Prediction of Orthogonal Cutting Forces Based on Multi-fidelity Modeling

by

Jianhua Lyu

A report submitted in partial
fulfillment of the requirements for the degree of

Master of Engineering

Department of Mechanical Engineering

Faculty of Engineering and Computer Science

University of Victoria

Victoria, BC, Canada

Prediction of Orthogonal Cutting Forces Based on Multi-fidelity Modeling

by Jianhua Lyu

Supervisory Committee:

Faculty Supervisor:	Dr. Keivan Ahmadi
Committee Member:	Dr. Bosco Yu
Reader:	

ABSTRACT

Simplified analytical models of chip formation mechanics (e.g. the well-known Merchant's model) are widely used to compute the machining forces in orthogonal cutting operations. The accuracy of analytical models, however, diminishes when the cutting edge has a rounded shape, known as edge (or hone) radius, which is common for most cutting tools. Finite element (FE) simulation can be used to obtain more accurate predictions of the forces in the presence of edge radius, but FE is computationally expensive because it should numerically solve a thermo-mechanical contact problem with nonlinear material properties to model the plastic deformation and damage of the workpiece. The high computational cost of FE simulations indeed becomes crucial when the force model is used for process optimization or for online simulations in the digital twin of the machining process. In this research, we present a computationally efficient data-driven model with acceptable accuracy when compared to the FE simulation.

The presented model combines the predictions of FE simulations (i.e., high-fidelity dataset) and the predictions of the analytical model (i.e., low-fidelity dataset) and generates a new regression multi-fidelity model. The high-fidelity dataset is generated by an FE simulation in Abaqus and using Johnson-Cook constitutive equation to model the plastic deformation and damage of an aluminum workpiece during chip formation. The low-fidelity dataset is generated by Merchant's analytical model. In both datasets, the inputs are the tool rake angle and uncut chip thickness, and the outputs are the cutting and feed forces. In total, 440 data points (40 high-fidelity points and 400 low-fidelity points) are generated. Based on this dataset, a multi-fidelity model is trained and tested through the emulator-embedded neural network (E2NN) method. The Root Mean Squared Error (RMSE) is then computed between the predictions of the trained model and the predictions from the FE simulation to quantify the performance of the presented multi-fidelity model.

The results show a close agreement between the predictions of the high-fidelity and the multi-fidelity models. The computed RMSE was less than 8.5%. Yet, the accuracy would gradually improve by increasing the high-fidelity samples. Moreover, note that the computational time of a FE simulation is typically a few (~5) minutes while it is less than a second for the presented multi-fidelity model. The presented modelling approach therefore can efficiently replace high-cost FE simulations in process optimization or online simulations.

Keywords**: Orthogonal cutting; Machining forces; Edge radius; Multi-fidelity surrogates; Neural networks**

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

HF	High-fidelity
LF	Low-fidelity
Std. NN	Standard neural network
E2NN	Emulator embedded neural network
DNN	Deep neural network
h	Uncut chip thickness
α_r	Rake angle
FEM	Finite Element Model
PINN	Physics-Informed Neural Networks

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude and appreciation to my supervisor, Dr. Keivan Ahmadi, who made this work possible. His support and counsel carried me through all the stages of my project. I also want to thank my committee members for making my defense a pleasant experience and for their insightful comments and ideas.

In addition, I would like to thank my DDM teammates (Meisam, Ali, Ayberk, and Duncan) for their support and assistance with my research and report writing.

Finally, thank my parents for providing me with hope and support whenever I felt down. All of them kept me going and helped me overcome obstacles.

Chapter 1. Introduction

1.1 Orthogonal Cutting

The machining process is a complex three-dimensional operation. Orthogonal cutting is a simplified two-dimensional machining model that ignores much of the geometric complexity but describes the process well. Orthogonal cutting uses a wedge-shaped tool with the cutting-edge perpendicular to the cutting direction. As the tool is forced into the material, the chip forms a shear deformation along the shear direction in a plane at an angle ϕ_c (shear angle) to the workpiece surface[1], as shown in Fig. 1.1. The cutting edge separates the chip from the workpiece throughout the cutting process. The cutting edge will therefore form two angles with the work surface. The leading-edge surface used to guide the chip flow is orientated at a specific angle called the rake angle, α_r. As seen in Fig. 1.1, the side of the tool offers a space between the tool and the freshly generated work surface, known as the relief angle, thereby preserving the surface from wear[1]. The tool's cutting edge is positioned a specified distance below the original working surface during the cutting process. The parameter h represents the thickness of the chip prior to formation. Several relevant forces are generated during the cutting process, partly acting on the chip and partly on the tool. The study of cutting forces is therefore essential for the selection of optimal process parameters and surface finish for the process machining, as well as for extending the tool life and identifying the causes of failures in machining. Various models have been proposed to analyze orthogonal cutting, including the Merchant model[1], the Slip-Line model[2], and the Finite Element model[3]. The following paragraphs explain and compare these three models and their advantages and disadvantages.

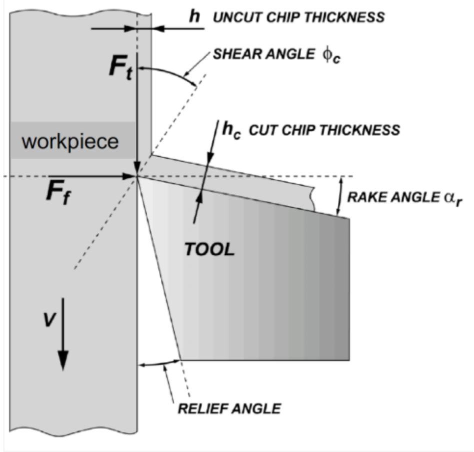


***Fig. 1.1** Chipping scheme [1]*

The Merchant model is a force-based analytical model commonly employed to forecast cutting forces, chip production, and tool wear in orthogonal cutting. It is determined by the equilibrium of forces operating on the cutting tool and workpiece[1]. Thus, Merchant model is a simple way to have a rapid computation time for estimating orthogonal cutting forces for initial estimations. But this analytical model is based on several assumptions, such as homogeneity and isotropy of the workpiece material, which predictions cannot match the real simulation results. The accuracy of analytical models, however, diminishes when the cutting edge has a rounded shape, known as edge (or hone) radius, which is common for most cutting tools.

The Slip-Line model is a semi-analytical approach focusing on workpiece plastic deformation during orthogonal cutting. It describes the material flow and stress distribution using slip-line fields. Oxley modeled an slip line field with experimental flow fields[2]. The Slip-Line model outperformed the Merchant model in terms of prediction accuracy, and it can handle materials with more complex mechanical properties, such as strain-hardening or anisotropy. Because the Slip-line model is more complicated than the Merchant model, solving slip-line field equations can be time-consuming and computationally intensive, particularly for complex geometries and materials.

Finite Element model is a numerical approach that splits the workpiece and tool into discrete components and solves each element's governing equations. Complex geometries, material characteristics, and boundary conditions are precisely modeled in FEM models[3]. The FEM accurately predicts orthogonal cutting forces, chip formation, and temperature distributions. However, the FEM process is computationally intensive. In the analysis of orthogonal cutting, the Merchant, Slip-Line, and Finite Element models reflect distinct levels of fidelity and complexity. The Merchant model is simple and rapid. However, less accurate technique, whereas the Slip-Line model is more sophisticated but provides a more accurate description of material flow and stress distribution. FEM is the most accurate and adaptable but is also the most computationally expensive and requires specialized knowledge. Therefore, a multi-fidelity model can leverage the strengths of each orthogonal cutting analytical method. The simpler models can be utilized for a significant amount and quick estimations as the LF model. In contrast, the more sophisticated models can generate small amounts of high-accuracy data as the HF model and validate the MF model. By combining the different models, computational resources can be used more efficiently, allowing for faster analysis and optimization.

In conclusion, the Merchant, Slip-Line, and Finite Element models analyze orthogonal cutting with varying degrees of fidelity, complexity, and precision. By merging these models into a multi-fidelity method, engineers can balance computational economy and accuracy, resulting in a more efficient instrument for designing, optimizing, and validating cutting processes.

1.2 Multi-fidelity modeling

After the structure of the model has been developed through the machine learning process, the data determine the model's parameters. The data's quantity and precision significantly impact the parameters' accuracy, which in turn influences the precision (also known as fidelity) of the prediction outcomes. In many practical situations, particularly engineering design, vast quantities of correct data are unavailable due to the high cost of calculating exact data. In the context of multi-fidelity modeling, the data is usually split into two parts: $\mathbf{X}_l = \left\{\boldsymbol{x}_l^{(1)}, \dots, \boldsymbol{x}_l^{(n_l)}\right\}$ (low-fidelity data) and $\mathbf{X}_h = \left\{\boldsymbol{x}_h^{(1)}, \dots, \boldsymbol{x}_h^{(n_h)}\right\}$ (high-fidelity data). These two sets of data represent different levels of fidelity or accuracy in the modeling process, and they are used in combination to enhance the overall performance of the multi-fidelity model.

The low fidelity data, $\mathbf{X}_l$, is typically obtained from a simpler model. This data is cheaper to generate in terms of computational resources and time. Examples of low-fidelity data sources include analytical models, like the Merchant model in orthogonal cutting analysis. The high-fidelity data, $\mathbf{X}_h$, is obtained from more complex, accurate, and computationally expensive simulations. This data is more costly to generate, but it provides higher accuracy and better representation of the real-world problem. Examples of high-fidelity data sources include advanced numerical models, like the Finite Element Model in orthogonal cutting analysis.

