COMPARISON OF ARTIFICIAL NEURAL NETWORK-BASED AND ADAPTIVE QUADRATIC NEURAL NETWORK-BASED MULTI-FIDELITY ALGORITHMS FOR BUCKLING LOAD PREDICTION OF STIFFENED PANELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HÜSEYİN AVNİ YAŞAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENTIFIC COMPUTING

APRIL 2023

Approval of the thesis:

COMPARISON OF ARTIFICIAL NEURAL NETWORK-BASED AND ADAPTIVE QUADRATIC NEURAL NETWORK-BASED MULTI-FIDELITY ALGORITHMS FOR BUCKLING LOAD PREDICTION OF STIFFENED PANELS

submitted by **HÜSEYİN AVNİ YAŞAR** in partial fulfillment of the requirements for the degree of **Master of Science in Scientific Computing Department, Middle East Technical University** by,

Prof. Dr. Sevtap Selçuk Kestel
Dean, Graduate School of **Applied Mathematics** ________________

Assoc. Prof. Dr. Önder Türk
Head of Department, **Scientific Computing** ________________

Assoc. Prof. Dr. Ercan Gürses
Supervisor, **Aerospace Engineering, METU** ________________

Assoc. Prof. Dr. Hamdullah Yücel
Co-supervisor, **Scientific Computing, METU** ________________

Examining Committee Members:

Prof. Dr. Altan Kayran
Aerospace Engineering Department, METU ________________

Assoc. Prof. Dr. Ercan Gürses
Aerospace Engineering Department, METU ________________

Assoc. Prof. Dr. Hamdullah Yücel
Scientific Computing Department, METU ________________

Prof. Dr. Erdem Acar
Mechanical Engineering Department, TOBB ETÜ ________________

Assoc. Prof. Dr. Serdar Göktepe
Civil Engineering Department, METU ________________

Date: 14.04.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: HÜSEYİN AVNİ YAŞAR

Signature :

ABSTRACT

COMPARISON OF ARTIFICIAL NEURAL NETWORK-BASED AND ADAPTIVE QUADRATIC NEURAL NETWORK-BASED MULTI-FIDELITY ALGORITHMS FOR BUCKLING LOAD PREDICTION OF STIFFENED PANELS

Yaşar, Hüseyin Avni
M.S., Department of Scientific Computing
Supervisor : Assoc. Prof. Dr. Ercan Gürses
Co-Supervisor : Assoc. Prof. Dr. Hamdullah Yücel

April 2023, 75 pages

This thesis presents a novel approach for predicting the buckling load of stiffened panels using multi-fidelity modeling based on the quadratic neural networks (QNNs) with adaptive activation functions. The effectiveness of the proposed approach is demonstrated through a series of simulations on a range of stiffened panel configurations, and the results are compared to those obtained from traditional multi-fidelity modeling methods in terms of accuracy and computational efficiency. Numerical experiments demonstrate that the model can accurately and efficiently predict the buckling load of stiffened panels, while significantly reducing the computational cost of evaluating the surrogate model. This approach can significantly improve the design and optimization of aerospace structures by easily and quickly exploring various design configurations and finding stable and efficient configurations. Overall, this study highlights the potential of multi-fidelity modeling for predicting the buckling load of aerospace structures, and the effectiveness of using QNNs with the adaptive activation functions.

Keywords: Multi-Fidelity Models, Surrogate Model, Adaptive Activation Functions, Artificial Neural Networks, Buckling, Stiffened Panels

ÖZ

GÜÇLENDİRİLMİŞ PANELLERİN BURKULMA YÜKÜ TAHMİNİ İÇİN YAPAY SİNİR AĞI TABANLI VE ADAPTİF KUADRATİK SİNİR AĞI TABANLI ÇOKLU DOĞRULUK SEVİYELİ ALGORİTMALARIN KARŞILAŞTIRILMASI

Yaşar, Hüseyin Avni
Yüksek Lisans, Bilimsel Hesaplama Bölümü
Tez Yöneticisi : Doç. Dr. Ercan Gürses
Ortak Tez Yöneticisi : Doç. Dr. Hamdullah Yücel

Nisan 2023, 75 sayfa

Bu tez, uyarlanabilir aktivasyon fonksiyonlarına sahip kuadratik sinir ağlarına (KSA-lar) dayalı çoklu doğruluk seviyeli modeller kullanarak güçlendirilmiş panellerin burkulma yükünü tahmin etmek için yeni bir yaklaşım sunmaktadır. Önerilen yaklaşımın etkinliği, çeşitli güçlendirilmiş panel konfigürasyonu üzerinde bir dizi simülasyon yoluyla gösterilmiştir ve sonuçlar, doğruluk ve hesaplama verimliliği açısından geleneksel çok yönlü modelleme yöntemlerinden elde edilenlerle karşılaştırılmıştır. Sayısal deneyler, önerilen modelin güçlendirilmiş panel yapılarının burkulma yüklerini doğru ve etkin bir şekilde tahmin edebileceğini ve vekil modelin değerlendirilmesi için gerekli hesaplama maliyetini önemli ölçüde azaltabileceğini göstermektedir. Bu yaklaşım, çeşitli tasarım konfigürasyonlarını kolay ve hızlı bir şekilde keşfederek ve kararlı ve verimli konfigürasyonlar bularak havacılık yapılarının tasarımını ve optimizasyonunu önemli ölçüde iyileştirebilir. Genel olarak, bu çalışma, havacılık yapılarının burkulma yükünü tahmin etmek için çoklu doğruluk seviyeli modellemenin potansiyelini ve bu amaç için uyarlanabilir aktivasyon fonksiyonlarına sahip KSAları kullanmanın etkinliğini vurgulamaktadır.

Anahtar Kelimeler: Çoklu Doğruluk Modelleri, Vekil Model, Uyarlanabilir Aktivasyon Fonksiyonları, Yapay Sinir Ağları, Burkulma, Güçlendirilmiş Plakalar

To all my loved ones

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my thesis supervisor, Assoc. Prof. Dr. Ercan Gürses and co-supervisor, Assoc. Prof. Dr. Hamdullah Yücel, for their unwavering support, patient guidance, and enthusiastic encouragement throughout the development and preparation of this thesis. Their invaluable advice and willingness to share their experiences have been instrumental in shaping the direction and outcome of this research.

In particular, I am deeply grateful to Assoc. Prof. Dr. Ercan Gürses for his exceptional technical expertise and generous spirit, which he has demonstrated in both technical and non-technical conversations. The time I spent working with him was always productive and fruitful, and his kind and respectful attitude toward me has made this journey a truly enjoyable one.

Besides my advisors, I would like to thank the rest of my thesis committee: Prof. Dr. Altan Kayran, Prof. Dr. Erdem Acar, and Assoc. Prof. Dr. Serdar Göktepe for their insightful comments and contributions.

I wish to express my appreciation to my colleagues at Roketsan Inc. and friends at METU for supporting and motivating me during my thesis studies. I am genuinely grateful for Kemal Gürson's assistance in preparing technical drawings and constant support throughout my research journey. I would also like to thank Roketsan Inc. for the encouragement shown through the graduate program. The numerical calculations reported in this paper were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

Finally, I want to express my deepest gratitude to my amazing family - my wife, Melike, whose love and support kept me grounded and focused; my sister, Zeynep, whose unwavering belief in me gave me the strength to overcome any obstacle; and my wonderful parents, Abdil and Arife, whose love and guidance have been my guiding light throughout this journey. Their support and encouragement mean the world to me, and I could not have achieved this without them.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
DNN	Deep Neural Networks
MLP	Multi Layer Perceptron
QMLP	Quadratic Multi Layer Perceptron
MFM	Multi-Fidelity Model
HFM	High-Fidelity Model
LFM	Low-Fidelity Model
AAF	Adaptive Activation Functions
CAF	Constant Activation Functions
LFDNN	Low-Fidelity Deep Neural Network
HFDNN	High-Fidelity Deep Neural Network
DDNN	Discrepancy Deep Neural Network
MDNN	Multiplicative Deep Neural Network

CHAPTER 1

INTRODUCTION

Buckling, which can lead to sudden and catastrophic failure of a structure, is a critical concern in the design of aerospace structures, see, e.g., [8]. Accurate prediction of the buckling load of a structure is essential for ensuring its stability and safety and is a key challenge in the design of aerospace structures. Traditional methods for estimating the buckling load of a structure, such as finite element analysis, are computationally expensive and time-consuming, making it challenging to examine a wide range of design configurations and loading conditions [39].

In recent years, surrogate modeling has been proposed as a promising approach for addressing this challenge [30]. One type of surrogate modeling that is quite effective is multi-fidelity modeling. Multi-fidelity modeling is a technique that uses multiple models with varying levels of fidelity to approximate complex systems [45]. By combining the predictions of these models, this approach can reduce computational burden and improve prediction accuracy compared to using a single high-fidelity model.

This study proposes a novel approach for estimating the buckling load of stiffened panels using multi-fidelity modeling based on quadratic neural networks (QNNs) with adaptive activation functions. According to our best knowledge, this is the first study in which adaptive activation functions and QNNs have been applied in a multi-fidelity study for predicting buckling load in the stiffened panels. In this approach, using QNNs can enable the model to represent a more comprehensive range of nonlinear functions than the traditional ANNs, and using adaptive activation functions can improve the model's convergence speed and training efficiency.

The effectiveness of the proposed approach will be demonstrated through a series of simulations on a range of stiffened panel configurations. The results obtained using this approach will be compared to those obtained from the traditional multi-fidelity modeling methods in terms of accuracy and computational efficiency. For an in-depth comparison, ANN-based multi-fidelity models with optimal hyperparameters obtained through hyperparameter optimization will be utilized.

The proposed approach can effectively combine multiple fidelity levels to produce more accurate and reliable predictions of the buckling load of stiffened panels while significantly reducing the computational cost of evaluating the surrogate model. This method also allows for more efficient optimization of the panel design and has the potential for application to a wide range of real-world problems. The thesis is organized as follows:

The second chapter provides a literature review on surrogate modeling, multi-fidelity modeling, adaptive activation functions, and quadratic neural networks. This chapter serves as a background and motivation for the work presented in the subsequent chapters.

The third chapter presents a detailed description of the buckling analysis procedures, including the methods and assumptions used to model the buckling behavior of stiffened panels.

The fourth chapter presents an overview of the concept of multi-fidelity modeling and examines the correction methods used to improve the accuracy of multi-fidelity models.

The fifth chapter describes the methodology of our proposed approach in detail, including the algorithm for training and evaluating the model. This chapter comprehensively explains how our approach works and how it can be implemented in practice.

The sixth chapter presents the results of our numerical experiments and compares the performance of our approach to existing methods. This chapter demonstrates the effectiveness of our approach in achieving high accuracy with significantly reduced computational cost. Last, the seventh chapter provides a discussion of the results, the conclusion of this study, and suggestions related to future works.

CHAPTER 2

LITERATURE REVIEW

In engineering and scientific applications, systematically formulating the design problem within an optimization framework to thoroughly exploit the advantages in design is often reasonable. The objective functions can be evaluated by employing analytical functions or, more generally, numerical models such as finite difference, finite volume, or finite element simulations. Since a considerable amount of evaluations are often needed for optimization studies to achieve an optimal solution, the solution process results in high computational costs. In order to reduce the computational time spent on evaluations, global approximation techniques known as metamodels or surrogate models are developed using data acquired from analytical or numerical approaches [54]. These models are cheap to evaluate and can be used as a substitute for high-fidelity simulations by providing significant time savings to execute the optimization.

In the literature, there are several applications of surrogate modeling techniques in the optimization of stiffened panels that are excessively used in the aerospace structures. It has been demonstrated that the time needed to find the maximum buckling load of composite stiffened panels was effectively reduced using surrogate models based on the radial basis functions [25], second-order polynomials [47], and neural networks [7]. In Hao's study [22], an intelligent optimization framework was utilized using convolutional neural networks (CNN)-based surrogate models to enhance the optimization performance of structural panels with curvilinear stiffeners. Bisagni et al. [7] developed an artificial neural network (ANN)-based optimization procedure for obtaining the minimum weight design and load-carrying capacity of stiffened panels subject to buckling and post-buckling. The developed network was trained with non-linear finite element analysis results. Gomes et al. [18] conducted an optimization study using a genetic algorithm on anisotropic laminated composites. In this study, two alternative neural networks, radial basis and multi-layer perceptron neural networks, were used to reduce the computational cost. Both methods were determined to produce satisfactory solutions on an accuracy basis and to reduce the design process time. Lanzi and Giavotto [31] investigated the performance of radial basis functions, neural networks, and kriging surrogate models for the optimization of axially compressed composite stiffened panels with the multi-objective function of maximum post-buckling load and minimum weight goals. According to the results verified by tests, three methods were found to produce acceptable results with similar accuracy, and none of them can be identified as a superior method for this study. In Mallela and Upadhyay [34], a neural network-based optimization tool was developed for forecasting the shear-buckling load of composite stiffened panels under in-plane shear loads. The designed network has been checked for generalization capabilities utilizing test data created with finite element analysis results. Bacarreza and Aliabadi [5] proposed a multi-objective optimization approach for laminated composite panels in a post-buckling regime by considering the progressive failure modes of composites. This approach resulted in the maximization of load bearing capacity of stiffened panels. Another comparison analysis was performed by Kalnins et al. [28] to check the performance of different metamodels. They have utilized radial basis functions, multivariate adaptive regression splines, and polynomials regression metamodels to optimize the post-buckling attributes of curved composite stiffened structures with damages. It was concluded that since the surrogate models used in this study have cross-validation errors of less than 10%, they could be an efficient substitute for physical experiments in an optimization framework.

While the accuracy of metamodels is significantly dependent on the surrogate model choice and problem type, the strategy of the design of experiments (DOE) also has a considerable effect on the performance of metamodels. The sampling approach and size of samples dramatically affect the accuracy of the models, and an inappropriate sampling process might increase the computational costs. Therefore, to select experimental data strategically, the design of experiment methodology should be chosen carefully. It was observed that space-filling methods such as Latin Hypercube Sampling (LHS) were generally used for building datasets for surrogate models [49]. In Olsson's study [40], LHS was used to improve the efficiency of Monte Carlo Simulations by increasing the representation capacity of stochastic finite element analysis. Ding et al. [12] employed LHS to compute fracture probability with fewer simulations and acquired sufficient calculation error. Moreover, Ferrari et al. [16] chose LHS as the sampling method to create the sample set of experimental vibration data used for boosting the confidence level of the global optimization-based automated model for the historic bridge. It was indicated that there is a high correlation between the experiment and analysis results.

Surrogate models have been applied successfully and have reduced the computational cost of various optimization problems. Since the reliability of surrogate models can be determined based on the fidelity of training data, high-fidelity data is needed to yield a surrogate model with high accuracy. However, the computational burden of generating enough experiments to fit reliable metamodels is severe for most large-scale problems. Even conducting a few high-fidelity simulations can be burdensome for complex problems with highly nonlinear behavior. In order to overcome this issue, the multi-fidelity modeling approach is proposed for the optimization of engineering problems. Multi-fidelity models, which are implemented by incorporating metamodels trained with low-fidelity data and metamodels trained with high-fidelity data, can approach the accuracy obtained by high-fidelity metamodels while reducing the computational cost critically.

Zadeh et al. [64] proposed a multi-fidelity method for achieving the collaborative optimization of composite beams at a reasonable cost. High-fidelity simulations had a much higher computing cost when considering design variables, whereas low-fidelity simulations were less accurate despite having a lower computational cost. In this study, the presented multi-fidelity method has used fewer high-fidelity simulations for tuning low-fidelity models to reach the accuracy level of high-fidelity models, while the tuned model is less expensive. Peherstoler et al. [45] surveyed different model management techniques for multi-fidelity models to differentiate how different fidelity models are employed and combined during the multi-fidelity execution studies. These techniques have enabled multi-fidelity models to improve the solution accuracy and time savings for optimization problems by leveraging low-fidelity models for acceleration and by building correction methods for representing high-fidelity models to ensure high accuracy and convergence. One of the model management techniques is an adaptation that generates a multi-fidelity model by making the correction of the low-fidelity model using the updates obtained from the high-fidelity simulations. Multi-fidelity metamodels have been commonly constructed with either additive or multiplicative correction techniques [21] by utilizing the evaluation of low and high-fidelity simulations [37, 46, 66].

Padron et al. [41] developed the multi-fidelity method to optimize a RAE2822 wing subject to uncertain flow conditions. The method uses polynomial chaos expansion constructed from a low-fidelity model and a model correction to approximate the high-fidelity statistics. Results showed that 60% to 90% computational savings can be achieved compared to the high-fidelity method.

Multi-fidelity surrogate models are formed employing different fidelity models based on the specific details of the given problem combined. Yoo et al. [61] created four different MF models based on an artificial neural network and used them for reliability analysis and composite panel optimization. Results demonstrated that the proposed approach could reduce the computational cost by at least half compared to traditional methods. Tao and Sun [58] proposed a deep belief network-based multi-fidelity surrogate model for the robust aerodynamic design of an airfoil. In this framework, a deep belief network was employed as a low-fidelity model, and a linear regression model was used as a correction model. Fernández-Godino et al. [15] reviewed multi-fidelity model (MFM) trends in optimization, uncertainty quantification, and optimization under uncertainty fields. Additionally, they surveyed the proper way to determine how multi-fidelity model savings can be achieved.

Vitali et al. [60] first proposed the idea of multi-fidelity models via fracture progression in a composite structure. The concept is to employ the correction response surfaces for low-fidelity models to follow the global response behavior of high-fidelity models. Multi-fidelity models are produced by combining correction response surfaces with low-fidelity surrogate models; these models offer an accurate answer at a cheap computational cost. Alexandrov et al. [3] presented the trust region-based model management optimization framework for the multi-fidelity aerodynamic optimization of wings to reduce the computational cost. This strategy employs the first-order additive and multiplicative correction models for low-fidelity models to satisfy the first-order consistency constraints. These correction functions are constructed with Taylor approximations, which make the low-fidelity models follow the results of high-fidelity models while saving dramatic computational costs.

Park et al. [42] compared three different multi-fidelity surrogate frameworks and proposed a method to assess the performance of multi-fidelity models. The method is based on the ratio of high and low-quality simulation costs required to construct a multi-fidelity model. They showed that multi-fidelity models are helpful for computational time savings, and prediction performances of multi-fidelity models are highly correlated with the ratio of the sample sizes between high and low-fidelity models.

