ADVANCED NUMERICAL MODELING IN MANUFACTURING PROCESSES

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Arup Dey

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Industrial and Manufacturing Engineering

April 2022

Fargo, North Dakota

North Dakota State University
Graduate School

Title

ADVANCED NUMERICAL MODELING IN MANUFACTURING PROCESSES

By

Arup Dey

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Nita Yodo
Chair

Dr. David Grewell

Dr. Lokesh Narayanan

Dr. Zhibin Lin

Approved:

4 July 2022
Date

Dr. David Grewell
Department Chair

ABSTRACT

In manufacturing applications, a large number of data can be collected by experimental studies and/or sensors. This collected data is vital to improving process efficiency, scheduling maintenance activities, and predicting target variables. This dissertation explores a wide range of numerical modeling techniques that use data for manufacturing applications. Ignorance of uncertainty and the physical principle of a system are shortcomings of the existing methods. Besides, different methods are proposed to overcome the shortcomings by incorporating uncertainty and physics-based knowledge.

In the first part of this dissertation, artificial neural networks (ANNs) are applied to develop a functional relationship between input and target variables and process parameter optimization. The second part evaluates the robust response surface optimization (RRSO) to quantify different sources of uncertainty in numerical analysis. Additionally, a framework based on the Bayesian network (BN) approach is proposed to support decision-making. Due to various uncertainties, estimating interval and probability distribution are often more helpful than deterministic point value estimation. Thus, the Monte Carlo (MC) dropout-based interval prediction technique is explored in the third part of this dissertation. A conservative interval prediction technique for the linear and polynomial regression model is also developed using linear optimization.

Applications of different data-driven methods in manufacturing are useful to analyze situations, gain insights, and make essential decisions. But, the prediction by data-driven methods may be physically inconsistent. Thus, in the fourth part of this dissertation, a physics-informed machine learning (PIML) technique is proposed to incorporate physics-based knowledge with collected data for improving prediction accuracy and generating physically consistent outcomes. Each numerical analysis section is presented with case studies that involve conventional or additive manufacturing applications.

Based on various case studies carried out, it can be concluded that advanced numerical modeling methods are essential to be incorporated in manufacturing applications to gain advantages in the era of Industry 4.0 and Industry 5.0. Although the case study for the advanced numerical modeling proposed in this dissertation is only presented in manufacturing-related applications, the methods presented in this dissertation is not exhaustive to manufacturing application and can also be expanded to other data-driven engineering and system applications.

ACKNOWLEDGMENTS

My deep gratitude goes first to my adviser Dr. Nita Yodo, who expertly guided me through my graduate education and shared the excitement of four years of discovery. Her unwavering enthusiasm kept me constantly engaged with my research, and her generosity helped make my time at North Dakota State University (NDSU) enjoyable.

Besides my advisor, I would like to thank the rest of my committee members. Dr. David Grewell's mentoring and encouragement have been especially valuable, and his early insights launched the vital part of this dissertation.

Moreover, I am grateful to Dr. Lokesh Narayanan for his worthy compliments and suggestions. I enjoyed discussing many aspects of my research and moving forward to future discussions.

Thanks also go to Dr. Zhibin Lin, who dedicated his precious time to supporting my graduate study and helping with my professional development. This research and doctorate dissertation could not have been completed without the committee members' continued motivation.

Words are inadequate in offering thanks to my parents and family members, who are the ultimate source of inspiration for me. They have eagerly awaited long for the end of my doctoral studies. I would also like to thank all relatives, friends, and others who shared their support through this journey. This dream would not have been possible without their care and encouragement.

Above all, the praise goes to the Almighty, the author of knowledge and wisdom, for his countless love.

DEDICATION

This dissertation is dedicated

to my parents, Alomoynno Dey and Sanjo Rani Dey

and

my beloved wife, Anamika Mitra.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABS	Acrylonitrile butadiene styrene
AE	Acoustic emission
AI	Artificial intelligence
AM	Additive manufacturing
ANN	Artificial neural network
BFO	Bacterial foraging optimization
BN	Bayesian network
BNN	Bayesian neural network
BT	Build time
CAD	Computer-aided design
CCD	Central composite design
CFD	Computational fluid dynamics
CNN	Convolutional neural network
CS	Compressive strength
DAG	Directed acyclic graph
DoE	Design of experiment
ELU	Exponential linear unit
EMD	Empirical mode decomposition
FCCCD	Faced centered central composite design
FFF	Fused filament fabrication
GA	Genetic algorithm
ICA	Imperialist competitive algorithm
IMF	Intrinsic mode function
IoT	Internet of things

k-NN	k-nearest neighbor
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MC	Monte Carlo
MLE	Maximum likelihood estimation
MOO	Multiobjective optimization
MPa	Megapascal
MSE	Mean squared error
NN	Neural network
NSGA-II	Non-dominated sorting genetic algorithm II
OLS	Ordinary least squares
PCA	Principal component analysis
PHM	Prognostic and Health Management
PI	Prediction interval
PIML	Physics informed machine learning
PLA	Polylactic acid
PSO	Particle swarm optimization
RBF	Radial basis function
RBFNN	Radial basis function neural network
ReLU	Rectified linear units
RRSO	Robust response surface optimization
RSM	Response surface model
SVM	Support vector machine
WLS	Weighted least squares

CHAPTER 1. INTRODUCTION

1.1. Background

In today's technological era, the current trend of manufacturing industries is moving toward the fourth industrial revolution or Industry 4.0. In recent years, data-driven methods have significantly progressed due to cutting-edge technologies and high computational power. Industry 4.0 explores the internet of things (IoT), big data and data analytics, augmented reality, cybersecurity, collaborative robots, cloud computing, additive manufacturing (AM), artificial intelligence (AI), and finally, 5G networks in manufacturing for improving quality, increasing flexibility, reducing product launching time, and developing automated controlled systems [1].

Data is the primary driving fuel for Industry 4.0. The manufacturing data is collected, stored, processed, and analyzed using advanced technologies and data-driven methods [2]. Generally, data-driven numerical modeling techniques (e.g., machine learning algorithms) are applied to analyze the situations, gain insights, and make essential decisions from data. Data-driven methodologies can be used for diverse manufacturing purposes, such as online condition monitoring, remaining useful life predictions, remote device control, process parameters estimation, product quality improvement, and reduced downtime [3, 4].

Nowadays, manufacturing industries are slowly implementing data analysis algorithms to cope with Industry 4.0 and gain competitive advantages. The objectives of applying data-driven methods in manufacturing industries are machine anomalies detection, predictive maintenance, product inspection, cyber threat detection, etc. The experimental investigation, expert opinions, and sensors can be used to collect the data relevant to a target of interest. The collected data is further utilized for training (or estimating model parameters in) a numerical model.

The applications of data-driven methods are still limited in manufacturing due to the lack of data collection facilities on manufacturing floors, shortage of artificial intelligence (AI) and data science knowledge, and the complexity and diversity of machinery types. Most of the data-driven methods are not developed for specific manufacturing applications. It is also necessary to combine a data-driven approach with other algorithms. For example, data preprocessing and feature extraction algorithms can be applied to prepare a dataset. Besides, uncertainty quantification techniques can be applied to improve prediction accuracy and make a robust prediction. It is inevitable to use data-driven methods in the future as big data and analytics are integral parts of Industry 4.0 [3, 4].

The overarching goals of this dissertation aim to: (1) explore different data-driven numerical methods for various manufacturing processes, (2) modify data-driven methods to make them applicable for both conventional and additive manufacturing processes, (3) combine different data-driven methods that can be implemented in various manufacturing processes, and (4) develop holistic advanced data-driven algorithms that involve process parameter optimization as well as condition monitoring.

1.2. Current Limitations

Manufacturing systems are becoming complex and dynamic day by day. Adopting cutting-edge manufacturing technologies, utilizing AI systems, sustainability of manufacturing processes, and high demand for customized products are critical challenges for manufacturing industries [5-7]. Advanced numerical modeling is a viable option to overcome these challenges [8]. Currently, there are many data-driven methods available, including support vector machine (SVM), decision tree, Naive Bayes, *k*-nearest neighbor (*k*-NN), linear regression, artificial neural network (ANN), convolutional neural network (CNN), Bayesian network (BN), radial basis function neural network (RBFNN), and long short-term memory (LSTM). These data-driven methods can be applied to aid in decision-making in manufacturing applications for several purposes, such as predictive maintenance, fault diagnosis, and scheduling optimization.

Applying data-driven methods is also essential to improve efficiency and effectiveness. The accuracy of predictions is measured by comparing the predicted values with measured values of leveled data [9]. But there are some limitations to applying data-driven methods.

1. Due to different data sources, a plethora of data is available. High *computational power* is necessary to handle a plethora of data, and it is also essential to select the correct data information for a specific purpose.
2. *Data quality* is another vital factor for reliable and accurate predictions. The prediction from poor data quality may lead to a less reliable decision and eventually leads to loss of resources, competitiveness, and time. In many cases, decision-makers blindly rely on the prediction by data-driven methods. Another vital factor that impacts the performance of the manufacturing process and the data-driven methods is uncertainty.
3. *Uncertainty* is everywhere, and it is not possible to avoid uncertainty in real-life practices [10, 11]. Environmental noise, human error, measurement error, and natural randomness are potential sources of uncertainty in manufacturing data collection. Instead, uncertainty can be reduced by identifying the sources of uncertainty and taking appropriate corrective action. Additionally, uncertainty quantification can be incorporated into a decision-making process, and the impacts of uncertainty should be analyzed before making a decision.

4. Besides, data-driven models are developed under certain *assumptions*. Those model assumptions can affect the accuracy of predictions [11-15]. Data normality assumptions, linearity relation assumptions, convergence assumptions, small data size, and idealizing representations of actual performances are sources of model uncertainty.
5. Generally, data-driven algorithms rely on collected data for decision-making, not on the physics of a system from which the data is collected. The *ignorance of physics-based knowledge* may result in infeasible estimation by data-driven numerical models. Therefore, incorporating the physical principles of a system in data-driven methods is necessary to achieve better and more consistent predictions.

Data-driven methods involve some parameters, and the other parameters are often assumed to be constant and insignificant for the target variable. This limitation can cause the prediction results to be misleading when analyzing parameters that do not show the complete story of an analyzed target variable. Although there are several limitations, data-driven methods have many prospects, especially in manufacturing applications.

1.3. Dissertation Objective and Scope

This dissertation makes several contributions to the overall manufacturing applications objectives by proposing various advanced numerical methodologies to solve particular problem statements. The primary objectives of this dissertation are to: 1) develop and apply data-driven numerical modeling in various manufacturing processes for decision-making by analyzing the experimental and sensor data, 2) incorporate uncertainty in data-driven methods to make more robust decisions, and 3) propose a PIML for additive manufacturing applications.

This dissertation makes several contributions in the manufacturing sectors toward fulfilling the objectives. The scope of this dissertation includes the proposed methodology for the following problem statements (PSs):

- **PS 01:** RSMs are widely applied to develop a functional relationship between inputs and a target variable. Generally, RSMs are used to develop first-order, second-order, and third-order polynomial functional relationships that assumptions often result in poor performance and infeasible solutions. ANN is applied instead of RSM to overcome the shortcomings as ANN is flexible and capable of capturing complex functional relationships. In addition, multiobjective optimization formulations are developed for comparison purposes with ANN and RSM models used as the objective functions. The proposed methods for this problem statement fulfill this dissertation's first objective to explore the advanced numerical analysis models in manufacturing. The detail of the proposed methods will be detailed in Chapter 2.
- **PS 02:** The manufacturing environment has different sources of uncertainty. For instance, it is implausible that two products produced with identical parameters with the same machine and from the same raw materials have identical characteristics (e.g., mechanical properties and dimensions) due to different sources of uncertainty. RSM and ANN explored in the first problem statement do not incorporate uncertainty for prediction. The predicted values of a target variable by RSM and ANN are deterministic point values, which do not always represent the actual manufacturing systems due to several sources of uncertainty. Besides, the data-driven methods are developed under certain assumptions, such as the sigmoid activation function used in the ANN model to capture non-linearity.

To minimize the impact of uncertainty in predictions, the second scope of this dissertation seeks to apply various uncertainty quantification techniques such as Monte Carlo (MC) dropout and Bayesian network (BN) for condition-based data-driven maintenance and process parameter analysis. In Chapter 3, a robust response surface optimization (RRSO) technique, which includes model parameter uncertainty, is proposed for the distribution estimation of a target variable for a set of input variables. Moreover, the uncertainty integration for robust decision analysis for both numerical and categorical variables with BN will be covered in Chapter 3.

Interval prediction is a statistical estimate of an interval that incorporates uncertainty. This prediction estimates an interval or range instead of a point value in which a future observation will fall with a certain probability or confidence interval level. To incorporate model uncertainty in decision-making, Chapter 4 elaborates on the proposed data-driven interval prediction based on MC dropout for tool wear and the proposed conservative interval prediction technique for linear and polynomial regression.

- **PS 03:** A significant limitation of most traditional data-driven algorithms is that the physical phenomena of a system are not considered for model training and a target variable estimation [12, 13]. Thus, the physics-based knowledge can be incorporated during training data-driven models to make a physically consistent prediction by overcoming the limitation of traditional data-driven methods. This concept is known as physics-informed machine learning (PIML). PIML is a relatively new concept compared to traditional machine learning and is rarely explored for AM processes.

In this dissertation, a PIML is developed to predict the surface roughness of a fused filament fabrication (FFF) build part. This corresponds to the third objective of the overall research. In the proposed method, the physics-based empirical model prediction of the surface roughness has been used as input along with other inputs. The goal of the proposed concept is to make physically consistent predictions and improve prediction efficiency and accuracy. In addition, a customized loss function has been introduced based on physics-based constraints.

The proposed PIML model is a hybrid model that combines data-driven methods and physics-based models. A radial basis function neural network (RBFNN) model is trained by considering physics-based knowledge about the target variable as an input along with other input variables by minimizing the proposed customized loss function. In addition, four performance evaluation metrics are used for performance analysis. The proposed PIML model that incorporates physics-based knowledge is demonstrated in Chapter 5.

In short, this dissertation explores and improves the different advanced numerical models for manufacturing applications. For instance, ANN, MC dropout, and RSM models are employed to support the decision-making process in the manufacturing environment. As the shortcoming of the numerical models is identified, different methods such as RRSO, conservative interval estimation techniques, and the PIML approach are proposed to overcome the current applications’ drawbacks.

1.4. Dissertation Organization

The organization of the dissertation and the major contributions to fulfill the four objectives are summarized in Figure 1.1.

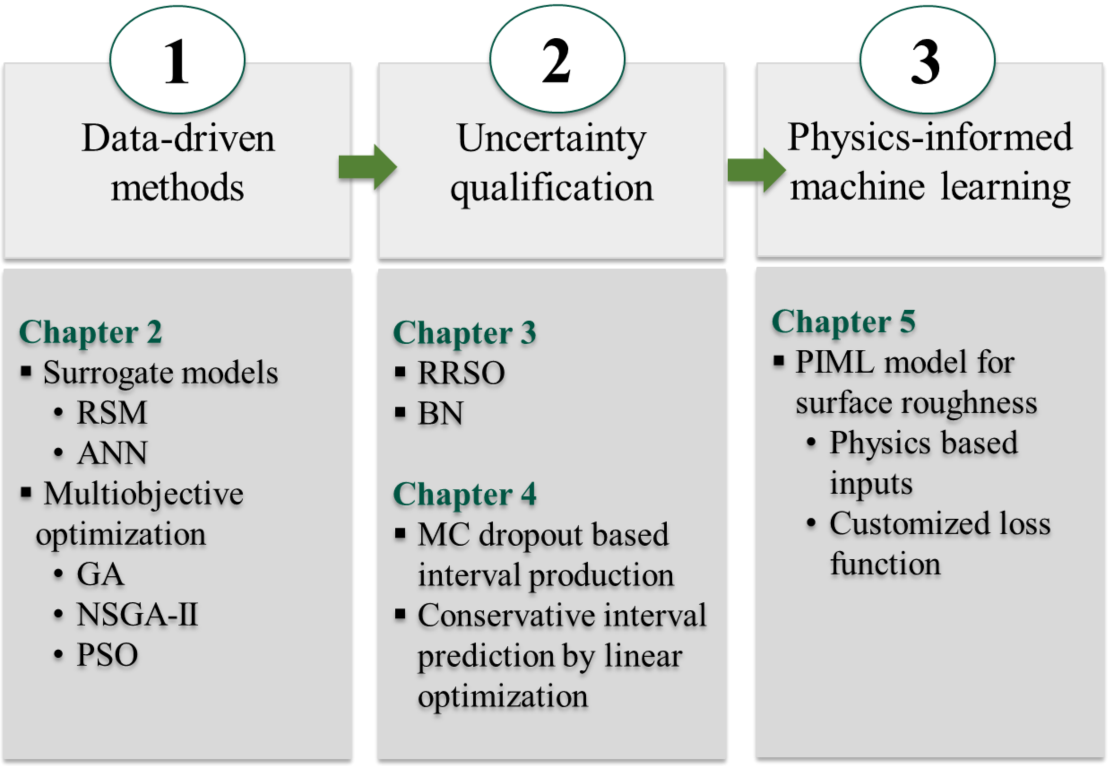

1
Data-driven methods
2
Uncertainty qualification
3
Physics-informed machine learning
Chapter 2
▪ Surrogate models
• RSM
• ANN
▪ Multiobjective optimization
• GA
• NSGA-II
• PSO
Chapter 3
▪ RRSO
▪ BN
Chapter 4
▪ MC dropout based interval production
▪ Conservative interval prediction by linear optimization
Chapter 5
▪ PIML model for surface roughness
• Physics based inputs
• Customized loss function

Figure 1.1. Dissertation organization

In Chapter 2, the details of different data-driven methods and optimization techniques applied in an FFF dataset have been discussed. The data collection procedure is also introduced in this chapter. The advantages of using ANN over RSM are demonstrated by applying them to the collected dataset. Chapter 3 provides the proposed RRSO technique, which can estimate the distribution of a target variable for a combination of input variables by multiobjective optimization. Besides, a Bayesian network-based approach to analyzing both numerical and categorical variables has also been introduced in Chapter 3.

Two interval prediction techniques demonstrate in Chapter 4. One interval prediction technique is proposed to predict conservative intervals and applied to an FFF dataset. The second proposed interval prediction technique is developed based on MC dropout and demonstrated by applying the proposed framework to a real-world computer numerical control (CNC) milling machine dataset to predict tool wear as intervals.

A PIML approach is proposed in Chapter 5 to predict surface roughness to improve prediction accuracy by incorporating physics-based knowledge. An RBFNN is trained using physics-based inputs and other input variables to estimate surface roughness. The performance of the proposed method is compared with other existing applied physics-based and data-driven approaches. Finally, Chapter 6 summarizes the work and describes the future research directions.

CHAPTER 2. NUMERICAL MODELING PROCESS FOR MULTIOBJECTIVE DECISION-MAKING

2.1. Overview

The response surface model (RSM) is a widely applied statistical technique that represents the relationship between some input variables (design parameters) and a target variable (response). RSM aims to analyze the impacts of input variables on the response and optimize responses to select an optimum combination of input variables. A combination of design parameters is estimated for an optimum response value by optimizing the developed response surface model under certain constraints. In developing a functional relationship, it is assumed a linear, quadratic, or third-order relationship exists between input variables and response. This assumption is not always valid, as the actual functional relationship may be more complex than the one obtained with the RSM method. Thus, this assumption may result in poor performance and infeasible solutions.

To overcome this problem, artificial neural networks (ANNs) are proposed. ANN has a flexible architecture. Any level of complexity can be captured by changing the number of hidden layers and the number of neurons in each hidden layer. In this work, ANN is used as an alternative to RSM, and the performance of ANN is compared with RSM. Both ANN and RSM surrogate models are used to develop the functional relationship between input variables and a target variable. When two objective functions are optimized simultaneously, the optimization is known as multiobjective optimization. Generally, RSMs are used as objective functions to determine the values of input variables through optimization. The developed surrogate models with RSM and ANN are further used as objective functions in the multiobjective optimization (MOO) process. In multiobjective optimization, two responses are optimized simultaneously.

Numerous algorithms can be employed to solve MOO formulations. Here, genetic algorithm (GA), non-dominated sorting genetic algorithm (NSGA-II), and particle swarm optimization (PSO) are applied to solve the MOO problem. MOO generates a set of non-dominated solutions, and the set of solutions is called Pareto optimal solution set. A decision-maker then chooses a solution from all non-dominated solutions based on requirements. A fused filament fabrication (FFF) dataset is collected, and the algorithms mentioned above are applied to analyze and compare the performance of ANN and RSM.

2.2. Methods Introduction

In this section, all methods applied for optimization are discussed. ANN and RSM are used to develop a functional relationship between inputs and output, also called the surrogate models. The MOO approach is used to optimize the outputs. The explored MOO techniques are GA, NSGA-II, and PSO.

2.2.1. Response Surface Model

The response surface model (RSM), often interchangeably known as the response surface method, is a statistical approach that explores the relationships between one or more input variables and a response (or target variable) [14]. RSM has been widely used in the area of reliability analysis [14], quality improvements [15], structural analysis [16], and many more, including FFF process parameter optimization [17, 18]. The actual functional relationship between input variables and the response is generally unknown but can be approximated by lower-order polynomial functions. The first-order RSM, which is based on linear approximation, can be expressed as:

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \epsilon \tag{2.1}$$

And the quadratic RSM, which is based on the polynomial approximation, can be formulated as:

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j + \epsilon \tag{2.2}$$

In the above equations, $y\ is$ the response variable; x_i and x_j are the input variables, where $i < j$. β_0 is a regression coefficient for the intercept, whereas β_i, β_{ii} and β_{ij} are the coefficients of linear, quadratic, and other interaction terms, respectively. k is the number of input variables. ϵ is an error term. The second-order RSM is preferred to the first-order RSM when presenting real-life applications. Each term of a mathematical model may not be significant. Different techniques such as forward selection and backward elimination are applied in developing RSM to remove insignificant terms from RSM.

2.2.2. Artificial Networks

The artificial neural network (ANN), simply called a neural network, is a multi-layered network architecture vaguely modeled based on the functionality of biological neurons in the brain. ANN has become a popular data analysis approach for various purposes, for example, classification, clustering, prediction, and pattern recognition [19]. ANN is widely used as it is flexible and can represent complex relationships between variables [20]. A significant advantage of using ANN is that in-depth knowledge about the physics of a system is not entirely necessary. Typically, ANN consists of three layers: an input layer, hidden layers, and an output layer. A general ANN architecture is given in Figure 2.1.

The ANN architecture varies. It can take different shapes and structures based on the number of neurons in each layer. In ANN, neurons, layers, and activation functions represent a mathematical relationship between input and output variables (responses). Each unit of a layer is called a neuron. In an ANN, a neuron of a layer is fully connected with all neurons of prior and post layers, and a neuron of a layer is not associated with any other neurons of the same layer. The output of a neuron acts as an input to the next layer. Figure 2.1 NN has one input layer with four input neurons, one hidden layer with seven neurons, and one output layer with one output neuron. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

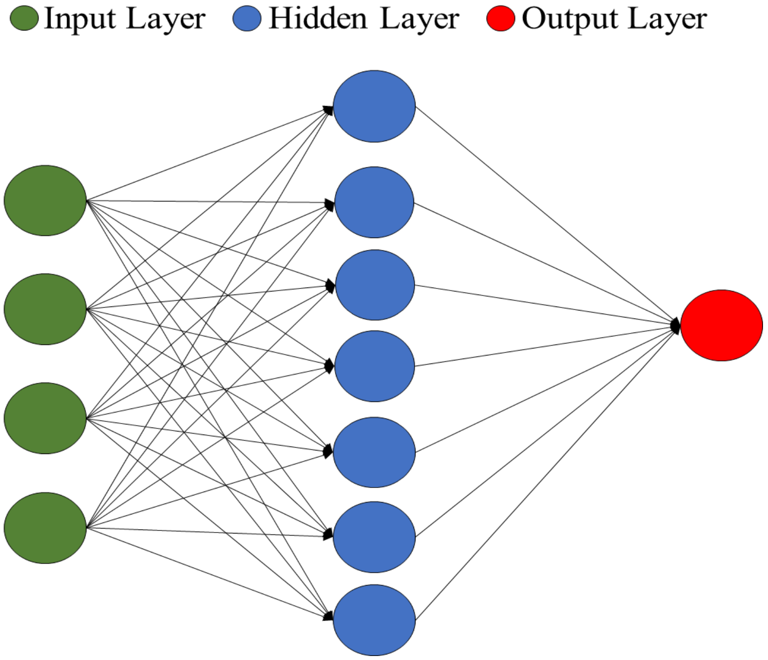


Figure 2.1. Artificial neural network

Figure 2.2 describes operations performed in a neuron. The neuron has three inputs $(x_1, x_2, and\ x_3)$. The weights of three inputs $x_1, x_2\ and\ x_3$ are $w_1, w_2\ and\ w_3$, respectively, and b is the bias term. $y = f(z)$ is the activation function that introduces non-linearity. Commonly used activation functions are sigmoid, tanh, relu, elu, and others [21]. y is a input for neurons of the next layer.

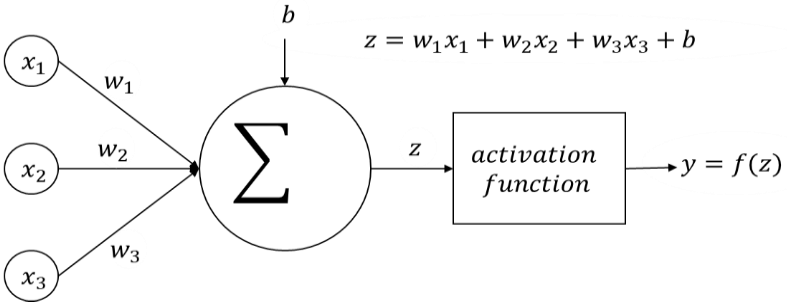


Figure 2.2. Operations performed in a neuron

To apply ANN, it is required to estimate weights and biases by minimizing loss function from data to predict the output for new observations. The least-square loss function is the most widely applied and can be defined as the summation of errors (losses) between the output of ANN and the given target value for all training data. The mathematical form of the least square loss function is given as

$$\mathcal{L}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.3}$$

where,$\{y_1, y_2, y_3, \dots \dots y_n\}$ are the output of n observations, and $\{\hat{y}_1, \hat{y}_2, \hat{y}_3, \dots \dots \hat{y}_n\}$ are predicted values by the ANN model corresponding to inputs $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \dots \boldsymbol{x}_n\}$. The above formula is applicable for any regression model, including RSM for estimating model parameters. Different algorithms such as gradient descent, stochastic gradient descent, and Levenberg-Marquardt algorithm are applied to minimize the loss function. The value of weights and biases are updated by different algorithms until a desired level of accuracy is achieved. In many cases, ANN is expected to perform well for training data, and prediction accuracy is low for new observations. This limitation is known as overfitting. This limitation can be overcome by applying different regularization techniques to avoid the risk of overfitting. Dropouts, L_1 regularizations, and L_2 regularizations are some examples of the popular regularization techniques [22].

2.2.3. Multiobjective Optimization

Multiobjective optimization involves two or more objective functions to be optimized simultaneously to make multi-criteria decisions. Unlike single-objective optimization, multiobjective optimization generates a set of non-dominated solutions, the set of solutions is called Pareto optimal solution set. A non-dominated solution is the improvement of one solution that results in at least one other worse solution [23]. In other words, the resulting solutions from multiobjective optimization are often called non-dominated solutions, meaning that there is no single solution that can simultaneously optimize all objectives [23, 24]. A decision-maker then chooses a solution from all non-dominated solutions based on requirements. The weighted sum approach, weighted metric method, Ɛ-constraint method, and goal programming are a few well-known techniques for solving multiobjective optimization problems [25]. Nowadays, different evolutionary algorithms (e.g., genetic algorithm (GA) and non-dominated sorting genetic algorithm (NSGA-II)) and biologically-inspired Optimization (e.g., particle swarm optimization (PSO) and bacterial foraging optimization (BFO)) are also used to get Pareto optimal solution set [26, 27].

2.3. Data Collection Details

An FFF dataset is collected for comparisons of ANN and RSM models. The impacts of four process parameters on compressive strength and build time are analyzed, and an experimental dataset is collected for this. The details of the data collection procedure are given in the following two subsections.