In various technical applications, multi-fidelity models have played an integral role. Multi-fidelity models have been effectively used for real-world engineering challenges such as optimization of aircraft design [4], [5], and mechanical design [6], [7]. Tyan et al. suggested a worldwide multi-fidelity modeling approach to optimize the wing design of RAE2822 supersonic and subsonic KLA-100 light aircraft by cross-comparing aerodynamic analytical data, including drag coefficient, lift coefficient, and approach angle [4]. By introducing the Nyström approximation of the sample covariance matrix, Zaytsev et al. overcome the computational burden of a multi-fidelity model based on Gaussian process regression in the presence of many samples and validate the proposed method on the problem of optimizing the shape of a high-speed turntable, which is an essential component of an aircraft engine [5]. Shi et al. introduced Co SVR, a multi-fidelity model based on Support Vector Regression-SVR[6]. They verified it using the aerodynamic study of the pressure relief valve, a standard safety valve in the industry. Le and Garnier presented the recursive co-Kriging model for deriving high-speed approximations to complex computational codes and successfully used it in the second-order turbulence simulation model MELTEM [7].

As for mathematical formulation, although there are various multi-fidelity models, the majority of those are based on the same assumption. Kennedy and O'Hagan were the first to suggest that the HF response of a system might be described as a linear trend relative to the LF response, as described in Eq. 2.1[8]. They considered that both $Z_l(\cdot)$ and the $Z_h(\cdot)$ are Gaussian processes, representing the high-fidelity and low-fidelity responses, respectively. Using auto-regressive modeling, the high-fidelity data is approximated as the low-fidelity data multiplied by a constant scaling factor ρ plus a Gaussian process $\delta(\cdot)$, which represents the difference between $\rho \cdot Z_l(\cdot)$ and $Z_h(\cdot)$. Then, Eq. 2.1 is used to predict HF values with samples $\mathbf{X}_h$ by the LF model. Zhang et al. proposed the least squares multi-fidelity model (LR-MFS) based on Eq. 2.1[9]. The approach is based on least squares regression, which aims to minimize the sum of the squares of the differences between the observed (high-fidelity) and predicted (surrogate) outputs. Zhou et al. derived the low-fidelity model $Z_l(\boldsymbol{x})$ and the difference function $\delta(\boldsymbol{x})$ using the Support Vector Regression (SVR) method[10]. Durantin et al. designed the co-Radial Basis Function (RBF) multi-fidelity model using the radial basis function technique, where $Z_l(\boldsymbol{x})$ and $\delta(\boldsymbol{x})$ are expressed as linear combinations of radial basis functions with a polynomial deviation[11]. Forrester et al. proposed the classical co-Kriging model[12]. They estimated the correlation coefficients of the LF model $Z_l(\boldsymbol{x})$ and the difference function $\delta(\boldsymbol{x})$ by optimizing the maximum likelihood function. It has also been argued that the projection relationship between the LF and HF models is nonlinear: $Z_h(\boldsymbol{x}) = \pi(Z_l(\boldsymbol{x}))$, where $\pi(\cdot)$ is a nonlinear function. The authors of [13] considered that the functions $\pi(\cdot)$ and LF model are artificial neural network and isothermal finite element analysis models, respectively, and that the multi-fidelity model is created by learning the pertinent parameters from the data.

$$Z_h(\boldsymbol{x}) = \rho \cdot Z_l(\boldsymbol{x}) + \delta(\boldsymbol{x}) \tag{2.1}$$

Even though these multi-fidelity models typically have well-defined forms and improved interpretability, they are predicated under certain assumptions, including the relationship between LF and HF response values, the distribution of data, and the nested data structure. These assumptions are rarely met in practice, so the applicability of these models to real engineering problems is diminished.

In addition to the models above, there are different kinds of multi-fidelity models that do not have a mapping relationship between high-fidelity and low-fidelity models. Jacobs et al. built a Bayesian support vector machine model as a LF model using LF training samples, computed the HF responses of its support vectors as HF training samples, and then created a Bayesian support vector machine for HF[14]. Consequently, this technique must fulfill one condition: the HF system must be known. However, most practical engineering issues know only the LF and HF training samples. Hence this method can only be used for specific problems.

Orthogonal cutting forces are influenced by various factors, potentially resulting in a high-dimensional problem. However, co-kriging assumes that data follows a Gaussian process, leading to many computations on covariance matrix inversion operations as the problem dimension increases. Moreover, orthogonal cutting forces exhibit nonlinearity due to the complex interplay between tool geometry and cutting speed factors. This nonlinearity may pose a challenge for LR-MFS. Different analytical models exist for orthogonal cutting force prediction, indicating various fidelity levels for the analysis. E2NN can adapt to varying fidelity levels and embed different fidelity models into its structure, reducing the risk of overfitting compared to other neural network structures. Thus, E2NN offers a more suitable approach for addressing the complexities of orthogonal cutting force prediction in high-dimensional and nonlinear problems.

1.3 The proposed approach in this work

In order to create a high-performance multi-fidelity model for orthogonal cutting analysis, low-fidelity data is obtained from the Merchant model. In contrast, high-fidelity data is sourced from the Finite Element model. The inputs for this analysis are the rake angle and uncut chip thickness of the orthogonal cutting, with the outputs being the orthogonal cutting forces. A multi-fidelity model called the E2NN model is constructed by combining low-fidelity and high-fidelity data. This multi-fidelity model yields results that closely approximate the high-fidelity output, providing increased accuracy while simultaneously reducing computation costs. Notably, the E2NN model achieves an acceptable root mean square error (RMSE) of just 8.5%, highlighting its effectiveness in predicting cutting forces in orthogonal problems.

The sections are organized as follows: Chapter 1 gave an overview of orthogonal cutting and introduced various multi-fidelity models. Chapter 2 is devoted to modeling, which consists of Kriging, co-Kriging, and E2NN models. The simulation results are presented in Chapter 3, including the 1D and 2D examples of the benchmark functions and prediction of orthogonal cutting forces. Finally, chapter 4 is the conclusion of the work.

Chapter 2. Classical multi-fidelity modeling

The concept of variable fidelity and multi-fidelity models was proposed by scientists as early as the 1980s. Numerous studies on multi-fidelity models have been conducted in engineering design and other related fields. Based on the practical application history of multi-fidelity problems, reasonable hypotheses and research methods have been proposed by numerous scholars. In multi-fidelity modeling, an important position is occupied by the assumption of a linear trend between HF and LF models. Multi-fidelity models with non-linear assumptions have also been the focus of a significant number of researchers. However, the belief in linear trend predominates in multi-fidelity problems, with models based on this assumption performing better in various situations. As a result, several classic multi-fidelity models will be presented below.

2.1 Kriging and Co-Kriging

2.1.1 Kriging

The Kriging method is a well-known local interpolation method that enables optimal estimation of regionalized variables in a finite region that is unbiased. The fundamental assumption of Kriging interpolation is that the attribute value of a point is related to and can be derived from the attribute value of its neighboring points. Kriging's interpolation is based on the principle that the difference between the attribute values of two points is proportional to the existence of a range of distances between them. The mathematical form of Kriging's interpolation's fundamental assumptions is as follows:

The estimated value $z'0$ of an unknown point in space (x0, y0) is a weighted sum of some known point values, as shown in Fig. 2.1.

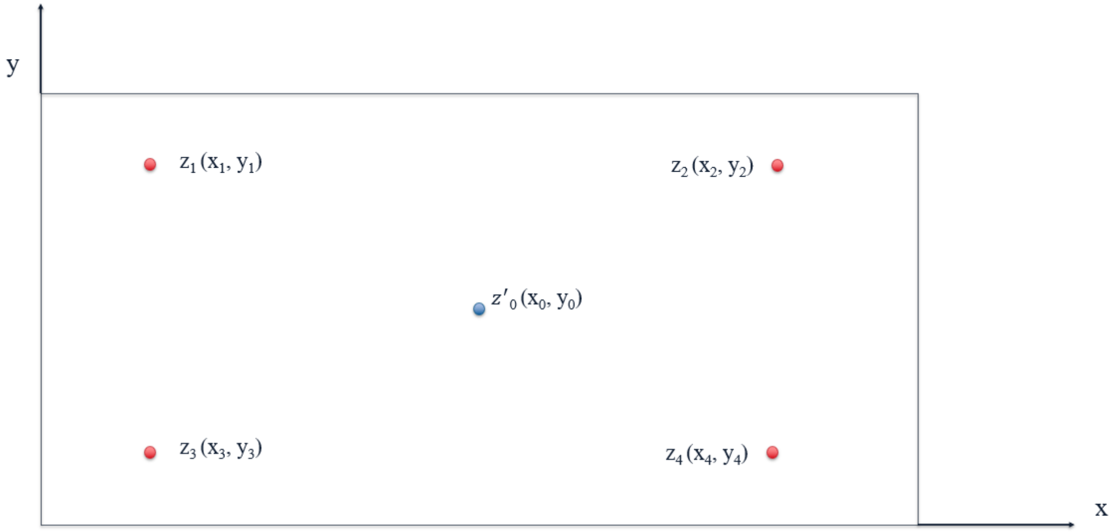


Fig. 2.1 *Image interpretation kriging approaches*

$$z'_0 = \sum_{i=1}^{n} z_i \cdot w_i \qquad (2.2)$$

where n is the number of points capable of determining an attribute value (generally a certain number of known points around the point are taken), z_i represents the attribute value of the i_{th} known point, and w_i represents the weight of the i_{th} known point).

The key component of Kriging is the semivariance matrix, which quantifies the spatial variability or dissimilarity between data points as a function of their separation distance. The semivariance is a measure of the dissimilarity between pairs of data points, and it is typically represented as half the squared difference between the values of the variable at two locations[15].