Recently, Yoo et al. [62] have proposed a multi-fidelity model-based framework that uses considerably fewer high-fidelity models than the conventional models with the aim of more significant computational time savings. With the help of fewer design variables with HFM, high-fidelity simulations are reduced, and low-fidelity simulations are used to explore the entire design space. The proposed multi-fidelity formulation achieved a significant computational advantage compared to conventional multi-fidelity models. Kudela et al. [30] examined the latest developments and state-of-the-art surrogate model applications for FEM-based computations, including publications on multi-fidelity. The review emphasizes theoretical and practical advancements in model development and validation, sensitivity analysis, uncertainty quantification, and surrogate-assisted optimization.

Lately, neural network-based metamodels have been widely used for engineering problems. Deep neural networks (DNNs), a division of ANNs, have demonstrated exceptional accomplishments in many applications, from natural language processing [33] to image analysis [51], by adjusting the number of hidden layers to enhance the functional capacity of neural networks for expressing complex functions. The learning capacity of DNNs should be appropriately balanced with the extent of supervision that is presented by training data to successfully capture patterns with high generalization, even though DNNs with higher capacity have a high chance of capturing complex patterns of a problem in a smaller number of epochs. In this situation, new approaches are required to expand the network's capacity without drastically increasing the number of trainable parameters, such as innovative network architectures or new types of activation functions.

It has been observed that quadratic neural networks have been one of the promising network architectures. Fan et al. [14] suggested that quadratic networks increase the capacity of networks by means of parametric efficiency and expressibility. Their study showed that complex radial functions could be approximated with a more compact network architecture than ordinary ANNs. Nguyen et al. [38] investigated the performance of quadratic neural networks (QNNs), artificial neural networks (ANNs), and high-order ANNs to predict the strength of foamed concrete under compressive loading. According to the validation results, high-order ANNs and QNNs were established to obtain higher confidence in the strength prediction than ANNs. Bu et al. [9] applied a quadratic neural network to solve physics problems involving PDEs and showed that QNNs could be an effective tool where high expressive power is required. They also conducted a study to demonstrate that QNNs are superior to ANNs in the sense of faster convergence speed and better parametric efficiency.

Adaptive activation functions have been recently developed activation functions for enhancing learning ability. Tezel et al. [59] implemented the ANN architecture with adaptive activation functions and compared the results of the proposed network with the classical ANN model, including non-adaptive activation functions. It has been shown that a model with an adaptive activation function converged faster than the traditional model and achieved superior performance compared to the traditional one. Lau et al. [32] investigated the performance of saturated, unsaturated, and adaptive activation functions for DNNs to compare the misclassification rate of networks. It is claimed that adaptive activation functions could solve the saturation problem for hyperbolic activation functions with two additional trainable parameters. Moreover, it has resulted in adaptive activation functions having better generalization performance compared to the other activation functions. Jagdap et al. [27] applied adaptive activation functions to accelerate the training speed of physics-informed neural networks (PINNs) by introducing a scalable hyperparameter. The prospective adaptive activation functions have better learning capability as well as preferred prediction accuracy compared to the traditional ones. It has been shown that the suggested method significantly improved the robustness, accuracy, and parametric efficiency of ANNs. Jagdap et al. [26] also presented two types of locally adaptive activation functions, referred to as neuron-wise and layer-wise, to enhance the convergence rate and performance further and to accelerate the training process of DNNs and PINNs.

In this study, state-of-the-art ANN architectures are utilized for developing multi-fidelity models for buckling load estimation of the stiffened panel. This work presents a comparative study of the performance of multi-fidelity models created using ANN and QNN with adaptive/non-adaptive activation functions. In addition, Optuna [2] is employed as the hyperparameter optimization framework to select optimal hyperparameters for neural networks. The aim of this study is to develop a strategy in order to generate optimal multi-fidelity models with high parametric efficiency and improved convergence speed, in addition to significant computational cost savings with high accuracy. Assessment of all models' performances will be made, and multi-fidelity models for each network architecture will be compared to the high-fidelity models. The best-performing multi-fidelity models created with each architecture will then be compared.

CHAPTER 3

BUCKLING ANALYSIS OF STIFFENED PANEL

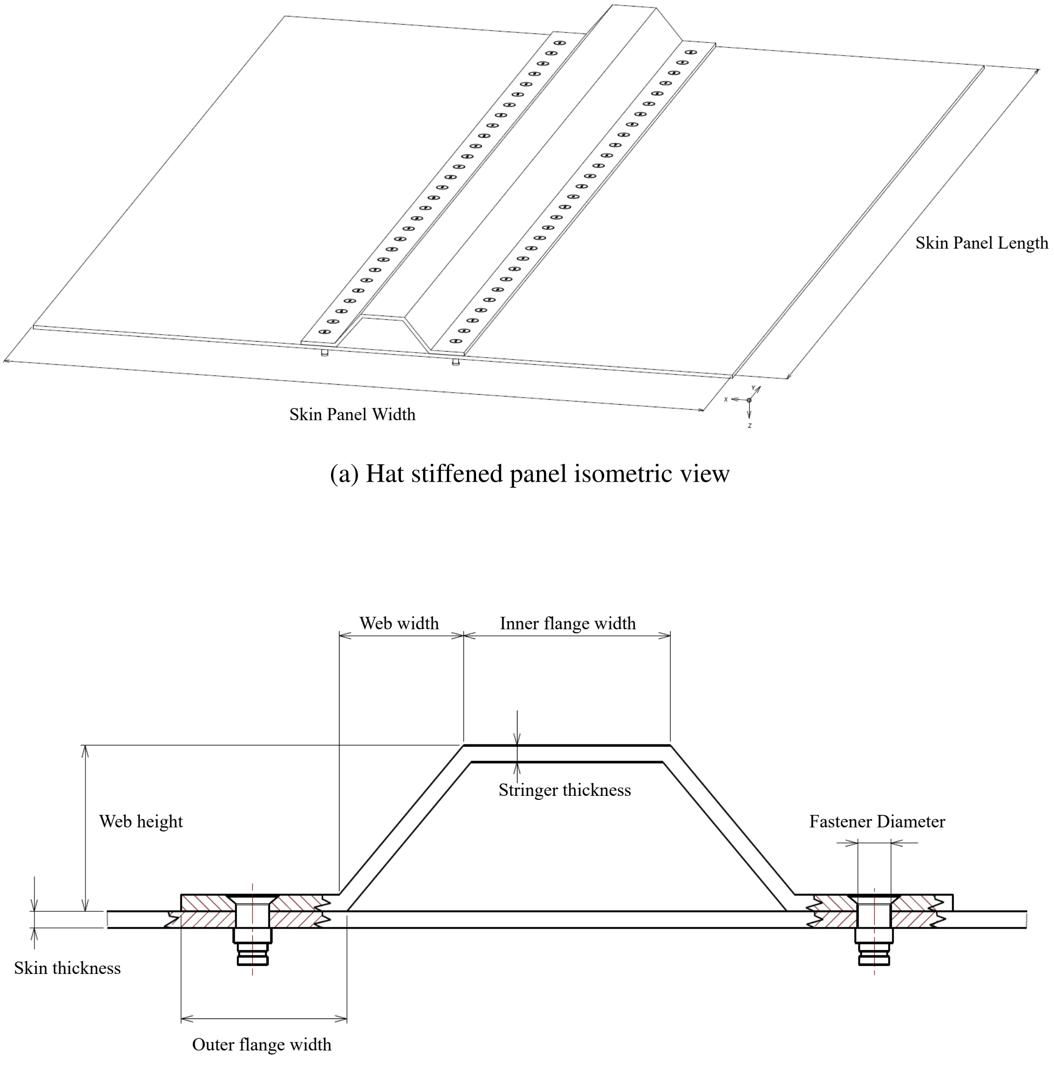
Finite element analysis of complex structures mandates a significant amount of computational resources and time to achieve accurate results. The primary sources of an increase in computational time involve nonlinearity in geometry, material properties, connections between parts, and finite element sizes.

This study aims to develop an ANN-based computational model to determine the first buckling load of stiffened panels with reasonable accuracy and reduced computational time. In order to accomplish this goal, multi-fidelity surrogate models are developed based on artificial neural networks using different fidelity models. Fidelities differ based on the size of the elements in finite element analyses. Low-fidelity datasets are obtained by analyzing finite element models with coarse mesh sizes, whereas high-fidelity datasets are obtained by analyzing models with fine mesh sizes. Since dataset quality is essential for training ANNs, a mesh convergence study is performed to choose the optimum element size for low and high-fidelity analyses. ABAQUS [53] is used for the structural analysis of stiffened panels. Low-high fidelity datasets involve the results of a large number of finite element analyses at design points determined using the sample strategies. These analyses are performed using ABAQUS, which can run a parametric algorithm written in Python, allowing all buckling analyses to be completed in a single step.

This chapter seeks to establish a baseline methodology for all finite element analyses conducted in this study. Firstly, geometric dimensions and connections between parts will be described. Details of load and boundary conditions are given next. Moreover, mesh properties and finite element solution methodology will be represented. In the last step, the results of the sample analysis study for fine and coarse meshes will be discussed.

3.1 Geometry and Dataset Description

The initial stage of finite element analysis is creating the geometry of the model. A rectangular panel and a stringer are used to create a skin-stringer assembly. A sample skin-stringer assembly model drawing and stringer cross-section drawing are given in Fig 3.1a and Fig 3.1b, respectively.



(a) Hat stiffened panel isometric view

(b) Hat type stringer cross section

Figure 3.1: Technical drawing for hat stiffened panel

Table 3.1: Constant inputs for dataset

Variable	Dimensions
Rivet Diameter	3.2 mm
Rivet Spacing	19.2 mm
Skin Panel Length	600 mm
Skin Panel Width	400 mm

Table 3.2: Variable inputs for dataset

Variable	Dimensions [mm]
skin thickness	1.5, 2.0, 2.5, 3.0
stringer thickness	1.5, 2.0, 2.5, 3.0
stringer web height	20, 25, 30, 35
outer flange width	15, 20, 25, 30
stringer web width	15, 20, 25, 30
inner flange width	20, 25, 30, 35

Stringer in skin-stringer assembly is selected as the hat type, with two flanges fastened to the skin and one upper flange. Hat section stiffeners are commonly used in the aerospace structures since they have a structurally efficient design and are easy to manufacture. The stringer with hat cross sections is created and connected to the skin panel with rivets of 3.2 mm diameter. Rivet connections are simulated using "Fastener" type connectors of ABAQUS. In order to avoid stress concentration around the rivet holes, it is recommended that the fastener spacing on an airframe structure should be between four and six times the fastener diameter [4]. In this thesis, the rivet diameter is specified as 3.2 mm, the most common and smallest size, and the fastener spacing is assigned five times of fastener diameter to all models. The wireframe view of the finite element model of a hat-stiffened panel is given in Figure 3.2.

The designed skin panel is a 600x400 mm sheet metal plate. The width and length of the skin panel are constant throughout the whole study. Constant inputs are given in Table 3.1. Six design parameters influence the buckling performance of the structure. These parameters are skin thickness, stringer thickness, stringer web height, stringer web width, stringer inner flange width, and stringer outer flange width. Each parameter has four design levels. Dimensions of these parameters are determined as typical aircraft design values and presented in Table 3.2.

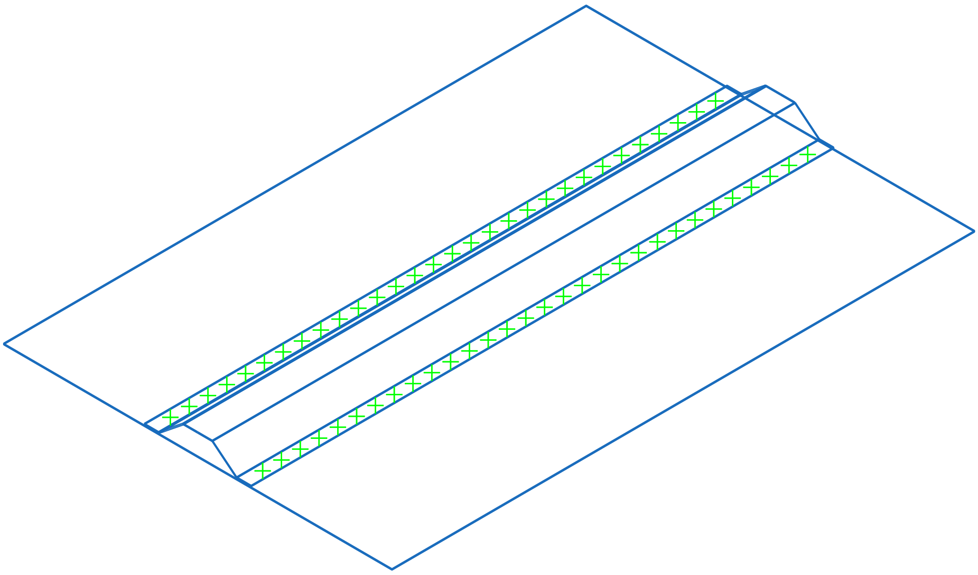


Figure 3.2: Finite element model of hat stiffened panel

3.2 Load and Boundary Conditions

The hat-stiffened panel assembly is subjected to compression loading. Two reference points are defined to be the master nodes of the hat-stiffened panel on the upper and lower edges (y-direction). Kinematic coupling is assigned between these edges and reference points. These reference points are used for loading and boundary conditions.

A concentrated force is applied from the load application point, defined as RP1 in Figure 3.3, to simulate the compressive load. The assembly is clamped at both ends from the reference points, except the loading edge is unrestricted to move in the loading direction (+y-direction). In order to simulate real-life skin-stringer connections, displacement, and moment boundary conditions are applied to the side edges of this panel. These boundary conditions allow rotation for side edges, while preventing them from translating in the out-of-plane direction.

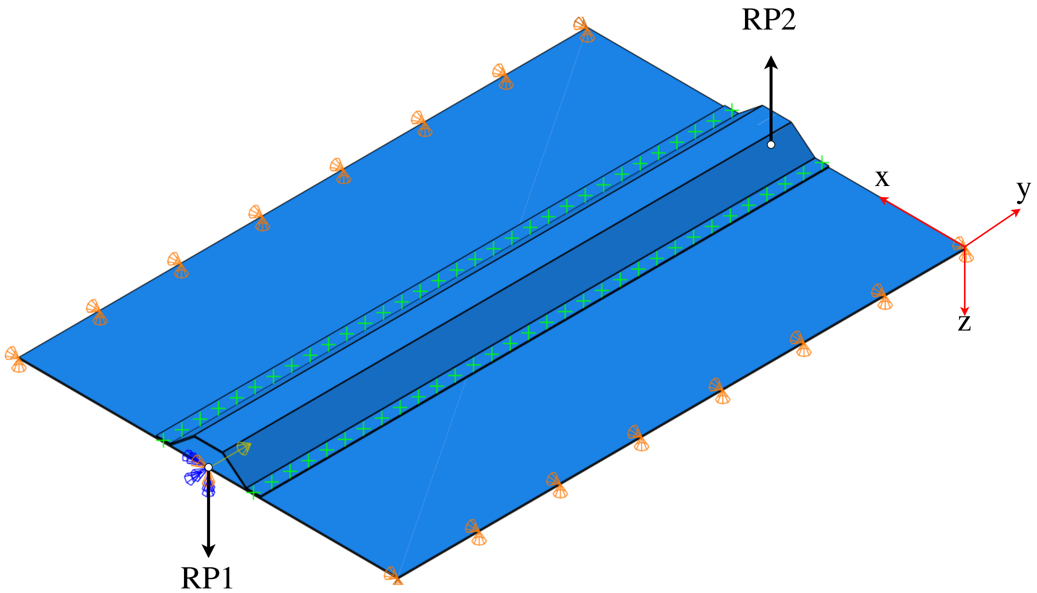


Figure 3.3: Compressive loading of stiffened panel

3.3 Material Properties

In this study, the material of stiffened panels is chosen as aluminum due to its lightweight properties, which makes it ideal for aerospace applications. Aluminum offers many options for its series. 2024-T3 clad sheet aluminum is selected for skin and stringer structures. The shell section for chosen aluminum properties is devised and assigned to the stiffened panel. The 2000 series are often utilized in the aerospace structures for sheet metal parts mandating a high strength/weight ratio and fatigue resistance. Fuselage and wing structures under high compression or tension are primary application areas for this. Material properties [1] of Al 2024-T3 used in modeling are given as follows:

- Compressive yield strength [MPa], F_{cy} : 296,
- Young's modulus [MPa], E : 73084,
- Density [kg/m^3], ρ : 2768,
- Poisson's ratio, ν : 0.33.

3.4 Mesh Quality

A mesh convergence study is being undertaken to determine the appropriate mesh size for both high-fidelity and low-fidelity models, utilizing an extensive range of eight analytical models with varying element sizes ranging from 2 to 20 millimeters. The buckling analyses in Abaqus are conducted using an Intel Xeon E5-2680 v3 CPU, which provided the high processing power needed for the complex computations required for this study. Table 3.3 includes the total computational time of buckling analysis for each mesh size and the percentage error of each load with respect to the first buckling load obtained from the finest mesh (2 mm).

Table 3.3: Mesh size, Error, and Time

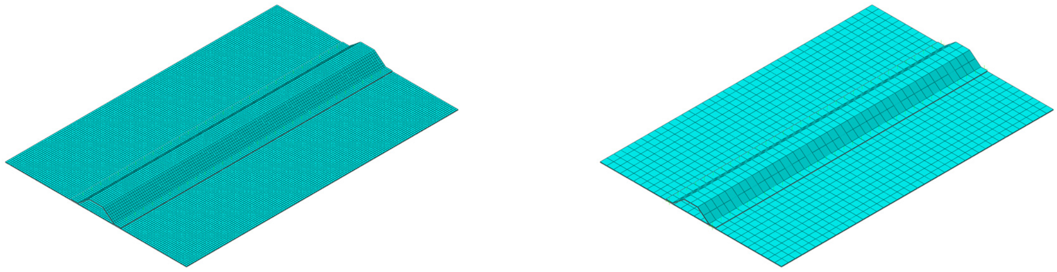
Mesh Size (mm)	Percentage Error in Load (%)	Time (s)
2	0.00	47092
4	1.51	11412
6	2.90	5086
8	11.31	3142
10	11.85	2068
12	17.00	1550
15	26.35	1130
20	53.17	782
25	498.50	544

It is of utmost importance to carefully choose the proper mesh sizes when conducting multi-fidelity simulations since it requires balancing accuracy with the computational expenses of simulations. The present study determines mesh sizes based on the mean absolute percentage error of buckling load and total computational time of the analyses, given in Table 3.3 for 200 design points.

