoS_{buck}) \tag{4.5}$$

4.1.2. Prepare the low-fidelity model

The success of any multi-fidelity framework is highly influenced by the availability and quality of low-fidelity data. The partner company, Ampelmann Operations B.V., provided an existing software for calculating the effect of load cases on the system. It uses Newtons 2^{nd} law as basis and calculates the resultant forces and moments in the assessment point for every load case. For the system described in Section 3.2, the assessment point is in the center of the slewing ring, which is where the gangway starts. However, this software is limited to reaction forces and moments, and gives no information on local responses. Calculations on buckling are also not included.

Therefore, one of the first steps should be to extend the low-fidelity model to a usable model for predicting load case severity based on stresses and buckling. Using reaction forces and moments, the

general stresses and buckling strength can be calculated with fundamental principles of engineering mechanics [31]. It is expected that the general predictions of stresses and buckling will give sufficient information on the structural severity of that load case, which can be fine-tuned with local information of high-fidelity data in a later stage.

Table 4.1: Structural properties of the gangway per top two girders and bottom two girders

	Main Boom		**T-Boom**	
	Top	Bottom	Top	Bottom
Length single beam $[mm]$ (L)	2080	1100	1100	1090
Area moment of inertia $[mm^4]$ (I)	6.99E6	2.14E8	9.04E6	6.08E7
Distance to neutral axis $[mm]$ (y)	1987	653	978	482
Cross-sectional area $[mm]$ (A)	4608	16988	3400	8172

For the low-fidelity model, the structure is simplified such that classical beam theory can be applied. First, it is assumed that all load is carried by the four main girders of both the main boom and the T-boom, which will be called the four load load-bearing beams. The dimensions of these beams are constant over the entire main boom and constant over the entire T-boom, but the values per boom differ. The dimensions can be found in Table 4.1 and were derived from the FEM model that the company provided. Please note that these values are for both top beams together, and both bottom beams together. For single beam evaluation the values need to be divided by two. The simplified cross-section is visualized in Figure 4.2.

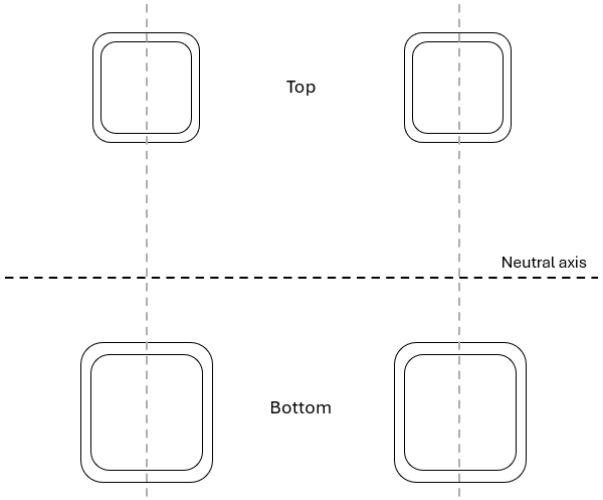


Figure 4.2: Simplified cross-section of the gangway with four load-bearing beams

Stress

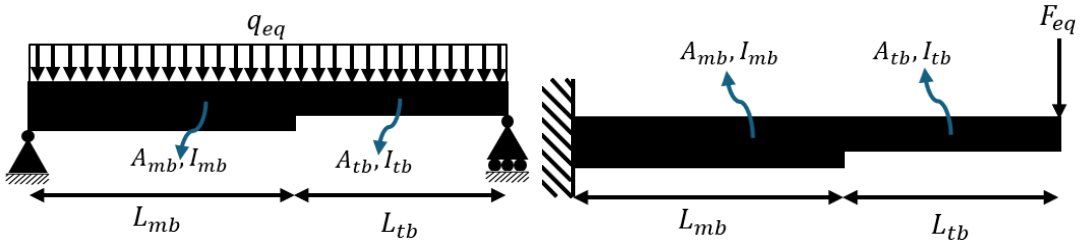
With the simplified cross-section, the maximum global bending stress can be assessed with the classic formula for bending stress of a homogeneous beam (Eq. 4.6) based on the bending moment of the structure (M), distance from neutral axis (y) and the area moment of inertia of the cross section (I).

$$\sigma = -\frac{My}{I} \tag{4.6}$$

However, the gangway is not a homogeneous beam, as the dimensions are different for the main boom and the T-boom. Therefore, the stress assessment must occur separately for both parts.

First, the available Python model gives the resultant forces and moments around the origin. From this, the maximum bending moment of the gangway 'tipping over' (M_{tipp}) is extracted. But this is

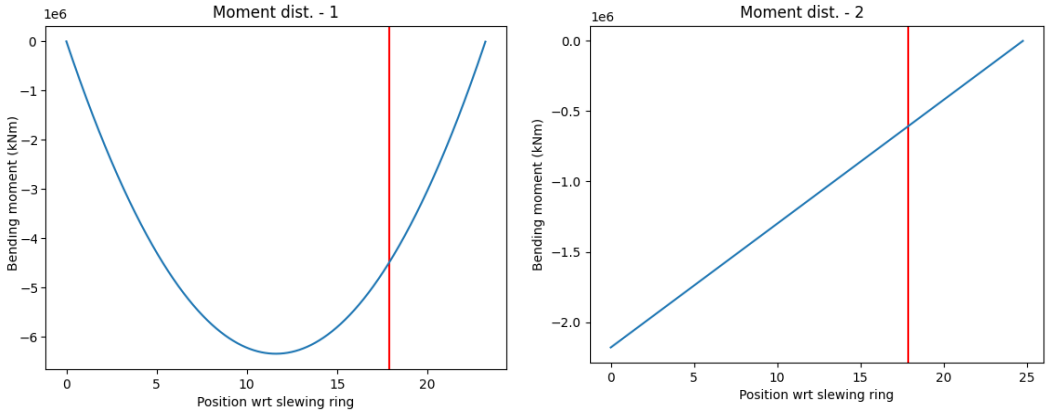
only the resulting moment in the support at the assessment point in the beginning of the gangway. The maximum moment needs to be known for the main boom as well as the T-boom. In order to do so, the moment function is determined. During operation, the gangway can usually be modelled as a clamped-free beam. However, in stowed condition the gangway is supported at the tip and therefore acts as a simply supported beam. Both conditions are visualized in Figure 4.3.



(a) Simply supported **(b)** Clamped-free

Figure 4.3: Sketch LF model

The moment function differs per support type as is visualized in Figure 4.4. Is is assumed that the moment is caused by an equivalent force F_{eq} in normal operation and an equivalent distributed load q_{eq} in stowed conditions as visualized in Figure 4.3. The tipping moment is transposed into an equivalent force F_{eq} applied to the tip (or a distributed force q_{eq} over the gangway length in stowed conditions). To account for the change in direction of this force in situations where the gangway is luffed upwards or downwards, the force is corrected with the cosine of the luffing angle.



(a) Simply supported **(b)** Clamped-free

Figure 4.4: Moment distribution along the gangway for two different support types. The red line indicates the location where the main boom ends and the T-boom starts.

With these moment functions, it can be concluded that:

- In stowed condition, the maximum moment of the main boom is always at L/2.
- In all other conditions, the maximum moment is at the beginning in the origin in the main boom.
- The maximum moment in the T-boom is in both cases where the main boom transitions into the T-boom (red line)

$$M(x) = \frac{4M_{tipp}x}{L^2}(L - x) \text{ if simply supported} \tag{4.7a}$$

$$M(x) = M_{tipp}\left(1 - \frac{x}{L}\right) \text{ if clamped-free} \tag{4.7b}$$

For all load cases, the maximum moment in both booms is calculated with the equations in Eq. 4.7. Then, together with the structural properties in Table 4.1, Eq. 4.6 gives two stresses: one which is maximum in the T-boom and one which is maximum in the main boom. The largest value among both is saved and a severity gets assigned regarding this stress following the method described in Section 4.1.1. Please note that only the bending moment is considered for this analysis, as the transversal moment was deemed less significant for this case.

$$RoS_{\sigma} = \frac{\sigma_{max}}{\sigma_y} \tag{4.8}$$

Buckling

Any structure with steel beams should also be checked on their buckling strength, as it might be the case that buckling failure happens before yielding. Buckling can happen when a beam is subject to a compressive load, which happens for clamped-free supports in the bottom two beams. For stowed condition, in clamped-pinned support, the buckling is evaluated in the upper two beams.

The classical beam theory of Euler and Bernoulli gives a formula to calculate the critical buckling load of a steel beam:

$$P_{cr} = \frac{\pi^2 EI}{(KL_{beam})^2} \tag{4.9}$$

Here, E is the Young's modulus of the material which is taken as 210GPa, I is the moment of inertia of the beam, and L_{beam} the beam length. The latter two properties can be found in Table 4.1. K is the effective length factor, which is dependent on the support of the local beam (see Table 4.2).

Support	K
pinned-pinned	1.0
clamped-free	2.0
clamped-pinned	0.7

Table 4.2: K-values for effective length determination

For normal operation where the compressive load is carried by the bottom girders, it is assumed that all beams are simply supported, resulting in a K-value of 1. However, for stowed condition it is more complex. During stowed condition, buckling occurs in the top girders rather than the bottom girders. The main difference is that the top girders in the T-boom are not supported horizontally, which makes them buckle in a different mode shape: along the full length of the T-boom. Therefore, the buckling length is this case is taken as the full length of the T-boom divided by three.

As the maximum stress occurring in the main boom and t-boom are already calculated in the previous section, the buckling load can be compared to this stress to find whether the structure exceeds the buckling limit. Please note that as there are two load-bearing beams, the maximum stress is divided by two to get the stress per beam.

$$\sigma_{cr} = \frac{P_{cr}}{A} \tag{4.10}$$

$$ROS_{buck} = \frac{\sigma_{max}}{\sigma_{cr}} \tag{4.11}$$

4.1.3. Prepare the high-fidelity model

The high-fidelity data can be created with finite-element software, which can be linked with python code. Then, a script can be written to iteratively calculate loading scenarios acting on existing FEM models which were provided by Ampelmann. The results will then be post-processed into readable data of a unit check representing the local structural integrity per member. Then the highest of all members is saved resulting in a severity ranking.

Experiments were simulated for three FEM versions: the full version, a simplified version of this full model, and a different FEM model which was used in an early-design concept stage.

Complex version

The full FEM model that was provided by Ampelmann for this gangway consisted of 24 structures into one file, each representing a different combination of luffing angle and telescoping stroke. It contained a pre-stress load by tensioned cables and all components such as telescoping wheels and the tip. Also the luffing system is modelled, along with the connection to the slewing ring. All load types were added and all possible load combinations could be simulated, as can be seen in the model input in Figures D.3 and D.4 in Appendix D. A screenshot of the FEM model can be found in Figure 4.5.

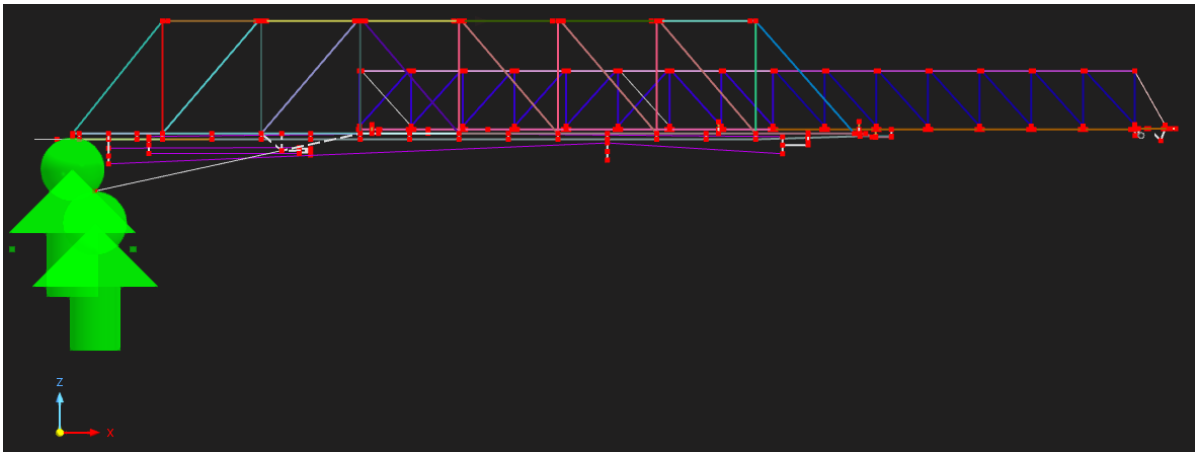


Figure 4.5: One of 24 configurations of the full FEM model

Simplifications

During setup of the data generation, it was found that the available FEM model has its limitations. Certain connections are implemented with a combination of surfaces and small beam elements, too short to comply with beam theory assumptions. But when running a beam model analysis, this results in local stress peaks that do not represent the real world. This problem should be solved, as this severely reduces the physical plausibility of the data and therefore the physical plausibility of the predictions when surrogates are trained on this data. In normal usage, one would decide with engineering judgment whether a member with exceeding stresses is indeed a problem for the structure. Then, irrelevant local peaks would not be considered for evaluation of the structural integrity. However, the approach of the data-driven framework does not contain this engineering judgment. Therefore, it must be avoided to get these irrelevant local peaks.

In order to obtain more consistent stress results that can automatically be analyzed, two measures have been taken. Firstly, it was decided to disregard all point loads. Hence, only load cases with varying vessel accelerations were considered. And secondly, a beam model cannot capture all details of a structure. Some areas will still need to be locally assessed. It was decided to ignore the beam members in these areas, which are visualized in yellow in Figure 4.6. These parts were modelled in detail in a different file outside of this thesis.

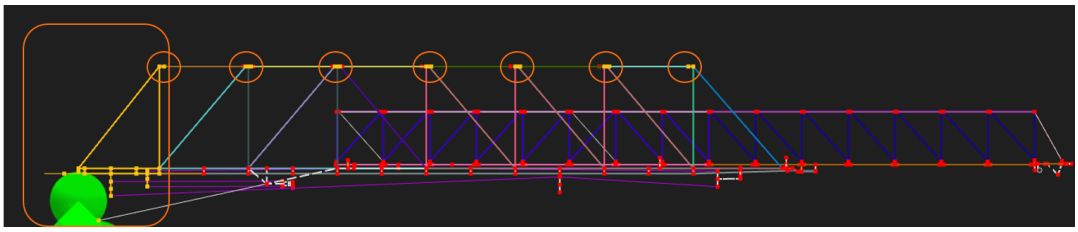


Figure 4.6: Areas that are ignored for the severity calculation in the simplified version

To simplify this model, it was decided to remove all loads except the vessel accelerations due to the heave motion of the ship. Additionally, the model will not take any variations in luffing angle or telescoping

stroke into account. It was decided to start with a luffing position of zero degrees with a telescoping stroke of four meters, as this configuration has the highest number of load cases that are relevant. The new model input with these simplifications can be found in Figure D.2 in Appendix D.

Truss model

Although the previous section removed the most significant modelling issues from the data, it was found that the nature of the FEM model is too complex to analyze in detail with just a beam model. As the previous FEM model contained parts that needed to be analyzed with a surface model, it was not possible to resolve the above modelling issues when performing an analysis using a beam model. Therefore, it was decided to use a different FEM model as well. This model is less accurate, as it does not use the latest dimensions and does not include complex components such as the cable tensioning, compression elements, wheels, etc. It is simply a truss of beam elements (see Figure 4.7). The reasoning for the use of this simple model lies in the scope of this research, which focuses on implementation of a well-predictive model for FEM data rather than implementing a FEM model that represents the real structure the best. With that, the local effects can be fully analyzed with a beam analysis.

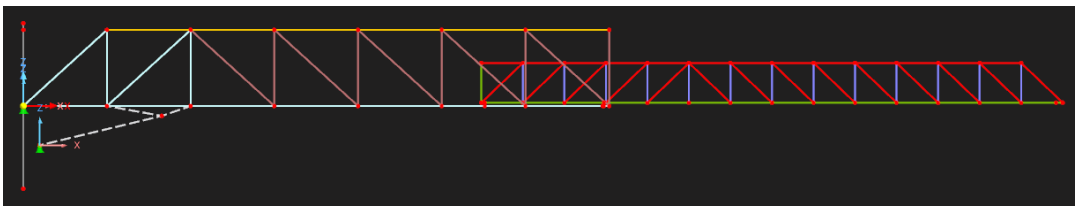
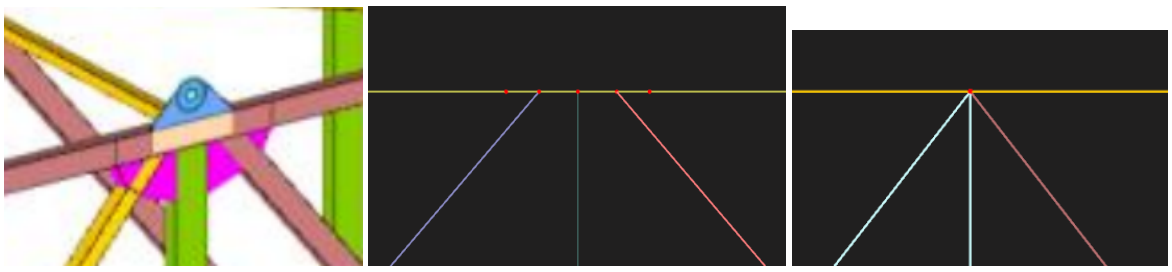


Figure 4.7: Alternative FEM model with a simple truss representing the gangway

Due to the simplicity, it does not contain the modeling issues from the previous model. As stated before, some parts of the complex model were modeled in detail in different files. This new approach refrains from trying to model these complex parts and differs from the complex model in those areas. Figure 4.8 illustrates one of these differences, which occurs at the connection point in the top girders in the main boom. In 4.8a, the most accurate form of the connection of the beams is illustrated, which was analyzed with a surface model in FEM to obtain the most accurate approximation of the real-world stresses. However, in the complex model from Section 4.1.3, this was modeled with beam elements and four different node connections as illustrated by the five red dots in Figure 4.8b. Between these dots, there are small beam elements present which cannot be analyzed correctly with a beam model. To refrain from this issue, the truss model in Figure 4.8c simply connects all beams to the same node. This prevents that the numerically unreliable value of the local peaks taints the severity ranking that gets assigned to the loading scenario.



(a) Correct connection **(b)** Complex model **(c)** Truss model

Figure 4.8: Differences in modelling the connection per FEM model

This was just one example, but there were many more parts of the gangway that showed numerically unreliable peaks, which this simple truss model does not take into account. It is expected that the behavior of this alternative model can be predicted and used for governing load case selection.

There are some limitations to keep in mind when using this FEM model. Firstly, the complex parts such as the connection point in Figure 4.8 must be modelled separately to be able to confidently say that

the structural integrity is intact. Secondly, this model is limited to only one configuration, so the load variations in luffing angle and telescoping stroke cannot be simulated with this model. Also the cargo load cannot be added, as there is no cargo hook simulated in this model that correctly distributes the cargo load over the gangway. The same holds for people on the gangway. This reduced the total amount of load cases to 625. The new model input with this reduced number of loads can be found in Figure D.1 in Appendix D. However, the remaining loads could be added in a later stage, which would allow for simulating high-fidelity data in the full range of the input space. For this research, it is considered out of scope to extend this model to that level of detail.

4.1.4. Sampling method

Section 2.2.1 elaborates on the state of the art regarding data sampling. It showed options for sequential sampling [70], where the design space gets larger in an iterative manner. Usually, this entails sampling adaptively based on the underlying response function, which is referred to as Bayesian optimization. Although this theory shows promise, it remains a challenge to implement a well-performing optimization framework for adaptive sampling for high-dimensional spaces. Although interesting to tackle, it was considered out of scope for this research.

Section 2.1 elaborates on traditional one-shot sampling methods. While this section shows that Latin Hypercube Sampling (LHS) is a popular method in literature due to its stratification along each input dimension, it does not ensure good space-filling properties when applied on a high-dimensional space. In contrast, Euclidean distance-based sampling focuses directly on maximizing the separation between points across the entire design domain, which is especially beneficial for building accurate surrogate models with limited samples such as the case in this research. Also, the multi-fidelity framework requires the data to be sampled in batches, iteratively adding data to the design space, whereas LHS is a one-shot sampling method. Together with the simplicity and geometric clarity of distance-based sampling, it was decided to use Euclidean distance-based sampling for as an alternative to simple random sampling.