2.1.2 Co-Kriging

Forrester's Co-Kriging method addresses multi-fidelity problems by combining low- and high-fidelity data to create an accurate surrogate model. It constructs separate Kriging models for low-fidelity data and the discrepancy function, representing the difference between low- and high-fidelity data. The co-Kriging model combines both Kriging models to make predictions for the high-fidelity response at unknown locations. This method improves the surrogate model's accuracy while reducing computational costs[12]. The correction processes of the multiple sets of data method named co-Kriging, described in Chapter 8 of [12].

The Co-Kriging model assumes that the data follows a Gaussian process; however, when the data exhibit significant fluctuations, the model's performance may degrade. Data is likely to change drastically in real-world problems, potentially leading to unexpected performance outcomes. Moreover, the Co-Kriging model involves numerous computationally expensive covariance matrix inversion operations, making it unsuitable for tackling complex practical problems.

Neural networks serve as an alternative approach in this report. The neural network structure is made up of an input layer, a hidden layer, and an output layer, as shown in Fig. 2.2. By adjusting the number of neurons, hidden layers, and the activation function before the output layer, and we can enhance the results and determine the ideal weights and biases. Although a large amount of training data often leads to more accurate results, issues like overfitting and extended fitting times may occur when using more hidden layers and neurons. We employ a physics-informed neural network as the foundational theory to address the challenge of limited training samples.

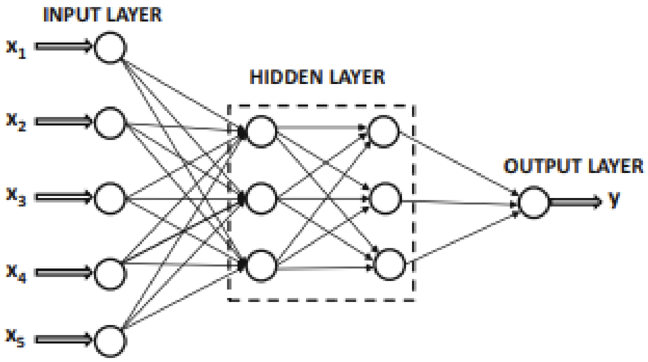


Fig. 2.2 *Structure of Neural Networks[16]*

2.2 Physics-Informed Neural Network

A group of researchers from the Department of Applied Mathematics at Brown University introduced Physics-Informed Neural Networks (PINNs) in 2019, with their findings published in the Journal of Computational Physics[17]. By incorporating the physics equation as a constraint in the neural network, PINNs ensure that the fitting results adhere to physical principles. The difference between successive iterations of the physics equation is integrated into the neural network's loss function, which trains the equation at each step. Consequently, the neural network optimizes the network's loss function and the discrepancy between each iteration of the physical equations throughout the training process, ensuring that the final trained model complies with a specific set of physical laws. This approach accelerates the training speed of the neural network and enables it to achieve high accuracy with a limited training set.

PINNs can handle various challenges related to partial differential equations, such as solving equations, inverting parameters, discovering models, controlling, and optimizing. As depicted in Figure 2.3, a simple, fully connected neural network and a partial differential equation are shown on the right side of the figure, respectively. The neural network is then employed to estimate the global loss function[17]:

$$\text{Total loss function} = \text{left part loss function} + \lambda \cdot \text{right part loss function}$$

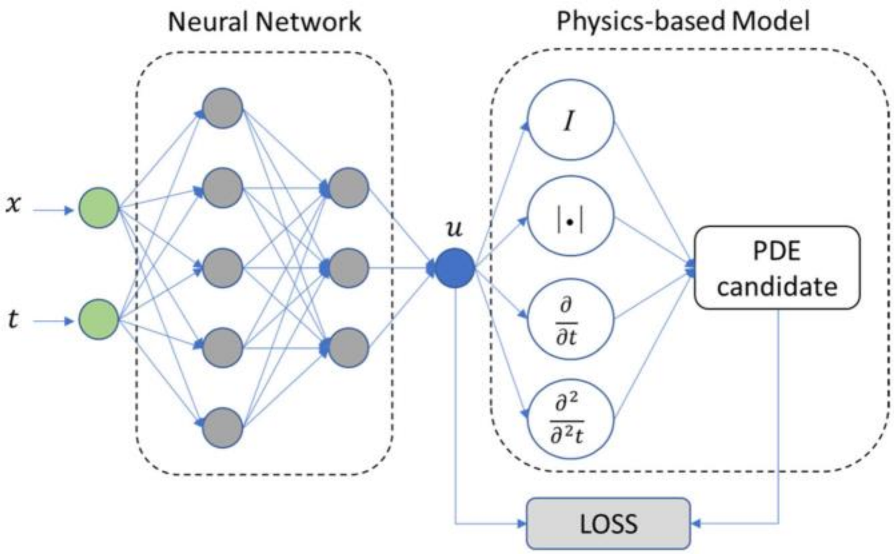


Fig. 2.3 *General structure of PINN[18]*

Liu and Wang investigated and experimented with multi-fidelity data using PINNs in 2019[19]. These models reduce the demand for costly high-fidelity experimental datasets by integrating physical constraints. They train a surrogate model using numerous low-cost, low-fidelity datasets and combine a limited number of high-fidelity data points within a neural network. This method identifies the differences between high and low-fidelity training models and incorporates physical information constraints to speed up the neural networks' fitting process. As shown in Figure 2.4, the model effectively captures the relationship between high and low-fidelity models in various applications, such as two-dimensional heat transfer, phase transition, and dendritic growth problems[19].

However, when the low-fidelity or physically informed model does not align well with the actual model, PINNs struggle to produce accurate predictions. In real-world engineering situations, low-fidelity data often exhibit only local relationships with high-fidelity data, or multiple low-fidelity models may correspond to a single high-fidelity model. When different fidelity levels display varying levels of uncertainty, it becomes difficult for PINNs to capture the actual model's global behavior. The embedded simulator model is introduced to improve PINNs' applicability to real-world engineering challenges.

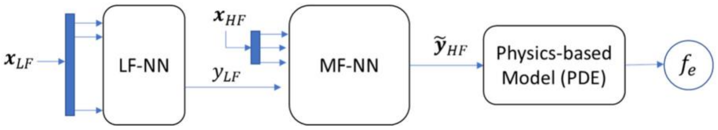


Fig. 2.4 *Architecture of MF-PINN. Samples are selected using a design of experiments method such as Latin hypercube sampling[18]*

2.3 Emulator embedded neural network (E2NN) with multi-fidelity data

Using the PINN method, we have discovered that incorporating a physical information model into a neural network can accelerate neural network fitting and address the issue of limited experimental data. Similarly, in multi-fidelity neural networks, low-fidelity models can be embedded within the neurons of the hidden layer rather than using physics models[18]. As a result, numerous low-cost, low-fidelity models can capture the impact of variables on outcomes. By replacing neurons with embedded low-fidelity models in the neural network and combining them with other neurons to enhance the signal, robust neural network fitting is achieved, leading to accurate neural network fitting and prediction.

Based on these two neural network models, the predicted value aims to closely approximate the actual response Z_h(x) for any given input. The structure of the E2NN model is based on the design proposed by Beachy and Bae in 2021[18]. Fig. 2.5 displays three main components: the input, hidden, and output layers. The subsequent section will provide a more detailed explanation of the connections between these parts.

Fig. 2.5 presents a fully connected neural network. The simulator embedded within the neuron contains multiple low-fidelity models combined with other neurons to capture information from the high-fidelity metric model for quick fitting and accurate output prediction. This low-fidelity metamodel is integrated within the neural network rather than concatenated in the output. The back-propagation of the neural network allows the embedded simulator to update the weights and biases at each iteration. Moreover, during training, each embedded simulator can connect with the high-fidelity model, capturing essential details and disregarding less important information. This approach addresses the issue where a single low-fidelity model cannot establish a global connection with the high-fidelity model.

As depicted in Figure 2.5, the E2NN structured neural network can accept arbitrary variables as embedded simulators and create connections for high-fidelity models with varying degrees of variation by distinguishing essential data from unimportant data. According to the authors' paper, the position of the embedded simulator for each problem may differ. Generally, authors choose two hidden layers, placing the less accurate simulator in the first and the more detailed simulator in the second[18]. This is because the lower fidelity model with higher accuracy will work more effectively when the activation function is applied to the output. The less accurate simulator can be placed in the first hidden layer for multi-dimensional mapping to capture different data types. Suppose the accuracy of an emulator is unknown. In that case, the same emulator can be embedded in multiple layers, and instances embedded in inappropriate layers will be disregarded due to regularization after training[18].

