Manufacturing Process Causal Knowledge Discovery using a Modified Random Forest-based Predictive Model

Swansea University
Prifysgol Abertawe

Meshari A. Al-Ebrahim
College of Engineering
Swansea University

Submitted to Swansea University in Fulfilment of the Requirements for the Degree of
"Doctor of Philosophy, Ph.D"

2020

Abstract
Manufacturing Process Causal Knowledge Discovery using a Modified Random Forest-based Predictive Model

A Modified Random Forest algorithm (MRF)-based predictive model is proposed for use in manufacturing processes to estimate the effects of several potential interventions, such as (i) altering the operating ranges of selected continuous process parameters within specified tolerance limits, (ii) choosing particular categories of discrete process parameters, or (iii) choosing combinations of both types of process parameters. The model introduces a non-linear approach to defining the most critical process inputs by scoring the contribution made by each process input to the process output prediction power. It uses this contribution to discover optimal operating ranges for the continuous process parameters and/or optimal categories for discrete process parameters. The set of values used for the process inputs was generated from operating ranges identified using a novel Decision Path Search (DPS) algorithm and Bootstrap sampling.

The odds ratio is the ratio between the occurrence probabilities of desired and undesired process output values. The effect of potential interventions, or of proposed confirmation trials, are quantified as posterior odds and used to calculate conditional probability distributions. The advantages of this approach are discussed in comparison to fitting these probability distributions to Bayesian Networks (BN).

The proposed explainable data-driven predictive model is scalable to a large number of process factors with non-linear dependence on one or more process responses. It allows the discovery of data-driven process improvement opportunities that involve minimal interaction with domain expertise. An iterative Random Forest algorithm is proposed to predict the missing values for the mixed dataset (continuous and categorical process parameters). It is shown that the algorithm is robust even at high proportions of missing values in the dataset.

The number of observations available in manufacturing process datasets is generally low, e.g. of a similar order of magnitude to the number of process parameters. Hence, Neural Network (NN)-based deep learning methods are generally not applicable, as these techniques require 50-100 times more observations than input factors (process parameters).

The results are verified on a number of benchmark examples with datasets published in the literature. The results demonstrate that the proposed method outperforms the comparison approaches in term of accuracy and causality, with linearity assumed. Furthermore, the computational cost is both far better and very feasible for heterogeneous datasets.

Declaration and Statements

DECLARATION

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .... Meshari Al-Ebrahim ....
Date .... 29/09/2020 ....

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .... Meshari Al-Ebrahim ....
Date .... 29/09/2020 ....

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed .... Meshari Al-Ebrahim ....
Date .... 29/09/2020 ....

Meshari A. Al-Ebrahim
29 September 2020

Acknowledgements

First, I would like to express my deepest gratitude to my first supervisor, Dr. Rajesh Ransing, for all the endless help, encouragement, patience, support, and guidance during the study and completion of this dissertation. In particular, I would like to thank him for sharing his knowledge and critical insights which are useful and have contributed immensely to the quality of my research. Also, I would like to thank him for his support, encouragement, and sympathy when I was taking care of my mom's cancer; he was always behind me to give me all the support I needed. If I could contribute anything to the field of smart manufacturing, it's because I stood on the shoulders of giants. I am also very grateful to my second supervisor, Prof. Perumal Nithiarasu.

I would like to thank my mom and my eldest sister for their support, as they have been always by my side in any matter or issue. I would like to dedicate this thesis to my mom, who died on May 2^{nd}, 2018. She did everything she could to empower and help me to be the best version of myself including moral and financial support. The values she had instilled in me continued to be my guiding star; these values have always helped me overcome challenges and stay focused. Her dream was to see me holding my Ph.D degree, and now her dream is about to come true. Also, I would like to dedicate this thesis to my eldest sister, who has always had faith in me and will always be the most significant support in my life. Without my mother, sister, and Dr. Ransing's encouragement, I couldn't reach this point.

I would like to thank everyone in the College of Engineering at Swansea University, especially Dr. Raed Batbooti, for sharing his expertise. A huge appreciation goes to my family, especially my parents, sisters, brother, and my wife, for their patience and encouragement. Finally, I would like to offer my special appreciation to my lovely daughters (Noor and Faiqa) for the joy and happiness they have brought to my life.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Abbreviations List

PCPHR	Principal Component of Proportional Hazard Regression
SMOTE	Synthetic Minority Over-Sampling Technique
NRMSE	Normalised Root Mean Square Error
FAMD	Factorial Analysis for Mixed Data Method
CART	Classification and Regression Tree
IRLS	Iterative Reweighted Least Squares
HRSG	Heat Recovery Steam Generators
CCPP	Combined Cycle Power Plant
RMSE	Root Mean Square Error
PFCs	Proportion of Falsely Classified Entities
CTQs	Critical to the Quality Characteristics
KPCs	Key Product Characteristics
SPC	Statistical Process Control
QRT	Quantile Regression Tree
QCA	Quality Correlation Algorithm
MRF	Modified Random Forest Algorithm
SOV	Stream of Variation Analysis

SVM	Support Vector Machines
PCA	Principal Component Analysis
PCS	Principal Component Subspace
PHM	Proportional Hazards Model
MFA	Multiple Factor Analysis
MCA	Multiple Corresponding Analysis
FDA	Fuzzy Data Analysis
ANN	Artificial Neural Networks
MTY	Mason, Tracy and Young Method
FEA	Finite Element Analysis
SVD	Singular Value Decomposition
GAM	Generalised Additive Model
PLS	Partial Least Square
SPE	Squared Prediction Error
CLI	Co-Linearity Index
KNN	K-Nearest-Neighbour
MSE	Mean Square Error
LAD	Least Absolute Deviations
TSR	Trimmed Score Regression Method
KDR	Known Data Regression Method
IDE	Integrated Development Environment
IQR	Inter-Quartile Range
DPS	Decision Path Search Algorithm

OOB	Out-of-Bag Measurements
HPC	High Performance Concrete
ID3	Iterative Dichotomiser-3
LSL	Lower Specification Limit
USL	Upper Specification Limit
AI	Artificial Intelligence
BI	Bayesian Inference
BN	Bayesian Network Method
CV	Cross Validation
DT	Decision Tree
FS	Feature Selection
GT	Gas Turbine
HB	Higher the Better
IG	Information Gain
LB	Lower the Better
LL	Lower Limit
NN	Neural Network
NV	Naive Predictor
PM	Penalty Matrix Approach
RF	Random Forest Algorithm
RS	Residual Subspace
ST	Steam Turbine
UL	Upper Limit

Dissertation Symbol

$I_{permute}(j)$	Permutation importance of the j_{th} feature
$E(Y - \hat{Y})^2$	Square expected difference between actual and predicted response
$S_{synthetic}$	New Generated synthetic sample
S_{sample}	Original minority class sample
$G(Q, \theta)$	Minimisation of impurity function
$x_{mis}^{(s)}$	The variables other than X_s with observations $i_{mis}^{(s)}$
$x_{obs}^{(s)}$	The variables other than X_s with observations $i_{obs}^{(s)}$
X_{new}	New generated observation from over-sampling technique
X^{true}	Complete data matrix
X^{imp}	Imputed data matrix
X_{old}^{imp}	Observation matrix before imputing the missing value
X_{new}^{imp}	Observation matrix after imputing the missing value
Y_{new}	New generated response from over-sampling technique
$y_{obs}^{(s)}$	The observed values of the observation X_s of missing values
$y_{mis}^{(s)}$	The missing values of the observation X_s
$Var(\epsilon)$	Variance of the uncertainty error
T_{max}	Maximum number of tree's in the forest
$ntree$	Number of trees in the forest
$h(T_b)$	The OOB prediction accuracy
Th_{op}	Optimal threshold
Th_{max}	Maximum penalty matrix threshold
Th_{min}	Minimum penalty matrix threshold

$H_e(X_m)$	Entropy impurity function
$H_g(X_m)$	Gini index impurity function
$nLabel$	The amount of categorise in the attributes
N_{mis}	Number of missing values in the categorical variables
∇RN	Continuous feature variables
∇F	Categorical feature variables
FTR	First time right approach
$f(X)$	Actual model relation
$\hat{f}(X)$	Regression model
P_N	Amount of SMOTE
O_P	Optimal number of trees
O_b	Out-of-bag sample in the decision tree
X_s	Observation including missing values
R^n	Observation range
R^l	Label response range
π_v	Probability of avoid
π_p	Probability of optimal
π_s	Probability of success
π_f	Probability of failure
R^2	Coefficient of determination
n_c	Number of correlated parameters resulted from applying CLI
m_0	Root node in the decision tree
t_m	Splitting threshold point

T_b	Sample OOB for a single b_{th} tree
S_T	Number of existing minority class sample
SS	Sum of squares
Q	Partition in the data set (data in the node)
N	Total data number in the node
θ	Split location point
θ^*	Optimal location point for minimising the impurity function
I	Class feature value
T	Decision tree
C	Set of classes
p	Class proportion
j	The feature/factor
A	Attribute in class splitting set
m	Node in the decision tree
B	Total number of bootstraps
b	Bootstrap sample
H	Entropy
ϵ	Error output
n	Normalisation by number of elements
Ω	Odds ratio
L	Data set values
Y	Actual response output
$\bar{Y}$	Average response

$\hat{Y}$	Predicted response
$\bar{X}$	Average observation
μ	Mean value
σ	Standard deviation
t	Time (second)
x	New unseen observation
X	Process observation input (Training data set)
γ	Stopping criterion
K	Number of nearest neighbour
k	Number of fold cross validation
f	Feature type
V	Response matrix size

Case Studies Factors

$\%Al$	Aluminium
$\%B$	Boron
$\%C$	Carbon
$\%Co$	Cobalt
$\%Cr$	Chromium
$\%Fe$	Iron
$\%Mo$	Molybdenum
$\%N$	Nitrogen
$\%Nb$	Niobium
$\%O$	Oxygen

$\%Ta$	Tantalum
$\%Ti$	Titanium
$\%W$	Tungsten
$\%Zr$	Zirconium
Ad	Glazing Area Distribution
Ag	Glazing Area
AP	Atmospheric Pressure
Ar	Roof Area
As	Surface Area
AT_c	Age of Testing
AT_p	Ambient Temperature
Aw	Wall Area
BS	Blast-Furnace Slag
C	Cement
CA	Coarse Aggregate
CL	Cooling Load
CS	Compressive Strength
F	Fly-Ash
FA	Fine Aggregate
H	Overall Height
HL	Heating Load
O	Orientation
P	Superplasticiser

PE	Electrical Power
Rc	Relative Compactness
RH	Relative Humidity
V	Vacuum
W	Water

Chapter 1

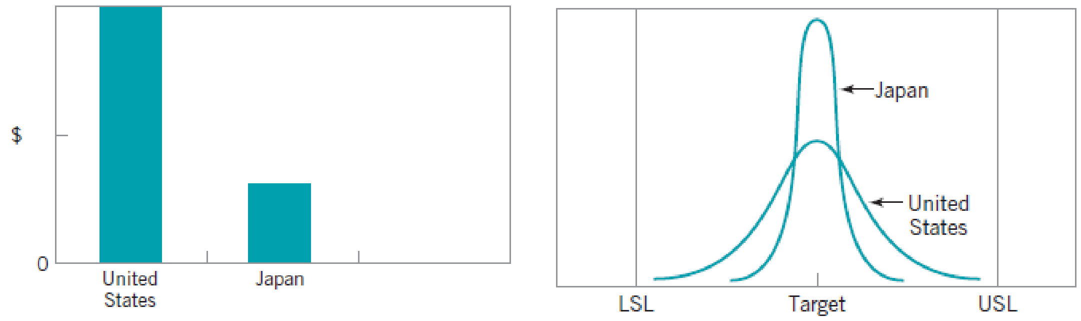
Introduction

The fourth industrial revolution, commonly referred to as 'Industry 4.0', will involve a complete digital transformation of several manufacturing operations. This transition will give rise to new developments in intelligent, co-operational, and integrated manufacturing processes that are capable of tracking real-time operating progress in order to monitor expenses, minimise downtime, and avoid system failures [1]. In this chapter, the main topics and challenges of the present dissertation are introduced and addressed. In Section 1.1, the manufacturing problem of interest and the necessary terminology pertaining to the statistical process control are defined, followed by the description of the investment casting process as a case study in Section 1.2. In Section 1.3, the aims and objectives of the study are outlined and discussed. Publishable results for potential journal papers are illustrated and included in Section 1.4. In Section 1.5, the research roadmap is depicted in the context of demonstrating how this work fits in with the Swansea University's overall research, as well as how it builds upon the findings of previous researchers group in the same field. Finally, Section 1.6 comprises the layout of this dissertation.

1.1 Industrial Research Context and Problem Definition

The development and enhancement of quality management systems have become core management practices for both industrial companies and service providers. Improved quality is a significant benefit in business. An organisation that can satisfy its consumers by monitoring to improve quality will overtake its rivals. Quality is inversely related to both volatility and variability, which implies that with a decline in the number of vital characteristics of the product, the product's condition will improve. As an example, one automotive manufacturer in the USA conducted a comparative survey on vehicle transmission systems that were locally manufactured and those produced by a

Japanese supplier. As Figure 1.1a shows, a remarkable distinction was observed between these sources (US manufacturers and the Japanese supplier) on terms of warranty claims and repair costs. The Japanese transmission had a much lower cost; moreover, by measuring several critical quality measures from random samples within each plant, as demonstrated in Figure 1.1b, the distributions for both principal dimensions are centralised at the target value. Nevertheless, the distribution of significant characteristics of transmissions produced in the US is approximately 75% of the width of the specifications. For the Japanese transmissions, on the other hand, the identical major features are allocated only around 25% of the specification band. Therefore, the root cause behind the cost-performance discrepancy was found to be the the smaller variability in the essential quality specifications of the Japanese-made transmissions compared to those manufactured in the US.

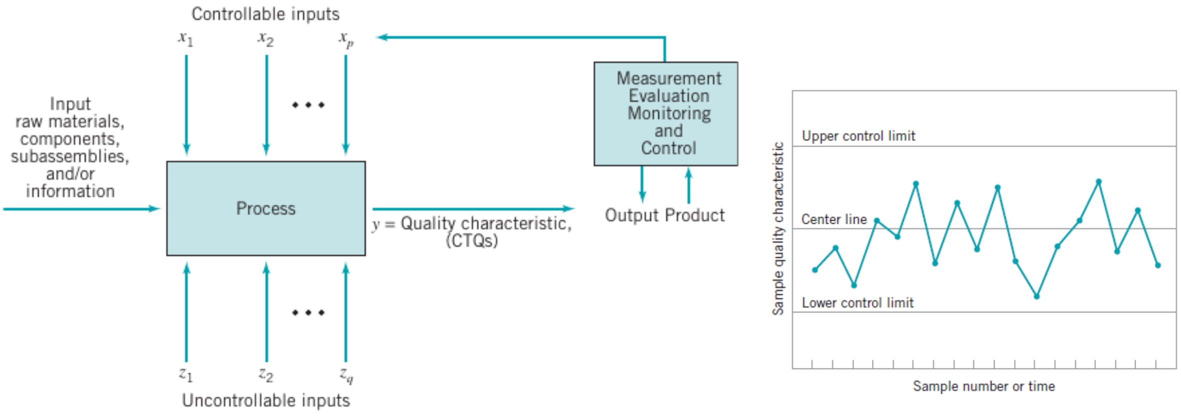


(a) Warranty costs for transmissions

(b) Critical dimensions for transmissions

Figure 1.1: Cost and critical dimensions variability range comparison [2]

When it comes to the mass-production of systems and components, reduction of variability in processes is the recipe for reducing the total cost, enhancing functionality, and boosting customer satisfaction. Excess variation leads to scrapping and reworking, more customer returns, and decreased serviceability and durability. In the foundry industry, in 2017, global casting production was around 109.8 million tons [3]. The mean cost of production for ferrous foundries was 1.7 billion Euros per million tons manufactured, and 4.94 billion Euros for every million tons of non-ferrous castings [4]. It can also be noted that the international casting industries created casts worth over 130 billion Euros in 2017. Tremendous advances have been implemented in the field of foundry technologies involving computer simulations, moulding machines, binder formulations, and alloy production. Commonly, factories would lose 4–5% of their annual revenues. In 2017, the direct expenses to the global economy contributed around 1.3 billion Euros based on the 1% refusal rate in foundries. These costs could potentially be reduced, or even eliminated, if efforts were made to derive product-specific process knowledge from the in-process statistics and to recycle it in order to optimise the casting procedures for current and emerging casting parts. According to the statistical process control framework, in any manufacturing process, (see Figure 1.2a), regardless of how well a process is designed, there is always a certain amount of natural variability around the target process output, referred to as critical-to-quality characteristics (CTQs). There are two types of variability. The first is inherited naturally from the combined effect of many small unavoidable causes. This type of variability is called **common cause variation**. This is an allowable variation, and the process response values remain during the process' upper (USL) and lower specification limits (LSL), as represented in Figure 1.2b. A process that operates in the presence of common causes is said to be an 'in-control' process. The second type of variability often occurs in the outcome of a process as a results of factors like improper machine adjustment, machine defects, operator errors, and defective raw material. Typically, the majority of these process results lie outside the specifications range compared to the natural variability and result in the unacceptable quality level of the process output. These sources of variability are typically referred to as **special or assignable causes variation**. A system that operates in the presence of assignable causes is considered to be an 'out-of-control' process [2]. The LSLs and USLs represent the total range for each factor in the process. These limits are extracted from the industry by the process engineer in order to visualise the range limits for each process.


Controllable inputs
x_1
x_2
x_p
Input raw materials, components, subassemblies, and/or information
Process
y = Quality characteristic, (CTQs)
Measurement Evaluation Monitoring and Control
Output Product
z_1
z_2
z_q
Uncontrollable inputs
Upper control limit
Center line
Lower control limit
Sample quality characteristic
Sample number or time

(a) Manufacturing process inputs and output

(b) Process output variability

Figure 1.2: Typical manufacturing process control [2]

These common and assignable causes of variation are illustrated in Figure 1.3. A process remains in-control until time t_1 when operating in the presence of common cause variation. The mean value (μ_0) and standard deviation (σ_0) of the system are centred around the in-control values. At time t_1, an assignable cause occurs, resulting in shifting the process mean μ_0 to the more considerable value μ_1. At time t_2, the other assignable cause appears, which leads to ($\mu = \mu_0$); however, the standard deviation of the output becomes a more significant value ($\sigma_1 > \sigma_0$). At time t_3, the next assignable cause emerges, which results in both the process mean value and the standard deviation moving to the out-of-control range. After time t_1, the existence of assignable causes leads to an out-of-control scenario. Moreover, as shown in Figure 1.3, the in-control process produces outputs that remain within the minimum and maximum control limits (LSL and USL respectively). On the other hand, the out-of-control process produces products or responses that do not comply with the desired specification limits.

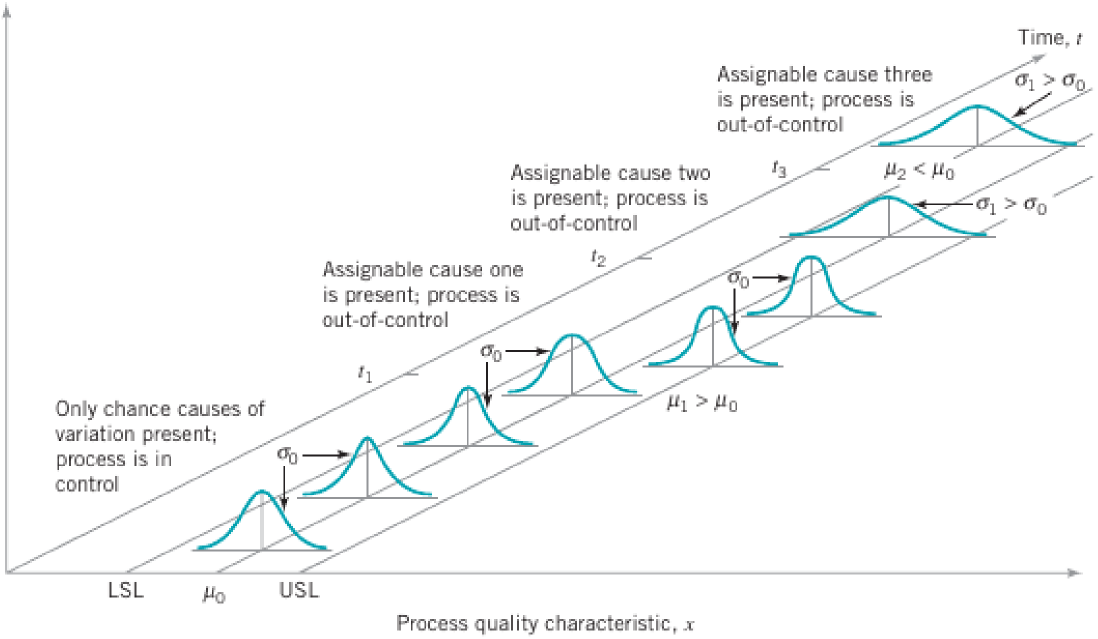

Time, t
Assignable cause three is present; process is out-of-control
$\sigma_1 > \sigma_0$
Assignable cause two is present; process is out-of-control
t_3
$\mu_2 < \mu_0$
$\sigma_1 > \sigma_0$
Assignable cause one is present; process is out-of-control
t_2
σ_0
σ_0
t_1
$\mu_1 > \mu_0$
Only chance causes of variation present; process is in control
σ_0
LSL
μ_0
USL
Process quality characteristic, x

Figure 1.3: Variation types of manufacturing process [2]

Typically, under these circumstances, statistical process control (SPC) is employed, along with other problem-solving techniques [5, 6] implemented to identify the presence of assignable causes of system variations and process investigation; this enables corrective action to be implemented before several non-conforming units are produced. The variation in the response values is usually associated with a variety of one or more process inputs or factor values. Thus, a reduction in inputs variance mitigates the variation of the response values [7]. Figure 1.4 illustrates the schematic representation of this effect. Extracting process knowledge from available measured data regarding of how and why a process behaves as it does is crucial to identifying causal relationships in the data and understanding the sources of variation and their role in process improvement. This information can then be used to enhance the process output characteristics. The work done by Batbooti [8] introduced an algorithm that investigated the optimisation of the investment casting process by applying the QCA algorithm to Swansea University's historical (nickel-based superalloy) dataset. This algorithm presumes linearity in the process data. The aim of this work is to propose a solution for non-linear datasets and discover opportunities to optimise process settings that will lead to a reduction in defects.

1.2 Investment Casting Process

The oldest known precise metal casting method, which is considered in this study, is the investment casting process, also referred to as lost-wax casting. Investment casting is used to produce components with complex-shaped that require a very smooth and detailed surface finish, tighter tolerances, and relatively thin sections such as aerospace turbine blades and automotive turbocharger wheels [9]. The investment casting process employs a series of operations, each of which involves a sub-process. The first step is the creation of wax patterns that resemble the shape of the final product using an injection mould. The wax pattern is then coated with ceramic to create disposable ceramic moulds. The moulds are gently heated to melt and drain away wax. The molten metal is then poured into the ceramic moulds. Subsequently, the ceramic moulds are removed to reveal the finished product. Finally, any excess material is removed from the finished product [9, 10]. As an example, producing a turbine blade takes a period of weeks. It often takes several days from the initial wax manufacturing phases to the final casting before the propeller is created. Typical continuous quality improvement studies incorporate more than 60 to 70 measurable procedure variables that control the quality of the finished blade or the propeller. Discussions with foundry process engineers revealed that they do try to reduce common cause variation by manually tuning the process using their domain expertise. The difficulty for foundry process designers is associated with making improvements to many system parameters (e.g., minor modifications to the operational ranges of different settings, such as alloy formulations at specific melting and pouring points, pouring temperatures, moulding criteria, etc). It is not adequate to make one single improvement at a time. It is also not straightforward for specialists to select the top crucial process factors that are seen to be accountable for the 3–5% rejection. The majority of waste produced by a foundry can be typically attributed to these factors. The schematic representation of this problem is illustrated in Figure 1.4, where the variation in the grey area is acceptable and hence considered as common cause variation. However, for processes with improved process capability, a reduction in spread or movement of target value may reduce the spread of causes. Figure 1.4 illustrates the input/output relationship of a typical single-input/single-output process. It is noticeable that as the input varies along the horizontal axis, the output varies by a corresponding amount. This variation is governed by the process itself and is represented by the blue curve in the figure, while the variance in the input value is represented by a normal distribution. The greater the spread of the input distribution, the greater the variance of the output value. Common cause variations (in grey), in this case, are substantial enough to cause the output to fall short of the process tolerance indicated by the dashed lines. The aim of this avenue of research [8, 11] is to develop an algorithm capable of confining the input parameter(s) to a narrow region (the white distribution) to guarantee that the process output remains within acceptable tolerance. The following two objectives have accordingly been pursued:

1. To quantify the effect of changes to inter-related process inputs on the variability of process output;

2. To explain this effect by quantifying causal relationships between the inputs and output(s).

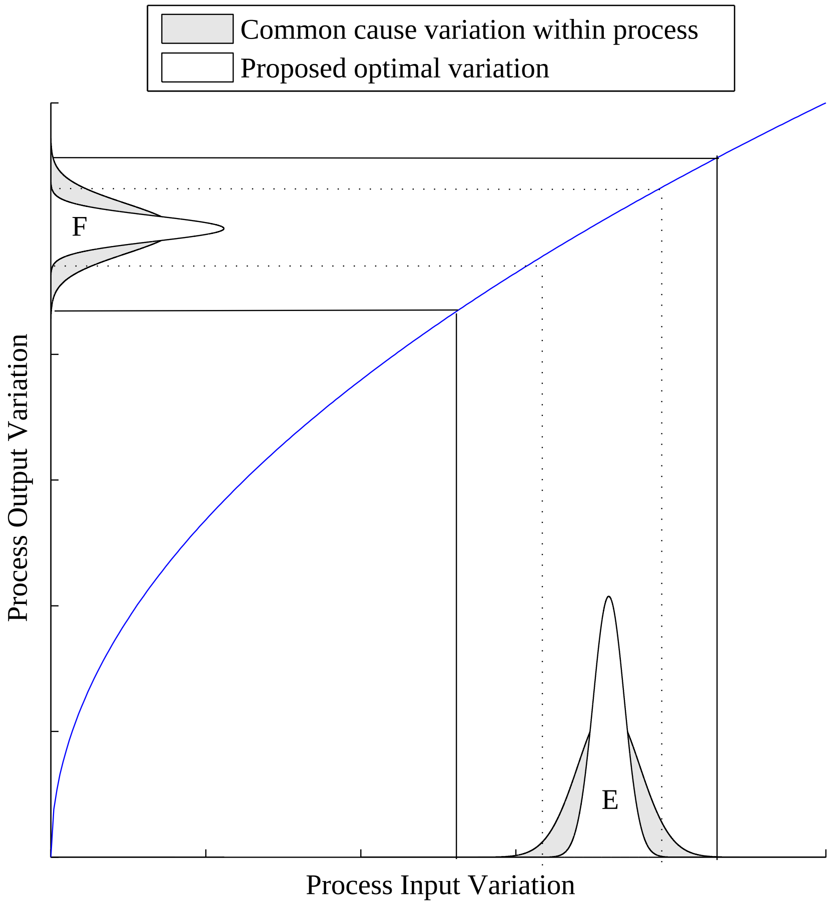


Figure 1.4: Typical common cause process revision [8]

Typically, the majority of historical manufacturing data contains incomplete observations (missing values), combines both categorical and continuous variables, and is impacted by non-linearly interacting process factors. In this study, a typical example of an investment casting foundry manufacturing nickel-based superalloy castings was utilised. The variation in a number of casts that were rejected due to shrinkage defects was observed and noted as a process response (i.e., the number of castings manufactured per fixed value of the molten metal, the rejection rate and the system parameters were observed for each melt or batch). The corresponding chemical composition readings for the batch are attached in Appendix A. The variation of the rejection rate and the variability of one process input, (e.g. factor $\%C$), is illustrated in Figure 1.5 along with the lower (LL) and upper operating range limits (UL). The USL and LSL represent the total range limits for the carbon composition, while LL and UL represent the optimal limits found in the QCA algorithm [8]. Figure 1.5 depicts the variation in the carbon composition for each batch (60 batches total) and the rejection rate variation for the complete process batches. In Figure 1.4, the variation of region 'E' corresponds to the variation of $\%C$ with lower and upper operating range limits of (LL) and (UL). The resulting variation in the output corresponds to the output region 'F'.

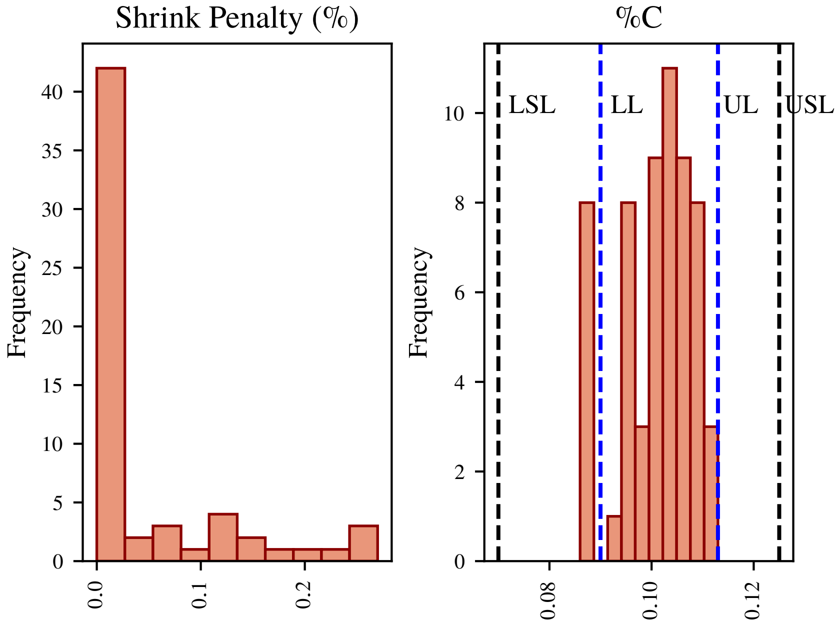


Figure 1.5: Variation in rejection rate and of the input factor %C for nickel-based superalloy [8]

To better understand the inter-dependence between the inputs, a co-linearity index (CLI) plot is constructed [11]. The co-linearity index is defined as the projection of an input variable along the response variable. The CLI plot describes the contribution of each factor to the shrinkage penalty (process output) in the form of strength and direction. Here, strength describes the weight of the factor in the process, while the direction describes the sign of its contribution to the shrinkage defect. Moreover, CLI represents the angle found between the loading vectors in the PCA subspace. This angle is defined by a set of selected principal components. On a graphical plot, the reference x-axis represents the loading vector, which corresponds to the response. The vectors of the other factors are plotted with respect to the angles and response of the loading factor. Moreover, the cosine of angles (i.e. CLI) is defined as the measure of correlation that is calculated with reference to the response direction. The factors with almost no correlations are removed, while the others are used for further consideration. This approach is considered to be an improved measure of correlation. A detailed computation of CLI is shown in the QCA algorithm [8]. The variables represented in the co-linearity plot can be categorised into different regions according to the strength of correlation, as follows:

- **No Correlation:** Central region with CLI between -0.2 and 0.2;
- **Weak Positive Correlation:** Intermediate region with CLI between 0.2 and 0.5;
- **Weak Negative Correlation:** Intermediate region with CLI between -0.5 and -0.2;
- **Strong Positive Correlation:** Extreme region with CLI between 0.5 and 1;
- **Strong Negative Correlation:** Extreme region with CLI between -1 and -0.5.

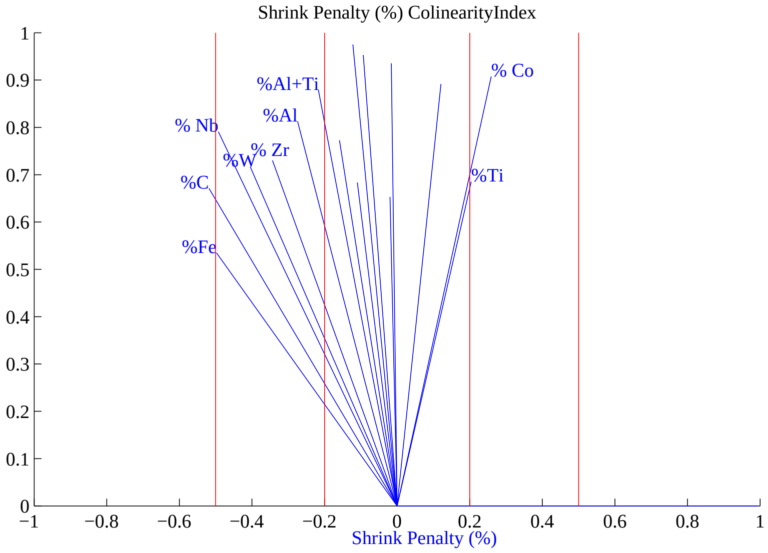
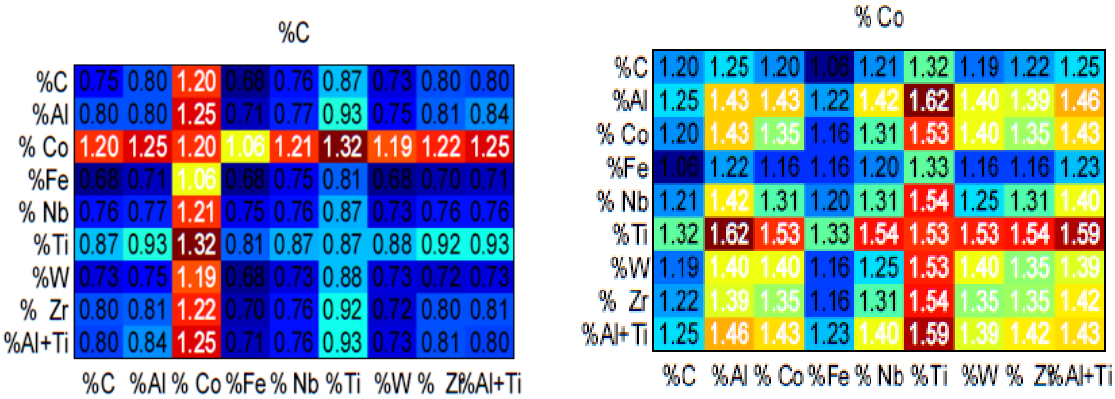


Figure 1.6: Co-linearity index plot for the process data used in nickel-based superalloy [12]

Figure 1.6 shows the CLI index plot for all process factors with reference to the normalised shrinkage penalty. The CLI index plot is used to find the strength of the correlations in the investment casting process. These correlations can be visualised by plotting the process response as a unit vector along the horizontal axis. All other factors are then represented as unit vectors with positions that are defined by the angle formed with the response variable. From Figure 1.6, it can be seen that there is no correlation in the region between shrinkage penalties -0.2% and 0.2% for $\%O$, $\%Ta$, $\%B$, $\%Ti/Ta$. Also, the factors with shrinkage penalty between 0.2% and 0.5%, and -0.5% and -0.2% exhibit a weak correlation, such as $\%Co$, $\%Ti$, $\%Al$, $\%C$, $\%Fe$. Thus, it can be concluded that there is no strong correlation between the factors in this case study.



%C

	%C	%Al	% Co	%Fe	% Nb	%Ti	%W	% Zr	%Al+Ti
%C	0.75	0.80	1.20	0.68	0.76	0.87	0.73	0.80	0.80
%Al	0.80	0.80	1.25	0.71	0.77	0.93	0.75	0.81	0.84
% Co	1.20	1.25	1.20	1.06	1.21	1.32	1.19	1.22	1.25
%Fe	0.68	0.71	1.06	0.68	0.75	0.81	0.68	0.70	0.71
% Nb	0.76	0.77	1.21	0.75	0.76	0.87	0.73	0.76	0.76
%Ti	0.87	0.93	1.32	0.81	0.87	0.87	0.88	0.92	0.93
%W	0.73	0.75	1.19	0.68	0.73	0.88	0.73	0.72	0.73
% Zr	0.80	0.81	1.22	0.70	0.76	0.92	0.72	0.80	0.81
%Al+Ti	0.80	0.84	1.25	0.71	0.76	0.93	0.73	0.81	0.80

(a) Odds ratio for interaction of $\%C$

% Co

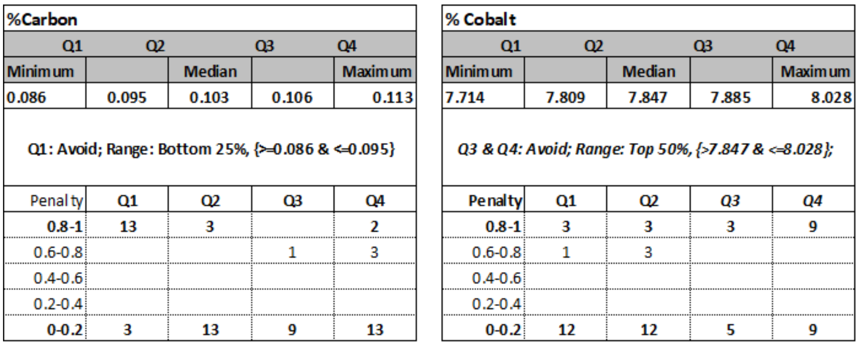
	%C	%Al	% Co	%Fe	% Nb	%Ti	%W	% Zr	%Al+Ti
%C	1.20	1.25	1.20	1.06	1.21	1.32	1.19	1.22	1.25
%Al	1.25	1.43	1.43	1.22	1.42	1.62	1.40	1.39	1.46
% Co	1.20	1.43	1.35	1.16	1.31	1.53	1.40	1.35	1.43
%Fe	1.06	1.22	1.16	1.16	1.20	1.33	1.16	1.16	1.23
% Nb	1.21	1.42	1.31	1.20	1.31	1.54	1.25	1.31	1.40
%Ti	1.32	1.62	1.53	1.33	1.54	1.53	1.53	1.54	1.59
%W	1.19	1.40	1.40	1.16	1.25	1.53	1.40	1.35	1.39
% Zr	1.22	1.39	1.35	1.16	1.31	1.54	1.35	1.35	1.42
%Al+Ti	1.25	1.46	1.43	1.23	1.40	1.59	1.39	1.42	1.43

(b) Odds ratio for interaction of $\%Co$

Figure 1.7: Odds ratio for interaction of factors [8]

Figure 1.7a above shows the prediction of the odds ratio of the interaction of $\%C$ with every two factors in the process. The odds ratio is a measurement of the relationship between the probability of the occurrence of desired and undesired process output values (outcomes). It can be used to predict the outcome with respect to an input factor. Another important application of the odds ratio is evaluating the behaviour of a known operating limits range by approximating the values of the related response. The computation of the odds ratio is explained in more detail in chapter five. The Bootstrap method is a statistical procedure that makes use of a resampling technique to estimate quantities of a population by means of random sampling of a single dataset with replacement. The Bootstrap method creates several random subsets from a single dataset. This technique generates Bootstrap samples from an initial dataset by randomly drawing with replacement. Moreover, the Bootstrap technique enables the calculation of standard errors and hypothesis testing from sample statistics. In machine learning, the Bootstrap method is used to evaluate the capability of a machine learning model when making predictions on data that is not included in the training data.

Batbooti [8] extended Gianetti's work [11] and found an interaction among the process factors. For example, based on Batbooti's analysis [8], the odds ratio of 1.32 in Figure 1.7a represents the effect of the interaction between $\%C$, $\%Ti$ and $\%Co$. In other words, the values for these factors were bootstrapped from the optimal limits; however, the values for the remaining factors were bootstrapped from the original process range. The interaction table reveals that if the values for factor $\%C$ are bootstrapped from its optimal range limit, the resulting odds ratio is 0.75; however, the $\%Co$ factor exhibits a higher odds ratio at 1.35 [8]. Although both $\%C$ and $\%Co$ have similar strength co-linearity indices (weak correlation), as shown in Figure 1.6, the corresponding odds ratio values are significantly different. This is likely because of the linear assumption of the quality correlation algorithm (QCA) prediction model [8]. Generally speaking, the QCA predictive algorithm underestimates odds ratio values for $\%C$. Based on the observations emerging from the research, Batbooti [8] recommended that a non-linear approach be adopted to try to capture the inputs/output relationship of the non-linear process at hand. It was concluded that the linearity assumption was the main point of weakness of Batbooti's proposed algorithm [8].