Upon careful consideration of the low-fidelity simulations, it has been determined that a mesh size of 20 millimeters is selected as an optimal choice. This decision is based on the fact that the load percentage errors increase too fast after the 20 mm mesh size. Therefore, choosing a larger mesh size would have increased the error while only marginally reducing the computational cost. Figure 3.4b shows the finite element model with a mesh size of 20 mm.

When conducting the high-fidelity simulations, a mesh size of 4 mm has been selected to balance accuracy and computational time. It was determined that using a finer mesh size of 2 mm would have produced more accurate results but with significantly increased computation time. Thus, choosing a 4 mm mesh allows for achieving a less than 2% percentage error in load which is a critical limit for this study, while keeping the computation time within reasonable limits. Figure 3.4a shows the finite element model with a mesh size of 4 mm.

The accuracy of the high-fidelity model (HFM) differs from the low-fidelity model (LFM) by 35%, and LFM costs 75% less than HFM. The choice of mesh size for the high-fidelity and low-fidelity models is based on a balance between accuracy and computation time. The 4 mm mesh size provides good accuracy while still being computationally efficient for the HFM. The 20 mm mesh size is sufficient for the LFM and allows for a faster computation time.



(a) Mesh size 4

(b) Mesh size 20

Figure 3.4: Mesh size 4 and 20 mm

In addition to the mesh size, the element type is critical for the finite element model. In this study, all parts of the hat-stiffened panel are modeled as the shell parts. In order to mesh the shell, S4R elements, defined as a four-node element with a single integration point (reduced integration), including hourglass control and finite membrane strains, are used. According to ABAQUS documentation [53], the S4R element type is recommended for modeling general-purpose shells. For thin shells with S4R elements, discrete Kirchhoff thin shell theory is applicable, whereas, for thick shells, thick shell theory is applicable [50]. The thickness of the element determines the theory to be used with S4R elements.

3.5 Finite Element Solution Methodology

Eigenvalue buckling analysis is used to predict structure stability under external loads. In order to perform the analysis, the eigenvalue problem must be solved using the equation $(K_0 - \lambda K_\Delta) \cdot u = 0$, where K_0 is the stiffness matrix at base state, K_Δ is the stiffness matrix induced from the incremental load, λ is the eigenvalue, and u is the eigenvector. The critical load at which the system becomes unstable and experiences buckling is represented by the eigenvalue λ, and the mode shape of the system at this critical load is represented by the eigenvector u.

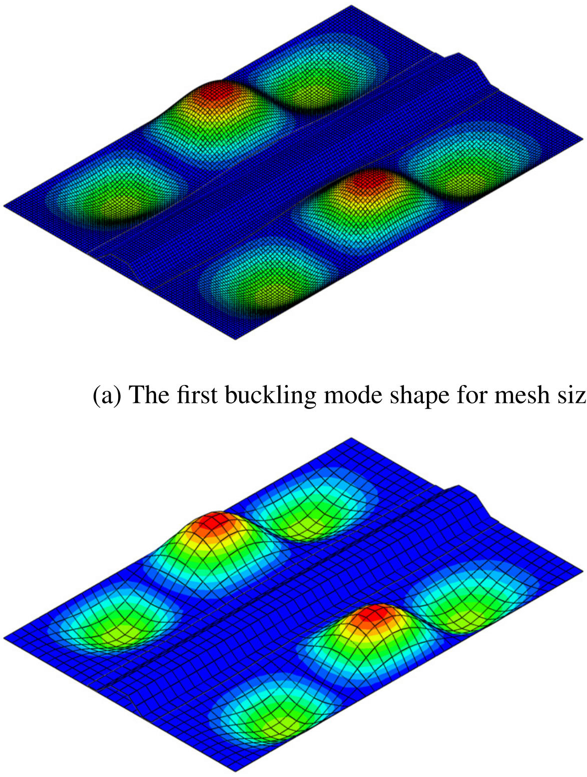
The ABAQUS software has two options for conducting eigenvalue buckling analysis: the Subspace solver and Lanczos solver. Deciding on which one to choose requires taking into account variables such as the characteristics of the problem and the desired level of accuracy and convergence. The Subspace solver is generally more efficient, stable, and particularly well-suited for the large-scale systems. According to ABAQUS documentation [53], the Lanczos solver is generally faster but may sometimes suffer from convergence issues. In this case, the Subspace solver is chosen as the solver.

In order to successfully tackle the eigenvalue problem in ABAQUS, one must first indicate "Linear Perturbation-Buckle" as the solution step and select a preferred solver. In the Subspace solver, the number of specified eigenvalues, the vectors used per iteration, and the maximum number of iterations must be defined. The number of specified eigenvalues determines the number of critical loads and mode shapes that will be solved. In contrast, the vectors used per iteration and the maximum number of iterations control the accuracy and convergence of the solution. The vectors used per iteration refer to the number of vectors used to construct the subspace in each iteration of the Subspace solver. The subspace is a set of orthonormal vectors used to iteratively solve the eigenvalue problem until the desired accuracy is achieved. In each iteration, the Subspace solver updates the subspace with the new vectors and refines the solution until convergence or the maximum number of iterations is reached. A higher number of iterations and vectors per iteration will result in a more accurate and converged solution and increase the computational time.

The number of eigenvalues is set as one to obtain the first buckling load. Balancing accuracy and computational efficiency is achieved by using 12 vectors per iteration while limiting maximum iterations to 3000.

3.6 Finite Element Analysis Results

In this study, the purpose of FE analyses is to determine the critical buckling load of hat-stiffened panels. Therefore, for each case, only one eigenvalue is requested. Since the load is applied as concentrated force, the resulting eigenvalue gives the buckling load without additional calculations.



(a) The first buckling mode shape for mesh size 4

(b) The first buckling mode shape for mesh size 20

Figure 3.5: The first buckling mode shapes for mesh size 4 and 20 mm

In addition to the buckling load, buckling mode shapes (eigenvectors) are also predicted by this analysis. Since the magnitudes of these eigenvectors are normalized to have maximum displacement as one, the buckling mode shapes do not represent the actual magnitudes of deformation in the structure. The mode shapes for the fine mesh (4 mm) and coarse mesh (20 mm) are shown in Figure 3.5a and Figure 3.5b, respectively. For the hat-stiffened panel with fine mesh, the first buckling load is found to be 33 kN, and the first buckling load with coarse mesh is 45 kN. Although the mode shapes look similar for these meshes, the resulting loads significantly differ.

CHAPTER 4

MULTI-FIDELITY MODELS

Surrogate models are simple representations of a complex system by mimicking the functional relationship between design variables and output. These surrogate models predict complex systems with acceptable accuracy while incurring computationally cheap evaluation costs. Therefore, surrogate models have been extensively applied to optimization problems in different disciplines to cope with the high computational costs brought on by expensive simulations. Kriging [52, 48], support vector machines (SVM) [20, 11], polynomial chaos expansion (PCE) [57], artificial neural networks (ANN) [7], radial basis functions (RBF) [10, 13], and various others are widely used surrogate models in the optimization.

The main goal of employing surrogate models is to imitate expensive to evaluate high-fidelity models, such as FEM-based models [7, 34, 5], using affordable statistical models. These surrogate models are created based on the input and output data of computationally expensive simulations with high accuracy. Although the surrogate models diminish the high computational cost of sophisticated models, it still takes significant time to construct them for complex problems. Different techniques such as dimensionality reduction, parallel computing, or simpler surrogate models can be employed to lower the cost of training time of surrogate models. However, high-fidelity data is required for large-scale problems with non-linear behavior to ensure the targeted accuracy. Because of the high simulation costs, gathering the high-fidelity data needed for the model to be of the desired reliability and high accuracy is frequently challenging. In contrast, utilizing low-fidelity data to reduce expenses may result in unreliable models. Since a trade-off between accuracy and computational time is essential while constructing these models for low and high-fidelity data, choosing the suitable model plays an important role.

In these circumstances, multi-fidelity methods have been proposed between these models to balance accuracy and computational time. Different fidelity surrogate models are combined to create a more accurate overall model while retaining low-fidelity models' computational efficiency. In order to balance the optimization cost and accuracy, multi-fidelity (MF) surrogate modeling methods that can incorporate data from high-fidelity (HF) and low-fidelity (LF) models have been suggested in the literature; see, e.g., [42].

The multi-fidelity models are constructed based on the information from the different fidelity models to achieve the desired accuracy at a reasonable cost. Generally, two types of models are introduced in multi-fidelity modeling: High-Fidelity Models and Low-Fidelity Models. Low-fidelity models can be obtained by dimensional reduction, linearization, use of simpler physics, coarser domain discretization, and partially converged results. These models are less accurate approximations of the high-fidelity models. The cost of acquiring high-fidelity data is higher than that of low-fidelity data. Thus, allocating computational resources between high and low-fidelity simulations can lower the general training expense.

Multi-fidelity modeling is a technique that involves using multiple models with different fidelity to make predictions. Surrogate model types of these single-fidelity models affect the performance of multi-fidelity models. In particular, the choice of a surrogate model can significantly impact the accuracy, reliability, and efficiency of the multi-fidelity modeling. This study focuses on deep neural networks (DNNs) as surrogate models for the multi-fidelity modeling. DNNs are a specific type of artificial neural network (ANN) consisting of multiple hidden layers between the input and output layers. These hidden layers enable DNNs to model complex relationships between inputs and outputs, making them powerful and effective for various applications. In this work, we explore different DNN alternatives to produce surrogate models for multi-fidelity modeling.

Multi-fidelity neural networks can be formed with a lower training cost and higher prediction accuracy than ordinary DNNs. Low-fidelity deep neural network (LFDNN) is trained with low-fidelity simulation results, whereas high-fidelity deep neural network (HFDNN) is trained with high-fidelity simulations. Low-fidelity networks estimate the overall pattern, while high-fidelity networks model local features and fluctuations.

In our problem, the training data for LFDNN and HFDNN are obtained from the finite element method (FEM). LF data are taken from the FEM simulations with coarser spatial discretizations in this work. The fundamental goal of multi-fidelity models is to develop metamodels for HFMs employing correction methods.

Multi-fidelity modeling commonly uses multiplicative, additive, and comprehensive correction function approaches to combine information from high-fidelity (HFM) and low-fidelity (LFM) models to create accurate and efficient models.

The multiplicative correction approach is given by

$$\hat{y}_{HF}(x) = \beta(x) \cdot y_{LF}(x), \tag{4.1}$$

where $\beta(x)$ is the ratio of LFM and HFM responses multiplied by the LFM response to estimate the HFM response.

The additive correction approach is given by

$$\hat{y}_{HF}(x) = \delta(x) + y_{LF}(x), \tag{4.2}$$

where $\delta(x)$ is the difference between the HFM and LFM responses added to the LFM response to estimate the HFM response.

The comprehensive correction approach is represented by

$$\hat{y}_{HF}(x) = \delta_c(x) + \rho \cdot y_{LF}(x), \tag{4.3}$$

where the equation is similar to the additive correction approach but includes a scaling factor ρ to weight the contribution of the LFM response. By adjusting the value of ρ, the comprehensive correction approach can balance the accuracy and efficiency of the resulting model.

Multiplicative and additive correction methods are often used in multi-fidelity applications since they are relatively simple and easy to implement in many applications. These correction methods incorporate information from high-fidelity simulations without requiring much additional computation. Therefore these methods are especially beneficial when high-fidelity simulations are computationally expensive. Additive methods capture variations where the difference between the output of the low and high-fidelity models varies significantly. In other cases, the multiplicative correction method may be superior, as discussed in Vitali's study [60], as it can more effectively capture the overall trend in the data.

In addition to these methods, the comprehensive correction method is also used in multi-fidelity modeling. The comprehensive model is the updated version of the additive correction method by multiplying the low-fidelity model with the scaling factor. The scaling factor has been shown to improve the performance of multi-fidelity models using additive correction, which can have bumpiness, defined as immediate and significant fluctuations in the difference between the HFM and LFM when combined. Bumpiness incorporates variation and waviness, which is defined as

$$B(f(x)) = \int |f''(x)|^2 \, dx, \tag{4.4}$$

where $B(f(x))$ is bumpiness of a function $f(x)$.

The scaling factor can be selected to minimize the error or bumpiness of the difference model [17]. The steps involved in the scaling factor optimization study are outlined in the Algorithm 1.

Algorithm 1 Optimizing Scaling Factor ρ using fmincon

```
Define the function ($\Delta$) as $\Delta = HF_{data} - \rho \cdot LF_{data}$.
Define the objective function $fun(\rho)$ = bumpiness($\Delta$) or $fun(\rho)$ = mape($\Delta$).
Set the upper bound $ub = 1$ and lower bound $lb = 0$.
Set the initial guess $x_0 = 0.5$.
Call the optimization function: $x$ = fmincon($fun, x_0, lb, ub$).
Return the optimal scaling factor $\rho = x$.
```

The presented pseudo-code given in Algorithm 1 first defines the difference model Δ as the difference between high-fidelity and scaled low-fidelity models. The objective function is then defined as the bumpiness or mean absolute percentage error of the difference model. Finally, the fmincon function in MATLAB [24] is called to find the optimal scaling factor ρ that minimizes the objective function.

After that, a deep neural network called Discrepancy Deep Neural Network (DDNN) is trained over additive correction based on the difference between high-fidelity and low-fidelity data. In the same way, a multiplicative deep neural network (MDNN) is trained based on the high-fidelity and low-fidelity data ratio. Alternative multi-fidelity networks can be created by combining predictions from LFDNN and DDNN or LFDNN and MDNN.

Neural networks created utilizing each training dataset, such as LFMs and response correction surfaces, are used to construct multi-fidelity surrogate models. The accuracy of multi-fidelity models depends on the quality of surrogate models developed by datasets sampled using the Latin Hypercube Sampling (LHS)[35].

The pursuing multi-fidelity models are employed to compare with different high-fidelity models,

$$\hat{y}_{MF_m}(x) = \beta^{ANN}(x) \cdot y_{LF}^{ANN}(x) \tag{4.5}$$

$$\hat{y}_{MF_d}(x) = \delta^{ANN}(x) + y_{LF}^{ANN}(x) \tag{4.6}$$

$$\hat{y}_{MF_c}(x) = \delta_c^{ANN}(x) + \rho \cdot y_{LF}^{ANN}(x) \tag{4.7}$$

where $\hat{y}_{MF_m}$, $\hat{y}_{MF_d}$, and $\hat{y}_{MF_c}$ are the outputs of the developed multi-fidelity surrogate models, ρ is a constant multiplicative factor, and $\delta^{ANN}(x)$, $\delta_c^{ANN}(x)$, and $\beta^{ANN}(x)$ are the metamodels for the response correction function.

CHAPTER 5

METHODOLOGY

Multi-fidelity modeling approaches have been used to reduce the computational cost of different engineering problems [3, 42, 60, 62]. While a significant decrease in cost can be achieved in multi-fidelity modeling with ANNs, for complex problems, choosing an appropriate algorithm to construct multi-fidelity models is still a problem. Depending on the complexity of the problem, high functional capacity is required for deep networks, bringing additional performance and resource issues.

This study aims to use state-of-the-art algorithms to create optimal multi-fidelity models with high parametric efficiency and convergence speed. In Table 5.1, the proposed algorithms, which are expected to achieve accurate results with fewer parameters, are given. Four different multi-fidelity models are created in this framework, and detailed evaluations are done.

It is decided to employ a conventional multi-fidelity network as a baseline approach for the performance assessment. This baseline procedure uses five classic ANNs to generate extensive and limited multi-fidelity models. The predictions of these multi-fidelity and high-fidelity models are assessed in this study. In the same way, the other algorithms are used to develop multi-fidelity models, and buckling load estimations of these models are evaluated.

The main steps of the methodology followed for all artificial neural network approaches developed in this study are shown in Figure 5.1.

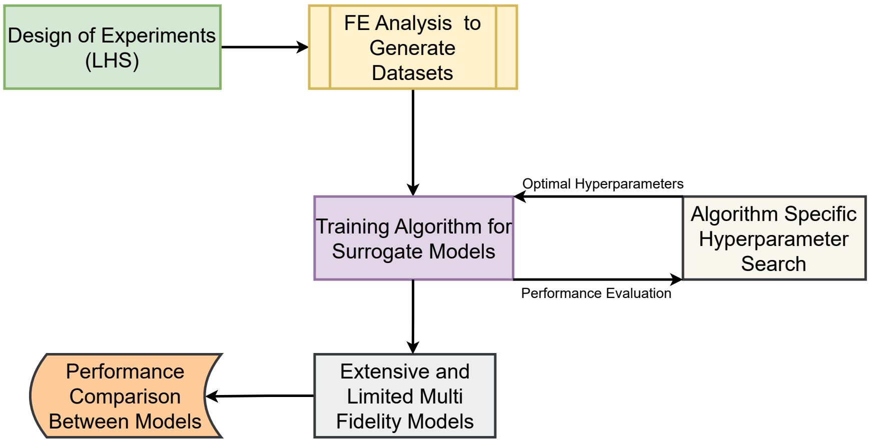

Design of Experiments (LHS)
FE Analysis to Generate Datasets
Training Algorithm for Surrogate Models
Optimal Hyperparameters
Algorithm Specific Hyperparameter Search
Performance Evaluation
Extensive and Limited Multi Fidelity Models
Performance Comparison Between Models

Figure 5.1: Multi-fidelity modeling methodology for neural network architectures

In this study, the dataset is created through FEM simulations run with ABAQUS using Python scripting. After dataset creation, the hyperparameter search space is determined for each previously mentioned model. In Table 5.1, algorithms for buckling load estimation in this research and related hyperparameters are shown. Then, using the Optuna optimization framework [2], the hyperparameters from which the highest-performing models obtained are chosen. After that, the best models trained with optimal hyperparameters are selected for comparison.