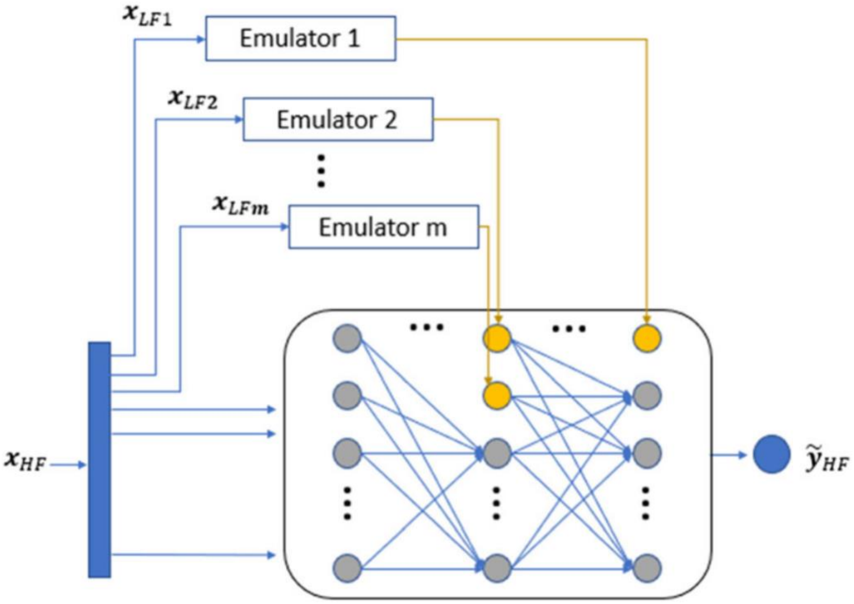


Fig. 2.5 *Emulator embedded neural network[18]*

Chapter 3. Numerical experiments

To scientifically validate the practicability of the multi-fidelity model proposed in this report, we compared the co-Kriging, standard NN, and E2NN models on two multi-fidelity benchmark functions. Every experiment was performed on the same computer and DACE toolkit[20] was used to implement co-Kriging within the multi-fidelity model. We compared the force predictions from Merchant's model and the Finite Element Method (FEM). Finally, the presented model combines the predictions of FE simulations (i.e., high-fidelity dataset) and the predictions of the analytical model (i.e., low-fidelity dataset) and generates a new regression multi-fidelity model named E2NN. We also developed a high-accuracy DNN model using a larger number of high-fidelity samples to assess the performance of the multi-fidelity model when utilizing fewer high-fidelity samples.

3.1 Numerical examples

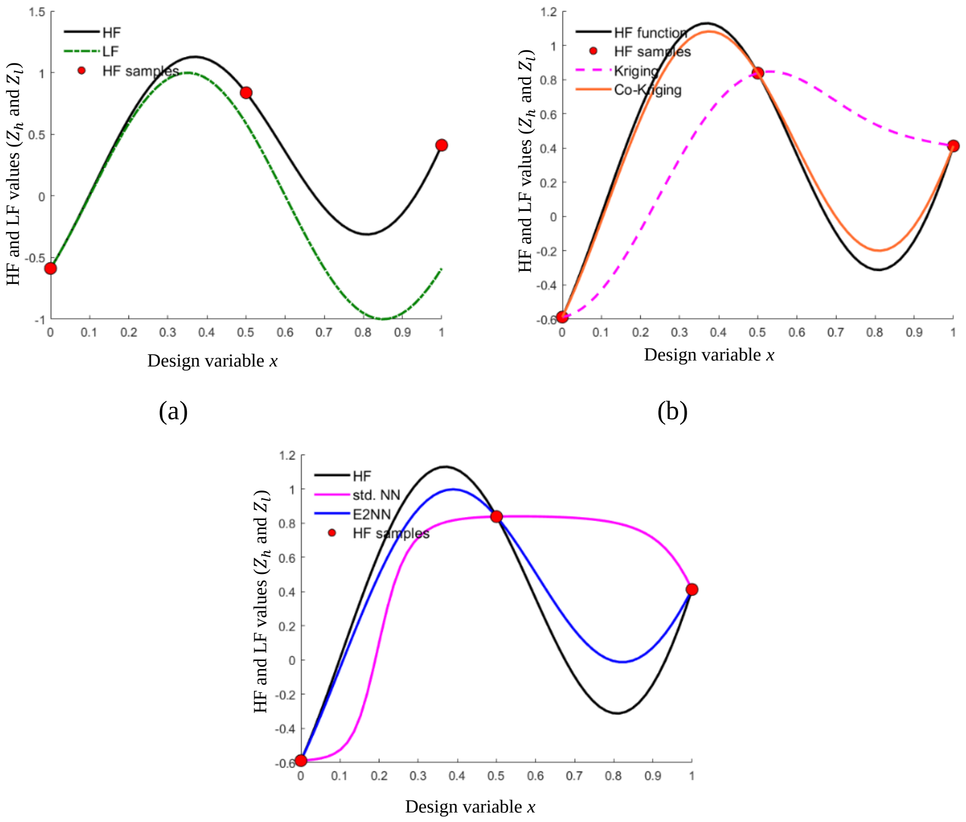
3.1.1 One-dimensional analytical example

This report's test functions were taken from[6]. Each function set includes both high-fidelity and low-fidelity datasets. The low and high fidelities functions are as follow:

$$Z_h\ (x) = \sin(2\pi(x\text{-}0.1)) + x^2 \qquad (3.1)$$

$$Z_l\ (x) = \sin(2\pi(x\text{-}0.1)) \qquad x \in [0.0\ \ 1.0] \qquad (3.2)$$

The high-fidelity samples discussed in this report are much smaller than the low-fidelity samples. High fidelity dataset was generated at 3 points: x = {0, 0.5, 1.0}. We generated 200 low-fidelity data points evenly distributed in the range of x = 0 to 1. Fig. 3.1a depicts the high-fidelity dataset and the low-fidelity dataset. They share the same pattern of variation. When only three high-fidelity samples are used, the kriging method cannot capture the changing trend of this function, resulting in a vast prediction error. In contrast, the co-Kriging model can learn the changing trend with high accuracy but has minor errors due to the small sample size. As shown in Figs. 3.1 b and c, the standard NN predicts similar results to those predicted by the Kriging model. For E2NN with the same network structure, however, the low-fidelity model is embedded as a simulator in the first neuron of the second hidden layer, thereby resolving the problem of low-fidelity and high-fidelity model bias in the process of the neuron transmission signal. The error is still more significant than the prediction error of the co-Kriging model, even though the E2NN multi-fidelity model captures the changing trend of the high-fidelity model. By adjusting the number of hidden layers and neurons in the neural network, we can later determine the optimal settings to improve E2NN's prediction accuracy.



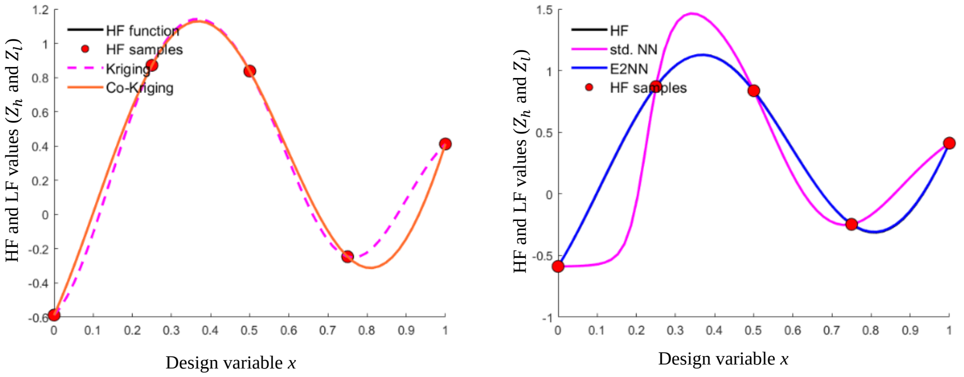
(a)

(b)

(c)

***Fig. 3.1** One-dimensional example. (a) HF and LF models. (b) Various modeling with three HF samples. (c) Std. NN and E2NN models.*

To determine the stability of the E2NN and co-Kriging models, we will increase the number of high-fidelity data from 3 to 5 (x = 0, 0.25, 0.5,0.75, 1.0) under the condition that the low-fidelity data set is 200, and the predicted results are shown in Fig. 3.2. The prediction curves of the E2NN and co-Kriging models overlap with those of the high-fidelity model. The Kriging and standard NN models can also capture the trend of the high-fidelity model, but their prediction accuracy is inferior to that of the E2NN and co-Kriging models. In the cases of both low-dimensional and minor HF datasets, the E2NN and co-Kriging models can capture the actual system characteristics and predict the same outcomes as the high-fidelity model.



(a) (b)

Fig. 3.2 *One-dimensional example. (a) Various modeling with five HF samples. (b) Std. NN and E2NN models.*

3.1.2 Two-dimensional analytical example

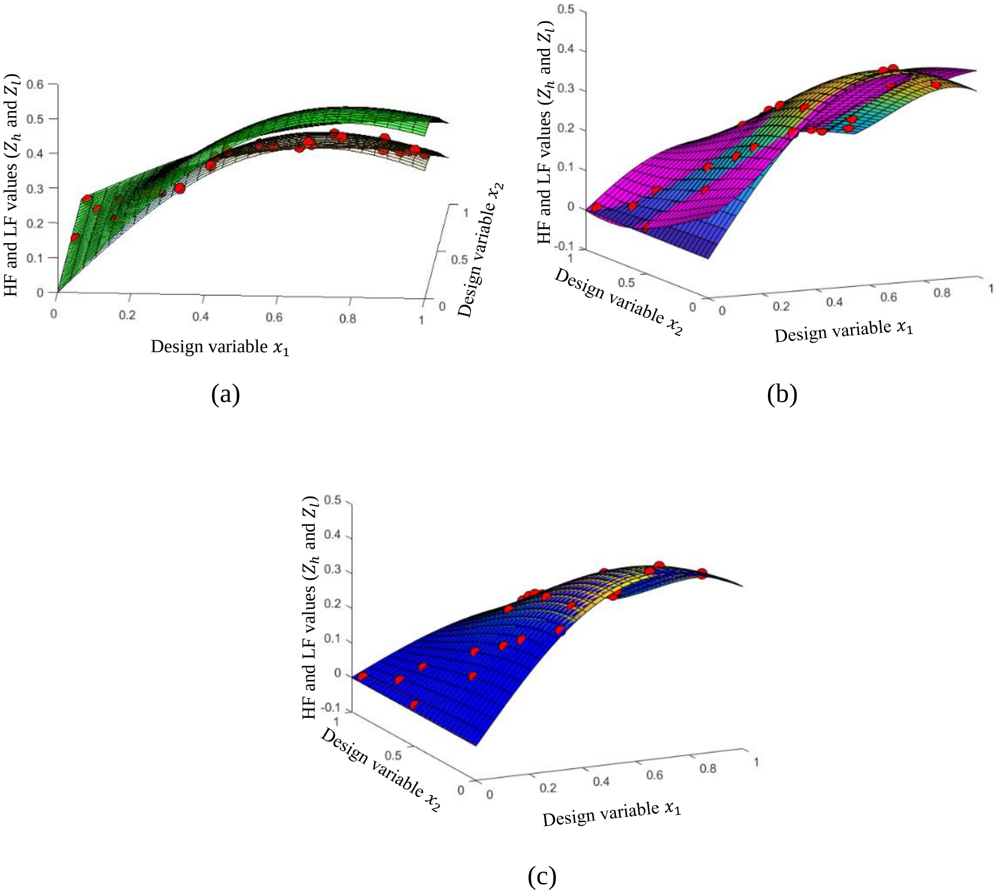
In high-dimensional problems, co-Kriging becomes computationally challenging due to extensive inverse matrix calculations, and its assumption that data adheres to a Gaussian process might not be suitable for real-world situations, resulting in suboptimal performance. Therefore, this section will cover the tests of Std. NN and E2NN for two-dimensional functions. The functions of multi-fidelity are described below:

$$Z_h\ (x) = x_1 \cdot \exp\ (-x_1^2 - x_2^2) \tag{3.3}$$

$$Z_l\ (x) = x_1 \cdot \exp\ (-x_1^2 - x_2^2) + \frac{x_1}{10} \qquad x \in [0.0\ \ 1.0] \tag{3.4}$$

The Std. NN consists of two hidden layers, each with 20 neurons. For the E2NN model, the $Emulator_1^1 = y_{LF}(x_1,\ x_2)\,|_{x_1=0.5}$ and $Emulator_2^1 = y_{LF}(x_1,\ x_2)\,|_{x_2=0.5}$ are embedded in the first hidden layer as the first two neurons, respectively. Then, we put the complete LF model into the second hidden layer as the first neuron, like $Emulator_1^2 = y_{LF}(x_1,\ x_2)$. Twenty-five high-fidelity data sets and 900 low-fidelity data sets are substituted for the two models, respectively. The results of the two sets of models' predictions are depicted in Fig. 3.3.

It is evident from Fig. 3.3a that the low-fidelity and high-fidelity models differ significantly. Fig. 3.3b shows that only a few HF points are located at the Std. NN predicted surface(magenta), calculating the RMSE (0.0018) for a 30 x 30 point grid of the test samples within the function domain. As shown in Fig. 3.3c, the E2NN model with the same set of training samples is well-matched to the HF surface; its RMSE (3.1597e-05) is significantly smaller than that in Std. NN. The orthogonal cutting forces are influenced by various independent parameters, including rake angle, uncut chip thickness, edge radius, and more, potentially resulting in 2D, 3D, or even high-dimensional problems. Therefore, we will use the E2NN model only in the next section.



(a)

(b)

(c)

Fig. 3.3 *Two-dimensional example: Std. NN and E2NN fitted with 25 HF samples. (a) HF surface, LF surface (green) and 25 HF samples. (b) Std. NN surface (magenta). (c)E2NN surface(blue)*

3.2 Multi-fidelity modeling of orthogonal cutting

3.2.1 Low-fidelity model

This work uses the Merchant's model as the LF model to predict orthogonal cutting forces[1]. The rake angle, α_r(deg) and the uncut chip thickness, h(mm), are the independent variables. The cutting width, b, is set at 1mm in the simulation. Below are several formulas that can be used to determine orthogonal cutting parameters for aluminum.

The feed force (F_f) and tangential force (F_t) are expressed as linear functions of uncut chip thickness (h) and width (b):

$$F_t = \mathrm{K}_{tc}\, b\, h + k_{te}\, b \tag{3.5}$$

$$F_f = \mathrm{K}_{fc}\, b\, h + k_{fe}\, b \tag{3.6}$$

where K_{tc}, K_{fc} are cutting force coefficients, depending on τ_s, shear stress, ϕ_c ,shear angle, β_a,friction angle of the material in Merchant’s model:

$$\mathrm{K}_{tc}\ [\mathrm{N/mm}^2] = \tau_s \frac{\cos(\beta_a - \alpha_r)}{\sin\phi_c \ \cos(\phi_c + \beta_a - \alpha_r)} \tag{3.7}$$

$$\mathrm{K}_{fc}\ [\mathrm{N/mm}^2] = \tau_s \frac{\sin(\beta_a - \alpha_r)}{\sin\phi_c \ \cos(\phi_c + \beta_a - \alpha_r)} \tag{3.8}$$

where k_{te}, k_{fe} are edge cutting force coefficients, τ_s, ϕ_c , β_a, are determined experimental as function of α_r *and* h:

$$k_{te}[\mathrm{N/mm}] = 39.2724 - 1.0756\alpha_r \tag{3.9}$$

$$k_{fe}[\mathrm{N/mm}] = 47.9158 - 1.4802\alpha_r \tag{3.10}$$

$$\tau_s[\mathrm{MPa}] = 122.512 + 151.492h + 1.7201\alpha_r \tag{3.11}$$

$$\phi_c\ [\mathrm{deg}] = 13.3202 + 8.498h + 0.9142\alpha_r \tag{3.12}$$

$$\beta_a[\mathrm{deg}] = 25.423 + 6.1526h + 0.3559\alpha_r \tag{3.13}$$

3.2.2 High-fidelity model

In this report, we will use ABAQUS software to construct an orthogonal cutting simulation model that outputs the effect of the tool rake angle and chip thickness on tangential and feeds force. Johnson-Cook constitutive equation is used to model the plastic deformation and damage of an aluminum workpiece during chip formation. The following assumptions are made to simulate the orthogonal cutting process with the finite element method:

1. Machining operation is a plane strain problem (2D).
2. Sliding friction at tool-chip interface is assumed.
3. Material properties are independent of temperature.

Fig. 3.4 displays the mesh models of the tool and workpiece. Table 3.1 to 3.4 outlines the characteristics of the tool and the workpiece.

Table 3.1*: Physical properties of workpiece[21]*

S.No	Parameter	Al2024
1	Density(kg/m^3)	2690
2	Young’s Modulus (GPa)	74
3	Poisson ratio	0.29
4	Thermal conductivity(W/m^0C)	121
5	Specific heat(J/Kg/^{0}C)	891.5

Table 3.2*: Johnson-Cook parameters for workpiece[21]*

A (MPa)	B (MPa)	C	n	m	Tm(^{o}C)	$T0$(^{o}C)
369	684	0.0083	0.73	1.7	520	20

Table 3.3*: Johnson-Cook fracture model parameters for workpiece[21]*

d1	d2	d3	d4	d5	ε_0
0.13	0.13	-1.5	0.011	0	1

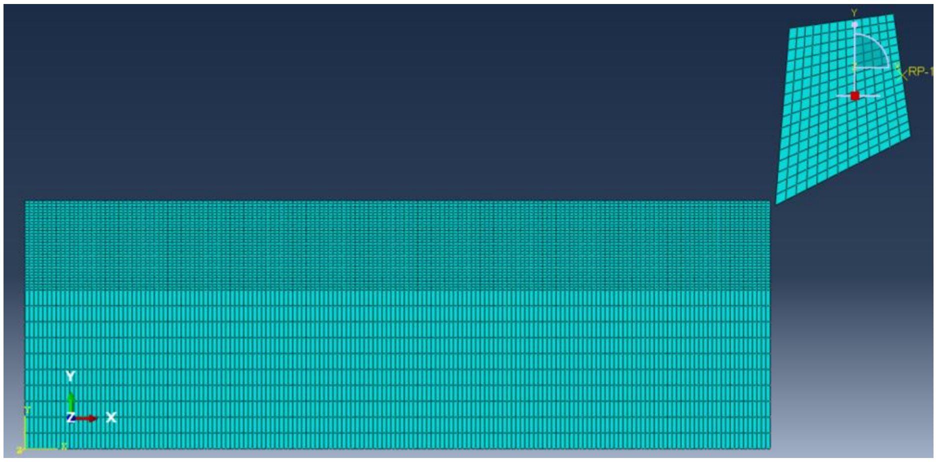


Fig. 3.4 *Orthogonal cutting simulation (F.E. mesh).*

After adjusting the remaining parameters, the simulation results are displayed in Fig. 3.6. The simulated tangential force under uncut chip thickness (0.1mm) and rake angle (4 deg) is 93.66 N. From Merchant’s model, the low fidelity value of Ft is 92.61 N. The high and low fidelity values are very close because the tangential force derived from eq.3.5 to 3.13 is very close to the actual value if the cutter edge is perfectly sharp.

As shown in Fig. 3.7 and 3.8, when the tool edge radius is increased to 0.1mm, the simulated tangential force increases to 176.99 N at the same h and α_r, significantly different from the force computed by Merchant’s model.

In machining processes, most of the cutting tool's edges are not always kept sharp. Therefore, we will create a high-fidelity simulation dataset with a 0.1 mm rounded tool edge. This adjustment will help ensure that our simulation closely mirrors the actual conditions and performance of the cutting tool in practical applications.