%Carbon				
Q1	Q2		Q3	Q4
Minimum		Median		Maximum
0.086	0.095	0.103	0.106	0.113
Q1: Avoid; Range: Bottom 25%, {>=0.086 & <=0.095}				
Penalty	Q1	Q2	Q3	Q4
0.8-1	13	3		2
0.6-0.8			1	3
0.4-0.6				
0.2-0.4				
0-0.2	3	13	9	13

% Cobalt				
Q1	Q2		Q3	Q4
Minimum		Median		Maximum
7.714	7.809	7.847	7.885	8.028
Q3 & Q4: Avoid; Range: Top 50%, {>7.847 & <=8.028};				
Penalty	Q1	Q2	*Q3*	*Q4*
0.8-1	3	3	3	9
0.6-0.8	1	3		
0.4-0.6				
0.2-0.4				
0-0.2	12	12	5	9

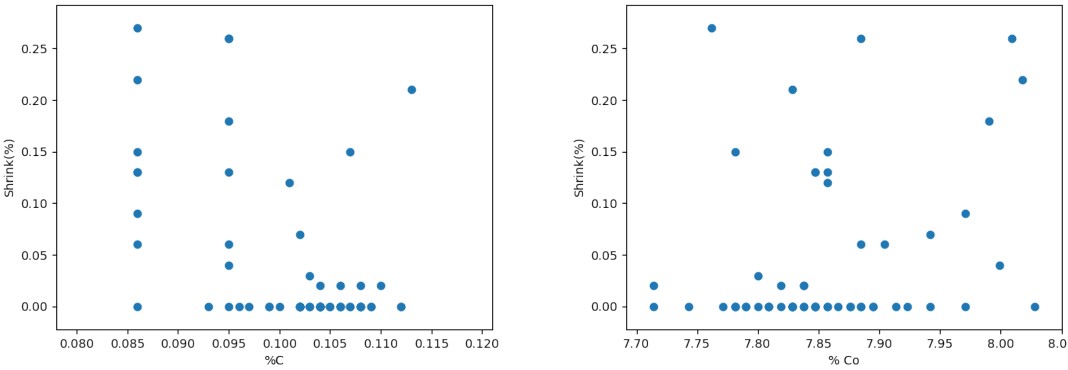
Figure 1.8: Penalty matrices for carbon and cobalt composition [13]

Another method of discovering the non-linearity in the dataset involves studying the effect introduced by Gianetti et al. [13]. Figure 1.8 depicts a method that can be used to standardise the response in order to determine the correlation between the input and the output [13]. With reference to its carbon composition, the variation in the batches is first sorted and then divided into four quartiles. For each quartile, the response penalty is normalised and further divided into five regions with an increment of 20% for each region, starting from zero until 100% penalty. Subsequently, each batch is assigned to a quartile based on its appearance limit. Finally, the table is constructed for each process factor using the same criterion. According to Gianetti [11], a linear relationship between the process factor and the response is defined as follows: If the bottom 50 percentile (Q1 and Q2) are defined as optimal, this means that the batches are within a penalty range between 0-40%, and the top 50 percentile (Q3 and Q4) are avoidance, as this indicates that the batches are in a penalised range of between 60-100%; and vice versa. Whereas, non-linearity is defined as a situation in which the bottom 25 percentile (Q1) or top 25 percentile (Q4) is avoidance; which means that the batches are within the penalised range between 0-40% and 60-100%, respectively. The same is true if the bottom 25 percentile (Q1), top 25 percentile (Q4), either/or middle 50 percentile (Q2 and Q3) are optimal, which means that the batches between 0-40%, 40-60%, or 60-100%, respectively are within the penalised range. These cases are shown in Figure 1.8. To sum up, linearity and non-linearity can be defined based on the following cases [11]:

1. **Non-Linearity**: From cases 1 and 2, a non-linear relationship is observed between process input and response. These relationships exhibit step changes and are, hence, non-linear.
    - **Case 1**: If the top quartile, bottom quartile or middle two quartiles are 'optimal' limits.
    - **Case 2**: If the bottom quartile or top quartile are 'avoidance' limits (undesirable).
2. **Linearity**: From cases 1 and 2, a linear relationship is observed between process input and response. These relationships have exhibited pattern variations and are, hence, linear.
    - **Case 1**: If the bottom two quartiles are 'optimal' limits and top two quartiles are 'avoidance' limits.
    - **Case 2**: If the bottom two quartiles are 'avoidance' limits and top two quartiles are 'optimal' limits.

Figure 1.8 presents the penalty matrices for both carbon $\%C$ and cobalt $\%Co$. It can be observed that carbon composition $\%C$ in the first quartile (bottom 25 percentile) is considered the avoidance limit since it has 13 observations correlated with high penalty values and three observations correlated with low penalty values. Whereas, the rest of the quartiles (second, third, and fourth) exhibit similar process performance in the way they are correlated with low penalty values. From the cases mentioned earlier, it can be observed that $\%C$ has a non-linear relationship since it is categorised as a case 2; this is, the avoidance limit lies within the bottom quartile. There is also a step change in the relationships when the $\%C$ is in the first quartile compared with the other quartiles. This problem has also been observed in Figure 1.9a, which presents a plot between $\%C$ and shrinkage defects, where it can be seen that for carbon values less than 0.095% the variability in defect values is high. On the other hand, for cobalt composition $\%Co$, it can be seen that the first and second quartiles exhibit similar process performance with an optimal range located in the bottom two quartiles since it has 12 observations correlated with low penalty values. Whereas, the third and fourth quartiles (top 50 percentile) are considered avoidance limits, since they are correlated with higher penalty values. Based on the abovementioned cases, it can be observed that

$\%Co$ exhibits a linear relationship since it is categorised as case 1. Thus, the bottom and top two quartiles are optimal and avoidance limits, respectively. There is also a variation in the range of $\%Co$ since it varies from low penalty values (first and second quartiles) to high penalty values (third and fourth quartiles). This variation is also illustrated in Figure 1.9b, which presents a plot between $\%Co$ and shrinkage.



(a) Carbon variation ($\%C$)

(b) Cobalt variation ($\%Co$)

Figure 1.9: The variability in shrinkage defect values across batches is shown with reference to a sample of both carbon and cobalt composition values for the batch

An example of the linear/non-linear variation is provided as follows. A linear correlation is characterised by an input/output relationship that is either monotonically increasing or decreasing. A non-linear interaction is characterised by having maxima, minima, or step changes in its process response versus input curve (Figure 1.9a). The input data at 0.085 and 0.095 exhibit high variability in their output response, which leads to an indication of non-linearity in the dataset studied. It can also be seen that there are multiple batches with defect values spread over a wide range for a single sample carbon composition value. There is a step change in the behaviour, with much smaller spread at higher composition values. This may suggest that the carbon's effect on the process output is not independent of other input parameters. By contrast, from Figure 1.9b, a pattern can be observed in the relationship between cobalt composition and shrinkage defect, which indicates linearity in the dataset being studied. Furthermore, in the previous work illustrated by Batbooti [8], it was described that both $\%C$ and $\%Co$ have the same strength with different interactions and it was concluded that the variation in carbon is non-linear. Moreover, Gianetti et al. [13] show that a non-linear relationship exists between carbon composition and shrinkage defect. Based on Figure 1.9, the previous work done by Batbooti [8], and the work of Gianetti et al. [13], it can be seen that a non-linear relationship exists between carbon composition and shrinkage, resulting in a non-linear problem to solve. Typically, historical manufacturing data is not widely available, as the observations are usually small in number and include missing data. Generally speaking, a statistical learning method requires sufficient data if it is to capture statistical knowledge from the historical data. Moreover, batches also take time to be produce (around 1 - 2 months); in addition, the process engineer makes decisions based on the amount of data available, and the decision made based on this limited data might not be accurate. Furthermore, a small dataset lowers both prediction performance and statistical power, and also increases overfitting, resulting in low prediction accuracy and weaker causal relationship discovery. This issue can be resolved by augmenting the model with new data. In chapter four, a new method is discussed and performed to augment the data to a sufficient extent. Another problem that the process engineer may face regarding the dataset is that of skewed data. As skewed data often increases the tendency of bias error, balancing this skewed data is an important step toward minimising bias error in the prediction model. Chapter three discusses the ways in which skewed data can be balanced. In the previous work conducted by Batbooti [8], the prediction performance of missing data was based on a linearity assumption. The results of linear prediction are affected by several limitations (see Figure 1.10), as follows:

1. Altering the effect of correlation for the non-linear interacted factor. This was shown by Batbooti [8] since the carbon composition correlation was changed after the prediction from being strongly to weakly correlated with the shrinkage defect.

2. The prediction accuracy performance will decrease dramatically after a certain level of missing data.

3. Computational cost is high, since it is an iterative method.

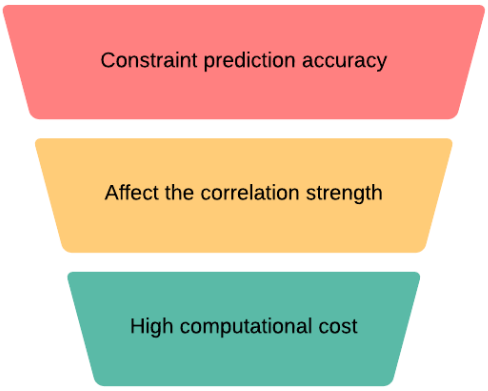

Constraint prediction accuracy
Affect the correlation strength
High computational cost

Figure 1.10: Limitations of linear prediction

From the above, it can be concluded that predictions based on the linear assumption are incapable of extracting statistical knowledge from non-linear interacted data. Moreover, this limitation will be propagated to both the extracted causal information and the optimal process range. The overall aim of this thesis was the same as that of Gianetti's [11] and Batbooti's [8] theses; however, the precise objective of this thesis is to overcome the limitations of Batbooti's [8] work. This is described in the next section.

1.3 Aims and Objectives

Typical historical foundry process data tend to be mixed, limited to a small number of observations (batchwise observations per part number are limited), and affected by missing values. The challenge of the present study is to create a data-based framework for variation reduction that can (i) diagnose the process variation; (ii) determine opportunities of optimal and avoidance operating range limits; and (iii) predict the process performance based on the optimal operating scales of the inputs. This thesis aims to develop a causal knowledge discovery model in order to find the optimal operating ranges for continuous process parameters and/or optimal categories for discrete process parameters that achieve the desired variation reduction in output with minimal interaction from domain expertise. This is achieved by (i) constraining the operating ranges of selected continuous process parameters within specified tolerance limits so that the spread is reduced, or changing the target value for process inputs; (ii) choosing particular categories of discrete process parameters; or (iii) choosing combinations of both types of process parameters. This leads to the following set of objectives:

- Development of a predictive machine learning model that has the capability of dealing with non-linearity in data, as described in Section 1.2.
- Development of a non-linear missing data algorithm to impute the missing values of both quantitative and categorical variables.
- Development of an augmentation technique to deal with a skewed and limited dataset.
- Demonstration of the ability to predict the output or response of any given choice of operating limits on selected mixed input factors.
- Development of a non-linear approach for defining the most critical process inputs.
- Verification of the performance efficiency of the proposed algorithm on published datasets and conduct comparisons with a number of published state-of-the-art models.

The following four novel techniques (as depicted in Figure 1.11) are proposed in this thesis to enhance the prediction ability of small non-linear datasets via improved computational efficiency and to enhance the interpretability of the traditional Random Forest algorithm. These novel techniques are as follows:

1. A novel automatic forest size optimisation. This algorithm maximises the inherited gains due to the use of ensemble uncorrelated multiple decision trees.

2. A novel iterative Random Forest training approach used for augmentation, balancing, and imputing missing values for non-linear mixed datasets.

3. A novel causal framework for Random Forest. This algorithm searches for the optimal decision path to facilitate interpretation of traditional Random Forest and provides a comprehensive inference analysis framework to enable causal knowledge extraction from any complicated process.This framework is designed to guide process engineers to optimise process performance through the estimation of optimal limits for each process factor that will minimise the process defects and maximise yield.

4. A scoring algorithm for ranking the most critical process factor based on the Decision Path Search algorithm.

The detailed development process of the first two novel techniques, which involve implementing the Modified Random Forest algorithm, are discussed in chapter four, while further details of the third and fourth novel techniques are illustrated in chapter five.

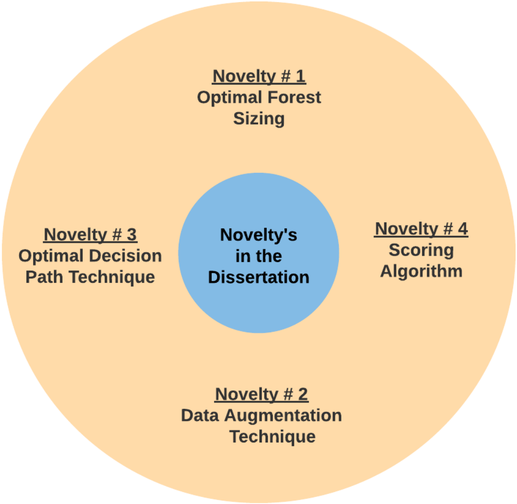

Novelty # 1
Optimal Forest Sizing
Novelty # 3
Optimal Decision Path Technique
Novelty's in the Dissertation
Novelty # 4
Scoring Algorithm
Novelty # 2
Data Augmentation Technique

Figure 1.11: Dissertation novelty techniques

1.4 Publishable Results for Potential Journal Papers

This section introduces the results that could potentially be published in future journal articles regarding the novel techniques identified in this research. These proposed articles can be summarised as follows.

1. "A Novel Approach to Industrial Process Optimisation Using a Modified Random Forest Algorithm."

    - Historical process data rarely deals with small datasets for quick observations and results. One of the key drawbacks of most non-linear machine learning algorithms is the inability to deal with a small dataset. Accordingly, the author proposes a novel approach combining both SMOTE [14] and missForest [15] techniques to resolve the issue of the small dataset and preserve the overall structure without affecting the factor correlations. This novel approach comprises the following steps: First, quantify the data into four different quartiles with a balanced dataset in each quartile so that SMOTE can be applied. Second, augment the dataset using SMOTE to achieve the optimal number of datasets that Random Forest can deal with and to preserve the overall structure of the data. Finally, use the missForest technique to predict the new response for the new observations in order to preserve the non-linearity relationships between the factors and the response (s). A novel approach is derived that involves combining both SMOTE and missForest in order to augment non-linear regression problems, enabling it, to deal with non-linear unbalanced datasets. In addition, Random Forest (RF) has an issue with choosing the optimal number of trees. Some hyper-parameter tuning tuning will thus be conducted to obtain the optimal number of trees that will lead to both lower computational cost improved and variance reduction. The idea is to determine the optimal number of trees that will obtain the most robust performance. Even if this number of trees is exceeded, there will be no change in performance results. This technique was accomplished by using the cross-validation method to sweep over a number of different trees in order to find the optimal estimator that will further minimise overfitting. The author proposes a novel technique for finding the optimal number of estimators for industrial process optimisation using a Modified Random Forest (MRF) approach. This paper is further discussed in chapter four.

2. "Causal Knowledge Extraction for Industrial Processes Using Decision Path Search Algorithm."

- Random forest is a non-parametric algorithm, which leads to difficulty in interpreting the ensuing decisions. The built-in function in the traditional algorithm is used to score the factors based on their importance, regardless of whether these factors are affected by outliers or noise. Accordingly, the author proposes a novel approach to discover the causal knowledge extraction by searching and tracing the decision trees all the way to reach the base estimators by traversing the tree. The proposed technique is used to define the optimal and avoidance limits for every factor in the dataset and to determine how important this factor is to the given response. This novel approach is called a Decision Path Search algorithm (DPS). The paper is discussed in more detail in chapter five.

3. "Robustness of the Modified Random Forest Algorithm."

    - In this paper, the Modified Random Forest algorithm is compared with a variety of published state-of-the-art models to prove the stability and robustness of the proposed algorithm, as well as to demonstrate the applicability of the MRF algorithms. Due to the novelty of the proposed algorithm, it can achieve superior performance relative to other state-of-the-art models. In the first stage of building the MRF algorithm, it was developed only for the foundry problem, after which it was used as a general algorithm for a wide variety of further problems that creates impact beyond foundry problems. The proposed algorithm performs better across the board for foundry-related and other problems; thus, the findings from the proposed algorithm are not only limited to the small subset of foundry problems but are generally applicable on a broad scale. This bench-marking highlights the main areas of weakness areas of the traditional Random Forest. Based on this, more work was carried out to enhance the base-line algorithm in terms of its robustness and its stability (even with large datasets). Comparisons with different state-of-the-art models and their results are presented in chapter six.

1.5 Knowledge Discovery and Variation Reduction Techniques Roadmap

Figure 1.12 depicts the roadmap of the various reduction techniques within the research group at Swansea University. It begins with a Penalty Matrix approach (PM) [16] moving on to other techniques while continuously enhancing the PM approach until the proposed algorithm is reached, which is a Modified Random Forest algorithm (MRF). The various reduction techniques shown in Figure 1.12 will be discussed in more detail in the literature review chapter.

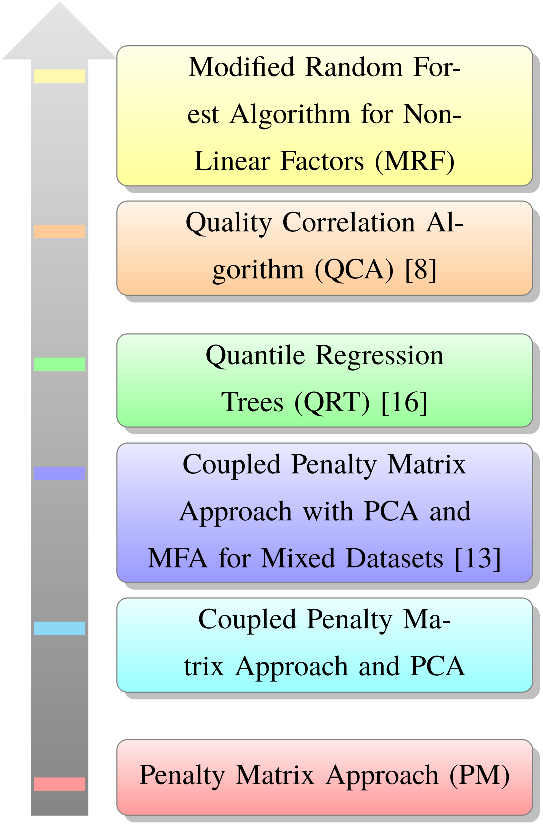

Modified Random Forest Algorithm for Non-Linear Factors (MRF)
Quality Correlation Algorithm (QCA) [8]
Quantile Regression Trees (QRT) [16]
Coupled Penalty Matrix Approach with PCA and MFA for Mixed Datasets [13]
Coupled Penalty Matrix Approach and PCA
Penalty Matrix Approach (PM)

Figure 1.12: Various reduction technique within the research group at Swansea University

1.6 Thesis Layout

This thesis describes the main research activities carried out as part of this PhD study. The thesis structure, which is illustrated in Figure 1.13, is as follows.

- **Chapter 2:** Introduces and discusses the main body of literature concerning statistical learning models, tree-based algorithms, and missing data algorithms. This chapter reviews the traditional and recent multivariate data analysis approaches to manufacturing output variation monitoring, diagnosis, and analysis.
- **Chapter 3:** Introduces the mathematical formulation of the predictive causal analytics, along with the state-of-the-art algorithm that is utilised in this thesis. This chapter lays the foundation for the novel work introduced in chapter four.

- **Chapter 4:** Introduces a predictive machine learning model based on the Random Forest algorithm, which was first developed by Leo Breiman [17]. The proposed algorithm solves the knowledge discovery issue from the non-linear relationship between the process factors, then adds this step to the research done by Batbooti [8]. The Modified Random Forest algorithm (MRF) is subsequently applied to the nickel-based superalloy dataset. This chapter demonstrates the algorithm's capability to predict process responses, even with non-linear interactions among process factors.

- **Chapter 5:** Presents a new variation reduction algorithm based on the graph search algorithm. This approach enhances the interpretability of the Random Forest algorithm (RF) and adds a causal relationship to discover the factor ranges corresponding to variation in the output, depending on the decision path in each decision tree in the forest. The non-linear optimal process limits of the algorithm are demonstrated on the nickel-based superalloy dataset. This chapter demonstrates the algorithm's ability to estimate the optimal and avoidance limits for the most critical factors.

- **Chapter 6:** Three publicly available datasets are studied to demonstrate the applicability of the proposed algorithms. In all three case studies, the optimal and avoidance variation of the factors were identified. The process improvement, as suggested by the optimal response values, was predicted for single and interaction factor effects. The algorithm's ability to discover new knowledge, visualise the data, and predict the behaviour of the data within specific limits is compared overall with recent studies that have been conducted on these datasets. The regression capability of the proposed algorithm is also compared with the published results. The novelty of the proposed RF approach is subsequently illustrated by comparing the MRF algorithm with three published RF models. This comparison reveals how MRF's features make it superior to other RF approaches found in the literature.

- **Chapter 7:** Summarises the main conclusions and research outcomes of this PhD work and identifies the scope for future work.

- **Appendix:** Presents both the nickel-based superalloy dataset studied in this thesis and the optimal number of trees for various published datasets. The plots presented in [8] summarising previous work on this were reproduced in Python and are also included in the Appendix.

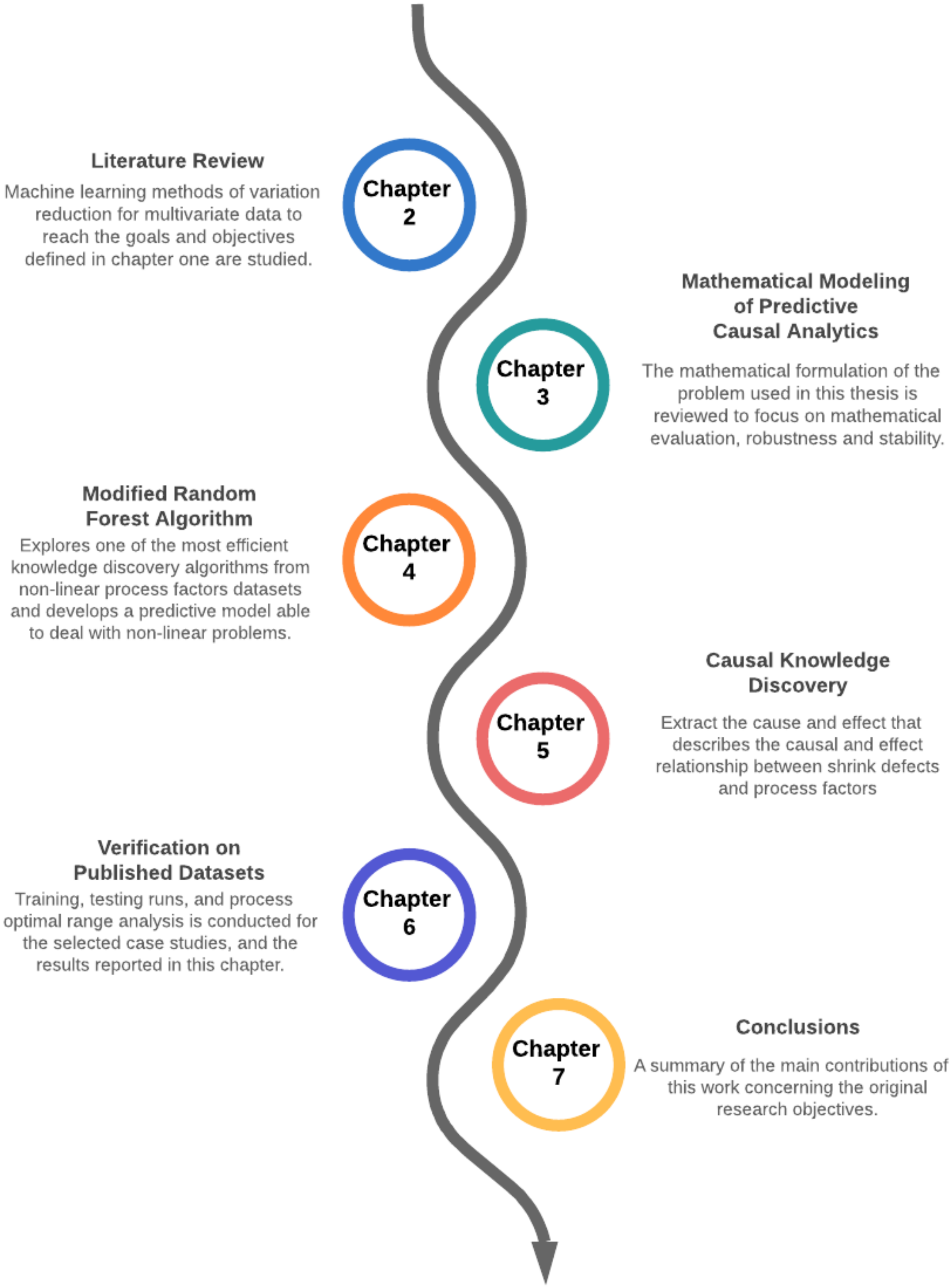

Literature Review
Machine learning methods of variation reduction for multivariate data to reach the goals and objectives defined in chapter one are studied.
Chapter 2
Mathematical Modeling of Predictive Causal Analytics
The mathematical formulation of the problem used in this thesis is reviewed to focus on mathematical evaluation, robustness and stability.
Chapter 3
Modified Random Forest Algorithm
Explores one of the most efficient knowledge discovery algorithms from non-linear process factors datasets and develops a predictive model able to deal with non-linear problems.
Chapter 4
Causal Knowledge Discovery
Extract the cause and effect that describes the causal and effect relationship between shrink defects and process factors
Chapter 5
Verification on Published Datasets
Training, testing runs, and process optimal range analysis is conducted for the selected case studies, and the results reported in this chapter.
Chapter 6
Conclusions
A summary of the main contributions of this work concerning the original research objectives.
Chapter 7

Figure 1.13: Dissertation road-plan and chapter outline

Chapter 2

Literature Review

2.1 Introduction

In several fields of technology, humanities, and the sciences, experts have employed past measurements or observations to predict future phenomena. For many years, experts have approached issues including the devising of analytic frameworks, particularly those based on fundamental principles or that combine knowledge to understand, model, and analyse the problem of interest. However, the limitations of conventional approaches begin to be revealed as the difficulty of these problems increases. Therefore, machines of increasing capacity and speed have been designed to break this cognitive barrier as well as advance the state of the scientific field. Consequently, many algorithms and techniques (powerful enough to deal with large and complex datasets) have originated from the field of machine learning. Notably, data is the 'magic word' for every decision-maker across all industries. Nevertheless, data is meaningless in the absence of a proper interpretation of its meaning. In the modern world, a number of tools are being developed to allow perceiving, analysing, and drawing conclusions from data, enabling its conversion into useful knowledge. In this chapter, machine learning methods of variation reduction for multivariate data are reviewed to meet the goals and objectives outlined in chapter one. This review includes a discussion of the advantages and disadvantages of the different methods, along with the reasons for selecting the Random Forest algorithm to discover the process knowledge from the historical data. The remainder of this chapter is structured as follows. Section 2.2 addresses different methods of machine learning and process knowledge discovery. Section 2.3 summarises the most common statistical learning models. An introduction to the Random Forest algorithm is presented in Section 2.4, while Section 2.5 expounds on the merits of this algorithm. In Section 2.6, the drawbacks of Random Forest are addressed. Section 2.7 then defines the multivariate feature imputation. Finally, Section 2.8 summarises the challenges this approach will encounter in the process of solving the problem.

2.2 Statistical Learning Methods for Multivariate Data

In this section, the previous statistical learning methods for multivariate data are addressed. The section is structured so that the limitations of each method are presented, followed by the contributions this thesis makes to overcoming these limitations. It further outlines the scope of each approach and discusses how those methods are unable to handle the challenges posed by the problem investigated.

Hotelling's T^2

Typically, in the SPC framework, Hotelling's T^2 statistics [18] are used to detect the occurrence of assignable causes. Process variables and key product characteristics (KPCs) are monitored, and the covariance matrix is estimated from common cause variation data; subsequently, T^2 statistics are applied to detect whether or not new process observations lie within the acceptable limit. However, this approach does not provide any diagnoses to identify the root cause variables. To improve the diagnosability of Hotelling's T^2, Mason, Tracy and Young (MTY) [19] proposed decomposing the T^2 into independent terms (conditioned and unconditioned), then charting the unconditioned terms individually (in a similar fashion to a Univariate Shewhart Control Chart [20]) in order to identify the factors that have significant a impact on an individual cause T^2 signal. In terms of the conditional part, it accounts for the contribution of different combinations of process variables to KPCs. However, it does not provide a detailed causal relationship to distinguish the root cause of the failure, as it focuses only on the occurrence of the defect; moreover, it is not computationally efficient, especially given its high number of process inputs [21]. Motivated by a desire to reduce computational cost and improve the diagnosability, a Bayesian-based T^2 approach was proposed [21] - also referred to as causation-based T^2 decomposition - that operates by integrating process knowledge with statistical data analysis. The limitations of this approach are that it requires heavy domain expertise to build the model and is also computationally expensive.

Principal Component Analysis (PCA) and Multiple Factor Analysis (MFA)

PCA is a multivariate data analysis technique designed to extract the most critical information from the data by projecting a series of observations of variables that are likely to be correlated on a subspace of linearly uncorrelated orthogonal variables, known as principal components. It provides a reduced-dimensional space that minimises the information loss. PCA is widely used in different areas of science, including computer vision [22], quality and process control [23], and pattern recognition applications with different types of analysis [24]. A multivariate computational (statistical) cycle control approach is suggested, which operates by splitting the initial data space into four different sub-spaces. The reduction in the dimensionality of variables in each subspace is focused on the principal component analysis and Bayesian inference (BI) [25]. At the PCA decomposition stage, each variable is divided into two significant sub-spaces: the Principal Component Subspace (PCS) and the Residual Subspace (RS). The squared Mahalanobis distance is used to assess the uncertainty within the PCS, while the squared Euclidean distance i used to calculate the RS variations. Each variable is considered as either relevant or irrelevant within each main subspace based on a binary numbering system. Multiple correlation coefficients are used in this work to determine the relevance between variables in PCS or RS. The BI is applied to divided samples based on conditional probability to infer the state of a new observation relative to the fault region defined by PCS and RS. Lin et al. [26] proposed a condition-based maintenance model that employs a Proportional Hazards Model (PHM) [27] and principal component analysis. The PCA was utilised in this model to reduce the number of original variables, as well as to decorrelate the linear set of random variables. In this work, the Principal Component of Proportional Hazard Regression (PCPHR) method is introduced, which uses a set of principal components as covariates in PHM to reduce the problem's complexity. Parameter estimation for the hazard model is achieved by maximising the likelihood function. A dataset of seventy data points was used: Forty-two of these data points failed, while twenty-eight led to a suspension. The scores in the lower subspace that were a result of the PCA or Partial Least Squares (PLS) formed the basis for the T^2 statistics [23]. The plot of scores for the first three principal components of the refinery's historical data [28] were used to cluster the data into five groups. The inference in the lower subspace provides another approach to studying the main causes of variation with greater ability from the classical T^2 approach. The squared prediction error (SPE) calculated during the PCA reconstruction method is referred to in the literature as $Q_{statistics}$. The T^2 and $Q_{statistics}$ are then used with PCA to highlight outliers in the data, such that the statistics are calculated and the points located outside the confidence interval of each statistic were identified as outliers [29–31]. When T^2 and $Q_{statistics}$ are used together in one plot, this is referred to as a leverage outliers test [31]. Another PCA-based procedure, the confidence intervals of the scores, was reviewed by Thennadil et al. [31] and Daszykowski et al. [32]. The concept of the squared prediction error was used to identify the variation in the new observations, as well as to estimate the corresponding contribution of the variables to this variation. It should be noted, however, that the methods described in this review were not designed to identify optimal and avoidance limits within the common cause variation to relate the process variables operating within new limits with KPCs. Instead, these methods were used to reduce the dimensions of the highly dimensional dataset (i.e. dimensional reduction methods). The number of projection axes (orthogonal) should be predefined before the dimensional reduction process begins.

The multiple factor analysis (MFA) methods [33] seek to compare a different group of variables by adapting the categorical variables according to multiple corresponding analysis (MCA), which is a counterpart of PCA for categorical data [22]. The MCA uses the Chi-Squared distance measure instead of the Euclidean distance used in PCA [22]; while the Chi-Squared distance is similar to Euclidean distance, it is weighted by the inverse of the sum of individuals. Giannetti et al. [5] used multiple factor analysis to extend the work presented in [13]. The proposed method integrates three data tables, specifically, quantitative process variables, categorical variables and the process responses. Data pre-processing is introduced to adapt the input variables and responses to plot the CLI for mixed data. The median and interquartile range (IQR) transformation is introduced to stabilise the standard deviation for variables with a Gaussian distribution and amplifies the standard deviation of those with skewed distribution. Each categorical variable is transformed into uncorrelated variables with a value of ONE when the corresponding category has occurred and ZERO otherwise. The main limitation of this approach is that it is sensitive to outliers and factor scaling, meaning that the data needs to be pre-processed. This approach also assumes that a linear relationship exists among the variables and is further unable to solve complex problems in which the majority of the variation is explained by more than the first two-thirds of PCs.

Stream-of-Variation Analysis (SOV)

Stream of Variation (SOV) [34] is a general mathematics-based procedure used for variation propagation analysis in multi-stage manufacturing processes. The main concept concerns the integration of multivariate statistics with control theory, along with design and manufacturing backgrounds. Ceglarek et al. [24] developed a novel approach based on the first-time-right $(FTR)^{design/diagnosis}$ approach for product/process performance analysis. In the design phase, FTR^{design} and SOV were used for analysis, prediction and optimisation. SOV simulates the variation propagation throughout the process and predicts the dimensional variation of the final product. In the production phase, $FTR^{diagnosis}$ and SOV provide a robust method for finding and segregating the underlying causes of dimensional variation by identifying the most significant dimensional defects. Liu [35] compared the SOV and statistical process control (SPC) approaches in terms of their variation propagation modelling, process monitoring, and diagnostic ability. In SOV modelling, the state transition equation accounts for the change of sources in a manufacturing stage that may cause variation in product characteristics during the same phase. It also incorporates the effect of upstream stage variation on the product, along with the effect of unmodelled variations. The observation equation used to model key product characteristic measurements assumes a linear dependence on product and process design information. In the SPC modelling, moreover, the model is built using KPC measurements rather than physical engineering knowledge pertaining to product and process design information.

Comparison of the results revealed that the SPC method was capable of identifying any faults that arose during the different stages, while SOV identified the root causes in the process. SOVs were used for the dimensional variation accumulated by the multi-stage manufacturing process data rather than the response variables that create categories where lower or higher values are associated with better optimisation function. The SOV model is based on a linear state-space model and is limited to dimensional and geometric variation, while the SPC model can be used for linear state transitions. The SOV assumes a linear transition model for the process phases and further requires domain expertise.

Fuzzy Data Analysis (FDA)

Shu and Wu [36] presented a fuzzy data analysis to estimate process yield as a quantification index of manufacturing process performance. The manufacturing of the touchscreen case presented in this work specified tolerance depending on the light-transmission rate. Youn et al. [37] developed a model that predicts the life of power generator stator bars under moisture absorption conditions resulting from water cooling. The presented work was a diagnostic and prognostic statistical model using directional Mahalanobis distance as an assessment of the health condition of these bars. Adopting the Mahalanobis distance approach enabled the high-dimensional dataset to be reduced to one dimension after normalisation was combined with a statistical model to discover the correlation between variables. The limitations of the fuzzy dataset are that it is restricted to a limited number of input variables and also requires a lot of data and expertise to develop a fuzzy system. Moreover, the dimensional reduction utilised leads to a linearisation of factor interaction.

Bayesian Belief Network (BN)

Yang and Lee [38] suggested applying the Bayesian belief network (BN) to investigate the qualitative and quantitative causal relationships between process variables, along with their effect on quality, by constructing BN models at various stages of the process. When a fault occurs, the measurement is given as proof in the BN model, and the backward likelihood of every sensor can be used to indicate the root cause of the problem. However, this approach requires a significant amount of technical expertise (domain knowledge) to define some configuration settings during the training process.

Recognising the features of the mechanism requires a highly resource-demanding algorithm with intensive training periods. Liu and Jin [39] devised a new Bayesian Networks (BN) modelling approach for assembly process fault diagnostics that considers small datasets. The network is constrained to two layers: The top layer contains the root nodes of the fixture variables, while the second layer represents the assembly variation sensor locations. Finite Element Analysis (FEA) was used to relate the causes of variance to the deformable component assembly variations. The critical causal relationship implemented was focuses on the visualisation of the sensitivity matrix of variance. The results for this relationship results were observed based on the testing of conditional mutual knowledge. If the system architecture has already been established, the Bayesian method can be used to achieve the conditional probability tables by integrating the previous probability distributions. Bayesian multivariate statistical inference approaches have been evaluated by Brown et al. [40], while Vehtari and Ojanen [41] have been addressed Bayesian-based prediction models.

Dey and Stori [42] developed a Bayesian belief network approach to monitor and diagnose the process incorporating sequential machine operations. The data on subsequent machining procedures collected from various sensors were then mixed through a causal belief network framework to supply a probabilistic diagnosis of the root cause of the process variation. A Bayesian belief network created using Bayes' theorem and based on the prior probability of observed data was used to determine the causal relationship between variables. Conditional probability values were obtained through maximisation of the likelihood function, which represented the data learning step following network initiation. The message transmission methodology developed in this work for Bayesian inference was based on sensor observations; in this process, each node received messages from its parents and transmitted these letters to its children. Data from two sensors (i.e. spindle power and acoustic emission) and two processes (i.e. face milling and drilling) was studied, with the obtained results showing that the belief network was capable of accurately diagnosing of the state of the drill tool wear at the 80% confidence level (10 to 18 trials). The limitations of the Bayesian network included the need for good domain expertise and its high computational cost.

Support Vector Machine (SVM)

A support vector machine (SVM) can be defined as a supervised machine learning algorithm that can be used for the purposes of both regression and classification. Specifically, SVMs are largely employed in problems of classification [43, 44]. In essence, they operate by finding the hyperplane that separates a dataset into two parts in the best way possible. Furthermore, support vectors are considered vital elements of a dataset, as they affect the location of the dividing hyperplane when removed. A hyperplane can be a line that linearly divides a set of data while classifying it at the same time. Intuitively, classification will be improved if the data points lie further away from the hyperplane. Hence, it is vital that the data points be as far from the hyperplane as possible. SVMs conduct regression by reversing the objective: more specifically, by finding a hyperplane around which the most data points lie in close vicinity. The limitations of SVMs include their lack of suitability for large datasets, their need for heavy normalisation, and their tendency to underperform when the number of features is higher than the number of observations.

Artificial Neural Networks (NN)

Neural Networks (NN) have been used in the SPC context to detect different patterns representing out-of-control situations. Pham [45] developed a Neural Network model to identify the abnormal patterns on control univariate charts. This trained network can detect different out-of-control patterns such as linear trends and sudden changes from target output. Pham and Oztemel [46] devised an Neural Network-based method for the detection of small shifts in the variation of the output from the target for univariate process control purposes. Cheng and Tannock et al. [47,48] devised an approach that detects assignable causes at the early stages of process monitoring using a single SPC chart. Their work look the form of sequential pattern analysis using a back-propagation learning network. The sequential analysis incorporated 'Sequence I', to recognise the abnormal patterns, and 'Sequence II', to identify the corresponding critical parameters of the mathematical model associated with trends in 'Sequence I'. A numerical simulation was conducted to validate this work [47].

Wang and Chen [49] proposed an Neural Network with a fuzzy model to detect mean shifts in the multivariate method. The severity of these changes in the mean was classified into different decision intervals using a fuzzy classifier and an external process decision rule for deciding the change status. A comparison of results using a Hotelling T^2 multivariate control chart revealed that the proposed model outperformed the T^2 charts in detecting the out-of-control causes. A Neural Network model was further proposed to complement multivariate Chi-Square chart-based methods. This model was used to evaluate the transition signals derived from the multivariate Chi-Square chart, then provide specialised specific shift knowledge for multivariate procedures [50]. Niaki and Abbasi [51] proposed a Neural Network model of a multivariate process to monitor a process shift from the desired target and identify different types of defects in a product. A more detailed review of Neural Network applications in the multivariate process control context was provided by Psarakis [52]. In terms of the limitations of this method, its computational cost is massive and it is difficult to interpret; moreover, it is also unsuitable for small datasets.

Coupled Penalty Matrix Approach for Mixed Datasets

The Penalty Matrix method (PM) is used to quantify divergence from predicted outcomes [13]. Process responses are grouped into three groups: (i) acceptable, (ii) unacceptable, and (iii) moderating between the two categories (see Figure 2.1). A zero-penalty value is applied to the acceptable response, while a 100-penalty value is applied to an unacceptable response. A penalty amount between zero and 100 is also allocated to the method response in the middle area. The rationale behind the penalty matrix data visualisation approach involves highlighting patterns related to both acceptable and unacceptable response values by classifying the input factor data into either quartiles or categories. The correlation between the factor and the penalty value for a given response is observed using the PCA-based CLI plot [13], as discussed in chapter one. However, this method is limited to continuous (quantitative) variables and is unable to account for missing data. The datasets used in this research required the use of mixed data types in which both continuous and categorical variables are employed in the presence of missing data.

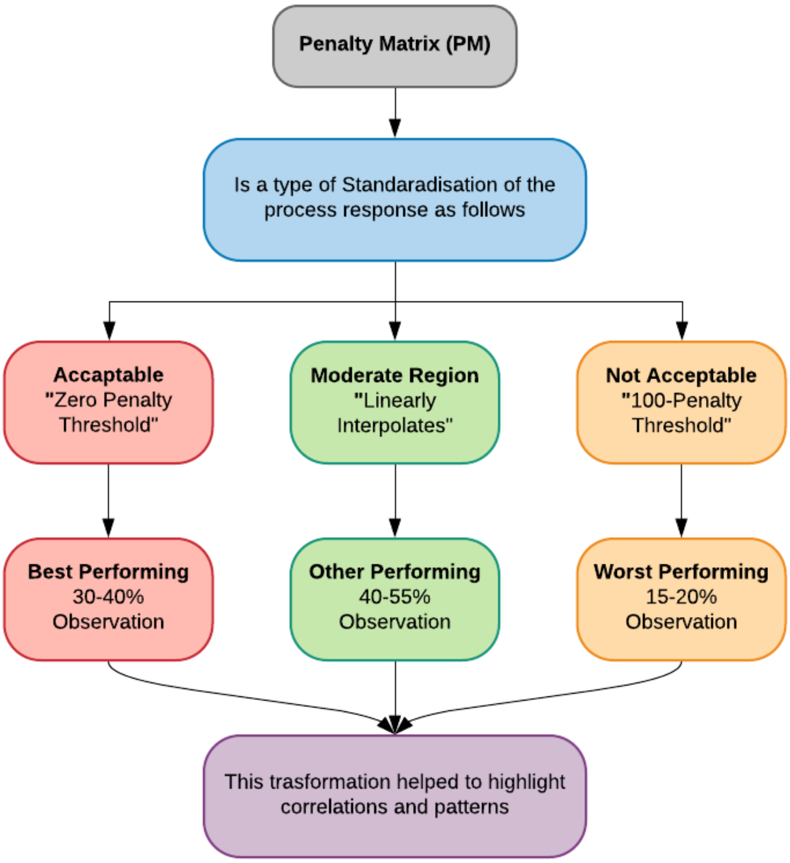

Penalty Matrix (PM)
Is a type of Standaradisation of the process response as follows
Accaptable "Zero Penalty Threshold"
Moderate Region "Linearly Interpolates"
Not Acceptable "100-Penalty Threshold"
Best Performing 30-40% Observation
Other Performing 40-55% Observation
Worst Performing 15-20% Observation
This trasformation helped to highlight correlations and patterns

Figure 2.1: Penalty matrix approach for visualisation

Quantile Regression Trees (QRT)

Quantile Regression Trees (QRT) take into consideration the effects of covariates on the full conditional distribution, rather than averaging regression trees as Random Forest does [16]. It extends a decision tree by providing predictions at percentiles, fitting a decision tree, and storing the full conditional distribution for all node data at the terminal node (target value). Giannetti [16], introduced a methodology for improving the robustness of manufacturing operations based on these quantile regression trees. The algorithm allows process engineers to discover and visualise optimal ranges. The significant drawbacks include the feature selection approach being based on the linearity assumption, an inability to handle missing data, and the number of trees being limited to the number of factors that leads to high variance, where the variance error is inversely proportional to the number of trees (i.e. limitation of trees tends to increase variance error) [53]. Figure 2.2 presents the steps of the algorithm.