Table 5.1: Hyperparameters for models

Models	Hyperparameters
MLP with CAF	Activation Function, Neuron Size, Number of Layers, Learning Rate, Optimizer, Batch Size
MLP with AAF	Activation Function, Neuron Size, Number of Layers, Learning Rate, Optimizer, Batch Size, α_{AAF}, n_{AAF}
QMLP with CAF	Activation Function, Neuron Size, Number of Layers, Learning Rate, Optimizer, Batch Size
QMLP with AAF	Activation Function, Neuron Size, Number of Layers, Learning Rate, Optimizer, Batch Size, α_{AAF}, n_{AAF}

5.1 Dataset Generation

In this study, there are six geometric design parameters, each with four levels. Definitions of these variables can be seen in Table 5.2.

Table 5.2: Input variables for dataset

Variable	Explanation
t_{sk}	skin thickness
t_{str}	stringer thickness
h	stringer web height
b_1	outer flange width
b_2	stringer web width
b_3	inner flange width

The referenced datasets for this problem are generated offline using the high-fidelity and low-fidelity structural buckling analyses carried out with ABAQUS. The FE model's element size determines the fidelity type that characterizes the high-fidelity and low-fidelity data. The high-fidelity analyses are conducted with a mesh size of 4 mm, whereas the low-fidelity analyses are made with a mesh size of 20 mm. In this study, high-fidelity and low-fidelity data are used to build High-Fidelity and Low-Fidelity Models, which are employed for developing the multi-fidelity models.

Multiple simulations are performed to generate the training data by sampling the parameters in Table 5.2, and these parameters are shown in Figure 3.1b. For each variable, minimum and maximum values are set, and the corresponding design intervals are divided into four equal intervals. For the baseline high-fidelity comparison model, the full factorial approach is used to create a dataset that includes all possible combinations of the design parameters at a discrete number of design points. The design points to create surrogate models for multi-fidelity models require a sampling technique. This study uses Latin Hypercube Sampling (LHS) [23] as the sampling method since it can provide good coverage of sample space to ensure that the experiment results are representative of the underlying distribution. Also, LHS can reduce the cost and complexity of the experiment by minimizing the number of required samples [56]. Using LHS, suitable design points are determined for creating the data to generate different multi-fidelity models. For generating multi-fidelity models and accomplishing comprehensive explanations by examining their performance and cost comparison with single-fidelity models, datasets with a sample size of 1060, 1375, and 3000 are sampled with LHS, and high-low fidelity analyses are carried out to generate them.

5.2 Artificial Neural Networks

Artificial neural networks (ANNs) use the brain and nervous system processing as a foundation to develop models that can be used to classify complex patterns and make correct estimations. ANNs are imitations and simplifications of biological neural networks. Like the human brain, ANNs consist of neurons organized into several layers.

Feed-forward neural networks are a widespread neural network that is the first and most straightforward type of artificial neural network; see, e.g., [19] for more details. Feed-forward neural networks comprise the input layer, the hidden layers, and the output layer. These layers consist of connected neurons which transmit input signals to the neurons in the subsequent layers. The input layer brings the external data into the system for further processing by the hidden layers. The output layer gives the model's prediction, which is the problem's solution. The hidden layers are the intermediate layers in which the neurons apply the weights to inputs and send them through the neurons of the next layer.

Multi-layer perceptron (MLP) is a layered feed-forward neural network consisting of interconnected neurons which process data through three or more layers. The basic structure of MLP follows one input layer, at least one hidden layer, one output layer, an activation function, and a group of weights and biases. MLPs with more than one hidden layer are called deep ANN or deep neural networks (DNN). In MLPs, neurons from one layer are connected to all neurons in the neighboring layer(s); in other words, these layers are fully connected, and the connection between neurons is constructed through the weights and bias. Once the output is generated from the output layer, the loss function between the actual and predicted values is calculated depending on the chosen loss metrics. In MLP, the training algorithm uses the back-propagation algorithm to learn the dataset. During training, neuron weights are tuned to minimize the loss function by using a backpropagation algorithm. Overall, the neural network's main focus is finding the optimal weight values to improve prediction accuracy.

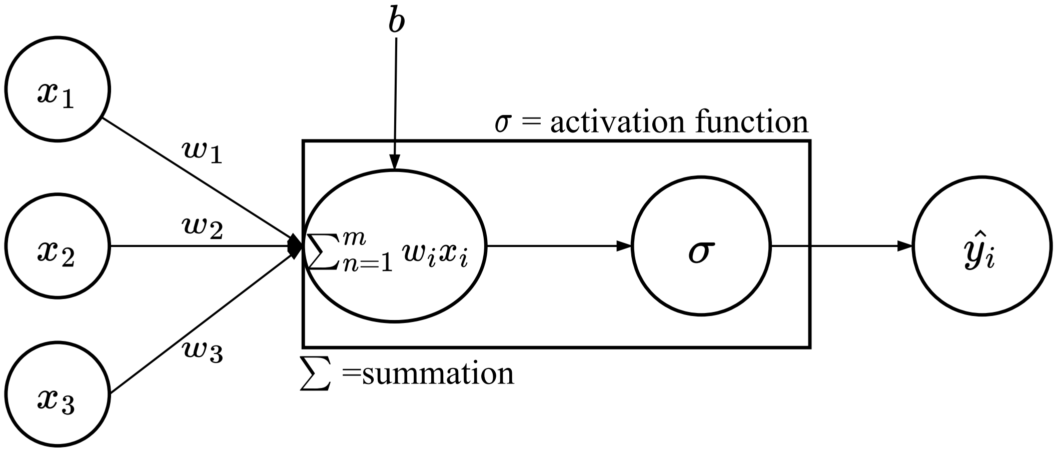


Figure 5.2: Mathematical operation steps of an artificial neuron

In Figure 5.2, the mathematical operation steps of an artificial neuron can be examined. The network produces the following output for the training set $\{(x_i, y_i)\}_{i=1}^{N}$:

$$\hat{y} = f(x, \theta) = \sigma(Wx + b) = \sigma\left(\sum_{i=1}^{N} w_i x_i + b\right), \tag{5.1}$$

where σ is the nonlinear activation function, and $w_i(i = 1, ..N)$ and b are learnable parameters named as weight and bias terms, respectively. The model parameters are described as $\theta = \{W, b\}$, where $W = \{w_i\}_{i=1}^{N}$ and $b = \{b_i\}_{i=1}^{N}$. The backpropagation algorithm utilizes gradient-based optimization techniques employing the loss function

$$\mathcal{L}(x; \theta) = d\big(y, f(x; \theta)\big). \tag{5.2}$$

The loss function can be set based on the chosen loss metrics d. The following minimization problem determines the optimum weights and bias terms:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(x; \theta), \tag{5.3}$$

where $\mathcal{L}(x; \theta)$ is the loss term to be optimized. This optimization problem seeks to minimize the loss function to reach optimum weights and biases, as given in

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla_\theta \mathcal{L}(x; \theta), \tag{5.4}$$

where η is the learning rate and $\theta = \{W, b\}$ is the pair of the weight and bias parameters of the neural network. In the weight update mechanism, the learning rate, a configurable hyperparameter, acts as the step size and controls how quickly the model weights fit the problem.

5.2.1 Activation Functions

Activation functions are classified as constant or adaptive depending on the slope of the activation function.

5.2.1.1 Constant Activation Functions

In this training process, the activation function plays a critical role by adjusting the outputs of neurons. For solving complex problems, nonlinear activation functions are needed in neural network design. Nonlinear activation functions attached to each neuron will activate the output, depending on whether the output value satisfies the threshold. In other words, the activation function serves as a mathematical formula to determine neuron switch on or off. Common activation functions in deep neural networks are hyperbolic tangent, sigmoid, and rectified linear unit (ReLU). While nowadays the most commonly used activation function is ReLU, before the discovery of ReLU, activation functions were mainly sigmoidal functions, which are hyperbolic tangent and sigmoid functions are defined, respectively, by

$$f(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right), \tag{5.5}$$

$$f(x) = \left(\frac{1}{1 + e^{-x}}\right). \tag{5.6}$$

The main problem with the sigmoidal activation functions is that their derivative tends to be zero as x goes to plus-minus infinity ($\frac{df(x)}{dx} \to 0$ as $x \to \pm\infty$). The problem is deepened in deep neural networks since the gradient propagates and multiplies throughout the layers, which results in excessive training time. This phenomenon is called the vanishing gradient problem, which can be avoided using ReLU. The rectified linear unit defined in (5.7) [36] has a gradient of one if the input is positive and zero if the input is negative.

$$f(x) = max(0, x). \tag{5.7}$$

ReLU solves the vanishing gradient problem and is faster than traditional activation functions since the function does not require any computationally expensive operation. ReLU has many variants, such as Exponential Linear Unit (ELU), Scaled Exponential Linear Unit (SELU), and Gaussian Exponential Linear Unit (GELU), which are simple and address different issues related to the performance of activation functions. Although these functions work well, it has been some interest in finding better activation functions.

5.2.1.2 Adaptive Activation Functions

Adaptive activation functions [26, 27] introduce an additional hyperparameter alpha (α), which changes the slope of the activation function. Adaptive activation functions (AAF) have better learning capabilities than the constant activation functions since it increases the convergence rate and training accuracy. Also, the vanishing gradient problem observed in sigmoidal activation functions can be solved with adaptive activation functions thanks to additional trainable parameters.

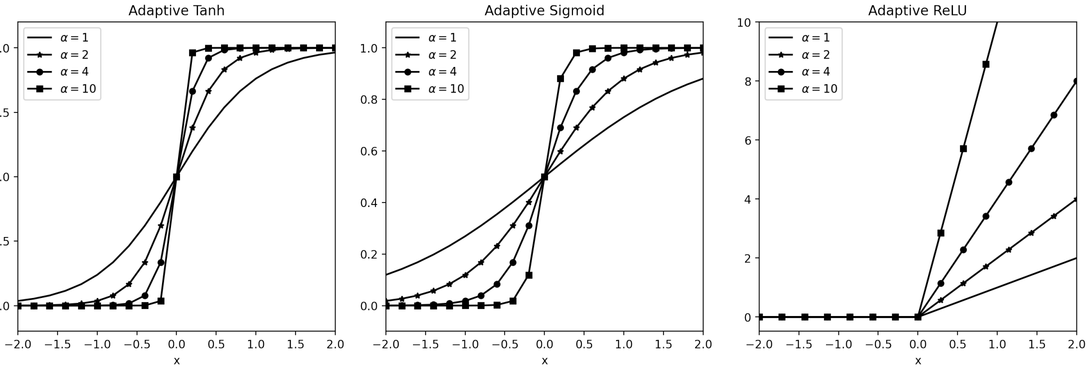
In addition to the trainable parameter α, the scaling factor n can also be helpful. For DNNs, choosing the correct learning rate is essential since it affects the search for the global minimum. In adaptive activation functions, the scaling factor behaves like a secondary learning rate toward finding minima. Figure 5.3 indicates adaptive activation functions with different α values, which change the slope of activation functions. In order to accelerate convergence to the optimum for the network, a scaling factor n multiplied by α can be chosen as $n \geq 1$. In this case, the final form of adaptive activation functions becomes as in (5.8- 5.10) for the hyperbolic tangent, sigmoid, and rectified linear unit functions, respectively,

$$\text{Adaptive Tanh} : \left(\frac{e^{n \cdot \alpha \cdot x} - e^{-n \cdot \alpha \cdot x}}{e^{n \cdot \alpha \cdot x} + e^{-n \cdot \alpha \cdot x}} \right), \tag{5.8}$$

$$\text{Adaptive Sigmoid : } \left(\frac{1}{1+e^{-n\cdot\alpha\cdot x}}\right), \tag{5.9}$$

$$\text{Adaptive ReLU : } \max(0, n\cdot\alpha\cdot x). \tag{5.10}$$

Adaptive activation function can be applied layerwise or neuronwise [26]. Neuronwise adaptive activation functions define α parameters for each neuron in the neural network, whereas layerwise adaptive activation functions define α parameters for each layer. In this research, a layerwise adaptive activation function has been employed.



(a) Adaptive Tangent (b) Adaptive Sigmoid (c) Adaptive RELU

Figure 5.3: Adaptive activation functions with different values of slope α

5.2.2 Quadratic Neural Networks

Combining numerous neurons with nonlinear activation provides Deep Neural Networks (DNN) with high functional capacity. For a dense neural network, the capacity to capture complex patterns in fewer training epochs increases as the network deepens. However, training becomes more challenging for DNNs with high capacity. The generalization performance of ANN is defined as the difference between the error on training data and unseen data. The complexity of the neural network and the amount of training data determines the generalization capability. Therefore, the number of trainable parameters should be balanced appropriately with the available training data to achieve generalization for the network.

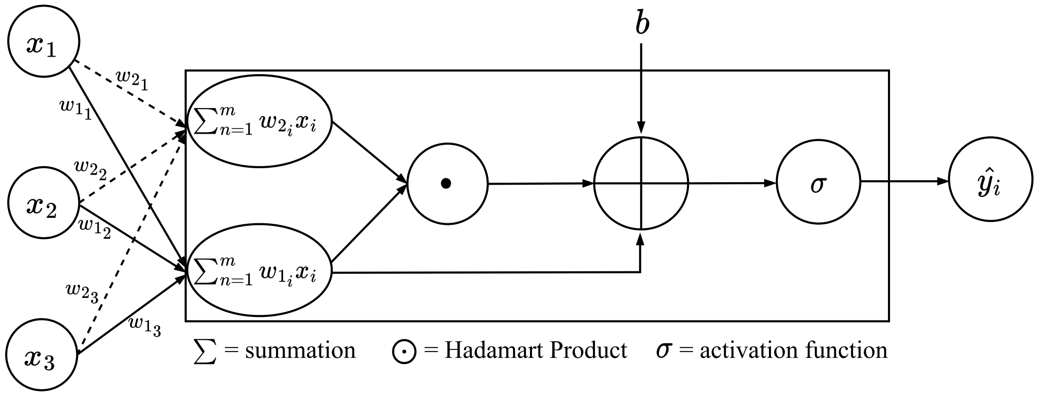

x_1
x_2
x_3
w_{2_1}
w_{1_1}
w_{2_2}
w_{1_2}
w_{2_3}
w_{1_3}
$\sum_{n=1}^{m} w_{2_i} x_i$
$\sum_{n=1}^{m} w_{1_i} x_i$
b
σ
$\hat{y_i}$
$\sum$ = summation
$\odot$ = Hadamart Product
σ = activation function

Figure 5.4: Mathematical operation steps of a quadratic neuron

Quadratic neural networks [9, 14, 38] are developed to approximate complex functions with less trainable parameters than ordinary DNNs. In Figure 5.4, the mathematical operation steps of a quadratic neuron can be examined. In this network architecture, an additional weight term (W_2) is introduced for each neuron, and an additional quadratic term ($W_1 \cdot x \circ W_2 \cdot x$) is added to the output of neurons before applying the activation function. This quadratic term provides supplemental nonlinearity in addition to the activation functions.

5.3 Multi-Fidelity Modeling

This study uses different artificial neural network options to develop surrogate models based on High-Fidelity Models (HFMs) and Low-Fidelity Models (LFMs). The correction response surfaces of $\delta(x)$ and $\beta(x)$ are used as the ratio and difference between the HFMs and the LFMs to produce the multi-fidelity models.

Four alternative surrogate model choices are investigated to find an optimal multi-fidelity model with low cost and high accuracy. These alternatives are MLP with constant activation, MLP with adaptive activation, quadratic MLP with constant activation, and quadratic MLP with adaptive activation functions. Although the network architectures differ, the methodology for multi-fidelity models will be identical. For each network choice, ANN creates five metamodels: High-Fidelity DNN, Low-Fidelity DNN, Multiplicative DNN, Discrepancy DNN, and Comprehensive Discrepancy DNN. Multiplicative, Discrepancy, and Comprehensive Discrepancy DNN are created by the difference and ratio between high and low-fidelity models, as demonstrated in Table 5.3.

Table 5.3: Response correction models

Model	Output approximation
Multiplicative DNN	$\beta^{ANN}(x) = \dfrac{Y_{HF_{1060}}(x)}{Y_{LF_{1060}}(x)}$
Discrepancy DNN	$\delta^{ANN}(x) = Y_{LF_{1060}}(x) - Y_{HF_{1060}}(x)$
Comprehensive Discrepancy DNN	$\delta_c^{ANN}(x) = \rho \cdot Y_{LF_{1060}}(x) - Y_{HF_{1060}}(x)$

For Comprehensive Discrepancy DNN, scaling factor ρ is optimized as 0.70 by minimizing bumpiness, whereas ρ is optimized as 0.75 by minimizing error. According to sample results in Table B.1, minimizing bumpiness gives better results than minimizing error, which aligns with the literature [43]. Therefore, ρ is chosen as 0.70 from now on.

The design points to create ANNs are obtained using the LHS for different sample sizes. At first, the datasets with 1060 and 3000 design points are sampled using LHS. Once the sampling has been completed, high-fidelity and low-fidelity analyses are done with determined input parameters. Datasets are divided into training, test, and validation sets as 70% for training, 15% for validation, and 15% for testing. For simplicity, surrogate models will be called upon fidelity and dataset size. For example, the low-fidelity surrogate model trained with 1060 samples dataset is called LFM1060. In this study, LFM1060, LFM3000, HFM1060, and HFM3000 surrogate models are trained for Low-Fidelity and High-Fidelity DNNs, whereas Diff1060, Diffc1060, and Ratio1060 surrogate models are trained for Discrepancy, Comprehensive Discrepancy, and Multiplicative DNNs.

The multi-fidelity models are produced by combining these low-fidelity surrogate models with discrepancy and multiplicative surrogate models, as demonstrated in Table 5.4. Multiplicative surrogate models, also known as ratio-based models, and discrepancy surrogate models, also known as difference-based models, are utilized in creating these multi-fidelity models. The extensive multi-fidelity models, MF1, MF2, and MF3, need costly computation time since they call for the low-fidelity models trained with a dataset with 3000 samples. On the other hand, limited multi-fidelity models, MF4, MF5, and MF6, are cheap since they call for low-fidelity models trained with 1060 samples.