2.3.1. Data Overview

An FFF process dataset is collected to demonstrate the effectiveness of ANN and to compare ANN with RSM. The applications of FFF-produced parts are limited due to high surface roughness, low dimensional accuracy, and inconsistent mechanical properties [27]. The inconsistency results in repeatability are also a restriction for applications of FFF build parts. It is known that the FFF process parameters have significant impacts on part properties [28, 29]. In the dataset, compressive strength and build time are considered as responses. The effects of four FFF process parameters, namely layer thickness, build orientation, infill density, and extrusion temperature, on the two responses are analyzed. Acrylonitrile butadiene styrene (ABS) and polylactic acid (PLA) are the two most widely used materials for the FFF filament [30]. This research uses non-toxic and biodegradable bioplastic PLA as the filament material. PLA is also used for producing functional parts such as medical devices. The specimen is built according to ASTM D695, the international standard for compressive properties of thermoplastics [31]. The dimension of the rectangular-prism-shaped specimen is 12.7 mm × 12.7 mm × 25.4 mm is selected according to ASTM D695, and the specimen is shown in Figure 2.3.

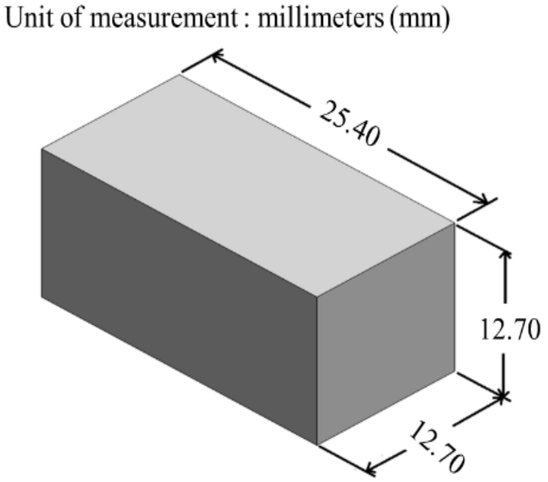


Figure 2.3. Specimen according to ASTM D695

To estimate the compressive strength and build time, the test specimen is initially designed in the Fusion 360 environment, a cloud-based computer-aided design (CAD) modeling software. A MakerBot Replicator Z18 is used to produce PLA samples from a CAD design.

MakerBot recommends using a smart extruder with a nozzle diameter of 0.4 mm for printing parts from PLA filament. The black color PLA filament with a diameter of 1.75 mm manufactured by 3D Solutech is used in the experiments. The compressive strength of the specimen is evaluated by an INSTRON compressive strength testing machine with a load of 30kN moves uniformly speed of 1.3mm/min. The chronological steps of data collection are represented as a flow diagram in Figure 2.4.

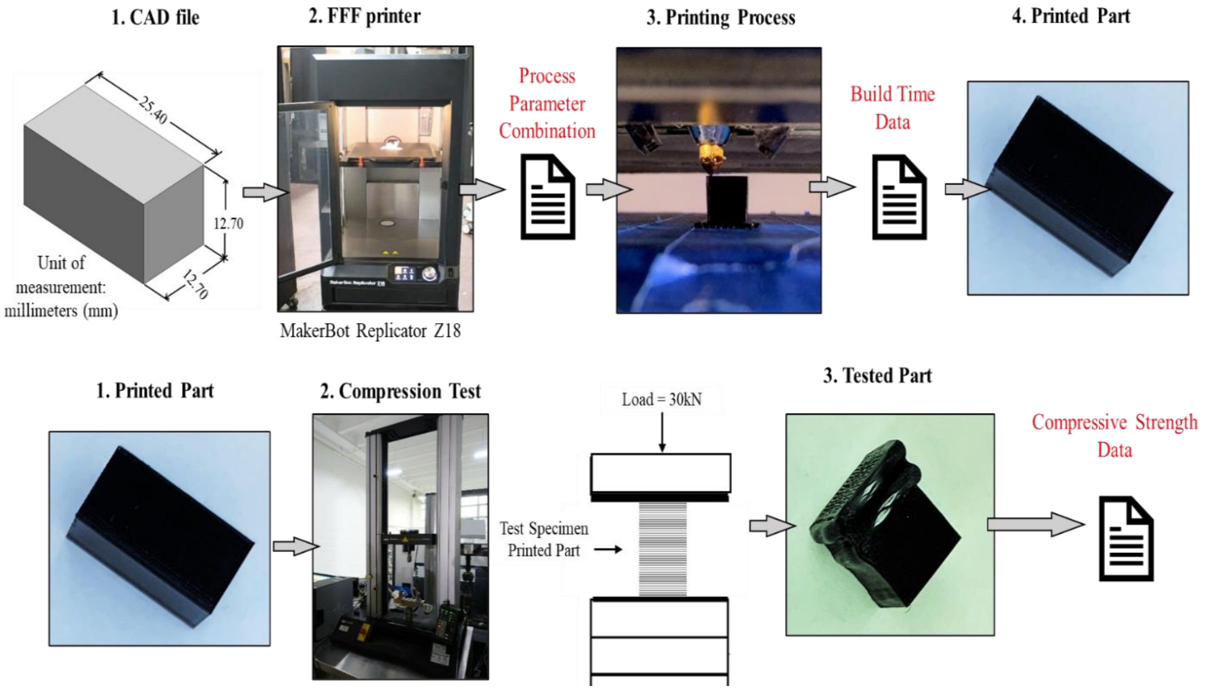


Figure 2.4. Data collection for process parameter combinations

Layer thickness and build orientation are the two most analyzed process parameters and significantly impact different part properties [27]. It is known from existing research on process parameter analysis that layer thickness and build orientation impact substantially on compressive strength and build time [32]. For this, layer thickness and build orientation are selected as process parameters to analyze the effect on compressive strength. The mechanical properties of a material and build time to print a layer depend on its density. The impact of infill density on FFF build parts' compressive properties is still unknown as infill density is one of the least analyzed process parameters. Like infill density, extrusion temperature is also the least analyzed process parameter, and the impact on compressive strength and build time is unknown [27]. The four different types of process parameters are selected in this research because the impact of those process parameters combination on compressive strength and build time is rarely analyzed. The levels of four (04) process parameters are given in the following table. For each process parameter, we are considering three levels (low, center, and high). The center level is the average of the high and low levels.

Table 2.1. Investigated parameters and their levels

Parameters	Units	Level		
		Low (-1)	Center (0)	High (+1)
1. Layer thickness, x_1	millimeters (mm)	0.1	0.22	0.34
2. Build Orientation, x_2	Degree	0	45	90
3. Infill Density, x_3	Percentage	20	50	80
4. Extrusion Temperature, x_4	°C	200	215	230

2.3.2. Experimental Design

The design of experiment (DoE) approach aims to obtain the possible maximum amount of information from a smaller number of experiments. As part of the DoE approach, the faced-centered central composite design (FCCCD) is used to reduce part production as the experimental run is costly and time-consuming. Central composite design (CCD) is a mathematical and statistical DoE tool used to develop a non-linear model from a reduced number of experiments. In CCD, the rotatability of design points is incorporated by a constant value, α [33]. When α=1, it is called FCCCD. FCCCD requires parameters with three levels: low (-1), center (0), high (+1). In this experimental design, -1, 0, and +1 are the coded levels of parameters. We considered four (04) parameters, and each has three (03) levels. Total design points consist of two-level full factorial design points (2^k), axial design points $(2k)$, and center design points (6 or more). Here, k=4 and is representing the number of factors (process parameters).

In this research, the influence of four (k=4) process parameters is investigated, and a total of 30 combinations of process parameters is generated according to two levels of full factorial design (16), axial (8), and center (6) design points. The total number of experiments with coded values of levels of process parameters using FCCCD is given in Table 2.2.

The actual levels of parameters can be converted into coded values using the following formula. This is called normalization, and the range of coded variables from -1 to +1 [34].

$$x_{ic} = \frac{2x_i - (x_{il} + x_{iu})}{x_{iu} - x_{il}}, i = 1,2, \dots k \tag{2.4}$$

In Equation (2.4), x_{ic} is the coded level of i^{th} parameter, x_{il} is the lower (uncoded) level of an i^{th} parameter, x_{iu} is the upper level of an ith parameter and x_i is the uncoded level of the i^{th} parameter that is intended to convert to the coded level.

According to experimental design, three (03) sets of parts, a total of 90 parts, are produced. The build time is recorded directly from the FFF machine dial box for each part. Then, the compressive strength of PLA parts is evaluated with an INSTRON compression testing machine. The average values of build time and compressive strength of the three specimen samples are used for further analysis. The measured values of compressive strength and build time for different combinations of process parameters are given in Table 2.2.

Table 2.2. Experimental data obtained by FCCCD

SL. No.	x_1	x_2	x_3	x_4	Compressive Strength (MPa)	Build time (min)
1	-1	-1	-1	-1	16.88	25.85
2	-1	-1	-1	1	17.64	25.85
3	-1	-1	1	-1	18.79	44.08
4	-1	-1	1	1	18.39	44.53
5	-1	1	-1	-1	28.52	35.13
6	-1	1	-1	1	29.00	35.12
7	-1	1	1	-1	28.01	50.28
8	-1	1	1	1	27.13	50.28
9	1	-1	-1	-1	20.22	9.74
10	1	-1	-1	1	21.20	9.62
11	1	-1	1	-1	32.98	14.70
12	1	-1	1	1	34.24	14.70
13	1	1	-1	-1	17.46	12.25
14	1	1	-1	1	19.48	12.22
15	1	1	1	-1	27.86	16.48
16	1	1	1	1	30.12	16.48
17	-1	0	0	0	15.98	49.78
18	1	0	0	0	18.68	15.41
19	0	-1	0	0	23.28	18.52
20	0	1	0	0	22.75	20.85
21	0	0	-1	0	11.32	15.70
22	0	0	1	0	22.10	23.83
23	0	0	0	-1	13.24	20.38
24	0	0	0	1	18.45	20.39
25	0	0	0	0	16.58	20.39
26	0	0	0	0	18.05	20.39
27	0	0	0	0	16.36	20.40
28	0	0	0	0	14.80	20.39
29	0	0	0	0	16.68	20.38
30	0	0	0	0	17.91	20.39

2.4. Results and Discussions

Based on the experimental data, quadratic response surface and ANN models for compressive strength and build time are developed. The coded levels of process parameters are used for surrogate models. For the rest of the chapter, the coded levels of variables is referred to instead.

2.4.1. Compressive Strength

Based on the experimental data, a quadric model for compressive strength is developed that represents the relationship of compressive strength with a layer thickness (x_1), build orientation (x_2), infill density (x_3), and extrusion temperature (x_4). For this, a response surface quadratic equation for compressive strength is generated using MINITAB 18 software. In a mathematical model, not all terms are always significant. In this case, all insignificant terms are eliminated by the backward elimination method with a 95% confidence level. The terms with p-values less than 0.05 are considered significant terms. Other than that, the t-value of all significant terms is determined at a 95% confidence level.

The coefficient, t-value, and p-value of all significant terms are given in the following Table 2.3. From Table 2.3, it is visible that extrusion temperature is not significant for compressive strength. Therefore, we do not need to consider extrusion temperature as a variable for further analysis of compressive strength. There are some square terms (x_1^2 and x_2^2) and interaction terms (x_2x_3) also insignificant. All insignificant terms are eliminated by the backward elimination method. Using a low extrusion temperature (in the range of 190^oC), is recommended as it will reduce extruder preheating time. A low extrusion temperature also increases dimensional accuracy and surface finishing, but the bond between layers becomes firmer at a high temperature.

The quadratic model of compressive strength (CS) is given in Equation (2.5).

$$\mathrm{CS} = 16.679 + 1.217\mathrm{x}_1 + 1.483\mathrm{x}_2 + 3.216\mathrm{x}_3 + 7.430\mathrm{x}_2^2 - 3.418\mathrm{x}_1\mathrm{x}_2 + 2.910\mathrm{x}_1\mathrm{x}_3 \qquad (2.5)$$

Table 2.3. Compressive strength analysis

Term	Coefficient	t-value	p-value
Constant	16.679	37.09	0.000
x_1	1.217	3.31	0.003
x_2	1.483	4.04	0.001
x_3	3.216	8.76	0.000
x_2^2	7.430	12.80	0.000
x_1x_2	-3.418	-8.78	0.000
x_1x_3	2.910	7.47	0.000
R^2		0.9457	
Adjusted R^2		0.9315	
Predicted R^2		0.9163	

In Table 2.3, R^2, adjusted $\mathrm{R}^{2,}$ and predicted R^2 values are given as 0.9456, 0.9315, and 0.9163, respectively. The high values represent that the mathematical model has good agreement with experimental data.

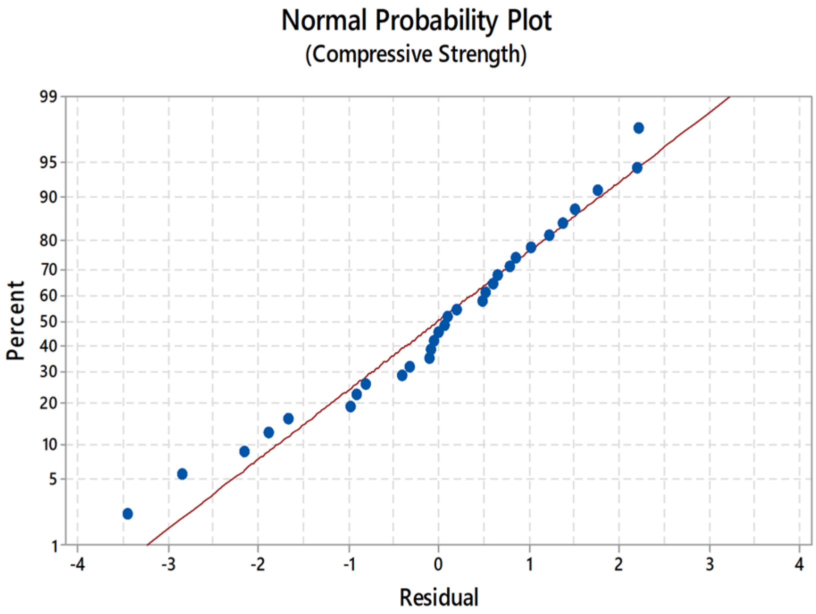


Figure 2.5. Normal probability plot of residuals for compressive strength

Another verification method is the normal probability plot used to determine the quadratic model's statistical significance. The fitted linear line in the normal probability plot shown in Figure 2.5 indicates that residuals are approximately normally distributed, and the predictions by this model have good agreement with experimental results.

Similarly, the ANN model is used to develop mathematical models representing the relationships of compressive strength and four process parameters. For compressive strength, the simplest ANN structure with only one hidden layer to capture non-linearity is considered. The rest of the ANN model parameters used to develop the surrogate models are as follows. The number of input variables (process parameters) is 4. The number of neurons in the hidden layer is 11. The activation function used in the hidden layer is the sigmoid function for compressive strength. Further, the Adam optimizer is used to train the network with a learning rate of 0.001, exponential decay control parameters of 0.99 and 0.999, and a smoothing term of 10-8. The learning rate for ANN models is set as 0.003. The details of the Adam optimizer can be found in Ref. [35]. The RSM and ANN models for compressive strength are used as objective functions in multiobjective optimization.

2.4.2. Build Time

A response surface quadratic model for build time is also developed using experimental results from Table 2.2. The model became statistically insignificant as the normal probability plot is not linear, and errors are not random. For this reason, we collected more data for build time, the levels of process parameters for additional experimental runs, and the build time represented in Table 2.4.

Table 2.4. Experimental data for build time

SL. No.	x_1	x_2	x_3	x_4	Build time (min)
1	1	-1	0	-1	12.46
2	0	-1	0	1	18.31
3	-1	0	0	1	46.98
4	1	0	1	1	17.35
5	1	1	0	1	14.60
6	-1	0	1	0	54.52
7	0	-1	-1	0	13.30
8	0	1	-1	-1	17.07
9	0	-1	1	-1	21.60
10	-1	0	-1	1	38.35
11	-1	1	0	0	43.72
12	0	1	1	0	23.98
13	1	0	-1	1	12.93
14	-1	-1	0	-1	37.32
15	-0.50	-0.33	-0.67	-0.33	25.88
16	-0.50	-0.33	0.67	-0.33	32.63
17	-0.50	0.33	-0.67	-0.33	26.43
18	-0.50	0.33	0.67	-0.33	32.43
19	0.50	-0.33	-0.67	0.33	15.13
20	0.50	-0.33	0.67	0.33	19.08
21	0.50	0.33	-0.67	0.33	15.30
22	0.50	0.33	0.67	0.33	19.67

Using all experimental data for build time from Table 2.2 and Table 2.4, a quadratic model generated consists of all significant terms at 95% confidence levels. The extrusion temperature is insignificant for build time, like compressive strength. Some square terms (x_3^2) and interaction terms (x_2x_3) are also insignificant. The backward elimination method eliminates all insignificant terms from the mathematical model. The results of the t-test and p-values with coefficients of all significant terms are given in Table 2.5.

Table 2.5. Build time analysis

Term	Coefficient	T-Value	P-Value
Constant	21.362	49.47	0.000
x_1	-13.680	-40.28	0.000
x_2	2.014	5.81	0.000
x_3	4.863	14.48	0.000
x_1^2	8.865	14.96	0.000
x_2^2	-3.441	-6.01	0.000
x_1x_2	-1.286	-3.17	0.003
x_1x_3	-2.965	-7.42	0.000
R^2		0.981	
Adjusted R^2		0.977	
Predicted R^2		0.973	

The quadratic mathematical model for build time (BT) is given in Equation (2.6).

$$\begin{gathered}\mathrm{BT} = 21.362 - 13.680\mathrm{x}_1 + 2.014\mathrm{x}_2 + 4.863\mathrm{x}_3 + 8.865\mathrm{x}_1^2 - 3.441\mathrm{x}_2^2 \ldots \\ \ldots - 1.286\mathrm{x}_1\mathrm{x}_2 - 2.965\mathrm{x}_1\mathrm{x}_3 \end{gathered} \tag{2.6}$$

In Table 2.5, the high value of R^2, adjusted R^2, and predicted R^2 indicates that the experimental data are well-fitted with the quadratic model.

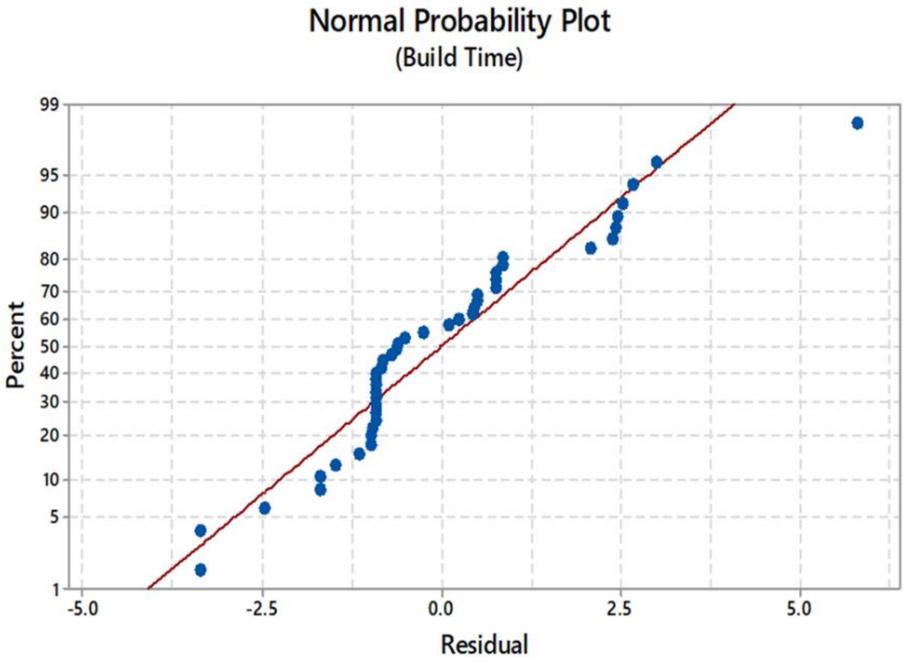


Figure 2.6. Normal probability plot of residuals for build time

In Figure 2.6, it is shown the normal probability plot of build time, and the linearity of the graph represents that the distribution of residuals is normal.

Similar to compressive strength, an ANN model is developed for build time. In the ANN model, the number of the hidden layer is 1, and the number of neurons in the hidden layer is 11. The activation function used in the hidden layer is the tanh activation function for build time. The Adam optimizer is applied to train the ANN model of build time, and the learning rate for ANN models is set as 0.003.

The surrogate models represent the relation of the analyzed process parameters with responses. The four developed mathematical models are employed as the objective functions in the multiobjective optimization problem that are solved by genetic algorithm (GA), non-dominated sorting genetic algorithm (NSGA-II), and particle swarm optimization (PSO). Before the surrogate models can be implemented as in the optimization formulations, the RSM models of compressive strength and build time are compared with the ANN models of compressive strength and build time.

2.4.3. Comparisons Results

Many metrics are available to evaluate the performance of surrogate models. In this research, the performance of RSM and ANN is compared using prediction accuracy. The mean squared error (MSE), the mean absolute error (MAE), and R^2 values are calculated for performance analysis. MSE measures the squared average error between actual data and the predicted data, whereas MAE measures the magnitude of average errors in a set of predictions by taking the average of the absolute difference between prediction and actual data. Besides, R^2-value is a measurement of the goodness of fit of a regression model. The formulations for both MSE, MAE, and R^2 values are given in the equations below.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.8}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}\left(y_i - \mu_{y_i}\right)^2} \tag{2.9}$$

where y_i is the measured value of the target variable at i^{th} observation, $\hat{y}_i$ is the predicted value at i^{th} observation, and μ_{y_i} is the mean of measured values. $i = 1,2,3, \dots n$. n is the total number of data points. MSE and MAE can range from 0 to ∞, with lower values preferred, representing a smaller margin of error. R^2-value lies between 0 and 1, and the higher value indicates a better model fit.

Table 2.6. MSE and MAE values

Properties of Printed Part (Target Variables)	MSE		MAE		R^2-value	
	RSM	ANN	RSM	ANN	RSM	ANN
Compressive Strength	1.861	0.582	1.040	0.509	0.946	0.983
Build Time	2.825	0.518	1.297	0.460	0.981	0.996

The results of MSE, MAE, and R^2 values obtained for RSM and ANN are shown in Table 2.6. It should be noted that the ANN shows a smaller degree of errors compared to RSM. It could also mean that the surrogate models obtained by ANN are more accurate than the RSM.

The scatter plots in Figures 2.7 (a) and 2.7 (b) present the predicted versus actual measured values for compressive strength for the RSM and ANN methods, respectively. For the scatter plot of the ANN model, coordinate points are close to the diagonal line compared to the scatter plot of the RSM model. It indicates that the ANN model is fitted well for compressive strength. Besides, the higher *R*-value for the ANN model demonstrates that the prediction performance overweighs the RSM model.

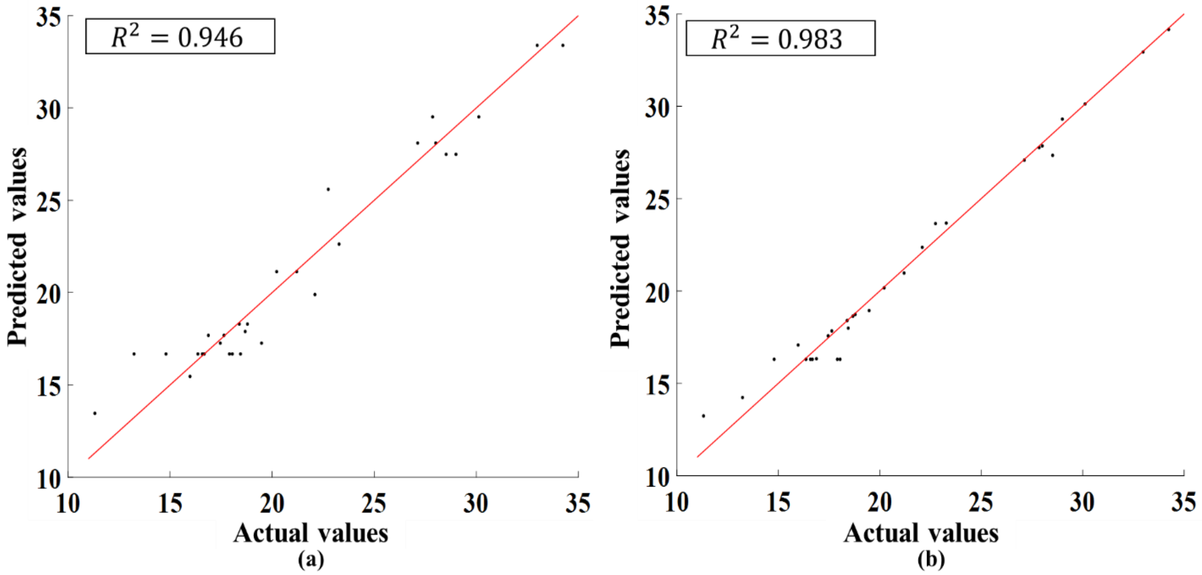


Figure 2.7. Scatterplots of predicted vs. actual measured values for compressive strength prediction by (a) RSM and (b) ANN

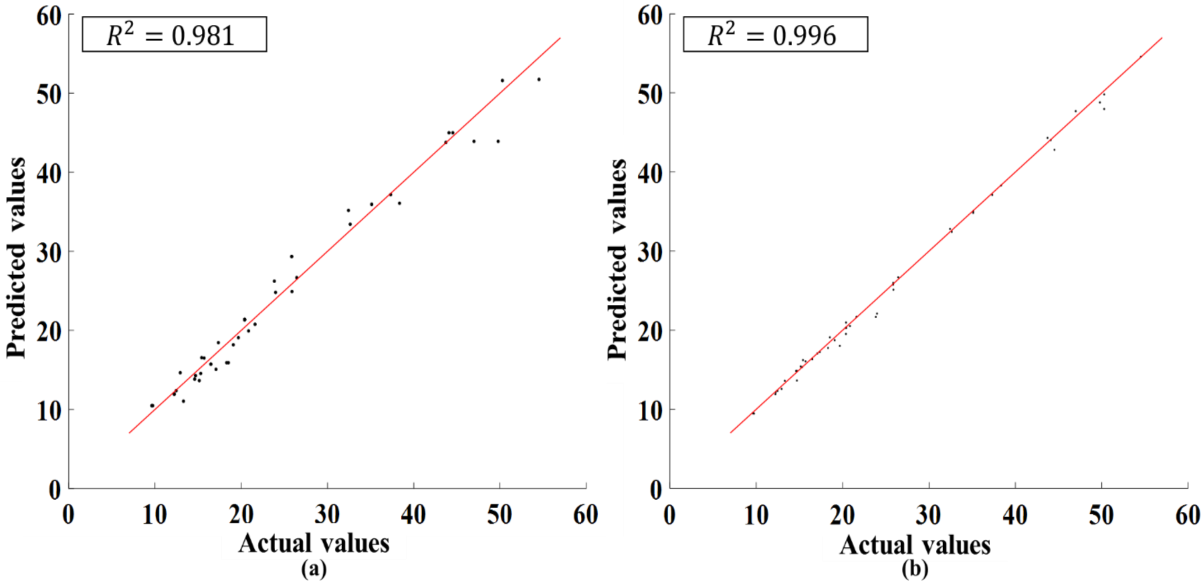


Figure 2.8. Scatterplots of predicted vs. measured values build time by (a) RSM and (b) ANN

Similar to the compressive strength prediction results, the ANN model for build time is fitted better than the RSM model, as shown in scatted plot in Figure 2.8. The high-value R^2 value of 0.996 for the ANN model proved that the ANN model fitted well, better than the RSM model.

2.4.4. Multiobjective Optimization Formulations

After the surrogate models have been developed, they can be incorporated into the MOO. The general MOO formulation to optimize compressive strength and build time, based on process parameters in the FFF process, is given below.

$$\max_{\boldsymbol{x}} CS$$

$$\min_{\boldsymbol{x}} BT$$

$$\text{subject to, } -1 \le x_j \le 1, j = 1,2,3,4$$

$$BT \ge 0, CS \ge 0 \tag{2.10}$$

where x_j is the process parameters. To be more precise, x_1 is layer thickness, x_2 is build orientation, x_3 is infill density, and x_4 is extrusion temperature. In this case study, two models developed by ANN and RSM are used as objective functions for the optimization problem. Further, three types of optimization algorithms, GA, NSGA-II, and PSO, are applied to optimize compressive strength and build time simultaneously for RSM and ANN models. There is a total of six combinations of hybrid methods compared in the case study. The combinations are listed in Table 2.7. The detail of the algorithm parameters and the discussion of the results is explained in the following section.

Table 2.7. Various approaches employed in the case study

Surrogate models	Optimization algorithms	Hybrid methods (surrogate models + Optimization algorithms)	
RSM	GA	RSM-GA	ANN-GA
ANN	NSGA-II	RSM-NSGA-II	ANN-NSGA-II
-	PSO	RSM-PSO	ANN-PSO

2.4.5. Optimization Results

The formulated MOO in this case study deals with optimizing conflicting objectives, where the increase in one objective decreases the other objectives. Thus, there is no one unique global solution to the MOO problem; instead, the optimal solution is represented by a set of non-dominated optimal solutions called the Pareto frontier. This Pareto frontier contains a set of solutions that describe the Pareto optimal solutions, often shown graphically as a chart. The Pareto frontier obtained from the six approaches listed in Table 2.7 is also detailed in this subsection.