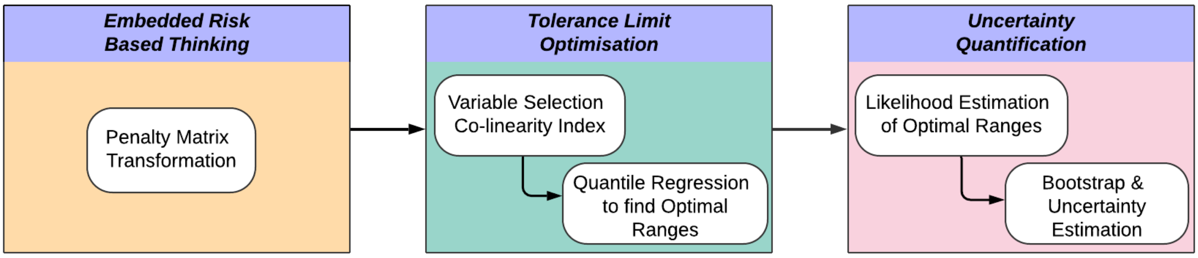

Embedded Risk Based Thinking
Penalty Matrix Transformation
Tolerance Limit Optimisation
Variable Selection Co-linearity Index
Quantile Regression to find Optimal Ranges
Uncertainty Quantification
Likelihood Estimation of Optimal Ranges
Bootstrap & Uncertainty Estimation

Figure 2.2: Quantile regression tree algorithm steps [16]: (a) Embedded risk-based thinking; (b) Tolerance synthesis; (c) Uncertainty quantification

Quality Correlation Algorithm (QCA)

The Quality Correlation Algorithm (QCA) [8] was developed based on modified PCA. It proposes a refined operating range for one or more of the process inputs by analysing the contribution of the PCA scores for every output. The uncertainty in the estimated results using the proposed algorithm was quantified by the Bootstrap sampling method [54], which was developed by simulating the error of the QCA model via a Bootstrap replacement strategy. The operating ranges discovered by the QCA algorithm are then further adjusted with reference to this uncertainty quantification, as seen in Figure 2.3. Regression-based algorithms are modified to take the missing values for the mixed dataset into account. It was shown that the algorithm proposed by Batbooti [8] had better prediction abilities than the commonly used factor analysis for mixed data (FAMD) method [55]. Batbooti's [8] algorithm was thus used to predict the process response corresponding to a given set of values for process inputs. The set of values for process inputs was generated using the operating ranges discovered by the QCA algorithm using the Bootstrap sampling method [54]. The main drawbacks of the QCA-based algorithm include the linearity assumption of PCA, the high computational cost of prediction, and restriction to a single imputation method.

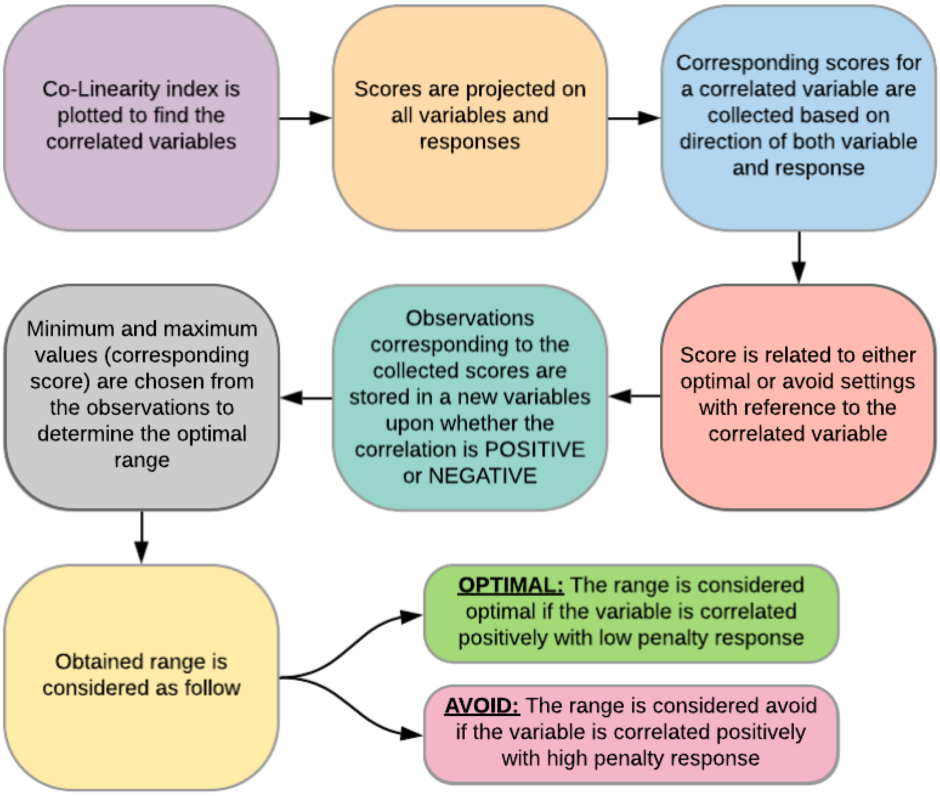

Co-Linearity index is plotted to find the correlated variables
Scores are projected on all variables and responses
Corresponding scores for a correlated variable are collected based on direction of both variable and response
Score is related to either optimal or avoid settings with reference to the correlated variable
Observations corresponding to the collected scores are stored in a new variables upon whether the correlation is POSITIVE or NEGATIVE
Minimum and maximum values (corresponding score) are chosen from the observations to determine the optimal range
Obtained range is considered as follow
OPTIMAL: The range is considered optimal if the variable is correlated positively with low penalty response
AVOID: The range is considered avoid if the variable is correlated positively with high penalty response

Figure 2.3: Optimal range process steps in QCA

In this thesis, the author has constructed his approach based on the following research direction, as proposed by Batbooti [8]:

- Identification of the optimal/avoidance range for each factor in terms of its tolerance (process) limits;
- Development of a confirmation trial plan to validate the optimised model performance versus the original limits;
- Conducting uncertainty quantification on the QCA algorithm to minimise the error in the model.

2.3 Summary of Statistical Learning Models

Table 2.1 summarises the statistical process control and machine learning algorithms most commonly applied to manufacturing process knowledge discovery. The table lists the application typically used with each technique. It also presents the advantages and disadvantages of each technique, as discussed above. Based on this comprehensive overview of the limitations of previous techniques, a new approach will be presented in this thesis.

Technique	Application	Advantages	Disadvantages
Hotelling's T^2	1) Multivariate air quality control [18]. 2) Hot forming process [19].	Real-time monitoring.	1) Computationally inefficient, with a high number of process inputs. 2) Lack of good causal relationships to identify the root cause of the fault.
(PCA)[a] /(MFA)[b]	1) Prediction of RNA-Seq Malaria vector [56]. 2) Stochastic Bottleneck: Rateless Auto-Encoder for flexible dimensionality reduction [57]. 3) Deterioration of historical buildings [58].	1) Easily visualise the results with the use of the Bi-plot. 2) Discover and explain the existence of dominant factors that contribute to the total variance.	1) Sensitive to outliers. 2) Unable to visualise complex problems where the majority of the variation is explained by more than the first 2/3 PCs. 3) Assumes linear relationship among variables.
(SOV)[c]	Multistage manufacturing processes (MMPs) [24].	Provide a variation propagation model from design to final product.	Requires domain expertise, and assumes a linear transition model between phases.
(FDA)[d]	Assessment of the health condition of bars [37].	Reduced the high dimensional datasets to one dimension and discover the correlation between variables.	Restricted to limited number of input variables, and requires heavy domain expertise.
(BN)[e]	1) Semiconductor manufacturing [38]. 2) Automotive body assembly process [39].	Provide a good causal relationship.	1) Resource-demanding algorithm that has long training cycles. 2) Requires substantial prior domain knowledge.
(SVM)[f]	A new GPU implementation of SVMs for fast hyperspectral image classification [59].	Relatively memory-efficient and more effective in high-dimensional spaces.	Not suitable for large datasets and requires heavy normalisation.

Technique	Application	Advantages	Disadvantages
(NN)[g]	1) Cost estimation for sheet metal [60]. 2) Cold rolled steel process [61]. 3) Applying Artificial NN in construction [62]. 4) COVID-19 incidence rates across the continental US [63].	Able to deal with highly non-linear problems.	1) Complex tools that require experience and mathematical understanding for proper use. 2) Unable to determine relationships between parameters. 3) Only relates inputs to outputs. 4) Care required to over-fitting the data.
(PM)[h]	Discovering product-specific foundry process knowledge from in-process data [13].	Able to help in highlighting the correlation and patterns.	Limited to quantitative variables, and inability to handle missing data.
(QRT)[i]	Risk-based uncertainty quantification to improve robustness of manufacturing operations [16].	Discover the critical process inputs.	Has a high variance, and feature selection based on the linear assumption.
(QCA)[j]	Nickel casting defect reduction [8].	Discover the optimal process settings for defect reduction.	Based on linear assumption of PCA, and has high computational cost.
(RF)[k]	Using data mining to predict secondary school student performance [64].	Overcomes the high variance in the regression trees, and deals with non-linear datasets.	Non-parametric, and requires a large dataset to implement the algorithm.

Table 2.1: Summary of multivariate data analysis methods

[a] Principal Component Analysis

[b] Multiple Factor Analysis

[c] Stream-of-Variation Analysis

[d] Fuzzy Data Analysis

[e] Bayesian Belief Network

[f] Support Vector Machine

[g] Artificial Neural Networks

[h] Penalty Matrix

[i] Quantile Regression Tree

[j] Quality Correlation Algorithm

[k] Random Forest Algorithm

2.4 Introduction to Random Forest (RF)

A decision tree is a representation of the possible outcomes that can be obtained from a wide array of related choices or actions. An individual or an organisation may use a decision tree to evaluate their possible options in order to identify the best action that will yield the optimal results. In essence, the decision tree is a decision support tool. The selection of the most rewarding action is influenced by the associated costs and the likely benefits derived from that action (decision). A typical decision tree is made up of several nodes and branches. The tree begins with a single node, which then branches into more than one possible outcome (i.e. from the root node). Each outcome produces more decision nodes, which also branch off into even more outcomes. This process continues until a terminal node is reached and no further splitting occurs. A well-developed decision map resembles a tree in shape.

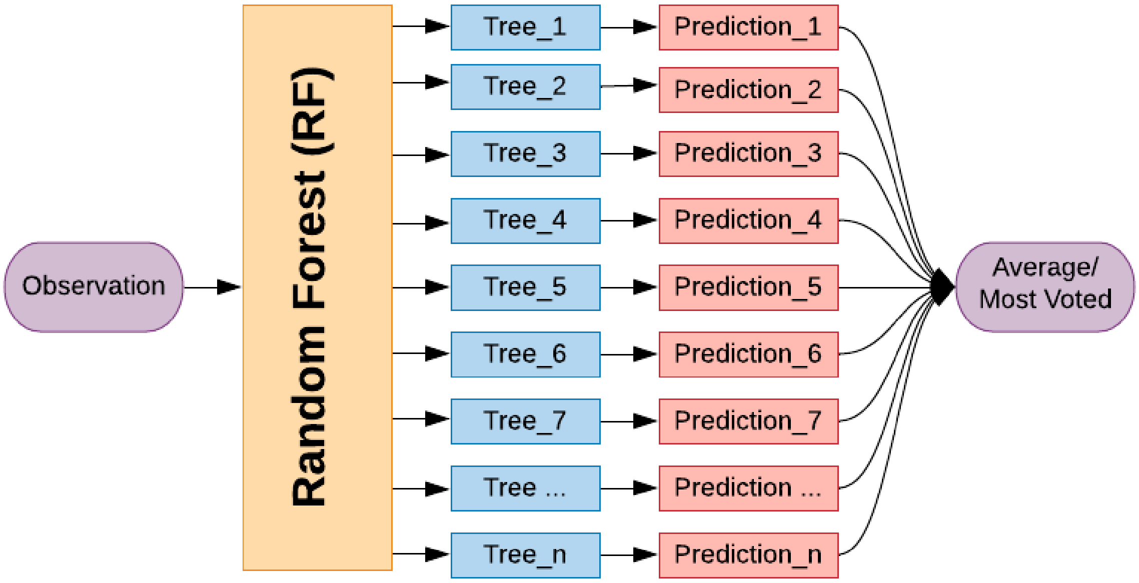


Figure 2.4: Random Forest algorithm model

Ho [65] discovered that an ensemble of trees can increase the accuracy (compared to a single decision tree) as they grow without the problem of overfitting, provided that (i) they are randomly selected to be insensitive to specific feature dimensions, and (ii) they are able to be separated with oblique hyperplanes. Later, the work of Geman and Amit [66] influenced the early development of the Random Forest concept proposed by Breiman, which introduced the concept of searching over a random subset of decisions at the time of splitting a node when growing a single tree. Ho's [67] random subspace selection concept was also influential in RF development. In this case, a forest of trees is first grown. Subsequently, there is an introduction of variation among the trees through the projection of the training data into a subspace that is randomly selected before fitting each node. Consequently, Dietterich first introduced the concept of the optimisation of a randomised node; in this process, a randomised procedure rather than a deterministic optimisation is employed to choose the decision at each node [68]. One of the primary tools used for data analysis is a decision tree, which is an example of a supervised machine learning technique. Random Forest (RF) is an ensemble learning classification and regression technique that works by creating a variety of randomly constructed decision-making trees at the training phase. The prediction of the random forest involves either a class label or numeric values for cases of classification and regression, respectively. RF algorithms can solve regression and/or classification tasks, and thus come up with the automated procedures for feature estimation through the use of past observations. RF is an example of an ensemble method, in which various weak learners are trained on the training sample and an arbitrary subgroup of variables. It has been argued that increased randomness also improves the precision of the estimation, eliminates bias, and integrates the different predictions in order to make the final forecast. Interestingly, decision trees and Random Forests are a class of algorithms that have proven to be accurate, successful, and robust tools for solving many machine learning tasks, including density estimation, classification, manifold learning, and regression. The model of the Random Forest algorithm is shown in Figure 2.4.

It is also important to note that Breiman properly introduced Random Forests in a paper that explains how to develop a forest of uncorrelated trees by using a procedure similar to Classification and Regression Trees (CART) [69], together with a randomised subset of the data and the use of the Bagging technique. Bagging is used to minimise the variance of a decision matrix. It is often used to eliminate weak models and to combine several predictions in order to select the most realistic one. In addition, the technique can be used by an individual or an organisation to develop stable decision tree models that significantly reduce variance. In so doing, these models, therefore, increase the accuracy of the decision trees and eliminate overfitting problems. Intriguingly, regression and classification trees represent a reasonable point from which to start understanding the CART model. The basic concept of CART is easy to understand. In essence, a set of observed features are used to separate the data recursively until the response variable values in every sub-partition become homogeneous. One measure of variable importance contributes toward this homogeneity. The steps of the CART algorithm in terms of regression and classification are addressed in detail in chapter three. Figure 2.5 illustrates the Random Forest cycle.

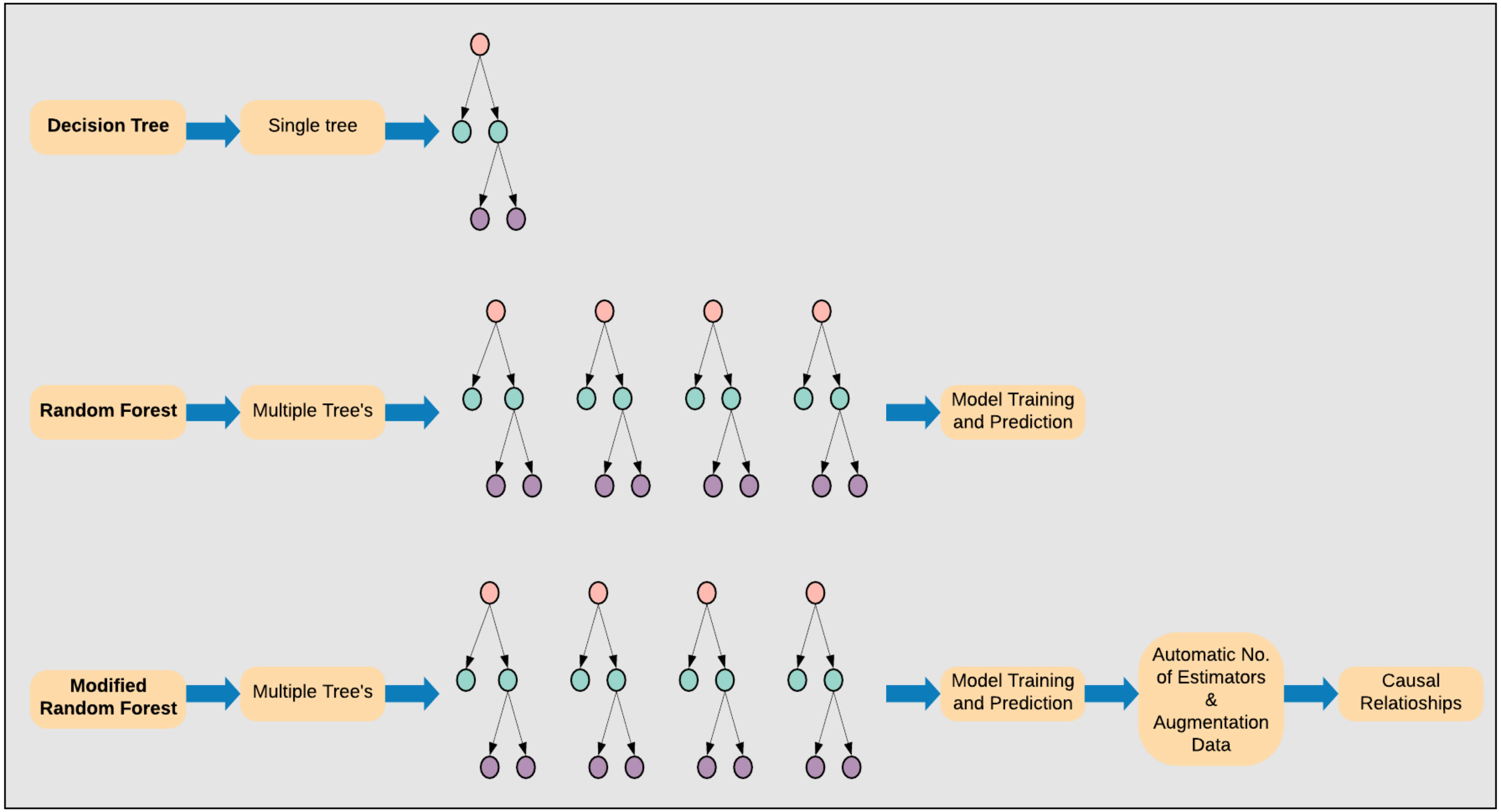


Figure 2.5: Random Forest cycle

2.5 Merits of the Random Forest Algorithm

In the current work, the Random Forest algorithm (RF) is considered due to the several advantages it offers. Figure 2.6 presents the advantages of Random Forest, which include the following:

1. The ability to efficiently handle large datasets with missing values;
2. No need for normalisation of the selected features;
3. The ability to scale up to thousands of variables;
4. Outliers do not significantly influence its accuracy;
5. The ability to handle classification and regression problems;
6. Requires minimal domain expertise to run and get accurate results;
7. The ability to determine variable significance (critical factors);
8. Computationally efficient.

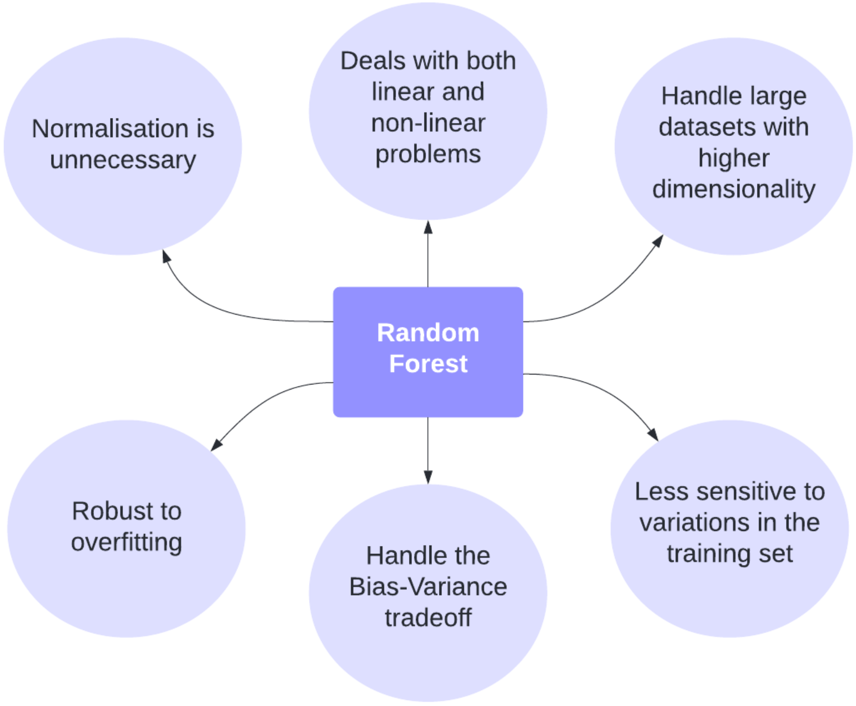

Normalisation is unnecessary
Deals with both linear and non-linear problems
Handle large datasets with higher dimensionality
Random Forest
Robust to overfitting
Handle the Bias-Variance tradeoff
Less sensitive to variations in the training set

Figure 2.6: Advantages of Random Forest features

2.6 Drawbacks of the Random Forest Algorithm

However, the Random Forest method also has significant drawbacks that need to be overcome. In this thesis, the challenge involves dealing with small datasets, eliminating overfitting, generating efficient response prediction, and dealing with causal relationships. The drawbacks of the Random Forest algorithm, as shown in Figure 2.7, are thus as follows:

- It is non-parametric, making it difficult to interpret;
- Requires a large dataset to perform efficiently;
- Needs precise tuning in order to limit overfitting.

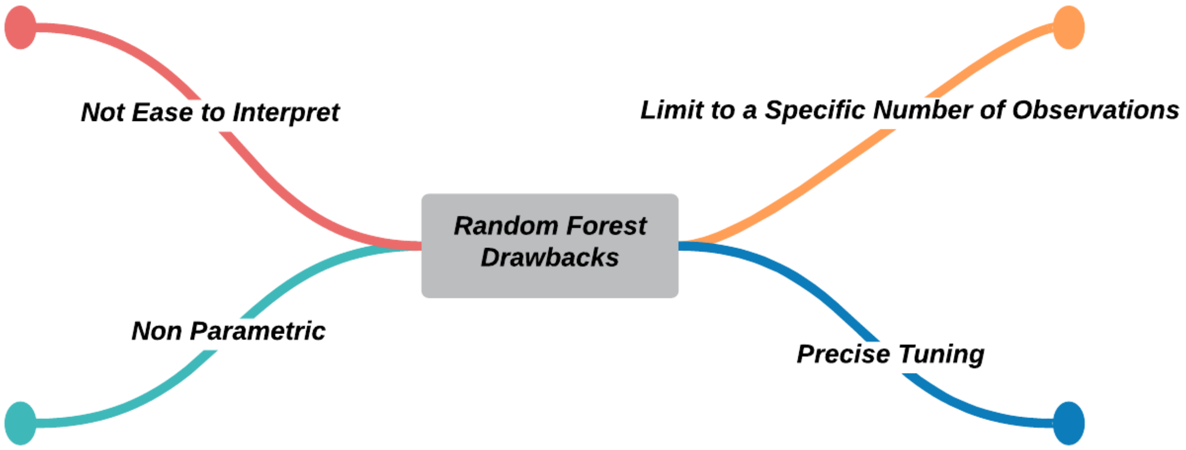


Figure 2.7: Limitations of the Random Forest algorithm

Although Random Forest does have some drawbacks, as noted above, its benefits far outweigh its limitations: Again, it is computationally efficient, can deal with outliers, handle classification and regression problems, naturally reduce variance, and requires minimal domain expertise. These merits overcome most of the drawbacks and gaps of the algorithm. In chapters four and five, the author will introduce a new proposed algorithm to overcome these limitations by producing a robust non-linear algorithm designed to deal with extracting the causal relationship between the process factor and response (shrinkage defect).

2.7 Missing Data Imputation Methods

Generally speaking, industrial datasets may contain missing data for a number of reasons, which include data collection errors, measuring sensor errors or any other cause that results in missing observations. Dealing with missing data is essential because the machine learning model cannot be trained using data with missing observations. Thus, the problem of missing data imputation must be solved by appropriate methods. Different approaches have been identified in previous studies to deal with missing data (see Figure 2.8).

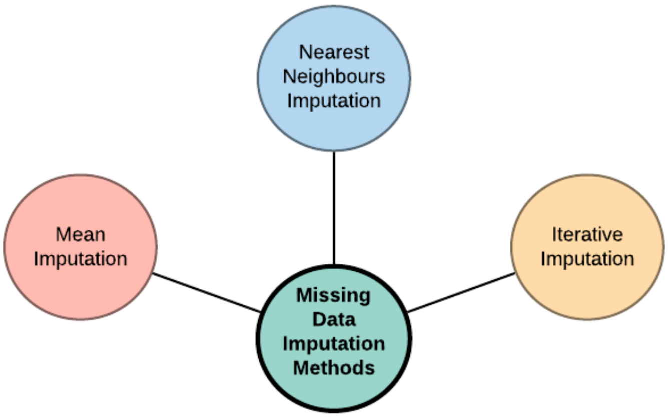

Nearest Neighbours Imputation
Mean Imputation
Iterative Imputation
Missing Data Imputation Methods

Figure 2.8: Missing data imputation methods

2.7.1 Mean Imputation

Mean imputation is a commonly utilised approach based on replacing the missing value using the variable mean. While it is straightforward, its limitation is that it underestimates the real variance of the variable, and even the standard deviation [70].

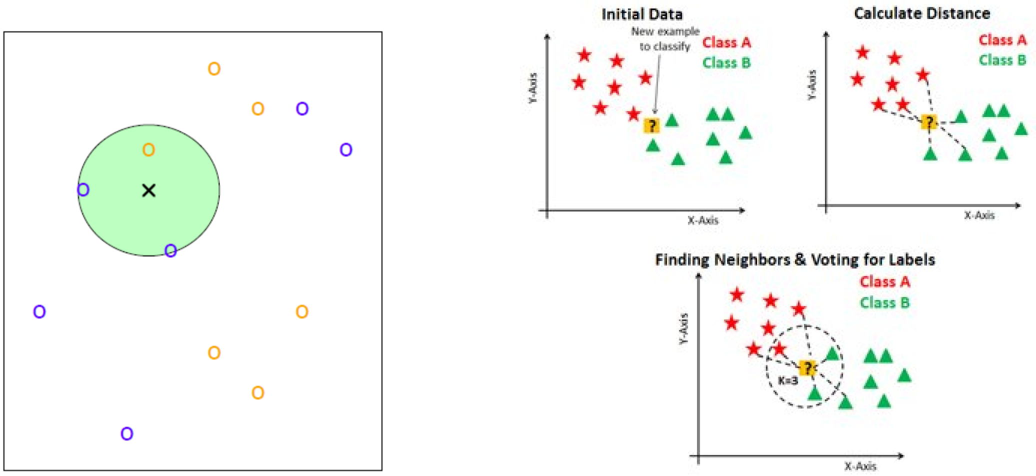
2.7.2 Nearest Neighbours Imputation

Laaksonen [71] introduced an imputation method based on the Nearest Neighbours algorithm, referred to as regression-based Nearest Neighbour hot decking. The distance between observations is employed to classify the data into clusters, while the missing observation is replaced with the mean of the Nearest Neighbour cluster. K-Nearest Neighbour algorithms have been designed based on K-Nearest Neighbour imputation [72], weighted K-Nearest Neighbour imputation [73], and fuzzy K-means clustering imputation [74].

Steps

This *K*-Nearest Neighbours (KNN) algorithm is broken down into three parts:

1. Cluster the data, and calculate Euclidean distance;
2. Get the Nearest Neighbours;
3. Make predictions.



(a) KNN Imputation

(b) KNN Steps

Figure 2.9: K-Nearest Neighbour (KNN) Imputation [75]

2.7.3 Iterative Imputation

The main concept underpinning this approach is that each feature with missing values fits a function of other features, and this function is used to predict the missing values. The candidate feature column is treated as output, while the other feature columns are treated as inputs. Using an iterative approach, this is done for each feature and then repeated until either the maximum number of iterations is reached or the convergence criterion is attained. Daniel, Stekhoven and Peter [15] proposed the so-called 'missForest' approach, which is RF-based. Using the built-in out-of-bag error estimation of Random Forest, it is possible to approximate the imputation error without the need for a validation dataset over other imputation approaches, particularly when complex relationships and non-linear connections are perceived for predictive accuracy and can therefore cope with high-dimensional data. This algorithm is discussed in more detail in chapter three.

2.8 Research Challenges

From the previous discussion, it emerged that optimising any production process requires a model capable of overcoming the limitations of each technique. Figure 2.10 presents a summary of these limitations. The candidate machine learning model is required to demonstrate the following abilities:

- It can act as a predictive non-linear model (as defined in Section 1.2);
- It can deal with a small dataset (as defined in Section 1.2);
- It can deal with high-dimensional data;
- It can handle missing data efficiently;
- It has low computational power;
- It is an easy-to-use tool for non-machine learning experts;
- It can extract the causal relationship between the process factor and the response.

In the next chapter, a model capable of tackling and overcoming most of all the previous limitations is presented and explained. The chosen model can outperform the other competing approaches. The problem discussed in this thesis follows a different path to the comparison methods; this approach requires a novel machine learning model that is capable of dealing with the abovementioned challenges.

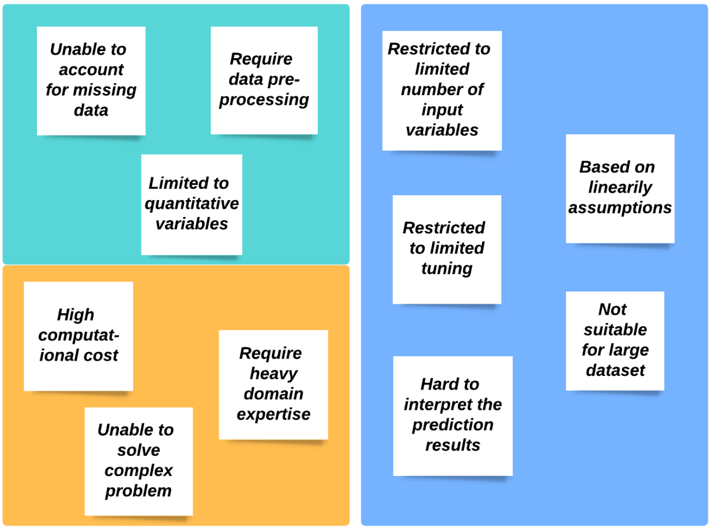

Unable to account for missing data
Require data pre-processing
Limited to quantitative variables
High computat-ional cost
Require heavy domain expertise
Unable to solve complex problem
Restricted to limited number of input variables
Based on linearily assumptions
Restricted to limited tuning
Not suitable for large dataset
Hard to interpret the prediction results

Figure 2.10: Limitations and drawbacks of the various methods described in this thesis

Chapter 3

Mathematical Modelling of Predictive Causal Analytics

3.1 Introduction

Machine learning is a subset of artificial intelligence (AI) that enables learners to be trained independently based on previous experiences. In other words, machine learning is the study of systems that are able to learn from data without the need to program such machines explicitly. More specifically, machine learning provides both algorithms and insights on the predictive data.

In this chapter, knowledge discovery algorithms are evaluated in order to develop a predictive model capable of dealing with non-linear problems. This chapter is structured as follows. The machine learning workflow is described in Section 3.2, while Section 3.3 discusses the data exploration and feature engineering methods. Section 3.4 presents the predictive causal analytics while also explaining the prediction accuracy. Section 3.5 reviews the comparisons of predictive model development. Section 3.6 explains the decision tree and its construction. The advantages and disadvantages of the decision tree are discussed in Section 3.7, after which Section 3.8 discusses the bias-variance trade-off. The ensemble methods based on randomisation, along with the algorithm of the traditional RF approach, are defined in Sections 3.9 and 3.10 respectively. Section 3.11 illustrates the pre-processing steps for the encoding of categorical variables. Section 3.12 outlines the missing data imputation methods, while Section 3.13 discusses data augmentation using different techniques. Model tuning and the assessment of model accuracy are explained in Sections 3.14 and 3.15, respectively. Various validation techniques are addressed in Section 3.16. A research roadmap is proposed in Section 3.17. Finally, the technological choice for this thesis is described in Section 3.18.

3.2 Machine Learning Workflow

Data analysis is a method of reviewing, cleaning, transforming, and modelling data in order to find valuable information, along with conclusions to support the selected decision. The data analysis process includes the following steps:

- Defining data requirements.
- Data collection through a variety of sources like sensors, satellites, recording devices, etc.;
- Data processing by sorting data in matrix form;
- Data cleaning (removing any possible error or wrong information within data);
- Exploratory data analysis for identifying correlations between features;
- Modelling and algorithm development, where algorithms are implemented for the identification of correlations between variables;
- Data product: The data product is dedicated to procedures in which data is input and the outputs are produced, then returns both inputs and outputs to the system. This could be based on models or algorithms.

The standard workflow of machine learning applications involves all relevant phases, beginning from raw data collection to pre-processing, data cleaning, the discovery of technologies and technical capabilities, and demonstrating the effect it has on the efficiency of machine learning models. Figure 3.1 illustrates a broad picture of the steps of model development.

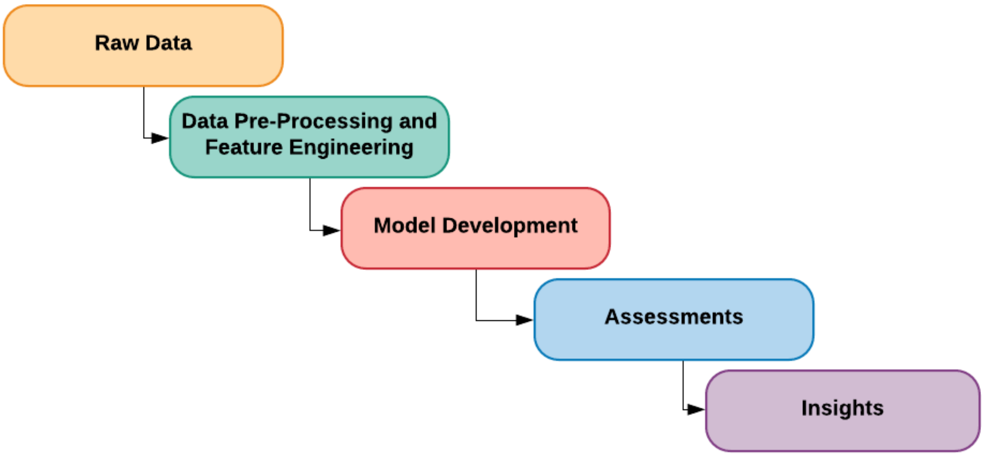


Figure 3.1: Machine learning workflow

3.3 Data Exploration and Feature Engineering

A nickel-based superalloy dataset is used for the manufacturing of cast components for an aerospace foundry. The dataset consists of 16 chemical composition factors (see Appendix A) that affect the process response, along with the percentage of defective components produced in a batch due to shrinkage defects. The input factors are quantitative variables and the process data for 60 observations is available. This example is taken as the benchmark example. This dataset is selected because Swansea Research used this example, making it a suitable benchmark. It also contains a limited number of observations, non-linear interactions, and highly skewed data (indicating high variance), while the data is also based on a real-world case. The process response of the dataset is depicted in Figure 3.2. Figure 3.3 further illustrates the distribution of process inputs aggregated for 60 batches/observations in the nickel-based superalloy dataset. Moreover, Figure 3.4 plots the factor values on the horizontal axis and the response values on the vertical axis to reveal correlations on the optimal or avoidance limits. The manual inspection of such graphs is difficult and can be unreliable when the number of data points increases or there are overlaps between points. Manual interpretation is also particularly difficult when no clear correlation exists between the process input and the process output, while understanding the effect of interactions is also difficult. It is far easier to interpret the data using a machine learning model that has been trained to estimate the recommended process limits. However, this is particularly challenging for non-linear systems, for which there is no standard procedure that achieves this. Accordingly, one of the main goals of this thesis is to devise an algorithm capable of achieving this functionality for generic datasets. The objective is thus to recognise such limits automatically to facilitate better understanding and interpretation.

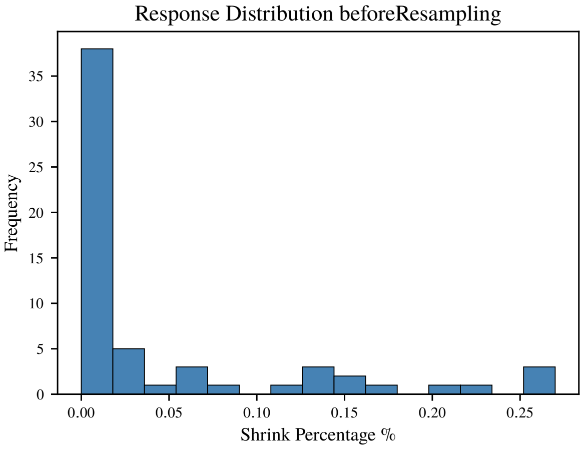


Figure 3.2: Process response distributions (Shrink Percentage %)

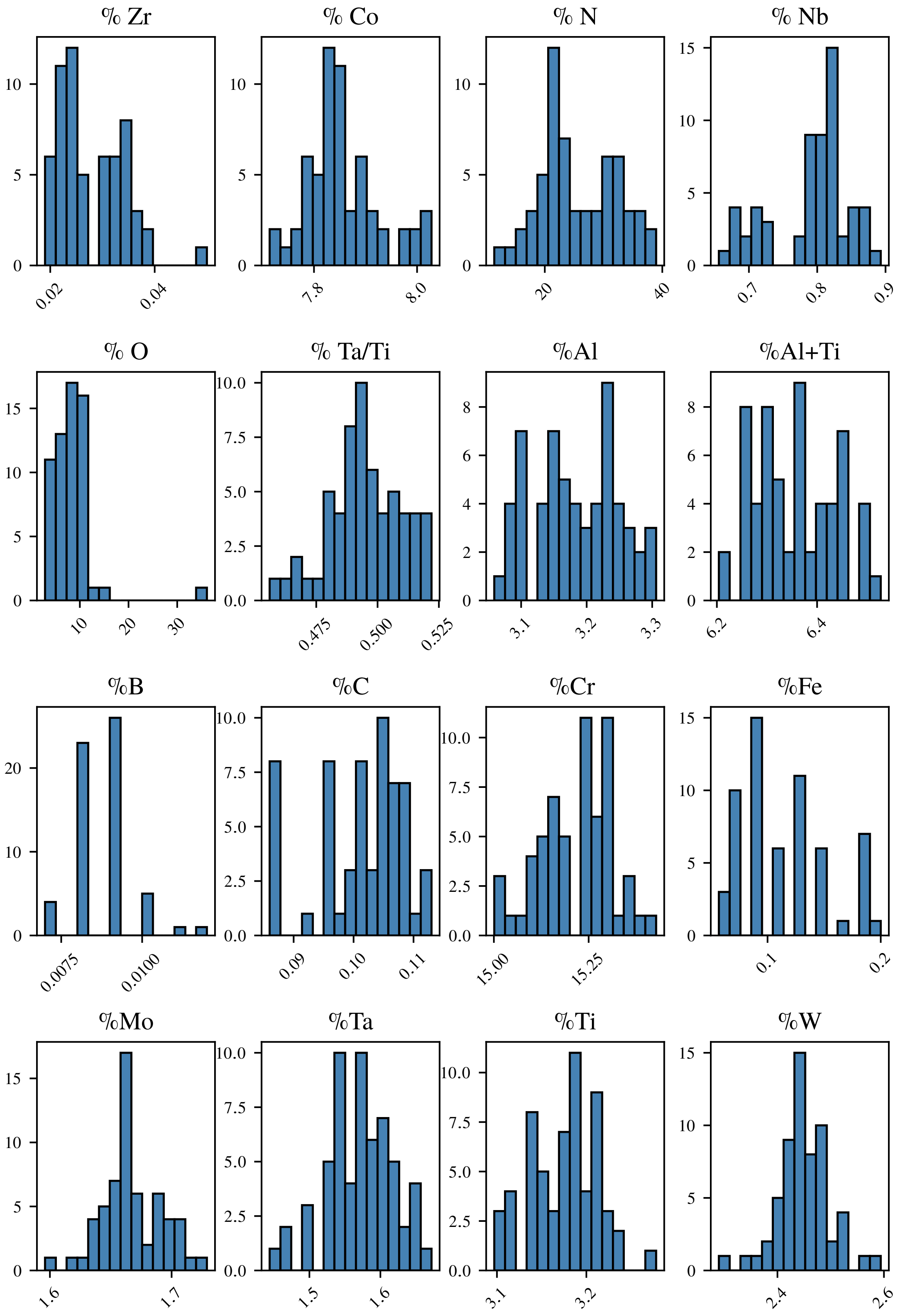


Figure 3.3: Process input distributions (16 factors)

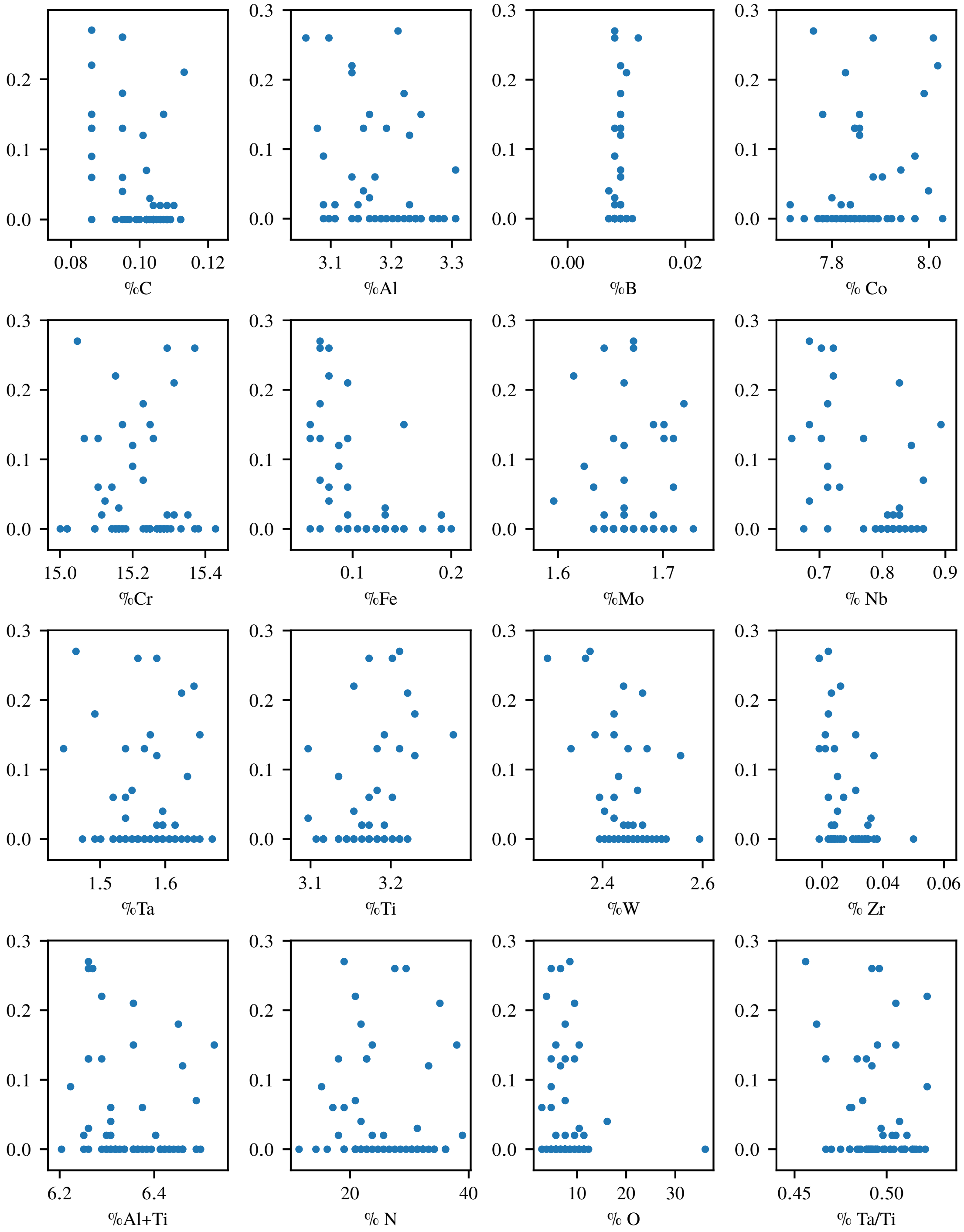


Figure 3.4: Process inputs versus output response (16 factors)

It is infeasible to directly reduce the shrinkage defects of the product. However, it is possible to control the 16 process inputs (further details of the process inputs and output are presented in Appendix A). Therefore, if a causal link is found to exist between the process inputs and shrinkage defects, it will be possible to instruct the manufacturer to modify the process inputs and thus indirectly decrease the shrinkage defects. Hence, the first goal is to develop an accurate regression model that is capable of predicting shrinkage defects based on the 16 process inputs. Within this configuration, the process inputs are the model's input data variables, while the shrink percentage is an output response. The input variables are represented using the symbol X, with subscripts denoting the following subtypes: X_1 might be the $\%Zr, X_2 \leftarrow \%Co, ..., X_{16} \leftarrow \%Fe$. The inputs thus represent the data features. Moreover, the output variable - in this case, the shrinkage percentage - is the dependent variable (response) and is generally represented by the symbol Y. Thus, with the quantitative response Y and the 16 different factors $(X_1, X_2, ..., X_{16})$, it is possible to generalise the relation between Y and $X = (X_1, .., X_{16})$ as in Equation 3.1 [75].

$$Y = f(X) + \epsilon \tag{3.1}$$

All statistical learning models use a specific set of strategies for the estimation of $f(X)$. Throughout this chapter, some of the basic analytical concepts that are employed during the evaluation of $f(X)$ are also outlined, while tools for evaluating the estimates are acquired. An important step in machine learning model development is feature engineering and exploring the quality of the data before proceeding. Feature engineering refers to the process of extracting raw data parameters (i.e. features) that are used to train a machine learning model. It also involves performing various permutations on the low-level features to come up with higher-level features. Feature engineering plays an important role in the machine learning pipeline. There are several metrics used to evaluate the quality of the data, namely historical data required, skewness, and kurtosis (see Figure 3.5).