It is also possible to use random sampling or distance-based sampling methods in an iterative manner, which will be the proposed method for this research. It was decided to start with random sampling, and implement sampling based on Euclidean distances afterwards to see if it improves the performance when data is scarce. To start off, the full design space will be sampled in low-fidelity, which is the full factorial of the load case table. Then, 25% of the full design space will be randomly selected for high-fidelity evaluation. When sampling more data in high-fidelity, this will also occur randomly. Afterwards, the data selection will be based on Euclidean distances. The Euclidean distance between two points is defined with the formula in Equation 4.12 [34], where N represents the dimensionality of the input space.

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{N} (x_{1,i} - x_{2,i})^2} \tag{4.12}$$

It is expected that sampling based on Euclidean distances will reduce the data size required for a well-performing model, as the full input space will be represented with fewer datapoints.

4.2. Implement the data-driven framework

Data-driven surrogate models will be fitted to the data of different fidelities. Before reaching full complexity of the research, it is beneficial to check the workflow on simple data. Therefore, while the low-fidelity data is generating, a machine-learning framework will be implemented to allow for this simple check. This framework can be used for the high-fidelity data as well. When the machine-learning workflow is fully implemented and the high-fidelity data can be iteratively sampled, the multi-fidelity framework can be implemented, which is described in Section 4.2.4.

The proposal is to implement one linear method for its simplicity, one tree-based method and one kernel-based method as surrogates. Due to its popularity in the field of response surface modelling [52] and ability to handle more complex relations than simple linear regression, the polynomial regression

model is selected from the linear model family. XGBoost is selected as the tree-based model because of the proven performance in current literature [30]. For the kernel-based method, Kernel Ridge Regression (KRR) is selected. It handles large dimensions better than Gaussian Process Regression (GPR) and can be extended with a confidence interval. Its process is similar to SVR, but KRR is better suited for implementation in closed-loop due to the large training time of SVR [51][20]. A short description of the working principles of Ridge Regression and KRR can be found in A. For information on XGBoost, the reader is referred to the research of T. Chen and C. Guestrin [13]. The knowledge in this section originates from books on statistical learning [24][34]. For python implementation, most was created without libraries, but for the algorithms of random hyperparameter sampling, KRR and XGBoost, scikit-learn [57] was used. The remainder of this section elaborates on the methodology of implementing these models regarding data preprocessing, hyperparameter tuning, performance evaluation and implementation in a multi-fidelity framework.

4.2.1. Data preprocessing

Before training and tuning the models, the data should be prepared. After generating the low- and high-fidelity data as described in previous sections, the resulting data will be prepared for use. This entails:

1. Encode categorical features
2. Remove constant features
3. Scale input features

After preparing the data, the tuning can begin. Any machine-leaning model needs to be tuned to find the hyperparameters that result in the best performance. To do this the proper way, the model should be trained on one part of the data and validated on another part. However, every tuned model needs to be evaluated to compare its performance with other models, so after tuning the final model performance on unseen data should be evaluated on the last part of the data. In other words, the data should be splitted in a training, validation and test set.

The most intuitive form for the splitting of data is one-shot splitting, where the data is splitted with a given percentage for the test and validation set. The model is trained, validated and tested only once. However, this method might be unbalanced and not representative of the real performance. Some of the data might not be represented in the validation set, which will lead to a misleading validation score. The way to deal with this issue is the use of K-fold cross-validation.

K-fold cross-validation splits the data K times (usually 3 or 10) where $1/K^{th}$ part of the data is used for validation. Each fold, different data is used for validation and after K folds, all data has been used for validation once. This leads to K different validation scores, and the average represents the cross-validated validation score of the tuned model.

However, this approach does not include a test set for final model evaluation. To avoid reusing data for performance evaluation and creating model bias, the preferred solution among ML engineers is the theory of double cross-validation. The working principle can be found in Appendix B. The downside of this method is the computational burden, as the amount of trained models is K-times as large when using K inner- and outer folds.

The alternative is to use cross-validation for tuning, but keep final model evaluation as a one-shot split. It was decided to follow this approach, as double cross-validation is not suited for a multi-fidelity framework where not all high-fidelity data is available. It was decided to set 20% of the available data apart for final model evaluation, and use 10-fold cross-validation for training and tuning of the model. These numbers were considered a good balance between computational time and trustworthiness of the performance scores.

4.2.2. Hyperparameter tuning

Any ML model needs to be tuned to find the hyperparameters that result in the best performance. These hyperparameters act as settings of the model and can highly influence the performance. As each model has a different working principle, there are also different parameters to be tuned.

For Ridge regression, there is only one parameter to be tuned: the regularization parameter α. As explained in Appendix A, α handles the smoothness of the predictions. However, if α is too large, the model might suffer from underfitting. But a α that is set too low might result in overfitting.

As KRR combines Ridge regression with the kernel trick as explained in Appendix A, the same α from Ridge needs to be tuned for this model. However, KRR has additional hyperparameters to be tuned: the type of kernel, and kernel-dependent hyperparameters. For example, when choosing a polynomial kernel, the degree of polynomial has to be tuned. But for a RBF kernel, the width γ needs to be tuned.

For XGBoost, there are more parameters to be tuned, as this model has a more complex nature compared to the previously described Ridge regression and KRR. The parameters to be tuned are:

- Number of trees
- The learning rate η
- The maximum depth of the trees
- The minimum weight of the children leaf nodes
- The ratio of training data used per tree (subsample)
- The ratio of features used per tree (colsample bytree)
- Regularization parameter for threshold of tree splits (γ)
- Regularization parameters for leaf weights (α and λ for L1- and L2-regularization respectively)

All parameters work together, and if not tuned very carefully, the model is prone to overfit. Especially if the data is very noisy, it can learn the noise by brute force and give a misleadingly great performance. Therefore, if the model is performing suspiciously well, it should be checked whether the model is not overfitting. This could be done by adding randomness to the model, like shuffling target variables. If the model still behaves well, it is clearly capable to learn noise and should be tuned for better regularization.

The preferred method of tuning any machine learning algorithm is to use random grid search in combination with cross-validation. Given a wide range for each hyperparameter, this method selects a combination randomly and trains a model with the approach of K-fold cross-validation, resulting in a trustworthy validation score. Then, this is performed for a large given number of iterations, and the combination of hyperparameters with the best cross-validated score is saved. Afterwards, the same can be performed for a smaller range of parameters for fine-tuning the parameters. This research proposes to use the above-described approach with 10-fold cross-validation and 1000 iterations of selecting hyperparameter combinations.

4.2.3. Performance metrics

The performance of the model needs to be evaluated. This can be done with multiple performance metrics, each having their own strengths and weaknesses. An intuitive form of computing the error is the Mean Absolute Error (MAE), for which the formula can be found in Equation 4.13. This approach sums the absolute differences between the predicted and actual values and calculates the average. The Mean Squared Error in Equation 4.14 or its version with a root (RMSE in Equation 4.15) is very similar, but squares the errors before summing them. Therefore, MSE is more sensitive to outliers than MAE.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.13}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.14}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4.15}$$

Both MAE and (R)MSE approaches are very intuitive and easy to interpret, but they suffer from the lack of scalability. To be able to compare the results with other cases, it is preferred to express the error as a

percentage. For example, the Mean Absolute Percentage Error (MAPE) is very similar to the MAE (see Equation 4.16) but divides the absolute difference between actual and predicted value by the actual value before summing, rather that simply summing the errors themselves. It's strength is the scalability of the result that is introduced by this division, but it also introduces a weakness: when the actual value is zero, or near-zero, the formula is numerically unstable.

$$MAPE = 100 \cdot \frac{1}{N} \sum_{i=1}^{N} \left(\frac{|y_i - \hat{y}_i|}{y_i} \right) \tag{4.16}$$

Another scalable metric is the Relative Error Percentage (REP), for which the formula can be found in Equation 4.17. It is very similar to MAPE, but it uses the squared errors and divides them by the sum of all actual values squared. It can be seen as a relative form of RMSE in Equation 4.15. Like MAPE, it's strength is its scalability but it suffers when the actual data is close to zero.

$$REP = 100 \cdot \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{N} y_i^2}} \tag{4.17}$$

Finally, there are two performance metrics that measure association rather than error: the Pearson Product Moment Correlation Coefficient (PPMCC) in Equation 4.18 and the R-squared score in Equation 4.19. These are both more useful as an addition to get a full picture of nature of the performance, rather than the performance itself. PPMCC represents the correlation of the predicted values versus the actual values. When 1, the values are directly proportional to each other. When -1, the values are inversely proportional to each other. When 0, there is no correlation at all.

$$PPMCC = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{4.18}$$

Rather than measuring correlation, R-squared measures the variance after predictions and compares is with the variance before predictions. For linear cases, this is mathematically the same as taking the square of PPMCC. R-squared of 1 represents the best possible score. R-squared below 0 indicates that the model performs worse than always predicting the mean. A weakness of this model is the scalability, as an R-squared of 0.85 might be great for some cases but not acceptable for others.

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2} \tag{4.19}$$

For this research, it was decided to calculate REP, MAPE, PPMCC and R^2. These four provide a good balance between information on model performance that can be compared with other cases and information that shows the relation of the predicted values compared to actual values.

4.2.4. Multi-fidelity approach

It is very time-consuming to setup a fully correct FEM model and run that model for all possible loading scenarios. It was found that it would take about 20 days to preprocess the loads, calculate them in the FEM simulation an post-process them into a severity if one would require the data for all loading scenarios form the tables in Appendix C. That does not even include the time to setup the FEM model. However, the simple python model as described in Section 4.1.2 would take only two hours for the same amount of data. This justifies the use of a multi-fidelity framework.

The main concept of any multi-fidelity model is that expensive high-fidelity (HF) data is leveraged by cheaper low-fidelity (LF) data, such that the computational load is reduced for an accurate model. By learning from the full low-fidelity dataset and only refining a few data points with high-fidelity simulations, predictions can be made for high-fidelity data of the full dataset. It was decided to follow the comprehensive approach, as this proved to have the best performance compared to the multiplicative

and additive approaches [19]. After each trained model, more data is added to see if the model improves in accuracy. Enlarging the dataset is continued until a certain stopping criterion is reached. The stopping criterion could be a number of things. One could stop the simulations due to a time-constraint, as the sampling is very computationally expensive. Or one could decide to stop running after a certain cross-validated score has been achieved. Another option is to train with known data only and stop when resampling is required. The latter was the approach used for the results of this research, as there is already quite some data sampled in high-fidelity when the final results are required, and it is impractical to always have the sampling setup up and running. The algorithm can be described in detail as follows:

1. Generate LF data on full dataset with the LF model (f_{LF}).

$$y_{LF} = f_{LF}(X_{LF}) \tag{4.20}$$

2. Use a small set for initial HF evaluation and generate that data with the HF FEM simulation (f_{HF}).

$$y_{HF} = f_{HF}(X_{HF}) \tag{4.21}$$

3. Set aside a part of the initial HF dataset for final model evaluation ($y_{HF,test}$).
4. Use cross validation on the remaining part of the HF set for model training and tuning.
   (a) Fit a model on the full LF set minus the test and validation sets to obtain a baseline model trained on all LF data ($\hat{f}_{LF}$).
   (b) Predict the corresponding LF data for the training set.

$$\hat{y}_{LF@HF,train} = \hat{f}_{LF}(X_{HF,train}) \tag{4.22}$$

   (c) Find the average scale factor (ρ) between the LF and HF data and multiple the LF predictions with this factor to obtain a benchmark score for the HF predictions.

$$\rho = \frac{1}{N}\sum_{i=1}^{N}\frac{y_{LF,train}[i]}{y_{HF,train}[i]} \tag{4.23}$$

   (d) Find the discrepancy between scaled predicted LF data and actual HF data.

$$\delta(X_{HF,train}) = y_{HF,train} - \rho \cdot \hat{y}_{LF@HF,train} \tag{4.24}$$

   (e) Learn the discrepancy by fitting a model ($\hat{\delta}$) on $X_{HF,train}$ and δ.
   (f) Make multi-fidelity predictions for the HF data by predicting both $\hat{y}_{LF@HF,val}$ and the discrepancy $\hat{\delta}(X_{HF,val})$.

$$\hat{y}_{HF,val} = \rho \cdot \hat{f}_{LF}(X_{HF,val}) + \hat{\delta}(X_{HF,val}) \tag{4.25}$$

   (g) Evaluate the predictions with performance metrics.
5. Check if model reached the stopping criterion. Sample more HF data if necessary and repeat model training until the cross-validated score converged or the maximum number of iterations is reached.
6. Make multi-fidelity predictions for the test set for final model evaluation.

$$\hat{y}_{HF,test} = \rho \cdot \hat{f}_{LF}(X_{HF,test}) + \hat{\delta}(X_{HF,test}) \tag{4.26}$$

7. Evaluate the predictions with performance metrics
8. Fit a model ($\hat{f}_{LF}$) on the full LF dataset minus the test set and evaluate against the test set to obtain a benchmark score to compare the results.

$$\hat{y}_{LF@HF,test} = \hat{f}_{LF}(X_{HF,test}) \tag{4.27}$$

$$\hat{y}_{HF,test,benchmark} = \rho \cdot \hat{y}_{LF@HF,test} \tag{4.28}$$

Using this algorithm, the model trains $\hat{f}_{LF}$ on the full set of low-fidelity data to generate a coarse estimation of the underlying relationship between input features and target variable. This acts as a baseline model, which can be refined by the discrepancy model $\hat{\delta}$, which is trained on the smaller high-fidelity dataset. This smaller dataset is enlarged by sampling new data and the training repeats.

It is expected that more data will be available the further along in the research. During other activities, the data could be sampled in the meantime. This extra availability of data gives a slight adjustment to the approach. Rather than always starting with the same size of initial dataset, it can be analyzed which percentage of the set needs to be sampled to obtain a well-predicting multi-fidelity surrogate model. When putting this percentage against the performance per used data percentage, it gives a clearer outline of the data requirement of the surrogate. It was decided to increase the high-fidelity data with 1% of the available data each new training to see whether it results in a successful multi-fidelity prediction.

5

Results and discussion

This chapter aims to elaborate on all findings, answering all secondary research questions required to tackle the main research question. Section 5.1 visualizes the generated data and draws some conclusions from the figures. Section 5.2 aims to answer the subquestion regarding data-driven models that can handle the complexity of the input features. Section 5.3 will elaborate on the findings regarding the subquestion about suitable models when data is scarce, followed by the results of the proposed solution of intelligent sampling in Section 5.4. Finally, Section 5.5 aims to answer the subquestion regarding the possibility of a multi-fidelity framework. All findings will be discussed in Section 5.6 and with that, the main research question will be tackled as well.

5.1. Exploratory data analysis

The data generation resulted in two high-fidelity datasets: For the complex model, there were 4016 datapoints in total. However, due to time and computational constraints, only $\tilde{2}5\%$ was sampled in high-fidelity. For the simple truss model, a dataset of 625 datapoints was obtained, for which $\tilde{7}0\%$ of the data was sampled in high-fidelity. This section provides a visualization of the sampled data, which was performed as an exploratory data analysis.

Figure 5.1 shows the severity of all loading scenarios of the for both the low- and high-fidelity data of the complex FEM model. As stated in Section 4.1.1, the structural integrity is respected when the severity is lower than one. The figure shows that the low-fidelity data heavily underestimates the high-fidelity data. However, when applying a scale factor, the correlation between low- and high-fidelity data significantly improves. Also, a lot of datapoints are exceeding the severity limit of one, which would mean that the structure is failing. At first, this result seems surprising as this data represents a functional and fully designed system. However, this can be explained by the local peaks that do not represent real world behavior as explained in Section 4.1.3. One should conclude from this visualization that the local stress peaks due to modelling issues significantly taint the physical plausibility of the high-fidelity model, and it should not be used to select governing load cases due to the untrustworthiness of the data.

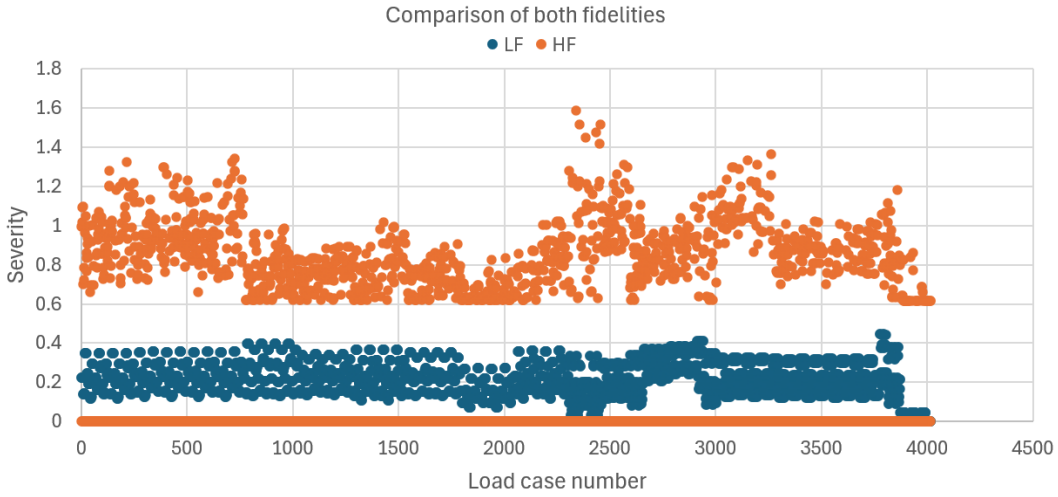


Figure 5.1: Correspondence of low- and high-fidelity data for the complex FEM model

The same analysis has been performed for the simple truss model in Figure 5.2. It shows that the low-fidelity data still heavily underestimates the high-fidelity data. However, similarly to the complex model, the correlation between both fidelities significantly improves when applying a scale factor. Also, the data seems to be more structured with respect to the data of the complex FEM model in Figure 5.1. This can be explained by the difference in model input between the complex FEM model and the simple truss model. The input only contains wind and vessel accelerations for the simple model, whereas the complex FEM model covers all possible loading scenarios. The load cases of the simple truss model in Figure 5.2 are structured as follows: the low-fidelity data shifts the value of the x-acceleration every 125 load cases on the x-axis. Each category of 125 load cases shifts y-acceleration value every 25th load case, z-acc value every 5th loadcase and wind every single load case. The values of the vessel accelerations shift between -5 and 5 m/s^2 with steps of 2.5 m/s^2 , whereas the wind varies between 0 and 20 m/s with steps of 5 m/s.

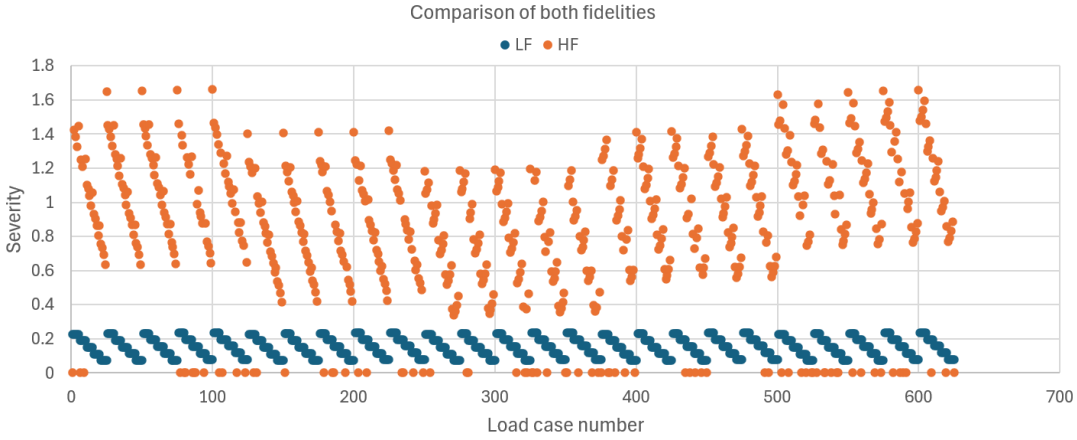


Figure 5.2: Correspondence of low- and high-fidelity data for the simple FEM model

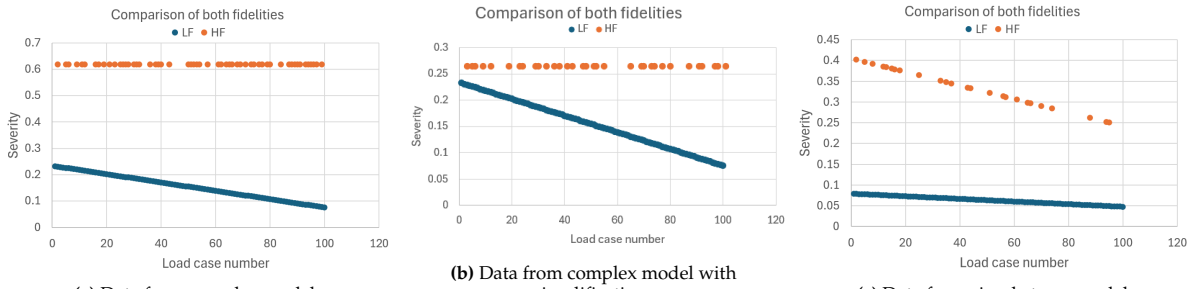
There are three conclusions that can be drawn from this visualization of the data:

1. Since the low-fidelity with every group of 5 adjacent load cases where the only difference in input is the wind force, it can be concluded that the low-fidelity data is insensitive to wind.
2. Both fidelities show barely any effect when varying the acceleration of the sway motion, and a large effect when varying the acceleration of the heave motion.