GA is a widely used evolutionary algorithm for optimization problems in general. To solve the MOO formulations, the GA parameters used to maximize compressive strength and minimize build time are set as follows. Population size: 50, the number of generations: 100, mutation rate: 0.01, crossover rate: 0.8, and chromosome length: 24. The Pareto frontier diagrams generated for RSM and ANN models with GA are shown in Figure 2.9.

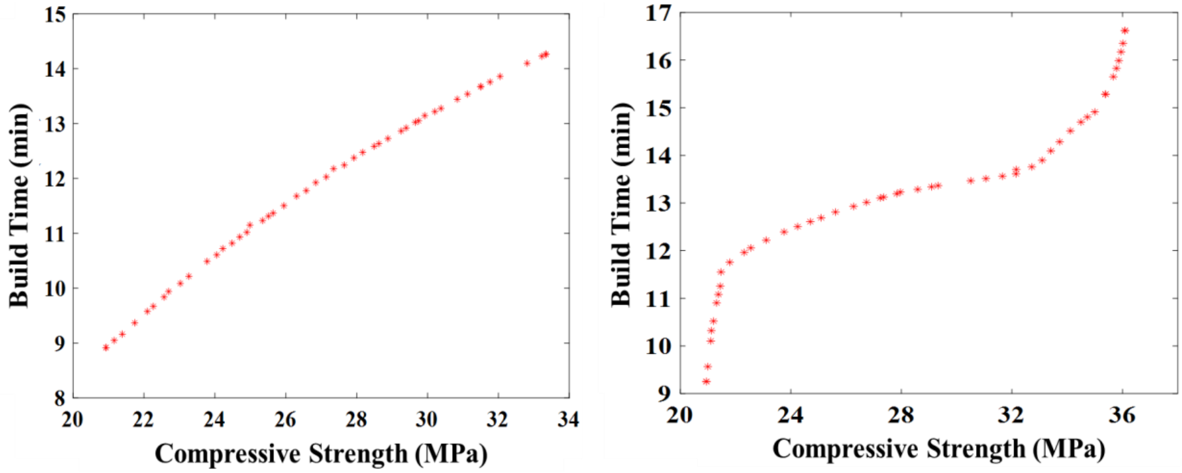


Figure 2.9. Pareto frontier by GA for (a) RSM and (b) ANN

NSGA-II is another evolutionary algorithm used to solve the multiobjective optimization problem in general. The Pareto frontier charts generated for RSM and ANN models by employing NSGA-II are shown in Figure 2.10. In this case study, the parameters considered for NSGA-II are the population size of 50, the maximum generation of 100, the crossover rate of 0.95, and the mutation rate of 0.1.

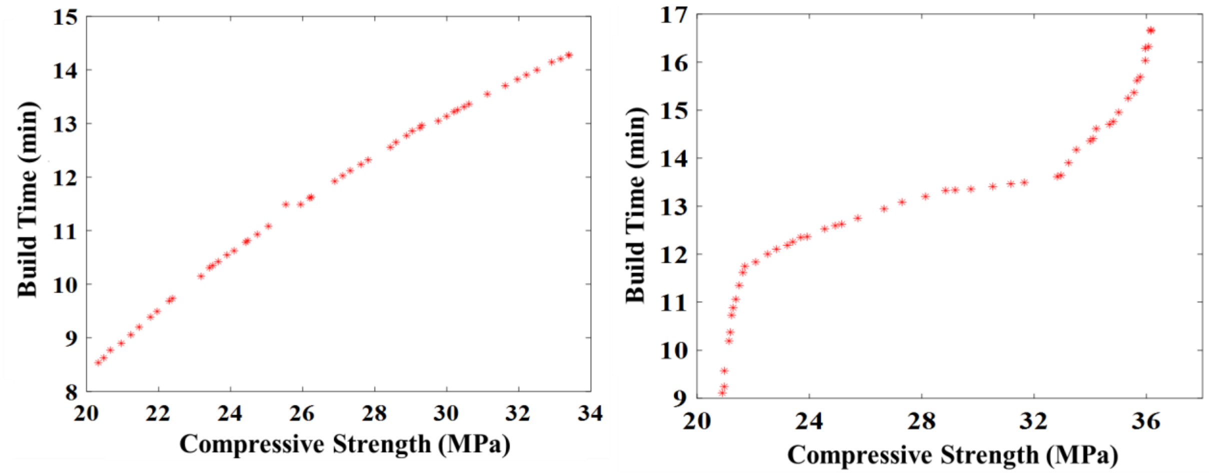


Figure 2.10. Pareto frontier by NSGA-II for (a) RSM and (b) ANN

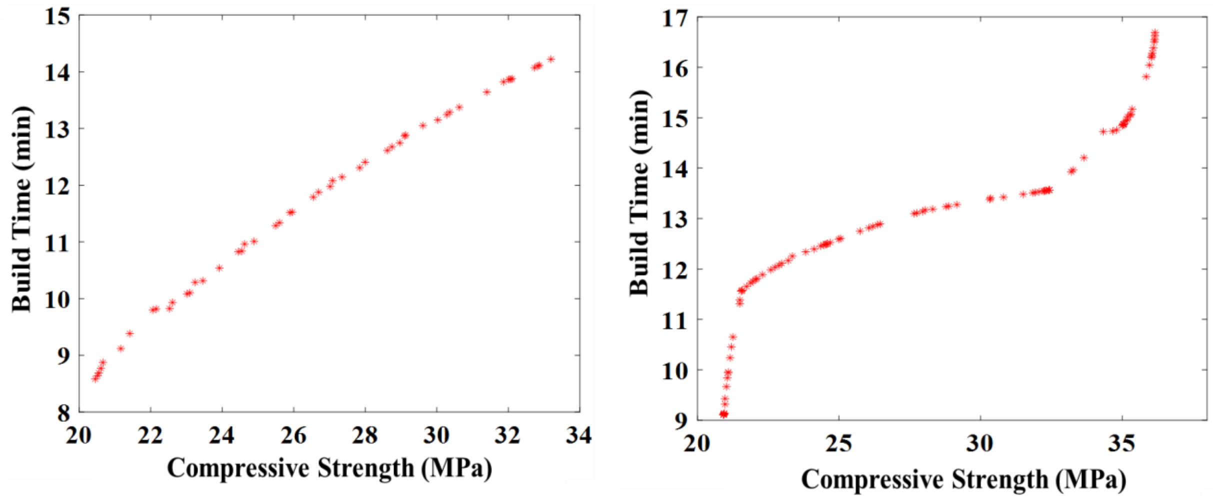


Figure 2.11. Pareto frontier by PSO for (a) RSM and (b) ANN

PSO is a popular biologically-inspired algorithm often employed to solve multiobjective optimization. In this case study, the parameters used for PSO algorithms are the population and repository size of 50, and the maximum generation is set to 200. The personal learning coefficient and global learning coefficient are adjusted to 2. Moreover, the mutation rate, inertia weight, and inertia dumping weight are 0.1, 0.5, and 0.99, respectively. The Pareto frontier graphs generated for RSM and ANN models obtained from the PSO approach are shown in Figure 2.11. In addition to the Paetro frontier, the quantitative results are obtained based on the maximum and minimum values and range values. The quantitative results for all combinations from the two surrogate models and three optimization models are summarized in Table 2.8.

Table 2.8. Maximum, minimum, and range values for all presented hybrid combinations

	RSM					
	GA		NSGA-II		PSO	
	Compressive Strength (MPa)	Build Time (min)	Compressive Strength (MPa)	Build Time (min)	Compressive Strength (MPa)	Build Time (min)
Min	20.329	8.5786	20.327	8.536	20.503	8.630
Max	33.365	14.269	33.387	14.276	33.387	14.276
Range	13.039	5.691	13.059	5.739	12.884	5.645
	ANN					
	GA		NSGA-II		PSO	
	Compressive Strength (MPa)	Build Time (min)	Compressive Strength (MPa)	Build Time (min)	Compressive Strength (MPa)	Build Time (min)
Min	20.911	9.196	20.172	9.443	20.922	9.119
Max	36.117	16.687	36.156	16.666	36.142	16.561
Range	15.205	7.490	15.984	7.223	15.220	7.441

2.4.6. Discussions

From the optimization results presented, it should be noted that the results vary for different hybrid approaches employed. Based on Table 2.8, the minimum value for compressive strength varies around 20 -20.9MPa for all combinations. In contrast, the maximum value for compressive strength is around 33.3MPa for all optimization algorithms that employed RSM for surrogate models and about 36.1MPa for those based on ANN as its surrogate models. Similarly, for the build time results, the minimum values for algorithms using RSM are around 8.5-8.6 minutes and 9.1-9.4 minutes for the ANN approach. The maximum values for build time with RSM as surrogate models are approximately 14.2 minutes and 16.6 minutes for ANN as surrogate models.

Although there are some deviations in maximum and minimum values, the range values for both part properties remain the same for all three optimization algorithms using the same surrogate models. The results also point out that GA, NSGA-II, and PSO, work comparatively well in this case study. The deviations in the results are more noticeable when comparing the two surrogate models, RSM and ANN. It can be further interpreted that surrogate models play a significant part in multiobjective optimization problems.

Qualitative results from Figure 2.9 to Figure 2.11 note that the Pareto frontier for all optimization algorithms follows the same pattern. However, the Pareto frontier patterns for RSM and ANN are different. This is because RSM and ANN developed different surrogate models. Although the Pareto frontier patterns are not precisely the same for RSM and ANN, both Pareto frontier charts show the same increasing trend. This trend also means a longer build time is required to achieve higher compressive strength. As the objective is also to minimize the build time, it should be noted that the points obtained and plotted on all the Pareto frontier charts are the non-dominated solutions. It is also essential to involve the decision-makers or implement another decision-making analysis to decide on the best combination of process parameters for compressive strength and build time from the non-dominated solutions obtained.

2.5. Summary

In this chapter, two approaches introduced to develop the surrogate model are RSM and ANN. MSE and MAE matrices are used to compare RSM and ANN. It has been shown that

MSE and MAE are the lowest for ANN. Therefore, the prediction accuracy obtained is better for ANN because ANN can capture complex functional relationships between process parameters and part properties. In addition, three evolutionary (or nature-inspired) algorithms, GA, NSGA-II, and PSO, are also presented to solve optimization problems with the surrogate models obtained by RSM and ANN approaches. The demonstrated case study aims to optimize compressive strength and build time by optimizing four process parameters: layer thickness, build orientation, infill density, and extrusion temperature. The optimization results from all hybrid combinations of surrogate models and optimization algorithms exhibit minimal deviations. It is also concluded that generating an accurate surrogate model is essential before employing it in any optimization solver. In addition to the FFF application, the proposed framework and algorithms presented in this chapter can be applied to other fields.

CHAPTER 3. UNCERTAINTY INTEGRATION FOR ROBUST DECISION-MAKING PROCESS

3.1. Overview

Continuing from the previous chapter, a response surface model (RSM) that explores the relationship between input variables (design parameters) and a response can be developed from the experimental outcomes. A combination of input variables is estimated for an optimum response value by optimizing the developed response surface model under certain constraints. In optimization, it is assumed that the response surface model parameters are deterministic, but model parameters are not deterministic in many real-world applications. To close this gap, a robust response surface optimization (RRSO) approach is proposed for optimizing the distribution of a response by considering response surface model parameter uncertainty. In the proposed framework, the mean and variance of response are optimized simultaneously, and the robustness of the response is achieved using the variance to measure the variation of the response. The proposed framework generates a Pareto frontier that consists of a set of non-dominated solutions and represents a tradeoff between mean and variance of compressive strength. The proposed method is more practical instead of a deterministic response surface model when a response is uncertain due to different factors such as environmental noise and natural randomness.

Different mathematical models such as RSM and ANN can be used to develop functional relationships between input variables (design parameters) and response. These mathematical models analyze quantitative (numerical) input variables. But, in experimental investigation, the input variables can be quantitative, qualitative (categorical), and a combination of both types. The information obtained from different sources, such as experimental data or expert knowledge (user experience), should be considered to make a relatively sound decision. Besides, selecting the optimal combination of process parameters is complicated further by uncertainties in different stages of the additive manufacturing processes. A framework based on the Bayesian network (BN) approach is proposed to enhance the decision-making process in determining the optimum combination of input variables for a desired level of response. The proposed approach has several advantages: 1) account for uncertainties, 2) incorporate information from multiple sources, and 3) analyze both quantitative and qualitative input variables. A FFF dataset is used to demonstrate the robust response surface optimization and Bayesian Network techniques.

3.2. Robust Response Surface Optimization

The response surface model (RSM) is a widely applied collection of statistical techniques that represents the relationship between some input variables (design parameters) and one or more output variables (responses). The concept of RSM was introduced by Box and Wilson in the 1950s [36]. RSM is now used in diverse fields, including manufacturing, biological and clinical science, and pharmaceutical science [37]. The goal of using RSM is to analyze the impacts of input variables on the response and optimize responses to select an optimum combination of input variables. In RSM, the coefficients of different terms (e.g., first-order, polynomial, and interaction terms) and intercept are known as RSM parameters. In the optimization of response by using RSM and experimental constraints, it is assumed that RSM parameters are fixed and deterministic [38]. But the deterministic assumption is not always accurate due to different sources of uncertainty in the experimental data collection environments [10]. Environmental noise, human error, measurement error, and natural randomness are possible sources of uncertainty in data collection.

More than one sample is produced for a combination of input variables to reduce the impacts of uncertainty sources. The average value of the response is used in developing RSM [39]. But still, model parameters are constant, and the estimated value of a response by an RSM is a deterministic point value for a combination of input variables. On the other hand, when an experiment is repeated several times for the same combination of input variables, it is unlikely that the response will be exactly the same for all experimental runs due to uncertainty. Therefore, the uncertainty aspect should be incorporated in optimizing a response with the RSM approach to obtain a more robust decision. The interval estimation and estimation of a probability distribution of a target variable are superior to deterministic point value estimation [40, 41].

Here, a robust response surface optimization (RRSO) approach that optimizes the distribution of the response to incorporate the model uncertainty is proposed. In the proposed method, the mean and variance of response are optimized simultaneously using a multiobjective optimization approach. In developing RSM, it is assumed that the error is normally distributed with a zero mean. The distribution of RSM parameters can be estimated from the distribution of the error. Further, the mean and variance of the response can be estimated from the distribution of model parameters. In the proposed RRSO approach, the mean of the response is optimized, and the variance of the response is minimized simultaneously, assisting in risk-informed decision-making while selecting input variables for a response.

3.2.1. Proposed Method

RSM is a mathematical model used to develop a functional relationship between input variables and response. In general, the actual functional relationship between input variables and the response is unknown but can be approximated by lower-order polynomial functions. RSM is represented as follows

$$y = f(x_1, x_2, x_3, I., x_k, \boldsymbol{\beta}) + \epsilon \tag{3.1}$$

In Equation (3.1), y is the response, $x_1, x_2, x_3, I., x_k$ are input variables, and $\boldsymbol{\beta}$ is the coefficient vector of different terms (e.g., first-order terms, interaction terms, and polynomial terms), including the intercept. ϵ is an error term. It is assumed that error is normally distributed with a zero mean and σ^2 variance. First-order, quadratic and third-order RSMs are applied in diverse fields based on the functional relationship between input variables and the response. In this research, the proposed model is developed for a quadratic RSM. The proposed RRSO approach can be applied to other RSMs as well. Generally, a quadratic RSM can be written as

$$y = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k} \beta_{jj} x_j^2 + \sum_{j=1}^{k} \sum_{l=j+1}^{k} \beta_{jl} x_j x_l + \epsilon = \boldsymbol{x\beta} + \epsilon \tag{3.2}$$

where, $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, I, \beta_k, \beta_{11}, \beta_{22}, I, \beta_{kk}, \beta_{12}, \beta_{13}, I, \beta_{(k-1)k})^T$. β_0 is an intercept, and β_j β_{jj}, β_{jl} are the coefficients of linear, quadratic, and interaction terms, respectively. k is the number of input variables and, $\boldsymbol{x} = (1, x_1, x_2, I, x_k, x_1^2, x_2^2, I., x_k^2, x_1x_2, x_1x_3, I, x_{k-1}x_k)$. $\epsilon \sim N(0, \sigma^2)$ is normally distributed with a zero mean and σ^2 variance.

In Equation (3.2), $\boldsymbol{\beta}$ is unknown and can be estimated from a collected dataset. The model parameters, $\boldsymbol{\beta}$ can be estimated by minimizing the least square error and can be expressed as

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \tag{3.3}$$

where, $\widehat{\boldsymbol{\beta}}$ is the least square estimator of $\boldsymbol{\beta}$, and $\boldsymbol{X} \in \boldsymbol{R}^{\boldsymbol{n}\times\boldsymbol{p}}$, where n is the number of observations, p is the number of terms in RSM, and each row of $\boldsymbol{X}$ represents an observation. $\boldsymbol{Y} \in \boldsymbol{R}^{\boldsymbol{n}}$ is the measured values of the response. The proposed method assumes that the error is normally distributed, and the error variance can be estimated using the following formula.

$$\hat{\sigma}^2 = \frac{1}{n-p-1}\left(\boldsymbol{Y} - \widehat{\boldsymbol{Y}}\right)^{\boldsymbol{T}}\left(\boldsymbol{Y} - \widehat{\boldsymbol{Y}}\right) \tag{3.4}$$

where, $\widehat{\boldsymbol{Y}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}$ is the estimated value of $\boldsymbol{Y}$. The mean of ϵ is zero, and the estimated variance is $\hat{\sigma}^2$. Now, the distribution of $\boldsymbol{\beta}$ can be estimated as

$$\boldsymbol{\beta} \sim N\left(\widehat{\boldsymbol{\beta}}, \hat{\sigma}^2 (\boldsymbol{X}^T \boldsymbol{X})^{-1}\right) \tag{3.5}$$

The response y is a function of $\boldsymbol{\beta}$. Therefore, the first two moments of y can be approximated from the distribution of $\boldsymbol{\beta}$ by using a first-order Taylor series approximation as follows [42]:

Performance function:

$$Z = f(z_1, z_2, z_3, \cdots z_m) \tag{3.6}$$

First-order approximation of mean:

$$E(Z) \approx f\left(\mu_{z_1}, \mu_{z_2}, \cdots, \mu_{z_m}\right) \tag{3.7}$$

First-order approximation of variance:

$$var(Z) \approx \sum_{p=1}^{m} \left(\frac{\partial f}{\partial z_p}\right)^2 var\left(z_p\right) + 2 \sum_{p=1}^{m} \sum_{q=\mathrm{p}+1}^{m} \frac{\partial f}{\partial z_p} \frac{\partial f}{\partial z_q} cov\left(z_p, z_q\right) \tag{3.8}$$

The mean and variance of the function Z given in Equation (3.6) can be estimated by using the first-order Taylor series approximation formulae given in Equations (3.7) and (3.8), respectively. In Equation (3.8), the covariance, $cov(z_p, z_q)$, is zero when z_p and z_q are independent. The mean and variance of a quadratic RSM can also be estimated using the first-order Taylor series approximation. In the proposed method, it is assumed that all model parameters are independent. Therefore, the covariance terms are ignored for the variance approximation.

All terms in Equation (3.2) may not be significant for a response, and the stepwise elimination techniques are applied to remove the insignificant terms from an RSM. For estimating the mean and variance of y, only the significant terms obtained after applying a stepwise elimination technique are used, and the coefficients of insignificant terms are set to zero. The estimated RSM is used to optimize response and determine the optimum level of input variables. The traditional formulation for optimizing response from RSM is given in Equation (3.9).

$$\max_{x}/\min \ y$$

$$\text{subject to, } LB_j \leq x_j \leq UB_j \quad \forall j = 1,2, \dots, k$$

$$g_i(x) \leq 0, \forall i = 1,2, \dots, l \tag{3.9}$$

In the above optimization formulation, Equation (3.9), the response y is optimized under some constraints. LB_j and UB_j represent the lower and upper bounds of j^{th} input variable, respectively, and $g_i(x)$ is an inequality constraint. k is the number of input variables, and l is the number of inequality constraints. In most RSM optimization, inequality constraints are not present; in such cases, the constraints are only the range of the input variables.

A limitation of the above optimization is that the optimized value of y is a deterministic point value, and the uncertainty impacts are ignored in the above optimization. But this deterministic assumption may lead to an inaccurate or infeasible estimation in many real-life problems.

In this research, the RRSO approach is proposed to incorporate uncertainty in RSM's response optimization. In the proposed method, the mean and variance of the response are optimized simultaneously to determine the distribution of the response. The robustness of the RRSO approach is achieved by using the variance to measure the variation of the response. The mathematical formulation of the proposed RRSO method is given in the following equation.

$$\underset{\boldsymbol{x}}{\text{Max/min}}\, E(y)$$

$$\min_{\boldsymbol{x}} \quad var(y)$$

$$\text{subject to, } LB_j \leq x_j \leq UB_j \; \forall j = 1,2,\dots,k$$

$$E\big(g_i(x)\big) + k_1\sqrt{var\big(g_i(x)\big)} \leq 0, \forall i = 1,2,\dots,l \tag{3.10}$$

In the above optimization formulation, Equation (3.10), $E(.)$ and $var(.)$ are the mean and variance of response y, respectively. In the proposed method, the mean and variance of response are optimized instead of a deterministic point value estimation. A multiobjective optimization approach can be applied to solve the proposed RRSO. The multiobjective optimization formulation generates a set of non-dominated solutions that represent a trade-off between the mean and variance of the response. From all non-dominated solutions, users will select a solution based on their requirements and the level of uncertainty that the users consider acceptable. The proposed method is applied to a FFF dataset to demonstrate its effectiveness of the proposed method. A Pareto frontier is generated for the dataset, where each point represents a non-dominated solution.

3.2.2. Case Study: Experimental FFF Dataset

To demonstrate the proposed RRSO method, the FFF dataset described in Chapter 2 is used. In the dataset, the FFF process is used to print parts from different combinations of four FFF process parameters. The compressive strength and build time are measured for each build part. To analyze the effectiveness of the proposed RRSO, the compressive strength is considered as the party property. The mean and variance of compressive strength are optimized simultaneously. Same as in the last chapter, a quadratic RSM is developed from the collected dataset by minimizing the least square error given in Equation (3.3). The quadratic RSM in Equation (3.11) consists of only significant terms, and y is the compressive strength. All other terms are insignificant and removed from the quadratic RSM. The backward elimination technique is used to reduce insignificant terms with a 95% level of confidence.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 \tag{3.11}$$

By using the formula given in Equation (3.5), the estimated distribution of β is given below.

$$\beta \sim N\left(\begin{bmatrix} 16.6792 \\ 1.2167 \\ 1.4839 \\ 3.2167 \\ 7.4292 \\ -3.4175 \\ 2.9100 \end{bmatrix}, diag(0.2023, 0.1349, 0.1349, 0.1349, 0.3372, 0.1517, 0.1517)\right) \tag{3.12}$$

In this research, it is assumed that all model parameters are independent, and all covariance terms are ignored and assumed to be zero. Thus, $var(\boldsymbol{\beta})$ is a diagonal matrix.

Coded levels of process parameters between -1 and +1 are used to estimate the distribution of $\boldsymbol{\beta}$. Therefore, the range of process parameters is between -1 and +1, and there are no other constraints for multiobjective optimization.

Table 3.1. The optimal solution set

x_1	x_2	x_3	Compressive Strength (MPa)	
			Mean	Variance
-0.037	0.008	-0.074	16.417	0.203
0.480	-0.369	0.612	21.158	0.326
0.694	-0.692	0.923	26.530	0.621
0.864	-0.806	0.949	29.184	0.830
0.935	-0.910	0.999	31.454	1.039
1.000	-1.000	1.000	33.385	1.248

The mean of compressive strength is maximized, and the variance of compressive strength is minimized simultaneously by the proposed RRSO formulation given in Equation (3.10). The ε-constraint method is employed to solve the multiobjective optimization problem formulated by the proposed method. The outcomes of the optimization are summarized in Table 3.1.

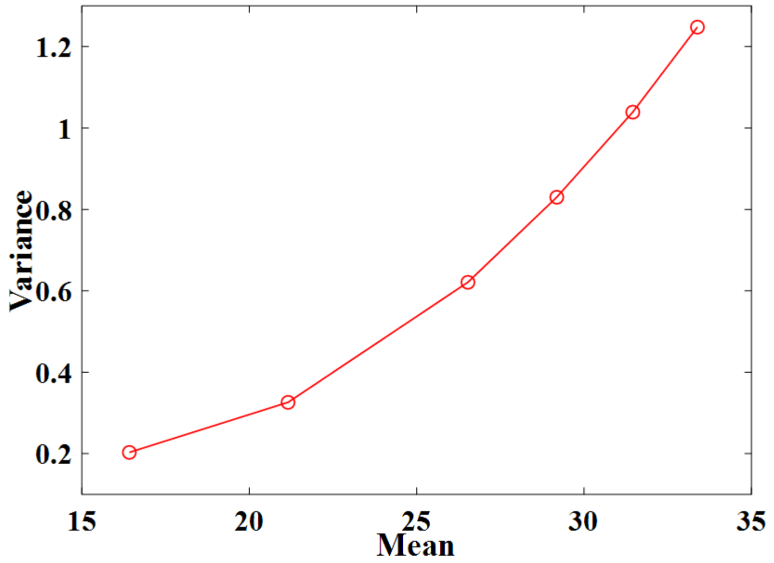


Figure 3.1. Pareto frontier

The Pareto frontier represents the trade-off between the mean and variance. The Pareto frontier for this case study is given in Figure 3.1. In the above figure, the mean is increasing with the increase of the variance. This phenomenon is expected as the goal of the optimization is to maximize the mean and minimize the variance of compressive strength.

The same problem is solved in Chapter 2 without considering the variance of compressive strength [18]. The compressive strength and build time are optimized simultaneously, and the obtained maximum value of the compressive strength is approximately 33.385 MPa when no weight is given to build time. In Table 3.1, the maximum compressive strength value is 33.385 MPa with a maximum variance of 1.248 MPa. In traditional RSM, the influence of the variance is ignored for compressive strength optimization. It indicates that ignoring variance generates a highly optimistic solution and does not always represent real-world situations that inherently have a high variance. The proposed RRSO method optimized the mean and variance with different combinations of input variables. The estimation of the distribution (mean and variance) gives additional information for selecting input variables for a response. The solution set generated by the proposed approach helps choose a solution with corresponding levels of input variables from all Pareto optimal solutions based on requirements and tolerance level compressive strength variation. In the following section, the Bayesian network (BN) is proposed to analyze input variables' impacts on responses. The proposed BN-based method incorporates uncertainty in estimating a combination of input variables for one or more responses.

3.3. Decision Analysis with Bayesian Network Approach

Generally, RSM is widely used for process parameter analysis. Additionally, it has been analyzed that ANN can be used over RSM for optimization, and a comparison study is performed in Chapter 2. Besides, an RRSO model is proposed in Chapter 3.2 for incorporating uncertainty in design parameter optimization. These methods are mainly useful only for quantitative variables. But, in the actual manufacturing environments, categorical design parameters also exist along with quantitative variables [43]. Thus, this section proposes a BN-based framework to estimate the levels of both quantitative and qualitative design parameters. In addition, the proposed framework also incorporates uncertainty in decision-making.

3.3.1. Bayesian Network Overview

Bayesian network (BN) is a directed acyclic graph (DAG) that represents the dependent relationships among variables through conditional probability distributions[44]. A simple BN with two levels is presented in Figure 3.2. In Figure 3.2, each node represents a random variable, and the edges (or arcs) show the dependent relationship among the variables. Figure 3.2 can be interpreted as random variables Y_j depend on all random variables $x_{i,}$ where I = 1,2,3,…k and j = 1,2,3,…m.