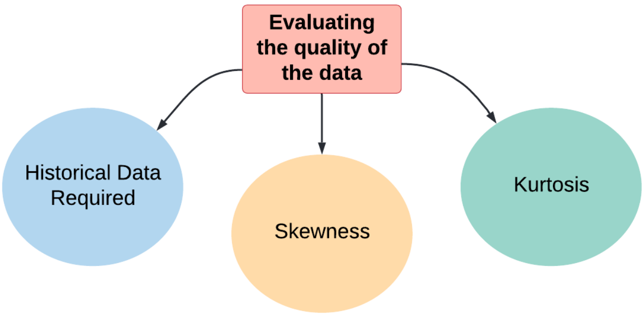


Figure 3.5: The evaluation metric for the data quality

Historical Data Required

A large dataset is essential if the machine learning model is to perform well. Statistical power reduces while overfitting increases when the sample size is small, which weakens the discovery of knowledge and accuracy of the estimation. As shown in Figure 3.3, the historical data available is limited to only 60 observations. To solve this problem, data augmentation should be performed to decrease overfitting in the model. Each dataset is unique and there is no such thing as a one-size-fits-all hyperparameter. The number of factors is also important here; small numbers of factors require a lower number of observations.

The sample size depends on the prevalence of rejection rates and the minimum magnitude of process improvement required to demonstrate success. For studies based on design of experiment guidelines, a very small sample size (10-20) is also sufficient as the magnitude of process improvement demonstrated is usually high. However, the algorithms developed in this thesis are based on the observational studies. The typical prevalence rejection rates for foundry industry are around 4-5%. One percent increase in the rejection rate constitutes to around 20% deterioration in the quality. A typical design of experiment study would aim a difference of at least 100% or more. The 20% magnitude of process improvement required to demonstrate the success is small in comparison to the prevalence rates. Statistically, it would require a very large sample size (e.g. couple of hundred observations) [76]. However, the top management in a foundry industry would not wait to produce couple of hundred batches with 20% higher rejection rates as the profit margins are around 10-20%. There is pressure on process engineers to intervene based on limited datasets. In the nickel-based superalloy case study used for this thesis, the process engineers were required to intervene after recording observations for 60 batches. In this case study (nickel-based superalloy), Random Forest would required a minimum of 320 observations [77] in order to achieve accurate prediction performance. Hence, the need for data augmentation and requirement of not introducing a bias and preserving the underlying cause and effect structure between factors and responses. In the literature [78–81], many oversampling techniques were found. However, these techniques cannot handle the non-linearity of the data. Therefore, a novel data augmentation technique was developed that is capable of dealing with mixed non-linear datasets; this technique will be introduced in chapter four.

Skewness

Skewness [82] demonstrates how non-symmetric data travels across the standard. Distortion from the normal distribution curve is used to determine the loss of symmetry in the distribution of the results. Typically, machine learning algorithms perform better when data is normally distributed;

thus, if the dataset has high skewness, it is preferable to eliminate this skewness by balancing the dataset before training. Skewness is calculated as in Equation 3.2 [82]:

$$skewness = \frac{1}{n}\frac{\sum_i^n (L-\mu)^3}{\sigma^3} \tag{3.2}$$

Where,

- L: The dataset values,
- μ: The mean value,
- σ: The standard deviation,
- n: The normalisation by number of elements.

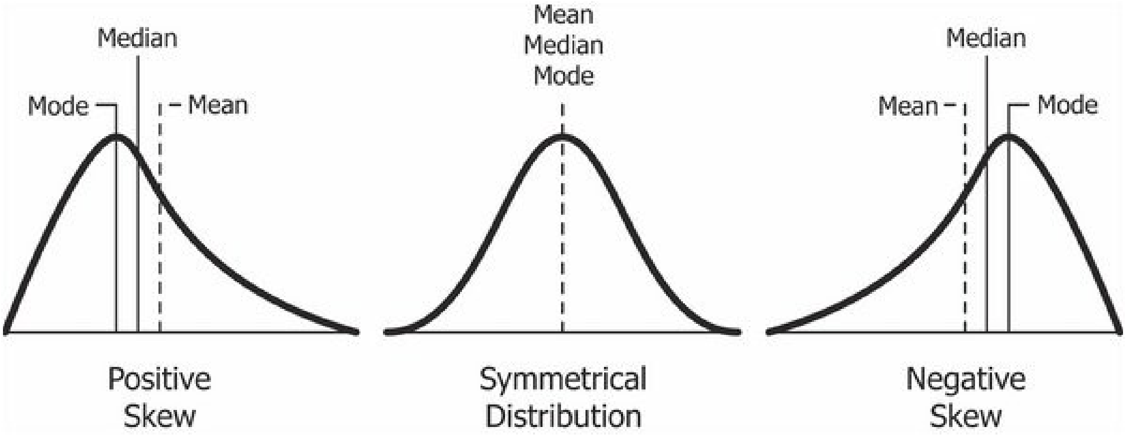


Figure 3.6: Schematic drawing for different skewness values [83]

The measures for skewness range and the different distributions (see Figure 3.6 for an illustration) are as follows:

- The normal (symmetrical) distribution has a skewness of zero.
- Positive skewness indicates a longer tail on the right side of the distribution. The mean and median are greater than the mode.
- Negative skewness implies that the tail on the left is more prominent. The mean and median are lower than the mode.
- Skewness in the range of (-0.5, 0.5) indicates relatively symmetrical data.

- Skewness in the range of (-1, -0.5) represents negative skewness, while skewness in the range of (0.5 and 1) represents positive skewness. Therefore, the data is partially skewed.

- A skewness level of lower than -1 (which means negatively skewed) or higher than 1 (positively skewed) indicates that the data is heavily skewed.

Figure 3.7 depicts the response of the nickel-based superalloy dataset, which is an extremely (positively) skewed distribution; accordingly, it induces imbalanced estimators that significantly impact predictive capacity.

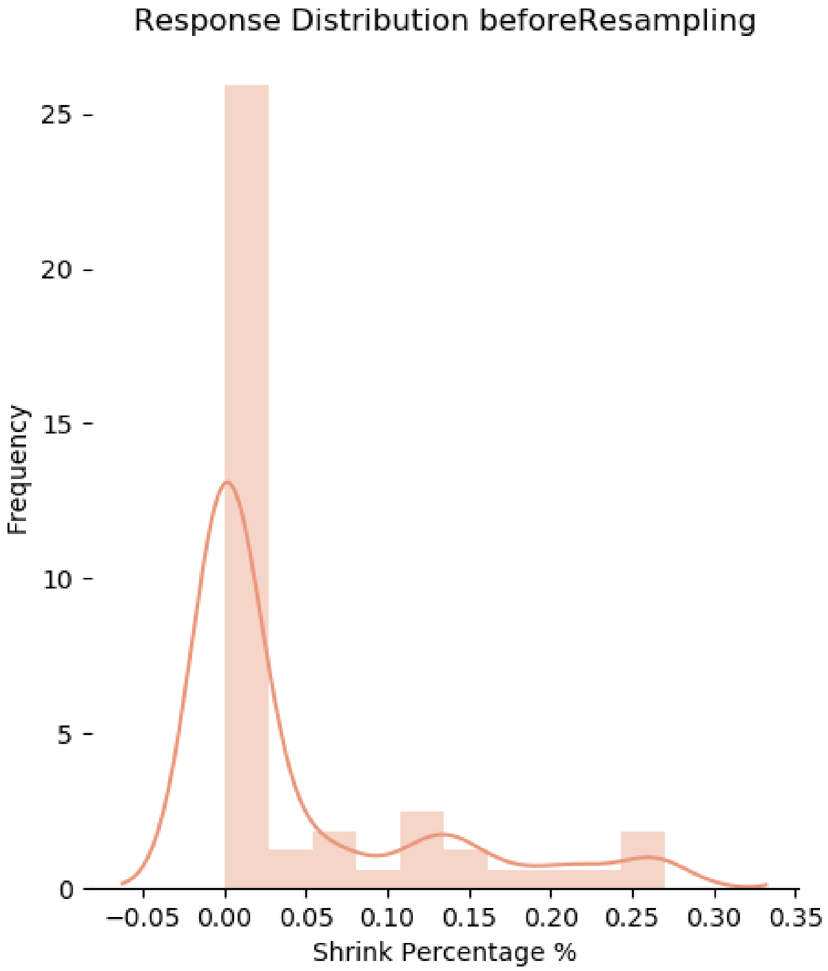


Figure 3.7: Response distribution of nickel-based superalloy dataset

Kurtosis

Kurtosis [84] mostly relates to the distribution tail. It is the approximation of the outliers found in the distributions. Kurtosis is calculated using Equation 3.3 [84].

$$kurtosis = \frac{1}{n}\frac{\sum_i^n (L-\mu)^4}{\sigma^4} \tag{3.3}$$

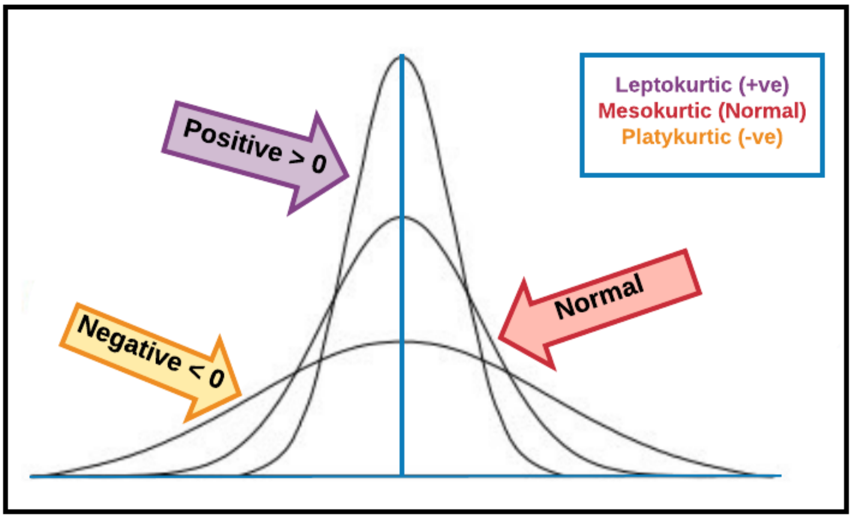


Figure 3.8: Schematic drawing for different kurtosis types [85]

Figure 3.8 indicates the disparity between the different types of kurtosis. Based on the formula above, kurtosis is categorised based on the following [85]:

- Mesokurtic (Kurtosis = 0): The distribution has a kurtosis margin close to that of the standard normal distribution. This implies that the maximum level of the distribution is identical to that of normally distributed data.

- Leptokurtic (Kurtosis > 0): The distribution is longer, has fatter tails and a higher and sharper peak than the normal distribution. High kurtosis implies that the results are heavy-tailed or a profusion of outliers exists. The more observers are involved in the results, the higher the probability of false-positive predictions.

- Platykurtic (Kurtosis < 0): The distribution is shorter, has thinner tails and a lower and broader peak than the normal distribution. Low kurtosis in the collected data is an indication that the sample has light tails or lacks outliers.

Distribution of Skewness and Kurtosis in the Nickel-based Superalloy Dataset

Figure 3.9 depicts the distribution of skewness and kurtosis in the nickel-based superalloy dataset, where the shrinkage defect and $\%O$ exhibit large values of skewness and kurtosis, respectively.

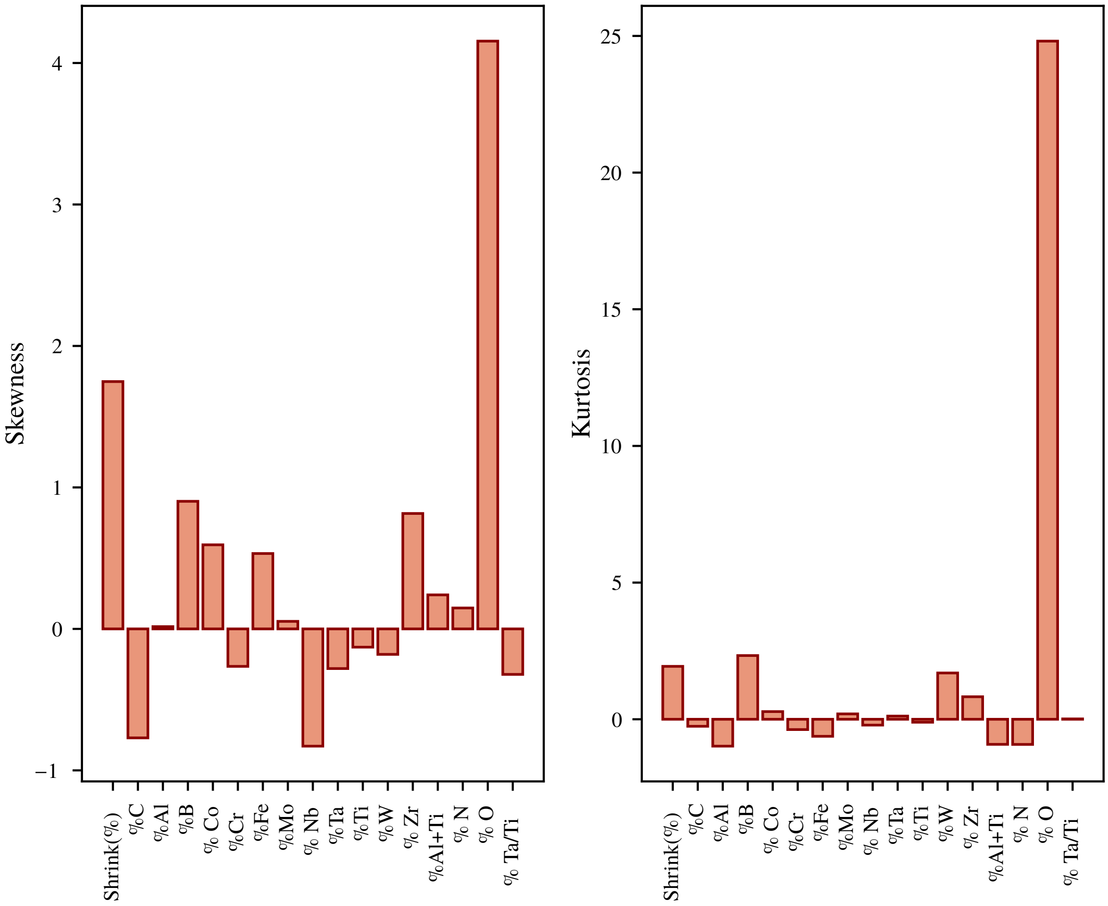


Figure 3.9: Skewness and Kurtosis in nickel-based superalloy dataset

In any tree-based algorithm, the normal distribution represents the optimum scenario. Any shift from the normal distribution will create an unbalanced tree. This will increase the tendency of the variance and thus lead to overfitting. Based on Figure 3.9, it can be seen from the output response that an unbalanced tree is expected. In order to resolve this matter and improve the performance, a data augmentation technique is utilised (this technique is introduced in chapter four) to minimise the skewness in the data and further balance the tree. High skewness and kurtosis in $\%O$ were observed. $\%O$ may contain outliers that cause this high variance. In this case, a confirmation from the process engineer is required in order to validate the correctness of the factor measurements obtained from the sensors, as there might be a defect or a miscalculation of results.

3.4 Predictive Causal Analytics

There are two main types of rationale behind all statistical learning models: The first is the discovery of causal knowledge, and the second is prediction.

3.4.1 Discovery of Causal Knowledge

A causal relationship is determined to exist between two occurrences if the incidence of the first causes the other. Here, the first event is called **the cause**, and the second event is called **the effect**. A relationship between two factors does not imply causation; on the other hand, if there is indeed a causal relationship between the two variables, they must be correlated. Notably, the predictive model cannot be treated as a black box; the interesting point to consider is the causal relationship between Y and X, or more specifically, an understanding of how the process response changes as a function of process inputs $X_i, ..., X_n$. Causal knowledge is discovered by identifying correlations that are supported by domain knowledge and verifying correlations with confirmation trials. This knowledge is then used for prediction. Accordingly, the purpose of extracting knowledge from the black box is to find an answer to the following questions:

- Which factors are associated with the response? (Finding a few critical factors among a broad set of process inputs can be useful.)
- What is the relationship between the response and each factor?
- Is it possible to properly articulate the relationship between Y and other critical factors mathematically?

3.4.2 Prediction

In any process, a set of observations X are accessible, but the output Y can only be approximated using Equation 3.4 [75].

$$\hat{Y} = \hat{f}(X) \tag{3.4}$$

Here, $\hat{f}(X)$ refers to the estimated $f(X)$, while $\hat{Y}$ refers to the resulting prediction for Y. In general, $\hat{f}(X)$ will not be an exact representation of $f(X)$. This inaccuracy can be attributed to two types of errors [75]: specifically, reducible and irreducible error. The source of **reducible error** is based on the choice of the mathematical or machine learning model used; using the best available analytical learning approach to evaluate $f(X)$ will improve this error. The source of **irreducible error** could be attributed to an unmeasured variable that may contain information useful for predicting Y. Moreover, if there are no measurements, $f(X)$ cannot be employed for the prediction. Consider a given model $\hat{f}(X)$ and a set of features X, which yields the prediction $\hat{Y} = \hat{f}(X)$. Considering both $\hat{f}(X)$ and X remain unchanged by [75],

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{Var(\epsilon)}_{\text{Irreducible}} \end{aligned} \tag{3.5}$$

Here, $E(Y - \hat{Y})^2$ in Equation 3.5 represents the squared error between the predicted and actual value of Y, while $Var(\epsilon)$ denotes the variance of the uncertainty correlated with the definition of error (ϵ). Finally, it can be inferred that all predictive approaches aim to reduce the reducible error. It should further be noted that the irreducible error always provides the upper limit for the precision of the prediction of Y. In chapter four, a novel approach for minimising the reducible error for the proposed predictive model is discussed.

3.4.3 Prediction Accuracy and Model Interpretability

There are several reasons why a more restrictive model might be chosen over a very flexible model when causality is of interest. Conditional structures are also easier to interpret under these circumstances. For example, when the primary purpose is to find a conclusion, the linear model could be the correct option, as the correlation between the response and the observations will be easier to comprehend.

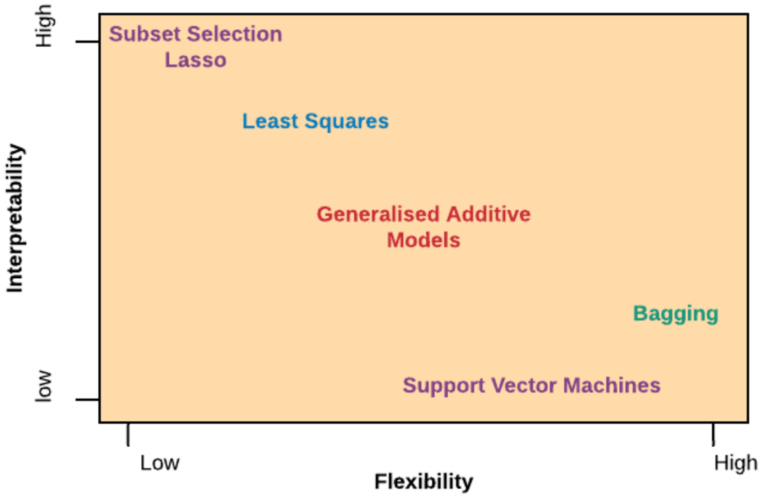


Figure 3.10: The trade-off between flexibility and interpretability [75]

Figure 3.10 illustrates the trade-off inconsistency in understanding when various statistical learning techniques are used. In this thesis, the challenge is how to increase flexibility in order to capture the non-linear relationship between the process factors while maintaining a high level of causality knowledge. Chapter five discusses the novel approach adopted to enhance the causal capability of the proposed algorithm.

3.5 Predictive Model Development

Machine learning is a data modelling approach that optimises the construction of computational models. The primary objective is to find patterns and correlations without having to explicitly program them. Rather than developing and programming logic, data are fed into a machine learning algorithm, which constructs logic depending on the statistics calculated. In this section, the terminology used in machine learning is discussed and the most important terms are defined.

3.5.1 Deep vs. Shallow Learning

Deep learning requires several levels of abstraction and several layers of non-linear processing systems. These techniques have been evolving quite rapidly over recent years and have seen practical implementations in fields such as computer vision and speech recognition. For their part, non-deep learning methods may also be categorised as shallow learning methods. Shallow learning methods are capable of learning from the available input features by extracting patterns; in deep learning methods, however, the algorithms are (inherently) capable of extracting complex patterns from the raw data. Table 3.1 explains the critical distinctions between shallow and deep learning.

Table 3.1: Key differences between deep and shallow learning

Shallow Learning	**Deep Learning**
Based on statistical characteristics of data.	Detect the features from the raw data.
The provided data will be saved with no additional comprehension.	Seeks to find correlation within previous and new knowledge.
Feature engineering is required for better accuracy.	Predefined feature engineering (Not required since it happens by itself).
Based on the provided features derived from heuristics of target.	No prior knowledge is required.

3.5.2 Supervised vs. Unsupervised Learning

The bulk of predictive learning challenges can be assigned to one of two categories: supervised or unsupervised (see Figure 3.11). Supervised learning is so called because the method of algorithm learning applied to the training dataset could be thought of as an instructor in control of the learning cycle. The right responses are already known; the algorithm then allows iterative assumptions about the training data and is corrected. Training ends once the algorithm reaches an appropriate degree of accuracy. The details addressed throughout this chapter all fall under the context of supervised learning. For each observation of the feature measurement(s) X_i, where $i = 1, ..., n$, there is a related Y response measurement.

The researcher's goal is to fit a model that relates the response to the features in order to accurately predict the response for future observations or to truly comprehend the relationship between the outputs and the critical feature input. Moreover, standard predictive learning approaches, such as linear regression and logistic regression, and more improved methods such as Generalised Additive Model (GAM) [86], Support Vector Machines (SVM) [43, 44], and Bagging (Chapter 4), also function in the supervised learning domain. Supervised learning problems may be further divided into regression and/or classification approaches. Further examples are listed in Table 3.2.

- **Regression:** Concerns the prediction of a continuous target value, based on the input feature(s) values.
- **Classification:** Concerns the prediction of a class label of the target, based on the input feature(s) values.

Table 3.2: Machine learning algorithm examples

Machine Learning	**Supervised**	**Unsupervised**
Continuous	Regression • Linear • Logistic • Decision Trees (RF)	Clustering and Dimensionality Reduction • Singular Value Decomposition (SVD) • PCA • K-means
Categorical	Classification • SVM • Naive-Bayes • Decision Trees (RF)	Association Analysis • Apriori • FP-Growth • Hidden Markov Model

For its part, unsupervised learning deals with unlabelled data. That is, for each set of observations X_i, the output response Y is unknown. It is not feasible to fit a linear regression model under these circumstances since there is no predictive response component. In this case, the task is referred to as unsupervised, since the response variable is not available for monitoring the analysis. Clustering is the statistical tool that can be applied in this case. The objective of clustering analysis is to classify input features in such a way that similar features are grouped together into their corresponding categories or ’bins’. Unsupervised learning tasks may be further classified into clustering and dimensionality reduction approaches. Table 3.2 presents explanations different unsupervised machine learning models.

- **Clustering:** Involves identifying the underlying groupings of data, such as classifying people through their purchasing behaviour. It also refers to the clustering of groups in such a way that objects in the same cluster are more analogous to one another compared to objects in different clusters.
- **Dimensional Reduction:** The goal is to reduce the number of features used for training while keeping as much variance in the data as possible.

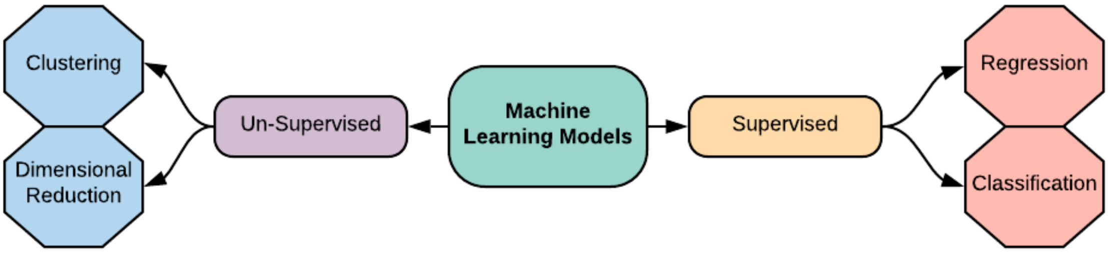


Figure 3.11: Machine learning approaches

Table 3.3 summarises the difference between supervised and unsupervised machine learning models according to three criteria; availability of labelled data, data knowledge, and end goal.

Table 3.3: Key differences between supervised and unsupervised learning

Supervised Learning	**Unsupervised Learning**
The data labels are known.	The data labels are missing.
Focused on training of dataset.	Learning with no initial knowledge.
Aim to classify future observations.	The goal is to explore the data.

3.5.3 Regression vs. Classification Problems

Variables can be characterised as either quantitative or categorical. Quantitative variables are continuous numerical data, while categorical variables are used to represent categories or groupings of data points. Problems that relate to a quantitative response are referred to as regression problems, whereas those associated with a qualitative response are referred to as classification problems.

3.5.4 Overfitting vs. Underfitting

Overfitting refers to a phenomenon such that a regression model tends to capture random errors in the data rather than the interaction between the variables, and occurs when the algorithm begins to 'memorise' training data rather than 'learning' to generalise from the pattern. On the other hand, an underfitted model is a configuration such that the model fails to capture the most important trends in the data. This will occur, for example, when a linear model is applied to a non-linear dataset and will lead to low predictive efficiency. Figure 3.12 illustrates how a prediction might be influenced by either underfitting or overfitting [87].

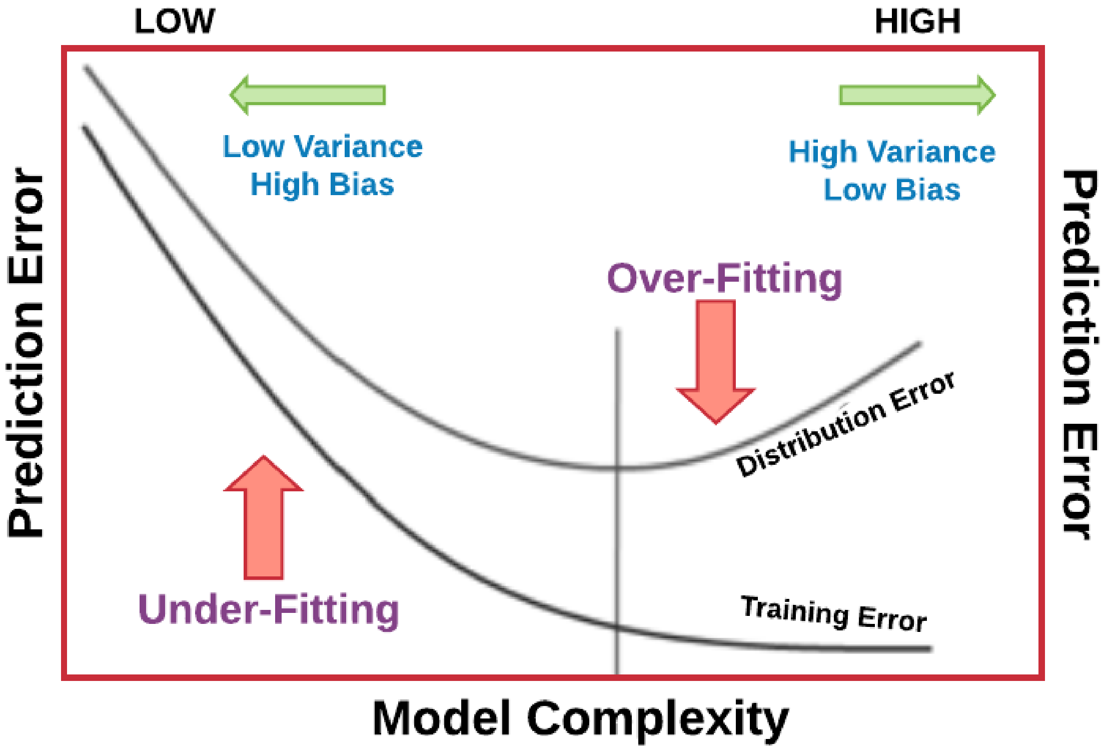


Figure 3.12: Overfitting vs. Underfitting in machine learning

3.6 Decision Trees

As discussed in the literature review, the prediction model introduced in [8, 12, 88] is affected by significant limitations in terms of its regression capability when compared to the Neural Network model; this was found to be especially true for problems involving non-linear datasets. This particular drawback, which affects the algorithms discussed above, can be attributed to the underlying assumption that the dataset is linear, which does not hold true for the dataset under investigation in this thesis (i.e. the nickel-based superalloy dataset). Linear-based models have certain advantages relative to other methods regarding their interpretation, inference, and ease of implementation [75, 89]. However, linear regression models also have significant constraints in terms of their predictive efficiency. It is therefore reasonable to move beyond the linearity assumption while still attempting to maintain as much interpretability as possible. This can be accomplished by examining more sophisticated approaches, such as tree-based models. The first step in understanding how Random Forests work is to learn decision trees. The earliest decision tree (DT) algorithm, developed by Quinlan in 1975, is known as Iterative Dichotomiser-3 (ID3) [90]. This algorithm was developed by applying the principle of Occam's Razor to arrive at a compact and practical decision tree. Quinlan [91] went on to create the C4.5 algorithm, after which the C5.0 algorithm was eventually developed.

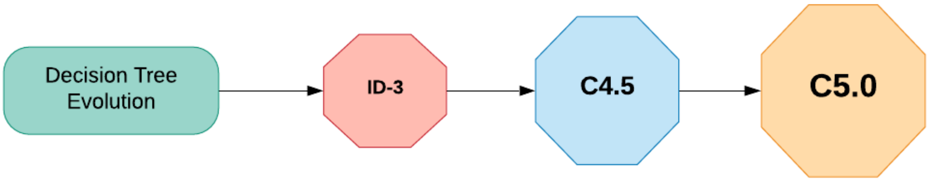


Figure 3.13: The evolution of decision tree

Figure 3.13 illustrates the evolution of a typical decision tree. It starts with the ID3 algorithm, which only works with categorical variables. The limitation of this algorithm is that it is unable to deal with missing data. Nonetheless, the C4.5 algorithm overcomes the limitations of ID3 by handling both the missing data and the quantitative variables. Another upgrade, referred to as C5.0, was developed as a significant enhancement of C4.5. The additional improvements include faster speed, more memory usage, and the ability to use a smaller decision trees. Decision trees are predictive supervised learning algorithms that utilise a layered splitting process. At each layer, the population is split into two or many groups so that all observations in the same group are homogeneous with each other, while the groups are significantly distinctive (heterogeneous). The splitting process in a decision tree follows specific criteria, such as the Gini Index, Chi-Square, Information Gain, etc. Some of the splitting rules are discussed in the next section. Depending on the output data type, decision trees can be divided into either regression or classification trees. The former is used to forecast continuous data, while the latter is employed in forecasting categorical data.

3.6.1 Decision Tree Structure

The decision tree utilises a flowchart-like layout, as depicted in Figure 3.14, where each internal node represents a condition on a sample of data. Each branch represents the response of the state, while the terminal (leaf) node corresponds to the final target response of the decision path. The terminology used in decision trees is as follows:

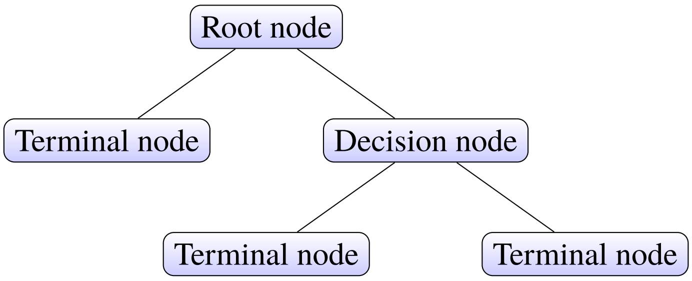


Figure 3.14: Decision tree structure

- **Root nodes:** The first node at the top of the decision tree is called the root node, which corresponds to the total population and is further split into two or more homogeneous parts.
- **Splitting:** The procedure employed for the classification of nodes into two or more sub-nodes.
- **Branch:** A sub-section of a complete tree.
- **Decision nodes:** Represent the nodes created by splitting processes, and rank lower in the hierarchy than roots with decision branches coming out of them.
- **Terminal nodes (leaf):** The bottom nodes of a decision tree with no branches coming out of them.
- **Pruning:** The opposite of the splitting procedure, in which the sub-nodes of a decision tree are removed.

3.6.2 Splitting Rules

Splitting the parent node into two or more sub-nodes (or ‘child nodes’) requires following specific rules to achieve homogeneity of data in each node. Before explaining the splitting procedures, it is first necessary to define an essential metric that drives the splitting process.

Entropy

Entropy is an indicator of unpredictability (measure of randomness) in the data being processed. Lower entropy reflects higher homogeneity of the child node, while higher entropy reflects the impurity of the child node. If the target attribute A can take on a different value C, then the entropy of the dataset (L) relative to this $C - wise$ classification is defined in Equation 3.6 as follows [89]:

$$H(L) = -\sum_{i=1}^{C} p_i \log_2 p_i \tag{3.6}$$

Where,

- L: The dataset that the entropy is calculated for.
- C: Set of classes in L.
- p_i: The ratio (proportion) of elements in class i to the total number of elements in set L.

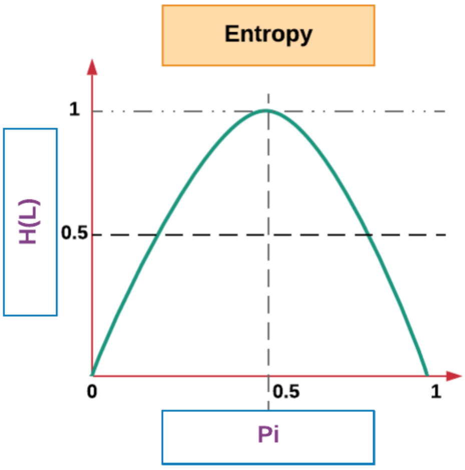


Figure 3.15: The effect of entropy in terms of proportion

The relation between the proportion and the entropy is illustrated in Figure 3.15. Entropy is defined as maximum/minimum based on the proportion value. Entropy is considered maximum when the proportion is equal to 0.5, since this represents complete randomness in the data. The minimum entropy corresponds to a proportion of either zero or one, since this represents pure homogeneity. Splitting rules in building decision trees include the following:

1. **Gini Index**: Seeks homogeneity in created subsets. A higher Gini Index reflects higher homogeneity.
2. **Chi-Square:** Statistically determines the significance of the difference between parent and child nodes. A higher Chi-Square reflects a higher significance.
3. **Reduction in Variance**: Determines splitting criteria based on the variance of child node from the parent-node. It is used for regression problems.
4. **Information Gain (IG)**: Measures the decrease in entropy after splitting. The difference is estimated between entropy before the splitting and the weighted average of the entropy' of the node after splitting the dataset. IG is calculated using Equation 3.7 [89].

$$IG(A,L) = Entropy(before) - \sum_{j=1}^{C} Entropy(j,after) \tag{3.7}$$

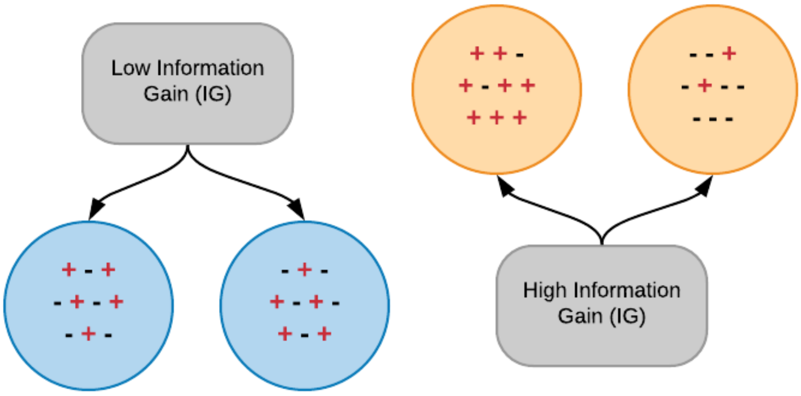


Figure 3.16: Information gain based on entropy after splitting relative to before splitting

$IG(A,L)$ is the information provided about the target function value, given the value of some other attribute A. Figure 3.16 illustrates the difference between low and high information gains, where the gain represents the entropy after splitting relative to before splitting. A high information gain represents low entropy, where it is the entropy after splitting relative to before splitting, and leads to more homogeneity. Low information gain represents high entropy, where it is the entropy after splitting relative to before splitting; this leads to a high level of randomness in the split data.

3.6.3 The Evolution of Decision Tree Construction Algorithms

The top-down induction approach to building decision trees [90] splits the feature space by recursively constituting the successor's child nodes [92]. The recursion stops if the subset at a given node includes all identical values of the output, or if no more information gain is added to the predictions when dividing.

3.6.3.1 ID3 Algorithm

The ID3 algorithm, which was introduced in 1986 by Quinlan [90] is a multi-way decision tree classification algorithm. The algorithm follows a greedy method of developing a decision tree by choosing the best categorical feature that will yield the maximum information gain (IG) (see Equation 3.7) or minimum entropy (H) (see Equation 3.6), for categorical targets.