3. The low-fidelity data shows barely any effect of varying the acceleration in surge motion, while the high-fidelity data gets affected.

One can conclude that variations in all other directions than the z-direction have no effect on the low-fidelity data. This can be explained by the simple nature of the low-fidelity data. Only the tipping moment is taken into account for the stress calculation, which only results in equivalent loads in the z-direction (see Section 4.1.2). All loads that do not affect the tipping moment will have no effect on the low-fidelity data.

Another analysis was performed to grasp the interpretibility and physical plausibility of the high-fidelity models: the full complex version, the version with added simplifications and the simple truss model. In order to understand the underlying relation, it was decided to decrease the z-acceleration input linearly over the load cases. With this input, it is expected that the output will be linear as well.



(a) Data from complex model

(b) Data from complex model with simplifications

(c) Data from simple truss model

Figure 5.3: Correspondence of LF and HF when varying z-acceleration

Figure 5.3 shows the results of varying the z-acceleration linearly. It was expected that the high-fidelity data was linearly correlated with the variations of the z-acceleration. However, the high-fidelity results were constant for this analysis in Figures 5.3a and 5.3b, which would indicate that the z-acceleration has no effect on the severity for both the complex FEM model and the same model with simplifications. It can be concluded that only the simple truss model behaves as expected, as Figure 5.3c shows a linear relationship of the high-fidelity data between the load cases with varying z-accelerations and the severity. This indicates that the data of the simple truss model can be trusted. However, the complex model, even with simplifications, shows untrustworthy results. No matter the value of z-acceleration, there is a local stress reporting the same value, removing all physical plausibility from this data. The true response to a varying z-acceleration appears to be hidden in the complexity of the model due to modelling issues as described in Section 4.1.3. This analysis shows that the simplifications described in that same section do not reveal the expected structural behavior. It only took away the constant local peaks with high values and shows a constant local peak with a lower magnitude, which is still untrustworthy and not physically plausible. This removes the added value from the simplified complex model and therefore will be disregarded for the remainder of this research.

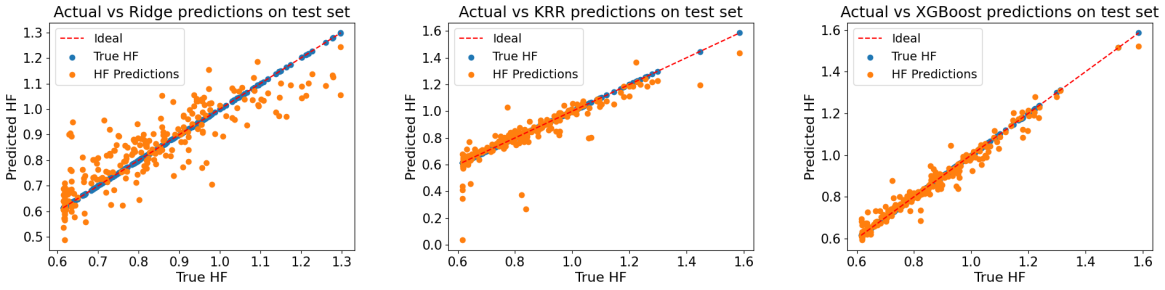
Finally, the nature of the severity of the structure was analyzed. Although Section 4.1.1 describes the severity as a combination of sensitivity to buckling and risk of exceeding the yield strength, it was found that the severity is always defined by stress rather than buckling for both the low- and high-fidelity model. For the low-fidelity model, this indicates that the proposed method of treating the gangway as an Euler-Bernoulli beam is not sufficient for estimating buckling responses for this type of complex structures. For the high-fidelity model, the fact that the Von-Mises stress was more governing than buckling can be explained by the fact that the case study represents a fully-designed structure, where beams prone to buckling already have been reinforced. This finding is heavily dependent on this case, and in earlier design phases buckling would most likely appear to be the failing mode for some cases.

5.2. Suitability for complexity of input features

As described in Section 4.2, three models were selected to be trained on the data of this case. It was suspected that not all models are able to handle the complex nature of the design variables. The three selected models were the linear method Ridge regression, the kernel method KRR and the tree-based

model XGBoost. This section describes the findings after all three models were implemented.

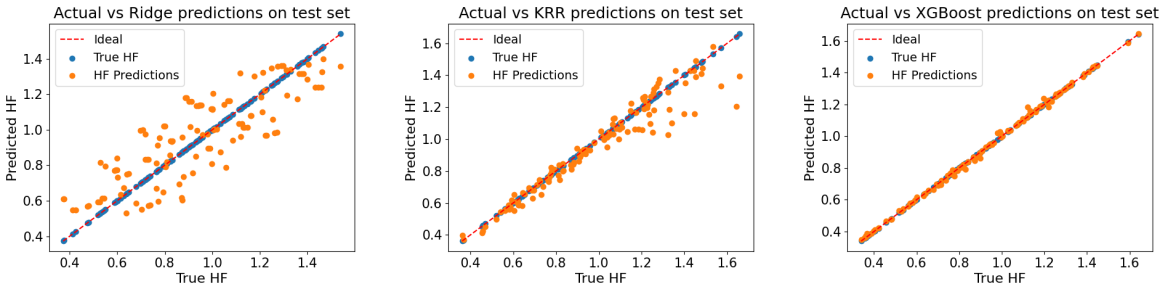
First, the results on the data with full complexity (see Section 4.1.3) should be discussed. Figure 5.4 shows the model performance of Ridge regression, KRR and XGBoost from left to right. All models were trained with 10-fold CV, and a hold-out test set of 20% of the available HF data was kept apart. The performance on the test set is visualized by plotting the actual HF data against the predicted HF data. Ideally, this would result in a straight line where the values on the y-axis always match the values on the x-axis, as this would imply that all predictions are 100% correct. This ideal line has been plotted as a red dashed line. On this line the actual HF datapoints can be found in blue. The predicted datapoints are shown in orange.



(a) Ridge **(b)** KRR **(c)** XGBoost

Figure 5.4: Model performances on test set of the complex HF model for Ridge regression, KRR and XGBoost

To rule out the suspicion that the the model might be thrown off by the untrustworthiness of the complex FEM model, the same has been performed for the simple truss model (see Section 4.1.3). Figure 5.5 shows the results.



(a) Ridge **(b)** KRR **(c)** XGBoost

Figure 5.5: Model performances on test set of the simple HF model for Ridge regression, KRR and XGBoost

With the results of both the complex model and the truss model, there are some conclusions to be drawn from these figures. XGBoost in Figures 5.4c and 5.5c performs really well. Especially the performance on the predictions of the truss model are exceptional, as all predictions are almost on the ideal line. The final REP for the complex model and the truss model are 3.80% and 1.34% respectively. As expected, the model with more complex data is more difficult to predict and shows a higher REP.

For Ridge regression in Figures 5.4a and 5.5a respectively, the performance is very similar. With a final REP on the test set of 10.3% and 15.8% for the complex model and truss model respectively, it can be concluded that the model has a significant error. Especially if you compare the performance with XGBoost in Figures 5.4c and 5.5c it is clear that there are better options out there. Ridge regression assumes linear relationships, and the effect of the input features of this case is too complex to assume linearity. Especially for the complex FEM model, with categorical values in the input, it can be concluded that the complexity of the design space is too much for Ridge regression to comprehend. Interestingly, as opposed to XGBoost, it does not get worse when trying to predict the complex data. In fact, the model struggles more with the simple truss data. This can be explained by the fact that the variance of

the simple truss data is larger than the variance of the complex data (see Figures 5.1 and 5.2). Ridge regression is not capable of capturing the underlying relationship as shown by the bad performance scores, which means that the model will benefit from data that is more centered around the mean. That way, bad understanding of the underlying relationship will lead to a lower relative error with respect to data that further deviates from the mean.

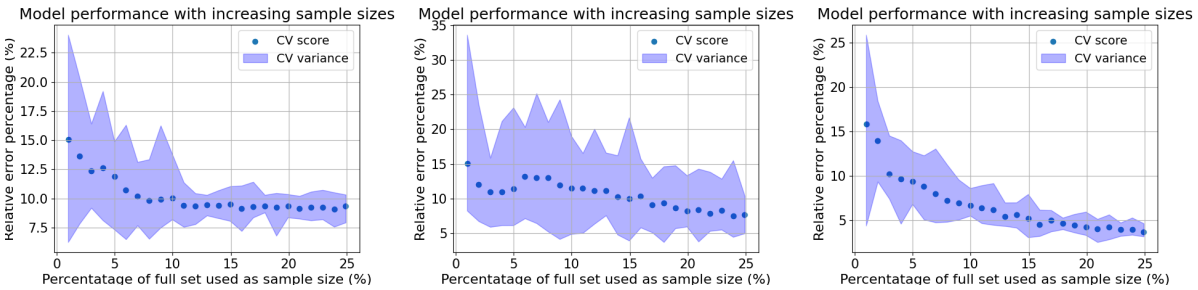
Finally, KRR performs better than Ridge, especially on the simple truss data. On the complex dataset, a REP of 9.45% was achieved, and the data from the truss model resulted in a similar performance: a REP of 8.52%. When these results are compared to the XGBoost models in Figures 5.4c and 5.5c, it can be concluded that XGBoost mainly outperforms KRR for the simple truss model. For the complex data, the performance appears to be very similar, but the REP on the test set shows that XGBoost clearly outperforms KRR: 3.8% versus 9.4% for XGBoost and KRR respectively. The performance gap between XGBoost and KRR can be explained by the structure of the data. As seen in Figures 5.1 and 5.2, the patterns in the data show sudden shifts and plateaus, rather than gradual transitions. KRR tends to perform poorly under such conditions, whereas XGBoost is known to excel in these situations, especially when working with tabular datasets that include categorical values or features where the number of intermediate data points is limited.

In conclusion, when it comes to complexity of the relationship between input features and target variable, it is clear that the linear regression model cannot handle the complexity due to non-linear patterns. KRR performs better, but cannot handle certain piece-wise relationships as well as XGBoost, which outperforms the other models on handling data complexity.

5.3. Suitability for data-scarcity

The problem statement of the case as described in Chapter 3 is heavily caused by the time-consuming process of generating FEM data. Therefore, it must be analyzed which machine learning approach works best when data is scarce. Following the methodology in Section 4.2, the same three models have been selected, trained and tuned on the high-fidelity data of both the complex model and the simple truss model. This section describes the findings.

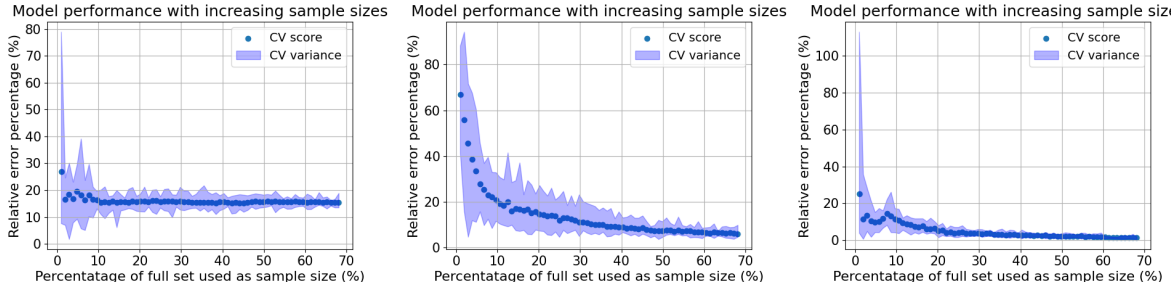
In order to rank the performance when data is scarce, the learning curve should be plotted. This curve shows the performance of the model when trained on different sample sizes. Then, one can compare the three models based on the sample size that is required to obtain a well-performing predictive surrogate model. This has been done for both the complex model and the simple truss model.



(a) Ridge **(b)** KRR **(c)** XGBoost

Figure 5.6: Model performances on test set of the complex HF model for Ridge regression, KRR and XGBoost

Figure 5.6 shows the learning curves on the data of the complex FEM model for Ridge regression, KRR and XGBoost respectively from left to right. Figure 5.7 shows the result of the truss model in the same format. On the x-axis, the percentage of the full dataset that was used for training is shown. The y-axis shows the REP of each cross-validated model. The scores are visualized in blue, with a blue shade representing the variance of the cross-validated score, which acts as a quantification of the uncertainty of the predictions. Each blue dot represents a new model that has been trained with a specific sample size. It was decided to add 1% of the available HF data each new training, which was found to be the best balance between computational capacity and capturing the full picture of the learning curve.



(a) Ridge **(b)** KRR **(c)** XGBoost

Figure 5.7: Model performances on test set of the simple HF model for Ridge regression, KRR and XGBoost

The figures in this section result in a few conclusions. First of all, the result of the previous section is further confirmed. It was found that Ridge regression cannot capture the complexity of the underlying relationship because the REP never dropped below 9%, no matter which FEM model was used. This is confirmed by the lack of improvement with larger sample sizes after a certain threshold as shown in Figures 5.6a and 5.7a. This rules out that the model performance is limited by the data availability, and therefore it must be due to the data complexity. However, it must be noted that Ridge regression reaches its best performance already with 8% known data for both the complex model and the truss model, so data-scarcity is not a problem for Ridge regression.

Secondly, it can be seen in Figure 5.6b that, although KRR improves with more data, it needs a lot of data to reach its optimal performance. Especially when comparing the figure with the performance of XGBoost in Figure 5.6c, which already drops below 10% REP after sampling 4% of the data. KRR on the other hand needs 17% for the same achievement. This leads to the conclusion that XBGoost outperforms KRR when data is scarce in the complex dataset.

For the simple truss dataset in Figure 5.7, the same conclusion can be drawn. The performances of KRR models quickly improve in the first 10% of the dataset, and afterwards the learning curve flattens out. For better visualization, a zoomed-in version of XGBoost on the simple truss data can be found in Figure 5.8. Here, it becomes clear that XGBoost once again outperforms KRR. XGBoost needs only 11% of the data to achieve a stable the REP below 10, whereas KRR requires 40% to achieve the same. With that amount of data, XGBoost already reports a REP of 2%. With that, it can be concluded that, for the simple truss dataset as well as for the complex dataset, XGBoost outperforms KRR when data is scarce.

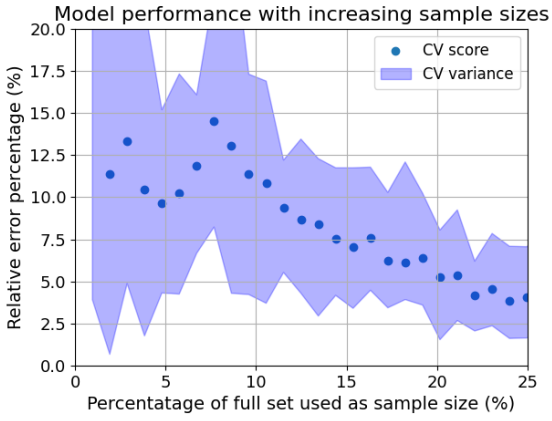


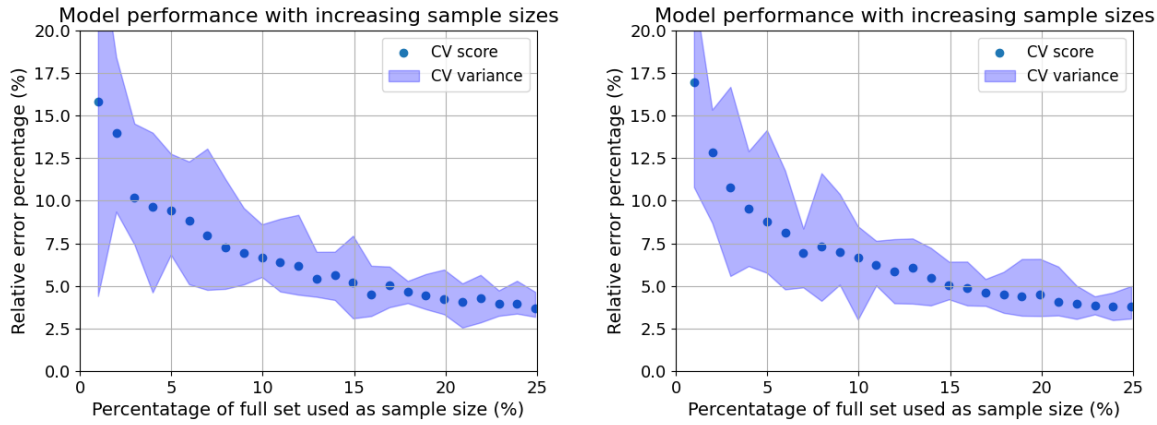
Figure 5.8: Model performances on test set of the simple HF model for XGBoost

5.4. Effect of intelligent sampling

One of the secondary research questions was to investigate the added value of intelligent sampling techniques. It was proposed to compare random sampling with sampling based on Euclidean distances

as reasoned in Section 4.1.4. It is expected that sampling based on Euclidean distances will reduce the data size required for a well-performing model, as the full input space will be represented with fewer datapoints. In order to obtain a full picture of the effect of a sampling strategy, the framework was run with both datasets, the simple truss model and the complex FEM model, both for random sampling and sampling based on the Euclidean distance. This was performed for all three machine learning models.

Figure 5.9 shows that the performance per sampled percentage does not change when sampling based on Euclidean distances rather than normally. In Appendix E the results of the other surrogates for the simple truss model and/or a different machine learning model can be found. These also include an attempt to improve the multi-fidelity results from next Section 5.5. However, neither of the results showed different behavior. With that, it can be concluded that sampling based on Euclidean distances rather than randomly has no significant effect on the performance of the surrogates.



(a) Resampling randomly

(b) Resampling Euclidean distance-based

Figure 5.9: Sampling randomly (left) and sampling based on Euclidean distances (right) - XGBoost, complex model, HF only

A possible explanation for this lack of improvement could be that both the performance of the predictive models for both the simple truss model and the complex model are not sensitive to clustering when sampling the data. For the simple model, the underlying relationship is quite linear. This indicates that the relationship is just as well captured when sampling two points that are close together, rather than sampling two points that represent a different part of the design space. For the complex dataset this is not the case. However, the complex dataset and the simple dataset were sampled differently. For example the vessel accelerations can either be 3 or -3 in the complex input, but they can be in the domain [-5, 5] with steps of 1 for the simple dataset. With this lack of representing the intermediate values in the complex dataset, the risk of clustering is very limited. All datapoints represent a different extreme and the datapoints are well-spread. Therefore, sampling random points will still give a good distance in the input space which reduces clustering. With that, sampling based on Euclidean distances is redundant.

5.5. Multi-fidelity effectivity

This section aims to answer the question whether multi-fidelity can improve model performance for a surrogate on this complex case study when data is scarce. The plots in this section use the same format as described in Section 5.3. Also, since Section 5.4 showed no significant improvement when resampling based on Euclidean distances rather than randomly, this section continues with the approach of sampling the new data randomly.