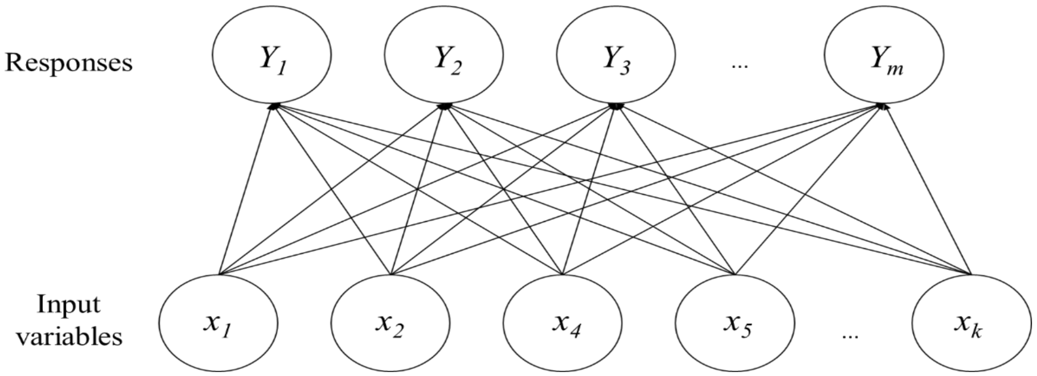


Figure 3.2. A representative of a two-level BN

Mathematically, based on the graphical representation above, the joint probability distribution can be represented as:

$$P(x,Y) = \prod_{j=1}^{m} P\big(Y_j \big| pa(Y_j)\big) \prod_{i=1}^{k} P(x_i) \tag{3.13}$$

where $x = (x_1, x_2, \dots x_k)$, $Y = (Y_1, Y_2, \dots Y_m)$, $pa(Y_j)$ is the set of parent nodes of Y_j. In this case, x_i are parent nodes of Y_j. $P(x_i)$ is the probability of the random variable x_i . $P\big(Y_j \big| pa(Y_j)\big)$ is the conditional probability of Y_j given $pa(Y_j)$. To calculate $P(x,Y)$, $P(x_i)$ for all i and $P\big(Y_j \big| pa(Y_j)\big)$ for all j must be determined.

All marginal and conditional probabilities can be determined from historical data, experimental data, expert opinion, or their combinations. In this research, all prior probabilities are determined from experimental data. Knowing the conditional probabilities between input variables and responses is necessary to develop a proper BN. The conditional probabilities represent the strength of dependency between input variables and responses. If prior information is available, various dependency tests can be employed before developing the BN. The selection of proper methods depends on the types of data and whether the variables are quantitative and/or qualitative (categorical) data. Dependency between two variables can be determined by different approaches such as Pearson correlation, point biserial correlation, Chi-squared test, or Cramer's V [45-47]. For instance, Pearson correlation is used to determine a linear correlation between two quantitative variables. The dependency between two categorical variables can be estimated by the Chi-squared test and Cramer's V. On the other hand, the biserial point correlation is used to estimate the dependency between a quantitative variable and a qualitative variable. All variables are quantitative in the dataset used to demonstrate the proposed BN. Therefore, Pearson correlation is used to find dependency relationships between input variables and response. The dependency relationship between an input variable and the response is determined through their Pearson correlation coefficient. Pearson correlation coefficient between two random variables (x and Y), r_{xY}, can be defined as:

$$r_{xY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{3.14}$$

where $\bar{x}$ and $\bar{Y}$ are the sample means of two random variables x and Y, respectively. I is the index of individual sample points. The total number of sample sizes or possible pairs between x and Y is n. The Pearson correlation coefficient lies between -1 and +1, where -1, 0, and +1 indicate a strong negative correlation, no correlation, and a strong positive correlation, respectively. In this research, it is assumed that a process parameter and a part property are independent if the Pearson correlation is lesser than 0.1, $|r_{xY}| \leq 0.1$.

3.3.2. Case Study: Experimental FFF Dataset

In this section, to analyze the proposed method's effectiveness, the impacts of four process parameters (layer thickness, build orientation, infill density, and extrusion temperature) on two-part properties, namely compressive strength and Young’s modulus, are analyzed. The collection process is described in Chapter 2. From the collected data, quadratic RSMs for both compressive strength and Young’s modulus are generated. The compressive strength and Young’s modulus can be estimated for any values of process parameters from the RSM. Since each process parameter has three (03) levels, and there is a total of four (04) process parameters, the total possible combinations are 3^4 or 81 combinations. For each possible combination, two to five data points are randomly produced. In data generation, a random value from a 95% confidence interval is picked for both part properties instead of estimating a constant value. The generated large dataset from small experimental data is used in the rest of the chapter.

For simplification purposes, all process parameters are assumed to be independent. Thus, as depicted in Figure 3.2, there are no edges between random variables x_i. All four process parameters and the two-part properties are quantitative. The dependency relationship between a process parameter and a part property is determined through their Pearson correlation coefficient. Based on the experimental data, the Pearson correlation coefficient for each possible combination of x and Y is summarized in Table 3.2.

Table 3.2. Pearson correlation coefficients

Process Parameters	Compressive Strength, MPa (Y_1)	Young’s Modulus, MPa (Y_2)
1. Layer thickness, x_1	0.17	-0.14
2. Build orientation, x_2	0.22	-0.14
3. Infill density, x_3	0.52	0.36
4. Extrusion temperature, x_4	0.05	0.13

From Table 3.2, it can be further interpreted that the compressive strength does not highly depend on extrusion temperature as the correlation coefficient found is 0.05. The most significant process parameter for both part properties is infill density. The updated BN, based on the outcomes of Pearson correlation, is demonstrated in Figure 3.3.

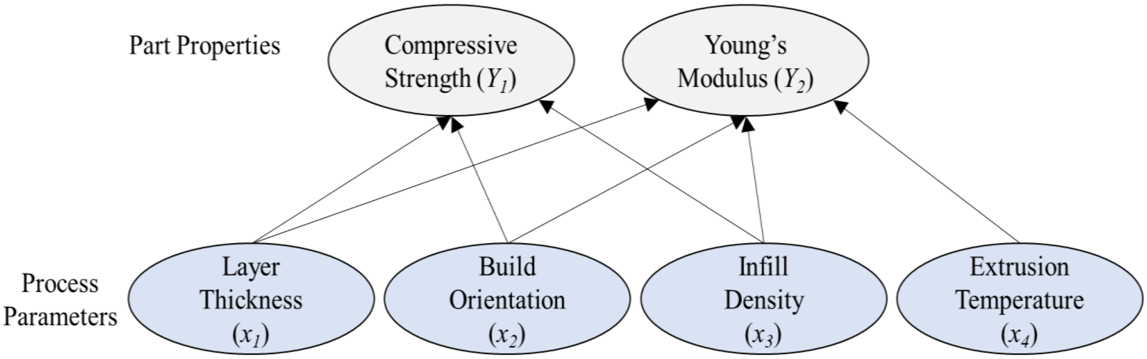


Figure 3.3. The BN structure obtained after Pearson correlation analysis

There is no arc between compressive strength (Y_1) and extrusion temperature (x_4) as both random variables are assumed to be independent. Therefore, compressive strength (Y_1) has only three parent nodes, layer thickness (x_1), build orientation (x_2) and infill density (x_3). On the other hand, Young's modulus (Y_4) has four (04) parent nodes. Young's modulus depends on all four (04) process parameters, layer thickness (x_1), build orientation (x_2), infill density (x_3), and extrusion temperature (x_4). The joint probability distribution of process parameters and part properties can be mathematically formulated based on the updated BN, as:

$$P(Y_1, Y_2, x_1, x_2, x_3, x_4) = P(x_1)P(x_2)P(x_3)P(x_4)P(Y_1|x_1, x_2, x_3)P(Y_2|x_1, x_2, x_3, x_4) \qquad (3.15)$$

In this scenario, the data is collected considering three levels of process parameters, high, center, and low. The continuous process parameters are converted into discrete parameters or random variables. In this way, the optimum combinations of both discrete and continuous process parameters can be determined together. The chance of selecting each level is equally likely for all parameters as a manufacturer can select any level of each parameter based on their requirements. Each level for all process parameters is assumed to be independent. Therefore, the probabilities of $x_1, x_2, x_3,$ and x_4 are assumed to be equal for all three levels, $P(x_1) = P(x_2) = P(x_3) = P(x_4) = \frac{1}{3}$. Thus, Equation (3.15) can be rewritten as Equation (3.16) to account for the prior probability.

$$P(Y_1, Y_2, x_1, x_2, x_3, x_4) \propto P(Y_1|x_1, x_2, x_3)P(Y_2|x_1, x_2, x_3, x_4) \tag{3.16}$$

The probability of achieving the desired requirement for compressive strength and Young's modulus simultaneously for each combination of $x_1, x_2, x_3,$ and x_4 can be evaluated by Equation (3.16). The combination of process parameters with the highest probability may be regarded as the possible optimum combination to achieve desired levels of part properties. The optimum levels of $x_1, x_2, x_3,$ and x_4 for $Y_1$and Y_2 can be determined by Equation (3.17).

$$X_{Y_1,Y_2} = \underbrace{argmax}_{x_1,x_2,x_3,x_4} P(Y_1, Y_2, x_1, x_2, x_3, x_4) \tag{3.17}$$

where, X_{Y_1,Y_2} represents the selected combination of process parameters $x_1, x_2, x_3,$ and x_4 that result in the highest probability of achieving the required Y_1 and Y_2. To demonstrate the proposed BN-based method for process parameter analysis, five (05) scenarios with different desired requirements for compressive strength and Young's modulus are considered. Compressive strength and Young's modulus must be achieved simultaneously with one combination of process parameters to mimic the actual printing process. Before the actual printing process, process parameters have to be predetermined. Changing any process parameter in the middle of the printing process is not recommended. All prior probabilities are determined based on the collected experimental data.

To determine the optimum combination of process parameters for all scenarios, the probabilities $P(Y_1, Y_2, x_1, x_2, x_3, x_4)$ for all combinations (81) of x_1, x_2, x_3 and x_4 are initially evaluated with Equation (3.16). Further, the optimum level for $x_1, x_2, x_3,$ and x_4 is obtained from Equation (3.17). A combination of process parameters is optimal for which the probability is maximum. In this research, the optimal combination of process parameters is determined for five scenarios; the different compressive strength requirements and Young's modulus. For each scenario, the selected level of process parameters is shown in Table 3.3. For instance, in the first scenario, the target value of compressive strength should be greater than or equal to 30MPa $(Y_1 \geq 30MPa)$, and the target value of Young's modulus should be greater than or equal to 900MPa $(Y_2 \geq 900MPa)$. For all 81 combinations of process parameters, the probability of $Y_1 \geq 30$MPa and $Y_2 \geq 900$MPa is determined from experimental data.

Table 3.3. The selected levels of process parameters for compressive strength and Young's modulus

Scenarios	Targeted values of part properties (MPa)	The selected level of process parameters				Probability $P(.)$
		Layer thickness, x_1	Build orientation, x_2	Infill density, x_3	Extrusion temperature, x_4	
1	$Y_1 \geq 30$ & $Y_2 \geq 900$	0.34mm	0°	80%	any levels	0.917
2	$20 \leq Y_1 \leq 25$ & $Y_2 \geq 900$	0.22mm	0°	80%	any levels	0.500
3	$25 \leq Y_1 \leq 30$ & $Y_2 \geq 900$	0.1mm	90°	20%	230°	0.875
4	$25 \leq Y_1 \leq 30$ & $700 \leq Y_2 \leq 800$	0.1mm	90°	50%	200°	0.917
5	$20 \leq Y_1 \leq 30$ & $700 \leq Y_2 \leq 800$	0.1mm	90°	50%	200°	0.917

The optimal combination is also given in Table 3.3. For example, in Scenario 1, the results can be further interpreted as there is a probability of around 92% to produce a part with compressive strength $Y_1 \geq$ 30MPa and Young's modulus $Y_2 \geq$ 900MPa, simultaneously, if the layer thickness is set at 0.34 mm, build orientation is set at 0°, infill density is set at 80%, and at any levels of extrusion temperature. Similarly, the selection of the level of process parameters for the remaining four scenarios can be interpreted.

From Table 3.3, it is clear that the selected levels of process parameters and the resulting probability depend on the target values for compressive strength and Young's modulus. Beyond the five scenarios in Table 3.3, the levels of process parameters with probabilities for any targeted values of compressive strength and Young's modulus can be determined. In addition, since the prior probabilities for BN construction are obtained from experimental data, the specimen design and the process parameter levels can impact the resulting probability. Thus, to ensure the accuracy of the resulting likelihood, the BN should be periodically updated when there is any new relevant information.

3.4. Summary

In this chapter, RRSO and BN are proposed for analyzing relationships between input variables and responses. Both methods have their advantages and are applied to FFF datasets.

The proposed RRSO approach is capable of optimizing the distribution of a response instead of optimizing the deterministic point value. In multiobjective optimization, the mean is optimized, and the variance is minimized to generate a set of Pareto optimal solutions. The RRSO incorporates model parameter uncertainty in determining an optimum combination of process parameters.

The FFF process has some qualitative process parameters, such as infill pattern and the number of shells. Thus, the BN approach is explored to enhance the decision-making process in selecting an optimum combination of process parameters that accounts for uncertainties. One of the advantages of BN is that it can handle both quantitative and qualitative input variables together. The BN is useful to determine optimum levels of quantitative and qualitative process parameters. The expected probability of achieving multiple part properties simultaneously can be evaluated by setting the desired target value for each part property.

CHAPTER 4. INTERVAL PREDICTION FOR MODEL UNCERTAINTY IN DECISION-MAKING

4.1. Overview

Data-driven algorithms (e.g., neural networks) have been widely applied in predicting tool wear because of the high prediction performance of the algorithms, availability of data, and advancements in computer powers in recent years. Although most algorithms are supposed to generate outcomes with high precision and accuracy, this is not always true in practice. Uncertainty exists in different phases of applying data-driven algorithms due to noises and randomness in data, the presence of redundant and irrelevant features, and model assumptions. Data uncertainty and model uncertainty in applying data-driven methods are the two most common types of uncertainty. To incorporate both types of uncertainties, empirical mode decomposition (EMD) is applied to reduce uncertainty from signal data. The model uncertainty is estimated by applying the MC dropout in the neural network algorithm. The unique feature of the proposed method is that it can predict the interval of tool wear. A neural network model can predict different tool wear values for a new observation by employing MC dropout. The predicted tool wear is represented in an interval, and the interval range represents the degree of uncertainty. The proposed method is applied in a real-world manufacturing dataset for tool wear prediction as intervals.

Other types of data-driven methods, such as linear and polynomial regression, are also widely used for target variable prediction. The MC dropout is only applicable for neural networks where neurons exist. Here, linear optimization-based techniques are proposed to predict conservative confidence intervals for linear and polynomial regression models. Two linear optimization models are proposed, one for ordinary least squares (OLS) regression and the other for weighted least squares (WLS) regression. The proposed methods are implemented on an experimental FFF dataset to demonstrate the effectiveness of the proposed methods for conservative interval prediction.

4.2. Interval Tool Wear Prediction by MC Dropout

A smart manufacturing system is a technology-driven approach that utilizes sensors, the internet of things (IoT), artificial intelligence, and data analysis algorithms to improve the efficiency and effectiveness of the production processes [48]. In many manufacturing industries, such as aerospace and automobile, different types of sensors (e.g., vibration and acoustic sensors) are installed to collect real-time data for condition monitoring and maintenance decision-making. Various data analysis approaches are applied to use the collected sensor data to support the decision-making processes in engineering maintenance and prognostic applications by predicting different target variables from the data. The goals of using data analysis in condition monitoring of a system are to avoid unscheduled failures, reduce spare parts inventory, and minimize maintenance costs.

The machining processes are widely used to produce the desired form of parts from different materials in different industries. Tool failure is one of the common reasons for the quality degradation of machined parts. Tool flank wear is a common failure affecting tool and workpiece properties [49, 50]. Different types of sensors (e.g., vibration sensors) are used to monitor the condition of cutting tools for tool flank wear analysis. Predicting tool flank wear from sensor data can reduce unscheduled shutdown by estimating tool changing time earlier. Due to the availability of data collected by sensors and different advanced algorithms, data-driven approaches are considered the new trend for tool condition monitoring. Data-driven methods such as support vector machine (SVM) and neural networks (NNs) have been used for tool flank wear estimation from condition monitoring data. In-depth knowledge about the physics of a system is not entirely necessary for data-driven methods [51]. A large volume of relevant historical data is required to train a data-driven model, and the accuracy of the prediction depends on the quantity and quality of data. Data-driven methods can be modified easily to capture non-linearity and apply to different systems.

In most cases, the prediction by data-driven methods is a deterministic point value, and it is assumed that models predict with high accuracy. However, real-life problems are not deterministic due to different sources of uncertainty, and this deterministic prediction may lead to infeasibility or poor performance [25, 52]. Therefore, a target variable's interval prediction or probability distribution estimation is preferable to deterministic point value estimation. The uncertainty arises due to incomplete information and the random nature of a system [53]. The sources of uncertainty significantly impact processes, products, and collected data. Different artificial assumptions to develop a mathematical model are considered a source of uncertainty. Limited historical data to represent a system by a mathematical model is also a source of uncertainty. Product quality, experiment results, production cost, and financial decisions are directly impacted by these sources of uncertainty. The cost of ignorance of these sources of uncertainty is high, leading to wrong conclusions.

Similarly, a prediction by applying data-driven methods also consists of uncertainty due to the randomness of data, the noise in data collection, and model selection assumptions [54]. Model uncertainty and data uncertainty are the two most common sources of uncertainty in applying data-driven prediction models [55]. The prediction accuracy deteriorates due to different sources of uncertainty. Model uncertainty arises due to model selection error, lack of sufficient training data, model bias, or model variance. The sensor imperfections, signal wire noise, and irrelevant and redundant features are common sources of data uncertainty. Both uncertainties are epistemic as they are reducible by collecting more data, undergoing data preprocessing steps, and selecting appropriate models. The accuracy and reliability of prediction by data-driven methods are influenced by uncertainty. Therefore, the interval prediction and prediction of a probability distribution of a target variable are superior to deterministic point value prediction to incorporate the influence of uncertainty in the application of data-driven methods.

A good standard or recommended method for considering uncertainty in tool wear prediction has not been established yet. Research on tool wear prediction under uncertainty is still in the beginning stage. Thus, a novel method for accounting uncertainty during tool wear prediction is proposed to contribute to the research area under the umbrella of data-driven tool wear prediction. Here, a dropout-based method for interval prediction by neural networks is proposed that can predict several values of a target variable (e.g., tool flank wear) for a new observation (e.g., sensor data) by applying a random dropout approach on a trained NN model. From the several predicted values, the prediction interval can be estimated. In the proposed method, the collected data from sensors are denoised using empirical mode decomposition (EMD) to reduce data uncertainty. And, MC dropout is applied to predict tool wear as intervals. MC dropout considers model uncertainty.

4.2.1. Proposed Dropout-based Prediction Framework

The MC dropout is used to predict tool flank wear in intervals by considering data and model uncertainty. The steps of the proposed framework are given in Figure 4.1.

The data collected from different sensors, such as vibration and force sensors, are used to tool flank wear prediction. In the first step, the collected sensor data are denoised by using EMD to reduce data uncertainty arising due to several reasons such as sensor noise, signal wire noise, and signal conditioners noise. In the next step, a total of eleven (11) time-domain statistical features are extracted from the denoised sensor data. After feature extraction, principal component analysis (PCA) is used to reduce the dimension of features and convert data with linearly independent features. Further, a NN model with a dropout approach is trained using the new data obtained from PCA. For the next step, the dropout approach is additionally applied to a new observation known as the MC dropout to get multiple tool wear values. In the final step, the interval of tool flank wear can be determined for the new observation from the predicted values. The details of each step are introduced in the following subsections.

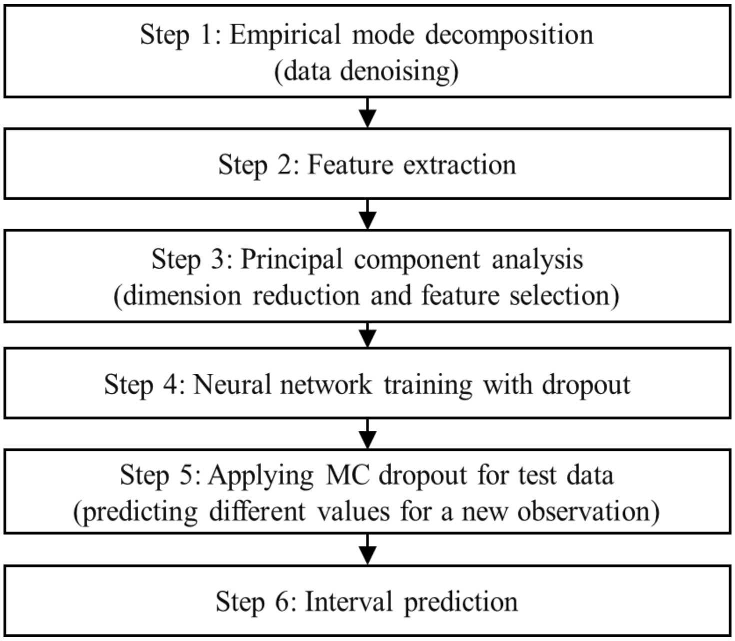

Step 1: Empirical mode decomposition
(data denoising)
Step 2: Feature extraction
Step 3: Principal component analysis
(dimension reduction and feature selection)
Step 4: Neural network training with dropout
Step 5: Applying MC dropout for test data
(predicting different values for a new observation)
Step 6: Interval prediction

Figure 4.1. The flow diagram of the proposed method

4.2.1.1. Empirical Mode Decomposition

The accuracy of prediction by NN significantly depends on the data quality. Several reasons that may cause sensor data to become uncertain are sensor noise, signal wire noise, and signal conditioners noise. This data uncertainty can be reduced by applying different data denoising algorithms. As this uncertainty is reducible, it is categorized as epistemic uncertainty. EMD method introduced by Huang et al. [56] is a well-known algorithm used for data denoising, reducing data's uncertainty (noise) to improve data quality. The EMD is an unsupervised data-driven signal decomposition algorithm and does not need any prior defined basis system. EMD decomposes a signal into a collection of intrinsic mode functions (IMFs) and a final residual [57]. An IMF must satisfy both of the following two criteria:

- For a given signal vector, the number of extrema and the number of zero crossings must either be equal or differ by at most one.
- At any point, the mean value of the envelope defined by the local maxima and the local minima is zero.

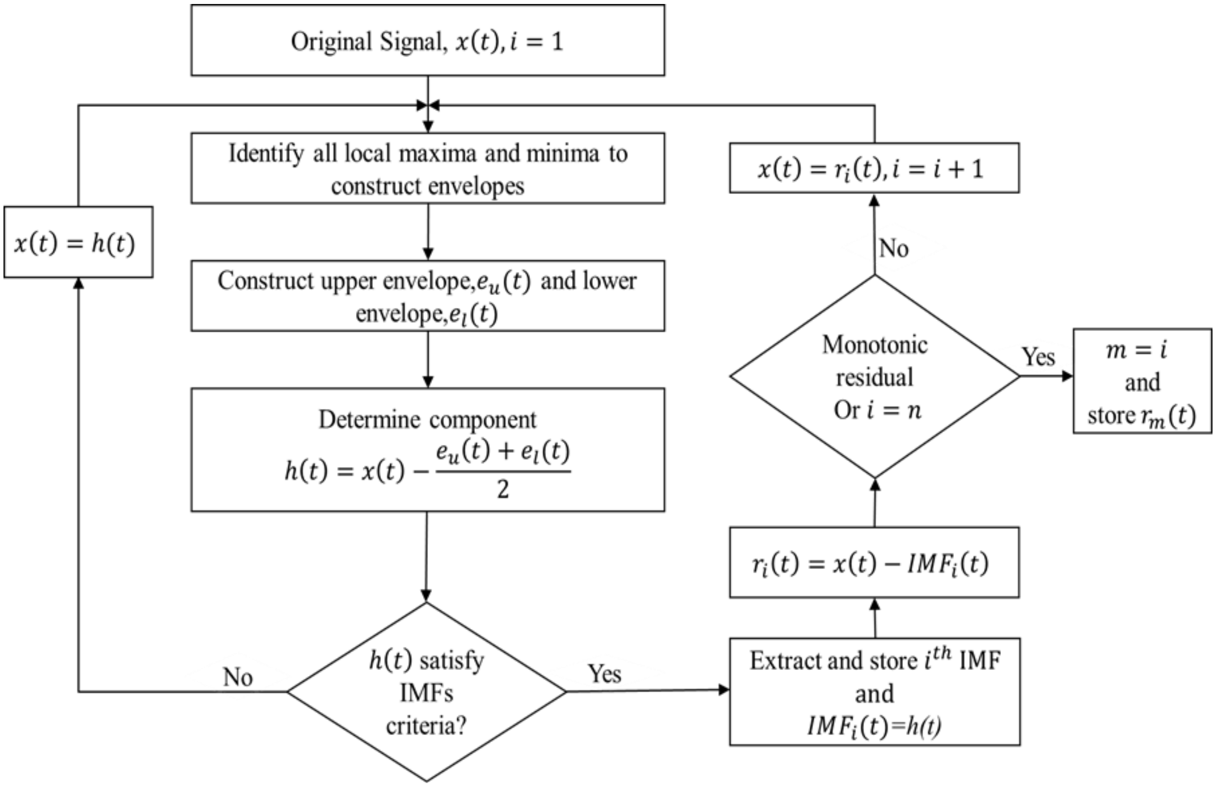

Original Signal, $x(t), i = 1$
Identify all local maxima and minima to construct envelopes
$x(t) = r_i(t), i = i + 1$
$x(t) = h(t)$
No
Construct upper envelope, $e_u(t)$ and lower envelope, $e_l(t)$
Monotonic residual Or $i = n$
Yes
$m = i$ and store $r_m(t)$
Determine component $h(t) = x(t) - \frac{e_u(t) + e_l(t)}{2}$
$r_i(t) = x(t) - IMF_i(t)$
No
$h(t)$ satisfy IMFs criteria?
Yes
Extract and store i^{th} IMF and $IMF_i(t)$=h(t)

Figure 4.2. A flow diagram of EMD

Some IMFs from all decomposed IMFs consists of noise, and the goal is to identify and subtract those IMFs from the original signal to get a noiseless signal. Before discussing the IMFs identification procedure, the signal decomposition algorithm to extract IMFs is given in Figure 4.2.

The extraction of IMFs is stopped when one of the following two conditions is satisfied: (1) a predefined number of IMFs are being extracted, or (2) the residual becomes monotonic from which no more IMF can be extracted [58]. In this research, it is assumed that the predefined maximum number of IMFs extracted is *m*=10.

When all IMFs are extracted, the signal can be expressed as,

$$x(t) = \sum_{i=1}^{m} IMF_i(t) + r_m(t) \tag{4.1}$$

where, *m* is the number of IMFs extracted from the original signal *x(t)*. $r_m(t)$ is the residual obtained after extracting the m^{th} IMF.

Consider the original signal *x(t)* is a collection of the noiseless signal $\tilde{x}\ (t)$ and noise *η(t)* as,

$$x(t) = \tilde{x}(t) + \eta(t) \tag{4.2}$$

The target is to estimate the denoised signal $\tilde{x}\ (t)$ from the original signal *x(t)* by removing the noise *η(t)*. In the decomposed signal, the first IMF contains the high-frequency terms, and the last IMF has the low-frequency terms [59]. It is also well-established and well-proven that high-frequency terms consist of more noise than low-frequency terms.

Consider the first *k* IMFs consist of noise. Therefore, $\tilde{x}\ (t)$ can be written as

$$\tilde{x}(t) = x(t) - \sum_{i=1}^{k} IMF_i(t) \tag{4.3}$$

The value of *k* can be determined by the correlation coefficient, *σ,* which is defined as,

$$\sigma = \frac{x(t)'\tilde{x}(t)}{\sqrt{x(t)'x(t)}\sqrt{\tilde{x}(t)'\tilde{x}(t)}} \tag{4.4}$$

Assume that the threshold value of σ is ρ. Then, the value of k can be determined by,

$$k^* = \min\{k \mid \sigma \leq \rho\} \tag{4.5}$$

when k^* has been determined, $\tilde{x}(t)$ is further estimated from Equation (4.3) by considering k=k^*. Generally, the threshold value of σ is assumed to be between 0.75 and 0.85 [59]. Here, it is considered that the threshold value σ is ρ = 0.8.

4.2.1.2. Feature Extraction

Feature extraction is a process by which an initial raw data set is reduced to more manageable groups of features for processing. It is necessary to extract all possible relevant features to increase prediction accuracy.

Time-domain, frequency-domain, and time-frequency domain features can be extracted from the denoised data. Among the three types of features, it is easy to extract time-domain features directly from denoised data [60]. Further data processing is required to extract the frequency-domain features [60, 61]. A high computation effort is necessary to extract time-frequency domain features from denoised data [62]. In this section, 11 time-domain features are extracted from denoised data obtained after applying EMD. The extracted features are mean, standard deviation, root mean square, square mean root, skewness, kurtosis, crest factor, shape factor, impulse factor, marginal factor, and peak to peak. The mathematical formulae for all features are given in Table 4.1.