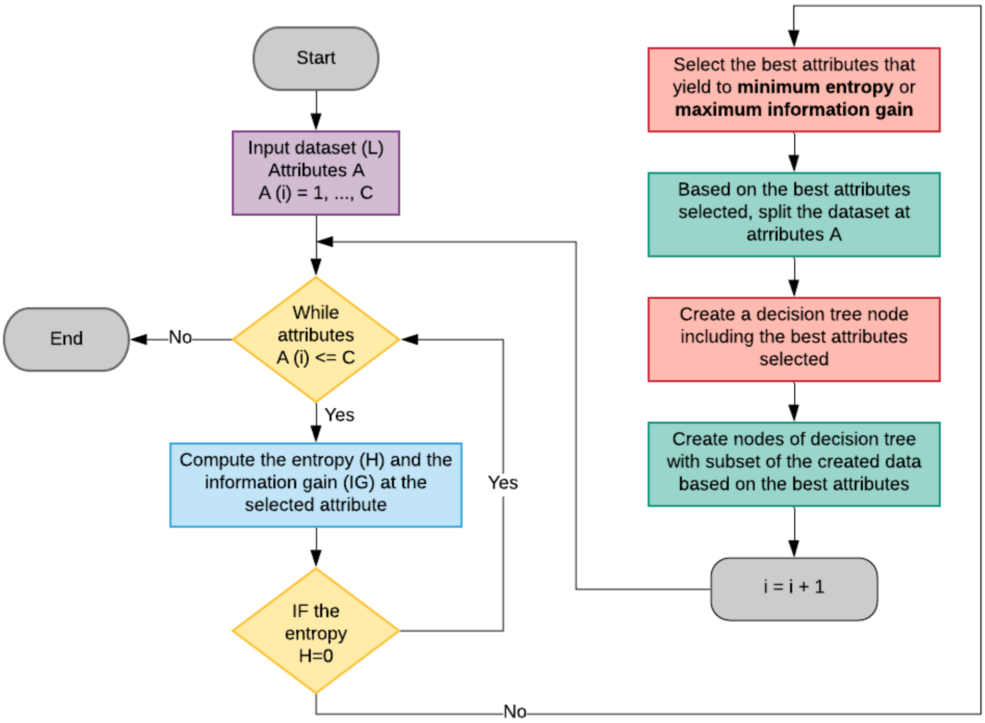

Start
Input dataset (L)
Attributes A
A (i) = 1, ..., C
While attributes A (i) <= C
No
End
Yes
Compute the entropy (H) and the information gain (IG) at the selected attribute
IF the entropy H=0
Yes
No
Select the best attributes that yield to **minimum entropy** or **maximum information gain**
Based on the best attributes selected, split the dataset at atrributes A
Create a decision tree node including the best attributes selected
Create nodes of decision tree with subset of the created data based on the best attributes
i = i + 1

Figure 3.17: The steps of the ID3 algorithm

In ID3, information gain can be calculated for each remaining attribute. The attribute with the most significant information gain is used to split the set on that particular iteration. ID3 follows the principle that a branch with an entropy of zero is a terminal (leaf) node, while any branch with entropy above zero requires additional splitting. More detailed steps are presented in Figure 3.17.

3.6.3.2 C4.5 Algorithm

C4.5 is a group of supervised learning algorithms for classification problems. Input datasets are attribute values that are described by collections of attributes and belong to one of the collection of classes. The algorithm extracts knowledge to map from attribute values to classes that can be applied to classify new unseen data. This algorithm is the successor to ID3 [91] and works by removing constraint features that should be categorical by partitioning the continuous attribute value into a discrete set of intervals.

3.6.3.3 Classification and Regression Trees [CART]

The first building block in the proposed algorithm is **CART (Classification and Regression Trees)** [69]. CART is analogous to C4.5 but differs in that it supports numerical target variables (regression). CART creates binary trees using the feature and threshold that yield the best split at each node. The conceptual framework behind the CART model is outlined below. Given training vectors $X_i \in R^n$, $i = 1, .., n$ and a label vector $Y \in R^l$, a decision tree recursively partitions the space such in a way that samples with identical labels are grouped together. Assume the data at node m is denoted by Q. For each dataset split $\theta = (j, t_m)$, including a feature j and threshold t_m, partition the data into $Q_{left}(\theta)$ and $Q_{right}(\theta)$ subsets using Equation 3.8 [93].

$$\begin{aligned} Q_{left}(\theta) &= (X, Y) | X_j \leq t_m \\ Q_{right}(\theta) &= Q \setminus Q_{left}(\theta) \end{aligned} \tag{3.8}$$

The impurity at m is estimated by an impurity function $H()$, the choice of which depends on the task being solved (either classification or regression). The equation is defined as follows [93].

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)) \tag{3.9}$$

The parameters for minimising the impurity are chosen as follows [93]:

$$\theta^* = \operatorname{argmin}_\theta G(Q, \theta) \tag{3.10}$$

This process is repeated recursively for subsets $Q_{left}(\theta^*)$ and subsets $Q_{right}(\theta^*)$ until the maximum permitted depth is reached, where $N_m \leq \min_{samples}$ or the output in the node is homogeneous.

Part 1: Classification Criteria in CART

A target is a classification outcome that takes on the values 0, 1..., $(C-1)$, for node m, representing R_m a region with N_m observations [93]. Let p_{mC} be the proportion of class C observations in node m, as follows:

$$p_{mC} = 1/N_m \sum_{X_i \in R_m} I(Y_i = C) \tag{3.11}$$

Common measures of impurity are the Gini Index (using Equation 3.12) or entropy (using Equation 3.13) as below:

$$H_g(X_m) = \sum_C p_{mC}(1 - p_{mC}) \tag{3.12}$$

$$H_e(X_m) = -\sum_C p_{mC} \log_2(p_{mC}) \tag{3.13}$$

Part 2: Regression Criteria in CART

Every node m describes a zone with R_m observations, all with continuous output values. Standard regression metrics for evaluating possible splits include the mean squared error (MSE), which minimises error by using the average value at the terminal nodes, or least absolute deviations (LAD), which decreases error by using the median value at the terminal nodes. MSE and LAD are evaluated using the following equations [93]:

Mean Squared Error (MSE):

$$\begin{aligned} \bar{Y}_m &= \frac{1}{N_m} \sum_{i \in N_m} Y_i \\ H(X_m) &= \frac{1}{N_m} \sum_{i \in N_m} (Y_i - \bar{Y}_m)^2 \end{aligned} \tag{3.14}$$

Least Absolute Deviations (LAD):

$$\begin{aligned} median(Y_m) &= \underset{i \in N_m}{\mathrm{median}}(Y_i) \\ H(X_m) &= \frac{1}{N_m} \sum_{i \in N_m} |Y_i - median(Y_m)| \end{aligned} \tag{3.15}$$

Where,

- X_m - A subset of the training data at node m.
- $H(X_m)$ - The impurity function $H()$ at node m.
- N_m - The number of observations in the node m.
- Y_i - The output of observation i of node m.
- $\bar{Y}_m$ - The mean of output at node m.

The building blocks for every node in the entire decision tree can be formulated using on the above formula. Each node is created by splitting the data into subsets. Estimating the location of the best split, as in Figure 3.18, is achieved by following the steps described in Algorithms 2 and 3. This block of code is then called recursively to build the whole tree, as described in Algorithm 1.

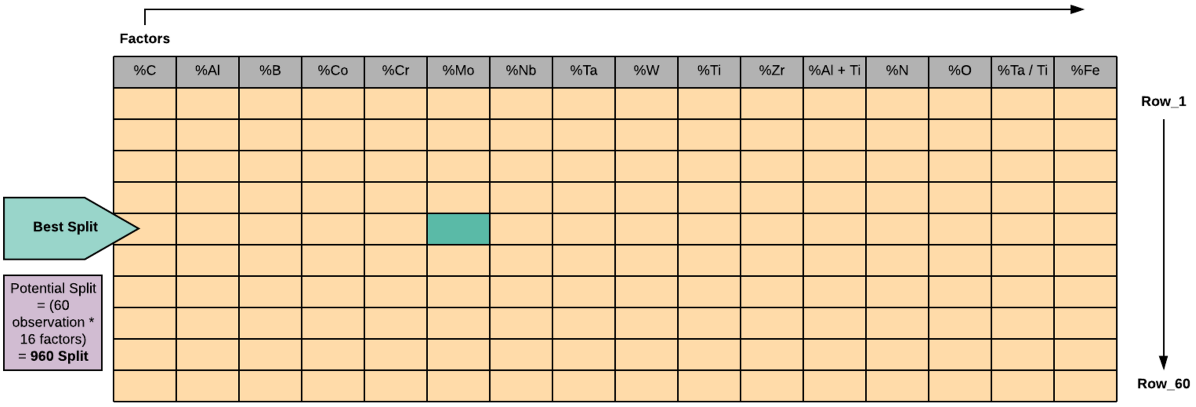


Figure 3.18: Possible split locations for nickel-based superalloy dataset

The tree splitting phases in the regression context could be summarised in three major steps, as follows: First, find each feature's best splits, which minimise the impurity function that contain the best split (one in each feature). Secondly, find the node's best splits that minimise the impurity function, which is the minimum weighted average. Finally, split the node using the best node split and repeat the first steps until the stopping criterion is met. In more detail, the steps of splitting CART used in this thesis are as follows:

1. Initially, create a decision tree (T) with root node (m_0), which contains the entire dataset (L).
2. Decision trees are trained by dividing the data into two parts recursively, based on the selected splitting rules.

3. The nickel-based superalloy dataset has 16 factors with 60 data observations in each column, meaning that a sum of 960 divisions is probable; accordingly, each of such splits that is best for the results should be found.

Regarding the first feature (j), the steps are as follows:

1. Begin by searching for the best split (θ_j^*) in each feature. Starting from the first feature at the zero index location ($t_{m,s}$).
2. Sort the node samples from the smallest data point until the largest data point.
3. Next, compute the mean square error (MSE) for the remaining data points of the feature and initialise the mean square error of the zero index in the feature equal to zero.
4. Therefore, the weighted average score of the MSE of the two subsets needs to be decreased, after which the greedy method is utilised to find a split by splitting the data into two halves with each observation in the column and measuring the weighted average score of the MSE of the two halves to determine the minimum as follows:
    - Initially, increment the index location ($t_{m,s+1}$), and start updating the MSE of the data before and after this index location.
    - The best split (θ_j^*) will be at the location that results in the minimum weighted average score for this feature; this will be the split with the lowest ranking score.
5. The same steps are considered later for all training features and are compared greedily in order to find the best (minimum) score.
6. Accordingly, the best split in the node (θ^*) is chosen based on the minimum MSE from the entire group of features. The data will be partitioned into two child nodes according to the best split (θ^*) location.
7. Finally, the previous steps are repeated recursively at each node until the stopping criterion (γ) is satisfied.

Algorithm 1: Greedy Induction of a Binary Decision Tree [93]

```
Function BuildDecisionTree($L$):
    Create a decision tree $T$ with root node $m_0$
    Create an empty stack $S$
    $S$.push($m_0, L$)
    while $S$ is not empty do
        $m, Q = S$.pop( )
        if $\gamma$ is met for $m$ then
            $\hat{Y}_m$ = average of the node output
        else
                                        // Find the split that minimise the 'impurity decreases'
            $\theta^* = \text{argmin}_\theta\, G(Q, \theta)$
            Partition $Q$ into $Q_{left} \cup Q_{right}$ according to $\theta^*$
            Create the left $m_L$ and the right child node $m_R$ of $m$, respectively
            S.push($m_R, Q_{right}$)
            S.push($m_L, Q_{left}$)
        end
    end
    return $T$
End Function
```

Algorithm 2: Finding the Best Split θ^* for Partitioning Q [93]

```
Function FindBestSplit $\theta^*(Q)$:
    $G(Q, \theta) = \infty$
    for $j = 1, ..., n$ do
        Find the best binary split $\theta_j^*$ defined on $X_j$
        if $G(Q, \theta_j^*) < G(Q, \theta)$ then
            $G(Q, \theta) = G(Q, \theta_j^*)$
            $\theta^* = \theta_j^*$
        end
    end
    return $\theta^*$
End Function
```

Algorithm 3: Finding the Best Split θ_j^* on X_j for Partitioning Q [93]

```
Function FindBestSplit $\theta_j^*(Q, X_j)$:
    $G(\theta, Q) = \infty$
    $s = 0$
    $i = 1$
    $t_{m,s} = -\infty$
    Initialise the MSE for $m_L$ to 0
    Initialise the MSE for $m_R$ to those of $i(m)$ by computing the MSE
    Sort the node samples $Q$ such that $X_{1,j} \leq X_{2,j} \leq ... \leq X_{N_{m,j}}$
    while $i \leq N_m$ do
        while $i + 1 \leq N_m$ do
            $i = i + 1$
            if $i \leq N_m$ then
                $t_{m,s+1} = \frac{X_{i,j}+X_{i-1,j}}{2}$
                Update the MSE from $t_{m,s}$ to $t_{m,s+1}$
                if $G(\theta_j^{t_{m,s+1}}, Q) < G(\theta, Q)$ then
                    $G(\theta, Q) = G(\theta_j^{t_{m,s+1}}, Q)$
                    $\theta_j^* = \theta_j^{t_{m,s+1}}$
                end
                $s = s + 1$
            end
        end
    end
    return $\theta_j^*$
End Function
```

3.7 Advantages and Disadvantages of Decision Trees

Out of all tree-based methods, decision trees have been the most successful, primarily because of several factors that make them useful in practice. These factors include the following:

- Known to be non-parametric, in that they do not require any prior assumptions to model arbitrarily complicated connections between outputs and inputs.
- Capable of implementing a selection of features and making them robust to variables that are noisy or irrelevant.
- Ability to minimise the error introduced by outliers in labels.
- Easy to use; detailed statistical knowledge is not necessary.
- Provide a way to find homogeneous subsets of data in heterogeneous datasets.
- Can be categorised into 'Categorical' or 'Regression' trees for categorical and continuous problems respectively; can therefore work with a variety of data types.

Equally important, however, is that decision trees have drawbacks. The models tend to suffer from one of two errors, specifically bias and variance. Interestingly, a simple model will tend to exhibit more bias and less variance; that is, it is likely to exhibit more deviation of the predicted answer from the actual answer and less change in predicted response with changing samples. The opposite is true of complex models. Hence, a good model should consider this trade-off between the two extremes. This trade-off is accomplished through the use of ensemble methods, which include Bagging, Boosting, and Stacking [68]. Bagging is performed by applying multiple features/categories to different samples from the same dataset, then combining all features/categories. In so doing, the variance is reduced by $1/n$, where n denotes the number of features/categories. Random Forests are an implementation of Bagging. By creating multiple trees using different feature subsets, the model chooses the answer that gets most of the votes for categorical problems or, alternatively, takes an average of the outputs for regression problems. More importantly, many state-of-the-art [94] and modern algorithms (such as forests of randomised trees) use decision trees as the foundation to build blocks for forming larger models. As the objective is to extract the causal relationship from the process knowledge, it is necessary to understand all algorithmic details of a single decision tree. One shortcoming of decision trees relates to their use of a greedy algorithm, meaning that only the best solution for the current node is considered with no regard for future solutions. They also tend to overfit the training data, which can lead to poor performance, mainly when the model is implemented on a set of new data. This occurs most commonly when the terminal nodes represent small subsets of the training data while the tree has grown enormously. At the same time, it is possible to overcome this challenge by employing a process known as pruning. This process involves eliminating terminal nodes that are likely connected to noise in the data, thus simplifying the tree. As a result, the effects of over-fitting are minimised, while at the same time, the predictive power of the model is enhanced. However, there is still a challenge involved, in that the process of pruning is somewhat subjective; thus, pruning a tree can be cumbersome, thereby yielding the lowest error in a given test dataset. The other method that can be used to overcome this challenge is to constrain the tree sizes. Restricting tree size can be done in one of two ways. The first way involves setting a threshold on a minimum number of samples to allow a node to split; the second involves limiting the depth of the tree to a specific value. In the next section, the critical measurements utilised in the evaluation of regression models are considered.

3.8 The Bias-Variance Trade-Off

Bias relates to the error created by the comparison of a real-life problem, which could be incredibly complex, to a somewhat simplified one. It is unlikely to achieve a precise forecast by using a restrictive/simple algorithm in cases where the real relationship is extremely complicated. On the other hand, variance corresponds to the sum by which the estimation of $f(X)$ will shift if measured using a specific collection of training data. Ideally, however, the approximation for $f(X)$ will not differ between training sets. Nevertheless, if the procedure exhibits significant variance, even minor changes in the dataset could lead to substantial changes in $f(X)$.

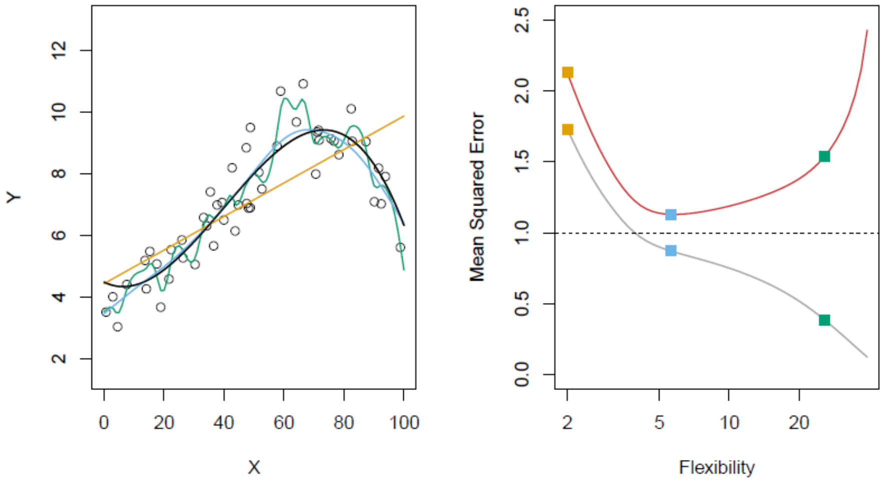


Figure 3.19: The bias-variance trade-off [75]

As shown in Figure 3.19, the yellow curve is a linear pattern; moreover, the blue one is a projection of a marginally non-linear model, while the green curve is a predictor of a strongly non-linear/flexible model. The optimum model minimises both bias and variance error, as seen in the black curve. The U-shape found in the MSE test curves in Figure 3.19 tends to be the product of two conflicting properties of statistical learning methods. It can be seen that the predicted MSE test, for a defined value of X, can often be broken down into the sum of the fundamental quantities: the variance of $\hat{f}(X)$ (representing the red curve), and the squared bias of $\hat{f}(X)$ (representing the grey curve). High system complexity has been shown to result in more noise in the learning range. In the case of a given dataset, the lowest MSE level provides an important statement regarding the bias and variance of the error types. Therefore, with improved variability, bias can be reduced more than variation increases. While there is no further decrease in bias at any stage, the variation continues to increase rapidly due to overfitting. Moreover, each adjustment in the data collection can yield a new result, which is quite reliable, by utilising a mathematical approach that aims to fit the data points very carefully. The basic theory is as follows: when a predictive approach works to predict data points more precisely, or as a more robust system is used, bias decreases; however, variation also increases.

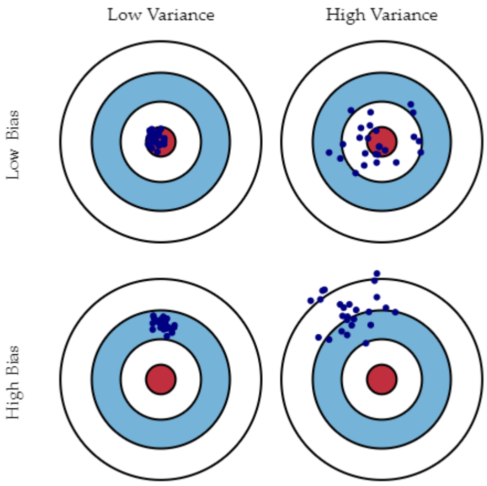


Figure 3.20: Graphical illustration of bias and variance [95]

The mean squared error (MSE) of a statistical model can be represented as the sum of the squared bias of its forecasts, the variance of such predictions, and the variance of any error expression. Because both squared bias and variance are non-negative, while other errors that catch randomness in the data will not be influential, MSE is reduced by decreasing the variance and bias of the model. As Figure 3.20 indicates, the centre of the target is a model that accurately forecasts the correct values. The figure present four separate instances reflecting variations of high and low bias and variation. High bias occurs where all dots are far from the bulls-eye, while high variance is where the dots are more dispersed. To minimise the prediction error, a predictive learning approach must be selected that concurrently maintains low variance and minimal bias.

Trade-off Reflection on the Proposed Algorithm

A decision tree has a high variance and low bias. As shown in Figure 3.20, the optimal model should have the lowest possible values of both variance and bias. Ensemble methods introduce a robust approach to reducing the variance error in a decision tree by building randomised trees from the original data to create a forest of random decision trees. Moreover, a novel technique is introduced in chapter four for estimating the optimum number of trees to further minimise the variance error.

3.9 Ensemble Methods

In this chapter, a generalised framework for error decomposition of any statistical learning model $f(X)$ is presented. This framework was first introduced by Geman [53] as a tool for diagnosing model underfitting and overfitting of the model. In the regression context, the MSE is decomposed into bias and variance terms. Bias measures the discrepancy within the average prediction and the forecasting of the data model, while the variance term estimates the variability of the predictions. As can be observed from the graphic presented in Figure 3.20, a reasonable approach to reducing generalisation error would drive down the prediction variance and either keep the bias the same or avoid increasing it. Ensemble approaches aim to achieve this by building several uncorrelated different estimators from randomised subsets of features of a single learning series L, then aggregating the predictions of those models to arrive at the prediction of the ensemble (see Figure 3.21).

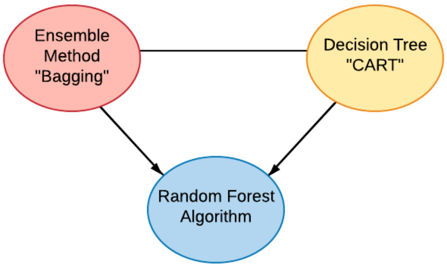


Figure 3.21: Ensemble methods of the Random Forest algorithm

3.10 Random Forest Algorithm (RF)

The black-box version of the Random Forest method explains the challenge associated with adequately analysing Random Forests. The Bagging and the CART split criterion are among the essential ingredients of the RF approach and play crucial roles. In essence, the Bagging technique involves two processes [96]: namely, aggregation and bootstrapping. Bootstrapping is a statistical procedure whereby a random sample is selected from a single dataset using the replacement sampling method. After the selection of random samples, the learning algorithm is then run. In the next step, the model produces several predictions that are subsequently aggregated to produce the final prediction, which considers all possible outcomes. In other words, Bagging is a general aggregation (as shown in Figure 3.22) that is among the most effective computationally intensive mechanisms for enhancing unstable estimates, particularly for large datasets with high dimensionality. In large datasets of this kind, it is impossible to find a suitable model in one step due to the sheer scale and complexity of the problem. It is also essential to understand that the CART-split criterion is based on Breiman's influential CART program, which is employed in the establishment of the individual trees to select the splits that are perpendicular to the axes [16]. Indeed, the best split at each node of each tree is chosen by maximising the CART-splitting measure based on either the squared prediction error (for regressions) or Gini impurity (for classification). However, both the CART-splitting scheme and bagging are difficult to evaluate even with rigorous mathematical processes, although they play essential roles in the mechanism of Random Forests, which is the reason why theoretical studies tend to prefer simplified versions of the first algorithm. Under these circumstances, what typically happens is that a simpler split protocol replaces the CART-split selection and the bagging step is simply ignored. Moreover, each terminal node of the individual trees in Breiman's forests consists of some number of observations (usually between one and five).

Algorithm 4: Random Forest Training Procedure for Regression or Classification [89]

```
Input: Training Data ($L$), Number of Trees ($ntree$)
Output: Forest of trained estimators(trees)
RF=[]
for $b = 1 : B$ do
    Draw a bootstrap sample $b$ from the training data $L$.
    Grow a tree $T_b$ using the bootstrapped sample $b$, by implementing Algorithm 1.
    Append tree in the RF.
end
return RF
```

Random Forest is a hybrid algorithm first introduced by Breiman [17]. When used for discrete predicted results, it is referred to as Random Forest classification, while for continuous values, it is called Random Forest regression. Algorithm 4 explains the procedure of training an Random Forest algorithm. For prediction using the trained model with a new unseen observation x, the estimation of each tree (as in Algorithm 5) would be consolidated based on the output form, as follows [89]:

- Regression

$$\hat{f}(x) = \sum_{b=1}^{B} T_b(x) \tag{3.16}$$

- Classification

$$\hat{f}(x) = MajorityVote\{T_b(x)\}_1^B \tag{3.17}$$

Algorithm 5: Prediction of the Output Value $\hat{Y} = \hat{f}(X)$ in a Decision Tree [89]

```
Function Predict($X$):
    $m = m_0$
    while $m$ is not a terminal node do
        $m$= the child node of $m$
        $\hat{Y}_m$ = Average response in $m$ node
    end
    $\hat{Y}_m$ = Average response in the node
    return $\hat{Y}_m$
End Function
```

The steps involved in using Random Forest for training and testing are illustrated in Figure 3.22 and can be summarised as follows:

1. In the training phase, the data (L) is randomly bootstrapped to generate bags (b), based on the number of estimated trees ($ntree$), for different subsets with replacement. The subsets created from this bootstrapping process are then used to build a regression tree (T_b).

2. The trained trees are appended to the forest.

3. In the testing phase, the unseen observations (x) acquired during training are used to predict the response and test the predictions generated from the decision tree by using the validation set (by splitting the data and/or using the k-fold cross-validation).

4. The prediction of the process response is calculated as follows:

   - Starting from the root node (m_0), each tree follows the decision path until the terminal node is reached.
   - The predicted value will be the average of the response value at terminal node ($\hat{Y}_m$).
   - Finally, the average prediction produced from each tree is computed.

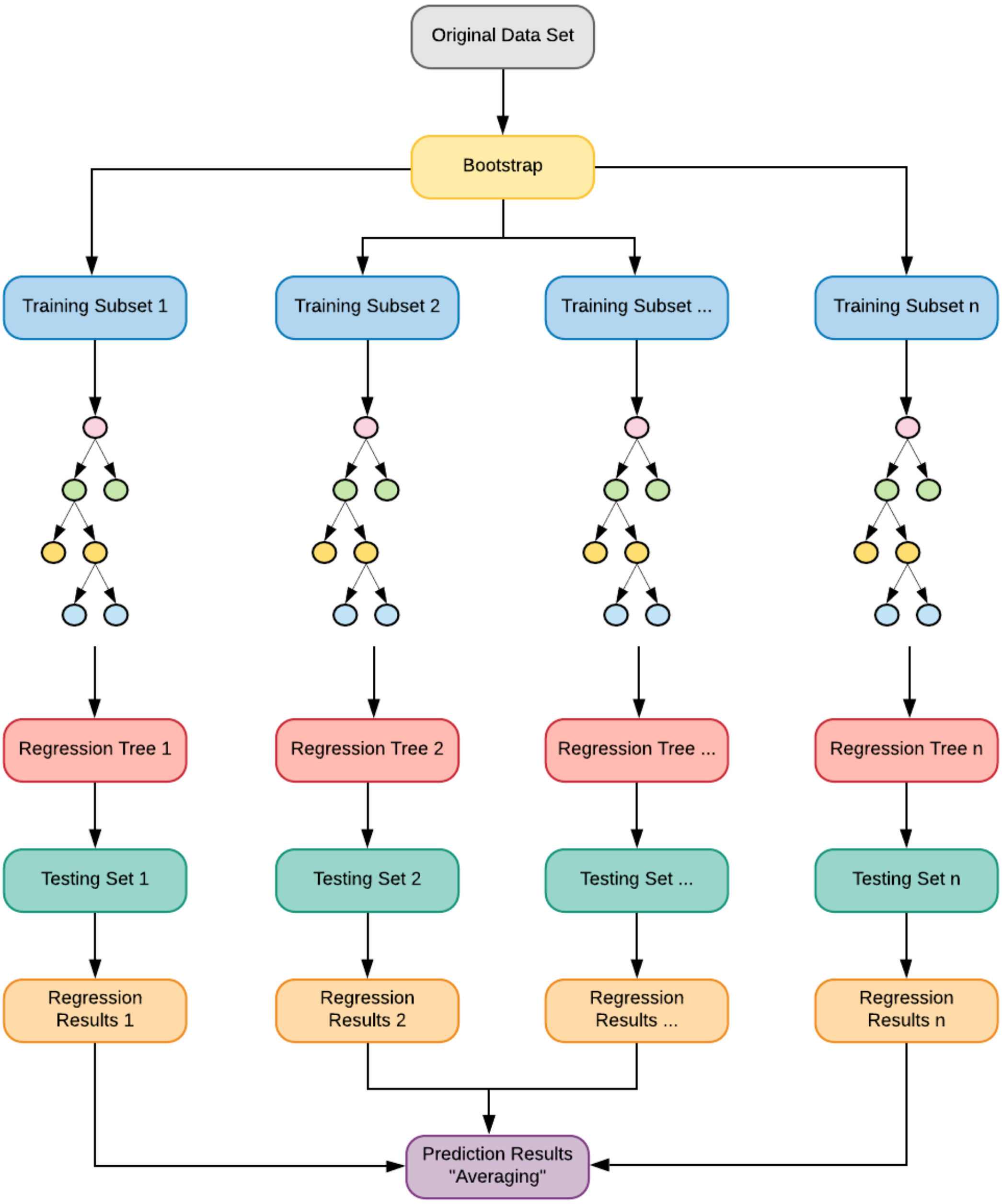


Figure 3.22: Random Forest algorithm procedure

3.11 Encoding of Categorical Features

Encoding [93] is a pre-processing step used to encode the categorical features into numeric values. In the current work, a simple encoding feature called **LabelEncoder** is used, which is suitable for the mixed dataset provided in this case study. The algorithm simply collects all attributes (A) for each feature, then computes the amount in $nLabel$ that represents the number of categories present in the attributes. The encoded value is equal to some value between zero and ($nLabel - 1$). Finally, it creates (quantitative) features based on the total $nLabel$ numbers established in the previous steps. The detailed steps involved in LabelEncoder are illustrated in Figure 3.23.

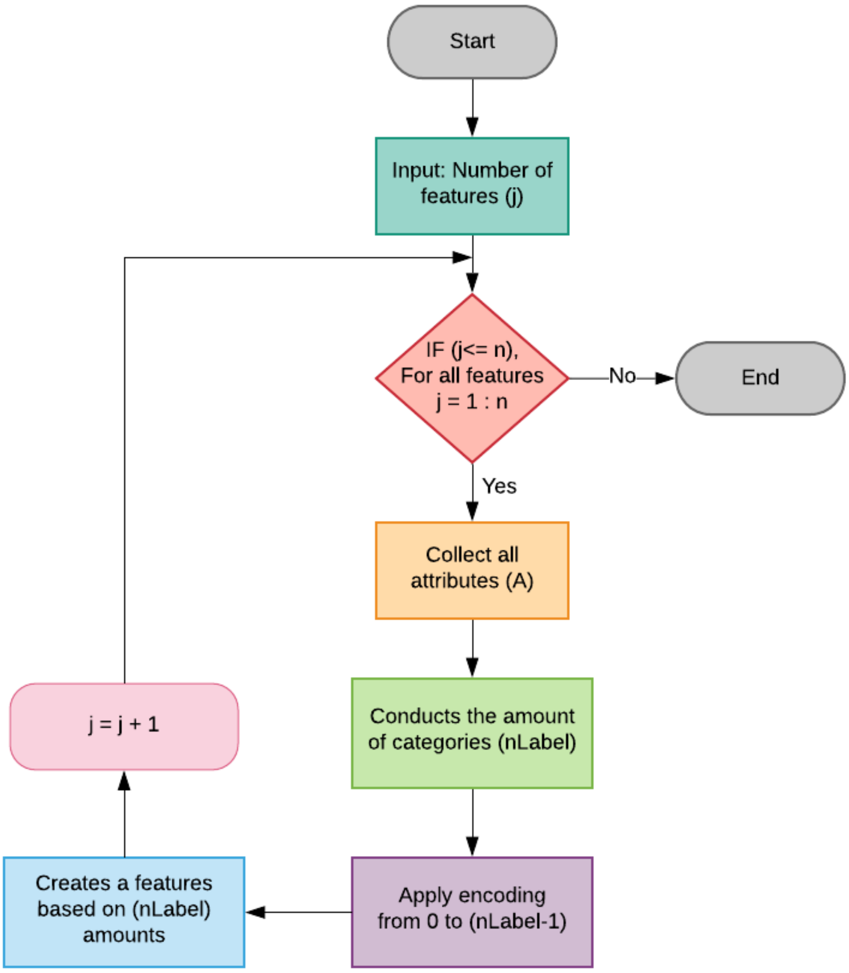

Start
Input: Number of features (j)
IF (j<= n), For all features j = 1 : n
No
End
Yes
Collect all attributes (A)
Conducts the amount of categories (nLabel)
Apply encoding from 0 to (nLabel-1)
Creates a features based on (nLabel) amounts
j = j + 1

Figure 3.23: Steps of categorical feature encoding using the LabelEncoder algorithm

3.12 Missing Data Imputation

As discussed above, manufacturing data collection is confronted with the issue of missing data. Missing data imputation techniques are accordingly applied to solve this issue. However, the majority of such imputation methods are limited to one data format (either continuous or categorical). For mixed-type data, continuous and categorical data is typically handled separately; therefore, these techniques cannot obtain the potential non-linear relationships between variables. Considering the nature of the Random Forest, which can provide either a classification or regression capability, the next section discusses the missForest approach and how it can be used to handle the mixed types of variables simultaneously.

missForest Algorithm

missForest [15] is an iterative imputation approach based on the Random Forest algorithm. It utilises a built-in Random Forest out-of-bag error calculation, such that an imputation error may be calculated without the need for a test set. This approach thus outperforms other imputation approaches, particularly in cases involving non-linear interactions, computational performance, and high-dimensional data. The imputer continues to arrange the data columns based on the least amount of missing values. The first column is called the candidate column ($y_{mis}^{(s)}$). The absent values in the remaining non-candidate columns ($x_{mis}^{(s)}$) are filled using an iterative method (initial guess). The initial approximation is the mean value for the quantitative columns and the mode for the categorical columns. After that, the imputer trains a Random Forest model with the candidate column as the response variable ($y_{obs}^{(s)}$) and the non-candidate columns as the features ($x_{obs}^{(s)}$). The training set requires training matrix rows in which the candidate column values are not missing. Upon fitting, the missing values of the candidate column are imputed ($y_{mis}^{(s)}$) using prediction from the trained Random Forest through ($x_{mis}^{(s)}$). The rows of the non-candidate columns act as the data input for the trained model. Thus, the imputer switches to the next nominee column with the second-smallest number of missing values, and the cycle repeats itself with each column with a missing value over several iterations before the stopping condition (γ) is satisfied. The stopping criterion relies on the 'difference' within the imputed arrays between (X_{new}^{imp} and X_{old}^{imp}) over successive iterations. The stopping criterion is reached as long as the gap between the newly imputed given dataset (X_{new}^{imp}), and the prior one increases for the first time concerning the selected variable forms (X_{old}^{imp}). In this case, the various set of continuous variables ∇RN is introduced by Equation 3.18. For this set of categorical variables, ∇F is introduced by Equation 3.19, where N_{mis} is the number of missing values in the categorical variables [15],

$$\nabla RN = \frac{\sum_{j \in RN} (X_{new}^{imp} - X_{old}^{imp})^2}{\sum_{j \in RN} (X_{new}^{imp})^2} \tag{3.18}$$

$$\nabla F = \frac{\sum_{j \in F} \sum_{i=1}^{n} I_{X_{new}^{imp} \neq X_{old}^{imp}}}{N_{mis}} \tag{3.19}$$

After imputing the missing values, when the Random Forest matches the observed component of the equation, an out-of-bag (OOB) error calculation is included for that component. After the stopping criterion γ has been met, the correct imputation errors are averaged over the set of variables of the same type. The performance of this estimation is then assessed by measuring the absolute discrepancy between the actual imputation error and the OOB imputation error calculation over both simulation tests. Algorithm 6 is a depiction of the missForest algorithm procedure as described below, where the variables represent the following:

- $y_{mis}^{(s)}$ - The missing values of observation X_s;
- $y_{obs}^{(s)}$ - The observed values of observation X_s;
- $x_{mis}^{(s)}$ - The variables other than X_s with observations $i_{mis}^{(s)}$;
- $x_{obs}^{(s)}$ - The variables other than X_s with observations $i_{obs}^{(s)} = \frac{1,\ldots,n}{i_{mis}^{(s)}}$.

Algorithm 6: Impute Missing Values with missForest Technique [15]

```
Result: $X_{new}^{imp}$ an $n \times r$ matrix, stopping criterion $\gamma$
Make initial guess for missing values
$g \leftarrow$ vector of sorted indices of columns in $X_{old}^{imp}$    // w.r.t. increasing amount of missing
while $\gamma$ is not met do
    $X_{old}^{imp} \leftarrow$ store previously imputed matrix
    for $s$ in $g$ do
        Fit a Random Forest: $y_{obs}^{(s)} \approx x_{obs}^{(s)}$
        Predict $y_{mis}^{(s)}$ using $x_{mis}^{(s)}$
        $X_{new}^{imp} \leftarrow$ update imputed matrix, using predicted $y_{mis}^{(s)}$;
    end
    Update $\gamma$
end
return Imputed matrix $X_{new}^{imp}$
```

3.13 Data Augmentation Techniques

The randomised bootstrapped subsets approach used in RF produces more trees that are likely to be uncorrelated and thus reduce overfitting, as well as more in-depth knowledge discovery from the process data. However, these benefits come at a cost. Having a large dataset is crucial for the performance of the RF model. A small sample size decreases statistical power and increases overfitting, resulting in weaker knowledge discovery and low prediction accuracy. The sample size acts as a control parameter for the degree of randomness in the forest. An increased sample size in a smaller number of trees tends to produce correlated trees; by contrast, decreasing the sample size in a larger number of trees will produce trees that are more likely to be uncorrelated and that will thus tend to prevent overfitting, although typically at the expense of model performance. A beneficial side effect is that smaller sample sizes reduce the time required to train the model. As noted above, a large dataset is vital to the good performance of a machine learning model. Accordingly, the model performance could be enhanced by augmenting the data. Data augmentation is a technique used for the synthetic generation of new training data from raw training data.

As shown in Figure 3.24, the nickel-based superalloy historical data available is limited to 60 observations only. As the RF model cannot be trained with this limited number of observations, data augmentation is a crucial task here. The primary considerations when choosing the augmentation technique are the types of data and the nature of the relationship between process factors (in other words, whether or not there is non-linearity in the data). The problem faced in this study is the need for a robust technique that is capable of handling non-linear mixed datasets. The most common approaches in the literature are discussed, beginning from the simple random oversampling method and progressing to the more advanced technology developed especially for balancing classification problems, referred to as the Synthetic Minority Over-sampling TEchnique (SMOTE) [14]. This section of the study concludes with the realisation that it is important to consider developing a more robust technique capable of augmenting non-linear process data, especially when dealing with regression problems. The detailed algorithms for each method, as described in detail in the next sections, are as follows:

- Details and results of the random oversampling technique are described in Section 3.13.1.
- The steps of the SMOTE are described in Section 3.13.2.
- A novel augmentation technique for non-linear regression, including the algorithm and results, is discussed in chapter four.