For the analysis on the effectivity of the multi-fidelity framework, both the complex FEM data and the data resulting from the simple truss model were used for training the surrogates. Also, to give a complete picture of the effectiveness of using multi-fidelity rather than single fidelity, a model will be trained on high-fidelity data only. All figures in this section show the cross-validated REP score with increasing sample sizes. As stated before, the blue shade around the datapoints represent the variance during cross-validation. At 0 on the x-axis, it means that 0% of the high-fidelity data was used

for training. For the multi-fidelity plots, the location at 0% represents the performance of an attempt to predict high-fidelity data when only trained on the low-fidelity dataset. The resulting score can be treated as the benchmark score. For the high-fidelity plots, when 0% of the dataset was used for training, it indicates that there was no training at all and therefore the results start at 1%. For completeness, the final model performance on the test set after training on all available high-fidelity data for all four surrogates in this section can be found in Tables 5.1 and 5.2 for XGBoost and KRR respectively.

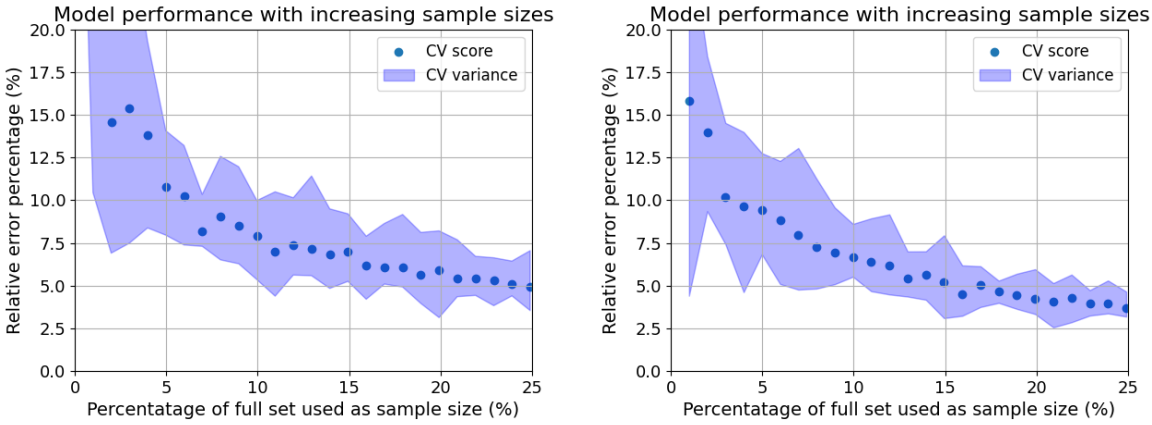
Table 5.1: Performance of XGBoost on the test set of the MF surrogate next to their corresponding HF only surrogates

Metric	**Complex model**		**Truss model**	
	MF	HF	MF	HF
REP	5.00%	3.80%	0.35%	1.35%
MAPE	3.17%	2.52%	0.33%	1.03%
PPMCC	0.97	0.98	0.99	0.99
R^2	0.93	0.96	0.99	0.99

Table 5.2: Performance of KRR on the test set of the MF surrogate next to their corresponding HF only surrogates

Metric	**Complex model**		**Truss model**	
	MF	HF	MF	HF
REP	6.97%	9.45%	3.21%	8.52%
MAPE	4.81%	4.78%	2.62%	4.90%
PPMCC	0.94	0.90	0.99	0.96
R^2	0.89	0.81	0.99	0.93

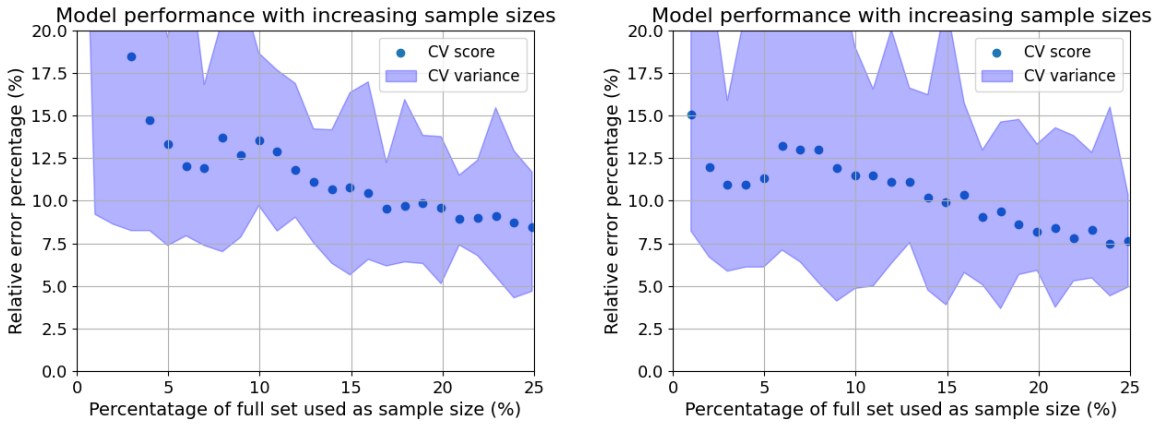
Figure 5.10 and Figure 5.11 show the resulting figures for the complex dataset for XGBoost and KRR respectively. It is clear that the multi-fidelity does not improve the data size required for full predictions. In fact, it even appears to worsen the performance from XGBoost if the final performance metrics on the test set are considered, which can be found in Tables 5.1 and 5.2 for XGBoost and KRR respectively. The performance of XGBoost reduced from 3.8% REP to 5% REP. This can be explained by the common knowledge in statistical learning, stating that adding uncorrelated data during training only confuses the model and might worsen the performance.



(a) Multi-fidelity complex model

(b) High-fidelity complex model

Figure 5.10: Model performance of XGBoost on the complex model with increasing sample size for the multi-fidelity framework (left) and a baseline model trained in high-fidelity only (right).



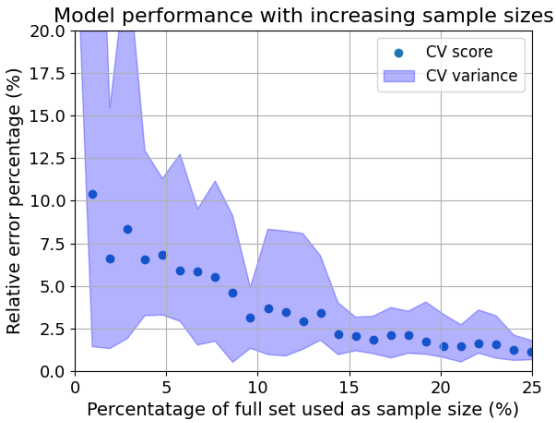
(a) Multi-fidelity complex model

(b) High-fidelity complex model

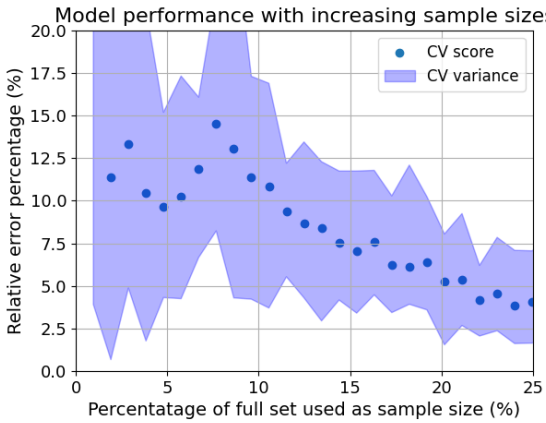
Figure 5.11: Model performance of KRR on the complex model with increasing sample size for the multi-fidelity framework (left) and a baseline model trained in high-fidelity only (right).

However, the performance of KRR increased as the REP changed from 9.45% to 6.97% by using multi-fidelity as shown in Table 5.2. It appears that training on the low-fidelity dataset, although badly correlated with the high-fidelity, does improve performance for KRR. However, KRR never manages to achieve the same accuracy as XGBoost and therefore it is not recommended to use KRR, even if multi-fidelity appears to improve performance. When comparing the high-fidelity plot of XGBoost in Figure 5.10b with the multi-fidelity plot of KRR in Figure 5.11a the performance is better for XGBoost in high-fidelity only. This indicates that it would be just as good to simply train XGBoost on high-fidelity only, which consumes a fraction of the time that would be required to implement a multi-fidelity framework with KRR.

If the same analysis is performed on the dataset of the simple truss model for both models (see Figures 5.12 and 5.13), a different conclusion can be drawn. Here, adding low-fidelity data to the training actually decreased the data size required to obtain a well-performing model. The multi-fidelity approach with XGBoost in Figure 5.12a achieves a REP of 5% after 9% trained in high-fidelity. The corresponding high-fidelity surrogate in Figure 5.12b only shows this achievement when 23% of the high-fidelity data is available. Also, a better REP can be achieved by this approach. Similar results were found for KRR in Figure 5.13, where KRR even reports a REP of almost 5% after 25% of the data, a performance that had not been achieved by KRR before. This is a great improvement when comparing it to its corresponding surrogate trained on high-fidelity only in Figure 5.13b, and it is again confirmed that multi-fidelity works for the simple truss model.

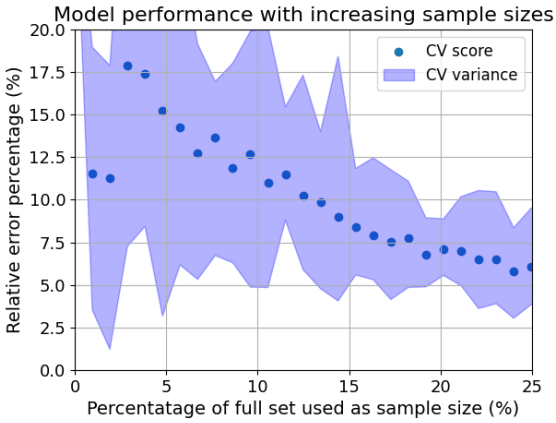


(a) Multi-fidelity simple model

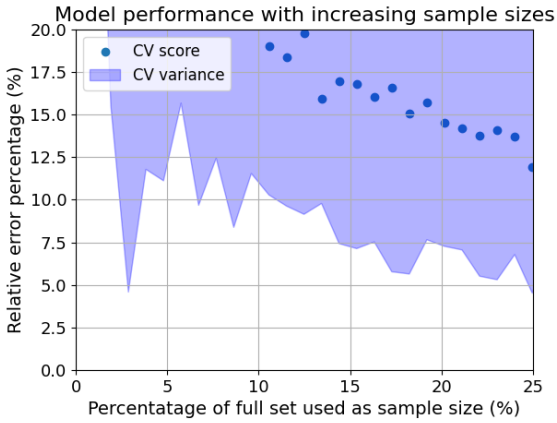


(b) High-fidelity simple model

Figure 5.12: XGBoost performance on the simple truss model with increasing sample size for the multi-fidelity framework (left) and a baseline model trained in high-fidelity only (right).



(a) Multi-fidelity simple model



(b) High-fidelity simple model

Figure 5.13: KRR performance on the simple truss model with increasing sample size for the multi-fidelity framework (left) and a baseline model trained in high-fidelity only (right).

When considering the exploratory data analysis in Section 5.1, it comes as no surprise that the effectivity of the multi-fidelity framework is limited to the dataset generated with the simple truss model. The low-fidelity data corresponding to the complex FEM model cannot capture the complexity of the high-fidelity data at all, resulting in noise in the data which worsens the performance for XGBoost.

In conclusion, the results of this section show that multi-fidelity can be a good option for this case study, using a simple python model to guide a surrogate model for a FEM model. However, the FEM data must represent the true response rather than local stress peaks due to modelling issues such that the low-fidelity data is well-correlated to the high-fidelity data. Only then, the multi-fidelity approach can effectively link a severity based on FEM results to a parametrized loading scenario. With that, the outcome of the structure in FEM can be linked to the input features, which enables surrogate-based optimization. However, one must be really careful in constructing the FEM model. Local peaks are allowed, but they should be trustworthy. If the FEM model requires engineering judgment to disregard certain exceedings of structural limits, it is not suited for this approach as shown in Figures 5.10 and 5.11. But the model can still be useful to predict which loading scenarios result in these exceedings, such that the engineer can be guided towards beam members that need attention. Both results should really speed up the design process.

5.6. Discussion

Although the results have shown a successful achievement regarding predictive surrogates for the high-fidelity data, and a multi-fidelity framework that works on the simple truss model, there are some aspects to keep in mind. This section discusses these aspects.

First of all, there are some considerations regarding the usability of the framework for structural design due to the source of the high-fidelity data. The essence of high-fidelity data is that it should be highly accurate. However, as stated in Section 4.1.3, there are some limitations regarding the high-fidelity data used in this research. The approach to generate high-fidelity data is only trustworthy if one can trust all modelling aspects. Only then, the precise method of FEM actually results in accurate local responses. If this is not the case, these modelling issues might introduce local effects in the structure that do not represent the real-life structural response. This severely lowers the trustworthiness of the high-fidelity data. However, this does not indicate that the surrogate is not useful for detailed engineering. It simply changes perspective. The purpose of predicting governing load cases is not relevant for this case since the data does not represent real-world responses. But it can still accelerate the structural design, because all exceeding members need to be checked by an engineer who can decide whether the member should be reinforced or disregarded. In the case of this complex model which has already been fully designed, it will always be the latter. But for future purposes, this surrogate, proven to have well-predictive capabilities even though the data represents modelling issues, can point engineers to all load cases that result in exceeding members. Therefore, the surrogate could aid engineers by acting as a pointer to load cases that require engineering judgment.

It could have been expected that the multi-fidelity approach stops adding value when the data is governed by modelling issues rather than real-world structural response, and it explains that multi-fidelity only works for the simple truss model. All parts of this FEM model can be fully trusted. With that, all local responses represent real-world responses and the low-fidelity data does correspond to the high-fidelity data as opposed to the complex model, resulting in a successful multi-fidelity surrogate. However, this brings other limitations to the surface. The complexity of the case has been reduced tremendously, as this new version of high-fidelity data only originates from varied wind force and vessel accelerations. Further work should include an attempt to extend this simple truss model with the remaining loads in such a way that all modelling aspects can be fully trusted. Without this extension, the usability of the surrogate presented in this research is limited. It is expected that it could be extended to the level of detail required to cover all loads when modelled very carefully. When successful, the main purpose of the research applied to this case can be realized. The high-fidelity data will represent real-life structural responses and the surrogate can be used to determine the governing load cases of all described in the tables in Appendix C.

It should also be kept in mind that one unsuccessful multi-fidelity approach does not imply that multi-fidelity is not an option. Rather, it could indicate that the method for data generation of both fidelities is inadequate. For this research, it is important to note that the low-fidelity model is very basic. As shown in Section 5.1, wind does not affect the results. And there were more loads given as input that did not affect the output. As the model input for the complex dataset contains a lot of these loads that did not affect the low-fidelity results, it was to be expected that the low-fidelity data gives little to added value to the multi-fidelity framework. Future work could attempt to extend the low-fidelity approximations such that all load variables have an effect on the global stress. Alternatively, one could make a different low-fidelity dataset by creating a simple FEM model with a coarse grid. Then, the fidelities would be better correlated which would lead to a more successful multi-fidelity surrogate for the complex data.

Then, there are some aspects of the multi-fidelity framework that should be noted. First of all, the hyperparameter tuning was not performed in the most optimal manner. When trying to plot learning curves as the ones in Section 5.3 up until Section 5.5, the computational burden quickly explodes. Ideally, the hyperparameters should with a random grid search combined with cross-validation be tuned for every point in these plots. This random grid search typically needs around 1000 iterations of selecting a combination of hyperparameters, which leads to 1000 different models to be trained. But the multi-fidelity approach contains two models that require training: one for the low-fidelity predictions, and one for the predictions of the discrepancy, leading to 1.000.000 combinations for every multi-fidelity surrogate. And as cross-validation of 10 folds is advised, the number of trained models becomes 10

million per datapoint in the learning curves. With about 50 points per learning curve, two different datasets, two different model types, both a single-fidelity surrogate and a multi-fidelity surrogate, this would result in more than 500 million trained machine learning models. And the grid refinement that is usually required was not even taken into account for this calculation. The computational capacity in this research was too low to follow this methodology. Therefore, some workarounds were followed.

Firstly, XGBoost models were not tuned during runtime. Although extensively attempted, the approach of tuning the hyperparameters did not result in any combination that performed better than the default parameters. And even if there was an optimum to be found that failed to come to the surface, it was decided that it should not matter that much when the model is performing as well as reported in Section 5.2, given that the model is not overfitting. The latter was checked by shuffling the target variables. When the performance would still be great after shuffling, the model is clearly capable of learning the noise by brute force. However, this was not the case. Therefore, it was decided to keep using the default hyperparameters for XGBoost.

The second workaround relates to the issue of training two models within one surrogate in the multi-fidelity framework. It was found that XGBoost can reproduce the low-fidelity data almost perfectly due to its simple mathematical origin. Therefore, it was decided to keep using XGBoost for the low-fidelity predictions, even when running KRR for the discrepancy model. This disregards the need to tune the hyperparameters for the low-fidelity predictions and the discrepancy predictions simultaneously.

Lastly, it is of utmost importance to consider the time required to implement a multi-fidelity framework for future applications. It should be noted that the benefits of the multi-fidelity framework in this research are marginal, and only apply when a multi-fidelity framework has already been constructed. One should always consider whether the time saved by sampling in high fidelity outweighs the time required to construct a multi-fidelity framework. If the current surrogate model were constructed exclusively for the current use case, training a multi-fidelity model would not be a time-efficient option. However, it is expected that the current surrogate will work for future designs as well. Therefore, the current multi-fidelity model is seen as a valuable time-investment for cases where similar structures are expected to be designed in the future.

6

Conclusion

Surrogate modelling is a promising field that is largely in development, as described in Chapter 2. There are still significant limitations to this approach. For complex mechanical systems such as the case described in Chapter 3, the development of useful surrogate models is mainly limited by the complex nature of the input variables, data-scarcity and the fact that it is often not possible to extend the model to a surrogate-based optimization framework. These limitations formed the research questions. Table 6.1 shows per research question where the method can be found that led to the answer, and the location of the elaboration on the answer.

Table 6.1: Overview of the research questions and the location of their corresponding methodology and answers

Question	**Method**	**Answer**
Which machine learning technique is best suited for the complex nature of design variables?	Section 4.2	Section 5.2
Which surrogate modelling technique can provide accurate predictions when data is scarce	Section 4.2	Section 5.3
How can intelligent sampling improve the model performance when data is scarce?	Section 4.1.4	Section 5.4
How can a multi-fidelity framework effectively leverage low-fidelity data to multi-fidelity output, such that a higher accuracy is reached with less data?	Section 4.2.4	Section 5.5
How can data-driven surrogate modelling techniques, embedded in a multi-fidelity framework, aid and accelerate the structural design process when applied to a complex mechanical system with complex moderate-dimensional input?	Chapter 4	Chapter 5

The main research question of this report was stated as follows: How can data-driven surrogate modelling techniques, embedded in a multi-fidelity framework, aid and accelerate the structural design process when applied to a complex mechanical system with complex moderate-dimensional input? This research proposes to answer this question in a structured matter.

First, a complex mechanical case was found and analyzed, such that useful data could be generated in both low- and high-fidelity. With that, the desired in- and output for the surrogate became clear. The input features would represent the load on the structure during real-life operations, and the output would be a ranking of severity for the structural integrity during that scenario. Then, three different machine learning techniques were selected to train as a surrogate model: Ridge regression, KRR and XGBoost. It was found that Ridge regression was not suited for the complex nature of the design space. KRR performed better, but suffered from the discontinuous relationships in the data. Contrarily,

XGBoost could provide accurate predictions for both the complex data and the simple truss data. This answered the first subquestion, which stated the need for a machine learning technique that can handle the complexity of the input features.

The same techniques were analyzed on their suitability for data-scarcity in order to answer the second subquestion. This led to the conclusion that Ridge regression, although inaccurate, reaches its maximum performance the quickest: only 10% of both analyzed datasets needed to be sampled in high-fidelity in order to achieve a stable maximum performance. XGBoost performed second-best regarding data-scarcity and does not show much more improvement after 25% of the dataset. KRR required the most data for achieving its best performance. Around 50%, KRR starts to report good performance, but it still improves, although very gradually, when adding more data after that.

For the third subquestion, the difference between sampling randomly and sampling based on Euclidean distances was analyzed. However, there was no significant difference to be found in performance of the surrogates. It was found that the limited performance in data-scarce regions is caused by other factors than clustering in the data, as distance-based sampling would have improved performance otherwise.