Table 4.1. Time-domain features extracted from denoised data

Features	Formula	Features	Formula
1. Mean	$\mu = \frac{1}{N_s}\sum_{i=1}^{N_s}\tilde{x}_i$	7. Crest factor	$f_c = \frac{\max(\tilde{x}_i)}{rms}$
2. Standard deviation	$\sigma = \sqrt{\frac{\sum_{i=1}^{N_s}(\tilde{x}_i - \mu)^2}{N_s - 1}}$	8. Shape factor	$f_s = \frac{rms}{\frac{1}{N}\sum_{i=1}^{N_s}\lvert\tilde{x}_i\rvert}$
3. Root mean square	$rms = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}\tilde{x}_i{}^2}$	9. Impulse factor	$f_i = \frac{\max(\tilde{x}_i)}{\frac{1}{N_s}\sum_{i=1}^{N_s}\lvert\tilde{x}_i\rvert}$
4. Square mean root	$smr = \left(\frac{1}{N_s}\sum_{i=1}^{N_s}\sqrt{\lvert\tilde{x}_i\rvert}\right)^2$	10. Marginal factor	$f_m = \frac{\max(\tilde{x}_i)}{smr}$
5. Skewness	$f_{sk} = \frac{\sum_{i=1}^{N_s}(\tilde{x}_i - \mu)^3}{(N_s - 1)\sigma^3}$	11. Peak to peak	$\mathrm{f_{pp}} = \max(\tilde{x}_i)$-min $(\tilde{x}_i)$
6. Kurtosis	$f_k = \frac{\sum_{i=1}^{N_s}(\tilde{x}_i - \mu)^4}{(N_s - 1)\sigma^4}$	-	-

4.2.1.3. Principal Component Analysis

Principal component analysis (PCA) is one of the most widely used dimension reduction techniques that capture significant dataset variability and minimize the loss of information. The goal of PCA is to reduce the number of features with minimum loss of information. PCA is an unsupervised statistical technique that identifies new features along with new directions, which are linear combinations of the original features and uncorrelated. In this research, PCA is applied for two purposes: (1) to reduce the number of features and (2) to convert the features into linearly independent features. Reduced and uncorrelated features increase computational efficiency, model stability, and prediction accuracy. The steps of PCA are demonstrated as follows.

Step 1: Standardized data by subtracting the mean of features from all observations of the features.

$$\widetilde{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{1}^n\overline{\boldsymbol{X}} \tag{4.6}$$

where, $\boldsymbol{X} \in \boldsymbol{R}^{n\times p}$ is the feature matrix (data) obtained by feature extraction, $\overline{\boldsymbol{X}} \in \boldsymbol{R}^{1\times p}$ is the mean vector of features, $\widetilde{\boldsymbol{X}} \in \boldsymbol{R}^{n\times p}$ is the standardized data, and $\boldsymbol{1}^n$ is a column vector with *n* elements. *n* and *p* are the number of observations and number of features, respectively.

Step 2: Compute the covariance matrix, $\boldsymbol{S} \in \boldsymbol{R}^{p\times p}$ of *X*.

$$\boldsymbol{S} = \widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}} \tag{4.7}$$

where, $\widetilde{\boldsymbol{X}}^T$ represents the transpose of the matrix $\widetilde{\boldsymbol{X}}$.

Step 3: Compute the eigenvalues and eigenvectors of $\boldsymbol{S}$. In PCA, eigenvectors are the principal components. Then, order the eigenvectors according to the descending order of eigenvalues.

$$\boldsymbol{V} = [\boldsymbol{v}_1\ \boldsymbol{v}_2\ \boldsymbol{v}_3 \ldots \boldsymbol{v}_p] \tag{4.8}$$

where, each column of $\boldsymbol{V} \in \boldsymbol{R}^{p\times p}$ represents an eigenvector. v_i $(i = 1,2,3,\ldots,p)$ is the eigenvector corresponding to the *i*th largest eigenvalue, $\lambda_i\left(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p\right)$. In eigenspace, the direction of the eigenvector (v_1) corresponding to the largest eigenvalue (λ_1) represents the direction along which the data variance is maximum.

Step 4: Set a threshold value, τ of the explained variance, and determine a new dimension, d^* by the following formula.

$$d^* = \left\{\min d \mid \frac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{p}\lambda_i} \geq \tau\right\} \tag{4.9}$$

where, $d^*(d^*<p)$ is the reduced dimension.

Step 5: Determine the new matrix, $V^* \in R^{p\times d^*}$ by taking the first d^* columns of *V*.

$$\boldsymbol{V}^* = [\boldsymbol{v_1}\ \boldsymbol{v_2}\ \boldsymbol{v_3} \ldots . \boldsymbol{v_{d^*}}] \tag{4.10}$$

Step 6: Finally, the reduced dataset, $\boldsymbol{X}^{*} \in \boldsymbol{R}^{n \times d^{*}}$ is computed by the formula given below.

$$\boldsymbol{X}^{*} = \widetilde{\boldsymbol{X}}\boldsymbol{V}^{*} \tag{4.11}$$

The dataset X^{*} obtained by PCA is then used to train the NN employed in the tool flank wear prediction.

4.2.1.4. Uncertainty Prediction with MC Dropout

The NN approach is widely used as a prediction approach in different fields, including manufacturing, medicine, supply chain, and many others. Generally, model parameters (weights) are estimated by minimizing the loss function, and the most common loss function is the least square loss function. All models are developed under certain assumptions; for instance, the neurons within a layer do not interact with the other neurons of the same layer in a NN. But, real-world situations do not always align with those assumptions. A great statistician of the 20th century, George Box, mentioned that all models are wrong, but some are very useful [63]. A NN is also considered a very useful model that has been known to predict with high accuracy. In most cases, a prediction obtained by a NN is a deterministic point value, and it is assumed that the level of prediction accuracy is relatively high. This deterministic prediction is highly optimistic, and the model uncertainty is often ignored. Thus, it is essential to know the level of uncertainty in a prediction to make decisions with high confidence. In this research, the MC dropout approach is employed to account for model uncertainty in predictions.

Srivastava et al. [64] proposed a standard dropout technique to prevent overfitting, improve model stability, and efficiently combine different NN architectures. In a NN, a neuron of a layer is fully connected with all neurons of prior and post layers, as shown in Figure 4.3(a). Dropout can be defined as deleting a neuron from networks and all its associated connections

(Figure 4.3(b)). The dropout decision is binary. It is whether a neuron is retained in a NN or is dropped out from a NN. If the dropout probability is *p*, then the probability of a neuron retaining in the network is *1-p*. Usually, dropout is applied to all neurons in a NN layer with the same probability. If the dropout is applied to *q* number of neurons, then the possible number of NN architecture is 2^q. It is recommended to apply the dropout technique only in hidden layers as a dropout in the input layer results in loss of data information for some network architectures.

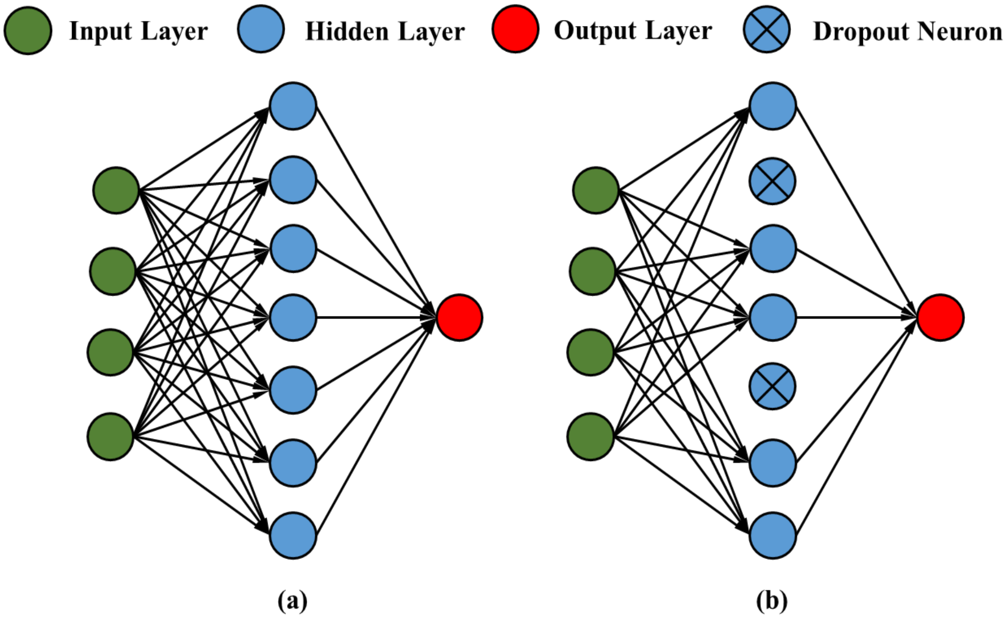


Figure 4.3. (a) Standard NN and (b) NN after applying dropout

Dropout is applied randomly until obtaining a desired level of accuracy. During training, the weights of the retained neurons update in each epoch, and the weights of dropped-out neurons remain unchanged. After training a NN with the dropout, a fully connected network is further applied to predict the target variable for a new observation. And the final weight is estimated as *(1-p) w* for applying a fully connected network. Although the dropout reduces overfitting, the prediction by incorporating the dropout technique is still a point estimation and does not consider uncertainty. Thus, to overcome this challenge, instead of predicting by implementing a fully connected NN, a NN is employed to predict different values for test data (new observation) by applying dropout. This concept is known as MC dropout.

Unlike standard dropout, MC dropout is applied for training and test data. During training a model, MC dropout is the same as the typical dropout. For a new observation, the dropout is applied to a trained NN to predict multiple values to get a distribution of a target variable. When dropout is applied in NN several times, some neurons are masked out. A slightly different network architecture is obtained for every time dropout. For a new observation, different target variable values are predicted from different network architectures. From different values, the distribution of the target variable is estimated. More details, including theoretical proof of MC dropout, can be found in Ref. [65]. Gal and Ghahramani [65] demonstrated that using MC dropout in neural networks is a Bayesian approximation of the Gaussian probability model. It is also shown that the prediction by MC dropout is equivalent to the prediction by the Bayesian neural network, where the distribution of weights is estimated by training. Therefore, the model uncertainty can be estimated by MC dropout [65].

The model uncertainty is incorporated for tool flank wear prediction by applying MC dropout. When a NN model predicts different values of tool flank wear for a new observation, the first two moments (mean and variance) of the tool flank wear can be determined from the predicted values by following formulae.

$$\mu = \frac{1}{D}\sum_{i=1}^{D}\hat{y}_i \tag{4.12}$$

$$\sigma^2 = \frac{1}{D-1}\sum_{i=1}^{D}(\hat{y}_i - \mu)^2 \tag{4.13}$$

where dropout is randomly applied *D* times for a new observation, μ and σ are the mean and standard deviation of different predicted tool flank wear. And $\hat{y}_i$ is the predicted tool wear for i^{th} dropout. When μ and σ are known, the prediction interval (PI) for a new observation for a certain level of significance *(α)* can be estimated as follows.

$$PI = \left[\mu - z_{\alpha\backslash 2}\sigma, \mu + z_{\alpha\backslash 2}\sigma\right] \tag{4.14}$$

where *PI* is the prediction interval and $Z_{\alpha/2}$ is the z-value for α significance level. The range of *PI* represents the level of uncertainty. A narrower range of *PI* indicates a lower level of uncertainty, and a wider range represents a higher level of uncertainty. Based on the level of uncertainty, the maintenance decision will be taken to reduce unwanted failure. If the level of uncertainty is high, then manual monitoring can be performed by observing different process indicators in determining tool changing time. Therefore, interval prediction by considering data and model uncertainty provides more detailed information for decision-making like maintenance scheduling and tool changing. In the following section, the proposed tool flank wear prediction method is implemented on the 2010 PHM Challenge Dataset [66].

4.2.2. Case Study: Tool Wear Prediction

In this section, the proposed method is applied to the 2010 Prognostic and Health Management (PHM) Challenge Dataset [66] to demonstrate the performance and applicability of the proposed method. Further, to verify the results, the performance of the proposed method is compared with other equivalent known methods.

Seven sensors were installed on the Röders Tech RFM760 CNC milling machine to collect tool condition data of a tungsten carbide cutter with three (03) flutes. Feed rate (1555 mm/min), *Y*-axis depth of cut (radial 0.125 mm), and *Z*-axis depth of cut (axial 0.2 mm) were the constant parameters. Among seven sensors, three (03) force sensors collected signals for force along *X*, *Y*, and *Z* axes, and three (03) vibration sensors collected signals for vibrations along *X*, *Y*, and *Z* axes. The other sensor was an acoustic emission (AE) sensor. Tool flank wear was set as the target variable and measured for sensor data. The data of six cutters *(*c_1, c_2, c_3, c_4, c_5, and c_6*)* were collected. The data of 315 observations were collected for each cutter. The tool flank wear was measured for three cutters, c_1, c_4, and c_6, and the measurement unit is 10^{-3} *mm*. Tool wear for all three flutes was measured for the three cutters, and the average tool flank wear was used as the target variable. The measured tool flank wear for the three cutters is shown in Figure 4.4.

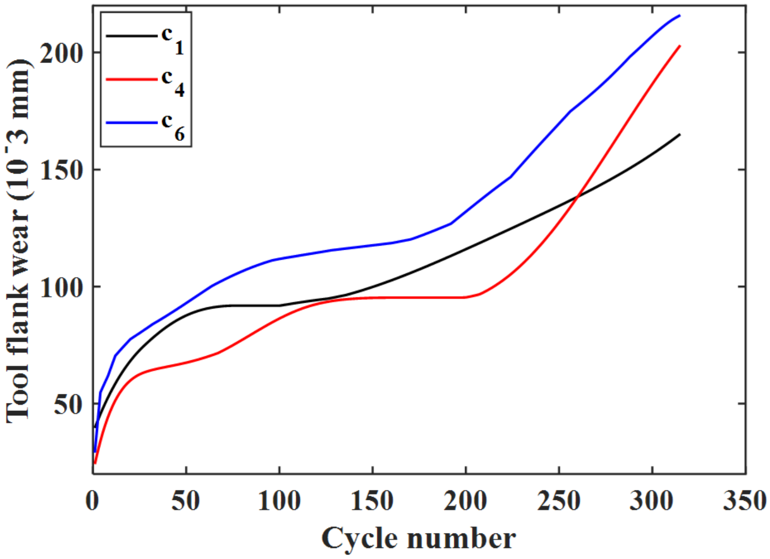


Figure 4.4. The measured tool flank wear for cutters $c_1, c_4, and\ c_6$

Although the machining parameters are the same for all three cutters, the tool flank wear patterns are different for each cutter. Therefore, deterministic assumptions are not useful for this case as capturing the exact pattern of a tool flank wear from a cutter is hard. The sensor data of the cutters with known tool flank wear is used in this dissertation. Among three (03) cutters with known tool flank wear, the data of cutters c_1 and c_4 are used to train the NN with dropout, and the data of the third cutter c_6 is used as test data to predict tool flank wear. Cai et al. [67] and

Zhao et al. [68] predicted tool flank wear of cutter c_1, c_4, and c_6. Both publications get the least prediction accuracy for the tool flank wear prediction of the cutter c_6. For this, the tool flank wear of cutter c_6 is predicted to demonstrate the proposed method. Therefore, the training dataset has 630 observations from cutter c_1 and c_4. The testing dataset consists of 315 observations from the cutter c_6.

In the first step, the data collected from seven (07) sensors are denoised by EMD to reduce data uncertainty. The IMFs are extracted from sensor data by using the EMD algorithm demonstrated in Figure 4.2. The IMFs extraction process is ended if the maximum number (n=10) of IMFs has been extracted or the residual becomes monotonic for a smaller number of IMFs. The threshold value of the correlation coefficient is $\rho=0.8$ for removing IMFs that consist of noise. After reducing data uncertainty by EMD, the 11 time-domain features listed in Table 4.1 are extracted from the denoised data of each sensor. As there are seven (07) sensors, a total of 77 (7×11) features are extracted from all denoised sensor data for each observation. All extracted features may not be relevant to tool flank wear, and some features may linearly depend on each other. PCA is used for feature reduction and generation of independent features. For this dataset, the extracted 77 features are reduced to 25 features that capture 96.7% variation. The new 25 features are applied to train NN. According to eigenvalues of features obtained by PCA, sensor signals for force are most significant as 15 principal components among the 25 components are obtained from force sensor signal-related data.

The reduced dataset obtained after applying PCA is applied to train NN with dropout for interval prediction. For tool flank wear prediction, we considered NN with only one hidden layer to capture non-linearity. The activation function used is the exponential linear unit (ELU), and the number of neurons in the hidden layer is 20. Random dropout rates are applied for tool wear prediction to determine the best dropout rate, and the prediction accuracy is found at its highest for the 0.1 dropout rate. For this research, a 0.1 (p=0.1) dropout rate is chosen. Adam optimizer is used to train NN with a learning rate of 0.001, exponential decay control parameters of 0.99 and 0.999, and smoothing term of 10^{-8} [35]. By applying the dropout technique 50 times (D=50) on the trained NN, for new data, different tool flank wear prediction values can be obtained. The mean (μ) and standard deviation (σ) of all estimated values of tool flank wear are determined from Equations (4.12) and (4.13), respectively. Finally, the confidence interval is calculated from Equation (4.14) with a 95% confidence level.

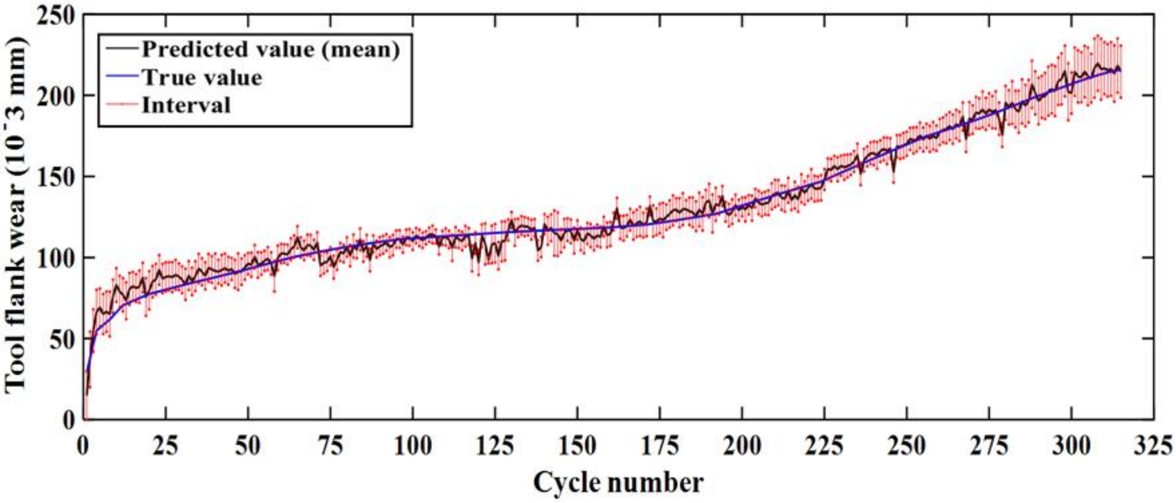


Figure 4.5. Interval prediction result for tool flank wear

The result of the estimated interval tool flank wear with the MC dropout is shown in Figure 4.5. It is visible that the range of the intervals is different for different data points, as expected. Another important observation is that the tool wear prediction accuracy is comparatively low for the first few cycles. It is acceptable as the initial conditions of all cutters are not always the same because manufacturers sometimes use new tools or sharpen the current tools. If the initial conditions of the two cutters are not similar, then there is a high chance that sensor data will be different. In the dataset, there is no information about the cutters' initial condition. Therefore, it is not possible to draw obvious conclusions. But it can still be concluded that with different initial conditions of cutters, the prediction accuracy is lower for the first (approximately) 125th cycles compared to the later cycles (after the 225th cycle in Figure 4.5).

After machining for a certain amount of time, tool conditions are inspected to be the same for most cutters. The chance of tool failure is high for the later cycles, approximately after the 200th cycle in Figure 4.5. Accurate predictions of tool flank wear are essential for these later cycles to estimate an optimum tool changing time and make other maintenance decisions to avoid significant downtime. The proposed model can capture this scenario, which is indicated by the increased prediction accuracy with an increase in the number of cycles.

Another observation is that the prediction intervals are comparatively wider for the last few cycles (after the 295th cycle in Figure 4.5). The reason for this is the range of the tool wear is different for test and training data. The maximum tool flank wear in the training data is 203.08 mm, whereas the maximum tool flank wear on the test data is 215.94 mm. For the test data that are further from the training data, the resulted interval is comparatively wider because the uncertainty is higher for extrapolation [65]. As the distance of the new data is further away from the training data, the interval will also be more expansive, and the degree of uncertainty will also be higher compared to the new data closer to the training data.

Standard regression and classification algorithms do not capture uncertainty for point value prediction. However, this point value prediction is highly optimistic and often applied to support different decision-making. In such cases, it is assumed that a highly reliable model can predict with high confidence, but it is not always accurate [69]. An input may be subjected to noise and outliers, and model uncertainty may arise due to model parameters and structure [70, 71]. Interval prediction is a way to represent these uncertainties in quantitative form. For example, a 95% prediction interval for a new observation can be defined as 95% confident that the true value for the observation will fall within the upper and lower bounds of the interval. The mean value is the most likely predicted value, and the prediction interval determines the variability of prediction around the mean. A wider prediction interval represents the high prediction variability around the mean predicted value. A wider prediction interval is a less reliable prediction. Still, it is useful for analyzing a system as it explains its condition instead of solely relying on optimistic point value estimation as the point value estimation fails to explain variability. A narrower interval is preferable and indicates a low level of uncertainty. The corrective actions would be based on the prediction interval's mean and variance and a part requirement. The mean indicates the estimation of the true value, and an interval represents the degree of belief. To produce a high-quality product, the maintenance department can use the worst (upper bound) value of tool flank wear to take action. The interval prediction is helpful for robust decision-making based on diagnosis results.

4.3. Conservative Interval Prediction by Linear Optimization

Monte Carlo (MC) dropout technique applies only to neural networks with neurons. Other than neural networks, linear and polynomial regression models are widely used for various engineering applications [72]. Due to uncertainty in many applications, including additive manufacturing processes, when data collection is repeated several times for the same combination of input variables, there is a high likelihood that the target variable will differ for each data collection attempt. To cope with different sources of uncertainties, estimation of interval and probability distribution is often more valuable than deterministic point value estimation.

The MC dropout is not applicable for linear and polynomial regression. This section proposes linear optimization-based techniques to predict conservative confidence intervals for linear and polynomial regression models. Two linear optimization models are proposed, one for ordinary least squares (OLS) regression and the other for weighted least squares (WLS) regression. The results show that the proposed method is useful for applications where the level of uncertainty or the lack of knowledge of uncertainty sources is high. The proposed method can also be leveraged to the Bayesian neural network (BNN), where the optimization techniques for interval prediction will be non-linear optimization instead of linear optimization [73].

Regression analysis is a machine learning algorithm that establishes relationships between a target variable and one or more input variables (features) [74]. Usually, the target variable depends on input variables, and the target variable is changed with the change of input variables. The purposes of applying regression analysis are to analyze relationships among variables, develop a fitted model for interpretation, and predict a target variable for new data for which the target variable is unknown for different decision-making. The regression analysis has been applied to many fields such as manufacturing [75], supply chain [76], and medicine [77] to predict the value of a target variable for new observations to aid decision-making processes.

Generally, the regression models are developed by minimizing least-squares errors between fitted values and true values from a dataset. This model of regression is known as the ordinary least squares (OLS) regression. The coefficient of input variables and the intercept are estimated by minimizing the least-squares errors. The point value of the target variable is estimated for a new observation from the known coefficients and intercepts. The OLS regression assumes that the error is consistent for all observations, known as homoscedasticity [78]. Another commonly employed regression besides OLS regression is the weighted least squares

(WLS) regression. The WLS regression is an extension of the OLS technique that weighs the observations proportional to the reciprocal of the error variance. The WLS regression is preferred over OLS regression when the homoscedasticity assumption is violated. In WLS regression, the coefficients of input variables are estimated by minimizing the weighted least squares errors. Like OLS regression, WLS regression is also a point value prediction method. The point value estimation is an optimistic prediction under deterministic assumptions.

Different methods have been applied to predict intervals by regression models to overcome the limitations of point value estimation. The regression coefficients and intersection distribution can be estimated by assuming that the noise is normally distributed with zero means [74]. From this, the distribution of the target variable for a new observation can be estimated. Several methods have been developed to estimate a confidence interval for a prediction. The most common method for confidence interval prediction is based on the variance of coefficients. The interval is predicted under the assumption that the noise is normally distributed. A large data set is required to meet normality assumptions, and the Type I error rate becomes close to the significance level.

The normality assumption can easily be used to estimate the distribution of a target variable. Stine [79] proposed a bootstrapping resampling technique for interval prediction. The bootstrap is a simple and straightforward way to get an inference of the population standard deviation to form a sample by resampling. For bootstrap interval prediction, several datasets are generated from a dataset with the same size as the original dataset. For each set of the generated datasets, a regression model is trained. The mean and variance of the target variable are estimated for new observations from all outcomes of regression models. It is assumed that predictions by the regression models are unbiased estimators of the true value and are normally distributed.

The computation cost of training is high for resampling and training several regression models. Shrestha and Solomatine [80] proposed a fuzzy c-means clustering-based interval prediction method for regression models. One advantage of their method is that it does not require any prior distribution information for interval prediction and predicts asymmetric intervals. Another advantage of the method is that the predicted interval does not depend on the model structure applied to predict a target variable. Olive [81] proposed an asymptotic prediction interval estimation technique from the error percentile of a training dataset for multiple linear regression. The unique feature of the proposed method is that it does not require the normality assumption of error.

The advantages and disadvantages of different interval prediction methods for regression models have been discussed above. To the best knowledge, none of the existing techniques predict conservative intervals at the time of writing. Two new methods are proposed to predict the conservative confidence interval. One method is for OLS regression, and the other method is for WLS regression. The conservative interval represents the possible lower and upper bounds for the best- and worst-case scenarios. In some cases, where the maximum value is better, such as profit, the lower bound of a conservative interval represents the worst-case solution.

As the proposed methods estimate the possible lower and upper bounds, the proposed methods are useful when the level of uncertainty and lack of knowledge about the sources of uncertainty is high. On the other hand, when the lowest value is preferable, such as cost, the worst-case solution is the conservative upper bound. Thus, a conservative interval predicted by the proposed methods will help make robust decisions.

The linear optimization technique is applied to determine the upper and lower bounds of a confidence interval in the proposed methods. The first two moments (mean and variance) of all coefficients and the intercept can be determined from the training dataset. Further, the confidence intervals can be estimated from the first two moments for the desired confidence level. The linear optimization technique is then applied to determine confidence intervals. The regression model obtained by OLS or WLS regression is the objective function, and the confidence intervals of coefficients are the constraints for linear optimization models. The objective function and the constraints are the same for estimating the upper and lower bounds. The difference is that the objective function is maximized for the upper bound and minimized for the lower bound.

4.3.1. Regression

The regression analysis, a widely applied machine learning algorithm, is employed to represent a target variable as a function of input variables. The goal is to predict the target variable based upon the values of input variables from the mathematical function for that the value of the target variable is unknown. Suppose y is a target variable, and k input variables are $x_1, x_2, x_3, \dots, and\ x_k$. The general form of the relationship between the target variable and the input variables is given as

$$y = f(x_1, x_2, x_3, \cdots \cdots x_k) + \epsilon \tag{4.15}$$

where, f is a fixed but unknown function, and ϵ is random noise and independent of input variables. The noise term ϵ is also unknown. The regression analysis estimates f to predict y in the following form.

$$\hat{y} = \hat{f}(x_1, x_2, x_3, \cdots \cdots x_k) \tag{4.16}$$

where, $\hat{f}$ is an estimator of f and $\hat{y}$ is the predicted value of y for input variables. Linear regression is the most widely applied and known method in machine learning to predict a target variable. In linear regression, the target variable is a linear function of input variables. The target variable is represented as the following.

$$Y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots\cdots x_k\beta_k + \epsilon = \boldsymbol{x\beta} + \epsilon \tag{4.17}$$

where, β_0 is intercept, and $\beta_1, \beta_2, \cdots\cdots, \beta_k$ are the coefficients of $x_1, x_2, \cdots\cdots, x_k$, respectively. And, $\boldsymbol{x} = [1 \quad x_1 \quad x_2 \quad \cdots\cdots \quad x_k] \in \boldsymbol{R}^{k+1}$ and $\boldsymbol{\beta} = [\beta_0 \quad \beta_1 \quad \beta_2 \quad \cdots\cdots \quad \beta_k]^T \in \boldsymbol{R}^{k+1}$. Instead of only input variables, different intersection $(e.g.,\ x_{k+1} = x_1x_2, x_{k+2} = 4x_2 + 3x_5)$ and transformations $(e.g.,\ x_{k+1} = x_2^3, x_{k+2} = \ln x_2)$ of input variables can be used as input variables in linear regression. As a result, the curvature can be modeled by linear regression, but the regression model must be a linear function of the parameters, $\boldsymbol{\beta}$.