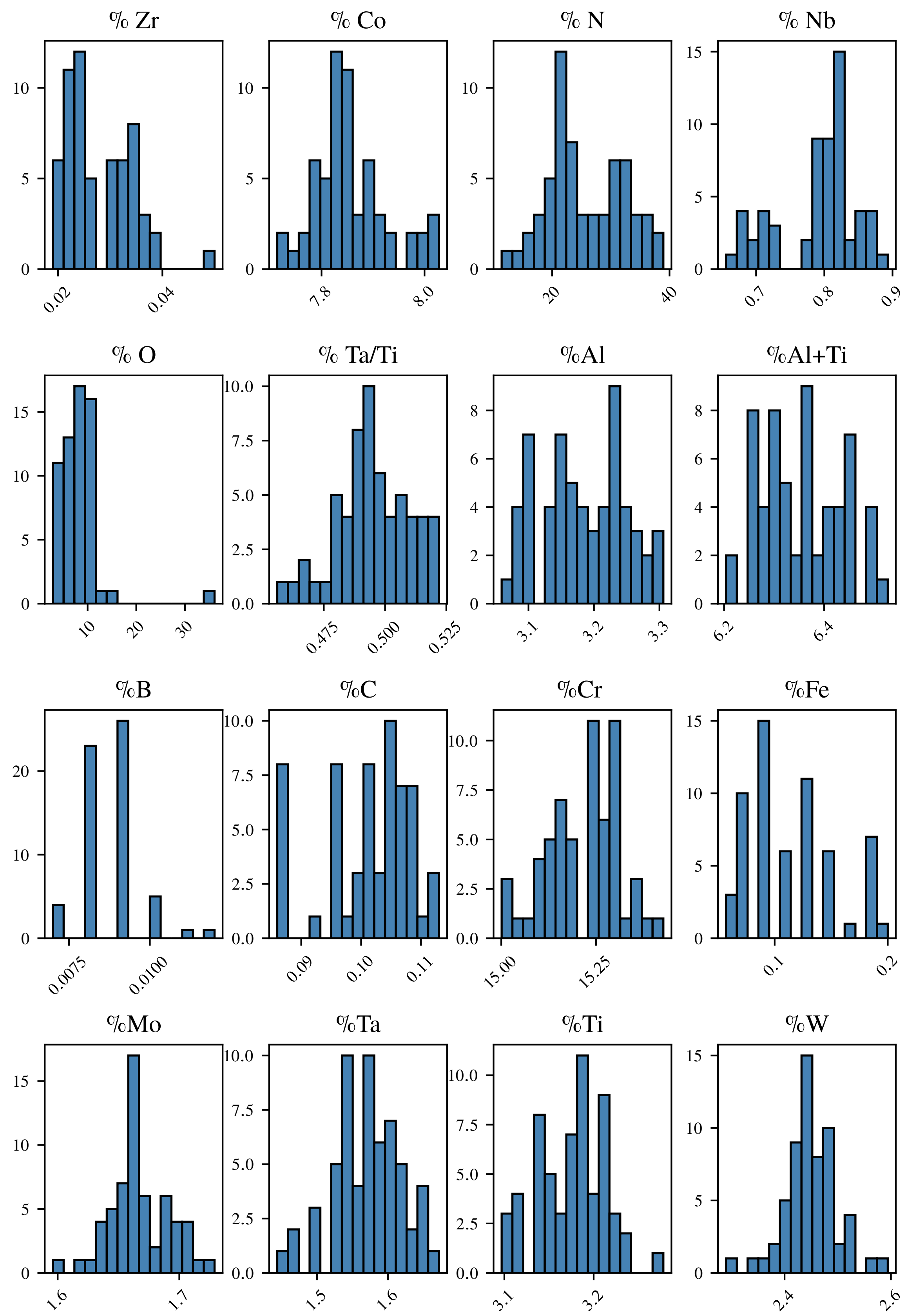


Figure 3.24: Nickel-based superalloy process inputs distributions

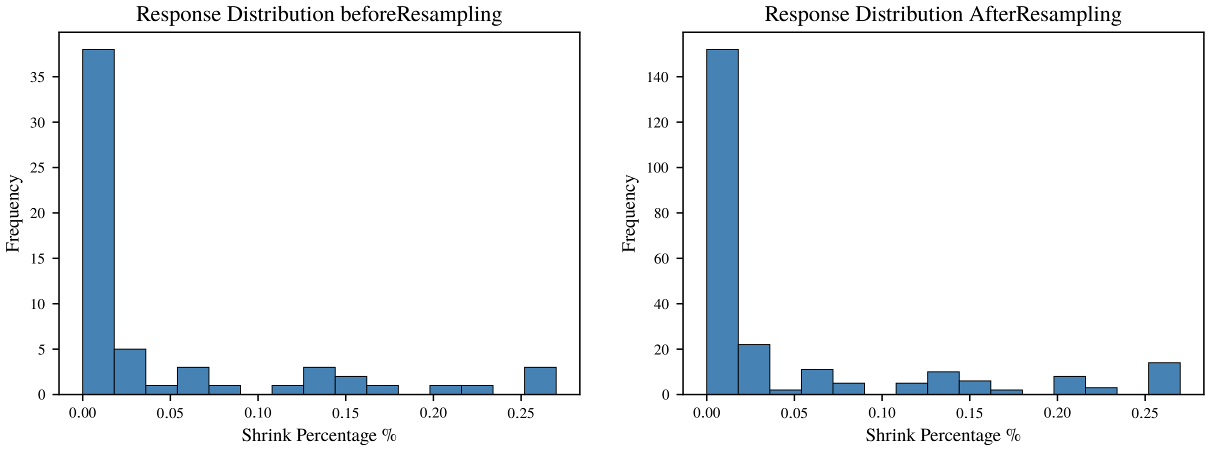
3.13.1 Random Oversampling

The random oversampling technique [97] is a simple technique based on one-step bootstrap resampling. First, a random index (i) is generated that contains between 1 and n original samples. Next, a lookup for this index is conducted in the original dataset (L), after which the corresponding observation is obtained and saved in a new data frame (L_{new}). This operation is repeated until the new sample size (n_{new}) required is reached. Algorithm 7 and the schematic diagram in Figure 3.26 provide an overview of the random over-sampler.

Algorithm 7: Random Oversampling Algorithm [97]

```
Input: New sample size $n_{new}$
Output: $L_{new}$
Data: Original set $L$
while $i \neq n_{new}$ do
    Generate random index from $1 : n$          // n is an original sample size $L$
    Look up in $L$ by using the generated index
    Save the selected index in $L_{new}$
    $i = i + 1$
end
```

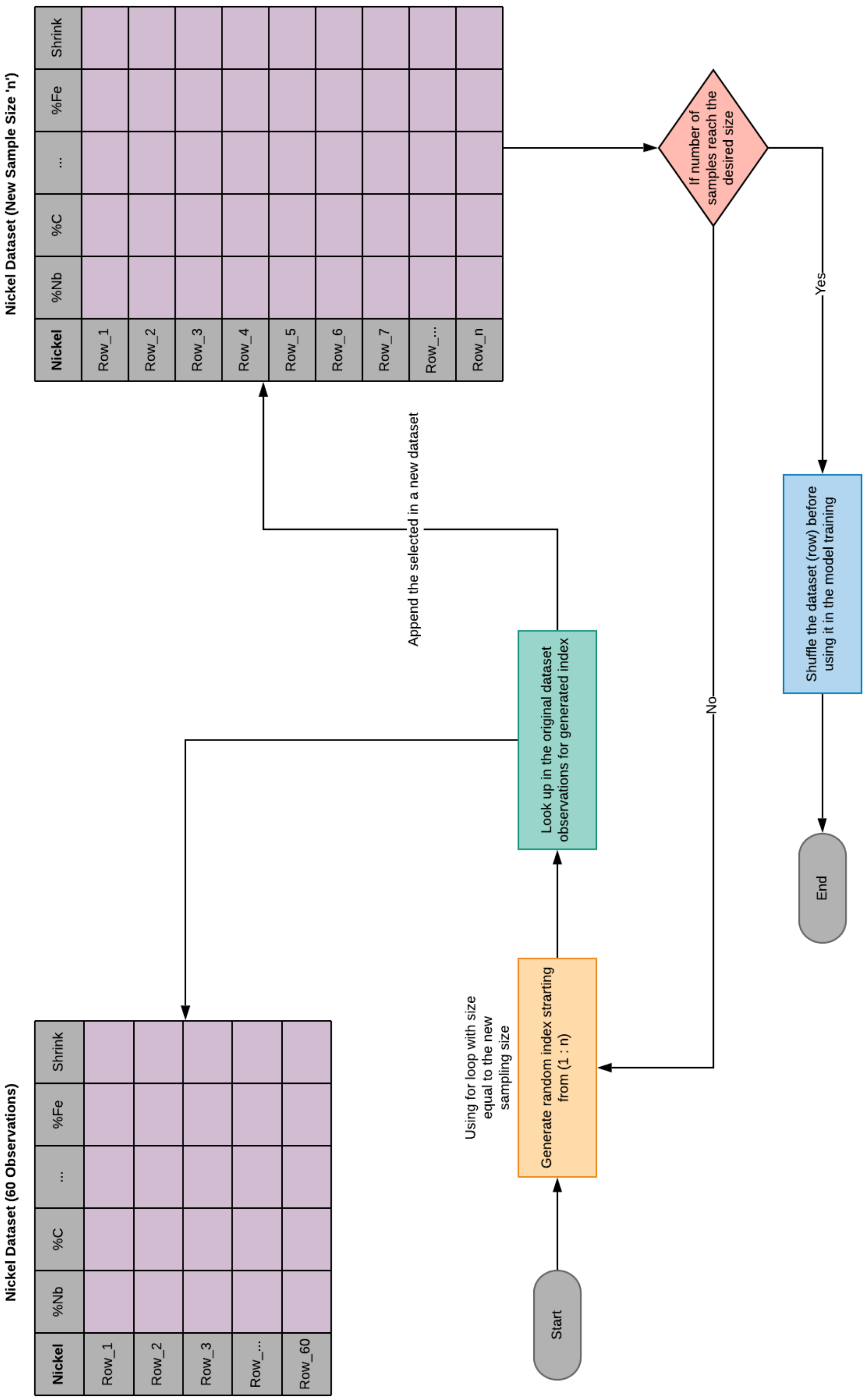
As can be seen from Figures (3.24, 3.25b, and 3.27), the major drawback of this technique is that it does not create new examples, but only draws examples randomly from the original data. Repeating observations increases the likelihood of overfitting during model training.



(a) Before re-sampling

(b) After re-sampling

Figure 3.25: Response distribution by random oversampling technique for nickel dataset



Nickel Dataset (60 Observations)
Start
Using for loop with size equal to the new sampling size
Generate random index strating from (1 : n)
Look up in the original dataset observations for generated index
Append the selected in a new dataset
Nickel Dataset (New Sample Size 'n')
If number of samples reach the desired size
No
Yes
Shuffle the dataset (row) before using it in the model training
End

Nickel Dataset (60 Observations)

Nickel	%Nb	%C	...	%Fe	Shrink
Row_1					
Row_2					
Row_3					
Row_...					
Row_60					

Nickel Dataset (New Sample Size 'n')

Nickel	%Nb	%C	...	%Fe	Shrink
Row_1					
Row_2					
Row_3					
Row_4					
Row_5					
Row_6					
Row_7					
Row_...					
Row_n					

Figure 3.26: Random oversampling technique steps

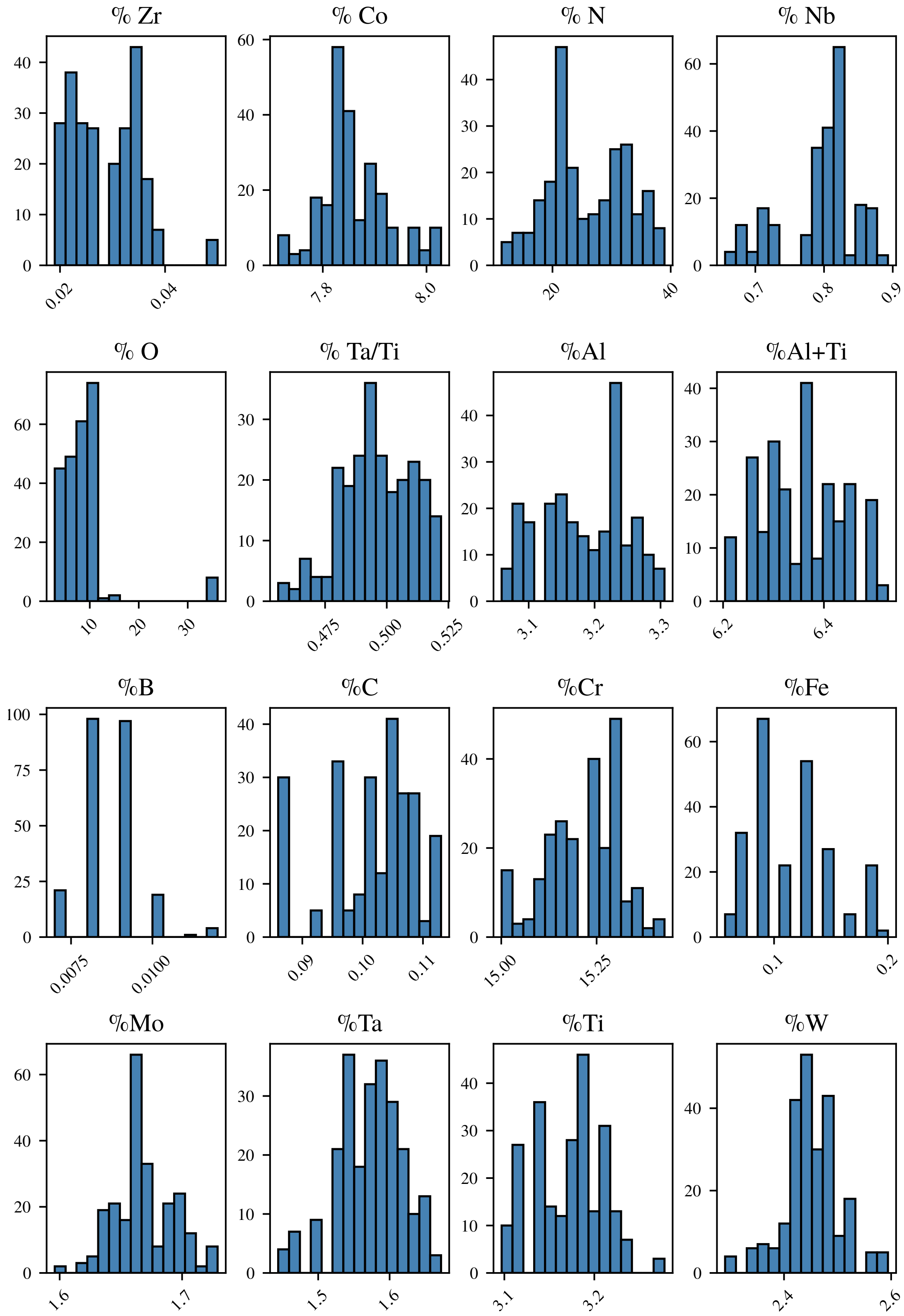


Figure 3.27: Distribution of factors while applying the random oversampling technique

3.13.2 SMOTE

The main drawback of the random oversampling technique is that it increases overfitting. To overcome this problem, the researchers [14] attempt to deal with imbalanced classes in data. The presence of imbalanced data can cause inaccurate predictions regarding the minority class due to insufficient representation of minority observations during training. With the goal of solving this problem, the Synthetic Minority Over-sampling TEchnique (SMOTE) was introduced for classification problems. The core concept behind SMOTE is to synthesise ideas for the minority class, with a focus on those that already exist. It operates by randomly selecting K-values from the minority class and calculating K-Nearest Neighbours for this stage. Synthetic points are then inserted between the selected point and its neighbours.

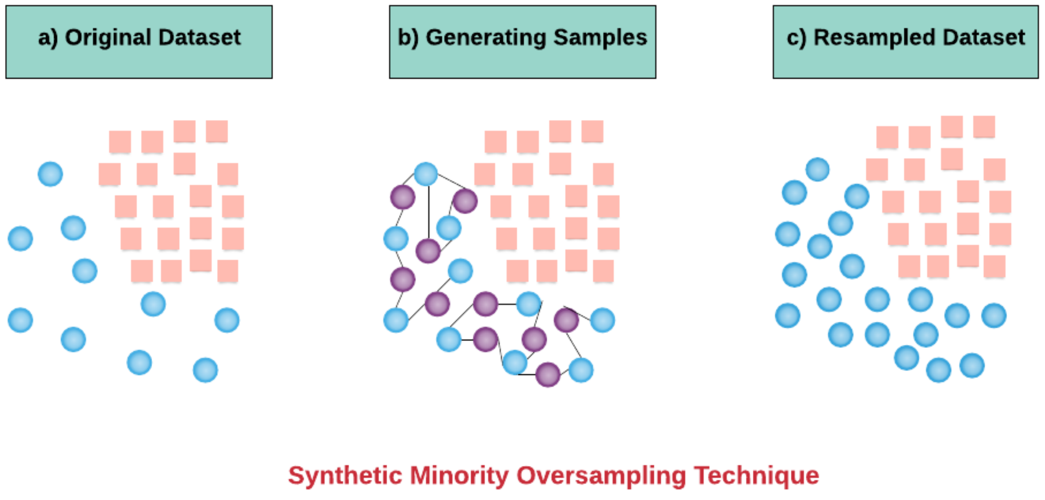


Figure 3.28: Illustration of the SMOTE algorithm

As can be seen from Figure 3.28, the synthetic data was generated based on the clustering of the feature space. This clustering was based on the response label and involved finding the K-nearest neighbours (KNN) and conducting linear interpolations to produce a new minority instance in the neighbourhood. Algorithm 8 presents the detailed implementation steps for SMOTE. The algorithm starts with the following steps.

1. Initially, this algorithm expects three inputs: the existing number in the minority class (S_T), the required value of augmentation for the minority class sample (P_N), and the KNN value.

2. Then, the algorithm begins to loop over every sample (S_{sample}) in the minority class (S_T) and computes the KNN.

3. Ultimately, one of the nearest neighbours (nn) is chosen randomly.

4. Afterwards, the algorithm loops over each attribute (A) to find the new synthetic sample ($S_{synthetic}$) by implementing the linear interpolations.

5. Repeat by choosing KNN randomly until reaching the required augmentation in the class.

6. Repeat the previous steps for every class in the dataset.

Algorithm 8: SMOTE Algorithm [14]

```
Input: Number of minority class samples $S_T$
Amount of required SMOTE $P_N$, Number of nearest neighbours $K$
Output: Required $P_N$ of synthetic minority class samples $S_T$
$P_N$ = (int)($P_N - S_T$)
$A_{number}$ = Number of attributes
$S_{sample}$[ ][ ]: Array for original minority class samples
$i_{new}$: Keeps a count of number of synthetic samples generated, initialised to 0
$S_{synthetic}$[ ][ ]: Array for synthetic samples
// Compute $K$-Nearest Neighbours for each minority class sample
for $i \leftarrow 1$ to $S_T$ do
    Compute $K$-Nearest Neighbours for $i$, and save the indices in the $nn_{array}$
    Populate($P_N$, $i$, $nn_{array}$)
end
// Function to generate the synthetic samples ($S_{synthetic}$).
Function Populate($P_N$, $i$, $nn_{array}$):
    while $P_N \neq 0$ do
        Choose a random number between 1 and $K$, call it $nn$. This step chooses one of
         the $K$-Nearest Neighbours of $i$
        for $A \leftarrow 1$ to $A_{number}$ do
            Compute: Difference = $S_{sample}[nn_{array}[nn]][A] - S_{sample}[i][A]$
            Compute: Gap = random number between 0 and 1
            $S_{synthetic}[i_{new}++][A] = S_{sample}[i][A]+$ (Gap × Difference)
        end
        $i_{new}++$
        $P_N = P_N - 1$
    end
```

3.14 Model Tuning

Tuning the model parameters limits the model's tendency to overfit. The most common tuning parameter in all machine learning models is choosing the most relevant features. In ensemble methods, the second most important parameter in tuning is choosing the optimal number of estimators.

3.14.1 Feature Selection

Feature selection is a process whereby the factors in the data that are most applicable to the predicted response or performance are selected. Getting irrelevant characteristics in the data will reduce the accuracy of several models. Three of the benefits of conducting feature selection before actually modelling are as follows:

- **Reduces Over-fitting:** Less redundant data means a reduced potential to arrive at noise-based decisions.
- **Improves Accuracy:** Less inaccurate data tends to improve the accuracy of modelling.
- **Reduces Training Time:** Fewer data helps the algorithm train faster.

In the scope of optimisation procedures, the feature selection can be used to select the essential variables that significantly influence the performance and process output of the method. A novel feature selection approach based on scoring the critical factor is introduced in chapter five.

3.14.2 Optimal Number of Estimators

The current literature does not provide an adequate maximum limit of estimators in the ensemble models [89]. A random subset of the training sample is used to calibrate a black-box estimator, after which a final prediction is developed by combining the estimations. In this research, the random bootstrapping of the forest results in uncorrelated estimators that negatively affect the variability of the base estimator; that is, a decision tree. This study's novel approach to the calculation of optimum forest size is discussed in chapter four.

3.15 Assessing Model Accuracy

The predictive performance of the machine learning model contributes to its ability to predict unseen test data. The evaluation of this success is essential in an implementation context, as it directs the selection of a machine learning approach and provides a product consistency evaluation method. In this section, the main performance assessment methods are considered.

Measuring the Quality of Fit

Regression is a means of matching a function to a set of data. To assess the efficiency of both the regression model and the data collection, it is necessary to find a way to calculate the extent to which its forecasts have generalised the training data. In other words, it is the error between the measured and predicted values. The most widely used measurement in the regression context is the mean squared error (MSE), given by Equation 3.20 [89],

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{3.20}$$

The coefficient of determination is a standard criterion for determining fitness. The definition of R-squared is a reflection of the regression model that is comparatively more appropriate than only having a horizontal line across the mean value. It can be calculated by Equation 3.21 [89],

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} = 1 - \frac{\sum_i (Y_i - \hat{Y}_i)^2}{\sum_i (Y_i - \bar{Y}_i)^2} \tag{3.21}$$

3.16 Validation Techniques

Validation techniques measure the performance of the models. These are commonly used to show how efficient the prediction is. In this section, two methods are described: the holdout method and the cross-validation technique.

3.16.1 Holdout Method

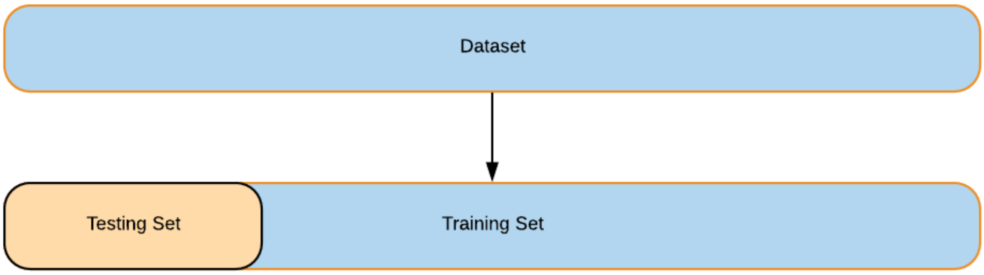


Figure 3.29: Holdout method

This method involves removing some parts of the training datasets and utilising them to gain insights (predictions) from the system trained on the remainder of the data [98]. Machine learning algorithms are trained on the training set, while the test set is used to evaluate the model predictive performance on unseen data. The error calculation then tells how this process will operate for the unseen data.

3.16.2 Cross-Validation Technique

Cross-validation [89] is a generalised re-sampling technique that can be used to test any supervised machine learning algorithm in a small sample of observations. This methodology has a single k parameter, which is an integer value that represents the number of splits of the original dataset. As such, the technique is sometimes referred to as k-fold cross-validation.

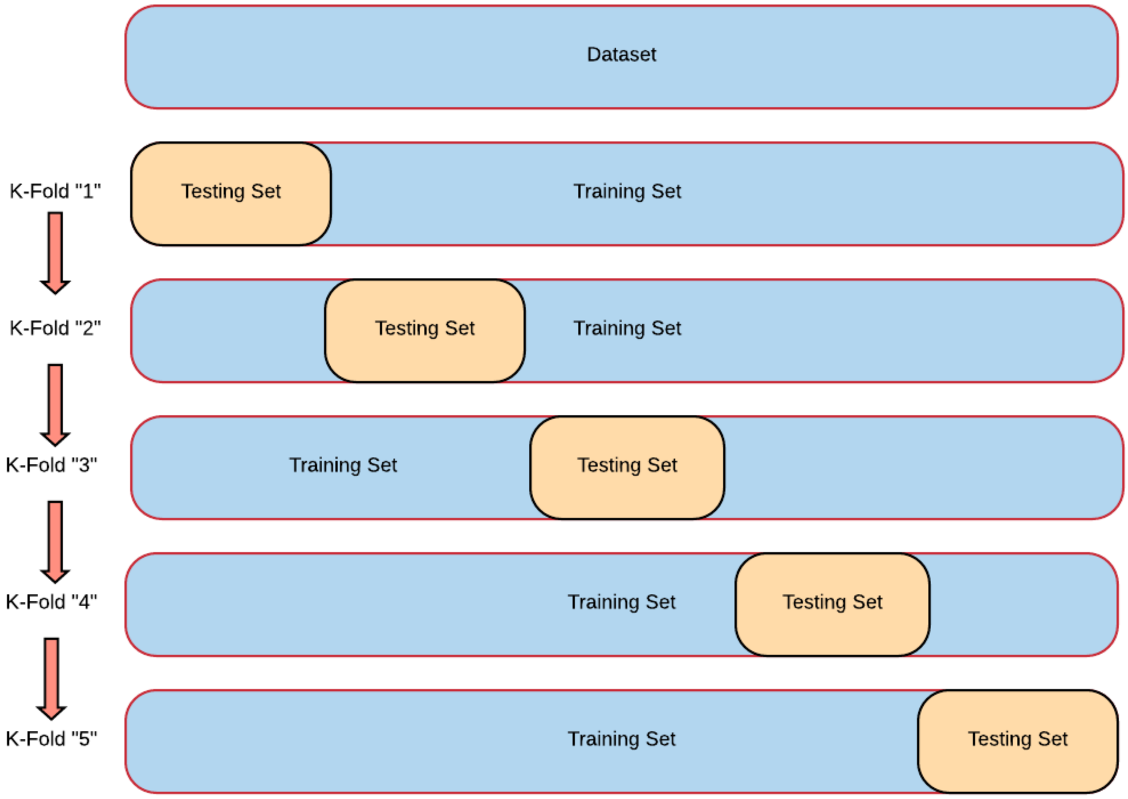


Figure 3.30: Cross-validation technique

Typically, a model output evaluation is a stricter and less biased performance assessment when compared to other approaches such as straightforward holdout methods. Figure 3.30 illustrates the protocol of the cross-validation technique in terms of a specific k-fold value.

The basic protocol can be described as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups.
3. For each particular group:
   (a) Select the group as a holdout or test dataset.
   (b) Take the remaining groups as a training dataset.
   (c) Fit a model to the training data and assessment of the test data.
   (d) Retain the evaluation score and discard the model.
4. Analyse the model performance using the average of the model evaluation scores.

It may be assumed that, in k-fold cross-validation, k holdouts from each of k subsets are utilised as a test set, while the other $(k - 1)$ subsets are combined to form the training set. The estimate of the error is averaged over all k trials to ensure that the model perform adequately. This significantly decreases the bias and variance, as much of the data for fitting as well as the validation data is deployed. In this thesis, the value for k-fold validation (i.e. the number of cross-validation subsets) was chosen as k=10. This value was obtained from the literature [89]. Multiple tests were conducted to determine the best value for k-fold. The tests revealed that k=5 or k=10 are the best choices with the least bias and the best prediction accuracy [89]. Thus, the author used k=10 in this thesis in order to apply the k-fold cross-validation technique.

3.17 Research Roadmap

The research roadmap was established based on several criteria. From the literature review in chapter two, knowledge from prior research and other competing algorithms gradually led to the devising of a solution to solve a non-linear mixed dataset based on traditional Random Forest. The Random Forest algorithm explains some of the problems, along with how to retrieve a solution to modify those gaps. In essence, one of the major goals is to have lower computational costs compared to previous works [8]. For example, the QCA algorithm [8] takes approximately 14-15 minutes for a dataset of only 60 observations (a more detailed analysis is presented and illustrated in Section 6.5). The author chose to focus on shallow learning to further minimise the computational power by using a predictive algorithm capable of doing this. The provided dataset has a known output and the features deal with a mixed dataset. Therefore, supervised learning was deemed the best solution to consider. Before building the model and choosing a specific algorithm tuning strategy to move forward from, it is important to explore the data and the feature engineering in order to improve the prediction accuracy. Skewness and kurtosis were implemented to further observe the input and output processes. It was discovered that the data has a highly skewed shrinkage defect, leading to overfitting. Moreover, one of the drawbacks of traditional Random Forest is that it requires a large dataset to perform efficiently. Hence, it is necessary to use the data augmentation technique to solve this problem. Furthermore, to limit overfitting, five major observations need to be covered in order to build a robust model: small dataset, non-linear, highly skewed, mixed dataset, and handling missing data. In consideration of the non-linear and highly skewed (and small dataset) properties, a novel approach to data augmentation was proposed to further minimise overfitting and improve prediction accuracy. For mixed data, a LabelEncoder was used to encode the categorical factors. Regarding the missing observations, Random Forest has its own built-in imputation methods to overcome this approach. Based on the data exploration, it was concluded that encoding the categorical factors, handling missing data imputations, and data augmentation techniques could be discovered to overcome those gaps. The next steps are building the required model. Before proceeding with Random Forest, the decision tree is the first step to discover. In the decision tree context, there are two things to consider: first, which tree constructing algorithm is required, and second, what splitting rule should be used in order to achieve the best split. The CART algorithm was selected when moving forward, since it deals well with both classifications and regression trees. In regression problems, Variation in Reduction is used for the splitting criteria, while in classification problems, the Gini Index is chosen. After the tree construction algorithm and the splitting rule were defined, discovering the Random Forest was the next step. Random Forest is an ensemble learning method that deals with both classification and regression problems and operates by constructing multiple decision trees. It is considered both a flexible model and an algorithm capable of dealing with complex problems. In the above discussion of the bias-variance trade-off, it was observed that a complex algorithm usually has low bias and high variance. In Random Forest, due to the way the model is built (i.e. generating bags by bootstrapping to create multiple decision trees), it leads to minimising the variance; thus, the model inherently has low bias and low variance. Random Forest also requires precise model tuning to achieve a good prediction result. Regarding the tuning parameters, it is necessary to choose precise parameters to achieve a low computational cost (which was the major concern, since it is one of the drawbacks in the previous research and finding a way to resolve it is necessary), and yield a high-performance model. Thus, another novel approach is established to resolve those issues by obtaining an automatic number of estimators.

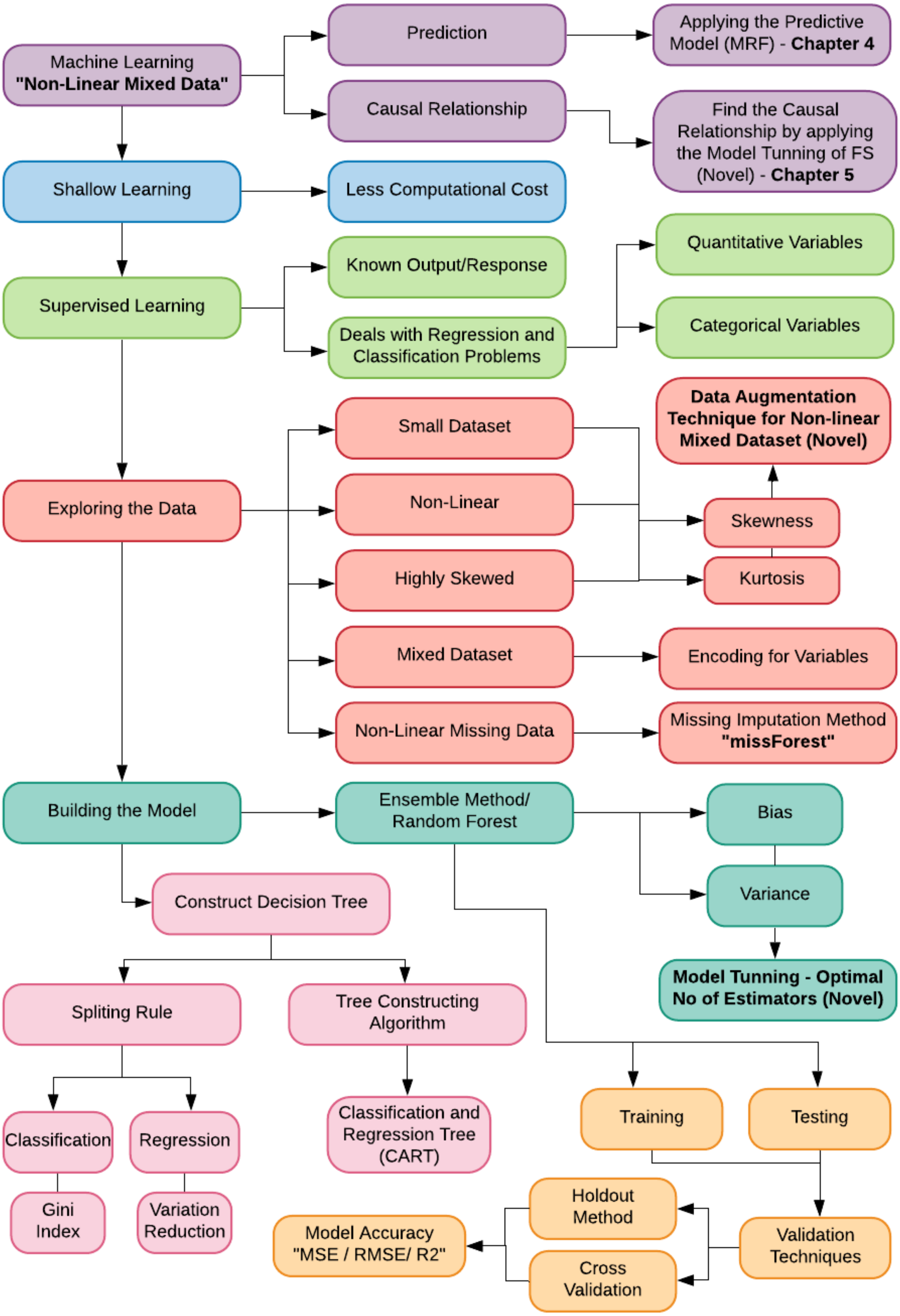


Figure 3.31: Summary of algorithm proposed to overcome the limits

The algorithm can then be used for training and testing on the dataset. After implementing the algorithm, a validation technique is required in order to check the accuracy of the predictive algorithm. Two techniques were used: namely, the holdout method and the cross-validation technique. In order to compute the model accuracy, the mean square error (MSE) and coefficient of determination (R^2) were used. At the end, all previous techniques will be used in order to bring them into one package and enable proceeding to the next step. The proposed algorithm is now ready to be used in order to achieve robust performance and prediction capacity. The steps and techniques were combined to further proceed to the proposed algorithm, as shown in Figure 3.31. Taking into consideration the fact that everything was scattered, the initial task was to apply all existing methods using the traditional Random Forest. The second step was then to put the pieces together so that the algorithm can be used to either solve the problem discussed in chapter one or not. Therefore, there was a need to combine all steps into a single package. Nevertheless, several remaining limitations were identified. As a result, a decision was made to select the most suitable techniques to address the research objectives. This led to the development of the Modified Random Forest algorithm. The MRF algorithm is explained in more detail later on in the thesis (chapters four and five). Accordingly, based on this roadmap (see Figure 3.31), it is likely that all weak spots will be identified and be fixed by applying the novel techniques.

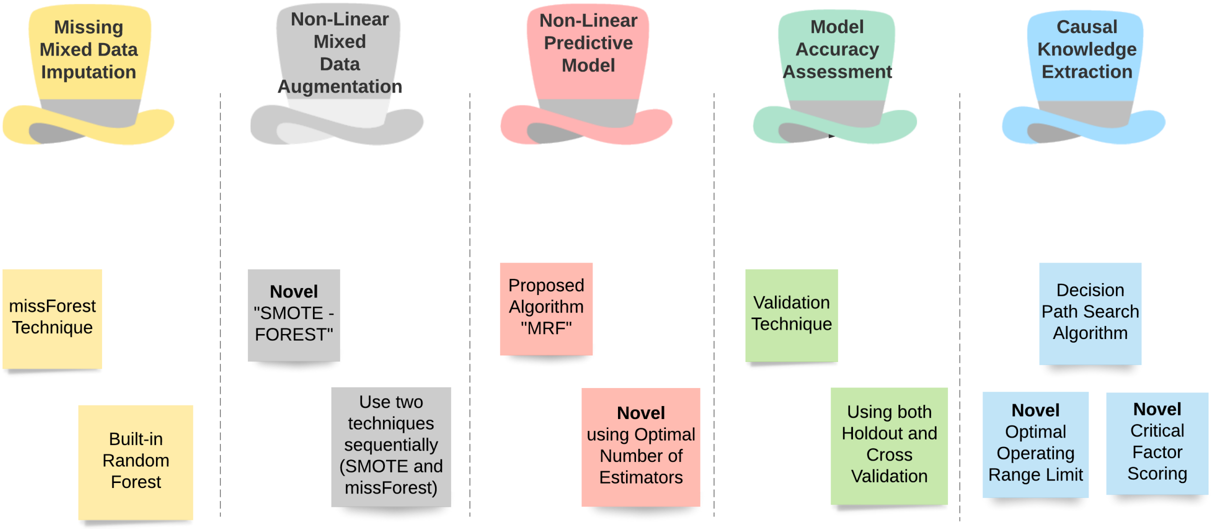

Missing Mixed Data Imputation
Non-Linear Mixed Data Augmentation
Non-Linear Predictive Model
Model Accuracy Assessment
Causal Knowledge Extraction
missForest Technique
Built-in Random Forest
Novel "SMOTE - FOREST"
Use two techniques sequentially (SMOTE and missForest)
Proposed Algorithm "MRF"
Novel using Optimal Number of Estimators
Validation Technique
Using both Holdout and Cross Validation
Decision Path Search Algorithm
Novel Optimal Operating Range Limit
Novel Critical Factor Scoring

Figure 3.32: Research Roadmap

The research roadmap began with the missing mixed data imputation by using the built-in Random Forest missing technique, called ‘missForest’. Subsequently, it goes through the non-linear mixed data augmentation technique using the up-sampling technique. This is achieved by applying two novel techniques sequentially; namely: SMOTE and missForest (both discussed in more detail in chapter four). The research introduces the non-linear predictive model using the MRF algorithm with an automated number of tree estimators (discussed in chapter four), followed by the model accuracy assessments by performing the validation technique (holdout and cross-validation techniques). Finally, the causal knowledge extraction is implemented by using a novel development proposed in this research (introduced in chapter five), which is called the 'Decision Path Search (DPS) algorithm'. This study's research roadmap, presented in Figure 3.32, indicates the steps involved in the proposed solutions; together, they form a model and an approach. The features and the learning algorithm are the two pillars of any data models, and these will be studied further in chapter four, Moreover, the details about how the causal knowledge is extracted from the model will be introduced and discussed further in chapter five.

3.18 Technological Choices

Python is a programming language. In addition to the scripting language and the compiler, Python consists of an extensive standard library. This library is regarded as the primary scripting and incorporates components for various operations, threading, networking, databases, etc. It is a general-purpose programming language that is easy to learn. The language is open-source and incorporates tools for mathematics and data analysis (for both statistics and analytics). As a result, Python has become the most widely used programming language in research and scientific investigations. Python can be used to execute multiple series of instructions at once or one block at a time. **MATLAB**, on the other hand, is a commercial computational programming and coding tool system that is mostly used by engineers for numerical computations. While the class library does not contain quite enough generic programming features, it does include an arithmetic matrix-based calculation and an extensive library for processing and plotting data. In this thesis, the author selected Python rather than MATLAB or R-Code to run the MRF algorithm. Although MATLAB and R-Code are both good software packages, Python represents a new generation of machine learning paths. R-Code was developed by statisticians to resolve statistical problems, while Python is a general-purpose programming language. Although most machine learning code is based on the R-Code language, Python was eventually selected because it offers a number of advantages. First, in comparison to MATLAB and R-Code, Python offers more libraries for a variety of applications (including libraries for maths, statistics and as artificial intelligence), as well as a faster run-time. Moreover, Python is regarded as the most improved tool for Machine Learning Integration and Deployment. In addition, Python can perform almost the same functions as R-Code but is also a fully integrated programming language (unlike the R-Code). In the previous thesis [8], MATLAB was used to program the code; in this research, however, Python was selected as the programming language for the implementation of the algorithms, as it offers the following advantages described in Table 3.4 below.

Table 3.4: Advantage of Python over MATLAB and R-Code

Python	MATLAB	R-Code
Large open-source ecosystem.	With default IDE and limited ecosystem.	Spread across several packages.
Supported cross-platform.	Expensive and commercial license.	Lacks basic security.
Name-spaces can be imported.	The default core does not have name-space.	Utilises more memory in comparison to Python.
Supports object-oriented programming.	Limited portability.	Not a fully integrated programming language.
Offers tremendous support of libraries.	Limited libraries, defined separately.	A smaller subset of statistical data.
Introspection is easy; easier access to internal sections also provided.	Closed-source libraries make it difficult to inspect the internal components.	Complicated language.

Python Processing Code of MRF and DPS Algorithms

The processing code used to implement both algorithms, namely MRF and causal relationships (DPS), was developed by the author using Python. The results illustrated in chapters four and five are derived from this processing code. Moreover, the simulation parameters used for the MRF and DPS algorithms are listed in Table 3.5. The code is applied to both algorithms used in this thesis. The code of the proposed algorithm is based on pre-existing libraries in Python; however, the code was also modified in order to overcome the limitations that were identified in this thesis. All the code refinements developed in this thesis were implemented from scratch. From Table 3.5, the minimum (Th_{min}) and maximum (Th_{max}) thresholds for the process were determined by the process engineers and then measured at the end of the investment casting process. In addition, the optimal threshold value (Th_{op}) was used to check the performance of the optimal limit process, as well as how these ranges affect the overall prediction accuracy. The optimal threshold was also measured by the process engineer, and hence provides the process input and output. The code was modified so that it would work on both the QCA [8] and the MRF algorithms. The user can choose whether to apply the MRF algorithm or the QCA algorithm for further visualisation and analysis. In order to verify the performance of the QCA algorithm in Python language, the author reproduced all the results of the current work by applying the QCA algorithm (see Appendix C). The MRF and the DPS processing code flowchart are illustrated in Figure 3.33.