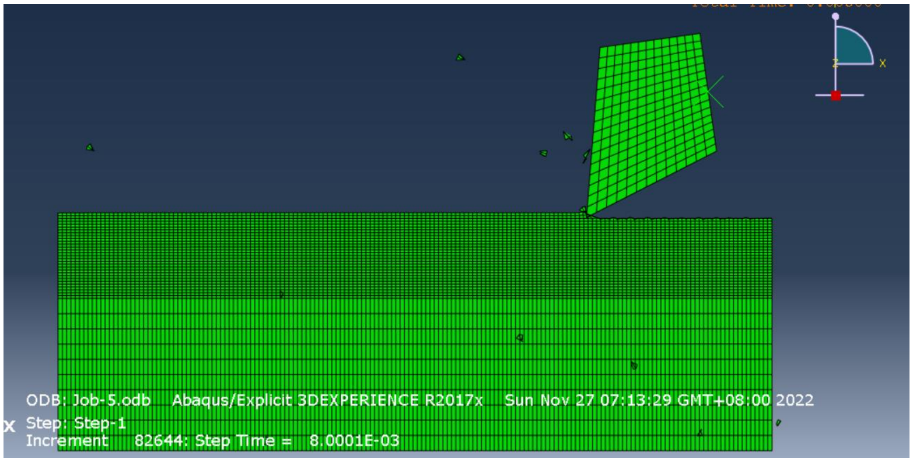

```
ODB: Job-5.odb    Abaqus/Explicit 3DEXPERIENCE R2017x    Sun Nov 27 07:13:29 GMT+08:00 2022
Step: Step-1
Increment    82644: Step Time =    8.0001E-03
```

Fig. 3.5 *Orthogonal cutting simulation.*

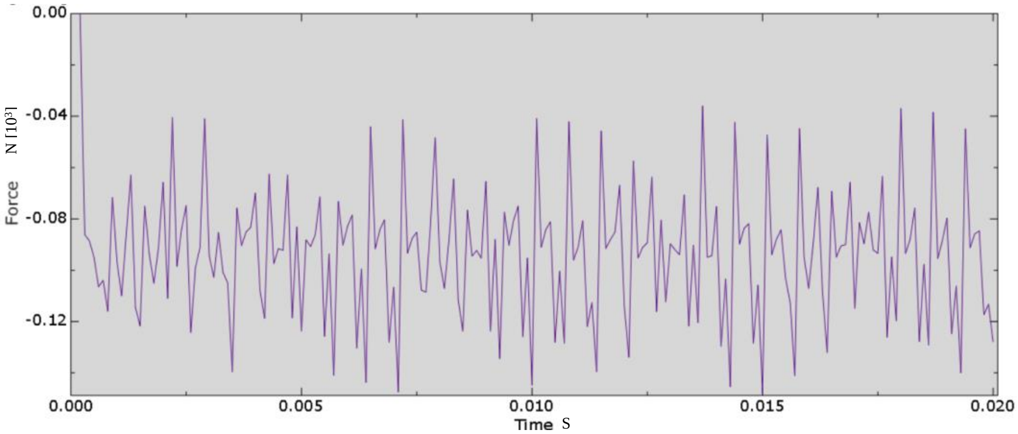


Fig. 3.6 *Tangential force of Orthogonal cutting simulation in sharp tool edge.*

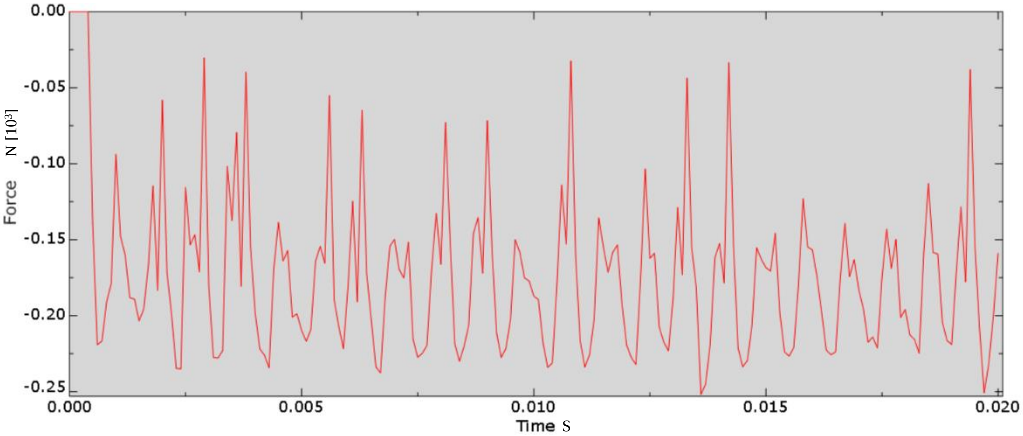


Fig. 3.7 *Tangential force of Orthogonal cutting simulation in round tool edge.*

Fig. 3.8 *Sharp (left) and round (right) tool edges*

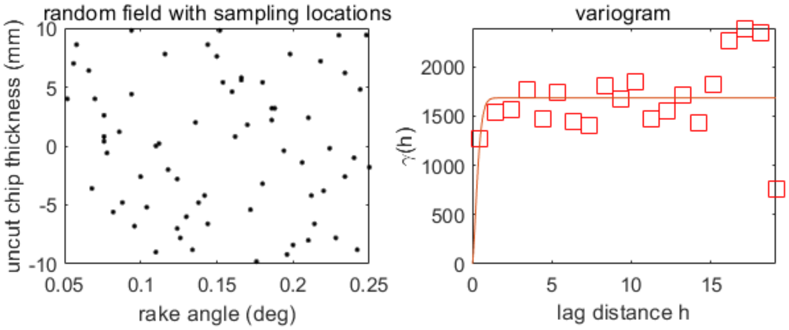
3.2.3 Surrogate model of orthogonal cutting forces using HF data

Kriging and DNN are used to develop a surrogate model of the cutting forces based on HF training data obtained from FEM. The following section will use these surrogates to study the prediction accuracy and training efficiency of the E2NN surrogate model trained by a combination of LF (Merchant's model) and HF (FEM) data.

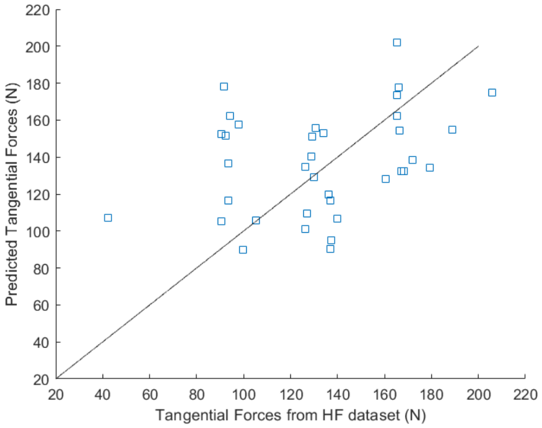
The FEM was executed at 106 combinations of rake angle and uncut chip thickness values using a cutting tool with a 0.1 mm edge radius, the results for generating the HF data set required for training the surrogate models. Latin Hypercube[22] was used to determine the optimal distribution of the training points, and the resulting distribution is shown in Fig. 3.9 (a).

3.2.3.1 Surrogate with Kriging method

Shown in Fig. 3.9. (b) is the correlation between each pair in the dataset and their distance, i.e., the variation of the forces at any two data points in terms of their distance. As shown in this figure, the variation of the force values for the distance between forces does not follow a Gaussian process, which is necessary for the Kriging method. As a result, the predictions of the developed Kriging surrogate are not close to the actual forces, as shown in Fig. 3.9 (c). The DNN method does not require such correlation between data points and thus is expected to perform better, which will be confirmed in the next section.



(a) (b)



(c)

Fig. 3.9 *Kriging method:(a) Random field with sampling locations. (b) Relationship of distance matrix and semi-covariance matrix (c) Predicted tangential forces VS. Tangential forces from HF model*

3.2.3.2 Deep Neural Network (DNN) surrogate

In the DNN model, the input layer data consists of uncut chip thickness and rake angle, while the output layer data represents tangential forces. The four middle hidden layers utilize the ReLU activation function to pass the weighted sum of the input nodes to the output nodes. This study evaluates the trained DNN model's output response using RMSE and the R-squared coefficient of determination. The current dataset consists of 106 simulation data points, with 70% used for training and 30% for testing. Parameters are set using Adam's gradient descent method, with 30 training samples each time and an initial learning rate of 0.002 for 2500 training sessions.

Fig. 3.10 compares the test and training set prediction results with RMSE values of 11.497 N and 13.1489 N, respectively. The corresponding R-squared value approximates regression results between 0 and 1, where a higher value indicates a better prediction model. The model yields an R-squared value of at least 0.90 for the training and test sets. The model's predictive ability will significantly improve with more simulation data.

Despite its favorable predictive capabilities, the DNN model has limitations. Due to its fully connected structure, the links introduce many weight parameters, which can easily lead to overfitting and getting stuck in a local minimum.

Comparing the Kriging and DNN methods, the inability to fit the distance and semi-covariance matrix of the two points into a Gaussian function prevents the determination of corresponding weight values using the kriging interpolation method. The reason is due to an insufficient sample size or inadequate correlation between the attribute values of the sampled points. However, using the same dataset, the DNN model achieves accurate prediction results. Consequently, the performance of the E2NN model and other models will be compared to the DNN prediction results to evaluate their effectiveness.