The least squares approach is widely used to estimate the regression coefficients by minimizing the least squares error [82]. The least squares estimation is a maximum likelihood estimation (MLE) if the noise is normally distributed. Most of the machine learning and deep learning algorithms, including the proposed methods, assume that the noise is normally distributed. With this assumption, the least squares approach is the same as MLE. Based on the dependency of errors on input variables and the consistency of errors, the least squares approach is divided into two broad categories: ordinary least squares (OLS) regression and weighted least squares (WLS) regression. The details of the two categories are given in the following subsections.

4.3.1.1 Ordinary Least Squares (OLS) Regression

OLS technique is a commonly used approach to estimate regression parameters, $\boldsymbol{\beta}$. OLS regression assumes that the variance of residuals is consistent for all observations, and it is known as homoscedasticity. The regression parameters are estimated by minimizing the least squares error. The parameters are estimated as [83]

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-\mathbf{1}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{Y} \tag{4.18}$$

where, $\boldsymbol{X} = \begin{bmatrix} \mathbf{1} & \boldsymbol{x}_{\mathbf{1}} & \boldsymbol{x}_{\mathbf{2}} & \cdots & \boldsymbol{x}_{\boldsymbol{p}} \end{bmatrix} \in \boldsymbol{R}^{n\times(p+1)}, \boldsymbol{Y} \in \boldsymbol{R}^{\boldsymbol{n}}$. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are input matrix and target variable vector, respectively, on the training dataset. And, *p* and *n* are the input dimension and the number of observations in a training dataset, respectively. $\widehat{\boldsymbol{\beta}}$ is an unbiased estimator of $\boldsymbol{\beta}$.

Another assumption of OLS is the noise is normally distributed with zero mean and σ^2 variance. The variance, σ^2, can also be estimated as follows [83].

$$\hat{\sigma}^{\mathbf{2}} = \frac{1}{n-p-1}\big(\boldsymbol{Y}-\widehat{\boldsymbol{Y}}\big)^{\boldsymbol{T}}\big(\boldsymbol{Y}-\widehat{\boldsymbol{Y}}\big) \tag{4.19}$$

where, $\widehat{\boldsymbol{Y}} = \widehat{\boldsymbol{\beta}}\boldsymbol{X} \in \boldsymbol{R}^n$ is the predicted values of the target variable. $\hat{\sigma}^2$ is an unbiased estimator of σ^2 [82]. As the noise is normally distributed, $\boldsymbol{\beta}$ also has a normal distribution with $\widehat{\boldsymbol{\beta}}$ mean and $\hat{\sigma}^2(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-\mathbf{1}}$ covariance. The distribution of $\boldsymbol{\beta}$ can be written as follows [83].

$$\boldsymbol{\beta} \sim N\big(\widehat{\boldsymbol{\beta}}, \hat{\sigma}^2(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-\mathbf{1}}\big) \tag{4.20}$$

In the proposed method, the conservative interval of the target variable is determined by using the distribution of $\boldsymbol{\beta}$. The proposed method for confidence interval prediction for OLS regression is described in the following subsection. A dataset may not meet the homoscedasticity assumption. When the homoscedasticity assumption is violated, the WLS regression is applied to train a model.

4.3.1.2 Weighted Least Squares (WLS) Regression

The homoscedasticity assumption is not always valid for all datasets. When the homoscedasticity assumption is violated, the WLS regression is applied to train a regression model. In WLS regression, the weights are assigned to each observation to reduce the effects of heteroscedasticity. In WLS, the parameters, $\boldsymbol{\beta}$ is estimated by minimizing weighted least squares error, and it can be estimated as

$$\widehat{\boldsymbol{\beta}}_w = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{Y} \tag{4.21}$$

where, $\widehat{\boldsymbol{\beta}}_w$ is WLS estimator of $\boldsymbol{\beta}$. $\boldsymbol{W} = diag(w_1, w_2, w_3, \cdots\cdots w_n) \in \boldsymbol{R}^{n\times n}$ is a diagonal matrix and w_j $(j = 1,2,3, \cdots n)$ is the weight of j^{th} observation. The weights are inversely proportional to the estimated variance of errors that can be estimated from OLS regression.

For WLS regression, the variance of errors and the distribution of $\boldsymbol{\beta}$ are estimated as

$$\hat{\sigma}_w{}^2 = \frac{1}{n-p-1}\left(\boldsymbol{Y} - \widehat{\boldsymbol{Y}}_w\right)^{\boldsymbol{T}}\boldsymbol{W}\left(\boldsymbol{Y} - \widehat{\boldsymbol{Y}}_w\right) \tag{4.22}$$

$$\boldsymbol{\beta} \sim N\left(\widehat{\boldsymbol{\beta}}_w, \hat{\sigma}_w{}^2(\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\right) \tag{4.23}$$

where $\widehat{\boldsymbol{Y}}_w = \widehat{\boldsymbol{\beta}}_w\boldsymbol{X} \in \boldsymbol{R}^n$ is the predicted values of the target variables obtained from the WLS regression model.

In OLS regression, each data point has an equal impact on estimating model parameters. In other words, it is assumed that the variance of residuals is constant over all values of predictors. On the other hand, the impact of each data point is accounted for by weights in WLS regression. In most cases, the weights are unknown but can be estimated from the residuals of OLS regression. The steps for estimating weights by OLS regression are given as follows.

Step 1: Determine absolute values of residuals, $\boldsymbol{\epsilon}^*$ for an OLS regression.

$$\boldsymbol{\epsilon}^* = |\boldsymbol{\epsilon}| = \left|\boldsymbol{Y} - \widehat{\boldsymbol{\beta}}\boldsymbol{X}\right| \tag{4.24}$$

The residual can be defined as the difference between the true value and the predicted value for an observation.

Step 2: Regress $\boldsymbol{\epsilon}^*$ versus $\widehat{\boldsymbol{\beta}}\boldsymbol{X}$ by considering $\boldsymbol{\epsilon}^*$ as a dependent variable. The fitted values of $\boldsymbol{\epsilon}^*$ is $\widehat{\boldsymbol{\epsilon}}^*$ and estimated as follows.

$$\widehat{\boldsymbol{\epsilon}}^* = [\hat{\epsilon}_1^* \quad \hat{\epsilon}_2^* \quad \hat{\epsilon}_3^* \quad \cdots \quad \hat{\epsilon}_n^*] \tag{4.25}$$

Step 3: Estimate weights, $\boldsymbol{W}$ from the fitted values, $\hat{\boldsymbol{\epsilon}}^*$.

$$\boldsymbol{W} = diag\left(\frac{1}{\hat{\epsilon}_1^{*2}}, \frac{1}{\hat{\epsilon}_2^{*2}}, \frac{1}{\hat{\epsilon}_3^{*2}}, \cdots \frac{1}{\hat{\epsilon}_n^{*2}}\right) \tag{4.26}$$

In the proposed method, the distribution of $\boldsymbol{\beta}_w$ is used to determine the conservative interval of the target variable. The proposed method for confidence interval prediction for WLS regression is described in the following subsection.

4.3.2. Conservative Interval Prediction Methods

The OLS and WLS regressions are selected based on the homoscedasticity of error. The WLS regression is applied over OLS regression when the homoscedasticity assumption is violated. The distribution of parameters $\boldsymbol{\beta}$ is changed based on the selection of regression models. The proposed methods are developed under two assumptions. Those are:

- Noise, ϵ is normally distributed with a mean zero and σ^2 standard deviation. This is a common assumption for most machine learning and deep learning algorithms.
- The second assumption is that all $\beta_i (i = 0,1,2, \cdots p)$'s are independent. In other words, the covariance between β_i's is zero.

$$cov(\beta_i, \beta_j) = 0 \quad \forall\, i, j = 0,1,2, \cdots p, i \neq j \tag{4.27}$$

An assumption of developing a linear regression model is that the input variables are independent. If the assumption is true, and a transformed dataset is used instead of a collected dataset, then the covariances between $\beta_i's$ are also zero. The new form of a dataset is

$$\widetilde{\boldsymbol{X}} = \boldsymbol{X}^* - \overline{\boldsymbol{X}}^* \tag{4.28}$$

where, $\boldsymbol{X}^* \in \boldsymbol{R}^{n \times p}$ is the collected data, $\overline{\boldsymbol{X}}^*$ is the mean and $\widetilde{\boldsymbol{X}}$ is the new transformed dataset. Now, $\boldsymbol{X}$ can be written as

$$\boldsymbol{X} = [\boldsymbol{1} \quad \widetilde{\boldsymbol{X}}] \tag{4.29}$$

Data normalization or transformation is common for most machine learning and deep learning algorithms. It is also recommended to use various feature reduction techniques to reduce irrelevant and redundant features before applying a machine learning algorithm. Besides, principal component analysis (PCA) can also be used to generate linearly independent features and reduce the dimension of a dataset. Therefore, the assumptions of the proposed methods are the same as other machine learning algorithms, and these assumptions are not unique and new. The proposed methods can predict conservative intervals and aid in robust decision-making with the same assumptions. The proposed method is beneficial when there is a high lack of knowledge about uncertainty sources. The proposed methods for interval prediction by linear optimization are demonstrated in this section.

4.3.2.1 OLS Interval Prediction

In OLS regression, $\boldsymbol{\beta}$ is estimated as $\widehat{\boldsymbol{\beta}}$ by minimizing the least squares error, and the formula for estimating $\boldsymbol{\beta}$ is given in Equation (4.18). Then, the distribution of $\boldsymbol{\beta}$ can be estimated from Equation (4.20). An assumption of the proposed method is that β_i's are independent. Therefore, the covariance matrix of $\boldsymbol{\beta}$ becomes a diagonal matrix and can be written as

$$\widehat{\boldsymbol{\sigma}}_{\beta}^{\mathbf{2}} = \hat{\sigma}^2(\boldsymbol{X}^T\boldsymbol{X})^{-1} \approx diag\left(\hat{\sigma}_{\beta_0}^2, \hat{\sigma}_{\beta_1}^2, \cdots, \hat{\sigma}_{\beta_p}^2\right) \tag{4.30}$$

When the mean and variance of all β_i's are known, the conservative confidence interval can be estimated from the linear optimization models for estimating lower and upper bounds are given in Equations (4.31) and (4.32), respectively.

$$y_L^* = \min_{\boldsymbol{\beta}} \boldsymbol{x\beta}$$

$$\text{s. t.} \quad \hat{\beta}_i - k_1\hat{\sigma}_{\beta_i} \leq \beta_i \leq \hat{\beta}_i + k_1\hat{\sigma}_{\beta_i} \quad \forall i = 0,1,\ldots,p \tag{4.31}$$

$$y_U^* = \max_{\boldsymbol{\beta}} \boldsymbol{x\beta}$$

$$\text{s. t.}\ \ \hat{\beta}_i - k_1\hat{\sigma}_{\beta_i} \le \beta_i \le \hat{\beta}_i + k_1\hat{\sigma}_{\beta_i} \quad \forall i = 0,1,\dots,p \tag{4.32}$$

In Equations (4.31) and (4.32), the constraints are the confidence intervals of β_i. $k_1$is a constant, and it depends on the level of conservatism. k_1 can be defined as the critical t value from T-distribution at $(1-\alpha/2)$ level of confidence and $(n-p-1)$ the degree of freedom, $k_1 = t_{1-\alpha/2}^{n-p-1}$. $\boldsymbol{x} = [1 \quad x_1 \quad x_2 \quad \cdots \quad x_p]$ is a new observation for which the confidence interval of the target variable is estimated. In the above optimization formulations, the constraints and objective functions are both linear functions of $\boldsymbol{\beta}$**,** and $\boldsymbol{\beta}$ is a bounded variable. Therefore, the optimization models are convex, which guarantees that the proposed optimization models can estimate the minimum and maximum values of the objective functions. The proposed interval prediction models in Equations (4.31) and (4.32) are simple linear optimization models with bounded constraints. Therefore, the exact solution of the optimization formulation in Equations (4.31) and (4.32) can be obtained analytically. For instance, in estimating the lower bound of *y* in Equation (4.31), the objective function is minimum at the lower bound of a variable with the positive coefficient and the upper bound of a variable with the negative coefficient. Similarly, the upper bound of y can be estimated analytically. The proposed optimization approach is not computationally expensive. However, in many cases, additional constraints may require based on the target variables or others. For instance, the production time is always a positive value. Thus, when the production time is the objective function for optimization, another constraint ($\boldsymbol{x\beta \geq 0}$) needs to consider in Equation (4.31) so that the solution is never a negative value. The analytical solution approach may not work well in such cases.

The current most-employed procedure (in this work, it is defined as the traditional method) to predict a confidence interval of a target variable y is with

$$CI = [y_L, y_U] = \left[E(y) - k_1\hat{\sigma}_y, E(y) + k_1\hat{\sigma}_y\right] \tag{4.33}$$

The mean and variance of a function are estimated by using a first-order Taylor series approximation as follows [42].

- Performance function:

$$Z = f(z_1, z_2, z_3, \cdots z_p) \tag{4.35}$$

- First − order approximation of mean:

$$E(Z) = f\big(E(z_1), E(z_2), \cdots, E(z_p)\big) \tag{4.36}$$

- First − order approximation of variance:

$$\hat{\sigma}_Z^2 = \sum_{i=1}^{p}\left(\frac{\partial f}{\partial z_i}\right)^2 var(z_i) + 2\sum_{i=1}^{p}\sum_{j=i+1}^{p}\frac{\partial f}{\partial z_i}\frac{\partial f}{\partial z_j}Cov(z_i, z_j) \tag{4.37}$$

Therefore, the mean and variance of y can be estimated from the first-order Taylor series approximation, and the approximate mean and standard deviation of y are given below.

$$\begin{aligned} E(y) &= \mu_{\beta_0} + x_1\mu_{\beta_1} + x_2\mu_{\beta_2} + \cdots + x_p\mu_{\beta_p} \\ &= \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots. + \hat{\beta}_p x_p \end{aligned} \tag{4.37}$$

$$\hat{\sigma}_y^2 = \hat{\sigma}_{\beta_0}^2 + x_1^2\hat{\sigma}_{\beta_1}^2 + x_2^2\hat{\sigma}_{\beta_2}^2 + \cdots + x_p^2\hat{\sigma}_{\beta_p}^2 \tag{4.38}$$

For the variance, all covariance terms are removed as we are assuming that β_i's are independent. Therefore, the lower bound of the confidence interval of y in Equation (4.33) can be demonstrated as

$$\begin{aligned} y_L &= E(y) - k_1\hat{\sigma}_y \\ &= \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots. + \hat{\beta}_p x_p - k_1\sqrt{\hat{\sigma}_{\beta_0}^2 + x_1^2\hat{\sigma}_{\beta_1}^2 + x_2^2\hat{\sigma}_{\beta_2}^2 + \cdots + x_p^2\hat{\sigma}_{\beta_p}^2} \\ &\geq \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots. + \hat{\beta}_p x_p - k_1\left(\hat{\sigma}_{\beta_0} + x_1\hat{\sigma}_{\beta_1} + x_2\hat{\sigma}_{\beta_2} + \cdots + \hat{x}_p\sigma_{\beta_p}\right) \\ &= \left(\hat{\beta}_0 - k_1\hat{\sigma}_{\beta_0}\right) + \left(\hat{\beta}_1 - k_1\hat{\sigma}_{\beta_1}\right)x_1 + \cdots + \left(\hat{\beta}_p - k_1\hat{\sigma}_{\beta_p}\right)x_p \end{aligned} \tag{4.39}$$

The intercept and coefficients of $x_1, x_2, \cdots x_p$ in Equation (4.39) are within the bounds of $\boldsymbol{\beta}$ in Equation (4.31). And the optimization problem introduced in Equation (4.31) is a convex optimization problem. Therefore, $y_L^* \leq y_L$. Now, the upper bound of the confidence interval of y can be written as

$$
\begin{aligned}
y_U &= E(y) + k_1 \hat{\sigma}_y \\
&= \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots . + \hat{\beta}_p x_p + k_1 \sqrt{\hat{\sigma}_{\beta_0}^2 + x_1^2 \hat{\sigma}_{\beta_1}^2 + x_2^2 \hat{\sigma}_{\beta_2}^2 + \cdots + x_p^2 \hat{\sigma}_{\beta_p}^2} \\
&\leq \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots . + \hat{\beta}_p x_p + k_1 \left(\hat{\sigma}_{\beta_0} + x_1 \hat{\sigma}_{\beta_1} + x_2 \hat{\sigma}_{\beta_2} + \cdots + \hat{x}_p \sigma_{\beta_p} \right) \\
&= \left(\hat{\beta}_0 + k_1 \hat{\sigma}_{\beta_0} \right) + \left(\hat{\beta}_1 + k_1 \hat{\sigma}_{\beta_1} \right) x_1 + \cdots + \left(\hat{\beta}_p + k_1 \hat{\sigma}_{\beta_p} \right) x_p
\end{aligned}
\tag{4.40}
$$

Similarly, the intercept and coefficients of $x_1, x_2, \cdots x_p$ are within the bounds of $\boldsymbol{\beta}$ in Equation (4.32). The optimization model, Equation (4.32), for estimating the upper bound of y is a convex optimization. Therefore, it can be concluded that $y_U^* \geq y_U$. Thus, the proposed optimization models in Equations (4.31) and (4.32) guarantee that the estimated lower and upper bounds are, respectively, lower and greater than the bounds estimated from Equation (4.33). For this reason, the proposed method ensures that the predicted interval is conservative.

4.3.2.2. WLS Interval Prediction

Similar to OLS regression, for WLS regression, the $\boldsymbol{\beta}$ is estimated as $\widehat{\boldsymbol{\beta}}_w$ by minimizing the weighted least squares error. It is assumed that β_{wi}'s are independent. Therefore, the covariance matrix of $\boldsymbol{\beta}$ becomes a diagonal matrix and can be written as

$$
\widehat{\boldsymbol{\sigma}}_{\beta_w}^{\mathbf{2}} = \hat{\sigma}_w{}^2 (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \approx diag\left(\hat{\sigma}_{\beta_{w0}}^2, \hat{\sigma}_{\beta_{w1}}^2, \cdots, \hat{\sigma}_{\beta_{wp}}^2 \right) \tag{4.41}
$$

From estimated mean $\widehat{\boldsymbol{\beta}}_w$ and variance $\widehat{\boldsymbol{\sigma}}_{\beta_w}^{\mathbf{2}}$ of $\boldsymbol{\beta}$**,** the conservative interval of a target variable for WLS regression can be estimated. The mathematical models for estimating $\boldsymbol{W}$ and $\boldsymbol{\beta}$

are given in Equations (4.26) and (4.21), respectively. The linear optimization models for estimating the lower and upper bounds of conservative intervals are given below.

$$y_{wL}^{*} = \min_{\boldsymbol{\beta}} \boldsymbol{x\beta}$$

$$\text{s. t.} \quad \hat{\beta}_{wi} - k_1\hat{\sigma}_{\beta w_i} \le \beta_i \le \hat{\beta}_{wi} + k_1\hat{\sigma}_{\beta w_i} \quad \forall i = 0,1,\dots,p \tag{4.42}$$

$$y_{wU}^{*} = \max_{\boldsymbol{\beta}} \boldsymbol{x\beta}$$

$$\text{s. t.,} \quad \hat{\beta}_{wi} - k_1\hat{\sigma}_{\beta_{wi}} \le \beta_i \le \hat{\beta}_{wi} + k_1\hat{\sigma}_{\beta_{wi}} \quad \forall i = 0,1,\dots,p \tag{4.43}$$

In Equations (4.42) and (4.43), $\hat{\beta}_{wi}$ is the *(i+1)* element of $\widehat{\boldsymbol{\beta}}_w$. Similar to Equations (4.39) and (4.40), it can be proved that the proposed method for WLS regression can also estimate conservative confidence intervals. Similar to optimization formulations in Equations (4.31) and (4.32), there may also have other constraints in Equations (4.42) and (4.43). In the following section, the proposed methods are implemented on three datasets to demonstrate the performance of the proposed methods.

4.3.3. Case Study: Experimental FFF Dataset

The proposed method is applied to the collected experimental FFF dataset. The data collection process, the initial study on the uncertainty in the FFF process, and the interval prediction results using the proposed method on the experimental FFF dataset are described in this section.

The data collection procedure is described in Chapter 2. This chapter uses the original levels of process parameters instead of the coded levels. In addition to the dataset given in Chapter 2, ten (10) other combinations of process parameters are selected randomly for data collection to use a portion of collected data as test data for validation. Therefore, the data is collected for a total of 40 combinations of process parameters. Three parts are printed for each combination of process parameters. Therefore, a total of 120 parts are produced. The compressive strength of the produced parts is obtained by testing the printed test specimen using the INSTRON compressive strength equipment with a load of 30kN, moving at a uniform speed of 1.3 mm/min. The maximum compressive strength before 10% compression along the compressive load direction is used as the target variable for the conservative interval prediction. The collected dataset has 120 observations, four input variables, and one target variable, compressive strength measured in the MPa unit.

Table 4.2. Partial experimental data collection outcome

Process Parameter	Combination 1	Combination 2	Combination 3
Layer Thickness, mm	0.22	0.34	0.28
Build Orientation, degree	45	90	0
Infill Density, %	20	50	65
Extrusion Temperature, ℃	215	215	200
Compressive Strength (MPa)			
Part 1	9.964	23.033	27.62
Part 2	10.845	28.500	28.365
Part 3	13.139	29.007	32.132

The objective of the experimental study is to analyze the effect of different process parameter combinations, namely, layer thickness, build orientation, infill density, and extrusion temperature with respect to compressive strength. Three identical parts of the FFF process are produced for one combination of process parameters. The compressive strength is measured as a target variable for each combination. There are 40 combinations of process parameters studied. Table 4.2 shows the numerical compressive strength results of three process parameter combinations.

The variations of compressive strength of three identical parts for three (03) combinations in Table 4.2 are also portrayed as a bar graph in Figure 4.6. It is visible from Figure 4.6 that there is a significant variation in part properties for the same combination of process parameters. This indicates that the impact of uncertainty is evident in the FFF part properties.

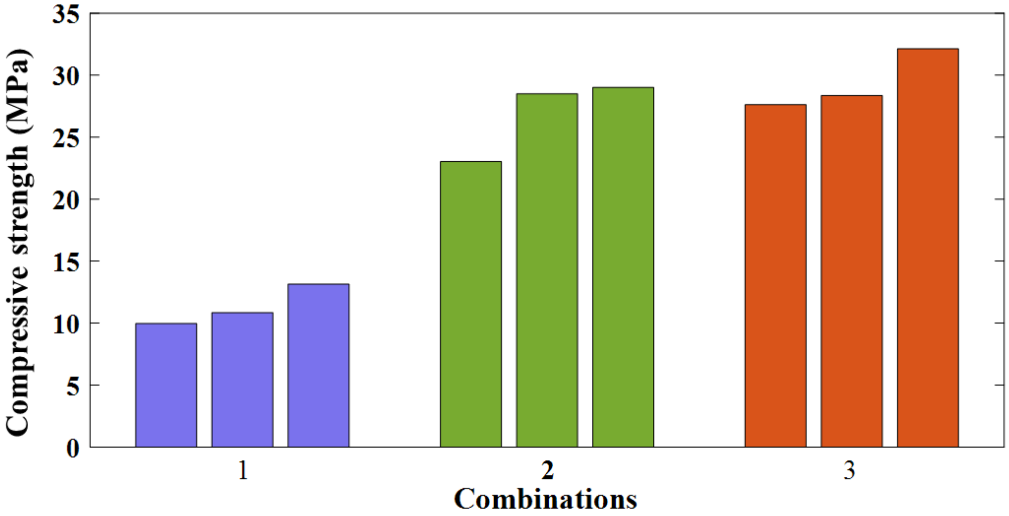


Figure 4.6. Compressive strength variation for different combinations of process parameters

Thus, the point value estimation for a combination of process parameters may not always be sufficient to predict the variation in the FFF process. To overcome this challenge, interval prediction or probability distribution estimation is more beneficial to capture variability in the FFF process. The proposed method can be employed to obtain the conservative interval prediction, and the results are detailed below.

Generally, the relationships between FFF process parameters and part properties are quadratic polynomial relationships [84, 85]. In this research, polynomial regression is applied for interval compressive strength prediction. From the collected data for 40 combinations, 30 combinations with 90 observations are randomly selected as the training data. The remaining 30 observations from the leftover 10 combinations are used as the test data.

To check the homoscedasticity assumption, the residual plot is used, and the residual plot for the training data is shown in Figure 4.7. It is visible that residuals are random and do not take any definite shape or pattern. In other words, if a fitted line is drawn between residuals and fitted values, the fitted line will be horizontal. Therefore, the training dataset is deemed to meet the homoscedasticity assumption. For this, OLS regression is applied to the FFF dataset.

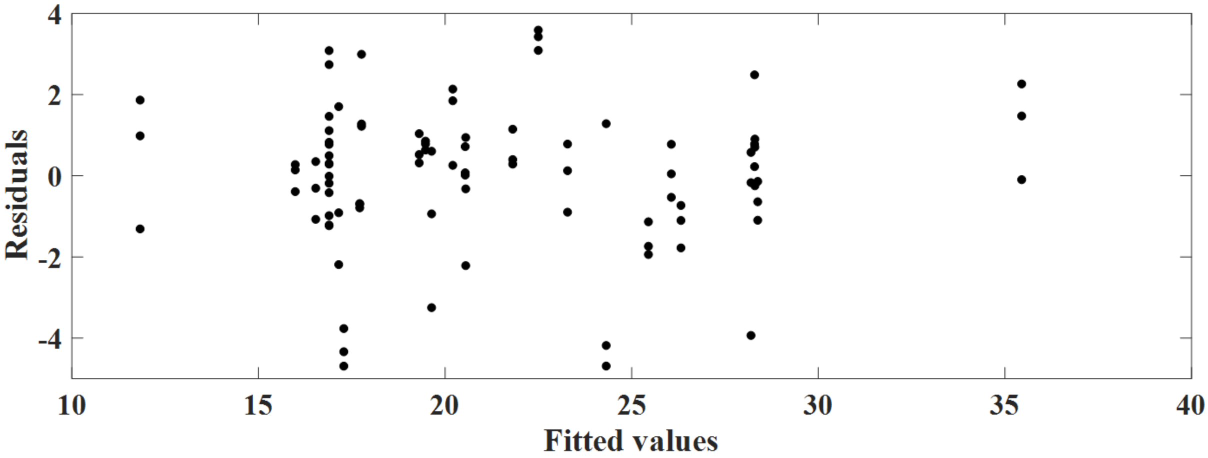


Figure 4.7. Residual plot for training data

The intercept and the coefficients of all first-order terms, second-order terms, and their interaction are estimated by OLS regression. The quadratic regression model for the FFF dataset is given in Equation (4.41).

$$y = \beta_0 + \sum_{i=1}^{4} \beta_i x_i + \sum_{i=1}^{4} \beta_{ii} x_i^2 + \sum_{i=1}^{4} \sum_{j=i+1}^{4} \beta_{ij} x_i x_j = \boldsymbol{x\beta} \qquad (4.44)$$

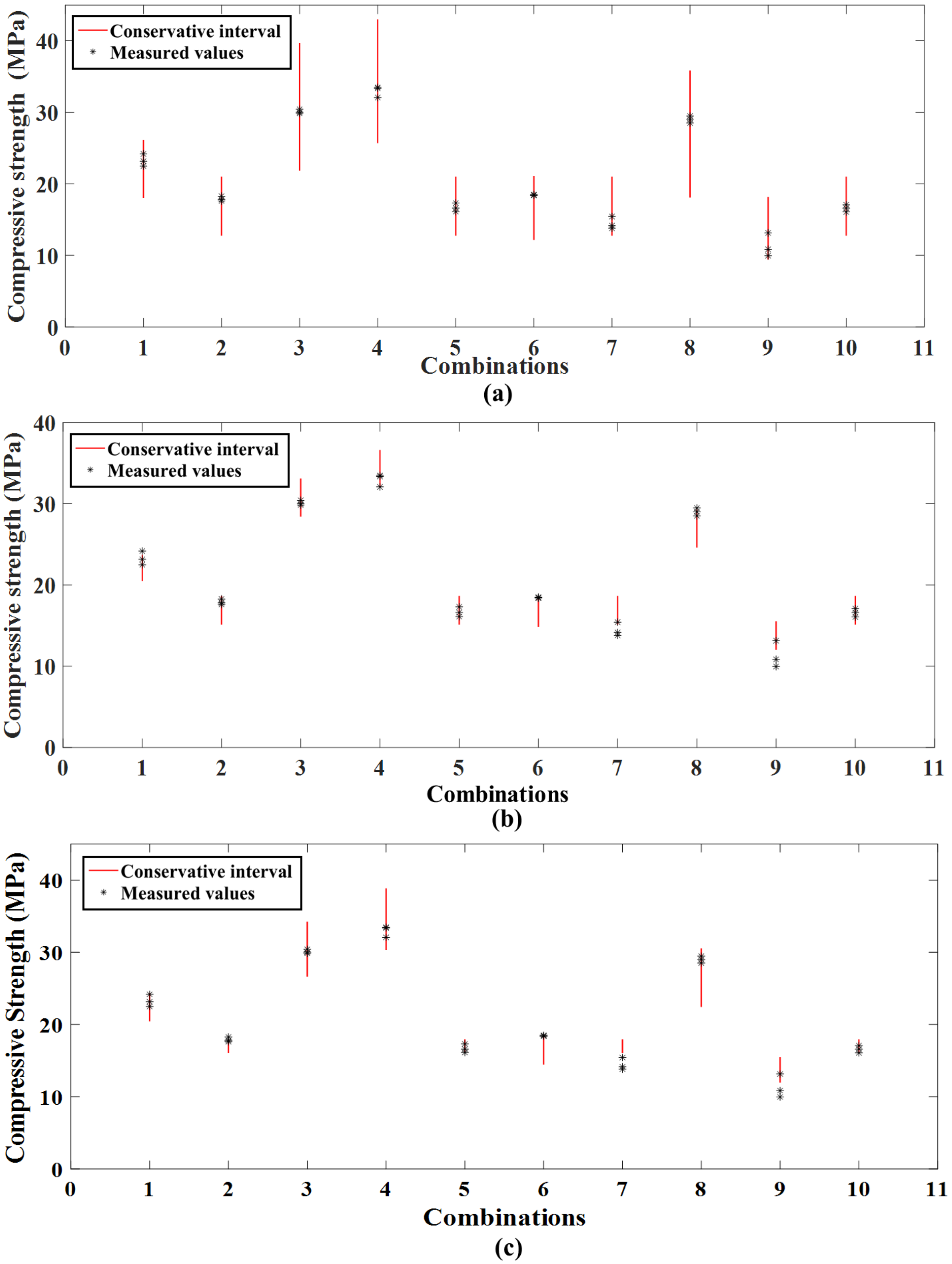
where $\boldsymbol{\beta} = \beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_{11}, \beta_{22}, \beta_{33}, \beta_{44}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{23}, \beta_{24}, \beta_{34}]^T$. β_0 is an intercept, and β_i β_{ii}, β_{ij} are the coefficients of linear, quadratic and interaction terms, respectively. The input variables and the target variable are normalized for predicting regression parameters. The predicted intercept and coefficients are the mean values of $\boldsymbol{\beta}$. Then, the residuals' variance is estimated using the formula given in Equation (4.19). The covariance of $\boldsymbol{\beta}$ is estimated from Equation (4.30) by using the estimated variance of residuals. The estimated mean and covariance of $\boldsymbol{\beta}$ for the FFF dataset are given in Equation (4.45).

$$\boldsymbol{\beta} = \left(\begin{bmatrix} 0.018 \\ 0.267 \\ 0.121 \\ 0.508 \\ 0.063 \\ 0.098 \\ 0.510 \\ 0.035 \\ -0.063 \\ -0.424 \\ 0.349 \\ 0.049 \\ -0.082 \\ 0.050 \\ -0.008 \end{bmatrix}, diag \begin{pmatrix} 0.0015 \\ 0.0016 \\ 0.0019 \\ 0.0019 \\ 0.0020 \\ 0.0026 \\ 0.0031 \\ 0.0028 \\ 0.0031 \\ 0.0020 \\ 0.0018 \\ 0.0020 \\ 0.0021 \\ 0.0018 \\ 0.0020 \end{pmatrix}^T \right) \tag{4.45}$$

Infill density (x_3), the square of build orientation (x_2^2), the interaction between layer thickness and build orientation (x_1x_2), and interaction between layer thickness and infill density (x_1x_3) are four significant terms with a high mean value of coefficients. From an analysis of the distribution of model parameters, it can be concluded that extrusion temperature is the least significant process parameter for compressive strength. On the other hand, three other process parameters and their interaction in some forms are significant for compressive strength.