Table 3.5: User inputs for prediction and confirmation for nickel-based superalloy dataset

MRF		Confirmation Trials	
Type	LB	Bootstrap	1000
Th_{min}	0	Simulation No	100
Th_{max}	0.03	Total Run's	100000
No of Factors	16	Th_{op}	0.2, 0.3, 0.4

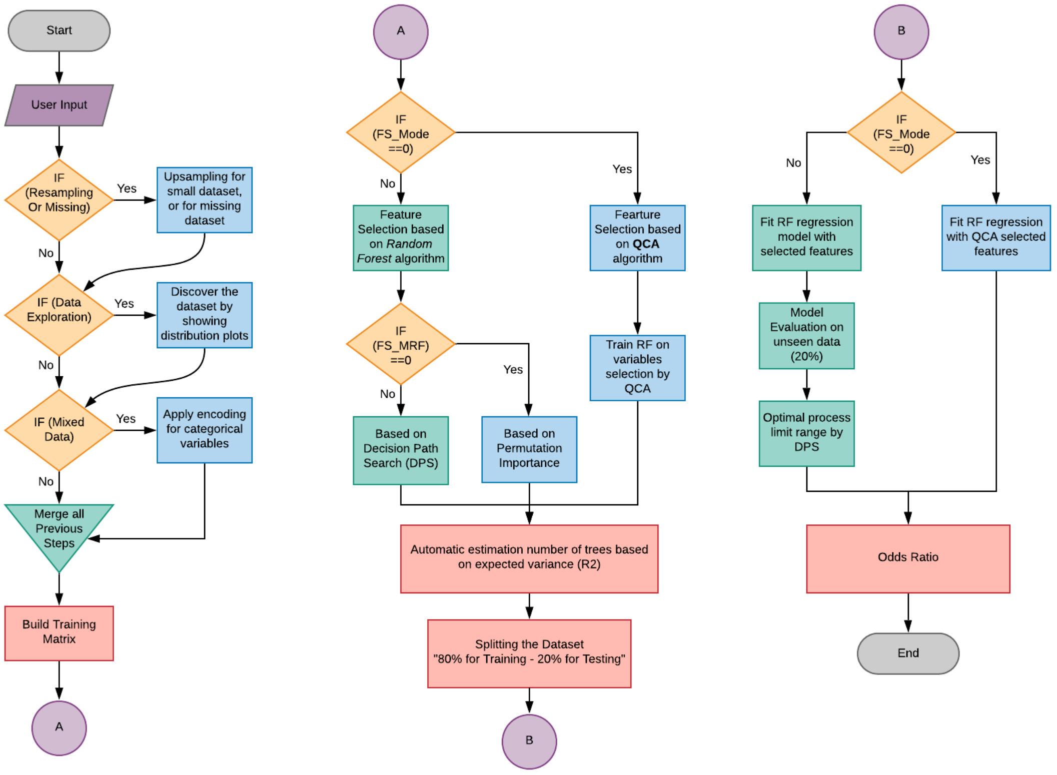

Start
User Input
IF (Resampling Or Missing)
Yes
Upsampling for small dataset, or for missing dataset
No
IF (Data Exploration)
Yes
Discover the dataset by showing distribution plots
No
IF (Mixed Data)
Yes
Apply encoding for categorical variables
No
Merge all Previous Steps
Build Training Matrix
A
A
IF (FS_Mode ==0)
No
Yes
Feature Selection based on *Random Forest* algorithm
Fearture Selection based on **QCA** algorithm
IF (FS_MRF) ==0
No
Yes
Based on Decision Path Search (DPS)
Based on Permutation Importance
Train RF on variables selection by QCA
Automatic estimation number of trees based on expected variance (R2)
Splitting the Dataset "80% for Training - 20% for Testing"
B
B
IF (FS_Mode ==0)
No
Yes
Fit RF regression model with selected features
Fit RF regression with QCA selected features
Model Evaluation on unseen data (20%)
Optimal process limit range by DPS
Odds Ratio
End

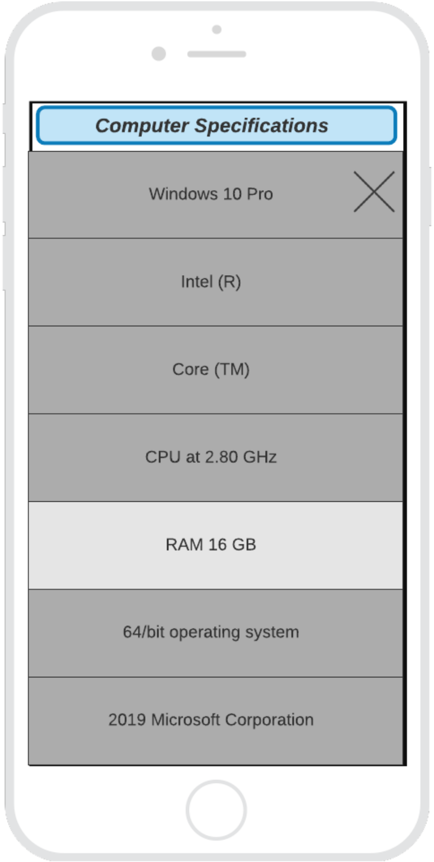
Figure 3.33: The proposed algorithms (MRF and DPS) processing flowchart code

Computer Specifications

The computational cost indicated for each case studies was found by running the code on a personal computer (laptop) with the specifications given in Table 3.6 and Figure 3.34.

Table 3.6: The computer specifications of the proposed code

Parameter	Specifications
Operating System	Windows 10 Pro
Edition	2019 Microsoft Corporation
Processor	Intel(R) Core(TM) CPU at 2.80 GHz
Installed Memory (RAM)	16 GB
System Type	64/bit operating system, x64



Computer Specifications

Windows 10 Pro

Intel (R)

Core (TM)

CPU at 2.80 GHz

RAM 16 GB

64/bit operating system

2019 Microsoft Corporation

Figure 3.34: Computer running code specifications

Chapter 4

Modified Random Forest Algorithm

4.1 Introduction

The primary objective of a machine learning study is to develop a generally applicable algorithm for use in predicting future unseen data. From a machine learning perspective, the solution to the generalisation problem is two-fold: specifically, the generalisation ability of the algorithm on both the training and testing dataset. The methods optimise the objective function and learn the abstractions that collectively and compactly form the model. The study was carried out on the nickel-based superalloy dataset, consisting of 16 chemical profile variables and the process output represented by shrinkage defect percentages.

In this chapter, the Modified Random Forest algorithm is developed. The goal here is to adapt the traditional RF algorithm so that it can deal with a small dataset through a novel technique of non-linear data augmentation, as well as further minimise the variance error and computational cost through the use of a novel technique to estimate the optimal forest size. The remainder of this chapter is structured as follows. Section 4.2 defines and explains the Modified Random Forest algorithm (MRF) and lists the steps for its implementation. This section consists of three parts: the novelty of the non-linear augmentation technique, the novelty of the optimal forest sizing, and a summary of the proposed algorithm steps. The model accuracy assessments are introduced in Section 4.3, which shows the results obtained by using both the holdout method and cross-validation techniques. In Section 4.4, missing data imputation based on the SMOTE-FOREST technique is presented, while it is further shown how effective this technique is compared to what has been done previously. Finally, the chapter's discussion and conclusion are summarised in Section 4.5 and 4.6, respectively.

4.2 Modified Random Forest Algorithm (MRF)

The Modified Random Forest algorithm is built over a traditional Random Forest algorithm; however, it also incorporates extra features to adapt the original algorithm so that it can deal with any non-linear process, such as the foundry process historical dataset test case (introduced and discussed in Section 3.3), which is high-dimensional and unbalanced and tends to have missing values. These modifications alleviate some of the limitations inherited from the traditional Random Forest algorithm. The MRF algorithm also aims to resolve the problems previously encountered with non-linear datasets – namely, predictive power, non-linearity, computational efficiency, and interpretability – by introducing the following two novel techniques.

1. The first technique is a novel iterative Random Forest training approach used for augmentation, balancing, and imputing missing values for non-linear mixed datasets.

2. The second technique is a novel automatic forest size optimisation. This technique maximises the inherited gains obtained from ensemble uncorrelated multiple decision trees.

The Modified Random Forest (MRF) algorithm is illustrated in Figure 4.1 and can be summarised as follows:

- Missing data imputation for non-linear mixed datasets (as discussed in chapter three);
- Novel data augmentation technique for non-linear mixed datasets;
- Novel automated optimal forest size estimation.

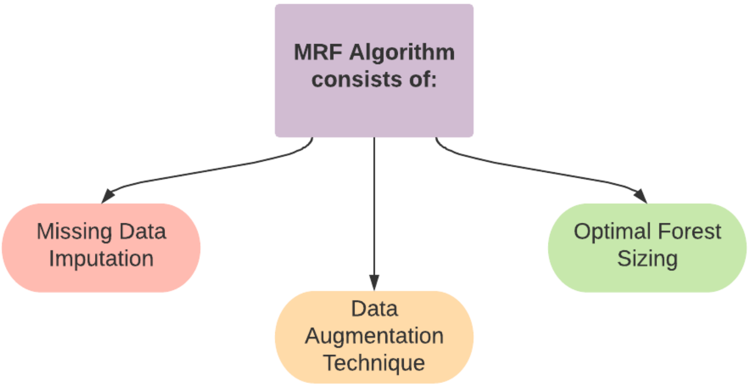


Figure 4.1: Summary of the MRF algorithm

The development process, along with the detailed steps involved in the novel techniques of the Modified Random Forest algorithm (as shown in Figure 4.2), will be discussed in the next sections.

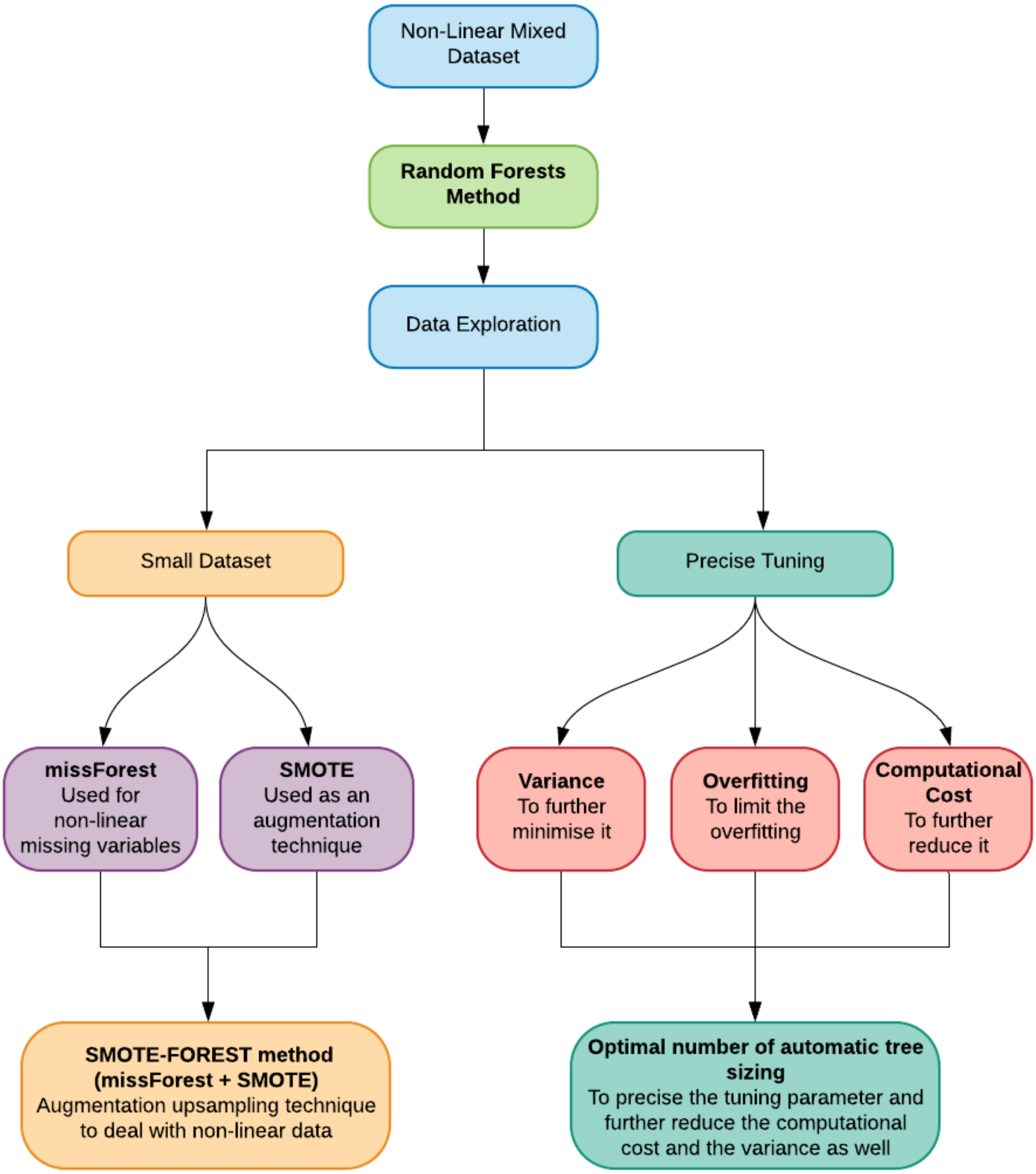

Non-Linear Mixed Dataset
Random Forests Method
Data Exploration
Small Dataset
Precise Tuning
missForest Used for non-linear missing variables
SMOTE Used as an augmentation technique
Variance To further minimise it
Overfitting To limit the overfitting
Computational Cost To further reduce it
SMOTE-FOREST method (missForest + SMOTE) Augmentation upsampling technique to deal with non-linear data
Optimal number of automatic tree sizing To precise the tuning parameter and further reduce the computational cost and the variance as well

Figure 4.2: Relational overview of the techniques used to devise the novel MRF

4.2.1 Novelty of Non-Linear Augmentation Technique

As discussed in chapter three, the major drawbacks of SMOTE relate to linear interpolation and its restriction to classification problems only (i.e. it is not valid for regression problems). Motivated by combining missForest and SMOTE sequentially, a novel synthetic oversampling technique is proposed that resolves the significant drawbacks of the SMOTE algorithm. The main idea here is to apply the SMOTE and missForest algorithms sequentially to create SMOTE-FOREST technique, as described in Algorithm 9. In the first step, the problem is temporarily converted into a classification problem by classifying the process response into four quarters based on the output penalty, as follows: quartile-I (0 - 25%), quartile-II (25% - 50%), quartile-III (50% - 75%), and quartile-IV (75% - 100%). Accordingly, the SMOTE can be used for oversampling four classes using Algorithm 8; after that, the response values of the new data example are marked as missing. For its part, the iterative missForest algorithm is used to predict the missing responses for the new observations using Algorithm 6. The flowchart in Figure 4.3 summarises all previous steps. SMOTE-FOREST inherits the strengths of both methods and eliminates the drawbacks; accordingly, the regression can over-sample data even in the case of complex interactions and non-linear relationships. The data augmentation technique is applied to preserve the overall structure for each factor in the process. In addition, this technique is also used to maintain the non-linearity in the shrinkage penalty. In Figure 4.4, each factor was augmented to 360 synthetic data points while maintaining and preserving the overall structure of each factor. The raw data was limited to 60 observations, which led to inaccurate predictions by the original RF algorithm. Figure 4.5 further illustrates each factor after the data was augmented. The RF algorithm requires a minimum number of data points (based on the number of features) to enable efficient algorithmic performance. Data augmentation is a procedure used to maintain the efficiency of RF; the algorithm will not work properly without it. Figures 4.4 and 4.5 depict the SMOTE-FOREST technique resulting from the oversampling process.

```
Algorithm 9: Novelty of SMOTE-FOREST Technique
  Input: Imbalanced training $X$, Target response $Y$
  Output: Over-sampled $X_{new}$,$Y_{new}$
  Calculate quartiles for response column
  Categorise original samples based on response quartiles
  $X_{new}, Y_{new}$ =SMOTE($X, Y$);            // Generate new samples using SMOTE
  Mark response values for the new generated samples as missing
  Correct response value $Y_{new}$ with missForest iterations for all data
  return $X_{new}, Y_{new}$
```

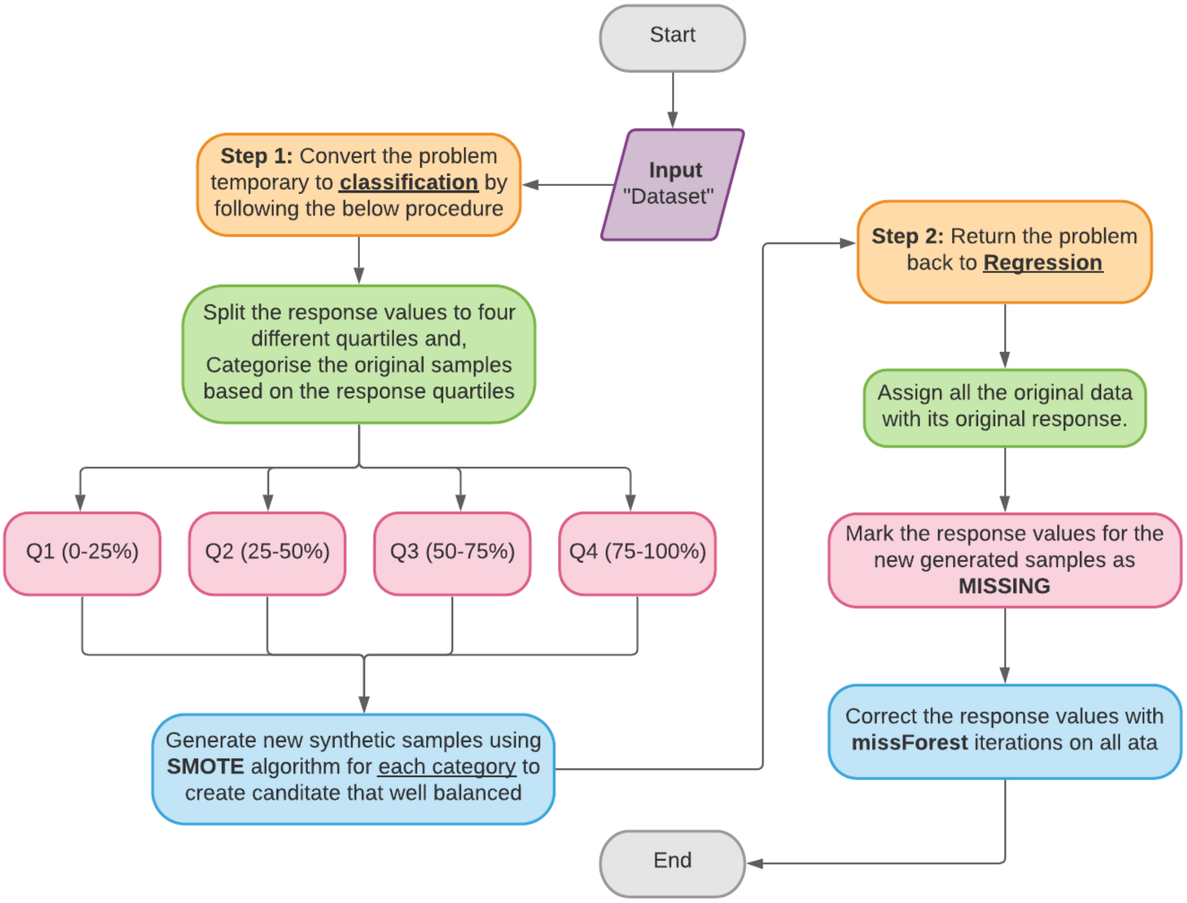
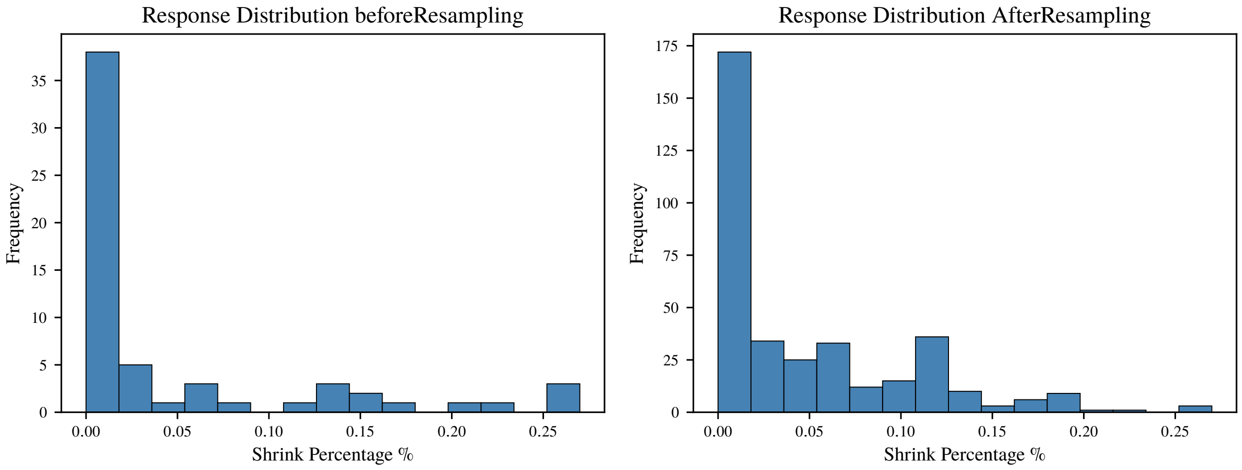

Start
Input "Dataset"
Step 1: Convert the problem temporary to classification by following the below procedure
Split the response values to four different quartiles and, Categorise the original samples based on the response quartiles
Q1 (0-25%)
Q2 (25-50%)
Q3 (50-75%)
Q4 (75-100%)
Generate new synthetic samples using SMOTE algorithm for each category to create canditate that well balanced
Step 2: Return the problem back to Regression
Assign all the original data with its original response.
Mark the response values for the new generated samples as MISSING
Correct the response values with missForest iterations on all ata
End

Figure 4.3: Proposed novel regression oversampling technique (SMOTE-FOREST)



(a) Raw Data

(b) After oversampling

Figure 4.4: Response distribution using the SMOTE-FOREST technique

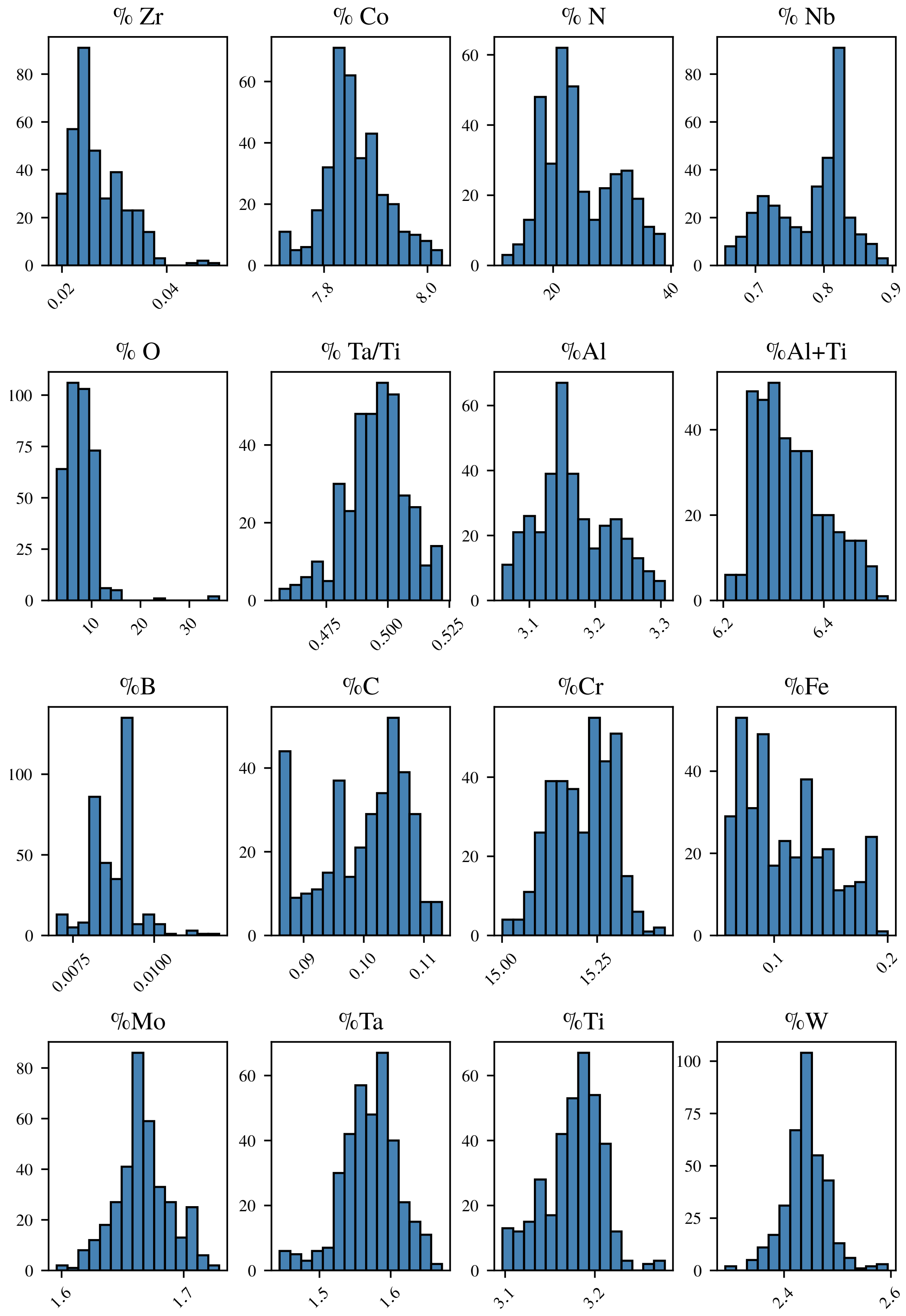


Figure 4.5: Factors distribution by SMOTE-FOREST technique for the nickel-based superalloy dataset

Figure 4.6 depicts the difference between QCA [8] and MRF algorithms in terms of the augmentation technique used. In the previous work by Batbooti [8], the QCA prediction was found to be ineffective for the non-linear interaction factors; this was due to the weakness of linear prediction when used on non-linear factors. Based on the previous results [8], the correlation between carbon variation and shrinkage penalty was downgraded from strong to weak. The augmenting technique employed is limited when dealing with non-linear datasets, which led to a reduction in prediction accuracy. Figure 4.7 illustrates the difference between the variations in carbon and cobalt before and after applying the augmentation technique. The technique was tested on both carbon and cobalt since these two factors were discussed both earlier in the thesis and in the previous work (QCA algorithm) [8]. In this thesis, an attempt is made to show how the proposed technique overcomes the limitation identified in the previous work of Batbooti [8] by undertaking an analysis on both carbon and cobalt. It can be observed that the overall structure was maintained as much as possible in order to preserve any existing correlation and pattern between the input and the output. Furthermore, the augmentation technique preserved the overall structure of both the linear (cobalt) and non-linear (carbon) factors, which demonstrates the robustness of the novel technique. At this point, the technique provides an indication that the process will achieve generally good results; however, further investigations to ascertain the efficiency of this augmentation method are necessary.

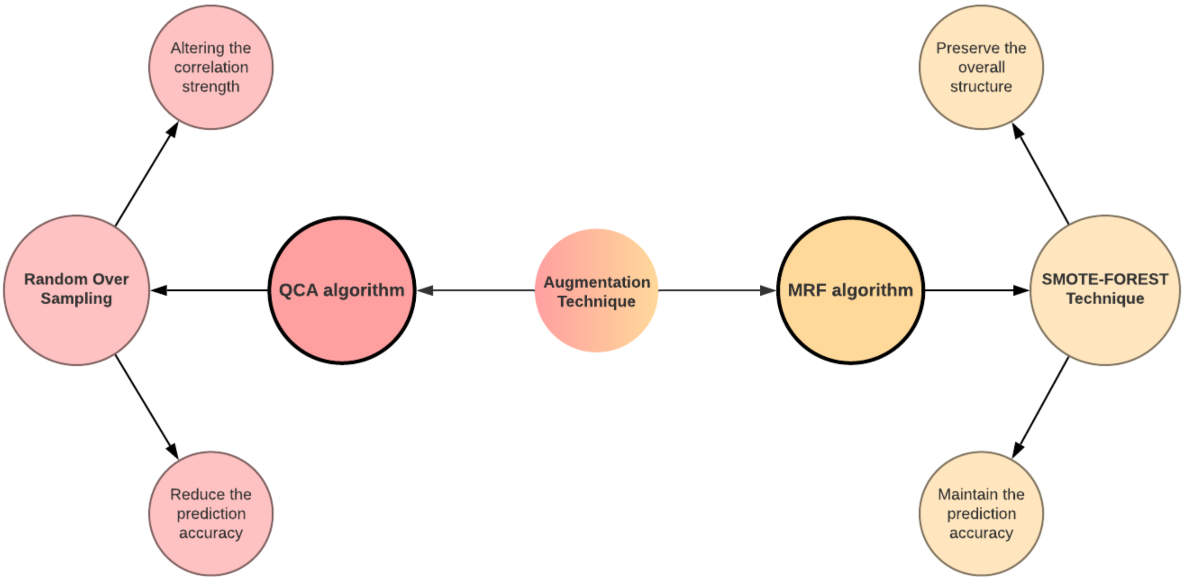
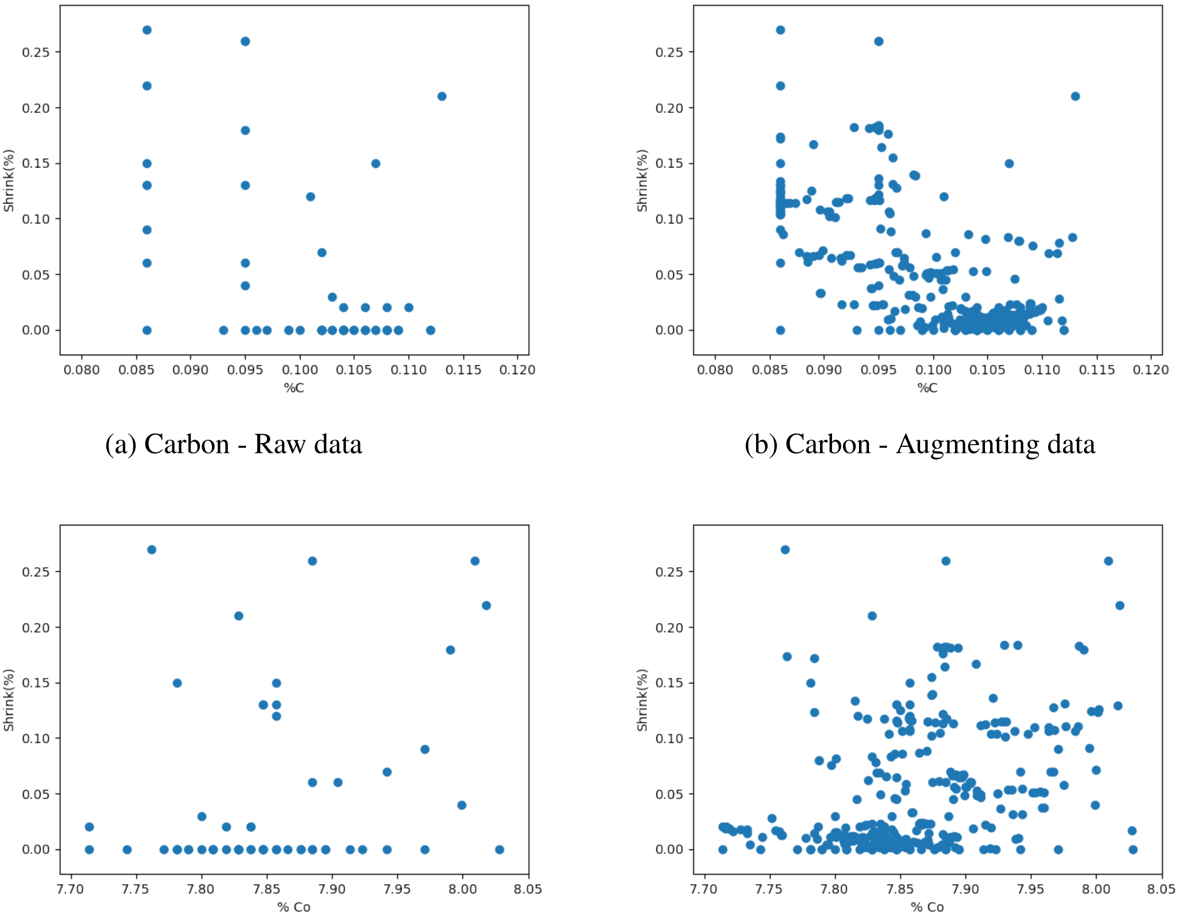

Altering the correlation strength
Preserve the overall structure
Random Over Sampling
QCA algorithm
Augmentation Technique
MRF algorithm
SMOTE-FOREST Technique
Reduce the prediction accuracy
Maintain the prediction accuracy

Figure 4.6: The difference between QCA and MRF algorithm in relation to the augmentation technique



(a) Carbon - Raw data

(b) Carbon - Augmenting data

(c) Cobalt - Raw data

(d) Cobalt - Augmenting data

Figure 4.7: Carbon and cobalt variation before and after applying the novel augmentation technique

4.2.2 Novelty of Optimal Forest Sizing

The existing literature provides almost no direction regarding the optimal number of trees that should be used to generate a forest. In the ensemble algorithms, several example black-box estimators are trained on random subsets of the original training set, and their predictions are aggregated to form a final prediction. The randomised bootstrapped subsets approach used in the RF algorithm in this study produces more trees that are likely to be uncorrelated, which reduces the variance of a base estimator (i.e., a DT). Thus, theoretically speaking, increasing the number of trees will result in more accurate predictions. At a certain point, however, the cost of increasing the forest size will be higher than the benefit in accuracy obtained from such a gigantic forest. After conducting several experiments on different datasets, the researcher concludes that there is an optimal forest size for each dataset, and that exceeding this size will result in an insignificant performance increase accompanied by increases in the computational cost (see Appendix B). The research reported here introduces an approach to the automatic estimation of the optimal number of trees used to generate the forest by simply testing a wide range of forest sizes (from 10 to 420 trees). Each time, a Random Forest algorithm is fitted with a certain tree size ($ntree$), after which the model is evaluated using 10-fold cross-validation. The MSE and standard deviation (σ) of the error in every fold are stored. Finally, the best forest size (O_p) is selected based on minimum MSE and two times the standard deviation ($2 \times \sigma$), as described in Equation 4.1. The steps involved are described in more detail in Algorithm 10 and in Figure 4.9. Moreover, after running the algorithm using the nickel-based superalloy dataset, the results are summarised in Table 4.1 and Figure 4.8. It can be concluded that a forest size of 90 trees is sufficient for obtaining an excellent representative prediction model.

Table 4.1: R^2 scores of the optimal number of estimators for the nickel-based superalloy dataset

Trees	R^2 Score	$2 \times \sigma$	Trees	R^2 Score	$2 \times \sigma$
10	84.4	(+/- 0.1874)	100	87.1	(+/- 0.1831)
20	86.0	(+/- 0.1920)	140	87.0	(+/- 0.1888)
30	86.2	(+/- 0.2069)	180	87.0	(+/- 0.1828)
40	86.5	(+/- 0.1975)	220	86.7	(+/- 0.1821)
50	87.2	(+/- 0.1859)	260	86.9	(+/- 0.1794)
60	86.9	(+/- 0.1850)	300	87.1	(+/- 0.1806)
70	87.2	(+/- 0.1846)	340	86.9	(+/- 0.1797)
80	87.2	(+/- 0.1849)	380	86.9	(+/- 0.1802)
90	**87.5**	**(+/- 0.1825)**	420	86.9	(+/- 0.1801)

Algorithm 10: Novelty of Optimal Forest Size

```
Input: Training matrix $X$, Maximum number of trees $T_{max}$
Output: Optimal number of trees $O_p$
$dT = 10$                                                // Increment size
$i \leftarrow 1, MSE = [], \sigma = []$
while $ntree \leq T_{max}$ do
    $ntree \leftarrow i \times dT$
    $Y \leftarrow f(X)$                                  // Fit data on a forest with size $ntree$
    $MSE, \sigma \leftarrow CV($k-folds $\leftarrow 10, 'MSE')$   // Cross-validation with 10-folds
    $i++$
end
$O_p = min(MSE, 2 \times \sigma)$                       // See Equation 4.1
return $O_p$
```

$$O_p = \underset{(MSE, 2\times\sigma)}{\arg\min} \{ntree\}_{i=10}^{T_{max}} \tag{4.1}$$

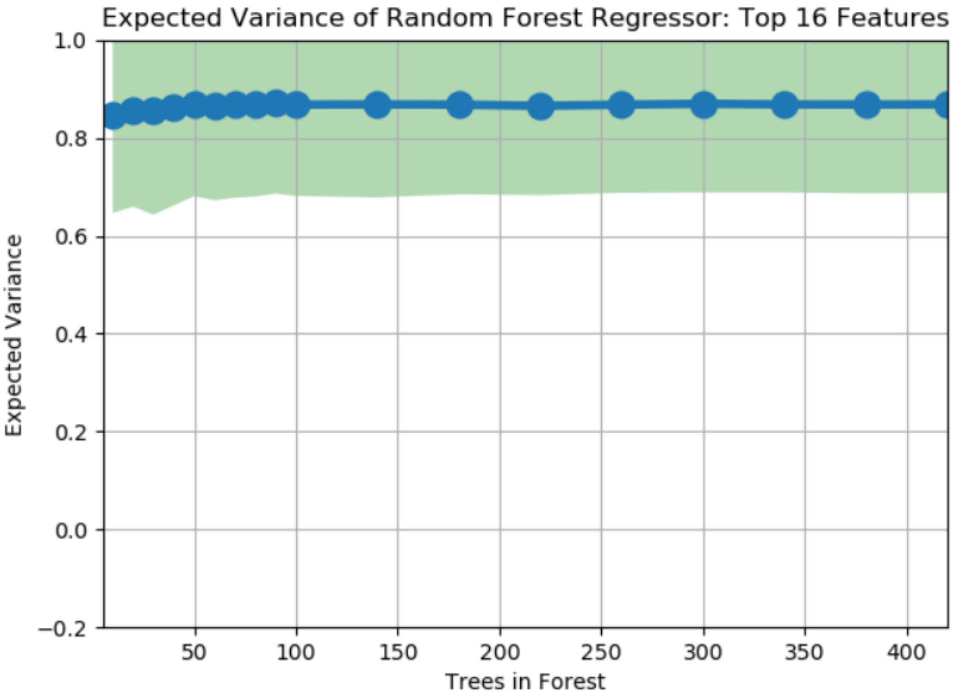


Figure 4.8: R^2 score of different numbers of estimators; here, the blue line represents the R^2 score, while the green area represents $2 \times \sigma$

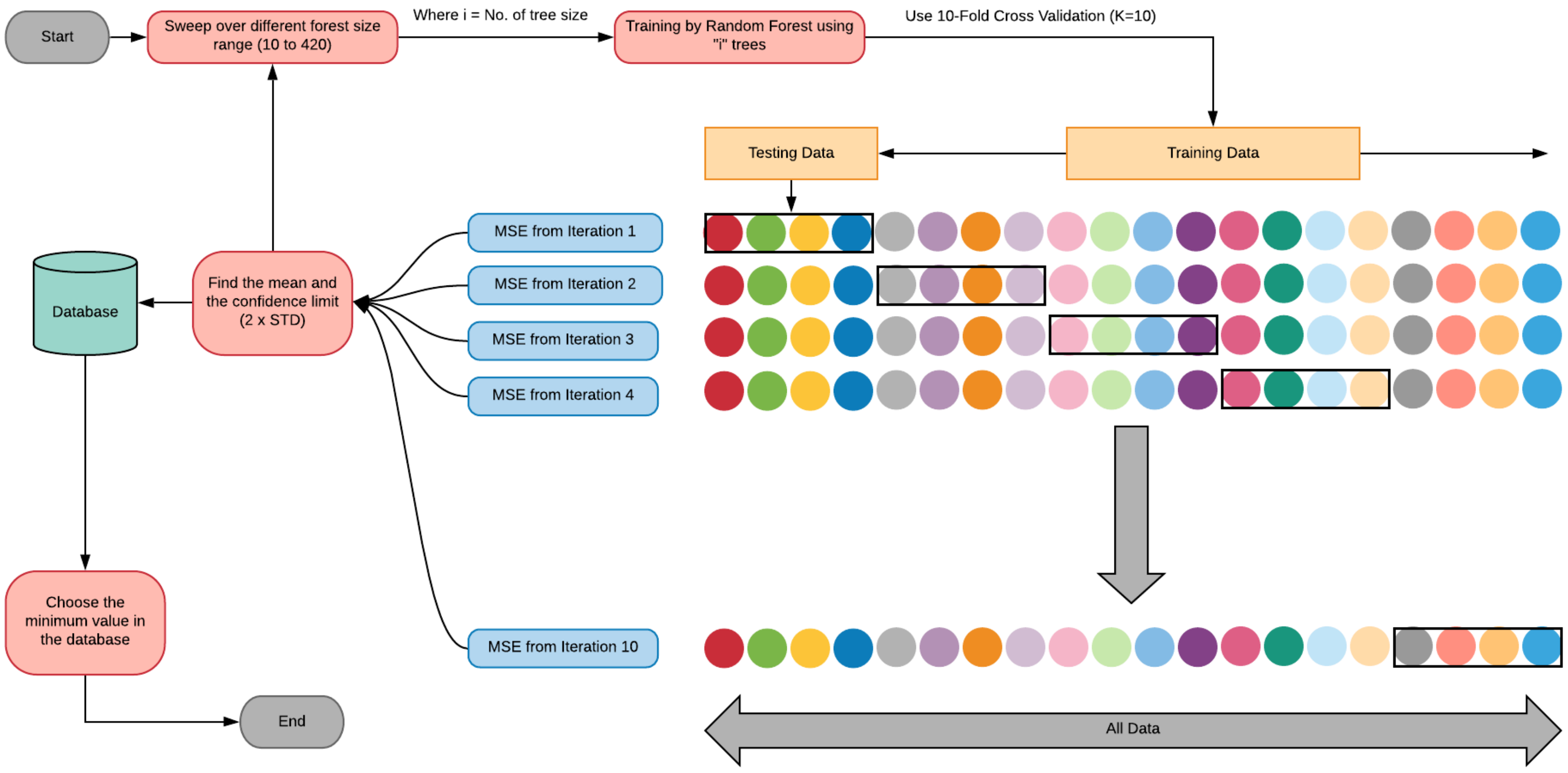

Start
Sweep over different forest size range (10 to 420)
Where i = No. of tree size
Training by Random Forest using "i" trees
Use 10-Fold Cross Validation (K=10)
Testing Data
Training Data
MSE from Iteration 1
MSE from Iteration 2
MSE from Iteration 3
MSE from Iteration 4
MSE from Iteration 10
Find the mean and the confidence limit (2 x STD)
Database
Choose the minimum value in the database
End
All Data

Figure 4.9: Schematic diagram for estimating the optimal number of forest size steps for the nickel-based superalloy dataset

4.2.3 Summary of MRF Algorithm (Prediction Analysis)

The steps involved in the Modified Random Forest are as follows. The algorithm starts by checking on three different criteria: mixed dataset, missing data, and required data augmentation. First, if the data includes mixed features (continuous and categorical), then categorical factor encoding is required; second, if there are missing observations in the data, the missForest technique is used to predict the missing values. Finally, if the dataset is small, the data augmentation technique is applied through the combined SMOTE-FOREST method. After applying all previous pre-processing steps, the proposed algorithm can finally be used to predict the data, since the algorithm faces no issues anymore. Initially, apply the sensitivity study of automatic optimal tree sizing by using the 10-fold cross-validation technique in order to reduce the computational cost and conduct robust tuning to further minimise both overfitting and variance. Then, the model is trained by fitting the Random Forest algorithm using the technique of optimal forest sizing. In the testing phase, validation techniques such as 10-fold cross-validation and the holdout method are applied; here, 20% of unseen data is proposed. Finally, the proposed algorithm is finalised. The next steps in the process, which involve extracting the causal relationships, are discussed in chapter five. The pre-processing steps, model tuning, and training are summarised in the algorithm presented below (Algorithm 11).