Finally, the fourth subquestion could be answered. The comprehensive approach of the multi-fidelity theory was applied on both datasets, where the low-fidelity data was scaled with a scaling factor and a model was trained on the discrepancy between the scaled low-fidelity data and the high-fidelity data. It was found that the complex dataset lacked well-correlated low-fidelity data, which resulted in a surrogate that did not improve when using low-fidelity data for predictions of the high-fidelity data. However, the data from the simple truss model showed the expected results. The multi-fidelity surrogate required only 10% of the high-fidelity data to obtain a REP of 3%, whereas the surrogate trained in high-fidelity only required 25% of the high-fidelity data to achieve the same performance. This confirmed the hypothesis of the research, which stated that a multi-fidelity framework would require less expensive high-fidelity data to obtain a well-performing surrogate.

With that, the main research question could be answered. With the accurate predictions when data is scarce, it becomes an option to link the outcome of the structure in FEM to the input features. This enables surrogate-based optimization, which could aid and accelerate the design of the structure, as less expensive FEM simulations are required to obtain information on the severity of the loading scenarios. However, the results showed that the multi-fidelity framework only had marginal benefits. Exclusively for the simple truss model with physically plausible data, there was an improvement found in the datasize required for a well-performing model. It should be possible to extend this model very carefully with the remaining load case factors, without introducing modelling issues. That would result in a working multi-fidelity framework for the full complexity of the case study in Chapter 3 that could provide accurate and physically plausible predictions. In the current state, the use case of multi-fidelity is limited, but it could be promising when extending the truss model.

For FEM models with modelling issues, the data-driven surrogate could also be valuable. Although trained on numerically unreliable stress peaks that come from modelling issues rather than real structural response, it pinpoints the user to load cases that result in beam members exceeding structural limits in the FEM software. And as every exceeding member needs to be judged by an engineer to decide whether the member should be reinforced or ignored, it could be very valuable to have a surrogate pointing to all load cases that result in exceeding members. This will be a great aid for the design engineers and could accelerate the process.

While this is a successful result, future work is recommended to further enhance the value for the scientific world. For example, the issues of interpretability and extrapolation capacities to different geometries have not been tackled in this research. A surrogate could be trained on other gangway types, and the surrogate could be expanded with techniques to improve the interpretibility of the predictions. Also, it could be very beneficial to dive deeper into an analysis of time cost versus profit of implementing a multi-fidelity data-driven surrogate for real applications. It should also be stated that the scope for intelligent sampling techniques should be expanded in future work. This research only compares random sampling with Euclidean distance-based sampling, but there are many more sampling strategies to be tested. It is recommended that future work explores possibilities for Bayesian optimization for complex cases as the one in this research. Regarding the case study in this research, it is recommended to expand the simple truss model, preferably for multiple geometries, such that it can

calculate all possible loading scenarios for multiple gangway types. This would enable surrogate-based optimization for the structure of the gangway, significantly improving the speed at which new designs can be developed.

References

[1] Naks Alex et al. "Optimization with Variable-Fidelity Models Applied to Wing Design". In: 2000 (Mar. 2000). DOI: 10.2514/6.2000-841.

[2] Reza Alizadeh, Janet K. Allen, and Farrokh Mistree. "Managing computational complexity using surrogate models: a critical review". en. In: *Research in Engineering Design* 31.3 (July 2020), pp. 275–298. ISSN: 1435-6066. DOI: 10.1007/s00163-020-00336-7. URL: https://doi.org/10.1007/s00163-020-00336-7 (visited on 11/04/2024).

[3] Esther Andrés-Pérez and Carlos Paulete-Periáñez. "On the application of surrogate regression models for aerodynamic coefficient prediction". en. In: *Complex & Intelligent Systems* 7.4 (Aug. 2021), pp. 1991–2021. ISSN: 2198-6053. DOI: 10.1007/s40747-021-00307-y. URL: https://doi.org/10.1007/s40747-021-00307-y (visited on 10/15/2024).

[4] Omar A. I. Azeem and Silvestre T. Pinho. "A machine learning assisted multifidelity modelling methodology to predict 3D stresses in the vicinity of design features in composite structures". In: *International Journal of Solids and Structures* 301 (Sept. 2024), p. 112946. ISSN: 0020-7683. DOI: 10.1016/j.ijsolstr.2024.112946. URL: https://www.sciencedirect.com/science/article/pii/S0020768324003056 (visited on 11/20/2024).

[5] Ampelmann B.V. *Ampelmann*. en. URL: https://www.ampelmann.nl/ (visited on 12/09/2024).

[6] B. Barros et al. "Design and testing of a decision tree algorithm for early failure detection in steel truss bridges". In: *Engineering Structures* 289 (Aug. 2023), p. 116243. ISSN: 0141-0296. DOI: 10.1016/j.engstruct.2023.116243. URL: https://www.sciencedirect.com/science/article/pii/S0141029623006582 (visited on 10/17/2024).

[7] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. "A comparative analysis of gradient boosting algorithms". en. In: *Artificial Intelligence Review* 54.3 (Mar. 2021), pp. 1937–1967. ISSN: 1573-7462. DOI: 10.1007/s10462-020-09896-5. URL: https://doi.org/10.1007/s10462-020-09896-5 (visited on 12/10/2024).

[8] George E. P. Box and Norman R. Draper. *Empirical model-building and response surfaces*. Empirical model-building and response surfaces. Pages: xiv, 669. Oxford, England: John Wiley & Sons, 1987. ISBN: 978-0-471-81033-9.

[9] S.L. Burgee et al. "Parallel multipoint variable-complexity approximations for multidisciplinary optimization". In: *Proceedings of IEEE Scalable High Performance Computing Conference*. May 1994, pp. 734–740. DOI: 10.1109/SHPCC.1994.296714. URL: https://ieeexplore.ieee.org/abstract/document/296714?casa_token=FlViyWp9ylMAAAAA:UjkNKLEBYUXzAYxlgylS_xeksgQbAM8-J3WrgLAC3vgQYGUPQ61P9WFYY0o5r8REdwrTFBN1yA (visited on 12/03/2024).

[10] Xiwen Cai et al. "Adaptive Radial-Basis-Function-Based Multifidelity Metamodeling for Expensive Black-Box Problems". In: *AIAA Journal* 55.7 (July 2017). Publisher: American Institute of Aeronautics and Astronautics, pp. 2424–2436. ISSN: 0001-1452. DOI: 10.2514/1.J055649. URL: https://arc.aiaa.org/doi/10.2514/1.J055649 (visited on 12/04/2024).

[11] German Capuano and Julian J. Rimoli. "Smart finite elements: A novel machine learning application". In: *Computer Methods in Applied Mechanics and Engineering* 345 (Mar. 2019), pp. 363–381. ISSN: 0045-7825. DOI: 10.1016/j.cma.2018.10.046. URL: https://www.sciencedirect.com/science/article/pii/S0045782518305541 (visited on 02/19/2025).

[12] Shi-Zhi Chen et al. "Embedding Prior Knowledge into Data-Driven Structural Performance Prediction to Extrapolate from Training Domains". EN. In: *Journal of Engineering Mechanics* 149.12 (Dec. 2023). Publisher: American Society of Civil Engineers, p. 04023099. DOI: 10.1061/JENMDT.EMENG-7062. URL: https://ascelibrary.org/doi/10.1061/JENMDT.EMENG-7062 (visited on 11/26/2024).

[13] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: https://dl.acm.org/doi/10.1145/2939672.2939785 (visited on 07/11/2025).

[14] Ankang Cheng and Ying Min Low. "A new metamodel for predicting the nonlinear time-domain response of offshore structures subjected to stochastic wave current and wind loads". In: *Computers & Structures* 297 (July 2024), p. 107340. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2024.107340. URL: https://www.sciencedirect.com/science/article/pii/S0045794924000695 (visited on 11/22/2024).

[15] Kai Cheng and Zhenzhou Lu. "Structural reliability analysis based on ensemble learning of surrogate models". In: *Structural Safety* 83 (Mar. 2020), p. 101905. ISSN: 0167-4730. DOI: 10.1016/j.strusafe.2019.101905. URL: https://www.sciencedirect.com/science/article/pii/S0167473019303698 (visited on 10/15/2024).

[16] DNV. *DNVGL-ST-0358-2022-Offshore gangways*. 2022. URL: www.dnv.com.

[17] DNV. *DNVGL-ST-0378-2021-Offshore and platform lifting appliances*. 2021. URL: www.dnv.com.

[18] Cédric Durantin et al. "Multifidelity surrogate modeling based on radial basis functions". en. In: *Structural and Multidisciplinary Optimization* 56.5 (Nov. 2017), pp. 1061–1075. ISSN: 1615-1488. DOI: 10.1007/s00158-017-1703-7. URL: https://doi.org/10.1007/s00158-017-1703-7 (visited on 12/04/2024).

[19] M. Giselle Fernández-Godino et al. "Issues in Deciding Whether to Use Multifidelity Surrogates". en. In: *AIAA Journal* (Mar. 2019). Publisher: American Institute of Aeronautics and Astronautics. DOI: 10.2514/1.J057750. URL: https://arc.aiaa.org/doi/10.2514/1.J057750 (visited on 12/02/2024).

[20] Alexander I. J. Forrester and Andy J. Keane. "Recent advances in surrogate-based optimization". In: *Progress in Aerospace Sciences* 45.1 (Jan. 2009), pp. 50–79. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2008.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0376042108000766 (visited on 12/02/2024).

[21] Jerome H. Friedman. "Multivariate Adaptive Regression Splines". In: *The Annals of Statistics* 19.1 (Mar. 1991). Publisher: Institute of Mathematical Statistics, pp. 1–67. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1176347963. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-19/issue-1/Multivariate-Adaptive-Regression-Splines/10.1214/aos/1176347963.full (visited on 12/02/2024).

[22] Jerome H. Friedman. "Stochastic gradient boosting". In: *Computational Statistics & Data Analysis*. Nonlinear Methods and Data Mining 38.4 (Feb. 2002), pp. 367–378. ISSN: 0167-9473. DOI: 10.1016/S0167-9473(01)00065-2. URL: https://www.sciencedirect.com/science/article/pii/S0167947301000652 (visited on 12/10/2024).

[23] Jan N. Fuhg, Amélie Fau, and Udo Nackenhorst. "State-of-the-Art and Comparative Review of Adaptive Sampling Methods for Kriging". en. In: *Archives of Computational Methods in Engineering* 28.4 (June 2021), pp. 2689–2747. ISSN: 1886-1784. DOI: 10.1007/s11831-020-09474-6. URL: https://doi.org/10.1007/s11831-020-09474-6 (visited on 11/04/2024).

[24] Aurelien GERON. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools and techniques to build intelligent systems*. OReilly, 2019.

[25] Tushar Goel et al. "Pitfalls of using a single criterion for selecting experimental designs". en. In: *International Journal for Numerical Methods in Engineering* 75.2 (2008), pp. 127–155. ISSN: 1097-0207. DOI: 10.1002/nme.2242. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2242 (visited on 11/04/2024).

[26] Xingquan Guan et al. "Seismic Drift Demand Estimation for Steel Moment Frame Buildings: From Mechanics-Based to Data-Driven Models". EN. In: *Journal of Structural Engineering* 147.6 (June 2021). Publisher: American Society of Civil Engineers, p. 04021058. ISSN: 1943-541X. DOI: 10.1061/(ASCE)ST.1943-541X.0003004. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29ST.1943-541X.0003004 (visited on 11/29/2024).

[27] Dan Guo et al. "Heterogeneous Ensemble-Based Infill Criterion for Evolutionary Multiobjective Optimization of Expensive Problems". In: *IEEE Transactions on Cybernetics* 49.3 (Mar. 2019). Conference Name: IEEE Transactions on Cybernetics, pp. 1012–1025. ISSN: 2168-2275. DOI: 10.1109/TCYB.2018.2794503. URL: https://ieeexplore.ieee.org/abstract/document/8276629?casa_token=mCY3zdfiFrQAAAAA:VcWAuoYPXEVUhk41NXPiDZJyCOkxwKun7-tl3kflP_hd-dwD_rHnFyuDCIHzEOYkDcOyYU5upEs (visited on 11/18/2024).

[28] Raphael T. Haftka. "Combining global and local approximations". In: *AIAA Journal* 29.9 (Sept. 1991). Publisher: American Institute of Aeronautics and Astronautics, pp. 1523–1525. ISSN: 0001-1452. DOI: 10.2514/3.10768. URL: https://arc.aiaa.org/doi/10.2514/3.10768 (visited on 12/03/2024).

[29] W. Q. Hao et al. "A physics-informed machine learning approach for notch fatigue evaluation of alloys used in aerospace". In: *International Journal of Fatigue* 170 (May 2023), p. 107536. ISSN: 0142-1123. DOI: 10.1016/j.ijfatigue.2023.107536. URL: https://www.sciencedirect.com/science/article/pii/S0142112323000373 (visited on 11/19/2024).

[30] Ali Hashemi, Jinwoo Jang, and Javad Beheshti. "A Machine Learning-Based Surrogate Finite Element Model for Estimating Dynamic Response of Mechanical Systems". In: *IEEE Access* 11 (2023). Conference Name: IEEE Access, pp. 54509–54525. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3282453. URL: https://ieeexplore.ieee.org/abstract/document/10143077 (visited on 10/11/2024).

[31] Russel C. Hibbeler. *Mechanics of Materials*. 10th ed. Pearson, 2017.

[32] Chunlai Hou et al. "A predicting method for the mechanical property response of the marine riser based on the simulation and data-driven models". In: *Ocean Engineering* 293 (Feb. 2024), p. 116612. ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2023.116612. URL: https://www.sciencedirect.com/science/article/pii/S0029801823029967 (visited on 10/10/2024).

[33] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential Model-Based Optimization for General Algorithm Configuration". en. In: *Learning and Intelligent Optimization*. Ed. by Carlos A. Coello Coello. Berlin, Heidelberg: Springer, 2011, pp. 507–523. ISBN: 978-3-642-25566-3. DOI: 10.1007/978-3-642-25566-3_40.

[34] Gareth James et al. *An introduction to statistical learning: with applications in R*. Springer Publishing, 2013.

[35] R. Jin, W. Chen, and T.W. Simpson. "Comparative studies of metamodelling techniques under multiple modelling criteria". en. In: *Structural and Multidisciplinary Optimization* 23.1 (Dec. 2001), pp. 1–13. ISSN: 1615-1488. DOI: 10.1007/s00158-001-0160-4. URL: https://doi.org/10.1007/s00158-001-0160-4 (visited on 12/02/2024).

[36] Chandrika Kamath. "Intelligent sampling for surrogate modeling, hyperparameter optimization, and data analysis". In: *Machine Learning with Applications* 9 (Sept. 2022), p. 100373. ISSN: 2666-8270. DOI: 10.1016/j.mlwa.2022.100373. URL: https://www.sciencedirect.com/science/article/pii/S2666827022000640 (visited on 11/07/2024).

[37] MC Kennedy and A O'Hagan. "Predicting the output from a complex computer code when fast approximations are available". In: *Biometrika* 87.1 (Mar. 2000), pp. 1–13. ISSN: 0006-3444. DOI: 10.1093/biomet/87.1.1. URL: https://doi.org/10.1093/biomet/87.1.1 (visited on 11/28/2024).

[38] Jakub Kudela and Radomil Matousek. "Recent advances and applications of surrogate models for finite element method computations: a review". en. In: *Soft Computing* 26.24 (Dec. 2022), pp. 13709–13733. ISSN: 1433-7479. DOI: 10.1007/s00500-022-07362-8. URL: https://doi.org/10.1007/s00500-022-07362-8 (visited on 10/14/2024).

[39] Xiaonan Lai et al. "A multi-fidelity surrogate model based on design variable correlations". In: *Advanced Engineering Informatics* 59 (Jan. 2024), p. 102248. ISSN: 1474-0346. DOI: 10.1016/j.aei.2023.102248. URL: https://www.sciencedirect.com/science/article/pii/S1474034623003762 (visited on 12/03/2024).

[40] P. Lancaster and K. Salkauskas. "Surfaces generated by moving least squares methods". en. In: *Mathematics of Computation* 37.155 (1981), pp. 141–158. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-1981-0616367-1. URL: https://www.ams.org/mcom/1981-37-155/S0025-5718-1981-0616367-1/ (visited on 12/04/2024).

[41] Haiquan Li, Qinghui Zhang, and Xiaoqian Chen. "Deep Learning-Based Surrogate Model for Flight Load Analysis". In: *Computer Modeling in Engineering & Sciences* 128.2 (Aug. 2021), pp. 605–621. DOI: 10.32604/cmes.2021.015747.

[42] Qilin Li et al. "Machine learning prediction of structural dynamic responses using graph neural networks". In: *Computers & Structures* 289 (Dec. 2023), p. 107188. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2023.107188. URL: https://www.sciencedirect.com/science/article/pii/S0045794923002183 (visited on 11/22/2024).

[43] Xueren Li et al. "Evaluation of supervised machine learning regression models for CFD-based surrogate modelling in indoor airflow field reconstruction". In: *Building and Environment* 267 (Jan. 2025), p. 112173. ISSN: 0360-1323. DOI: 10.1016/j.buildenv.2024.112173. URL: https://www.sciencedirect.com/science/article/pii/S0360132324010151 (visited on 12/10/2024).

[44] Yinqi Li et al. "Optimisation of transmission towers under multiple load cases and constraint conditions with the KSM-GA method". en. In: *Advances in Mechanical Engineering* 15.6 (June 2023). Publisher: SAGE Publications, p. 16878132231183764. ISSN: 1687-8132. DOI: 10.1177/16878132231183764. URL: https://doi.org/10.1177/16878132231183764 (visited on 10/17/2024).

[45] Dehao Liu, Pranav Pusarla, and Yan Wang. "Multifidelity Physics-Constrained Neural Networks With Minimax Architecture". In: *Journal of Computing and Information Science in Engineering* 23.031008 (Dec. 2022). ISSN: 1530-9827. DOI: 10.1115/1.4055316. URL: https://doi.org/10.1115/1.4055316 (visited on 12/04/2024).

[46] Jiaqi Liu et al. "Knowledge embedding synchronous surrogate modeling for multi-objective operational reliability evaluation of complex mechanical systems". In: *Computers & Industrial Engineering* 196 (Oct. 2024), p. 110482. ISSN: 0360-8352. DOI: 10.1016/j.cie.2024.110482. URL: https://www.sciencedirect.com/science/article/pii/S036083522400603X (visited on 11/22/2024).

[47] Jie Liu et al. "Net-HDMR Metamodeling Method for High-Dimensional Problems". In: *Journal of Mechanical Design* 145.091706 (June 2023). ISSN: 1050-0472. DOI: 10.1115/1.4062669. URL: https://doi.org/10.1115/1.4062669 (visited on 11/18/2024).

[48] Christian Málaga-Chuquitaype. "Machine Learning in Structural Design: An Opinionated Review". English. In: *Frontiers in Built Environment* 8 (Feb. 2022). Publisher: Frontiers. ISSN: 2297-3362. DOI: 10.3389/fbuil.2022.815717. URL: https://www.frontiersin.org/journals/built-environment/articles/10.3389/fbuil.2022.815717/full (visited on 10/03/2024).

[49] Levi D. McClenny and Ulisses M. Braga-Neto. "Self-adaptive physics-informed neural networks". In: *Journal of Computational Physics* 474 (Feb. 2023), p. 111722. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2022.111722. URL: https://www.sciencedirect.com/science/article/pii/S0021999122007859 (visited on 11/19/2024).

[50] Douglas C. Montgomery. *Design and Analysis of Experiments*. en. Google-Books-ID: Py7bDgAAQBAJ. John Wiley & Sons, 2017. ISBN: 978-1-119-11347-8.

[51] David Moreno-Salinas et al. "Modelling of a surface marine vehicle with kernel ridge regression confidence machine". In: *Applied Soft Computing* 76 (Mar. 2019), pp. 237–250. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2018.12.002. URL: https://www.sciencedirect.com/science/article/pii/S1568494618306847 (visited on 01/14/2025).

[52] Iván Negrin, Moacir Kripka, and Víctor Yepes. "Metamodel-assisted design optimization in the field of structural engineering: A literature review". In: *Structures* 52 (June 2023), pp. 609–631. ISSN: 2352-0124. DOI: 10.1016/j.istruc.2023.04.006. URL: https://www.sciencedirect.com/science/article/pii/S2352012423004782 (visited on 10/14/2024).