Table 5.4: Multi-Fidelity Models

Model	Output approximation
MF1	$\hat{y}_{MF1} = \beta^{ANN}(x) \cdot Y_{LF_{3000}}(x)$
MF2	$\hat{y}_{MF2} = \delta^{ANN}(x) + Y_{LF_{3000}}(x)$
MF3	$\hat{y}_{MF3} = 0.70 \cdot Y_{LF_{3000}}(x) + \delta_c^{ANN}(x)$
MF4	$\hat{y}_{MF4} = \beta^{ANN}(x) \cdot Y_{LF_{1060}}(x)$
MF5	$\hat{y}_{MF5} = \delta^{ANN}(x) + Y_{LF_{1060}}(x)$
MF6	$\hat{y}_{MF6} = 0.70 \cdot Y_{LF_{1060}}(x) + \delta_c^{ANN}(x)$

5.4 Performance Evaluation Metrics

In this study, the accuracy of the model's performance is assessed by comparing the actual (y) and predicted ($\hat{y}$) values for each network and multi-fidelity model. R-squared (R^2), mean absolute error (MAE), mean absolute percentage error (MAPE), and mean squared error (MSE) values are used as the performance metrics in this work. The R-squared (R^2) value measures the proportion of variance of the y value, which is the solution for the problem. In other words, it quantifies the degree of linear dependency between the actual and predicted value of the model. While R-squared (R^2) is close to one, the fit between the actual and predicted value is almost perfect.

Although R-squared (R^2) is an essential and commonly used metric in regression analysis, it is not sufficient on its own to determine the accuracy of the neural network

[55]. This is because R-squared assumes linear relationships between the independent and dependent variables, which may not hold for the nonlinear models. In addition, R-squared only considers the variance explained by the model and does not take into account other aspects of model performance, such as bias or error. Therefore, MAE, MSE, and MAPE are additional metrics to assess the performance of a neural network. Generally, a higher value of R^2 and a lower value of other metrics result in a decrease of loss between actual and prediction values. Definitions of these metrics are given in (5.11- 5.14):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}, \tag{5.11}$$

$$\mathcal{L}_{MAE} = \left(\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|\right), \tag{5.12}$$

$$\mathcal{L}_{MSE} = \left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2\right), \tag{5.13}$$

$$\mathcal{L}_{MAPE} = \left(\frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right), \tag{5.14}$$

where y and $\hat{y}$ are the actual and prediction values of the model, respectively, $\overline{y}$ is the mean of the actual values of the model, and N is the number of samples.

Finally, the number of trainable parameters is measured for each model to compare the parametric efficiency of alternative neural network architecture used to create multi-fidelity models.

5.5 Numerical Experiments

To implement ANN, we leverage Pytorch [44], developed by the Facebook AI research team, an open-source machine-learning framework for developing and training deep-learning models. Pytorch enables us to change activation functions and neural network structure, in which we need to implement quadratic and adaptive activation functions. Also, PyTorch facilitates GPU acceleration for the neural networks, which can remarkably enhance the training efficiency of deep neural networks. In this study, a multi-layer perceptron neural network is adopted as a baseline model. The generic configuration of this network with six inputs, one output, and N hidden layers is shown in Figure 5.5.

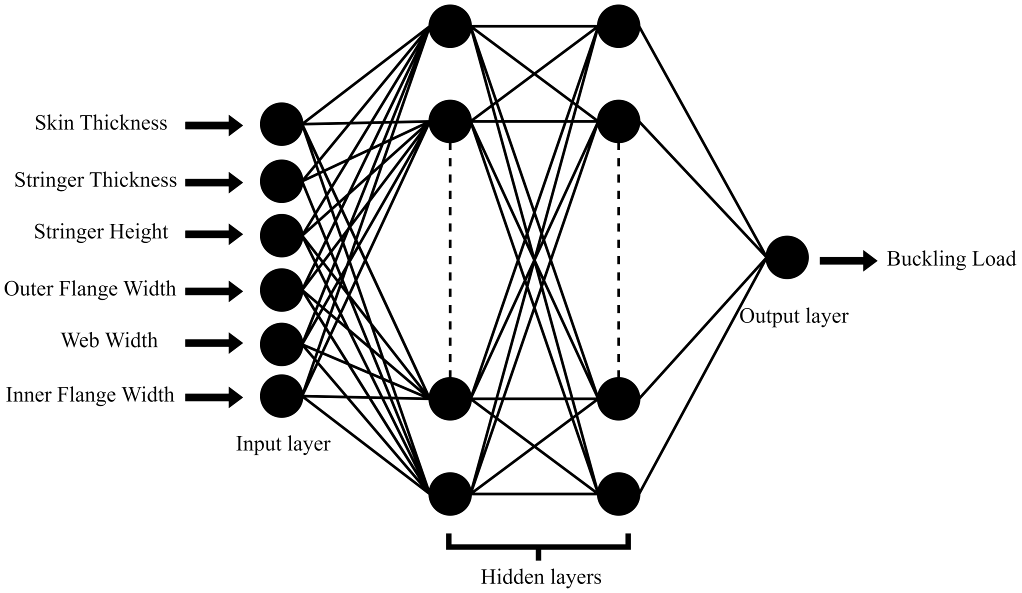


Figure 5.5: Multi layer perceptron model

This study employs four different network structures to create multi-fidelity models, and six surrogate models are trained for each multi-fidelity model series. Each corresponding surrogate model is trained with the same dataset, such as the low-fidelity surrogate model in MLP and QMLP has the same training set. Since a detailed comparison is needed to reasonably compare these multi-fidelity models created with different architectures given in Table 5.5, an extensive hyperparameter search is needed for every model. The performance of models is evaluated based on the performance metrics given in Section 5.4.

Hyperparameter optimization is one of the most challenging research domains in neural network architectures, especially for DNN. While the search space includes all the possible hyperparameters, the goal is to find the best hyperparameters which lead to the minimum loss function. Hyperparameters in search space are similar for all four algorithms. Hyperparameters are determined as the activation functions, neuron size, learning rate, optimizers, and batch size for algorithms with constant activation functions. On the other hand, two additional parameters, α and n, are added to the hyperparameters for the algorithms with adaptive activation functions. Search space for these hyperparameters is determined based on the literature and the number of resources allocated for this study. Although the activation functions are given in Table 5.5, search space for each architecture involves only one activation function because, based on initial experiments, it has been observed that the ReLU is more successful as a constant activation function and the tanh as an adaptive activation function.

Table 5.5: Hyperparameters search space

Models	Hyperparameters	Search Space
MLP with CAFs	Activation Function	ReLU
	Neuron Size	[16, 256]
	Number of Layers	[1, 8]
	Learning Rate	$[1 \times 10^{-5}, 1 \times 10^{-2}]$
	Optimizer	Adam, Yogi, Lamb
	Batch Size	[50, 250]
MLP with AAFs	Activation Function	Adaptive Tanh
	Neuron Size	[16, 256]
	Number of Layers	[1, 8]
	Learning Rate	$[1 \times 10^{-5}, 1 \times 10^{-2}]$
	Optimizer	Adam, Yogi, Lamb
	Batch Size	[50, 250]
	α (AAF)	[1, 2]
	n (AAF)	[1, 2]
QMLP with CAFs	Activation Function	ReLU
	Neuron Size	[16, 256]
	Number of Layers	[1, 8]
	Learning Rate	$[1 \times 10^{-5}, 1 \times 10^{-2}]$
	Optimizer	Adam, Yogi, Lamb
	Batch Size	[50, 250]
QMLP with AAFs	Activation Function	Adaptive Tanh
	Neuron Size	[16, 256]
	Number of Layers	[1, 8]
	Learning Rate	$[1 \times 10^{-5}, 1 \times 10^{-2}]$
	Optimizer	Adam, Yogi, Lamb
	Batch Size	[50, 250]
	α (AAF)	[1, 2]
	n (AAF)	[1, 2]

Well-known optimizers (Adam [29], Yogi [65], Lamb [63]) used in literature are one of the hyperparameters to see the effect of the optimization algorithm on the training performance. These optimizers utilize mini-batch, a fixed portion of training sets that are less than the actual training set. The batch size is defined as the number of samples passed through to the network at each group. It may have a significant impact on the accuracy of the network and the generalization performance of the networks. For comprehensive hyperparameter search, Optuna optimization framework [2], designed for machine learning studies, has been used in this study. Optuna combines an efficient search and pruning algorithm, enhancing cost-effectiveness for hyperparameter optimization.

Intending to optimize the performance of the neural network, we have defined the objective function as the mean absolute percentage error (MAPE) of the predictions in the test set. MAPE is selected as the objective function for the buckling load estimation of the stiffened panel due to its ability to effectively represent prediction error, mainly when working with actual values encompassing varying magnitudes. The optimization algorithm, called Tree-structured Parzen Estimator (TPE) [6], which belongs to a class of Bayesian algorithms designed explicitly for optimization purposes, is used to optimize the objective function.

In search of optimal hyperparameters, this study has opted to use TPE due to its ability to balance exploration and exploitation. The algorithm starts by randomly selecting a subset of hyperparameters and creates a prior probability distribution over the search space of hyperparameters. Then algorithm updates the probability distribution after each trial based on the results of the objective function evaluation. This capability of TPE allows it to focus on the most promising regions of the search space while still exploring other regions, making it a convenient choice for optimizing neural networks' hyperparameters.

The steps involved in conducting a hyperparameter optimization study are presented herein through the utilization of the Algorithm 2.

Algorithm 2 Hyperparameter Optimization Study with Optuna

```
procedure OPTIMIZE
    Define the objective function: minimize MAPE.
    Define the search space for the hyperparameters.
    Create an Optuna study using TPE as the optimization algorithm.
    Run the study.
    Get the best hyperparameters found by the optimization.
    Return the best hyperparameters.
end procedure
```

The study uses Optuna with the given steps in Algorithm 2 and minimizes the loss functions defined as MAPE, resulting in the best hyperparameters for the neural network. These hyperparameters are then used to train optimal neural networks, providing improved prediction accuracy.

5.5.1 Statistical Results of Generated ANNs

One trial study result is shown here to check the validation performance of the Quadratic MLP trained with a hyperparameter set from the search space. The network is trained with 3000 low-fidelity samples. As illustrated in Figure 5.6, the mean square error for training and validation sets are close to each other, and the error is reasonably small.

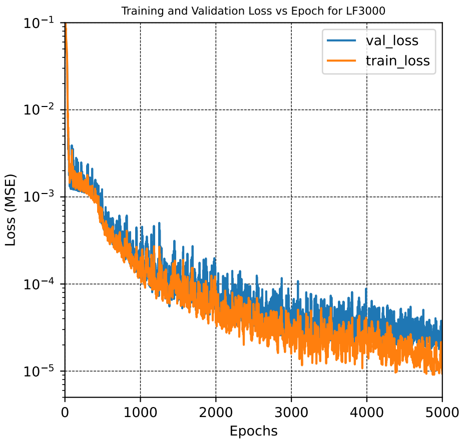


Figure 5.6: Training and validation loss vs. Epoch

The network continues to learn to last epochs; the best epoch is 4956. In Figure 5.7,

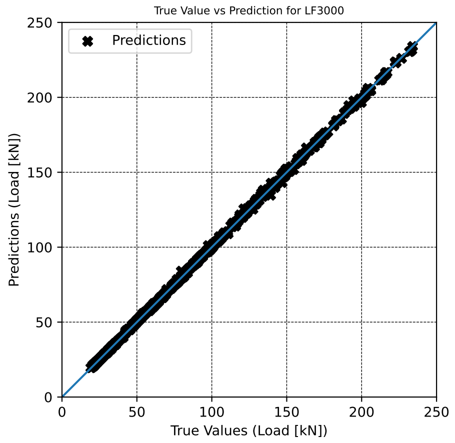


Figure 5.7: True value versus Prediction

true values are given in the horizontal axes, and ANN predictions are in the vertical axes. This figure shows the distribution of ANN predictions compared to the diagonal line. The points around the diagonal line constitute an almost bold line means the prediction fits almost perfectly with the actual value. Also, it means that the sample size of the chosen dataset for ANN is sufficient for developing multi-fidelity models. Based on these results, it can be inferred that developed ANN gives promising results for further study.

CHAPTER 6

RESULTS AND COMPARISONS

In this chapter, different types of multi-fidelity models are formed and examined for their ability to model complex systems. These models comprise quadratic multi-layer perceptrons (QMLPs) and multi-layer perceptrons (MLPs), with constant or adaptive activation functions (CAFs and AAFs), respectively.

Within this initial section of the chapter, the individual and cross-architecture performances of the evaluated models are presented and compared utilizing performance metrics. The purpose here is to improve comprehension of each model's strengths and weaknesses while identifying any conceivable trends or patterns that may have emerged.

In the analysis, a second section evaluates different surrogate algorithms concerning their efficiency and effectiveness, which are MLPs and QMLPs with CAFs and AAFs. Validation loss, time-saving, and the use of parameters are considered in this analysis.

The multi-fidelity models are evaluated in the third section of this chapter to determine their generalization performance on an additional dataset. This evaluation involves comparing the results of these models with high-fidelity analysis results for the additional design points included in the new data set.

Eventually, the fourth section assesses the performance of multi-fidelity models based on QMLP and AAFs on a dataset with different mesh sizes. The objective is to ascertain these models' efficiency and ability to handle various levels of mesh sizes.

The findings offer a significant perspective concerning the effectiveness and efficiency of multi-fidelity modeling for complex systems. The following sections expound upon what these results entail while underlining how employing such models could prove advantageous in different circumstances. Potential benefits are also highlighted within said discussions.

6.1 Performance Comparison of Multi-Fidelity Models

In this section, the performances of four different architectures are evaluated. Each model has been trained on a corresponding training dataset and has had extensive hyperparameter searches conducted on a validation dataset in order to select the model with the best performance. The final results for each model will be reported on an independent test dataset that is not used in the training or hyperparameter search processes.

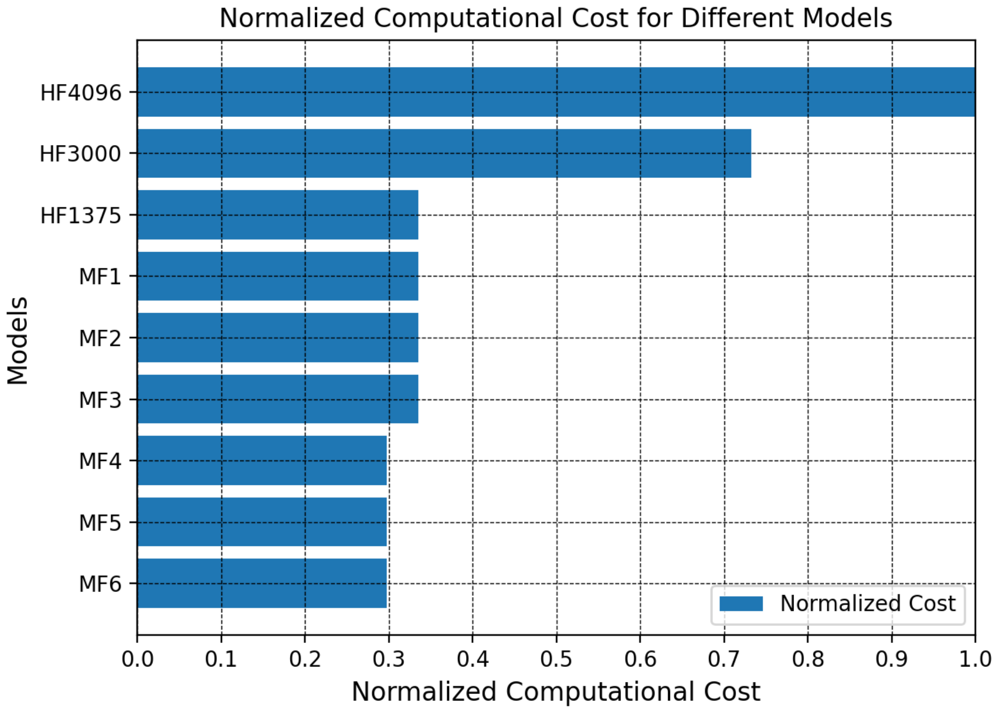


Figure 6.1: Normalized computational cost for different models

For each architecture, multi-fidelity models (MF1, MF2, MF3, MF4, MF5, and MF6) and high-fidelity models (HF1375, HF3000, and HF4096) are compared using the performance metrics; see Section 5.4. In order to normalize each model's computational time, the high-fidelity model's computational time has been used as a reference. The costs of running a high-fidelity and low-fidelity analysis are 32 seconds and 2.45 seconds, respectively.

Four multi-fidelity models are constructed in this study to balance the trade-off between accuracy and computational cost; see Table 5.4. The models MF1, MF2, and MF3 are built using 3000 low-fidelity design points and 1060 high-fidelity design points. In contrast, MF4, MF5, and MF6 are constructed using equal numbers of high and low-fidelity design points, with 1060 of each. Normalization results in MF1, MF2, and MF3 having a 1375 unit cost, whereas MF4, MF5, and MF6 have a 1220 unit cost.

In order to assess the accuracy of the multi-fidelity models and determine the high-fidelity model equivalent, high-fidelity surrogate models using various numbers of design points are created. Figure 6.1 shows the normalized computational costs for the multi-fidelity and high-fidelity models, with the cost of the HF4096 model used as the reference for normalization. The performance comparison results will be discussed in the following subsection for each architecture.

6.1.1 Comparison of Different Architectures

This section evaluates the cross-architectural performance of the different multi-fidelity models. The results are presented in Table 6.1, which shows the performance of each model for the MLP with CAFs, MLP with AAFs, QMLP with CAFs, and QMLP with AAFs architectures, respectively.

Table 6.1: Comparison of different architectures

Architectures	Metrics	MF1	MF2	MF3	MF4	MF5	MF6	HF1375	HF3000	HF4096
MLP with CAFs	MAPE	0.970	1.687	1.511	1.471	1.943	1.890	1.164	0.610	0.284
	MAE	0.702	1.189	0.959	1.029	1.322	1.325	0.627	0.326	0.149
MLP with AAFs	MAPE	0.894	1.828	1.347	1.467	2.224	1.817	1.064	0.588	0.262
	MAE	0.596	1.173	0.802	1.012	1.416	1.181	0.538	0.330	0.145
QMLP with CAFs	MAPE	0.793	1.576	1.090	1.148	1.732	1.426	1.094	0.539	0.220
	MAE	0.499	1.044	0.725	0.796	1.172	0.960	0.542	0.294	0.120
QMLP with AAFs	MAPE	0.662	1.273	1.035	1.097	1.617	1.341	0.793	0.510	0.193
	MAE	0.480	0.857	0.657	0.722	1.062	0.865	0.398	0.257	0.101

When comparing the different multi-fidelity models within each architecture, it is found that ratio-based models (MF1 and MF4) performed better than the comprehensive models (MF3 and MF6), and comprehensive models are better than difference-based models (MF2-MF5) in terms of MAPE and MAE. These results suggest that ratio-based models better capture the underlying relationships between high and low-fidelity samples than other multi-fidelity models. Meanwhile, comprehensive models are more adept at reducing errors than difference-based models.