From the estimated distribution of $\boldsymbol{\beta}$, the conservative intervals for all 10 combinations of test data are estimated by using the linear optimization models given in Equations (4.31) and (4.32). For estimating compressive strength by Equations (4.31) and (4.32), an additional constraint $(\boldsymbol{x\beta} \geq 0)$ is considered as it is commonly known as compressive strength should always be a positive value. The predicted intervals for test data are shown in Figure 4.8(a).

As an experimental validation effort, three parts are printed and further tested for each combination to obtain the compressive strength. The measured compressive strength is also shown in Figure 4.8(a).



(a)

(b)

(c)

Figure 4.8. Interval prediction for FFF dataset (a) proposed method (b) traditional method and (c) bootstrap approach

The confidence intervals are predicted for a 95% confidence level in this case. As shown in Figure 4.8(a), although the ranges of measured compressive strength seem large for some combinations of process parameters such as 3rd, 4th, and 8th, the measured compressive strength for all parts is guaranteed within the conservative interval bounds. It is indicated that the proposed method can capture uncertainty and predict the conservative interval for the compressive strength of FFF build parts.

As a comparison, the intervals are also predicted by the traditional interval prediction method and bootstrap approach. The predicted interval and the measured compressive strength obtained from the traditional interval prediction method and bootstrap approach are shown in Figures 4.8(b) and 4.8(c), respectively. Figure 4.8(b) shows that not all measured values are within the predicted intervals for all combinations, such as the 1st, 4th, 6th,7th, 8th, and 9th combinations. Figure 4.8(c) for the bootstrap approach shows that all measured values are not within the predicted intervals for some combinations, such as the 2nd, 5th, 6th, 7th, 9th, and 10th combinations. Therefore, the proposed conservative interval prediction methods are more suitable than the traditional interval and bootstrap approaches when the uncertainty is high.

In addition, it is known that one of the limitations of the FFF process is inconsistent mechanical properties. This phenomenon is reflected in the collected data, as well. There is a significant deviation in part properties for two parts produced by the same values of process parameters. Due to sources of uncertainty, the point value estimation is not always suitable to predict part properties for the FFF process. When the uncertainty or inconsistency is high, the proposed interval prediction method can predict a conservative interval that guarantees that all the values are within the desired confidence level. Although the case study presented may not seemingly do justice to the importance of conservative intervals, FFF printed parts are slowly being incorporated into systems that require exceptionally high reliability, such as in military or space applications. If the quality of the FFF printed part is not fully guaranteed, this can initiate a failure and jeopardize the entire system performance.

4.4. Summary

In real-world environments where the uncertainty is high due to different sources of uncertainty, it is encouraged to employ interval prediction methods. The MC dropout-based interval prediction framework is proposed for tool flank wear prediction under data uncertainty and model uncertainty. The data and model uncertainty are incorporated into two steps. The data uncertainty is reduced in the first step by using the EMD approach. In the second step, for a new observation (sensor data), the model uncertainty is incorporated in tool flank wear prediction as a prediction interval.

Additionally, linear optimization-based conservative interval prediction techniques are proposed. As the proposed methods are used to estimate conservative confidence intervals, the decision based on the interval predicted by the proposed methods will be more robust. The conservative interval plays an important role when a wrong prediction is costly or when there is a high level of uncertainty in data or systems from which the data is collected. The proposed method can estimate the best case and worst case values by the interval bounds. Sometimes, when the goal is to obtain a minimum value of a target variable (e.g., cost), the lower bound is the best case, and the upper bound is the worst case. In other cases, when the goal is to obtain a maximum value of a target variable (e.g., profit), the upper and lower bounds are the best case and worst case, respectively.

CHAPTER 5. PHYSICS INFORMED MACHINE LEARNING

5.1. Overview

Different advanced data-driven methods such as machine learning and deep learning algorithms have massive potential for applications in manufacturing. Now, applications of data-driven approaches in manufacturing are increasing to make steps towards Industry 4.0. In Chapter 2, ANN and RSM are applied to determine combinations of the process parameters for multiple responses. In Chapter 3, the RRSO approach is proposed to incorporate uncertainty quantification with data-driven methods to make a robust decision and estimate the distribution of a response (target variable). Besides, a BN-based approach has been proposed to analyze numerical and categorical variables together with information from multiple sources. MC dropout-based approach to estimate a target variable as the interval is proposed in Chapter 4. Another interval prediction approach is proposed in Chapter 4 to estimate conservative intervals from model parameters distribution using a linear optimization approach. Incorporating uncertainty in prediction by a data-driven approach increases the robustness of prediction.

Another major shortcoming of data-driven methods is that the physical principles of a system from which data is collected do not account for developing models to predict a target variable. It is essential to incorporate physical knowledge in data-driven methods to make more robust and physically consistent decisions based on data-driven methods' prediction. The limited availability of high-quality data is another obstacle to applying a data-driven method in manufacturing. A model can be pre-trained using physics-based knowledge to train an accurate model from a small collected dataset. Physics-informed machine learning (PIML) is a viable option for this. Incorporating physics-based knowledge in a data-driven method depends on the physics of a system and the physics-based constraints. This chapter proposes a PIML approach to estimate the surface roughness of an FFF build part. Physics-based knowledge has been incorporated using a physics-based empirical model output as an input variable to train a data-driven model. Besides, a customized loss function has been developed based on physics-based constraints to regularize the model and generate physically consistent outcomes. In this research, a radial basis function neural network (RBFNN) is trained by incorporating physics-based knowledge. In addition, the performance of the proposed PIML model is compared with other models using mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and R^2 value.

5.2. Physics Informed Machine Learning

Nowadays, data-driven methods are applied to gain insights, monitor system performance, and support decision-making processes. Using data-driven algorithms to estimate a target variable from experimental and sensor data is getting significant attention from researchers to make steps toward Industry 4.0. An unforeseen amount of information is available in the manufacturing sector due to data availability, data collecting devices, and advanced equipment for experiments [86]. The flexibility and power of data-driven methods increase the usage of the methods in different fields, including manufacturing [87]. Besides, an exponential increase in computation power also accelerates the applications of data-driven algorithms [88].

A limitation of the data-driven algorithms is that these algorithms merely rely on collected data for prediction, not on the physics of a system from which the data is collected [12]. The ignorance of the physical principles of a system may result in physical inconsistency in prediction by data-driven algorithms. It is difficult to interpret the prediction of a data-driven method. Besides, a large dataset is required to apply data-driven numerical modeling to make decisions [13]. In many cases, collecting a large dataset is time-consuming and costly.

In addition, prediction accuracy mainly depends on the quality and forms of data. Data can be available in different formats such as images, audio, videos, and numerical. Processing large datasets for training data-driven numerical models can be expensive in terms of time spent and memory used. Therefore, machine learning and deep learning algorithms are not always necessarily applicable and represent the physics of a system. PIML models are preferable to traditional data-driven numerical algorithms to overcome the above limitations of traditional data-driven methods.

In PIML models, the physical phenomena of a system are embedded with data-driven algorithms. The physics-based model used the physical phenomena of systems to develop a mathematical model for predicting a target variable [86]. A physics-based model can predict a target variable accurately if the physical phenomena of a system are modeled precisely. Therefore, an in-depth understanding of the physical behavior of a system is essential [87]. It is not always possible to accurately develop physics-based models for a system as many sources of uncertainty, and uncontrollable parameters exist. Generally, physics-based models are developed under many assumptions, such as independence between system parameters and deterministic assumptions. It is easier to interpret physics-based models compared to data-driven models. A mathematical model is developed from historical sensor data and experimental data to predict a target variable by data-driven models. A goal of developing PIML models is to get advantages of physics-based and data-driven models for predicting a target variable.

PIML models are physically consistent and scientifically sound predictive models. The prediction accuracy also can be achieved by training a model with fewer data points. Another advantage is that PIML enhances the interpretability of a model. There are many ways to incorporate physics-based knowledge in developing a PIML model. Physics-based knowledge can be incorporated into the loss function. Physics-based constraints can be used as regularization terms in the loss function. As a result, the loss function has additional terms along with the traditional loss function [88]. Another approach to embedding physics-based knowledge in a data-driven method is to use the physics-based model's output as an additional input with other input variables to training a model [89]. Data-driven models are required a lot of label data to develop an accurate model. It is often not always feasible to collect much data due to cost and time. The computational expense increases exponentially when a large dataset is used to train a model. A simulated dataset can be generated based on a physics-based model to pre-train a data-driven model. Then, a collected small dataset can be used to fine-tune a pre-trained model for obtaining an accurate model from a small dataset [90]. Another approach to incorporating physics-based features with other input variables is to train a data-driven model [91].

5.3. Physics Informed Machine Learning in Additive Manufacturing Applications

In AM, machine and deep learning algorithms are increasingly used to optimize process parameters, examine powder spreading, and in-process defect monitoring [92]. The use of PIML in additive manufacturing processes is at the beginning stage. As AM experiments are time-consuming and costly, generating a large dataset for training a machine learning model may not be feasible [93].

Limited research has been carried out on applying PIML concepts in AM processes. Liu et al. [94] developed a PIML model to predict pore generation for the laser powder bed fusion metal AM process. A unique feature of the proposed model is that the model is a generalized model for any laser bed fusion-based AM machine. Besides, a model, a combination of data-driven and physics-based models, was developed to predict melt pool width for the laser powder bed fusion process and compared with a computational fluid dynamics (CFD) model [95]. It was shown that the proposed approach improves prediction accuracy without compromising computational expense. Du et al. [96] combined machine learning and mechanistic modeling to predict defects before experimentations for the same AM process. The authors identified important variables for defect formation and the physics behind each variable to identify and reduce defects.

Zhu et al. [93] developed two PIML models for the metal AM process to accurately estimate the temperature and melt pool fluid dynamics from a small dataset. Kapusuzoglu and Mahadevan [89] used three strategies to develop PIML approaches for bond quality and porosity of FFF build parts. A total of eight deep learning models are trained for different combinations of three methods for both target variables. A total of eight deep learning models are trained for different combinations of three strategies for both target variables. Wang et al. [97] proposed a PIML approach for uncertainty quantification for the metal AM process. Kats et al. [98] trained a PIML model to predict grain structure characteristics for the direct energy deposition process and physics knowledge considered in selecting features. Mondal et al. [91] used thermophysical and mechanical features along with process parameters to predict creak formation in the metal AM process. Ren et al. [99] also developed a PIML model for a metal AM process to predict thermal behavior.

Most of the PIML approaches are developed for metal-based additive manufacturing. There are a few research works on PIML models for the FFF process. Data availability, physical-based modeling limitations, data imbalance, and data preprocessing are several common problems in applying PIML in additive manufacturing [100]. This chapter proposes a PIML approach to predict the surface roughness of an FFF build part. The proposed approach uses the physics-based model output as an additional input to train a radial basis function neural network

(RBFNN). Besides, a physics-based constraint is used as another term with the MSE term in developing a customized loss function. The custom loss function is minimized to estimate the model parameters of an RBFNN.

5.4. The Proposed PIML Model

In this chapter, a PIML model is proposed to predict surface roughness of FFF build parts. PIML models can be used over traditional machine/deep learning algorithms to incorporate the physical phenomena of a system. In the proposed model, physics-based knowledge is incorporated in two ways 1) using physics-based model output as an input in the PIML model and 2) using a physics-based term in the loss function. An RBFNN is trained using the new physics-based input with other inputs by minimizing the customized loss function. The proposed PIML is demonstrated in this section.

5.4.1. Physics-based Inputs

A physics-based model's prediction is used as an input with other input variables to develop a PIML model. A framework for incorporating physics-based knowledge in a data-driven method is given in Figure 5.1.

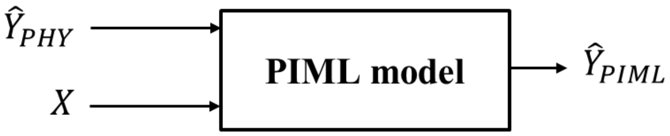


Figure 5.1. A framework of the PIML model

In data-driven methods, experimental and sensor-based input variables, $X,$ are used to train a model to predict a target variable. In addition to $X,$ another input variable, $\hat{Y}_{PHY}$, that is the output of a physics-based model has been used as an input to incorporate physics-based knowledge for more robust and physically consistent prediction. A physics-based model is developed under many assumptions, and the data-driven part of the PIML reduces the impacts of the assumptions [89]. The physics-based model should be a function of at least one input variable and other process constants. The target variable can be estimated for each observation from the physics-based model to use as an input. In this work, the target variable is surface roughness, and a physics-based empirical model from Ref. [101] is used to estimate the surface roughness. The estimated surface roughness is used as an input in the PIML model. The physics-based empirical model for the surface roughness estimation is given in Equation (5.1).

$$\hat{Y}_{PHY} = \begin{cases} (69.28\sim 72.36)\frac{t}{\cos\theta} & if\ \ 0^o \le \theta \le 70^o \\ \frac{1}{20}\left(90\hat{Y}_{PHY_{70^o}} - 70\hat{Y}_{PHY_{90^o}} + \theta\left(\hat{Y}_{PHY_{90^o}} - \hat{Y}_{PHY_{70^o}}\right)\right) & if\ 70^o \le \theta \le 90^o \\ 117.6 \times t & if\ \theta = 90^o \\ \hat{Y}_{PHY_{(\theta-90)}}(1+w) & if\ \ 90^o \le \theta \le 180^o \end{cases} \tag{5.1}$$

where t is the layer thickness and θ is the build orientation with the z-axis of an FFF machine's build platform. $\hat{Y}_{PHY}$ is the estimated surface roughness by the physics-based empirical model, and $\hat{Y}_{PHY_{\theta^o}}$ is the estimated surface roughness at θ^o build orientation. To incorporate physics-based knowledge in the proposed PIML, $\hat{Y}_{PHY}$ is be used as an input with other input variables. The inputs can be defined as

$$X_{PIML} = \left[X, \hat{Y}_{PHY}\right] \tag{5.2}$$

where, X_{PIML} is the features for the PIML model, and X can be process parameters, filament properties, and sensor data. $\hat{Y}_{PHY}$ is the physics-based knowledge that converts a data-driven model into a hybrid model, or the PIML model. In addition, a customized loss function is proposed to make a physically consistent prediction.

5.4.2. Customized Loss Function

Another strategy for incorporating physics-based knowledge is to use a physics-based term in the loss function to develop a customized loss function. A physics-based term is a regularization to reduce overfitting and physical inconsistency. A physics-based term in loss function is developed based on physics laws and physical constraints. Surface roughness is estimated as the average heights and depths across the surface. According to the definition and formula of the surface roughness, it is always a positive value, and a data-driven method prediction cannot guarantee it. For this, physics-based terms can be introduced with a data-driven method's loss function, for instance, mean square error (MSE). The generally used loss function to train a data-driven model is MSE, as given in Equation (5.3).

$$\mathbb{L}(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.3}$$

where, $\mathbb{L}(Y, \hat{Y})$ is the MSE loss function. Y and $\hat{Y}$ are the measured and prediction values of the target variable, respectively. A physics-based loss function is proposed in this chapter that regularizes the data-driven loss function by the constraint shown in Equation (5.4).

$$Y \geq 0 \tag{5.4}$$

The measured value of the target variable, surface roughness, is Y. A loss function is proposed that combines traditional loss function and physics-based regularization.

$$\mathbb{L}_{PIML}(Y, \hat{Y}_{PMIL}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_{i_{PIML}}\right)^2 + \frac{\lambda}{n}\sum_{i=1}^{n} Relu(-\hat{y}_{i_{PIML}}) \tag{5.5}$$

where, $\mathbb{L}_{PIML}(Y, \hat{Y}_{PMIL})$ is the proposed customized loss function that consists of MSE term and physics-based regularization. λ is a positive regularization hyper-parameter, and the value of λ is selected based on the importance of the physics-based loss function. Rectified linear unit (ReLU) function is used as regularization to incorporate physics-based knowledge. The output of

$Relu(-\hat{Y}_{PIML})$ is maximum between $-\hat{Y}_{PIML}$ and 0. The PIML model is trained by minimizing the loss function, $\mathbb{L}_{PIML}(Y, \hat{Y}_{PMIL})$. The target is to minimize both MSE terms and physics-based terms. In the physics-based term, the output of $Relu(-\hat{Y}_{PMIL})$ is zero when $\hat{Y}_{PIML}$ is positive and $-\hat{Y}_{PIML}$ when $\hat{Y}_{PIML}$ is negative. As the loss function is minimized during training, the physics-based term tries to be zero to generate physically consistent results. The MSE term in the loss function reduces the difference between actual and predicted surface roughness values. The proposed loss function simultaneously minimizes the MSE and ensures $\hat{Y}_{PIML}$ is positive. An assumption of using the proposed PIML model is that the trained model will also generate physically consistent results for test data. In other words, the predicted surface roughness for a new observation will also be positive if the PIML model is used to predict the surface roughness.

In the proposed PIML model, the physics-based empirical model output is used as input to train a PIML model. Besides, a physics-based term in the loss function is integrated with the MSE loss function term to train the proposed PIML model. Based on the physics-based input with other inputs and customized loss function, an RBFNN model is trained as a PIML model. The proposed PIML is demonstrated on an FFF process dataset.

5.4.3. Radial Basis Function Neural Network

A radial basis function neural network (RBFNN) is a one hidden layer neural network. The RBFNN model developed for PIML purposes has three layers, including the input and output layers, as shown in Figure 5.2. Each hidden layer neuron consists of a radial basis function (RBF). RBF is a non-linear activation function that converts input data based on the parameters of the RBF. The output obtained after converting input data by the RBF is linearly connected with the output layer neuron.

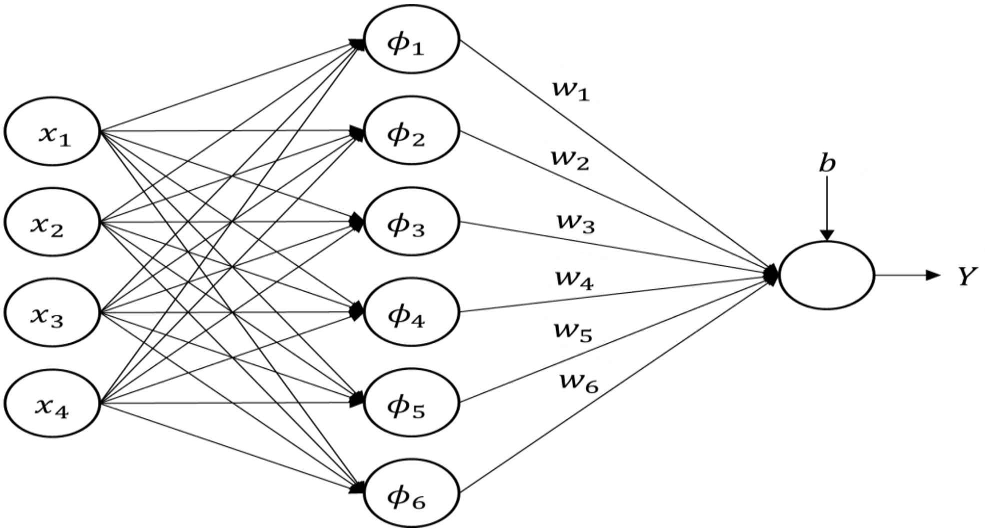


Figure 5.2. Radial basis function neural network

The training of RBFNN is two steps process. In the first step, the input data is converted by RBF, and the dimension of RBF layer output is the same as the number of neurons in the hidden layer. Usually, the number of neurons in the hidden layer is more than the number of input variables. The weights to connect RBF output with the target variable are estimated in the second step. This step can estimate the weights by simple linear regression or iteration.

The location of the function center and the deviation of data points from the center are two parameters of an RBF. There are many RBFs used to convert input data. The most widely used RBF is the Gaussian RBF. The Gaussian RBF is defined as

$$\phi_i = e^{-\frac{||x-c_i||^2}{2\sigma_i^2}} \tag{5.6}$$

In Equation (5.6), ϕ_i is the RBF. $c_i \in R^{1\times m}$ and σ_i are the center and deviations, respectively, for i^{th} RBF neuron. $x \in R^{1\times m}$ is an observation, and m is the number of input variables. A clustering technique can be used to estimate c_i and σ_i. The K-means clustering technique is applied to estimate the RBF parameters in this work. In this technique, the input data is partitioned into K clusters, and the number of neurons in the RBF layer is the value of K. The value of K is less than the number of observations (data points). The c_i and σ_i of the i^{th} cluster can be determined using the following formulas.

$$c_i = \frac{1}{p}\sum_{j=1}^{p} x_j \tag{5.7}$$

$$\sigma_i = \sqrt{\frac{1}{p}\sum_{j=1}^{p} || x_j - c_i ||^2} \tag{5.8}$$

In the above equations, p is the number of data points belonging to the i^{th}cluster. And, x_j is the j^{th} observation in the cluster. When, c_i and σ_i are known for each cluster, the input variables are transferred by using RBF in the hidden layers. The new dimension is the same as the number of neurons in the hidden layer. The target variable, surface roughness, is a linear combination of the outputs of the RBF neurons. To estimate the c_i and σ_i for each cluster, physics-based model output, $\hat{Y}_{PHY}$, is used as an input variable with X. The customized loss function given in Equation (5.4) is minimized to estimate the weights of the outputs of RBF neurons for the target variable. Using a FFF process surface roughness dataset, the RBFNN is trained as a PIML model by incorporating physics-based knowledge. This PIML model proposes a customized loss function by incorporating a physical constraint. In addition, a physics-based model estimation of the surface roughness is used as an input.

5.5. Case Study

In this section, the proposed PIML model is applied to an AM dataset to demonstrate the performance of the proposed method in terms of MSE, MAE, MAPE, and R^2 value. The proposed method is also compared with other methods.

5.5.1. Data Description

A real-world FFF process dataset is used to analyze the performance of the PIML model and compare the performance of the method with the physics-based model and data-driven RBFNN model. In the dataset, the input variables are layer thickness and build orientation. According to different studies, layer thickness and build orientation are the two most significant FFF process parameters for surface roughness [27, 102]. The data sources are given in Refs. [102, 103]. For the data collection, truncheon test parts are fabricated by the FFF process for different combinations of layer thickness and build orientation. The used dataset has a total of 106 observations. The combinations of the process parameters and the measured surface roughness are known for all observations.

The dataset is used to train the PIML model for performance evaluation. The RBFNN model is trained as a PIML model using physics-based input variables and customized loss function. A data-driven model, RBFNN, is trained without considering physics-based knowledge by the same dataset for comparison with the PIML model. The performance of the PIML model is compared with the physics-based model and the data-driven model in terms of MSE, MAE, MAPE, and R^2 value. In the PIML model, $\widehat{Y}_{PHY}$ estimated from Equation (5.1) is used as physics-based input with $\boldsymbol{X}$. The customized loss function given in Equation (5.5) is minimized to train RBFNN. On the other hand, for the data-driven model, input is only $\boldsymbol{X}$. The MSE loss function has been minimized to train the model. Equation (5.1) is used as a physics-based model to compare with the proposed PIML model. Besides, the PIML model is compared with a data-driven model from a published article.

5.5.2. Model Training

The dataset described in the above section is used to train RBFNN to estimate surface roughness. There are 106 observations (data points) in the dataset. For all observations, the physics-based model given in Equation (5.1) is used to estimate surface roughness, $\hat{Y}_{PHY}$. The estimated $\hat{Y}_{PHY}$ is used as an input with two other input variables, layer thickness and build orientation. Therefore, a total of three input variables are in the proposed model. In the next step, the dataset is divided into the test data and train data. Of the 106 observations, 86 observations are used to train the model, and the remaining 20 observations are applied to evaluate the performance of the proposed model. The number of neurons used in the hidden layer is 30. The training dataset is divided into 30 clusters to determine c_i and σ_i for all neurons in the RBF layer. After determining the cluster for all the 86 observations, c_i and σ_i for all neurons can be estimated by Equations (5.7) and (5.8), respectively. Then, the input variables' dimensions are expanded to 30 from 03 by RBF at the hidden layer. The constant value of the regularization hyperparameter is 0.01 ($\lambda = 0.1$). The surface roughness is a linear combination of the RBF neurons' outputs. The weights for the linear combination are estimated by minimizing the customized loss function given in Equation (5.5).

The proposed PIML model's performance is further compared with the physics-based and data-driven models. MSE, MAE, and MAPE for unseen data or test data are estimated as the basis of comparison between the proposed PIML with other methods. The case study results are introduced and discussed in the following section.

5.5.3. Results

The proposed PIML is demonstrated by applying it to predict surface roughness. An RBFNN is trained as the PIML model using a physics-based input and customized loss function.

The performance of the proposed PIML model is compared with other methods using MSE, MAE, MAPE and R^2 values.