[53] Mehdi Nourbakhsh, Javier Irizarry, and John Haymaker. "Generalizable surrogate model features to approximate stress in 3D trusses". In: *Engineering Applications of Artificial Intelligence* 71 (May 2018), pp. 15–27. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2018.01.006. URL: https://www.sciencedirect.com/science/article/pii/S095219761830006X (visited on 10/17/2024).

[54] Manolis Papadrakakis, Nikos D. Lagaros, and Yiannis Tsompanakis. "Structural optimization using evolution strategies and neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 156.1 (Apr. 1998), pp. 309–333. ISSN: 0045-7825. DOI: 10.1016/S0045-7825(97)00215-6. URL: https://www.sciencedirect.com/science/article/pii/S0045782597002156 (visited on 12/02/2024).

[55] Fabio Parisi et al. "On the use of mechanics-informed models to structural engineering systems: Application of graph neural networks for structural analysis". In: *Structures* 59 (Jan. 2024), p. 105712. ISSN: 2352-0124. DOI: 10.1016/j.istruc.2023.105712. URL: https://www.sciencedirect.com/science/article/pii/S2352012423018003 (visited on 10/16/2024).

[56] Ben Parsonage and Christie Maddock. "A multi-fidelity model management framework for multi-objective aerospace design optimisation". English. In: *Frontiers in Aerospace Engineering* 2 (Feb. 2023). Publisher: Frontiers. ISSN: 2813-2831. DOI: 10.3389/fpace.2023.1046177. URL: https://www.frontiersin.org/journals/aerospace-engineering/articles/10.3389/fpace.2023.1046177/full (visited on 11/22/2024).

[57] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[58] Pero Prebeg, Vedran Zanic, and Bozo Vazic. "Application of a surrogate modeling to the ship structural design". In: *Ocean Engineering* 84 (July 2014), pp. 259–272. ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2014.03.032. URL: https://www.sciencedirect.com/science/article/pii/S0029801814001279 (visited on 10/17/2024).

[59] Nestor V. Queipo et al. "Surrogate-based analysis and optimization". In: *Progress in Aerospace Sciences* 41.1 (Jan. 2005), pp. 1–28. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2005.02.001. URL: https://www.sciencedirect.com/science/article/pii/S0376042105000102 (visited on 11/26/2024).

[60] LLoyd's Register. *Code for Lifting Appliances in a Marine Environment | LR*. en. URL: https://www.lr.org/en/knowledge/lloyds-register-rules/code-for-lifting-appliances-in-a-marine-environment/.

[61] Delbaz Samadian, Imrose B. Muhit, and Nashwan Dawood. "Application of Data-Driven Surrogate Models in Structural Engineering: A Literature Review". en. In: *Archives of Computational Methods in Engineering* (July 2024). ISSN: 1886-1784. DOI: 10.1007/s11831-024-10152-0. URL: https://doi.org/10.1007/s11831-024-10152-0 (visited on 10/10/2024).

[62] Wilhelmus H. A. Schilders et al., eds. *Model Order Reduction: Theory, Research Aspects and Applications*. en. Vol. 13. Mathematics in Industry. Berlin, Heidelberg: Springer, 2008. ISBN: 978-3-540-78840-9 978-3-540-78841-6. DOI: 10.1007/978-3-540-78841-6. URL: http://link.springer.com/10.1007/978-3-540-78841-6 (visited on 12/04/2024).

[63] Maolin Shi et al. "A multi-fidelity surrogate model based on support vector regression". en. In: *Structural and Multidisciplinary Optimization* 61.6 (June 2020), pp. 2363–2375. ISSN: 1615-1488. DOI: 10.1007/s00158-020-02522-6. URL: https://doi.org/10.1007/s00158-020-02522-6 (visited on 12/04/2024).

[64] Xueguan Song et al. "A radial basis function-based multi-fidelity surrogate model: exploring correlation between high-fidelity and low-fidelity models". en. In: *Structural and Multidisciplinary Optimization* 60.3 (Sept. 2019), pp. 965–981. ISSN: 1615-1488. DOI: 10.1007/s00158-019-02248-0. URL: https://doi.org/10.1007/s00158-019-02248-0 (visited on 12/04/2024).

[65] Han Sun, Henry V. Burton, and Honglan Huang. "Machine learning applications for building structural design and performance assessment: State-of-the-art review". In: *Journal of Building Engineering* 33 (Jan. 2021), p. 101816. ISSN: 2352-7102. DOI: 10.1016/j.jobe.2020.101816. URL: https://www.sciencedirect.com/science/article/pii/S2352710220334495 (visited on 10/15/2024).

[66] Amey Thakur and Archit Konde. "Fundamentals of Neural Networks". In: *International Journal for Research in Applied Science and Engineering Technology* 9 (Aug. 2021), pp. 407–426. DOI: 10.22214/ijraset.2021.37362.

[67] Li-Wei Tsai and Alice Alipour. "Physics-informed long short-term memory networks for response prediction of a wind-excited flexible structure". In: *Engineering Structures* 275 (Jan. 2023), p. 114968. ISSN: 0141-0296. DOI: 10.1016/j.engstruct.2022.114968. URL: https://www.sciencedirect.com/science/article/pii/S0141029622010446 (visited on 11/19/2024).

[68] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. en. Google-Books-ID: sna9BaxVbj8C. Springer Science & Business Media, Nov. 1999. ISBN: 978-0-387-98780-4.

[69] Stephen Vasilopoulos, Kendra McTavish, and Ahmed Elshaer. "Governing Lateral Load on Tall Buildings in Canadian Regions". en. In: *Sustainability* 16.16 (Jan. 2024). Number: 16 Publisher: Multidisciplinary Digital Publishing Institute, p. 6757. ISSN: 2071-1050. DOI: 10.3390/su16166757. URL: https://www.mdpi.com/2071-1050/16/16/6757 (visited on 02/18/2025).

[70] Felipe A. C. Viana, Christian Gogu, and Tushar Goel. "Surrogate modeling: tricks that endured the test of time and some recent developments". en. In: *Structural and Multidisciplinary Optimization* 64.5 (Nov. 2021), pp. 2881–2908. ISSN: 1615-1488. DOI: 10.1007/s00158-021-03001-2. URL: https://doi.org/10.1007/s00158-021-03001-2 (visited on 11/04/2024).

[71] G. Gary Wang and S. Shan. "Review of Metamodeling Techniques in Support of Engineering Design Optimization". In: *Journal of Mechanical Design* 129.4 (May 2006), pp. 370–380. ISSN: 1050-0472. DOI: 10.1115/1.2429697. URL: https://doi.org/10.1115/1.2429697 (visited on 11/18/2024).

[72] Shuo Wang et al. "A multi-fidelity surrogate model based on moving least squares: fusing different fidelity data for engineering design". en. In: *Structural and Multidisciplinary Optimization* 64.6 (Dec. 2021), pp. 3637–3652. ISSN: 1615-1488. DOI: 10.1007/s00158-021-03044-5. URL: https://doi.org/10.1007/s00158-021-03044-5 (visited on 12/04/2024).

[73] Xiaowei Wang et al. "Machine Learning for Risk and Resilience Assessment in Structural Engineering: Progress and Future Trends". EN. In: *Journal of Structural Engineering* 148.8 (Aug. 2022). Publisher: American Society of Civil Engineers, p. 03122003. ISSN: 1943-541X. DOI: 10.1061/(ASCE)ST.1943-541X.0003392. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29ST.1943-541X.0003392 (visited on 10/16/2024).

[74] Xilu Wang et al. "Recent Advances in Bayesian Optimization". In: *ACM Comput. Surv.* 55.13s (July 2023), 287:1–287:36. ISSN: 0360-0300. DOI: 10.1145/3582078. URL: https://dl.acm.org/doi/10.1145/3582078 (visited on 11/18/2024).

[75] Eamon Whalen and Caitlin Mueller. "Toward Reusable Surrogate Models: Graph-Based Transfer Learning on Trusses". In: *Journal of Mechanical Design* 144.021704 (Sept. 2021). ISSN: 1050-0472. DOI: 10.1115/1.4052298. URL: https://doi.org/10.1115/1.4052298 (visited on 10/10/2024).

[76] David H. Wolpert. "The Supervised Learning No-Free-Lunch Theorems". en. In: *Soft Computing and Industry: Recent Applications*. Ed. by Rajkumar Roy et al. London: Springer, 2002, pp. 25–42. ISBN: 978-1-4471-0123-9. DOI: 10.1007/978-1-4471-0123-9_3. URL: https://doi.org/10.1007/978-1-4471-0123-9_3 (visited on 11/26/2024).

[77] Chen Xu et al. "Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios". In: *Computer Methods in Applied Mechanics and Engineering* 405 (Feb. 2023), p. 115852. ISSN: 0045-7825. DOI: 10.1016/j.cma.2022.115852. URL: https://www.sciencedirect.com/science/article/pii/S0045782522008088 (visited on 11/19/2024).

[78] Raul Yondo, Esther Andrés, and Eusebio Valero. "A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses". In: *Progress in Aerospace Sciences* 96 (Jan. 2018), pp. 23–61. ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2017.11.003. URL: https://www.sciencedirect.com/science/article/pii/S0376042117300611 (visited on 11/07/2024).

[79] Mohsen Zaker Esteghamati and Madeleine M. Flint. "Do all roads lead to Rome? A comparison of knowledge-based, data-driven, and physics-based surrogate models for performance-based early design". In: *Engineering Structures* 286 (July 2023), p. 116098. ISSN: 0141-0296. DOI: 10.1016/j.engstruct.2023.116098. URL: https://www.sciencedirect.com/science/article/pii/S0141029623005126 (visited on 10/15/2024).

[80] Shuai Zhang et al. “Multi-fidelity surrogate model ensemble based on feasible intervals”. en. In: *Structural and Multidisciplinary Optimization* 65.8 (July 2022), p. 212. ISSN: 1615-1488. DOI: 10.1007/s00158-022-03329-3. URL: https://doi.org/10.1007/s00158-022-03329-3 (visited on 12/04/2024).

[81] Yiming Zhang et al. “Multifidelity Surrogate Based on Single Linear Regression”. In: *AIAA Journal* 56.12 (Dec. 2018). Publisher: American Institute of Aeronautics and Astronautics, pp. 4944–4952. ISSN: 0001-1452. DOI: 10.2514/1.J057299. URL: https://arc.aiaa.org/doi/10.2514/1.J057299 (visited on 12/04/2024).

[82] Shun-Peng Zhu et al. “Physics-informed machine learning and its structural integrity applications: state of the art”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 381.2260 (Sept. 2023). Publisher: Royal Society, p. 20220406. DOI: 10.1098/rsta.2022.0406. URL: https://royalsocietypublishing.org/doi/full/10.1098/rsta.2022.0406 (visited on 10/24/2024).

A

Machine learning background

Polynomial Regression
Polynomial regression expands the features by adding the linear combinations as extra features. Then, it fits the model to the data by assigning coefficients to each of the features. Below is an example for a polynomial expansion of two input features for a second-order polynomial.

$$X = [x_1, x_2] \rightarrow X = \left[1, x_1, x_2, x_1 x_2, x_1^2, x_2^2\right] \tag{A.1}$$

The fitted equation would be in the form:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2 \tag{A.2}$$

Let's revert back to simple linear regression to understand this principle. Linear regression uses the Ordinary Least Squares (OLS) method to optimize the fit of a linear function. Suppose we have data x and target variable y. The fitted line will be in the form of Equation A.3.

$$\hat{y} = ax + b \tag{A.3}$$

However, the data contains noise. Therefore, the target variables can be described with Equation A.4.

$$y = ax + b + \epsilon \tag{A.4}$$

Then, the total loss of the fit can be calculated with the loss function $\mathcal{L}$ in Equation A.5.

$$\mathcal{L} = ||y - \hat{y}||^2 \tag{A.5}$$

In terms of regression theory, coefficients a (slope) and b (intercept) would be put in an vector with the regression coefficients (β). Also, x becomes a vector with an additional 1 added to represent the intercept.

$$X = \begin{bmatrix} 1 & x \end{bmatrix}, \beta = \begin{bmatrix} b \\ a \end{bmatrix} \tag{A.6}$$

Then, the solution can be written in matrix form:

$$\hat{y} = X\beta \tag{A.7}$$

In order to find the coefficients a and b, the loss function needs to be minimized.

$$\mathcal{L} = ||y - X\beta||^2 = (y^T y - 2y^T X\beta + \beta^T X^T X\beta) \tag{A.8}$$

The minimum of any function can be found by taking the derivative with respect to the varying parameter and finding the zero crossings:

$$\nabla(\mathcal{L}) = -2X^T y + 2X^T X\beta = 0 \tag{A.9}$$

The solution for beta represents the regression coefficients for the minimized loss function.

$$\beta = \left(X^T X\right)^{-1} X^T y \tag{A.10}$$

However, if $X^T X$ is singular, which might occur when features are highly correlated, the inverse cannot be computed directly. A common solution to prevent instabilities is to use Singular Value Decomposition (SVD).

$$X = U\Sigma V^T \rightarrow \beta = V\Sigma^{-1}U^T y \tag{A.11}$$

There is a significant downside to this method. The matrix Σ is a diagonal matrix containing the singular values of X. But when the input parameters are highly correlated, these values can be zero. This leads to a singular matrix, which is not invertible. A common solution to prevent numerical instabilities is to ignore values that are almost zero.

$$\Sigma^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{\sigma_n} \end{bmatrix} \tag{A.12}$$

If any singular value (σ_i) approaches 0 within a numerical tolerance, the matrix entry is manually set to zero.

Regularization

The most simple form of regression as described above might perform well for simple cases. However, as the problem becomes more complex with a large input space and there is need for a large polynomial degree, polynomial regression with simple OLS as solving method is high at risk of overfitting and might become unstable. To solve this issue, there are multiple regularization techniques one can apply. The most common techniques are Ridge, Lasso and Elastic net, which are described in the remainder of this section.

Ridge regularization adds an additional term in the loss function, penalizing large coefficients.

$$\mathcal{L} = ||y - X\beta||^2 + \alpha||\beta||_2^2 \tag{A.13}$$

This function has a closed-form solution for β:

$$\beta = \left(X^T X + \alpha I\right)^{-1} X^T \hat{y} \tag{A.14}$$

Here, $||\beta||^2$ is the L2 norm, and α is the regularization parameter. Once again, SVD is used to solve this function, avoiding the calculation of the inverse.

$$\beta = VS^{-1}U^T y \text{ with } S^{-1} = \left(\Sigma^T\Sigma + \lambda I\right)^{-1}\Sigma^T \tag{A.15}$$

With this new function for β, the entries of the diagonal matrix S^{-1} now include the regularization parameter λ.

$$S_{i,i}^{-1} = \frac{\sigma_i}{\sigma_i^2 + \lambda} \tag{A.16}$$

With the additional term included, singular matrices are prevented. This makes the solution stable. Also, the additional term shrinks the coefficients. A large α leads to coefficients close to zero, making the prediction very smooth. However, α is too large, the model might suffer from underfitting. The opposite is also true: overfitting might occur when the regularization parameter is set too low. The value should be tuned properly with hyperparameter optimization techniques such as cross-validation.

Lasso is another technique that also adds an additional term to the loss function, but uses the L1 norm rather than the L2 norm.

$$\mathcal{L} = ||y - X\beta||^2 + \alpha||\beta||_1 \tag{A.17}$$

Here, $||\beta||_1$ is the L1-norm and α is once again the regularization parameter. As this objective function has no closed-form solution, this has to be solved iteratively using coordinate descent. Lasso assumes sparsity in the coefficients, and performs automatic feature selection by setting some coefficients to zero. Large α results in more coefficients set to zero, whereas a small α might result is the same behaviour as an OLS solver. Then, α might be too small to set coefficients to zero. The regularization parameter α needs to be tuned with hyperparameter optimization techniques. This kind of regularization is useful when one suspects that not all variables are important.

Finally, there exists a regularization technique that tries to combine both Ridge and Lasso regularization by adding both the L1 and L2 norm to the loss function:

$$\mathcal{L} = ||y - X\beta||^2 + \alpha_1||\beta||_1 + \alpha_2||\beta||_2^2 \tag{A.18}$$

This method combines the properties of both Ridge and Lasso, as it is stable and handles co-linearity like Ridge, but performs feature selection like Lasso. This method might be beneficial when the data is high-dimensional, where not all features are important and some features are highly correlated.

Kernel methods

Kernel methods are a family of machine leaning models, which have in common that they all use the so-called 'Kernel trick'. This enables evaluation in higher dimensions without directly computing the higher dimension. The feature matrix X gets transformed to the higher dimension Φ. Then, without fully computing Φ, a Kernel matrix is defined, representing the inner products of the transformed features:

$$K(x_i, x_j) = \left\langle (\varphi(x_i), \varphi(x_j)) \right\rangle \tag{A.19}$$

Here, x_i and x_j represent two different data points in the original feature space. The kernel matrix represents the similarities of each possible pair of data points - commonly referred to as pairwise similarities. With this trick, the relations of the data points in the higher dimension can be found without directly computing the higher dimension. This significantly reduces the complexity, since directly computing Φ is often expensive and sometimes even infeasible.

There are different types of kernel functions, and it is up to the user to find the best suited kernel for their application. Common kernels are described below.

- Linear Kernel
$$K(x_i, x_j) = x_i^T x_j \tag{A.20}$$
- Polynomial Kernel
$$K(x_i, x_j) = (x_i^T x_j + c)^d \tag{A.21}$$
where:
  - d is the degree of the polynomial.
  - c is an optional constant.
- Radial Basis Function (RBF) Kernel
$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{A.22}$$
where γ controls the spread of the kernel.
- Sigmoid Kernel
$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + c) \tag{A.23}$$
where:
  - γ is a scaling parameter.
  - c is a constant.
- Laplacian Kernel
$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|_1\right) \tag{A.24}$$
Similar to the RBF kernel but uses the L_1 norm instead of L_2.

- Exponential Kernel
$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|\right) \tag{A.25}$$
Similar to the Laplacian kernel but without squaring the norm.
- Rational Quadratic Kernel
$$K(x_i, x_j) = 1 - \frac{\|x_i - x_j\|^2}{\|x_i - x_j\|^2 + c} \tag{A.26}$$
This can be seen as a continuous mixture of RBF kernels with different length scales.
- Matern Kernel (Parameterized by ν)
$$K(x_i, x_j) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\|x_i - x_j\|}{l}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\frac{\|x_i - x_j\|}{l}\right) \tag{A.27}$$
where:
  - K_ν is the modified Bessel function of the second kind.
  - ν controls smoothness.

Kernel Ridge Regression (KRR) combines Ridge regression with the kernel trick. The loss function from Ridge regression can be found below.

$$\mathcal{L} = ||y - X\beta||^2 + \alpha||\beta||_2^2 \tag{A.28}$$

This function has a closed-form solution for β:

$$\beta = \left(X^T X + \alpha I\right)^{-1} X^T \hat{y} \tag{A.29}$$

In KRR, the feature matrix X gets replaced by the matrix of transformed features (Φ).

$$\mathcal{L} = ||y - \Phi\beta||^2 + \alpha||\beta||_2^2 \tag{A.30}$$

This function has a closed-form solution for β. Then, with the kernel method, XX^T gets replaced by the kernel $K(x_i, x_j)$, and the model solves for the dual coefficient matrix $\hat{\alpha}$:

$$\hat{\alpha} = (K + \alpha I)^{-1} y \tag{A.31}$$

Here, $\hat{\alpha}$ is the dual coefficient matrix and α is the regularization parameter. Predictions are made with:

$$\hat{y} = K\hat{\alpha} \tag{A.32}$$

The type of transformation depends on the selected kernel, for example polynomial, radial basis function or Sigmoid. It could also be linear, but then it would be mathematically equivalent to Ridge Regression. However, it is solved differently in KRR. The inverse is directly computed, rather than using SVD which Ridge regression uses. This leads to more numerical instabilities in the KRR computation with a linear kernel, so it is advised to use Ridge regression rather than KRR with a linear kernel.