Among the six multi-fidelity models in each architecture, MF1 consistently achieves the lowest MAPE and MAE values. Compared to the high-fidelity model HF1375, MF1 provides accurate solutions for buckling load prediction and shows a lower MAPE. Further, while MF1 has a computational cost that is only 33% of HF4096, it still outperforms the performance of HF1375. Thus, MF1 represents a more efficient and accurate choice for situations where computational resources are limited. Additional details on the performance comparison of models for each architecture can be found in Appendix A.

When comparing the results of MLP with CAFs to those of MLP with AAFs, the usage of AAFs generally improves the performance of the MLP model across most of the fidelity levels. For MF1, the MAPE and MAE values have been reduced by 7.8% and 15.2%, respectively, compared to MLP with CAFs.

The analysis of the QMLP architecture's impact on model performance shows that it generally outperforms MLP with CAFs across all fidelity levels, regardless of the activation function used. According to Table 6.1, for MF1, QMLP with CAFs decreases the MAPE and MAE values by 18.3% and 29.0%, respectively, compared to MLP with CAFs.

In the comparison of the QMLP architecture's performance using CAFs versus AAFs, it is found that using AAFs significantly improves the model's performance across all fidelity levels. For MF1, the MAPE and MAE values have decreased by 16.7% and 5.8%, respectively, compared to QMLP with CAFs, as shown in Table 6.1.

Finally, comparing MLP with AAFs to QMLP with AAFs, the QMLP architecture further improves the model's performance. As shown in Table 6.1, QMLP with AAFs outperforms MLP with AAFs, resulting in a reduction of 26.2% and 21.1% in MAPE

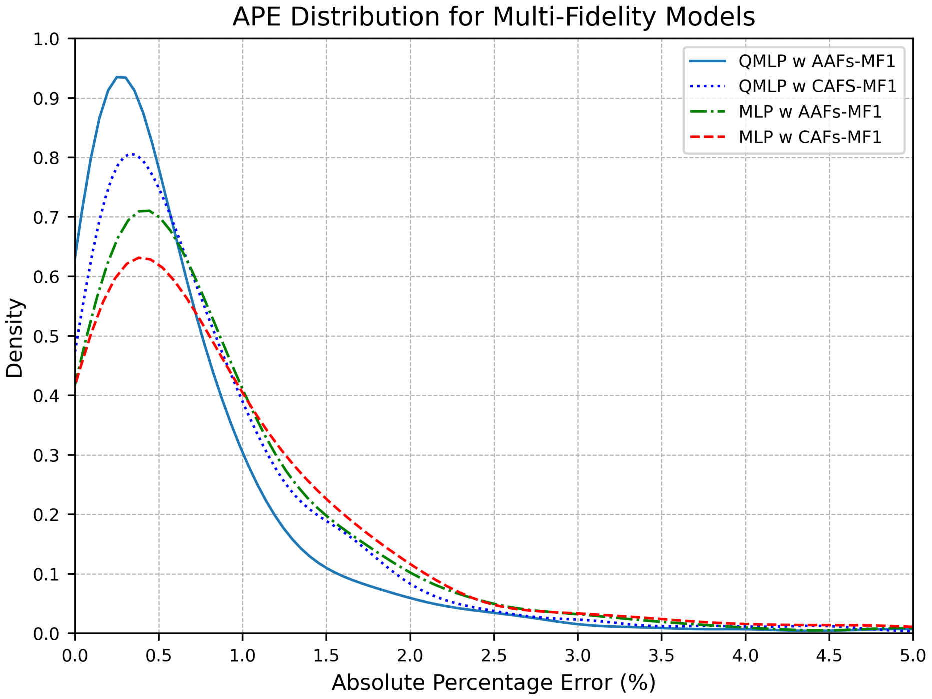


Figure 6.2: APE distribution for multi-fidelity models with different architectures

and MAE values, respectively, for MF1.

Overall, the results suggest that the QMLP with AAFs is the most promising approach for modeling complex systems, offering good performance with relatively low computational cost. The QMLP with AAFs outperformed the other multi-fidelity model approaches, making it a valuable tool for applications requiring efficient and accurate predictions. In addition, as shown in Figure 6.2, the QMLP with AAFs has the highest probability of having a low absolute percentage error (APE) value among the four models, followed by the QMLP with CAFs, MLP with AAFs, and MLP with CAFs in terms of APE performance.

6.2 Comparison of Multi-Fidelity Modeling Algorithms for Performance and Efficiency

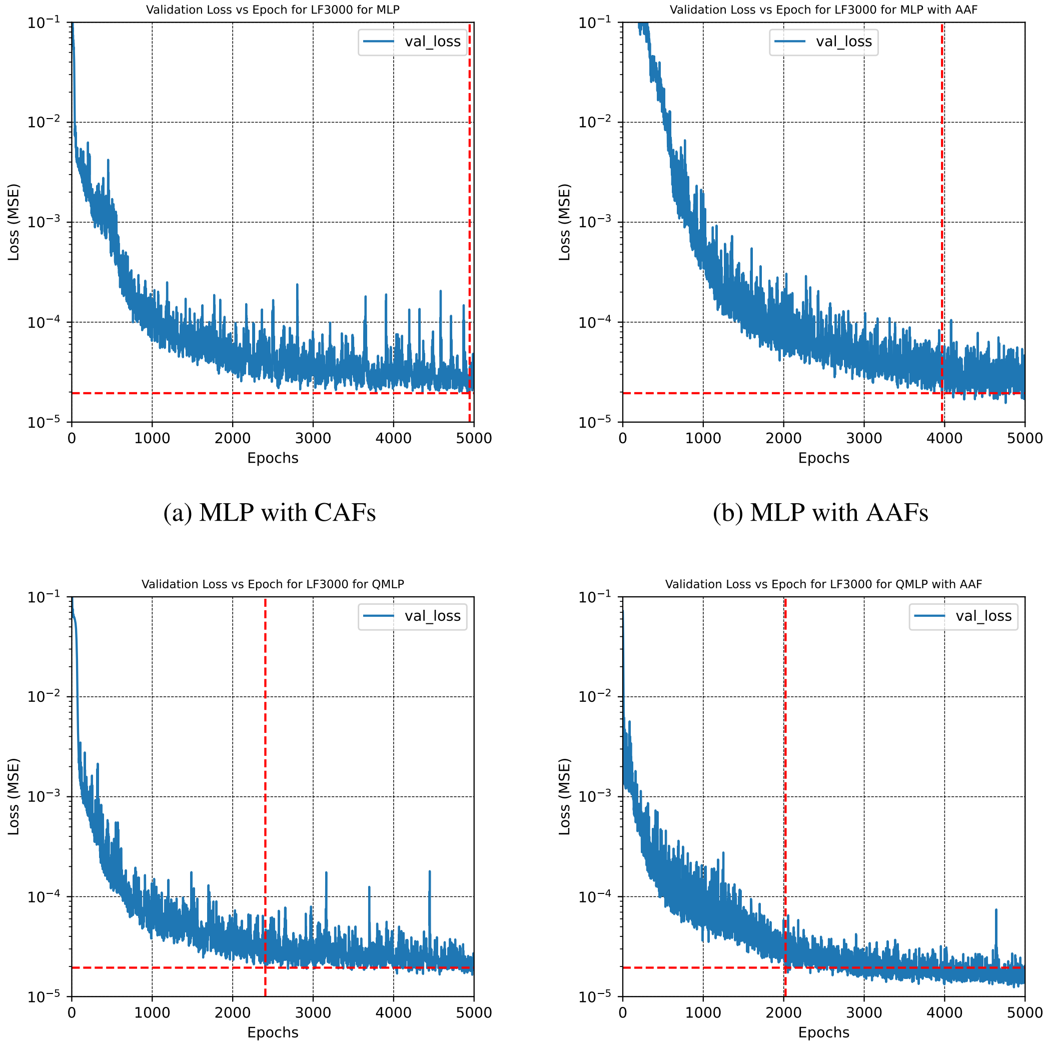
In the previous section, six different architectures have been evaluated for multi-fidelity and high-fidelity models. It has been found that QMLP with AAFs outperforms other multi-fidelity approaches, and the MF1 model, which is a ratio-based model that combines surrogate models of LF3000 and Ratio1060, is the best-performing multi-fidelity model for this architecture.

This section compares the performance efficiencies of four different multi-fidelity modeling algorithms using the LF3000 model. The LF3000 model is a low-fidelity model used as the base for the MF1 multi-fidelity model. The results are presented in Table 6.2, which compares the minimum validation loss, validation loss comparison, corresponding epoch, time-saving, the number of parameters, and model size reduction for each of the four algorithms: MLP with CAFs, MLP with AAFs, QMLP with CAFs, and QMLP with AAFs. In addition, in Figure 6.3, the validation loss graphs for these algorithms are given.

Table 6.2: Comparison between different surrogate model algorithms for LF3000

Models	Minimum Validation Loss	Validation Loss Comparison	Corresponding Epoch	Time Saving	Number of Parameters	Model Size Reduction
MLP w CAFs	0.000019	0.00%	4945	0.00%	83584	0.00%
MLP w AAFs	0.000016	20.51%	3968	19.76%	67077	19.75%
QMLP w CAFs	0.000017	15.23%	2406	51.34%	58752	29.71%
QMLP w AAFs	0.000012	36.41%	2023	59.09%	17475	79.09%

The minimum validation loss is the lowest loss value achieved during the training process of a model. A lower validation loss generally indicates a better model performance. The validation loss comparison is the percentage difference between the current model and the model with the highest validation loss. A higher percentage difference indicates a better model performance. According to the results in Table 6.2, the QMLP with the AAFs algorithm has the lowest minimum validation loss, with a value of 0.000012. This corresponds to a 36.41% improvement in validation loss compared to the MLP with the CAFs algorithm, with the highest validation loss of 0.000019.



(a) MLP with CAFs

(b) MLP with AAFs

(c) QMLP with CAFs

(d) QMLP with AAFs

Figure 6.3: Validation loss vs Epoch for LF3000 with different algorithms

The corresponding epoch is the epoch at which the minimum validation loss of MLP-CAF is achieved. A lower epoch number indicates a faster training process. The time-saving metric shows the percentage difference in training time between the current model and the model with the longest training time. A higher percentage difference indicates a more efficient model. The corresponding epoch for the QMLP with the AAFs algorithm is also the lowest, with a value of 2023. This represents a time saving of 59.09% compared to the MLP with the CAFs algorithm, which has the highest corresponding epoch of 4945.

The model size reduction metric is the percentage difference in the number of parameters between the current model and the model with the largest model size. A higher percentage difference indicates a more efficient model in terms of model size. In terms of the number of parameters, the QMLP with AAFs algorithm also performs the best, with a model size reduction of 79.09% and a reduction in the number of parameters of 17475 compared to the MLP with CAFs algorithm, which has the highest number of parameters at 83584.

Overall, these results suggest that the QMLP with AAFs algorithm is the most efficient in terms of model performance and computational cost among the four multi-fidelity modeling algorithms. It has the lowest minimum validation loss and corresponding epoch, as well as the greatest time-saving and model size reduction.

6.3 Performance of QMLP-AAF on MF1 Model with New Dataset

In this section, additional finite element analyses are performed to test the ANN. The input parameters of these analyses are given to ANN, and the results obtained with finite element analysis and neural network are compared.

Table 6.3: Variable inputs for unseen dataset

Variable	Dimensions [mm]
skin thickness	1.5, 2.0, 2.5, 3.0
stringer thickness	1.5, 2.0, 2.5, 3.0
stringer height	22, 27, 32
outer flange width	18, 22, 27
web width	18, 22, 27
inner flange width	22, 27, 32

This study needs new design space to create an additional dataset. Table 6.3 shows new dimensions for six design parameters by following typical aircraft design values. While skin and stringer thickness have identical search spaces to Table 3.2, the other parameters now have interpolated dimensions. This choice is related to the fact that skin and stringer thickness generally take particular values given in Table 6.3 in structural design for practical reasons. Using LHS, suitable design points are determined by sampling the parameters in Table 6.3 for creating the data to compare the results of additional finite element analysis and multi-fidelity model.

Table 6.4: Performance of QMLP-AAF on MF1 model with unseen dataset

		sk_t	str_t	h	b_1	b_2	b_3	y_{fem}	y_{pred}	$error$
		$[mm]$	$[mm]$	$[mm]$	$[mm]$	$[mm]$	$[mm]$	$[kN]$	$[kN]$	[%]
Panel Configurations	1	2.5	3.0	32.0	27.0	22.0	27.0	92.17	92.00	0.19
	2	2.5	2.0	32.0	22.0	18.0	32.0	71.17	71.70	0.74
	3	2.0	2.0	22.0	18.0	22.0	22.0	37.96	38.42	1.20
	4	2.0	3.0	22.0	18.0	27.0	32.0	53.54	52.36	2.21
	5	1.5	2.5	22.0	22.0	27.0	27.0	23.85	23.34	2.11
	6	3.0	1.5	22.0	27.0	22.0	32.0	106.77	106.66	0.11
	7	1.5	1.5	32.0	27.0	18.0	22.0	16.51	16.38	0.77
	8	2.0	1.5	27.0	27.0	18.0	22.0	33.90	34.12	0.67
	9	1.5	2.0	22.0	18.0	18.0	32.0	18.75	18.63	0.67
	10	2.5	1.5	32.0	18.0	27.0	32.0	68.72	66.75	2.87
	11	3.0	2.0	27.0	22.0	27.0	22.0	117.62	112.84	4.07
	12	2.0	3.0	27.0	18.0	27.0	27.0	51.93	52.08	0.29
	13	3.0	1.5	27.0	18.0	27.0	32.0	111.15	110.09	0.95
	14	3.0	3.0	32.0	22.0	22.0	32.0	144.15	142.13	1.40
	15	3.0	2.0	22.0	22.0	18.0	27.0	109.69	107.43	2.06
	16	2.5	3.0	22.0	27.0	22.0	22.0	87.24	85.79	1.65
	17	2.5	2.5	32.0	18.0	27.0	22.0	81.13	80.49	0.78
	18	1.5	2.5	27.0	27.0	18.0	27.0	21.84	21.69	0.71
	19	2.0	2.5	27.0	27.0	22.0	27.0	47.71	48.05	0.71
	20	1.5	2.5	32.0	22.0	18.0	22.0	20.27	20.50	1.14

Table 6.4 presents the input parameters of the additional analyses, results of the Adaptive QMLP (i.e., QMLP-AAF) based MF1 model with a mesh size of 20 mm and finite element analysis on these input parameters, and percentage error between these results. Also, the MAPE and MAE values calculated at these design points are given as 1.265% and 0.924 in Table B.2, respectively. According to these results, the model performance on unseen data is acceptable for this problem.

Although performance demonstration on this additional dataset is enough for engineering purposes, an extra study is conducted in Appendix B.3 using all six design parameters with interpolated dimensions.

Also, in Appendix B.2, the performance comparison between different high-fidelity and multi-fidelity models are given for Adaptive QMLP architecture using additional design points shown in Table 6.4.

6.4 Performance of QMLP-AAF on MF1 Model with Different Mesh Sizes

In this section, the performance of the Adaptive QMLP (i.e., QMLP-AAF) based MF1 model is evaluated using different mesh sizes for low-fidelity analysis while keeping the mesh size of the high-fidelity analysis constant at 4 mm. Table 6.5 presents the performance of the Adaptive QMLP-based MF1 with three different mesh sizes (12 mm, 15 mm, and 20 mm) for the low-fidelity analysis and the high-fidelity model (HF1375) for comparison.

Table 6.5: QLMP-AAF with different mesh sizes for low-fidelity models

Metrics	MF1-Mesh12	MF1-Mesh15	MF1-Mesh20	HF1375
MAPE	0.59	0.69	0.66	0.79
MAE	0.39	0.48	0.47	0.40
Max APE	4.49	10.87	9.02	4.69
Cost	1375	1375	1375	1375
Number of HF	825	950	1060	1375

It is essential to consider the mesh size when modeling complex systems, as it determines the resolution of the computational grid used to simulate the system. A finer mesh size can provide more detailed and accurate predictions, but it also increases the computational cost of the simulation. In order to maintain a consistent cost of 1375, the number of high-fidelity samples reduces as the mesh size decreases; see Table 6.5.

The results show that the Adaptive QMLP-based MF1 with a mesh size of 12 mm performs the best among the four models, with a mean absolute percentage error (MAPE) of 0.59. The Adaptive QMLP-based MF1 with a mesh size of 12 mm performs better than the Adaptive QMLP-based MF1 with a mesh size of 20 mm, which has a MAPE of 0.66. Although the mean absolute percentage error values for both models are relatively similar, the low-fidelity model with a mesh size of 12 mm still demonstrates an improvement in MAPE of 10.66% compared to the low-fidelity model with a mesh size of 20 mm.

Additionally, the Adaptive QMLP-based MF1 with a mesh size of 12 mm has the lowest maximum absolute percentage error (MAX APE) value of 4.49, compared to the MAX APE values of 9.02 for the Adaptive QMLP-based MF1 with a mesh size of 20 mm and 4.69 for the HF1375 model. The Adaptive QMLP-based MF1 with a mesh size of 12 mm achieves the highest accuracy among the four models.