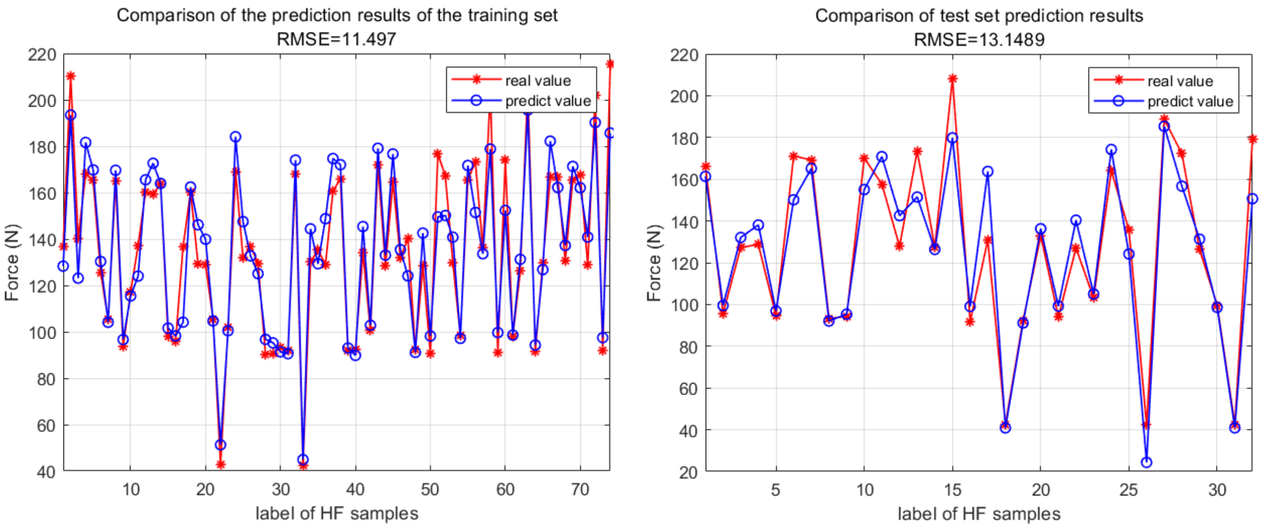


Fig. 3.10 *Predictions of training and test data set*

3.2.4 Multi-fidelity model

Previous sections detailed the generation of high-fidelity and low-fidelity datasets and the development of a high-accuracy DNN model using a large high-fidelity dataset. This section describes the structure of both the standard NN and multi-fidelity E2NN models. The standard NN comprises two hidden layers, each with ten neurons. The E2NN shares the same architecture as the standard NN but embeds three LF models as emulators within the hidden layer neurons. We will subsequently compare the prediction accuracy and efficiency of these models.

FEM computes various tangential forces for different uncut chip thicknesses and rake angle values involving a 0.1 mm edge tool radius. The experiment's two independent variables are uncut chip thickness (h) ∽ [0.05,0.25] (mm) and rake angle (α_r) ∽ [-10,10] (degree). The two design variables are parameterized with a normalized range of [0∽1], corresponding to the individually specified fields.

To observe the effect of the variation of the individual variables on the tangential force, as depicted in Fig. 3.11 and 3.12, we design the two low-fidelity models for the first hidden layer such that one of the variables varies within a specified range, while the other variable is fixed as its mean. Then, the two univariate tangential force responses are embedded in the first hidden layer as a low-fidelity model. The multivariate tangential force responses are embedded as another low-fidelity model in the second hidden layer. Fig. 3.13 illustrates the locations where the LF models are embedded into the E2NN structure.

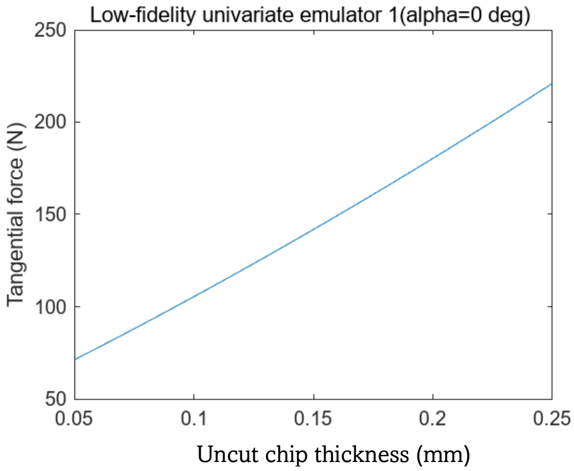


Fig. 3.11 *Low-fidelity univariate emulator 1(alpha=0 deg)*

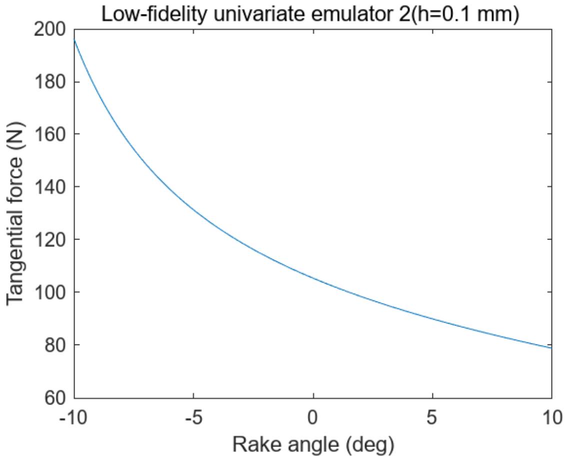


Fig. 3.12 *Low-fidelity univariate emulator 2(h=0.1 mm)*

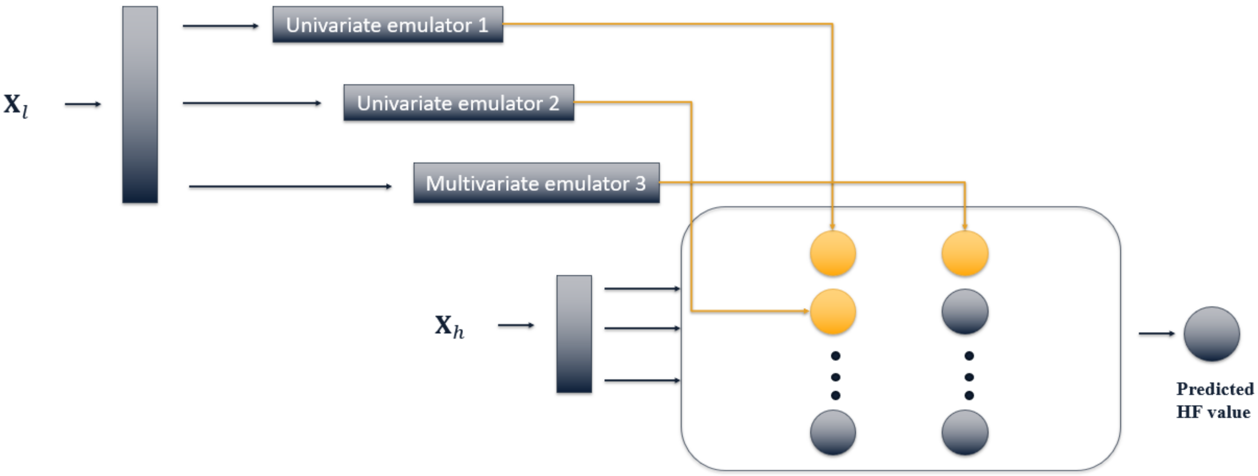


Fig. 3.13 *Low-fidelity model as an emulator embedded into E2NN structure*

We will place one set of tangential force values, with a data size of 40 HF samples, in the multi-fidelity model for training, as shown in Fig. 3.14. Then, we will randomly select 11 data points from another dataset to serve as the test set, the root mean square error (RMSE) values serve as the primary metrics for evaluating the prediction accuracy of each model, and results are shown in Table 3.5. The E2NN and DNN models demonstrate superior prediction accuracy, with RMSE values of 11.4505 N and 11.5769 N, respectively. In contrast, the standard NN prediction model's RMSE is twice as large, at 21.5919 N. The low-fidelity model, less reliable in predicting most data points, has an RMSE of 62.8649 N. The E2NN model achieves similar prediction accuracy to the DNN model while using only a third of the samples. Due to the high computational load in FEM, generating 40 samples saves 400 minutes compared to generating 120 samples while maintaining the same level of accuracy. Thus, when comparing the four different prediction models, the E2NN model provides high accuracy in a shorter time for complex engineering problems with insufficient data. In contrast, the standard NN model lacks the process of correcting the low-fidelity model with the high-fidelity model, potentially causing the predicted tangential force response to deviate from the actual value or even yield a lower forecast than the low-fidelity model.

Fig. 3.14 *Flowchart of E2NN built with 106 simulation data*

Table 3.4*: RMSE comparison among different models with 11 unique samples*

Force(N) No.	$F_{t\mathrm{E2NN}}$	$F_{t\mathrm{ABAQUS}}$	$F_{t\mathrm{Std.NN}}$	$F_{t\mathrm{LF}}$	$F_{t\mathrm{DNN}}$
1	132.6790	128.7402	138.8223	118.2349	138.0903
2	157.9290	134.1226	148.3975	186.2014	145.3823
3	96.7900	99.62385	83.6115	85.3967	98.6199
4	167.1128	166.5627	160.4577	242.5162	162.2219
5	130.5460	126.2840	129.8450	174.1635	131.2410
6	136.1028	136.7722	74.8556	69.9989	104.1584
7	98.3025	91.4975	106.5816	114.7546	94.2690
8	130.4680	136.7898	122.4104	100.6889	128.3154
9	173.7126	165.2792	173.8915	205.2594	169.8078
10	165.9143	188.724	179.9633	345.0731	185.3418
11	102.8292	90.36466	100.2461	96.6320	96.7320
RMSE	**11.4505**		**21.5919**	**62.8649**	**11.5769**

Furthermore, the model was validated using 24 and 40 feed-force HF values data sets. Subsequently, an independent data set of 12 points was employed as the test set. Tables 3.6

and 3.7 present the prediction errors for various multi-fidelity models. The E2NN model consistently exhibits lower prediction errors than the regular NN model, irrespective of the data set size (24 or 40). Moreover, the prediction errors of both multi-fidelity models are substantially smaller than those of the low-fidelity model. As the training data volume increases, the prediction errors of both multi-fidelity models decrease concurrently. With a data size of 40, the RMSE values for the E2NN and standard NN models were 12.3616 N and 19.4958 N, respectively, further illustrating the differences in their performance.