B

Double cross-validation

For double cross-validation (CV), the full data is split into a selected number of folds. This section uses 4 folds as example. Each fold has 25% of the data reserved for testing, and every fold uses a different part for testing (see the black test sets in Figure B.1 for visualization). These folds are used for model evaluation. Then, the training part of that fold gets split again following the same principle. In this example, the train set is split in 3 folds, which is also visualized in Figure B.1 These folds are used for hyperparameter tuning. Every combination of hyperparameters is trained on the training part of this inner fold and evaluated on the test part of this inner fold. The sum of the errors in each fold is then used as an evaluation of that hyperparameter combination. This is performed for every combination, and the best set of hyperparameters is selected as the set that resulted in the lowest sum of errors during tuning. Then, the model with these hyperparameters is tested on the outer test set for model evaluation. This happens for each fold, and the average error of the model represents the final score which can be used for comparison among other models.

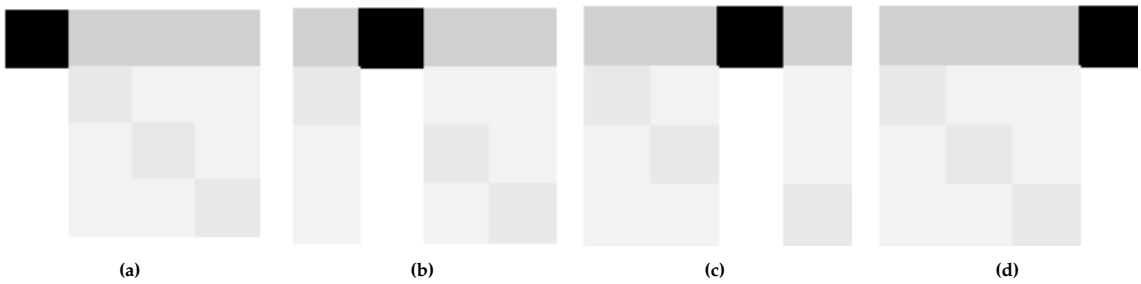


Figure B.1: Model selection approach with 4-fold CV. Model evaluation score based on black test sets, model tuning with 3-fold CV on training part of each fold.

The algorithm for double CV is as follows:

1. For every train and test set of fold k in 4 folds:
   (a) For every hyperparameter combination i:
      i. For every sub-train and sub-test of fold n in 3-fold split of the train set:
         A. Train model on sub-train set
         B. Evaluate model on sub-test set
         C. Save error as $\epsilon_{i,n,k}$
   (b) Save hyperparameter evaluation score as the sum of the errors.

$$\epsilon_{i,k} = \Sigma_{n=1}^{3} \epsilon_{n,i,k} \tag{B.1}$$

2. Select best hyperparameters based on the lowest total error during tuning.
3. For every train and test set of fold k in 4 folds:
   (a) Train the model with these hyperparameters on the train set of fold k
   (b) Evaluate on the test set of fold k. Save the error as ϵ_k
4. Find the average error of all outer folds and report that as the model evaluation error.

$$\epsilon = \frac{1}{4}\Sigma_{k=1}^{4}\epsilon_k \tag{B.2}$$

Fold outer	Hyperparameters	Fold inner	Performance $\epsilon_{n,i,k}$
1	$\alpha = 1$	1	$\epsilon_{1,1,1}$
		2	$\epsilon_{2,1,1}$
		3	$\epsilon_{3,1,1}$
	$\alpha = 100$	1	$\epsilon_{1,2,1}$
		2	$\epsilon_{2,2,1}$
		3	$\epsilon_{3,2,1}$
2	$\alpha = 1$	1	$\epsilon_{1,1,2}$
		2	$\epsilon_{2,1,2}$
		3	$\epsilon_{3,1,2}$
	$\alpha = 100$	1	$\epsilon_{1,2,2}$
		2	$\epsilon_{2,2,2}$
		3	$\epsilon_{3,2,2}$

Fold k	1						2					
HP i	$\alpha = 1$			$\alpha = 100$			$\alpha = 1$			$\alpha = 100$		
Fold n	1	2	3	1	2	3	1	2	3	1	2	3
Error $\epsilon_{n,i,k}$	$\epsilon_{1,1,1}$	$\epsilon_{2,1,1}$	$\epsilon_{3,1,1}$	$\epsilon_{1,2,1}$	$\epsilon_{2,2,1}$	$\epsilon_{3,2,1}$	$\epsilon_{1,1,2}$	$\epsilon_{2,1,2}$	$\epsilon_{3,1,2}$	$\epsilon_{1,2,2}$	$\epsilon_{2,2,2}$	$\epsilon_{3,2,2}$
CV error $\epsilon_{i,k}$	$\epsilon_{1,1} = \Sigma_{n=1}^{3}\epsilon_{n,1,1}$			$\epsilon_{2,1} = \Sigma_{n=1}^{3}\epsilon_{n,2,1}$			$\epsilon_{1,2} = \Sigma_{n=1}^{3}\epsilon_{n,1,2}$			$\epsilon_{2,2} = \Sigma_{n=1}^{3}\epsilon_{n,2,2}$		

Fold k	HP	HP error	Model error
1	$\alpha = 1$	$\epsilon_{1,1}$	ϵ
	$\alpha = 100$	$\epsilon_{2,1}$	ϵ
2	$\alpha = 1$	$\epsilon_{1,2}$	ϵ
	$\alpha = 100$	$\epsilon_{2,2}$	ϵ

The model with the lowest error during hyperparameter tuning (HP error) will be selected and reported with corresponding model error ϵ.

C

Load case tables

		Normal operation General / PT				Emergency operation people transfer				
Parameter	Unit	NO SE	NO AH	NO MC	NO PT	EC SE	EC 3P	EC UB	EC EL	EC PT CF
DNV states (see section 4.2.1)		2)	2)	1)	1)	3)	3)	3)	3)	3)
DNV acceptance criteria		I & II[1]	I & II[1]	I & II[1]	I & II[1]	III	III	III	III	III
Safety factor [DNV]		1.5 & 1.33	1.5 & 1.33	1.5 & 1.33	1.5 & 1.33	1.1	1.1	1.1	1.1	1.1
System state	[-]	Manual	Adjust height	Engaged	Free Float	Manual	Free Float	Manual	Engaged	M
Tip connected	Y/N	N	N	N	Y	N	Y	Y	N	N
Luffing angle range	[°]	-20°/+45°	-17°/+17°	-15°/+40°	-17°/+17°	-20°/+45°	-17°/+17°	-17°/+17°	-17° / +17°	-17° / +17°
Slewing range	[°]	±130°	±130°	±130°	±47.5°	±130°	±47.5°	±130°	±47.5°	±47.5°
Cursor height	[m]	Max	Max	Max	Max	Max	Max	Max	Max	Max
Telescopic stroke	[m]	0 - 8.8	0 - 4		2.3 - 8.8	8.8 - 11	2.3 - 8.8	11	11	2.3 - 8.8
People / SWL / Trolley on Tip	[kg]	0	0	0	120	0	0	0	120	120
People / Trolley on mid	[kg]	0	0	0	120	0	360	360 or 750+120	120	120
People / Trolley on Transfer Deck	[kg]	3000	3000	3000	3000	3000	3000	3000	3000	3000
People / Trolley on Balcony	[kg]	1200	0	1200	1200	1200	1200	1200	1200	1200
Live load factor people	[-]	2	2	2	2	2	2	2	2	2
Dynamic factor (DF) section 4.1.3	[-]	1	1.07	1	1	1	1	1	1	1
Wind speed	[m/s]	20	20	20	20	20	20	20	20	20
Accelerations due to vessels motions										
Tip compensating	Yes/No	No	No	Yes	Yes	No	Yes	No	Yes	No
x-acceleration of transfer deck	[m/s²]	±3	±3	±3	±3	±3	±3	±3	±3	±3
y-acceleration of transfer deck	[m/s²]	±3	±3	±3	±3	±3	±3	±3	±3	±3
z-acceleration of transfer deck	[m/s²]	±3	±3	±3	±3	±3	±3	±3	±3	±3
Tip residual acceleration (x)	[m/s²]	±3	±3	±0.5	±0.5	±3	±0.5	±3	±3	±3
Tip residual acceleration (y)	[m/s²]	±3	±3	±0.5	±0.5	±3	±0.5	±3	±3	±3
Tip residual acceleration (z)	[m/s²]	±3	±3	±0.5	±1.6	±3	±0.5	±3	±3	±3
Contact Loading at the tip (bumper loads)										
Fx at TB tip	[kN]	-	-	-	-20	-	-20	-20	-	-
Fy at TB tip	[kN]	-	-	-	-	-	-	-	-	-
Fz at TB tip	[kN]	-	-	-	-	-	-	-	-	-
Accelerations due to operational motions										
Slewing acceleration	[°/s²]	±5	0	±4.2	0	±5	0	0	±4.1	±4.3
Luffing acceleration	[°/s²]	±5	0	±4.1	0	±5	0	0	±4.0	±4.2
Telescoping acceleration	[m/s²]	±2	0	±2	0	±2	0	0	±2	±2
Height adjust acceleration	[m/s²]	0	±0.5	0	0	0	0	0	0	0
Torsion angle tip	[°]	0	0	0	±5	0	±5	0	0	0
Heel angle	[°]	±5/2	±5/2	±5/2	±5/2	±5/2	±5/2	±5/2	±5/2	±5/2
Trim angle	[°]	±2/5	±2/5	±2/5	±2/5	±2/5	±2/5	±2/5	±2/5	±2/5
Offlead	[°]	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
Sidelead	[°]	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.

Figure C.1: Source of the load cases when transferring people over the gangway

		Normal operation trolley transfer	Emergency operation trolley transfer			
Parameter	**Unit**	NO TT	EC TF SE	EC TF MC	EC TT EL	EC TT CF
DNV states (see section 4.2.1)		1)	3)	3)	3)	3)
DNV acceptance criteria		I & II[1]	III	III	III	III
Stress factor [DNV]		1.5 & 1.33	1.1	1.1	1.1	1.1
System state	[-]	Free Float	Manual	Engaged	Engaged	
Tip connected	Y/N	Y	Y	Y	N	N
Luffing angle range	[°]	-17°/+17°	-17°/+17°	-17°/+17°	-17° / +17°	-17° / +17°
Slewing range	[°]	±47.5°	±130°	±130°	±47.5°	±47.5°
Cursor height	[m]	Max	Max	Max	Max	Max
Telescopic stroke	[m]	2.3 - 8.8	2.3 - 8.8	2.3 - 8.8	11	2.3 - 8.8
Trolley + people on Tip	[kg]	0	750	750	750+120	750+120
Trolley + people on mid	[kg]	750kg+1pax	0	0	0	0
Trolley on Transfer Deck	[kg]	3000	3000	3000	3000	3000
People / Trolley on Balcony	[kg]	1200	1200	1200	1200	1200
Factor people / trolley	[-]	2	1	1	1	1
Dynamic factor (DF) section 4.1.3	[-]	1	1	1	1	1
Wind speed	[m/s]	20	20	20	20	20
Accelerations due to vessels motions						
Tip compensating	Yes/No	Yes	Yes	Yes	Yes	No
x-acceleration of transfer deck	[m/s^2]	±3	±3	±3	±3	±3
y-acceleration of transfer deck	[m/s^2]	±3	±3	±3	±3	±3
z-acceleration of transfer deck	[m/s^2]	±3	±3	±3	±3	±3
Tip residual acceleration (x)	[m/s^2]	±0.5	±3	±0.5	±3	±3
Tip residual acceleration (y)	[m/s^2]	±0.5	±3	±0.5	±3	±3
Tip residual acceleration (z)	[m/s^2]	±1.6	±3	±0.5	±3	±3
Contact Loading at the tip (bumper loads)						
Fx at TB tip	[kN]	-20	-20	-20	-	-
Fy at TB tip	[kN]	-	-	-	-	-
Fz at TB tip	[kN]	-7.36	-7.36	-7.36	-	-
Accelerations due to operational motions						
Slewing acceleration	[$°/s^2$]	0.0	±4.2	±4.2	±4.1	±4.2
Luffing acceleration	[$°/s^2$]	0.0	±4.1	±4.1	±4.0	±4.1
Telescoping acceleration	[m/s^2]	0	±2	±2	±2	±2
Height adjust acceleration	[m/s^2]	0.0	0.0	0.0	0.0	0.0
Torsion angle tip	[°]	±5	0	0	0	0
Heel angle	[°]	±5/2	±5/2	±5/2	±5/2	±5/2
Trim angle	[°]	±2/5	±2/5	±2/5	±2/5	±2/5
Offlead	[°]	n.a.	n.a.	n.a.	n.a.	n.a.
Sidelead	[°]	n.a.	n.a.	n.a.	n.a.	n.a.

Figure C.2: Source of the load cases when transferring a trolley over the gangway

Parameter	Unit	Normal operation Cargo transfer			Emergency operation Cargo transfer					
		NO CT	NO DL	NO PL	EC CT	EC PL	EC CT CF	EC CT SS	EC LR	EC-CTAOPS
DNV case (see section 4.2.1)		I & II[1]	I & II[1]	I & II[1]	III	III	III	III	III	III
Stress factor [DNV]		1.5 & 1.33	1.5 & 1.33	1.5 & 1.33	1.1	1.1	1.1	1.1	1.1	1.1
System state	[-]	Engaged CT	Engaged DL	Engaged CT	Engaged CT	Engaged CT	Manual	Manual	Engaged CT	
Luffing angle range	[°]	See Hoist Curve NO ()			-10°/+45°		-5° / + 40°			
Telescopic stroke	[m]		1.3 - 9.8		0 - 11		1.3 - 9.8			
Slewing range	[°]	±130°	±130°	±47.5°	±130°	±47.5°	±130°	±130°	±130°	±130°
Cursor height	[m]	Max	Max	Max	Max	Max	Max	Max	Max	Max
SWL on tip	[kN]				See Hoist Curve EC ()		See Hoist Curve NO ()			Hoist Curve NO * DF * 1.05
People on mid	[kN]	0	0	0	0	0	0	0	0	0
People / Trolley on Transfer Deck	[kg]	120	120	120	120	120	120	120	120	120
People / Trolley on Balcony	[kg]	0	0	0	0	0	0	0	0	0
Live load factor People transfer deck	[-]	1	1	1	1	1	1	1	1	1
Dynamic factor live load (Ψ)	[-]	DF_air	DF_DL	DF_PL	DF_air	DF_PL	DF_air	DF_air	-0.3	1
Wind speed	[m/s]	20	20	20	20	20	20	20	20	20
Accelerations due to vessels motions										
Tip compensating	Yes/No	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes
x-acceleration of transfer deck	[m/s²]	±1	±1	±1	±1	±1	±1	±1	±1	±1
y-acceleration of transfer deck	[m/s²]	±1	±1	±1	±1	±1	±1	±1	±1	±1
z-acceleration of transfer deck	[m/s²]	±1	±1	±1	±1	±1	±1	±1	±1	±1
Tip residual acceleration (x)	[m/s²]	±0.5	±1	±0.5	±1	±0.5	±1	±1	±0.5	±0.5
Tip residual acceleration (y)	[m/s²]	±0.5	±1	±0.5	±1	±0.5	±1	±1	±0.5	±0.5
Tip residual acceleration (z)	[m/s²]	±0.5	±1	±0.5	±1	±0.5	±1	±1	±0.5	±0.5
Contact Loading at the tip (bumper loads)										
Fx at TB tip	[kN]	-	-	-	-		-	-	-	-
Fy at TB tip	[kN]	-	-	-	-		-	-	-	-
Fz at TB tip	[kN]	-	-	-	-		-	-	-	-
Accelerations due to operational motions										
Slewing acceleration	[°/s²]	±5 to ±4.2	0.0	0.0	±5 to ±4.2	0.0	±5 to 4.2	±5 to 4.2	±5 to ±4.2	0[1]
Luffing acceleration	[°/s²]	±5 to ±4.1	±5 to ±4.1	±5 to ±4.1	±5 to ±4.1	±5 to ±4.1	±5 to 4.1	±5 to 4.1	±5 to ±4.1	0[1]
Telescoping acceleration	[m/s²]	±2	0.0	0.0	±2	0.0	±2	±2	0	0[1]
Height adjust acceleration	[m/s²]	0	0	0	0	0	0	0	0	0
Torsion angle tip	[°]	0	0	0	0	0	0	0	0	0
Heel angle	[°]	±5/2	±5/2	±5/2	±5/2	±5/2	±5/2	±5/2	±5/2	±5/2
Trim angle	[°]	±2/5	±2/5	±2/5	±2/5	±2/5	±2/5	±2/5	±2/5	±2/5
Gangway mode	M/FF/S	M	M	M	M	M	M	M	M	M
Offlead	[°]	±10	±7	±11	±13	±16	±42[2]	±10	±10	±42
Sidelead	[°]	±19	±16	±18	±23	±26.5	±42[2]	±19	±19	±42

Figure C.3: Source of the load cases when operating as a crane

Parameter	Unit	Stowed Survival											
		SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12
System state	[-]	Stowed											
Luffing angle range	[°]	-17° / +17°											
Slewing range	[°]	-100°/-80 & +100/+80°											
Telescopic stroke	[m]	4											
Wind speed	[m/s]	47.8											
x-acceleration of vessel	[m/s^2]	0	0	0	0	4.91	-4.91	4.91	-4.91	2.94	-2.94	2.94	-2.94
y-acceleration of vessel	[m/s^2]	8.83	-8.83	8.83	-8.83	0	0	0	0	5.30	-5.30	5.30	-5.30
z-acceleration of vessel (additional to gravity)	[m/s^2]	-9.81	-9.81	9.81	9.81	-9.81	-9.81	9.81	9.81	-9.81	-9.81	9.81	9.81
Heel angle	[°]	0	0	0	0	0	0	0	0	0	0	0	0
Trim angle	[°]	0	0	0	0	0	0	0	0	0	0	0	0

Figure C.4: Source of the load cases in stowed conditions

D

Model input

LoadCase	1
name	XYZ-WIND
abr	XYZ-WIND
group	XYZ-WIND
acceleration x-coordinate	[-5, -2.5, 0, 2.5, 5]
acceleration y-coordinate	[-5, -2.5, 0, 2.5, 5]
acceleration z-coordinate	[-5, -2.5, 0, 2.5, 5]
tipForce x-direction [N]	0
tipForce y-direction [N]	0
tipForce z-direction [N]	0
tipForce location x-direction (wrt luffing hinge)	0
tipForce location y-direction (wrt luffing hinge)	0
tipForce location z-direction (wrt luffing hinge)	0
tipConnected	0
dutyFactor	1
hoistingFactorPeople	1
gangwayExtension	4
heel	0
trim	0
slewingAccel	0
luffingAccel	0
telescopingAcceleration [m/s2]	0
luffingAngle [rad]	0
slewingAngle [rad]	0
windspeed	[0, 5, 10, 15, 20]
windAngle [rad]	0
peopleArray	[[]]
AddOn	[[]]
offleadDisplacement [m]	0
sideleadDisplacement [m]	0
actualCableLength [m]	0
isstowed	0
tipCompensation	0
passiveForceStowed	0
stowedPoseTBoom	0

Figure D.1: Model input for the simple truss model

LoadCase														
name	NO CT	NO SE	EC CT	EC SE	SC1	SC2	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12
abr	NO CT	NO SE	EC CT	EC SE	SC1	SC2	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12
group	NO CT	NO SE	EC CT	EC SE	SC1	SC2	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12
acceleration x-coordinate	[-1, 1]	[-3, 3]	[-1, 1]	[-3, 3]	0	0	4.91	-4.91	4.91	-4.91	2.94	-2.94	2.94	-2.94
acceleration y-coordinate	[-1, 1]	[-3, 3]	[-1, 1]	[-3, 3]	8.83	-8.83	0	0	0	0	5.3	-5.3	5.3	-5.3
acceleration z-coordinate	[-1, 1]	[-3, 3]	[-1, 1]	[-3, 3]	-9.81	-9.81	-9.81	-9.81	9.81	9.81	-9.81	-9.81	9.81	9.81
tipForce x-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipForce y-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipForce z-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipForce location x-direction (wrt luffing hinge)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipForce location y-direction (wrt luffing hinge)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipForce location z-direction (wrt luffing hinge)	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipConnected	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dutyFactor	1	1	1	1	1	1	1	1	1	1	1	1	1	1
hoistingFactorPeople	1	1	1	1	1	1	1	1	1	1	1	1	1	1
gangwayExtension	4	4	4	4	4	4	4	4	4	4	4	4	4	4
heel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
trim	0	0	0	0	0	0	0	0	0	0	0	0	0	0
slewingAccel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
luffingAccel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
telescopingAcceleration [m/s2]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
luffingAngle [rad]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
slewingAngle [rad]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
windspeed	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 47.8]	[0, 47.8]	[0, 47.8]	[0, 47.8]	[0, 47.8]	[0, 47.8]	[0, 47.8]	[0, 47.8]	[0, 47.8]	[0, 47.8]
windAngle [rad]	0	0	0	0	-90	-90	-90	-90	-90	-90	-90	-90	-90	-90
peopleArray	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]
AddOn	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]	[[]]
offleadDisplacement [m]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sideleadDisplacement [m]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
actualCableLength [m]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
isstowed	0	0	0	0	1	1	1	1	1	1	1	1	1	1
tipCompensation	0	0	0	0	0	0	0	0	0	0	0	0	0	0
passiveForceStowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0
stowedPoseTBoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure D.2: Model input for the complex model without point loads