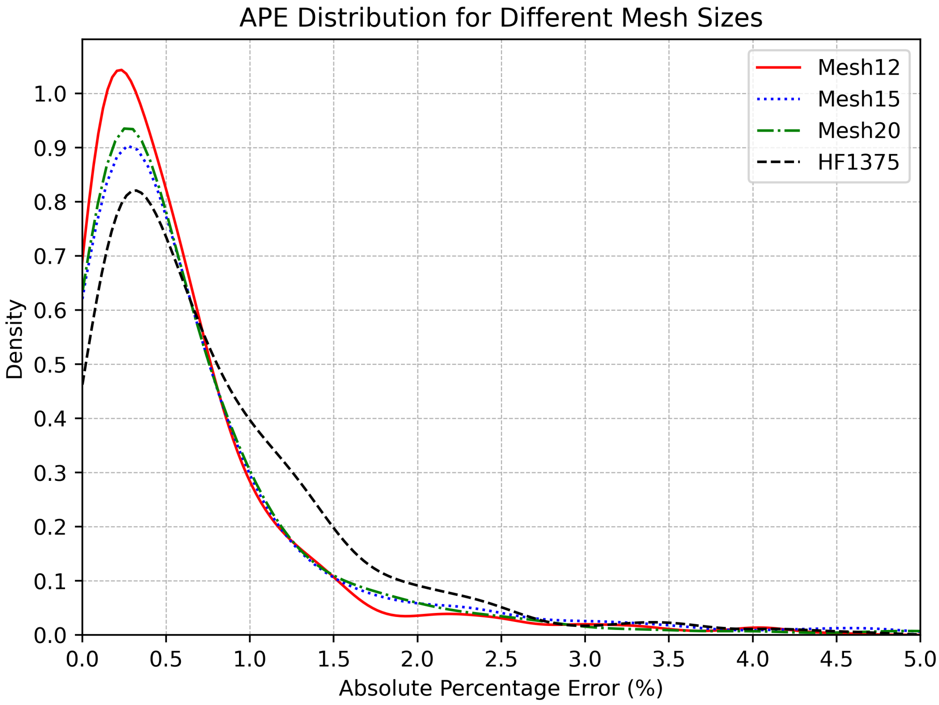


Figure 6.4: APE distribution of Adaptive QMLP with different mesh sizes

Figure 6.4 further illustrates the performance of the Adaptive QMLP model with different mesh sizes. As shown in the figure, the MF1 with mesh 12 has the highest probability of having a low absolute percentage error (APE) value among the four models, followed by the MF1 with mesh 20, MF1 with mesh 15, and HF1375 in terms of APE performance.

It is worth noting that the performance of the Adaptive QMLP model is not monotonically improving with decreasing mesh size. Choosing the optimal mesh size and the number of high-fidelity samples is not a monotonous optimization problem. In order to determine the optimal configuration, it may be necessary to consider the trade-off between mesh size, number of high-fidelity samples, and model performance.

These results highlight the importance of choosing the appropriate mesh size and the number of high-fidelity samples when using the Adaptive QMLP model for multi-fidelity modeling. The Adaptive QMLP-based MF1 with a mesh size of 12 mm achieves the best performance in terms of MAPE, MAX APE, and APE probability, making it a valuable tool for applications requiring accurate and efficient predictions. However, these findings cannot be generalized. Therefore, further studies are needed to determine the optimal combination of mesh size and the number of high-fidelity samples for different applications.

CHAPTER 7

DISCUSSIONS & CONCLUSIONS

7.1 Discussions

This study thoroughly explores the effectiveness of multi-fidelity models in various scenarios, such as multi-fidelity models with different correction methods, extensive-limited multi-fidelity models with different computational costs, multi-fidelity models with different neural network architectures, and multi-fidelity models with different numbers of high-fidelity samples and low-fidelity mesh size.

One significant result in this study suggests that multi-fidelity models constructed using the multiplicative correction method surpass those formed using the additive correction method when evaluated in terms of accuracy as aligned with the findings of Vitali's study [60]. This relationship is consistent with literature showing that the correction function's bumpiness can significantly affect the performance of multi-fidelity models [43] because the bumpiness of ratio-based models is 0.18 while the bumpiness of difference-based models is 0.29. Using a multiplicative correction function instead of additive correction makes the resulting error functions smoother and easier to optimize, leading to better overall performance.

Another important finding is that comprehensive models, which combine high-fidelity and low-fidelity data with a scaling factor optimized to minimize bumpiness, are an improved version of difference-based models and give better results than difference-based models [17]. The bumpiness values of the comprehensive model and difference model are given as 0.10 and 0.29, respectively. By reducing the bumpiness of the correction function, comprehensive models can better capture the true behavior of the system and make more accurate predictions. Although the bumpiness values of the comprehensive and ratio-based models are 0.10 and 0.18, ratio-based models still outperform the comprehensive models. Since comprehensive models are upgraded versions of difference-based models, it is reasonable to expect that comprehensive models are better than difference-based models; however, between comprehensive and ratio-based models, there is no direct connection to compare their performances based on the bumpiness values.

A detailed evaluation of the individual and cross-architecture performance of the different models shows that the QMLPs with AAFs perform the best overall. These models achieve the lowest values for the error metrics. The superior performance of the QMLP with AAFs makes it a valuable tool for applications requiring efficient and accurate predictions.

When comparing the performance and efficiency of different surrogate algorithms, the QMLP with AAFs algorithm emerges as the top performer. It has the lowest minimum validation loss, the most significant time saving, and the most efficient use of parameters. These results demonstrate the superiority of the QMLP with the AAFs algorithm compared to the other four multi-fidelity modeling algorithms tested in terms of model performance and computational cost.

Moreover, an analysis is undertaken to evaluate the efficiency of a multi-fidelity model that relies on QMLP with AAFs, based on low-fidelity data gathered from various mesh sizes. Results indicate that the Adaptive QMLP-based MF1 using 12 mm meshes produces the most favorable outcomes regarding MAPE, Max APE, and APE probability assessments. However, the model's non-monotonic behavior accentuates the significance of additional investigations to determine the optimal combination of the ideal mesh size and the number of high-fidelity samples for varying applications.

To sum up, the investigation gives significant knowledge into how multi-fidelity modeling works for complex systems and brings attention to the advantages of using QMLPs combined with AAFs in different situations. The results can guide future research or progress on creating multi-fidelity models for predicting complicated system performances.

7.2 Conclusions

In this thesis, a novel approach for estimating the buckling load of stiffened panels using multi-fidelity modeling based on quadratic neural networks (QNNs) with adaptive activation functions has been proposed. This study has presented the pioneering use of QNNs with adaptive activation functions for the buckling problem and multi-fidelity applications.

It has been found that the use of QNNs has enabled the model to represent a more comprehensive range of nonlinear functions than traditional ANNs, and the use of adaptive activation functions has improved the model's convergence speed and training efficiency. The results of the numerical experiments have shown that the proposed approach can effectively combine multiple fidelity levels to produce more accurate and reliable predictions of the buckling load of stiffened panels while significantly reducing the computational cost of evaluating the surrogate model. This method allows for more efficient optimization of the panel design and has the potential to be applied to a wide range of real-world problems.

7.3 Future Works

Future work in several areas might expand upon the findings of this study. One potential direction for future work is to investigate the proposed multi-fidelity modeling approach based on QNNs with adaptive activation functions on more complex problems, particularly those with high levels of nonlinearity. The ability of QNNs to represent a comprehensive range of nonlinear functions and the use of adaptive activation functions could make this approach particularly well-suited for these types of problems. Another area of interest is to explore the use of different types of multi-fidelity models, such as polynomial chaos expansion (PCE) [57] or Kriging [52, 48], and compare their performance to the QNN-based approach used in this study. In addition, it would be valuable to apply the proposed approach to a broader range of aerospace structures, such as curved stiffened panels, panels with other types of stiffeners, wing boxes or fuselages, to determine its generalizability and potential for use in more complex design problems. Furthermore, the post-buckling response and prediction of collapse load with multi-fidelity models could be considered. Moreover, the approach could be used in robust design optimization problems that include additional sources of uncertainty, such as uncertainties in the material properties or loadings, to enhance the accuracy and reliability of the predictions.

Finally, the hyperparameter optimization study conducted using Optuna can be extended to include scaling factor optimization to improve the accuracy of the predictions of the multi-fidelity models constructed using the comprehensive correction method.

REFERENCES

[1] F. A. Administration, *Metallic Materials Properties Development and Standardization (MMPDS)*, Metallic Materials Properties Development and Standardization (MMPDS): Scientific Report, National Technical Information Service, 2003.

[2] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

[3] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman, Approximation and model management in aerodynamic optimization with variable-fidelity models, Journal of Aircraft, 38(6), pp. 1093–1101, 2001.

[4] F. Aman, S. H. Cheraghi, K. K. Krishnan, and H. Lankarani, Study of the impact of riveting sequence, rivet pitch, and gap between sheets on the quality of riveted lap joints using finite element method, The International Journal of Advanced Manufacturing Technology, 67(1), pp. 545–562, 2013.

[5] O. Bacarreza, M. Aliabadi, and A. Apicella, Robust design and optimization of composite stiffened panels in post-buckling, Structural and Multidisciplinary Optimization, 51(2), pp. 409–422, 2015.

[6] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, Algorithms for hyper-parameter optimization, Advances in Neural Information Processing Systems, 24, 2011.

[7] C. Bisagni and L. Lanzi, Post-buckling optimisation of composite stiffened panels using neural networks, Composite Structures, 58(2), pp. 237–247, 2002.

[8] E. Bruhn, Analysis & design of flight vehicle structures, sr jacobs & associates, Inc., USA, 1973.

[9] J. Bu and A. Karpatne, Quadratic residual networks: A new class of neural networks for solving forward and inverse problems in physics involving pdes, in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 675–683, SIAM, 2021.

[10] M. D. Buhmann, *Radial basis functions: theory and implementations*, volume 12, Cambridge University Press, 2003.

[11] S. M. Clarke, J. H. Griebsch, and T. W. Simpson, Analysis of support vector regression for approximation of complex engineering analyses, Journal of Mechanical Design, 127(6), pp. 1077–1087, 08 2004, ISSN 1050-0472.

[12] K. Ding, Z. Zhou, and C. Liu, Latin hypercube sampling used in the calculation of the fracture probability, Reliability Engineering & System Safety, 59(2), pp. 239–242, 1998.

[13] N. Dyn, D. Levin, and S. Rippa, Numerical procedures for surface fitting of scattered data by radial functions, SIAM Journal on Scientific and Statistical Computing, 7(2), pp. 639–659, 1986.

[14] F. Fan, J. Xiong, and G. Wang, Universal approximation with quadratic deep networks, Neural Networks, 124, pp. 383–392, 2020.

[15] M. G. Fernández-Godino, C. Park, N.-H. Kim, and R. T. Haftka, Review of multi-fidelity models, arXiv preprint arXiv:1609.07196, 2016.

[16] R. Ferrari, D. Froio, E. Rizzi, C. Gentile, and E. N. Chatzi, Model updating of a historic concrete bridge by sensitivity-and global optimization-based latin hypercube sampling, Engineering Structures, 179, pp. 139–160, 2019.

[17] M. Giselle Fernández-Godino, C. Park, N. H. Kim, and R. T. Haftka, Issues in deciding whether to use multifidelity surrogates, Aiaa Journal, 57(5), pp. 2039–2054, 2019.

[18] H. M. Gomes, A. M. Awruch, and P. A. M. Lopes, Reliability based optimization of laminated composite structures using genetic algorithms and artificial neural networks, Structural Safety, 33(3), pp. 186–195, 2011.

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

[20] S. R. Gunn, Support vector machines for classification and regression, ISIS technical report, 14(1), pp. 5–16, 1998.

[21] Z.-H. Han, S. Görtz, and R. Zimmermann, Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function, Aerospace Science and Technology, 25(1), pp. 177–189, 2013.

[22] P. Hao, D. Liu, K. Zhang, Y. Yuan, B. Wang, G. Li, and X. Zhang, Intelligent layout design of curvilinearly stiffened panels via deep learning-based method, Materials & Design, 197, p. 109180, 2021.

[23] R. L. Iman, *Latin Hypercube Sampling*, John Wiley & Sons, Ltd, 2008, ISBN 9780470061596.

[24] T. M. Inc., Matlab version: 9.13.0 (r2022b), 2022.

[25] F.-X. Irisarri, F. Laurin, F.-H. Leroy, and J.-F. Maire, Computational strategy for multiobjective optimization of composite stiffened panels, Composite Structures, 93(3), pp. 1158–1167, 2011.

[26] A. D. Jagtap, K. Kawaguchi, and G. Em Karniadakis, Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks, Proceedings of the Royal Society A, 476(2239), p. 20200334, 2020.

[27] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, Journal of Computational Physics, 404, p. 109136, 2020.

[28] K. Kalnins, R. Rikards, J. Auzins, C. Bisagni, H. Abramovich, and R. Degenhardt, Metamodeling methodology for postbuckling simulation of damaged composite stiffened structures with physical validation, International Journal of Structural Stability and Dynamics, 10(04), pp. 705–716, 2010.

[29] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

[30] J. Kudela and R. Matousek, Recent advances and applications of surrogate models for finite element method computations: A review, Soft Computing, 26, pp. 13709–13733, 2022.

[31] L. Lanzi and V. Giavotto, Post-buckling optimization of composite stiffened panels: Computations and experiments, Composite Structures, 73(2), pp. 208–220, 2006.

[32] M. M. Lau and K. H. Lim, Review of adaptive activation function in deep neural network, in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 686–690, IEEE, 2018.

[33] X. Liu, P. He, W. Chen, and J. Gao, Multi-task deep neural networks for natural language understanding, arXiv preprint arXiv:1901.11504, 2019.

[34] U. K. Mallela and A. Upadhyay, Buckling load prediction of laminated composite stiffened panels subjected to in-plane shear using artificial neural networks, Thin-Walled Structures, 102, pp. 158–164, 2016.

[35] M. D. McKay, R. J. Beckman, and W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics, 42(1), pp. 55–61, 2000.

[36] V. Nair and G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in *ICML*, 2010.

[37] L. W.-T. Ng and M. Eldred, Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation, in *In Proceedings of the*

53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, p. 1852, 2012.

[38] T. Nguyen, A. Kashani, T. Ngo, and S. Bordas, Deep neural network with high-order neuron for the prediction of foamed concrete strength, Computer-Aided Civil and Infrastructure Engineering, 34(4), pp. 316–332, 2019.

[39] C. Niu and M. C. Niu, *Airframe structural design: practical design information and data on aircraft structures*, Conmilit Press, 1988.

[40] A. M. Olsson and G. E. Sandberg, Latin hypercube sampling for stochastic finite element analysis, Journal of Engineering Mechanics, 128(1), pp. 121–125, 2002.

[41] A. S. Padron, J. J. Alonso, and M. S. Eldred, Multi-fidelity methods in aerodynamic robust optimization, in *18th AIAA non-deterministic approaches conference*, p. 0680, 2016.

[42] C. Park, R. T. Haftka, and N. H. Kim, Remarks on multi-fidelity surrogates, Structural and Multidisciplinary Optimization, 55(3), pp. 1029–1050, 2017.

[43] C. Park, R. T. Haftka, and N. H. Kim, Low-fidelity scale factor improves bayesian multi-fidelity prediction by reducing bumpiness of discrepancy function, Structural and Multidisciplinary Optimization, 58, pp. 399–414, 2018.

[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, Pytorch: An imperative style, high-performance deep learning library, 2019.

[45] B. Peherstorfer, K. Willcox, and M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Review, 60(3), pp. 550–591, 2018.

[46] R. Pellegrini, U. Iemma, C. Leotardi, E. F. Campana, and M. Diez, Multi-fidelity adaptive global metamodel of expensive computer simulations, in *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 4444–4451, IEEE, 2016.

[47] R. Rikards, H. Abramovich, K. Kalnins, and J. Auzins, Surrogate modeling in design optimization of stiffened composite shells, Composite Structures, 73(2), pp. 244–251, 2006.

[48] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, Design and analysis of computer experiments, Statistical Science, 4(4), pp. 409–423, 1989.

[49] T. J. Santner, B. J. Williams, W. I. Notz, and B. J. Williams, *The Design and Analysis of Computer Experiments*, volume 1, Springer, 2003.

[50] A. Shahbazian and Y. C. Wang, Calculating the global buckling resistance of thin-walled steel members with uniform and non-uniform elevated temperatures under axial compression, Thin-Walled Structures, 49(11), pp. 1415–1428, 2011.

[51] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.

[52] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, Kriging models for global approximation in simulation-based multidisciplinary design optimization, AIAA journal, 39(12), pp. 2233–2241, 2001.

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

[54] A. Sobester, A. Forrester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, 2008.

[55] A. Spanos, *Probability theory and statistical inference: Empirical modeling with observational data*, Cambridge University Press, 2019.

[56] M. Stein, Large sample properties of simulations using latin hypercube sampling, Technometrics, 29(2), pp. 143–151, 1987.

[57] B. Sudret, Global sensitivity analysis using polynomial chaos expansions, Reliability Engineering & System Safety, 93(7), pp. 964–979, 2008.

[58] J. Tao and G. Sun, Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization, Aerospace Science and Technology, 92, pp. 722–737, 2019.

[59] G. Tezel and Y. Özbay, A new neural network with adaptive activation function for classification of ecg arrhythmias, in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 1–8, Springer, 2007.

[60] R. Vitali, R. T. Haftka, and B. V. Sankar, Multi-fidelity design of stiffened composite panel with a crack, Structural and Multidisciplinary Optimization, 23(5), pp. 347–356, 2002.

[61] K. Yoo, O. Bacarreza, and M. Aliabadi, A novel multi-fidelity modelling-based framework for reliability-based design optimisation of composite structures, Engineering with Computers, 38, pp. 595–608, 2022.

[62] K. Yoo, O. Bacarreza, and M. F. Aliabadi, Multi-fidelity robust design optimisation for composite structures based on low-fidelity models using successive high-fidelity corrections, Composite Structures, 259, p. 113477, 2021.

[63] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, Large batch optimization for deep learning: Training bert in 76 minutes, arXiv preprint arXiv:1904.00962, 2019.

[64] P. M. Zadeh, V. V. Toropov, and A. S. Wood, Metamodel-based collaborative optimization framework, Structural and Multidisciplinary Optimization, 38(2), pp. 103–115, 2009.

[65] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar, Adaptive methods for nonconvex optimization, Advances in Neural Information Processing Systems, 31, 2018.

[66] J. Zheng, X. Shao, L. Gao, P. Jiang, and Z. Li, A hybrid variable-fidelity global approximation modelling method combining tuned radial basis function base and kriging correction, Journal of Engineering Design, 24(8), pp. 604–622, 2013.