***Table 3.5**: RMSE comparison among different models with 12 unique samples*

Force(N) No.	$F_{f\,\text{E2NN}}$	$F_{f\,\text{ABAQUS}}$	$F_{f\,\text{Std.NN}}$	$F_{f\,\text{LF}}$
1	56.99206	47.82928	56.19761	71.84258
2	70.51380	60.60993	108.36067	132.48722
3	65.59275	70.36944	69.79977	69.95844
4	113.4817	73.79727	91.82555	174.92506
5	45.95913	60.17937	51.62610	130.28117
6	53.96278	73.11903	32.28336	56.44180
7	52.75784	40.29532	58.03757	87.24706
8	89.49916	80.56896	85.87521	61.50361
9	69.25818	62.67251	63.93514	137.43712
10	114.6665	109.5134	54.60754	251.04880
11	51.64779	37.19646	58.05653	71.30923
12	54.54158	51.28325	65.27245	102.27130
RMSE	**15.4740**		**26.4674**	**66.5306**

***Table 3.6**: RMSE comparison among different models with 12 unique samples*

Force(N) No.	$F_{f\,\text{E2NN}}$	$F_{f\,\text{ABAQUS}}$	$F_{f\,\text{Std.NN}}$	$F_{f\,\text{LF}}$
1	53.65168	47.82928	66.25063	71.84258
2	66.51380	60.60993	73.23846	132.48722
3	54.66232	70.36944	52.06062	69.95844
4	84.45536	73.79727	77.49385	174.92506
5	71.85457	60.17937	68.16398	130.28117
6	76.30908	73.11903	48.31218	56.44180
7	50.21936	40.29532	59.79117	87.24706
8	74.72558	80.56896	60.98423	61.50361
9	61.45031	62.67251	79.44327	137.43712
10	82.58030	109.5134	83.37696	251.04880
11	56.93550	37.19646	56.79805	71.30923
12	54.28325	51.28325	81.36864	102.27130
RMSE	**12.3616**		**19.4598**	**66.5306**

In conclusion, multi-fidelity models demonstrate enhanced predictive regression results compared to low-fidelity models. However, the performance of standard NN models is highly dependent on the availability of high-fidelity prediction samples. The standard NN may generate inaccurate predictions in particular local regions when the sample size is small. Embedding LF models into the E2NN model successfully integrates LF information, thereby improving overall prediction accuracy even with limited HF samples. The E2NN model mitigates the risk of overfitting by utilizing LF model emulators as a soft constraint.

In contrast, DNNs tend to be more prone to overfitting when a limited amount of high-fidelity data is available. Furthermore, the E2NN model can adapt to varying fidelity levels, allowing it to handle and deliver high performance in nonlinear problems effectively. Therefore, the E2NN model employed in this study, which incorporates multiple simulators in the hidden layer, demonstrates superior performance in global prediction for high-dimensional experimental designs.

Chapter 4. Conclusion

In this report, we discussed using finite element-based simulations to investigate the influence of machining parameters on manufacturing processes. We introduced a multi-fidelity model that combines a small amount of high-fidelity data with a large quantity of low-fidelity data to predict experimental results accurately. The multi-fidelity models employed in this study include co-Kriging, standard NN, and E2NN models. When applied to 1D data, both co-Kriging and E2NN models exhibit acceptable performance. However, co-Kriging's reliance on data following a Gaussian distribution may lead to inaccurate predictions when the data is unstable.

Additionally, co-Kriging requires numerous covariance matrix inversions, rendering it unsuitable for high-dimensional engineering problems. In contrast, the E2NN model performs excellent prediction when analyzing two-dimensional orthogonal cutting forces. The E2NN model allows the integration of any low-fidelity model with high-fidelity data as prior knowledge in the neural network, effectively mitigating the risks of overfitting and underfitting.

Standard NNs can yield accurate predictions with sufficient high-fidelity (HF) training samples and a strong relationship between input and output values. However, in the presence of limited HF training data, they may be prone to overfitting and need help capturing nonlinear relationships. In contrast, Deep Neural Networks (DNNs) can produce accurate predictions when trained on extensive and diverse HF samples, capturing intricate nonlinear relationships. Nevertheless, generating high-fidelity datasets is both expensive and inconvenient, and unlike E2NNs, DNNs cannot take advantage of various fidelity models. Moreover, E2NNs can achieve prediction accuracy similar to DNNs, using only one-third of the HF training samples required by DNNs.

During training, the E2NN model optimizes all models by incorporating a low-fidelity simulator, which biases the low-fidelity model towards the high-fidelity one. The low-fidelity model simulates magnifying the primary information from chaotic data and adapting it to the established path. In this research, Merchant's model serves as the source of the low-fidelity dataset. Three distinct low-fidelity models are generated by adjusting two independent variables (rake angle and uncut chip thickness) within a defined range. Using E2NN and a few high-fidelity datasets, it is possible to predict feed and tangential forces within the range of supplied independent variables.

This approach saves time and resources while enhancing the dimensionality of independent variables by incorporating multi-fidelity tool edge radius datasets into the neural network. As a result, the multi-fidelity model can offer comparably precise results in less time when analyzing the impact of processing factors to improve production quality.

In future studies, laboratory test data could be employed as the high-fidelity model. A simulation model with enhanced accuracy could serve as the low-fidelity model integrated into the neural network. Furthermore, by introducing a new independent variable, the influence of rake angle, uncut chip thickness, and tool edge geometries on tool life and process reliability could be predicted.

BIBLIOGRAPHY

[1] Altintas Y. ,. Author and Ber AA ,. Reviewer, “Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design,” *Applied Mechanics Reviews*, vol. 54, no. 5, pp. B84–B84, Sep. 2001, doi: 10.1115/1.1399383.

[2] P. L. B. Oxley and M. C. Shaw, “Mechanics of Machining: An Analytical Approach to Assessing Machinability,” *Journal of Applied Mechanics*, vol. 57, no. 1, pp. 253–253, Mar. 1990, doi: 10.1115/1.2888318.

[3] D. L. Logan, *First Course in the Finite Element Method, Enhanced Edition, SI Version*. Cengage Learning, 2022.

[4] M. Tyan, N. V. Nguyen, and J.-W. Lee, “Improving variable-fidelity modelling by exploring global design space and radial basis function networks for aerofoil design,” *Engineering Optimization*, vol. 47, no. 7, pp. 885–908, 2015.

[5] A. Zaytsev and E. Burnaev, “Large scale variable fidelity surrogate modeling,” *Annals of Mathematics and Artificial Intelligence*, vol. 81, pp. 167–186, 2017.

[6] M. Shi, L. Lv, W. Sun, and X. Song, “A multi-fidelity surrogate model based on support vector regression,” *Structural and Multidisciplinary Optimization*, vol. 61, pp. 2363–2375, 2020.

[7] L. Le Gratiet and J. Garnier, “Recursive co-kriging model for design of computer experiments with multiple levels of fidelity,” *International Journal for Uncertainty Quantification*, vol. 4, no. 5, 2014.

[8] M. C. Kennedy and A. O’Hagan, “Predicting the output from a complex computer code when fast approximations are available,” *Biometrika*, vol. 87, no. 1, pp. 1–13, 2000.

[9] Y. Zhang, N. H. Kim, C. Park, and R. T. Haftka, “Multifidelity surrogate based on single linear regression,” *AIAA Journal*, vol. 56, no. 12, pp. 4944–4952, 2018.

[10] Q. Zhou, X. Shao, P. Jiang, H. Zhou, and L. Shu, “An adaptive global variable fidelity metamodeling strategy using a support vector regression based scaling function,” *Simulation Modelling Practice and Theory*, vol. 59, pp. 18–35, 2015.

[11] C. Durantin, J. Rouxel, J.-A. Désidéri, and A. Glière, “Multifidelity surrogate modeling based on radial basis functions,” *Structural and Multidisciplinary Optimization*, vol. 56, pp. 1061–1075, 2017.

[12] A. Sobester, A. Forrester, and A. Keane, *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[13] H. S. Kim, M. Koc, and J. Ni, “A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network,” *International Journal of Machine Tools and Manufacture*, vol. 47, no. 2, pp. 211–222, 2007.

[14] J. P. Jacobs, S. Koziel, and S. Ogurtsov, “Computationally efficient multi-fidelity Bayesian support vector regression modeling of planar antenna input characteristics,” *IEEE transactions on antennas and propagation*, vol. 61, no. 2, pp. 980–984, 2012.

[15] J.-P. Chiles and P. Delfiner, *Geostatistics: modeling spatial uncertainty*, vol. 497. John Wiley & Sons, 2009.

[16] C. A. Charu, *Neural networks and deep learning: a textbook*. Spinger, 2018.

[17] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[18] A. Beachy, H. Bae, I. Boyd, and R. Grandhi, “Emulator embedded neural networks for multi-fidelity conceptual design exploration of hypersonic vehicles,” *Structural and*

Multidisciplinary Optimization, vol. 64, pp. 2999–3016, 2021.
[19] D. Liu and Y. Wang, “Multi-fidelity physics-constrained neural network and its application in materials modeling,” *Journal of Mechanical Design*, vol. 141, no. 12, 2019.
[20] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, *DACE: a Matlab kriging toolbox*, vol. 2. Citeseer, 2002.
[21] A. B. Nia *et al.*, “Dynamic response of aluminium sheet 2024-T3 subjected to close-range shock wave: Experimental and numerical studies,” *Journal of Materials Research and Technology*, vol. 10, pp. 349–362, 2021.
[22] M. D. Shields and J. Zhang, “The generalization of Latin hypercube sampling,” *Reliability Engineering & System Safety*, vol. 148, pp. 96–108, 2016.