LoadCase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
name	NO CT	NO DL	NO PL	EC CT	EC PL	EC CT CF	EC CT SS	EC-CTAOP	NO SE	NO AH	NO MC	NO PT	EC SE	EC 3P	EC UB	EC EL	EC PT CF	NO TT	EC TF SE	EC TF MC	EC TT EL	EC TT CF	
abr	NO CT	NO DL	NO PL	EC CT	EC PL	EC CT CF	EC CT SS	EC-CTAOP	NO SE	NO AH	NO MC	NO PT	EC SE	EC 3P	EC UB	EC EL	EC PT CF	NO TT	EC TF SE	EC TF MC	EC TT EL	EC TT CF	
group	NO CT	NO DL	NO PL	EC CT	EC PL	EC CT CF	EC CT SS	EC LR	EC-CTAOP	NO SE	NO AH	NO MC	NO PT	EC SE	EC 3P	EC UB	EC EL	EC PT CF	NO TT	EC TF SE	EC TF MC	EC TT EL	EC TT CF
acceleration x-coordinate	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	
acceleration y-coordinate	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	
acceleration z-coordinate	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	[-3, 3]	
tipForce x-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipForce y-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0 [-20, 0]	0	0 [-20, 0]	0 [-20, 0]	0	0	0 [-20, 0]	[-20, 0]	[-20, 0]	0	0	0
tipForce z-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 [-7.36, 0]	[-7.36, 0]	[-7.36, 0]	0	0	0
tipForce location x-direction (wrt luffing hinge)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipForce location y-direction (wrt luffing hinge)	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7
tipForce location z-direction (wrt luffing hinge)	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
tipConnected	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	0	0	0	1	1	1	0	1
dutyFactor	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
hoistingFactorPeople	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1
gangwayExtension	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
heel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
trim	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
slewingAccel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
luffingAccel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
telescopingAcceleration [m/s2]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
luffingAngle [rad]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
slewingAngle [rad]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
windspeed	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]	[0, 20]
windAngle [rad]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
peopleArray	[[7]]	[[7]]	[[7]]	[[7]]	[[7]]	[[7]]	[[7]]	[[7]]	[[0]]	[[]]	[[0]]	[[0, 3]]	[[0]]	[[0, 4]]	[[0, 4], [0, [0, 3]]	[[0, 3]]	[[0, 3]]	[[0, 3]]	[[0]]	[[0]]	[[0, 3]]	[[0, 3]]	[[0, 3]]
AddOn	[[13]]	[[6]]	[[14]]	[[6]]	[[14]]	[[14]]	[[9]]	[[10]]	[[5]]	[[5]]	[[5]]	[[5]]	[[5]]	[[5]]	[[]]	[[5]]	[[5]]	[[5]]	[[4, 5]]	[[4, 5]]	[[4, 5]]	[[4, 5]]	[[4, 5]]
offLeadDisplacement [m]	[-10, 10]	[-7, 7]	[-11, 11]	[-13, 13]	[-16, 16]	[-42, 42]	[-10, 10]	[-42, 42]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sideLeadDisplacement [m]	[-19, 19]	[-16, 16]	[-18, 18]	[-23, 23	[-26.5, 26	[42, 42]	[-19, 19]	[42, 42]	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
actualCableLength [m]	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
isstowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
tipCompensation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
passiveForceStowed	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
stowedPoseTBoom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

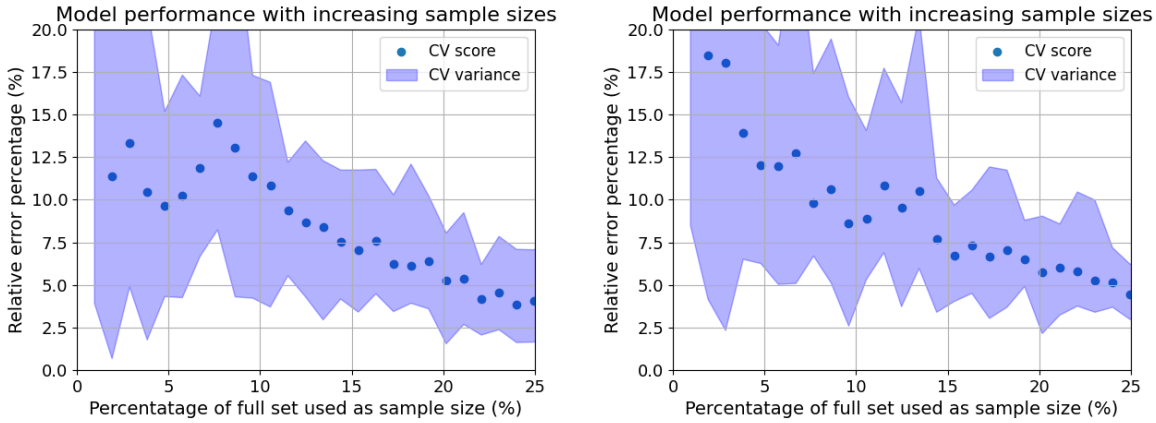
Figure D.3: Model input for the complex model - part - In operation

LoadCase	24	25	26	27	28	29	30	31	32	33	34	35
name	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12
abr	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12
group	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12
acceleration x-coordinate	0	0	0	0	4.91	-4.91	4.91	-4.91	2.94	-2.94	2.94	-2.94
acceleration y-coordinate	8.83	-8.83	8.83	-8.83	0	0	0	0	5.3	-5.3	5.3	-5.3
acceleration z-coordinate	-9.81	-9.81	9.81	9.81	-9.81	9.81	9.81	9.81	-9.81	9.81	9.81	9.81
tipForce x-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0
tipForce y-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0
tipForce z-direction [N]	0	0	0	0	0	0	0	0	0	0	0	0
tipForce location x-direction (wrt luffing hinge)	0	0	0	0	0	0	0	0	0	0	0	0
tipForce location y-direction (wrt luffing hinge)	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7
tipForce location z-direction (wrt luffing hinge)	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
tipConnected	0	0	0	0	0	0	0	0	0	0	0	0
dutyFactor	1	1	1	1	1	1	1	1	1	1	1	1
hoistingFactorPeople	2	2	2	2	2	2	2	2	2	2	2	2
gangwayExtension	4	4	4	4	4	4	4	4	4	4	4	4
heel	0	0	0	0	0	0	0	0	0	0	0	0
trim	0	0	0	0	0	0	0	0	0	0	0	0
slewingAccel	0	0	0	0	0	0	0	0	0	0	0	0
luffingAccel	0	0	0	0	0	0	0	0	0	0	0	0
telescopingAcceleration [m/s2]	0	0	0	0	0	0	0	0	0	0	0	0
luffingAngle [rad]	0	0	0	0	0	0	0	0	0	0	0	0
slewingAngle [rad]	-90	-90	-90	-90	-90	-90	-90	-90	-90	-90	-90	-90
windspeed	47.8	47.8	47.8	47.8	47.8	47.8	47.8	47.8	47.8	47.8	47.8	47.8
windAngle [rad]	-90	-90	-90	-90	-90	-90	-90	-90	-90	-90	-90	-90
peopleArray	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]
AddOn	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]
workspaceStructural	0	0	0	0	0	0	0	0	0	0	0	0
workspaceHydraulic	0	0	0	0	0	0	0	0	0	0	0	0
operationalMode	0	0	0	0	0	0	0	0	0	0	0	0
offLeadDisplacement [m]	0	0	0	0	0	0	0	0	0	0	0	0
sideLeadDisplacement [m]	0	0	0	0	0	0	0	0	0	0	0	0
actualCableLength [m]	0	0	0	0	0	0	0	0	0	0	0	0
isStowed	1	1	1	1	1	1	1	1	1	1	1	1
tipCompensation	0	0	0	0	0	0	0	0	0	0	0	0
residual acc. x-coordinate	0	0	0	0	0	0	0	0	0	0	0	0
residual acc. y-coordinate	0	0	0	0	0	0	0	0	0	0	0	0
residual acc. z-coordinate	0	0	0	0	0	0	0	0	0	0	0	0
slewingDrives_no	0	0	0	0	0	0	0	0	0	0	0	0
telescopingDrives_no	0	0	0	0	0	0	0	0	0	0	0	0
luffingCylinders_no	0	0	0	0	0	0	0	0	0	0	0	0
passiveForceStowed	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]
stowedPoseTBoom	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]

Figure D.4: Model input for the complex model - Stowed

E

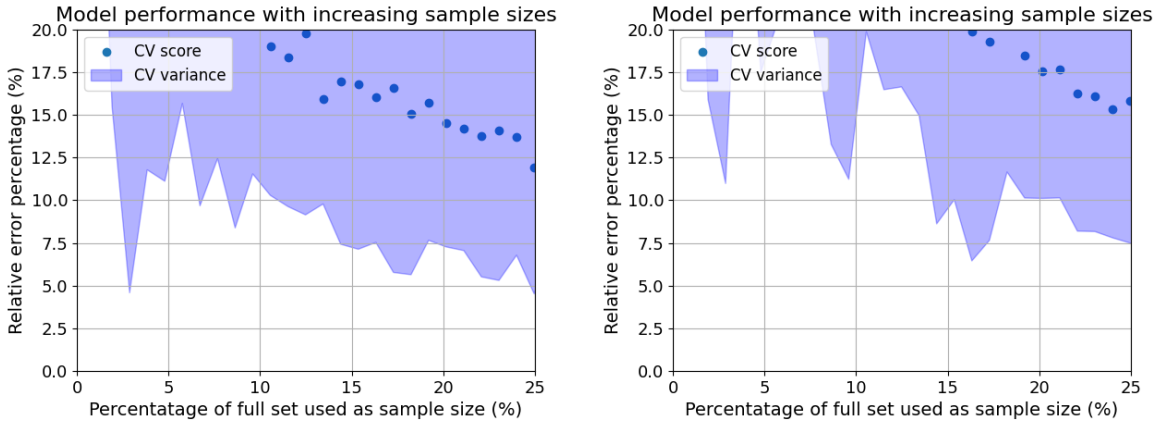
Intelligent sampling results



(a) Resampling randomly

(b) Resampling Euclidean distance-based

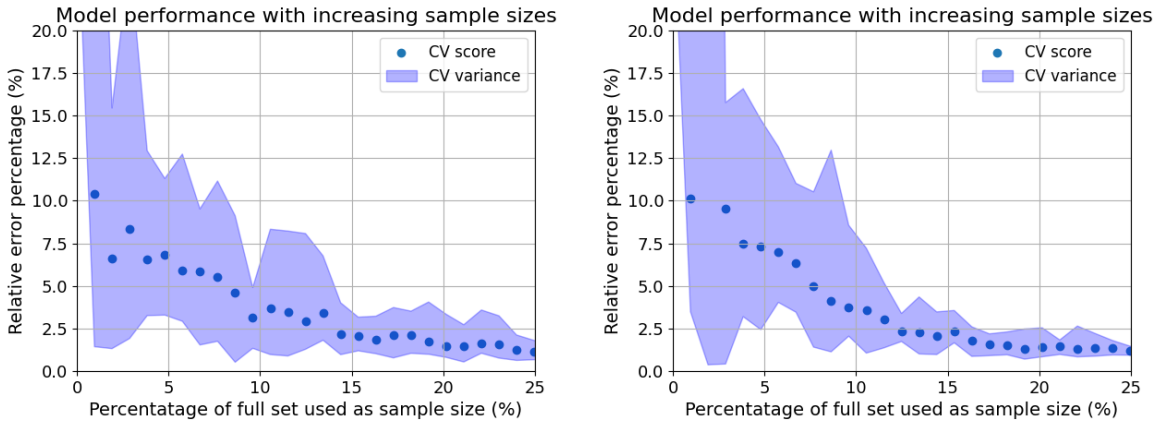
Figure E.1: Sampling randomly (left) and sampling based on Euclidean distances (right) - XGBoost, simple truss model, HF only



(a) Resampling randomly

(b) Resampling Euclidean distance-based

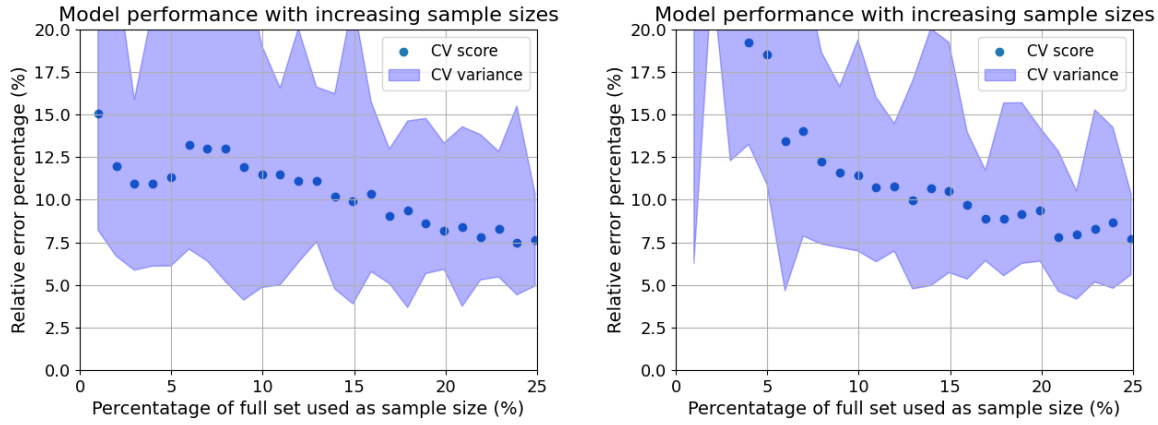
Figure E.2: Sampling randomly (left) and sampling based on Euclidean distances (right) - KRR, simple truss model, HF only



(a) Resampling randomly

(b) Resampling Euclidean distance-based

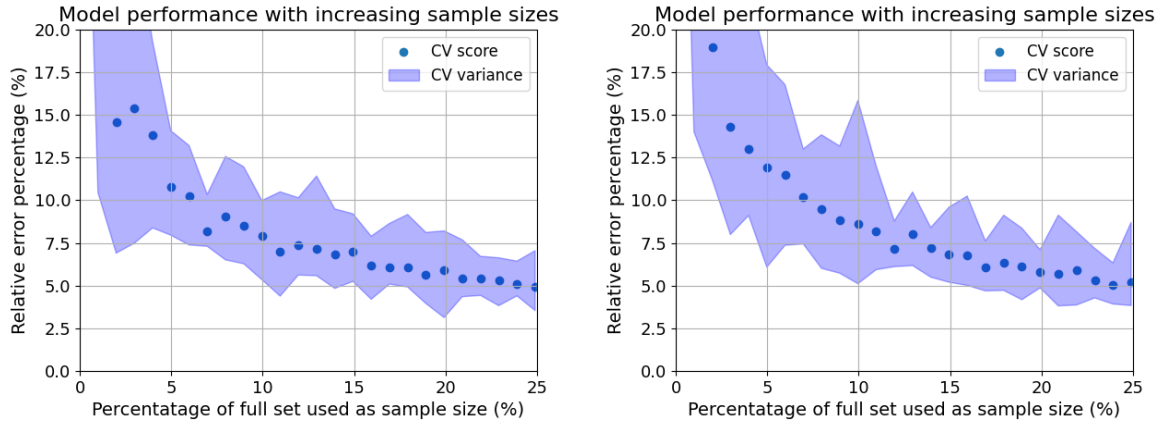
Figure E.5: Sampling randomly (left) and sampling based on Euclidean distances (right) - XGBoost, simple truss model, multi-fidelity



(a) Resampling randomly

(b) Resampling Euclidean distance-based

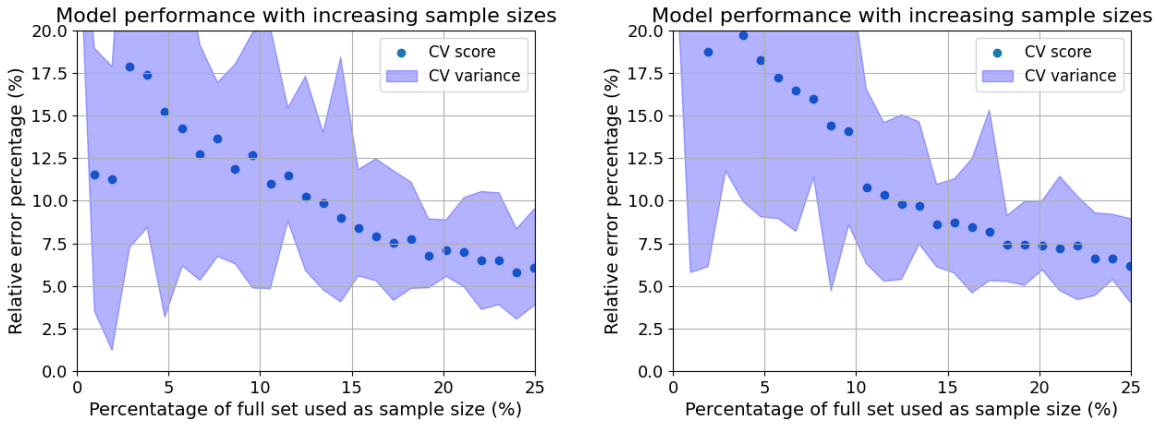
Figure E.3: Sampling randomly (left) and sampling based on Euclidean distances (right) - KRR, complex model, HF only



(a) Resampling randomly

(b) Resampling Euclidean distance-based

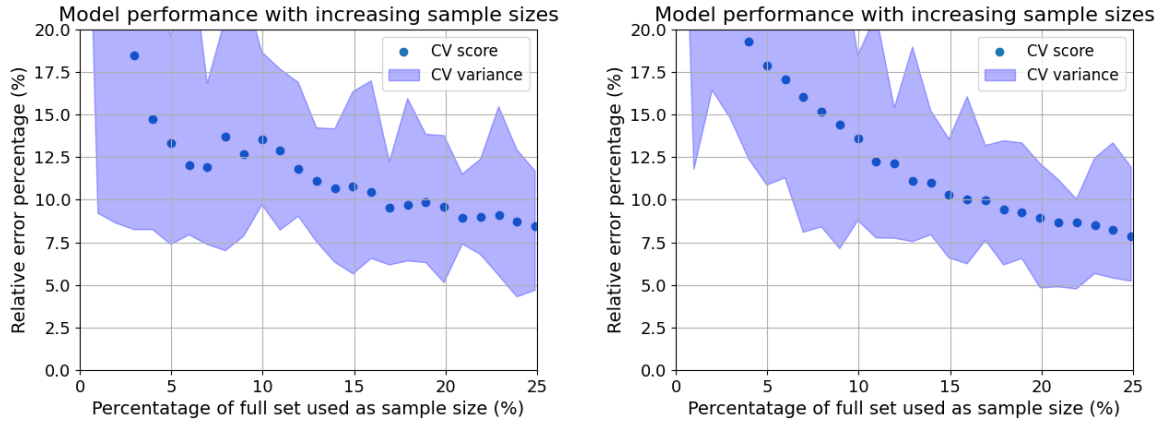
Figure E.4: Sampling randomly (left) and sampling based on Euclidean distances (right) - XGBoost, complex model, multi-fidelity



(a) Resampling randomly

(b) Resampling Euclidean distance-based

Figure E.6: Sampling randomly (left) and sampling based on Euclidean distances (right) - KRR, simple truss model, multi-fidelity



(a) Resampling randomly

(b) Resampling Euclidean distance-based

Figure E.7: Sampling randomly (left) and sampling based on Euclidean distances (right) - KRR, complex model, multi-fidelity