APPENDIX A

APPENDIX

A.1 Multi-layer Perceptron with Constant Activation Functions

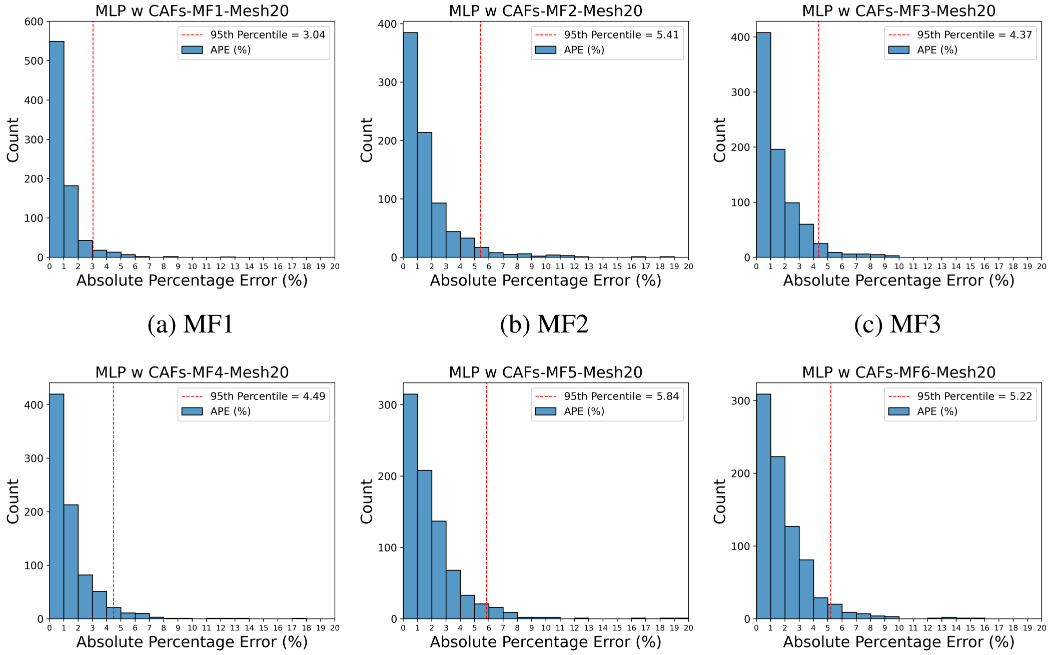
In this subsection, the performances of six multi-fidelity models (MF1, MF2, MF3, MF4, MF5, and MF6) that use multi-layer perceptron with constant activation functions are compared to that of three high-fidelity models (HF1375, HF3000, and HF4096) using the MAPE and MAE. The results are presented in Table A.1.

Table A.1: MLP with constant activation function

Metrics	MF1	MF2	MF3	MF4	MF5	MF6	HF1375	HF3000	HF4096
MAPE	0.970	1.687	1.511	1.471	1.943	1.890	1.164	0.610	0.284
MAE	0.702	1.189	0.959	1.029	1.322	1.325	0.627	0.326	0.149

MF1 has the lowest MAPE and MAE values among the six multi-fidelity models. In Figure A.1, absolute percentage error (APE) distribution histograms of the multi-fidelity models are shown. The histogram shows that the multi-fidelity models have a similar distribution of errors, with most of the errors being under 1 percent APE. 95th percentile values of the APE distribution for each of the six models (MF1, MF2, MF3, MF4, MF5, and MF6) are also given with red dashed lines in the figure. The MF1 model has the lowest 95th percentile value among the six models, indicating that it has the most accurate predictions.

When comparing MF1 to the high-fidelity models, it is noted that while MF1 is more accurate than HF1375, high-fidelity surrogates with more than 1375 samples have a MAPE of less than 1.164%. Using multi-fidelity models results in a considerable amount of computational time savings. While the MF1 model provides accurate so-



(a) MF1 (b) MF2 (c) MF3

(d) MF4 (e) MF5 (f) MF6

Figure A.1: APE distribution of MLP-based multi-fidelity models

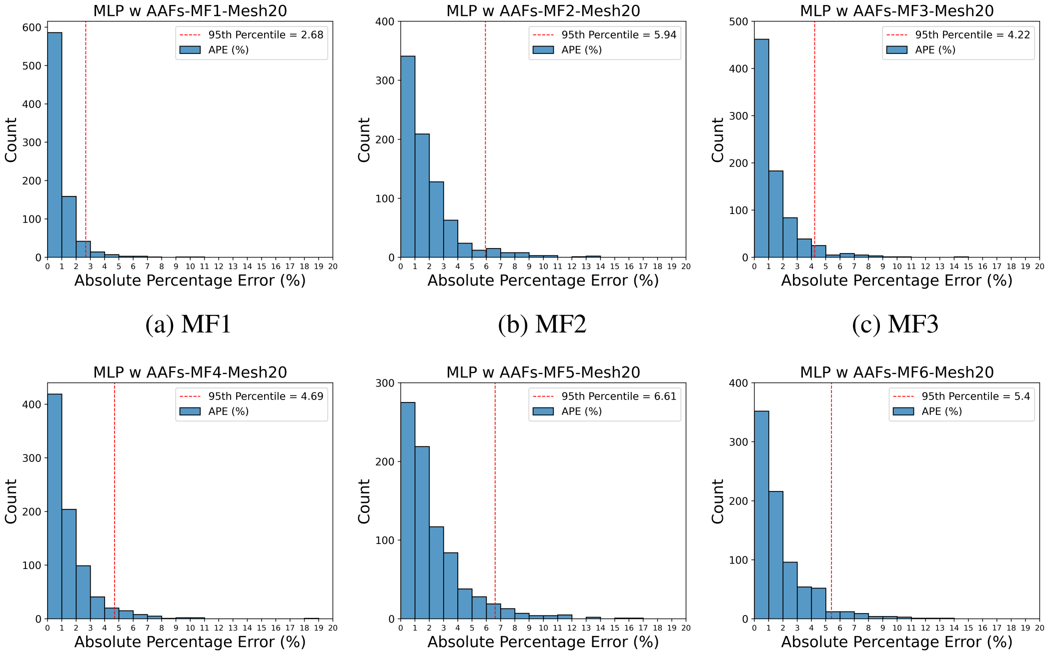
lutions for the buckling load prediction, the computational cost of MF1 is 33% of HF4096. MF1 has higher accuracy than the high-fidelity model HF1375, which has an equivalent computational cost to MF1. MF1 could be a more practical choice when computational resources are limited, as it can produce accurate predictions more efficiently.

A.2 Multi-layer Perceptron with Adaptive Activation Functions

In this study, the performances of six multi-fidelity models (MF1, MF2, MF3, MF4, MF5, and MF6) that use multi-layer perceptron with adaptive activation functions are compared to three high-fidelity models (HF1375, HF3000, and HF4096) using the MAPE and MAE. The results are presented in Table A.2.

Table A.2: MLP with adaptive activation function

Metrics	MF1	MF2	MF3	MF4	MF5	MF6	HF1375	HF3000	HF4096
MAPE	0.894	1.828	1.347	1.467	2.224	1.817	1.064	0.588	0.262
MAE	0.596	1.173	0.802	1.012	1.416	1.181	0.538	0.330	0.145



(a) MF1

(b) MF2

(c) MF3

(d) MF4

(e) MF5

(f) MF6

Figure A.2: APE distribution of Adaptive MLP-based multi-fidelity models

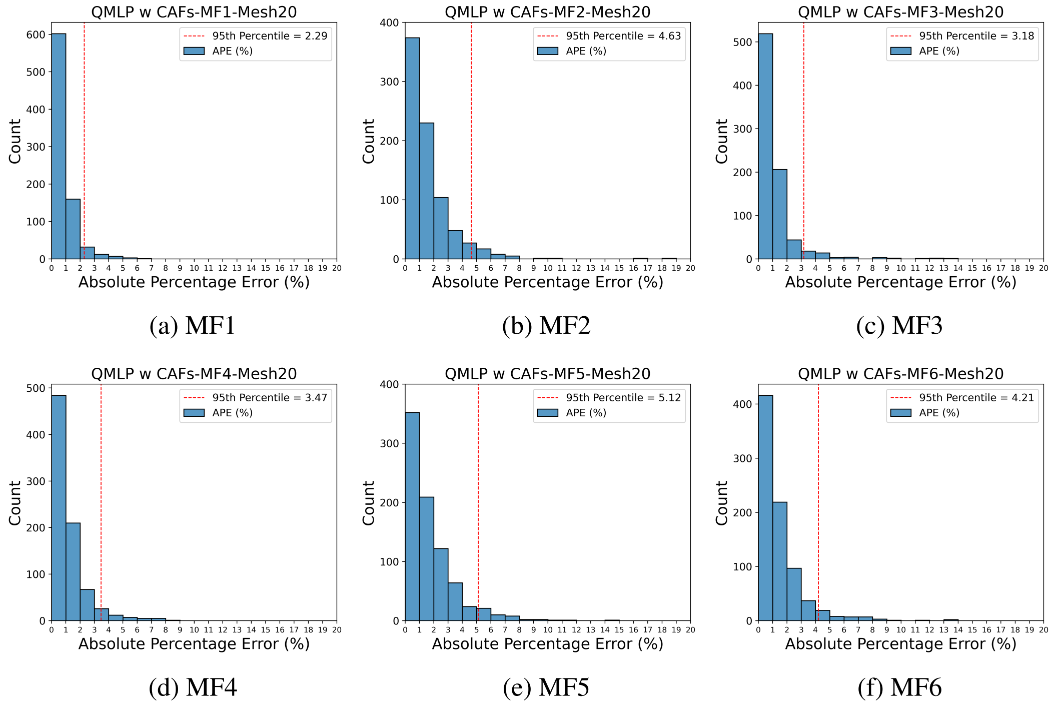
The same observations as in Appendix A.1 can be made among all multi-fidelity and high-fidelity models.

A.3 Quadratic Multi-layer Perceptron with Constant Activation Functions

In this subsection, we compare the performances of six multi-fidelity models (MF1, MF2, MF3, MF4, MF5, and MF6) that use quadratic multi-layer perceptron with constant activation functions to three high-fidelity models (HF1375, HF3000, and HF4096) using the MAPE and MAE. The results are presented in Table A.3.

Table A.3: QMLP with constant activation function

Metrics	MF1	MF2	MF3	MF4	MF5	MF6	HF1375	HF3000	HF4096
MAPE	0.793	1.576	1.090	1.148	1.732	1.426	1.094	0.539	0.220
MAE	0.499	1.044	0.725	0.796	1.172	0.960	0.542	0.294	0.120



(a) MF1

(b) MF2

(c) MF3

(d) MF4

(e) MF5

(f) MF6

Figure A.3: APE distribution of QMLP-based multi-fidelity models

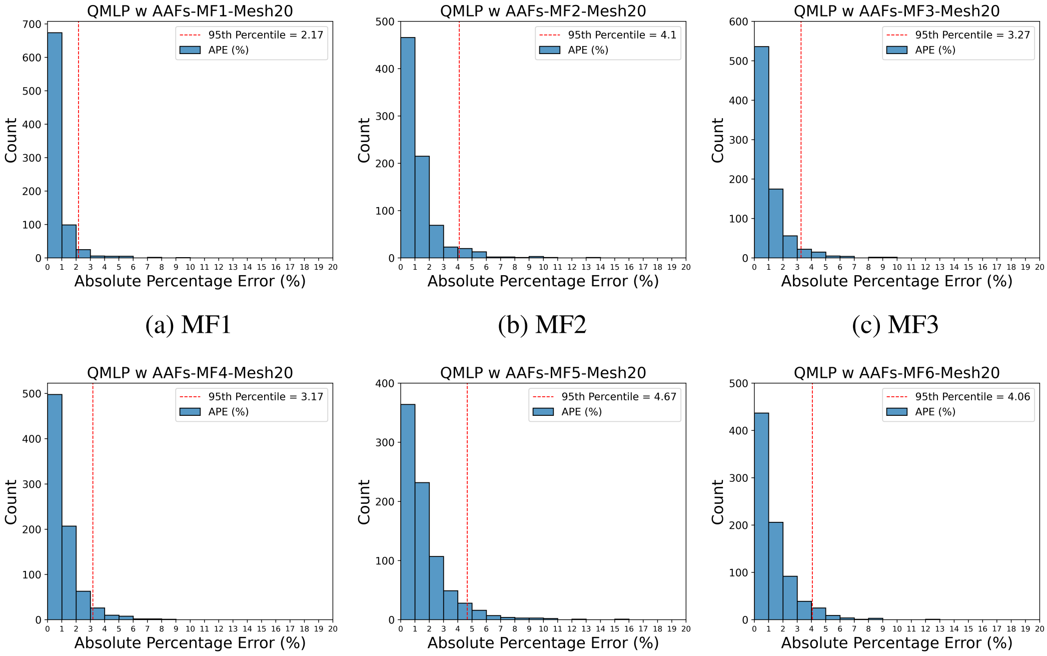
The same observations as in Appendix A.1 can be made among all multi-fidelity and high-fidelity models.

A.4 Quadratic Multi-layer Perceptron with Adaptive Activation Functions

In this subsection, the performances of six multi-fidelity models (MF1, MF2, MF3, MF4, MF5, and MF6) that use quadratic multi-layer perceptron with adaptive activation functions are compared to three high-fidelity models (HF1375, HF3000, and HF4096) using the MAPE and MAE. The results are presented in Table A.4.

Table A.4: QMLP with adaptive activation functions

Metrics	MF1	MF2	MF3	MF4	MF5	MF6	HF1375	HF3000	HF4096
MAPE	0.662	1.273	1.035	1.097	1.617	1.341	0.793	0.510	0.193
MAE	0.480	0.857	0.657	0.722	1.062	0.865	0.398	0.257	0.101



(a) MF1

(b) MF2

(c) MF3

(d) MF4

(e) MF5

(f) MF6

Figure A.4: APE distribution of Adaptive QMLP-based multi-fidelity models

The same observations as in Appendix A.1 can be made among all multi-fidelity and high-fidelity models.

APPENDIX B

APPENDIX

B.1 Comparison of Comprehensive Models with different ρ

Table B.1: Comparison of comprehensive models with different ρ

	$\rho = 0.70$	$\rho = 0.75$
MAPE	1.09	1.33
MAE	0.72	0.89

This section compares two comprehensive multi-fidelity models with different scaling factors to decide the objective function. Optimum scaling factors are found by minimizing bumpiness or minimizing error. The ρ is 0.70 when the objective function is determined as bumpiness, and the ρ is 0.75 when the objective function is an error. Table B.1 shows MAPE and MAE values on ρ values of 0.70 and 0.75 for the MF3 model with QMLP architecture.

B.2 Comparison of Different Models on Unseen Dataset

Table B.2: Comparison of different models on unseen dataset

	MF1	MF2	MF3	MF4	MF5	MF6	HF1375	HF3000	HF4096
MAPE	1.265	2.573	1.703	1.369	2.785	1.798	0.956	0.732	0.420
MAE	0.924	1.827	0.990	0.966	2.193	1.250	0.438	0.331	0.236

In this section, we compared the generalization performances of six multi-fidelity models (MF1, MF2, MF3, MF4, MF5, and MF6) to three high-fidelity models (HF1375, HF3000, and HF4096) on the new dataset given in Table 6.4 by using the MAPE and MAE. The results are shown in Table B.2

B.3 Performance of QMLP-AAF on MF1 Model with Unseen Dataset

In this section, additional finite element analyses are performed to test the ANN in new design space which involves six design parameters with interpolated dimensions given in Table B.3.

Table B.3: Variable inputs for unseen dataset

Variable	Dimensions [mm]
skin thickness	1.75, 2.25, 2.75
stringer thickness	1.75, 2.25, 2.75
stringer height	22, 27, 32
outer flange width	18, 22, 27
web width	18, 22, 27
inner flange width	22, 27, 32

Using LHS, suitable design points are determined by sampling the parameters in Table B.3 for creating the data to compare the results of additional finite element analysis and multi-fidelity model.

Table B.4: Performance of QMLP-AAF on MF1 model with unseen dataset

		sk_t	str_t	h	b_1	b_2	b_3	y_{fem}	y_{pred}	$error$
		$[mm]$	$[mm]$	$[mm]$	$[mm]$	$[mm]$	$[mm]$	$[kN]$	$[kN]$	[%]
Panel Configurations	1	1.75	2.25	27.5	27.5	27.5	32.5	36.78	37.26	1.30
	2	2.25	2.75	32.5	22.5	27.5	32.5	74.66	72.66	2.68
	3	2.25	2.75	22.5	27.5	27.5	27.5	73.41	77.93	6.17
	4	2.25	2.75	32.5	22.5	27.5	22.5	68.00	66.77	1.81
	5	2.75	2.25	32.5	17.5	27.5	27.5	101.78	94.62	7.04
	6	2.25	2.75	27.5	17.5	22.5	32.5	65.81	65.55	0.39
	7	1.75	2.75	27.5	17.5	17.5	27.5	30.73	31.63	2.91
	8	2.25	1.75	22.5	17.5	17.5	32.5	49.68	51.91	4.50
	9	1.75	2.25	22.5	22.5	17.5	27.5	29.23	28.80	1.46
	10	2.25	1.75	22.5	27.5	22.5	32.5	55.02	56.41	2.53
	11	2.75	1.75	22.5	17.5	27.5	27.5	90.51	90.69	0.20
	12	2.75	1.75	27.5	17.5	22.5	22.5	83.85	85.91	2.45
	13	2.75	2.25	32.5	22.5	17.5	32.5	95.78	101.58	6.06
	14	2.75	2.75	27.5	22.5	27.5	22.5	111.01	111.89	0.80
	15	2.75	2.25	27.5	22.5	17.5	22.5	89.36	88.03	1.49
	16	2.75	1.75	22.5	27.5	22.5	22.5	86.21	84.99	1.41
	17	1.75	1.75	32.5	27.5	22.5	22.5	27.94	27.15	2.84
	18	1.75	1.75	32.5	17.5	17.5	32.5	26.54	27.38	3.15
	19	1.75	2.25	22.5	27.5	17.5	22.5	29.22	30.03	2.77
	20	1.75	2.75	32.5	27.5	22.5	27.5	36.82	36.98	0.44

Table B.4 presents the input parameters of the additional analyses, results of the Adaptive QMLP (i.e., QMLP-AAF) based MF1 model with a mesh size of 20 mm, and finite element analysis on these input parameters, and percentage error between these results.