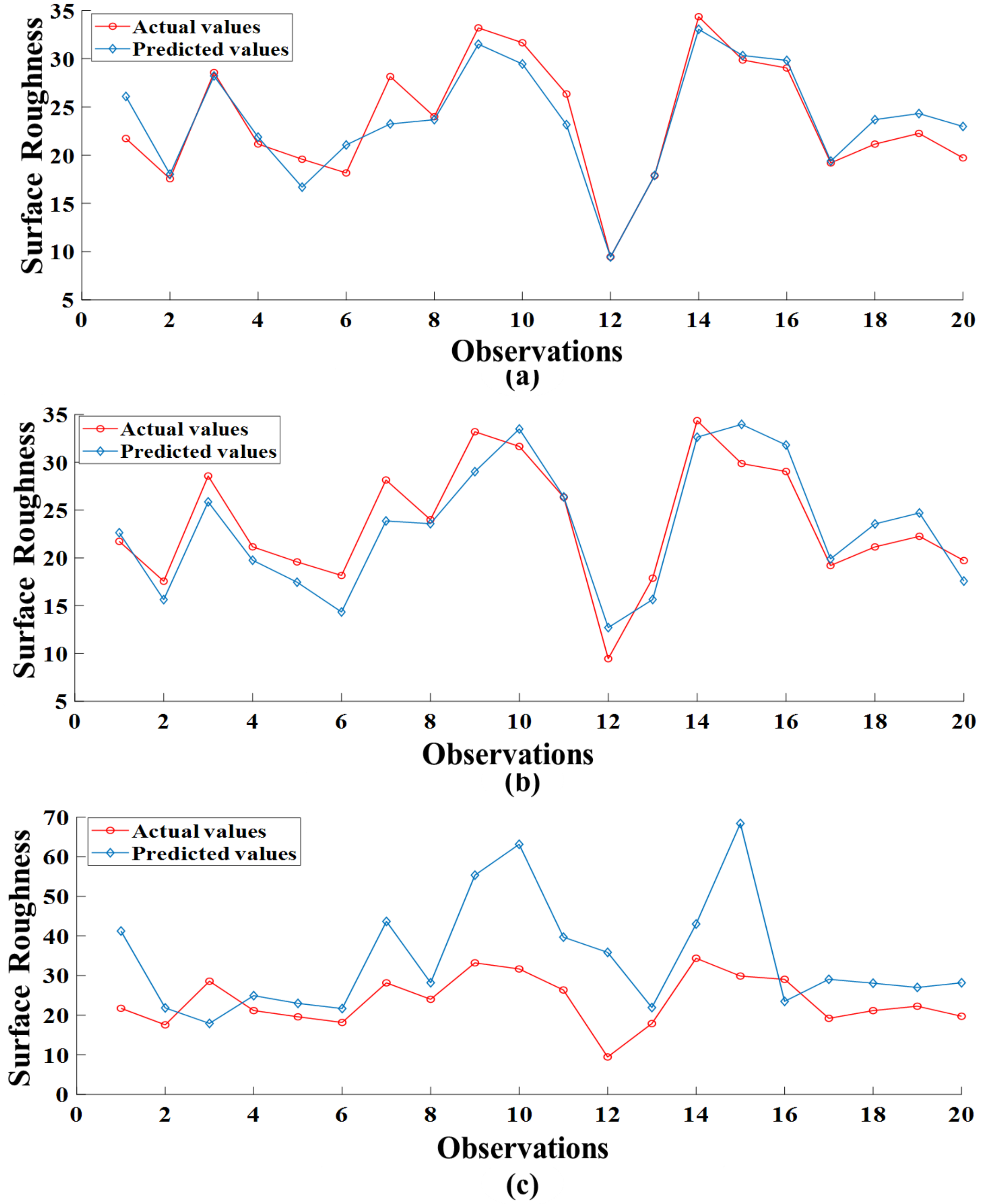


Figure 5.3. Surface roughness prediction by (a) proposed PIML model, (b) data-driven model, and (c) physics-based model

Table 5.1. Performance evaluation metrics for different models

Models	MAE	MSE	MAPE
PIML model	1.731	5.129	7.398
Data-driven model	2.270	6.608	10.533
Physics-based model	12.240	250.242	55.909

The minimum values of MAE, MSE, and MAPE indicate a low deviation of predicted values from the actual values. From Table 5.1, the values of all three-performance evaluation metrics are the lowest for the PIML model. The lowest values indicate the PIML predicted the surface roughness with high accuracy among the three models. The MAE, MSE, and MAPE of the PIML model are 24%, 22%, and 29%, respectively, lower than the data-driven model. The worst prediction accuracy for the physics-based model with the highest values for all three metrics. The low prediction accuracy of the physics-based model indicates the model is not able to estimate the surface roughness well. The data-driven model can predict well compared to the physics-based model. The data-driven model can estimate the surface roughness with a high level of accuracy. The prediction accuracy is further improved by adding embedding physics-based knowledge.

Besides, actual versus predicted scatter plots of surface roughness are further applied to compare the prediction performance of PIML and data-driven models. The actual versus predicted scatter plots for PIML, and data-driven models are shown in Figure 5.4. In Figure 5.4 (a), all coordinate points are close to the diagonal line with a high R^2-value, 0.865. It is visible in Figure 5.4 (b) that the model fitting is comparatively low with a 0.827 R^2-value. The higher R^2-value for the PIML model indicates high prediction accuracy and goodness of fit. Overall, the proposed PIML model performance overweighs the other two models.

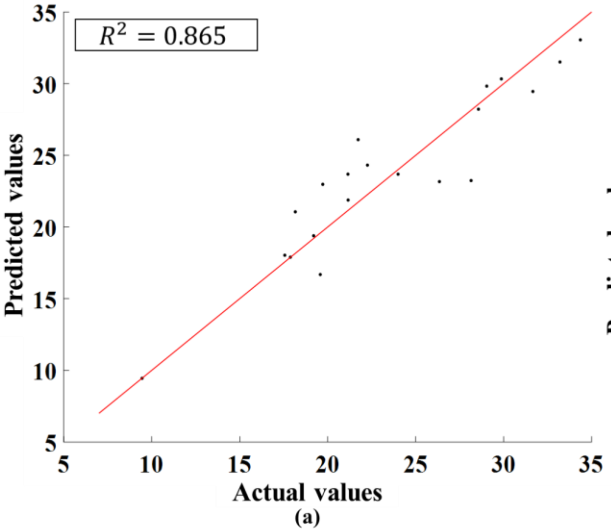


Figure 5.4. Scatterplots of predicted vs. measured surface roughness by (a) PIML model and (b) data-driven model

A published article by Vahabli and Rahmati [102] used a data-driven model to the surface roughness for the same data. In the proposed data-driven model, RBFNN and imperialist competitive algorithm (ICA), RBFNN-ICA, are combinedly applied to predict the surface roughness. The performance of the proposed PIML method compared with the RBFNN-ICA model. The MAE, MSE, MAPE, and R^2 values for both models are summarized in Table 5.2.

Table 5.2. Comparison between PIML model and RBFNN-ICA model

Models	MAE	MSE	MAPE	R^2-value
RBFNN-ICA model	2.01	8.87	7.19	0.9325
PIML model	1.73	5.129	7.398	0.865

In Table 5.2, MAE and MSE are improved by approximately 14% and 42% by the PIML model. On the other hand, MAPE and R^2 values are slightly worse than RBFNN-ICA for the PIML model. Overall, the prediction accuracy of the proposed method is slightly better than the RBFNN-ICA model.

The proposed PIML model is applied to predict the surface roughness for an FFF dataset. The PIML model is compared with three other methods. Overall, the prediction performance in terms of MAE, MSE, MAPE and R^2 values are evaluated for all the methods. The accuracy of the PIML model has overweighed the other methods.

5.6. Summary

A PIML model is proposed to estimate the surface roughness for FFF build parts. The proposed PIML model combines a data-driven model and physics-based knowledge. A physics-based model's surface roughness prediction is used as an input in the PIML model. Moreover, a customized loss function is proposed to incorporate physics-based knowledge in the PIML model. The proposed customized loss function combines physics-based regularization and MSE loss function to generate physically consistent estimation. An RBFNN model is trained as a PIML model using the physics-based input variable and the customized loss function. A real-world FFF process dataset is used to demonstrate the effectiveness of the PIML model. Besides, the MSE, MAE, MAPE, and R^2 values are estimated to compare the PIML model with a physics-based model and RBFNN data-driven model. An advantage of the proposed method is that the prediction by the PIML is physically consistent. The prediction accuracy of the proposed methods is improved significantly compared to physics-based and data-driven methods.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

6.1. Conclusions

Advanced numerical modeling techniques play a vital role in analyzing situations, gaining insights, and making essential decisions. The applications of advanced numerical modeling techniques are increasing in different fields, including manufacturing, due to the data availability, data-collecting devices, advanced algorithms, and high computational power. Applying data-driven methods in manufacturing, especially AM, is essential to take steps toward Industry 4.0 as AM is a technology of Industry 4.0. In this dissertation, different advanced data-driven numerical techniques are proposed and applied in manufacturing datasets for various purposes such as process parameter optimization, tool wear prediction, compressive strength estimation, and surface roughness estimation.

First, ANN is applied as an alternative to RSM to determine optimum combinations of FFF process parameters by optimizing two responses simultaneously, compressive strength and build time, as ANN is flexible and capable of capturing complex functional relationships. It is shown that ANN outperformed RSM. Besides, it has been concluded that the development of surrogate models plays a vital role in estimating and optimizing responses.

Second, an RRSO technique is proposed to estimate the distribution of a response instead of point value considering model parameter uncertainty. In addition, a BN-based approach is applied to determine an optimum combination of numerical and categorical process parameters to achieve desired requirements for one or more target variables.

Third, two linear optimization-based approaches are proposed to estimate conservative intervals for OLS and WLS regression models. The proposed approaches are compared with other exiting interval prediction techniques. It has been shown by applying an FFF dataset that the proposed conservative interval prediction methods are applicable when the level of uncertainty is high and wrong prediction is costly. Moreover, an MC dropout-based technique is applied to a CNC milling machine dataset to estimate tool wear as intervals by neural networks.

Finally, a PIML model is proposed to estimate surface roughness for FFF build parts. In the proposed approach, physics-based knowledge is embedded with a data-driven method to improve the physical consistency of prediction. According to an investigation of a FFF process dataset, the PIML outperformed a physics-based model and data-driven model.

Overall, there are many scopes of applying advanced numerical modeling techniques in manufacturing. This dissertation applies different approaches to predicting point values, approximating the intervals, and estimating probability distribution in additive manufacturing and traditional manufacturing datasets.

6.2. Future Work

Based on the conducted research in this dissertation, there are numerous scopes for further investigation and extensions in various domains. A list of further research scopes is given in the following.

- Research on *uncertainty quantification techniques* to increase robustness in prediction will be continued. The immediate plan is to develop a robust response surface model for multiple responses, and the distribution of model parameters will be used for robust estimation of the multiple responses. Additionally, other uncertainty sources in real-world manufacturing environments will be explored. Finally, different sources of uncertainty will be incorporated in interval prediction or distribution estimation by data-driven methods for making robust decisions.

- A challenge in applying data-driven methods is using *real-time different data sources and different data types* simultaneously. Other data types like videos, images, and numerical will be used for real-time process monitoring and quality analysis (e.g., internal defects) of AM build parts. Besides, cyber-attack is a common challenge in wireless communication for Industry 4.0. A future plan is to develop data-driven and game theory-based models to identify cyber-attacks and defects on CAD models due to cyber-attacks.
- Data-driven and hybrid condition monitoring techniques are currently applied to the energy sector. A limitation of the existing condition monitoring techniques is that these work for a system component. But, for better and more robust prediction, it is required to analyze the whole system. It is a plan to develop a condition monitoring setup to analyze a system instead of components individually. Besides, in the future, different data-driven methods will be explored to investigate the mechanical properties of composite materials and design composites with desired properties.

Scopes to apply data-driven methods, physics-informed machine learning methods, and other numerical models in manufacturing applications such as condition monitoring, quality control, and cyber-attack detection are essential to ease 21st-century manufacturing challenges and transition toward the Industry 4.0 era. Additionally, the advanced numerical methods proposed and attempted in this dissertation can be leveraged for innovative manufacturing technology development and applied in other sectors.

REFERENCES

1. Preuveneers, D. and E. Ilie-Zudor, *The intelligent industry of the future: A survey on emerging trends, research challenges and opportunities in Industry 4.0.* Journal of Ambient Intelligence and Smart Environments, 2017. **9**(3): p. 287-298.

2. Tao, F., et al., *Data-driven smart manufacturing.* Journal of Manufacturing Systems, 2018. **48**: p. 157-169.

3. Oztemel, E. and S. Gursev, *Literature review of Industry 4.0 and related technologies.* Journal of Intelligent Manufacturing, 2020. **31**(1): p. 127-182.

4. Pereira, A.C. and F. Romero, *A review of the meanings and the implications of the Industry 4.0 concept.* Procedia Manufacturing, 2017. **13**: p. 1206-1214.

5. Dingli, D.J., *The manufacturing industry–Coping with challenges*. 2012.

6. Thomas, A.J., P. Byard, and R. Evans, *Identifying the UK's manufacturing challenges as a benchmark for future growth.* Journal of Manufacturing Technology Management, 2012.

7. Ahuja, B., M. Karg, and M. Schmidt. *Additive manufacturing in production: challenges and opportunities*. in *Laser 3d manufacturing II*. 2015. International Society for Optics and Photonics.

8. Wuest, T., et al., *Machine learning in manufacturing: advantages, challenges, and applications.* Production & Manufacturing Research, 2016. **4**(1): p. 23-45.

9. Botchkarev, A., *Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology.* arXiv preprint arXiv:1809.03006, 2018.

10. Dey, A. and K. Zaman, *A robust optimization approach for solving two-person games under interval uncertainty.* Computers & Operations Research, 2020: p. 104937.

11. Wazed, M., S. Ahmed, and Y. Nukman, *Uncertainty factors in real manufacturing environment.* Australian Journal of Basic and Applied Sciences, 2009. **3**(2): p. 342-351.

12. Karpatne, A., et al., *Physics-guided neural networks (pgnn): An application in lake temperature modeling.* arXiv preprint arXiv:1710.11431, 2017.

13. Raissi, M., P. Perdikaris, and G.E. Karniadakis, *Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations.* arXiv preprint arXiv:1711.10561, 2017.

14. Khuri, A.I. and S. Mukhopadhyay, *Response surface methodology.* Wiley Interdisciplinary Reviews: Computational Statistics, 2010. **2**(2): p. 128-149.

15. Vuchkov, I.N. and N. Boyadjieva, *Quality improvement with design of experiments: A response surface approach*. Vol. 7. 2013: Springer Science & Business Media.

16. Bucher, C.G. and U. Bourgund, *A fast and efficient response surface approach for structural reliability problems.* Structural safety, 1990. **7**(1): p. 57-66.

17. Peng, A., X. Xiao, and R. Yue, *Process parameter optimization for fused deposition modeling using response surface methodology combined with fuzzy inference system.* The International Journal of Advanced Manufacturing Technology, 2014. **73**(1-4): p. 87-100.

18. Dey, A., D. Hoffman, and N. Yodo, *Optimizing multiple process parameters in fused deposition modeling with particle swarm optimization.* International Journal on Interactive Design and Manufacturing (IJIDeM), 2019: p. 1-13.

19. Hassoun, M.H., *Fundamentals of artificial neural networks*. 1995: MIT press.

20. Basheer, I.A. and M. Hajmeer, *Artificial neural networks: fundamentals, computing, design, and application.* Journal of microbiological methods, 2000. **43**(1): p. 3-31.

21. Ciaburro, G. and B. Venkateswaran, *Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles*. 2017: Packt Publishing Ltd.

22. Heo, S. and J.H. Lee, *Statistical process monitoring of the Tennessee Eastman process using parallel autoassociative neural networks and a large dataset.* Processes, 2019. **7**(7): p. 411.

23. Srinivas, N. and K. Deb, *Muiltiobjective optimization using nondominated sorting in genetic algorithms.* Evolutionary computation, 1994. **2**(3): p. 221-248.

24. Deb, K., *Multi-objective optimisation using evolutionary algorithms: an introduction*, in *Multi-objective evolutionary optimisation for product design and manufacturing*. 2011, Springer. p. 3-34.

25. Dey, A., *Methodology for solving two person game under interval uncertainty.* 2018.

26. Yodo, N. and A. Dey, *Multi-Objective Optimization for FDM Process Parameters with Evolutionary Algorithms.* Fused Deposition Modeling Based 3D Printing: p. 419.

27. Dey, A. and N. Yodo, *A systematic survey of FDM process parameter optimization and their influence on part characteristics.* Journal of Manufacturing and Materials Processing, 2019. **3**(3): p. 64.

28. Raney, K., E. Lani, and D.K. Kalla, *Experimental characterization of the tensile strength of ABS parts manufactured by fused deposition modeling process.* Materials Today: Proceedings, 2017. **4**(8): p. 7956-7961.

29. Nidagundi, V.B., R. Keshavamurthy, and C. Prakash, *Studies on parametric optimization for fused deposition modelling process.* Materials Today: Proceedings, 2015. **2**(4-5): p. 1691-1699.

30. Dey, A., I.N. Roan Eagle, and N. Yodo, *A Review on Filament Materials for Fused Filament Fabrication.* Journal of manufacturing and materials processing, 2021. **5**(3): p. 69.

31. D695-15, A.I.A., *Standard test method for compressive properties of rigid plastics*. 2015, Pennsylvania United States.

32. Popescu, D., et al., *FDM process parameters influence over the mechanical properties of polymer specimens: A review.* Polymer Testing, 2018. **69**: p. 157-166.

33. Chauhan, V.S., N.K. Bhardwaj, and S.K. Chakrabarti, *Application of response surface methodology and central composite design for the optimization of talc filler and retention aid in papermaking.* 2013.

34. Khuri, A. and J. Cornell, *Response Surfaces: Designs and Analyses, second edn, M*. 1996, Dekker, New York, United States.

35. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.

36. Box, G.E. and K.B. Wilson, *On the experimental attainment of optimum conditions.* Journal of the royal statistical society: Series b (Methodological), 1951. **13**(1): p. 1-38.

37. Bradley, N., *The response surface methodology*. 2007, Indiana University South Bend.

38. Dey, A. and N. Yodo. *Robust Response Surface Optimization under Model Parameter Uncertainty*. in *IIE Annual Conference. Proceedings*. 2021. Institute of Industrial and Systems Engineers (IISE).

39. Ledolter, J. and R.H. Kardon, *Focus on data: statistical design of experiments and sample size selection using power analysis.* Investigative Ophthalmology & Visual Science, 2020. **61**(8): p. 11-11.

40. Omlin, M. and P. Reichert, *A comparison of techniques for the estimation of model prediction uncertainty.* Ecological modelling, 1999. **115**(1): p. 45-59.

41. Cestnik, B. *Estimating probabilities: a crucial task in machine learning*. in *ECAI*. 1990.

42. Haldar, A. and S. Mahadevan, *Probability, reliability, and statistical methods in engineering design*. 2000: J. Wiley & Sons, Incorporated.

43. Dey, A. and N. Yodo. *Decision Analysis for Selecting FDM Process Parameters using Bayesian Network Approach*. in *IIE Annual Conference. Proceedings*. 2020. Institute of Industrial and Systems Engineers (IISE).

44. Yodo, N. and P. Wang, *Resilience Modeling and Quantification for Engineered Systems Using Bayesian Networks.* Journal of Mechanical Design, 2016. **138**(3): p. 031404-031404.

45. Benesty, J., et al., *Pearson correlation coefficient*, in *Noise reduction in speech processing*. 2009, Springer. p. 1-4.

46. McHugh, M.L., *The chi-square test of independence.* Biochemia medica, 2013. **23**(2): p. 143-149.

47. Kornbrot, D., *Point biserial correlation.* Wiley StatsRef: Statistics Reference Online, 2014.

48. Lasi, H., et al., *Industry 4.0.* Business & information systems engineering, 2014. **6**(4): p. 239-242.

49. Dhanasekaran, P., D.K. Kalla, and N. Yodo. *Recent advances in cutting tool materials and tool coatings for machining composite materials*. in *Tooling for Composites Conference 2010*. 2010.

50. Zhu, K. and B. Vogel-Heuser, *Sparse representation and its applications in micro-milling condition monitoring: noise separation and tool condition monitoring.* The International Journal of Advanced Manufacturing Technology, 2014. **70**(1-4): p. 185-199.

51. Xia, T., et al., *Recent advances in prognostics and health management for advanced manufacturing paradigms.* Reliability Engineering & System Safety, 2018. **178**: p. 255-268.

52. Dey, A. and N. Yodo. *A Dropout-based Neural Network Framework for Tool Wear Prediction under Uncertainty*. in *IIE Annual Conference. Proceedings*. 2021. Institute of Industrial and Systems Engineers (IISE).

53. Zaman, K., et al., *Robustness-based design optimization under data uncertainty.* Structural and Multidisciplinary Optimization, 2011. **44**(2): p. 183-197.

54. Hüllermeier, E. and W. Waegeman, *Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction.* arXiv preprint arXiv:1910.09457, 2019.

55. Segù, M., A. Loquercio, and D. Scaramuzza, *A general framework for uncertainty estimation in deep learning.* arXiv preprint arXiv:1907.06890, 2019.

56. Huang, N.E., et al., *The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis.* Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 1998. **454**(1971): p. 903-995.

57. Schlurmann, T. *The empirical mode decomposition and the Hilbert spectra to analyse embedded characteristic oscillations of extreme waves*. in *Rogue waves*. 2001.

58. Huang, N.E., *Hilbert-Huang transform and its applications*. Vol. 16. 2014: World Scientific.

59. Shang-yue, Z., L. Yuan-yuan, and Y. Gong-liu. *EMD interval thresholding denoising based on correlation coefficient to select relevant modes*. in *2015 34th Chinese Control Conference (CCC)*. 2015. IEEE.

60. Altın, C. and O. Er, *Comparison of different time and frequency domain feature extraction methods on elbow gesture's EMG.* European journal of interdisciplinary studies, 2016. **2**(3): p. 35-44.

61. Dargie, W. *Analysis of time and frequency domain features of accelerometer measurements*. in *2009 Proceedings of 18th International Conference on Computer Communications and Networks*. 2009. IEEE.

62. Wu, J., et al., *Multi-sensor information fusion for remaining useful life prediction of machining tools by adaptive network based fuzzy inference system.* Applied Soft Computing, 2018. **68**: p. 13-23.

63. Box, G.E., *Science and statistics.* Journal of the American Statistical Association, 1976. **71**(356): p. 791-799.

64. Srivastava, N., et al., *Dropout: a simple way to prevent neural networks from overfitting.* The journal of machine learning research, 2014. **15**(1): p. 1929-1958.

65. Gal, Y. and Z. Ghahramani. *Dropout as a bayesian approximation: Representing model uncertainty in deep learning*. in *international conference on machine learning*. 2016.

66. Society, P. *PHM data chalelnge 2010*. 2010; Available from: https://www.phmsociety.org/competition/phm/10 (accessed 7/25/2020, 2020).

67. Cai, W., et al., *A hybrid information model based on long short-term memory network for tool condition monitoring.* Journal of Intelligent Manufacturing, 2020: p. 1-14.

68. Zhao, R., et al., *Learning to monitor machine health with convolutional bi-directional LSTM networks.* Sensors, 2017. **17**(2): p. 273.

69. Kendall, A. and Y. Gal. *What uncertainties do we need in bayesian deep learning for computer vision?* in *Advances in neural information processing systems*. 2017.

70. Hammer, B. and T. Villmann. *How to process uncertainty in machine learning?* in *ESANN*. 2007. Citeseer.

71. Solomatine, D.P. and D.L. Shrestha, *A novel method to estimate model uncertainty using machine learning techniques.* Water Resources Research, 2009. **45**(12).

72. Myers, R.H., et al., *Generalized linear models: with applications in engineering and the sciences*. Vol. 791. 2012: John Wiley & Sons.

73. Dey, A. and N. Yodo, *Conservative Confidence Interval Prediction in Fused Deposition Modeling Process With Linear Optimization Approach.* ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg, 2022. **8**(1).

74. Sykes, A.O., *An introduction to regression analysis.* 1993.

75. Baturynska, I., *Statistical analysis of dimensional accuracy in additive manufacturing considering STL model properties.* The International Journal of Advanced Manufacturing Technology, 2018. **97**(5-8): p. 2835-2849.

76. Chiarini, A. and F. Brunetti, *What really matters for a successful implementation of Lean production? A multiple linear regression model based on European manufacturing companies.* Production Planning & Control, 2019. **30**(13): p. 1091-1101.

77. Sidey-Gibbons, J.A. and C.J. Sidey-Gibbons, *Machine learning in medicine: a practical introduction.* BMC medical research methodology, 2019. **19**(1): p. 64.

78. Jarque, C.M. and A.K. Bera, *Efficient tests for normality, homoscedasticity and serial independence of regression residuals.* Economics letters, 1980. **6**(3): p. 255-259.

79. Stine, R.A., *Bootstrap prediction intervals for regression.* Journal of the American Statistical Association, 1985. **80**(392): p. 1026-1031.

80. Shrestha, D.L. and D.P. Solomatine, *Machine learning approaches for estimation of prediction interval for the model output.* Neural Networks, 2006. **19**(2): p. 225-235.

81. Olive, D.J., *Prediction intervals for regression models.* Computational statistics & data analysis, 2007. **51**(6): p. 3115-3122.

82. James, G., et al., *An introduction to statistical learning*. Vol. 112. 2013: Springer.

83. Friedman, J., T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Vol. 1. 2001: Springer series in statistics New York.

84. Pandey, P.M., K. Thrimurthulu, and N.V. Reddy*, *Optimal part deposition orientation in FDM by using a multicriteria genetic algorithm.* International Journal of Production Research, 2004. **42**(19): p. 4069-4089.

85. Gurrala, P.K. and S.P. Regalla, *Multi-objective optimisation of strength and volumetric shrinkage of FDM parts: a multi-objective optimization scheme is used to optimize the strength and volumetric shrinkage of FDM parts considering different process parameters.* Virtual and Physical Prototyping, 2014. **9**(2): p. 127-138.

86. Wang, J., P. Wang, and R.X. Gao, *Enhanced particle filter for tool wear prediction.* Journal of Manufacturing Systems, 2015. **36**: p. 35-45.

87. Li, X., Q. Ding, and J.-Q. Sun, *Remaining useful life estimation in prognostics using deep convolution neural networks.* Reliability Engineering & System Safety, 2018. **172**: p. 1-11.

88. Kashinath, K., et al., *Physics-informed machine learning: case studies for weather and climate modelling.* Philosophical Transactions of the Royal Society A, 2021. **379**(2194): p. 20200093.

89. Kapusuzoglu, B. and S. Mahadevan, *Physics-informed and hybrid machine learning in additive manufacturing: application to fused filament fabrication.* Jom, 2020. **72**(12): p. 4695-4705.

90. Jia, X., et al., *Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles.* ACM/IMS Transactions on Data Science, 2021. **2**(3): p. 1-26.

91. Mondal, B., T. Mukherjee, and T. DebRoy, *Crack free metal printing using physics informed machine learning.* Acta Materialia, 2021: p. 117612.

92. Wang, C., et al., *Machine learning in additive manufacturing: State-of-the-art and perspectives.* Additive Manufacturing, 2020. **36**: p. 101538.

93. Zhu, Q., Z. Liu, and J. Yan, *Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks.* Computational Mechanics, 2021. **67**(2): p. 619-635.

94. Liu, R., S. Liu, and X. Zhang, *A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing.* The International Journal of Advanced Manufacturing Technology, 2021. **113**(7): p. 1943-1958.

95. Moges, T., et al., *Hybrid modeling approach for melt-pool prediction in laser powder bed fusion additive manufacturing.* Journal of Computing and Information Science in Engineering, 2021. **21**(5).

96. Du, Y., T. Mukherjee, and T. DebRoy, *Physics-informed machine learning and mechanistic modeling of additive manufacturing to reduce defects.* Applied Materials Today, 2021. **24**: p. 101123.

97. Wang, Z., et al., *Uncertainty quantification in metallic additive manufacturing through physics-informed data-driven modeling.* Jom, 2019. **71**(8): p. 2625-2634.

98. Kats, D., et al., *A physics-informed machine learning method for predicting grain structure characteristics in directed energy deposition.* Computational Materials Science, 2022. **202**: p. 110958.

99. Ren, K., et al., *Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning.* Computer Methods in Applied Mechanics and Engineering, 2020. **362**: p. 112734.

100. Guo, S., et al., *Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm.* Journal of Manufacturing Systems, 2022. **62**: p. 145-163.

101. Rahmati, S. and E. Vahabli, *Evaluation of analytical modeling for improvement of surface roughness of FDM test part using measurement results.* The International Journal of Advanced Manufacturing Technology, 2015. **79**(5): p. 823-829.

102. Vahabli, E. and S. Rahmati, *Application of an RBF neural network for FDM parts' surface roughness prediction for enhancing surface quality*. International journal of precision engineering and manufacturing, 2016. **17**(12): p. 1589-1603.

103. Nourghassemi, B., *Surface roughness estimation for FDM systems.* Master of Applied Science of Mechanical Engineering Ryerson University, Toronto, 2011.