Algorithm 11: The Proposed MRF Algorithm Steps

```
Load dataset $L$
if $L$ is mixed data type then
    Categorical factors encoding (Refers to Section 3.11)
if $L$ have missing values then
    Call missForest (Algorithm 6)                          // Check for missing values
if data augmentation required then
    Call SMOTE-FOREST (Algorithm 9)
Optimal forest size calculation using (Algorithm 10)
Fit Random Forest with optimal forest size using (Algorithms 4 and 5)
Cross-validation using 10-folds (Refers to Section 4.3.1)
Final assessment on unseen data (Refers to Section 4.3.2)
```

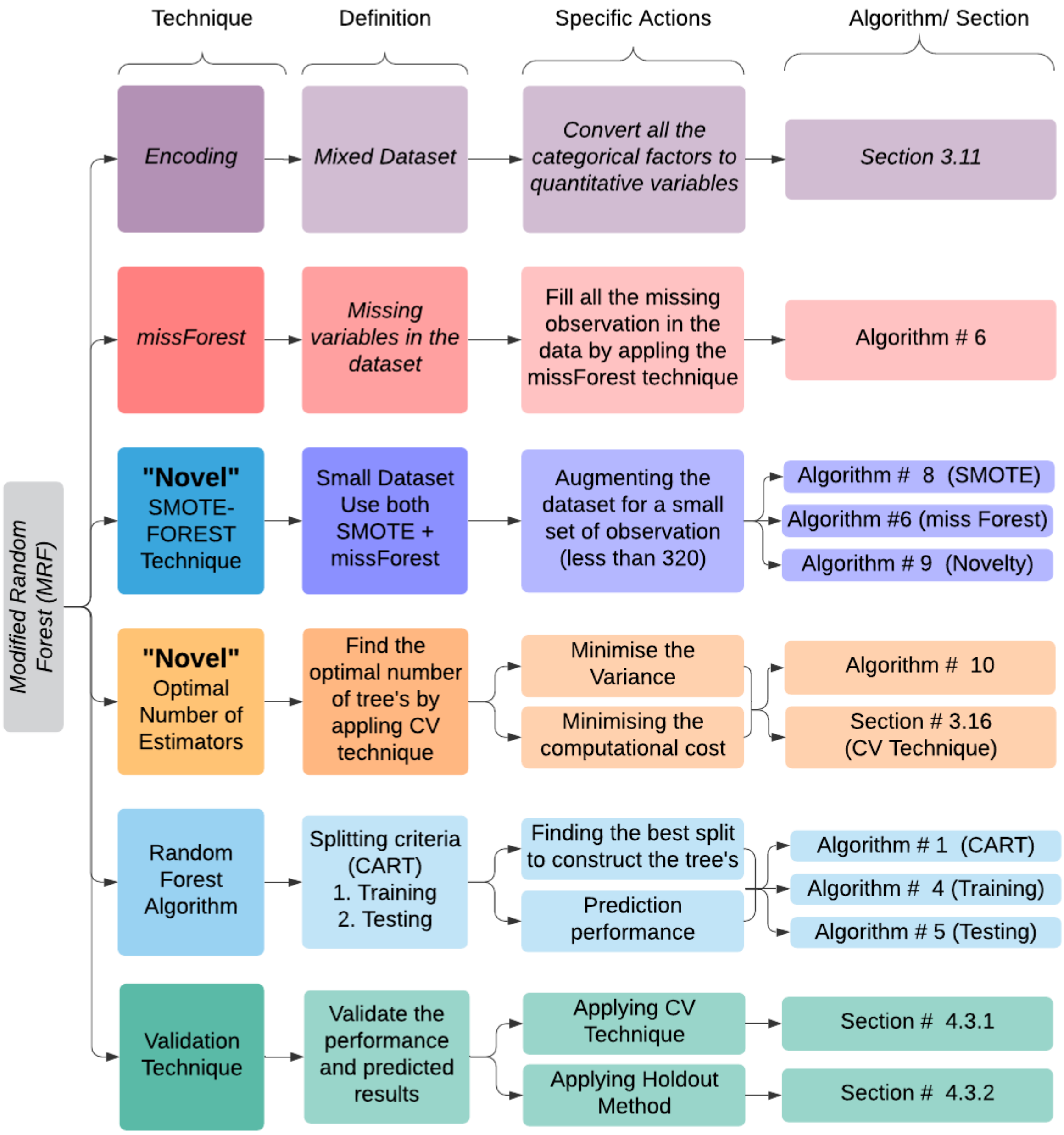

Technique
Definition
Specific Actions
Algorithm/ Section
Modified Random Forest (MRF)
Encoding
Mixed Dataset
Convert all the categorical factors to quantitative variables
Section 3.11
missForest
Missing variables in the dataset
Fill all the missing observation in the data by appling the missForest technique
Algorithm # 6
"Novel" SMOTE-FOREST Technique
Small Dataset Use both SMOTE + missForest
Augmenting the dataset for a small set of observation (less than 320)
Algorithm # 8 (SMOTE)
Algorithm #6 (miss Forest)
Algorithm # 9 (Novelty)
"Novel" Optimal Number of Estimators
Find the optimal number of tree's by appling CV technique
Minimise the Variance
Minimising the computational cost
Algorithm # 10
Section # 3.16 (CV Technique)
Random Forest Algorithm
Splitting criteria (CART) 1. Training 2. Testing
Finding the best split to construct the tree's
Prediction performance
Algorithm # 1 (CART)
Algorithm # 4 (Training)
Algorithm # 5 (Testing)
Validation Technique
Validate the performance and predicted results
Applying CV Technique
Applying Holdout Method
Section # 4.3.1
Section # 4.3.2

Figure 4.10: Chapter summary highlighting all work that has been done and the novel techniques proposed

4.3 Model Accuracy Assessment

Evaluating the machine learning model accuracy is an essential part of the development process. Moreover, the chosen evaluation metric should be representative of the primary purpose of the work. The MSE of the process response prediction from the real process response is selected as a metric for evaluating the predictive power of the model. The validation of the training phase output ensured the overall efficiency of the model and was used to prevent overfitting issues. The verification is then performed on a regression model at the testing stage by using a test dataset as input to the trained model. This test dataset is the remainder of the partitioned data from the original data collection; thus, it has the same features as the training dataset. The original dataset is partitioned into 80% training and 20% testing, respectively. Performance metrics are used for evaluation at both levels. Any overfitting issues can be identified by comparing the performance on the training and testing datasets; overfitting occurs when the training performance is comparatively better than the test results.

4.3.1 k-Fold Cross-Validation Technique

To evaluate the efficiency of the prediction model, ten runs of 10-fold cross-validation [89] were implemented on a defined dataset. The data is randomly shuffled, then split into ten subsets of the same size. Consecutively, one separate subset is evaluated (with 10% of the data), while the remaining dataset is used to fit the MRF algorithm. At the end of this step, the entire data collection of the analysed test set is used. Table 4.2 presents the results of the 10-fold cross-validation on the nickel-based superalloy dataset. The estimated variance (R^2) from the 10-fold cross-validation was 89.04% (+/- 0.1498). The k-fold cross-validation technique is also used in MRF to find the optimal number of estimators, as well as to verify the training data. A comparison with the previous work on the same dataset could not be conducted, as the researcher in [8] was unable to perform predictions on this dataset. As mentioned above, this problem occurred due to the simplifying assumption of linearity, which does not hold for this particular dataset. Comparisons with other published datasets are conducted and presented in chapter six; because the nickel-based superalloy dataset is Swansea-centric, only a few researchers have worked on it [8, 11].

Table 4.2: R^2 scores for 10-fold cross-validation on the nickel-based superalloy dataset

K-fold	1	2	3	4	5	6	7	8	9	10	**Mean**
R^2(%)	84.1	80.4	98.4	89.3	92.9	95.5	95.9	74.6	95.1	84.2	**89.04**

4.3.2 Holdout Method

The dataset was split into a training set and a testing set. The MRF was trained on the training set, while the testing set was used to see how well the generalised model could extract knowledge from unseen data. A typical split when using the holdout strategy involves using 80% of data for training and the remaining 20% for testing, as shown in Figure 4.11. This method was used for prediction testing in the MRF evaluation of the training phase on the nickel-based superalloy dataset. As Figure 4.12 shows, the model has a coefficient of determination R^2 = 81.3% and RMSE = 0.000699.

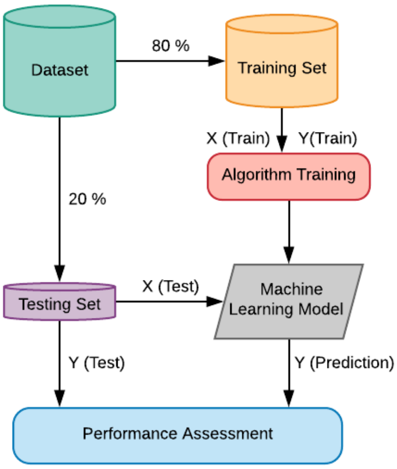


Figure 4.11: Validation on unseen data procedure

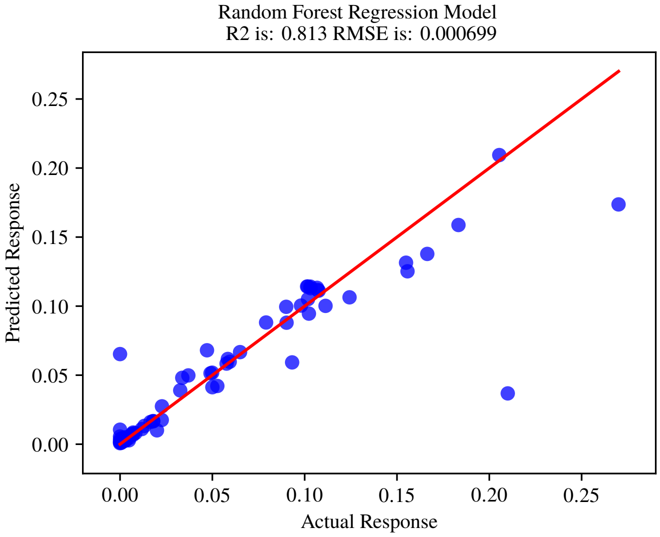


Figure 4.12: MRF regression model for nickel-based superalloy dataset

4.4 Missing Data Simulations

Missing data simulations are conducted for performance assessment and comparison between the SMOTE-FOREST and KDR algorithms. The strategy involves generating the missing data from a complete dataset by considering the following incremental levels: 5%, 10%, 15%, 20%, 25%, 30%, 40%, 50%, 60%. For each level, the missing data is randomly generated. The present work is assessed by calculating the Normalised Root Mean Squared Error (NRMSE) for continuous variables [99], which is defined as follows:

$$NRMSE = \sqrt{\frac{mean((X^{true} - X^{imp})^2)}{var(X^{true})}} \tag{4.2}$$

where

- X^{true}, represents the complete data matrix;
- X^{imp}, represents the imputed data matrix.

Moreover, $mean$ and var denote the evidential mean and variance calculated only using continuous missing values. Throughout the assessment of the categorical variables, the Proportion of Falsely Classified entries (PFCs) among the categorical missing values is considered, ∇F by using Equation 3.19. For all instances, outstanding success corresponds to a value close to zero, while inefficient output corresponds to a value of about one.

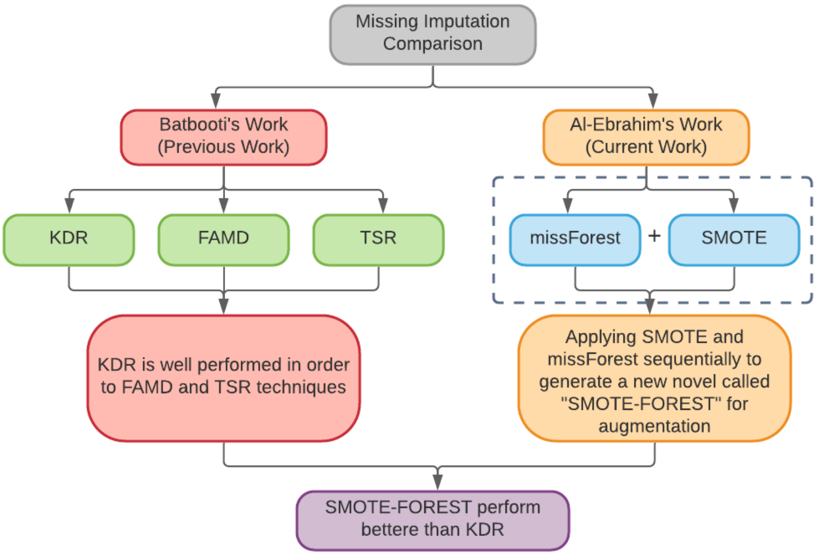

Missing Imputation Comparison
Batbooti's Work (Previous Work)
Al-Ebrahim's Work (Current Work)
KDR
FAMD
TSR
missForest
+
SMOTE
KDR is well performed in order to FAMD and TSR techniques
Applying SMOTE and missForest sequentially to generate a new novel called "SMOTE-FOREST" for augmentation
SMOTE-FOREST perform bettere than KDR

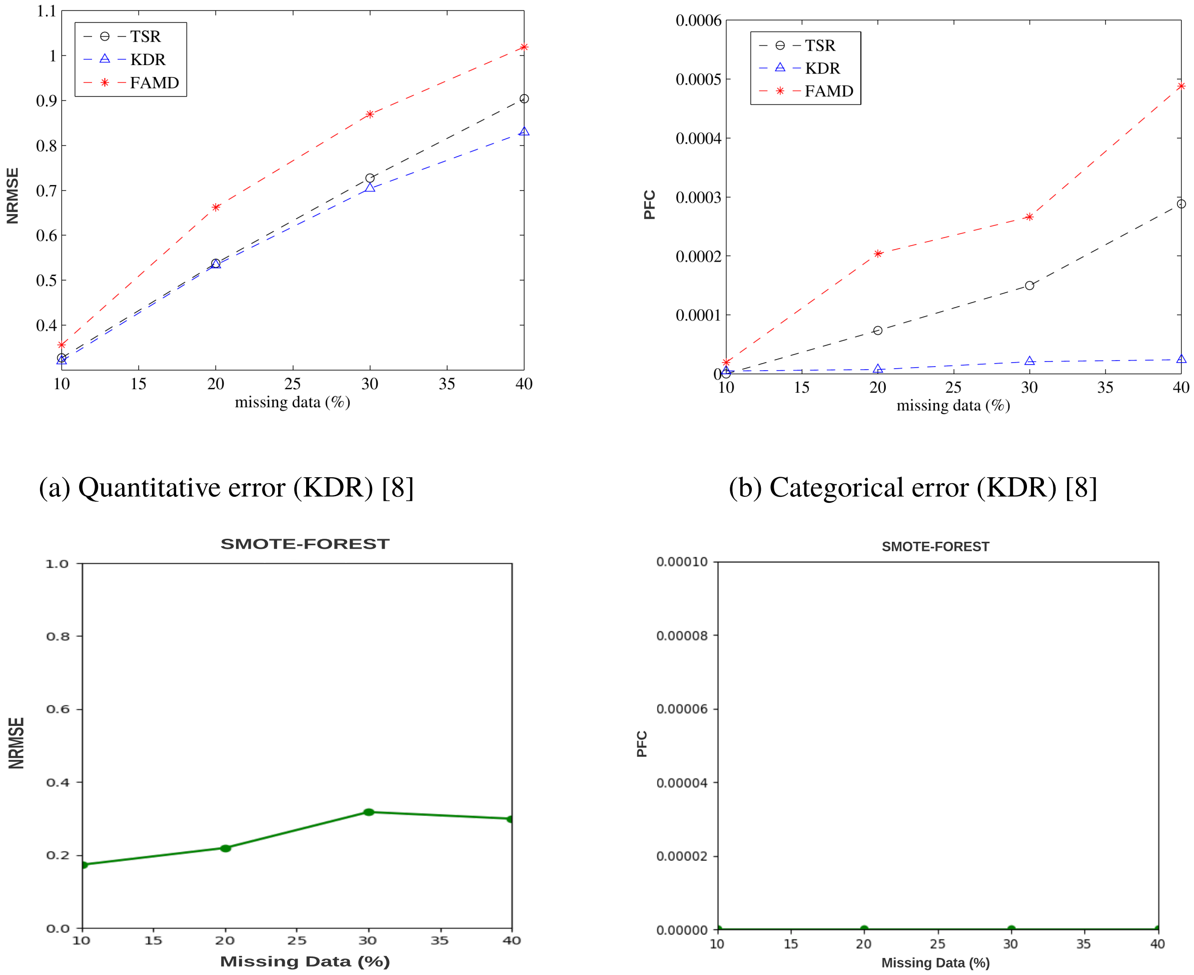
Figure 4.13: The path of comparing SMOTE-FOREST with KDR

Manufacturing Datasets

In order to assess the reliability of the SMOTE-FOREST algorithm, a new dataset with very large number of known observations with non-linear interactions among factors and responses was identified. A dataset consisting of 37 process factors that contribute to defects in the casting process was utilised. Here, 21 of the factors are categorical factors belonging to different categories, such as $\%Si$, $\%V$, $\%As$, etc; the other 16 factors are continuous factors, such as $\%Re$, $\%S$, $\%Hf$, etc. The total number of observations in this dataset was 20720. As a result, it is possible to randomly choose a significant proportion of observations and treat its response value as missing value. The accuracy of the missing data algorithm can then be verified. This data was used because it offers the following advantages:

- The dataset contains both quantitative and categorical variables.
- Non-linear interactions are observed.
- Because the data contains a large number of observations (20720), it can be used to check the computational cost and to visualise the performance of the algorithm on a large dataset.
- The data is based on a real-world example and was used in previous Swansea research.

Batbooti [8] used this dataset to assess the performance of his missing data algorithms: Known Data Regression (KDR), Factorial Analysis for Mixed Data (FAMD), and Trimmed Score Regression (TSR), as illustrated in Figure 4.13. He proposed that the best performance was achieved using the KDR algorithm as illustrated in Figures 4.14a and 4.14b. In the current work, moreover, a comparison was conducted between the SMOTE-FOREST and KDR method used in the QCA algorithm [8] for both quantitative and categorical data (see Figure 4.13). The proposed SMOTE-FOREST algorithm outperformed Batbooti's KDR algorithm as shown in Figure 4.14. For KDR algorithm [8], the error increased from 0.2 to 0.8 when the percentage of missing data increased to 40%. However for the proposed SMOTE-FOREST algorithm, the NRMSE error almost remained constant between 0.2-0.3. Moreover, the SMOTE-FOREST algorithm has a lower computational cost than the KDR algorithm [8], which represents an additional advantage (see Section 6.5 and Appendix B for a detailed review of the computational cost comparison). To sum up, the SMOTE-FOREST method performs better than the KDR method (that was used in the previous work) [8] in both quantitative and categorical contexts by reducing imputation error, in many cases by more than 50%, for the manufacturing data.



(a) Quantitative error (KDR) [8]

(b) Categorical error (KDR) [8]

(c) Quantitative error (SMOTE-FOREST)

(d) Categorical error (SMOTE-FOREST)

Figure 4.14: Missing data simulations - KDR versus SMOTE-FOREST

4.5 Discussion of Results

To verify the structure of the base estimators of Random Forest, the author extracted a random sample decision tree from the forest to demonstrate how decisions travel through the decision paths from the root to the leaf. To recognise what exists in the data or how decisions are made, the MRF model was trained utilising the nickel-based superalloy dataset with a forest size equivalent to 90 trees. One tree is randomly picked from the forest, as seen in Figure 4.16. The root node, as shown in Figure 4.15, has four different values, as described below. **The feature** $\%Nb$ value represents the splitting threshold point, **MSE** represents the weighted average score, **sample** represents the number of the samples in the node, and **value** represents the average response in the node. Every node represents a sub-tree with right and left branches: the left branch represents the true test value, while the right branch represents the false test value. At a growing node, all features are evaluated to determine the best split. Here, the best split is the one that yields the least weighted MSE within the actual response of the samples and the predicted response in the child nodes. The predicted response is the average response of all instances in that node. As can be seen in the tree graph, $\%Nb$ is the feature that best splits when considered at a threshold of 0.745, as described above. If the value is less than or equivalent to 0.745, the predicted response is 0.108; if the feature $\%Nb$ is greater than 0.745, the predicted response is 0.015. When a new unseen sample is tested, it traverses each of these nodes until it comes to an end at the last node. Thus, the average response of all instances in that node is regarded as the predicted response. A growing node is often divided into two other nodes. The control variables used to tune the model and limit overfitting are as follows: Number of estimators, Bootstrap with replacement, number of samples that need to be present for each tree node before split, and minimum sample leaf in each node. Figure 4.16 represents the $\%Nb$ feature, which has a minimum sample leaf equal to three (as used in the proposed algorithm). By contrast, Figure 4.17 represents the $\%Fe$ feature, which has a minimum sample leaf equal to five. The tree of the $\%Nb$ feature has more splits than the tree of the $\%Fe$ feature, since the minimum sample of three requires more splitting to be performed to reach the desired output.

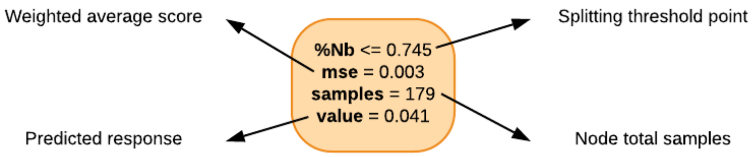


Figure 4.15: Root node elements for $\%Nb$ factor

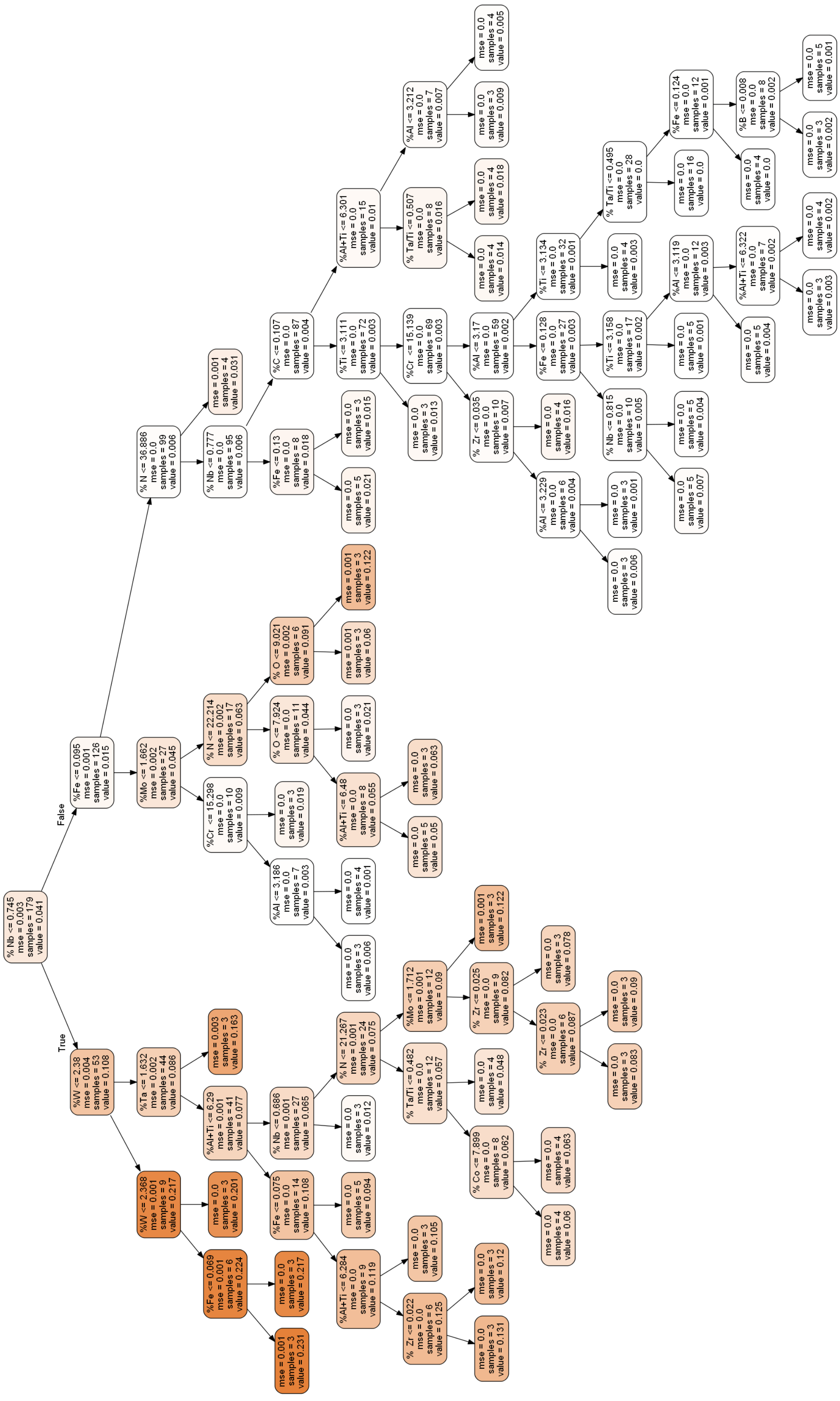


Figure 4.16: Sample tree from the generated forest for %*Nb* sample

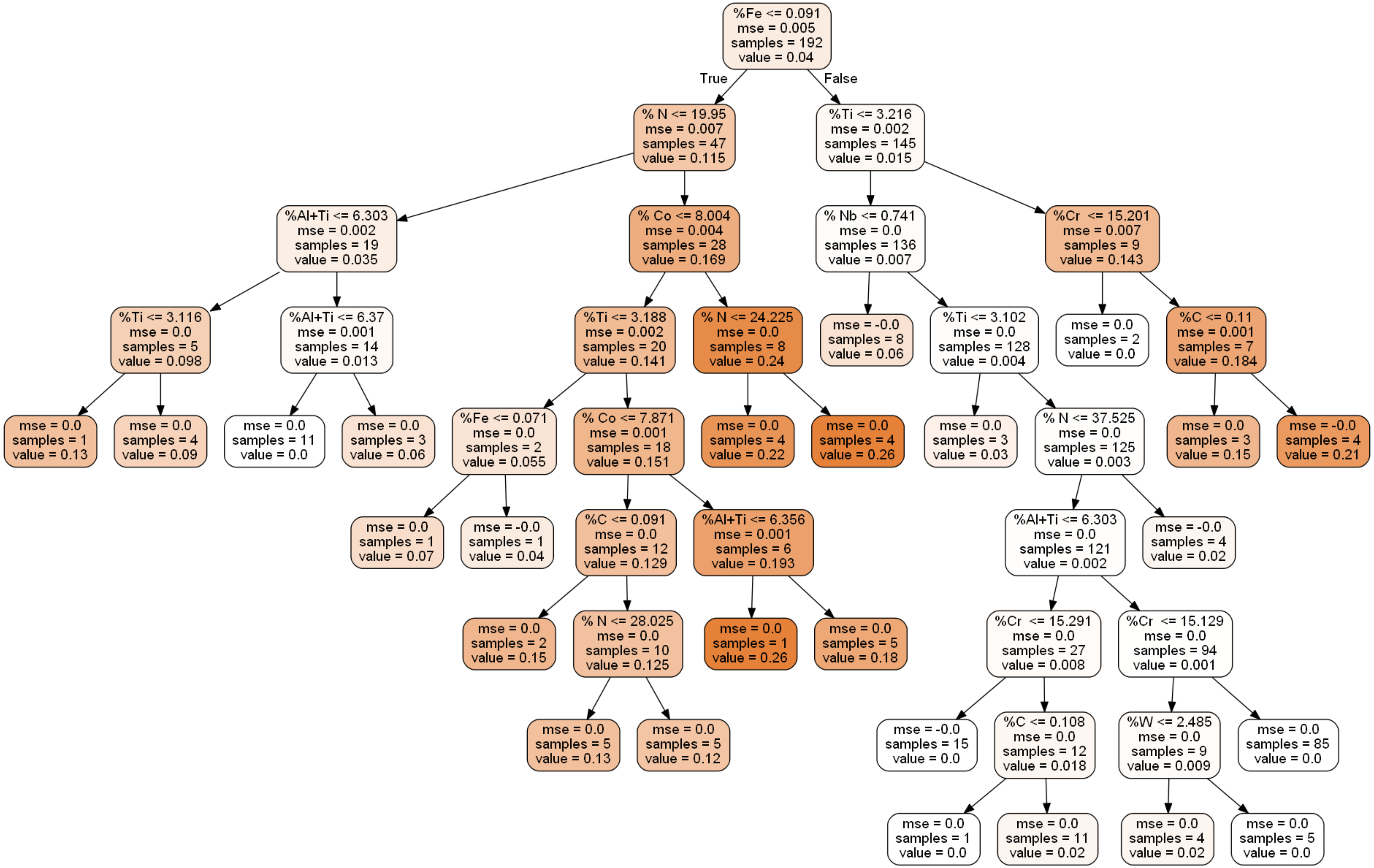


Figure 4.17: Sample tree from the generated forest for $\%Fe$ sample

The novel data augmentation technique presented in this chapter achieved good performance in augmenting 60 observations to 360 synthetic observations while maintaining the non-linear relationships between the process factors and the process response to the greatest extent possible. This method eliminates the drawbacks associated with repeating the original observation data, as in the random oversampling technique, and also extends the SMOTE algorithm to enable its use in non-linear regression problems. A more detailed comparison with the QCA algorithm [8] was presented in Section 4.2.1. The novel optimal forest sizing approach illustrated in this chapter significantly reduced the computational cost, especially when increasing the unnecessary size of the forest (see Appendix B). Moreover, the novel approach gives the traditional Random Forest an automatic tuning feature to minimise the variance error and overfitting in the prediction. To enhance the predictive performance, ten runs of the MRF algorithm were conducted, with 10-fold cross-validation used for a given run. This data is arbitrarily subdivided into ten subsets of the same size. Sequentially, one separate subset is evaluated, after which the remaining nine subsets are used to fit the MRF algorithm. Finally, the entire dataset of the analysed test set is used. The majority of the values are in/near the Gaussian distribution, indicating an intense fitting. Figure 4.18 displays the MRF scatter plot (predicted vs observed values) with the statistical regression method. The MRF prediction is close to the diagonal line, which represents the ideal forecast, throughout most of the output range. The results of the predictive evaluation of the training and test data further demonstrate that the proposed novel augmentation technique was robust for non-linearly related process factors.

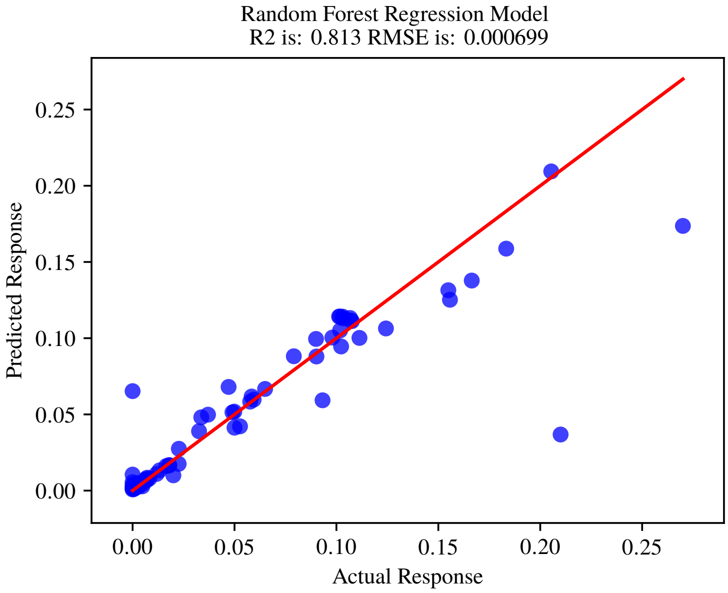


Figure 4.18: Te proposed regression model for nickel-based superalloy dataset

4.6 Conclusions

The MRF algorithm offers a good prediction accuracy by considering the non-linear relationships between investment casting process factors. The author could not conduct a comparison with other externally published work because the nickel-based superalloy dataset is Swansea-centric. In addition, a comparison with the author's previous work [8] could not be undertaken because the QCA algorithm is unable to deal with non-linear interactions [12]. Furthermore, the use of the QCA algorithm produces poor results. The most significant advantage of the proposed algorithm is its ability to make accurate predictions regardless of the occurrence of any non-linear interactions. The model achieved robust mapping of the relationship between the process inputs and the shrinkage defects as process output; therefore, it was a reliable base for the extraction of the causal relationship in the process. It can also be concluded that the missing data imputation algorithm used (SMOTE-FOREST) yields better results than the previous work (KDR) [8], as concluded in Figure 4.19. A detailed comparison with different datasets and case studies will be carried out in chapter six in order to compare the performance of the proposed algorithm with other published work. Moreover, obtaining the cause and effect knowledge will be discussed in chapter five.

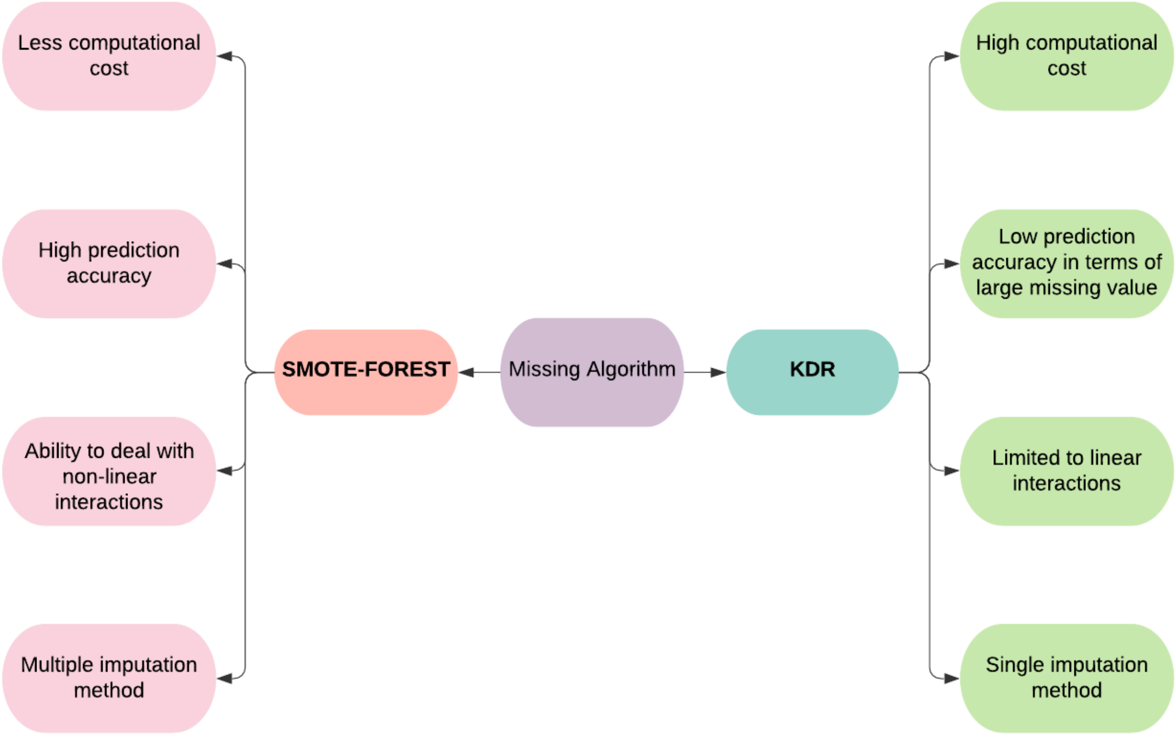

Less computational cost
High prediction accuracy
Ability to deal with non-linear interactions
Multiple imputation method
SMOTE-FOREST
Missing Algorithm
KDR
High computational cost
Low prediction accuracy in terms of large missing value
Limited to linear interactions
Single imputation method

Figure 4.19: Key differences between SMOTE-FOREST and KDR methods

Chapter 5

Causal Knowledge Discovery

5.1 Introduction

The predictive model introduced in chapter four acts as a transfer (regression) function between the process inputs and process response, albeit with results that are difficult to interpret. Using this transfer function, it is possible to predict the process response/quality output from given process inputs. However, this is not sufficient by itself to optimise the process and reduce defects. In other words, the manufacturers need more specific recommendations, such as which critical factors should be controlled and to what extent in order to to improve quality and reduce waste. Investment casting process optimisation is defined here as a methodology of using historical process data to discover a more refined process knowledge through studying the causal relationships between process factors in order to relate mechanical properties and casting defects with processing conditions. Chapter five will go deeper into this black-box and extract the cause and effect relationship that governs the relationship between shrinkage defects and process factors (i.e. chemical composition). This chapter introduces two novel techniques. The first technique is a completely new causal framework for the Modified Random Forest algorithm. This algorithm searches for the optimal decision path to make the traditional RF more interpretable and provide a comprehensive inference analysis framework that enables causal knowledge extraction from any complicated process. It further guides the process engineer to optimise process performance through the estimation of optimal limits for each process factor in order to minimise process defects and maximise yield. The second technique is a scoring algorithm for ranking the most critical process factor based on Decision Path Search (DPS). The remainder of this chapter is structured as follows. Section 5.2 describes the optimal process operating range to reduce defect outputs. This section is further subdivided into three parts: the novel Decision Path Search technique, the critical process factors that make the most significant contribution to defects by applying the novel technique, and the results of a foundry case study. Section 5.3 goes on to define the new process limits and odds ratio calculations. Finally, the discussion and conclusion can be found in Sections 5.4 and 5.5 respectively.

5.2 Optimal Process Operating Range

In this section, a novel approach for estimating optimal operating limits for mixed data is introduced, as shown in Figure 5.1. The decision tree structure can be analysed to gain further insight into the relationship between the process factors and the output target. In the present research, the author demonstrates how to retrieve the causal relationships required to score the critical factors and estimate their optimal operating range, which yields process output values that satisfy threshold limits.

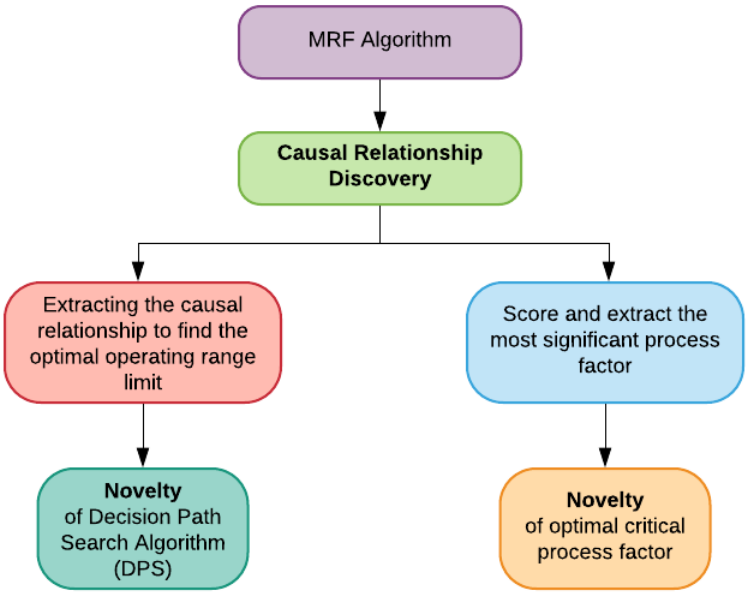

MRF Algorithm
Causal Relationship Discovery
Extracting the causal relationship to find the optimal operating range limit
Score and extract the most significant process factor
Novelty of Decision Path Search Algorithm (DPS)
Novelty of optimal critical process factor

Figure 5.1: Relational overview of the techniques used to devise the novel (Causal Relationship Discovery) algorithm

5.2.1 Novel Technique for Decision Path Search Algorithm

The novel technique introduced here is based on traversing the internal structures of decision trees to retrieve deeper process knowledge. The concept of the structured decision trees is not a complicated one. It involves evaluation of the partitions of the Bayes model through recursively partitioning the input data X into sub-spaces and subsequently assigning the prediction value of Y to all objects X within each sub-space. The decision tree used here has a binary structure. Nodes that do not have any children are leaf nodes, and the decision path consists of all nodes that were reached by a sample. All decision paths start from the root and end with the leaf node. The flowchart for all steps used in estimating the optimal process settings for quantitative and categorical variables is presented in Figure 5.2, while the pseudo-code is summarised in Algorithms 12 and 13. For each tree in the forest, the following main steps are implemented. First, all leaves are collected. The Penalty Matrix (PM) scaling is applied for all leaves to penalise undesired process responses as described in Algorithm 12. Subsequently, the distribution range for all leaves' output values is categorised into four quartiles. A leaf is considered an optimal leaf if its importance (value) is in the first quartile distribution range, while all other leaves are considered undesirable (avoidance) leaves. The optimal route is then drawn from the optimal leaf to the root node, as in Figure 5.3; moreover, an avoidance route drawn from an avoidance leaf to the root node is also presented in the algorithm. Both intervals, as shown in Figure 5.3, are constructed for every factor and aggregated from all trees. Finally, the algorithm averages all optimal and avoidance intervals for each factor and constructs the new limit ranges for each factor in the dataset.

Algorithm 12: Penalty Matrix Algorithm [13]

```
Function PenaltyMatrix($\hat{Y}$, $Th_{min}$, $Th_{max}$):
    $D = 0, E = 1$
        // For lower the better case (LB), while $D = 1$ and $E = 0$ for higher the better (HB)
    $V \leftarrow$ size of $(\hat{Y})$
    $PM \leftarrow$ Array of zeros($V$)
    for j=1 $\longrightarrow$ $V$ do
        if $\hat{Y} \leq Th_{min}$ then
            $PM = D$
        else if $\hat{Y} \geq Th_{max}$ then
            $PM = E$
        else
            $PM = \frac{\hat{Y} - Th_{min}}{Th_{max} - Th_{min}}$
        end
    end
    return $PM$
End Function
```

Algorithm 13: Novelty of Process Factor Limits Algorithm

```
Fit a Random Forest on the training set $L$ by using Algorithm 11 presented in chapter 4.
Optimal-Intervals=[]
Avoidance-Intervals=[]
for $b = 1$ to $B$ do
    Leafs = Collect all terminal nodes of tree $b$
    $PM_{Leafs} \leftarrow$ PenaltyMatrix($\hat{Y}, Th_{min}, Th_{max}$) (from Algorithm 12)
    $PM_{Q25} \leftarrow$ Quartile($PM_{Leafs}$,0.25)
    Optimal-Leaf=[]
    Avoidance-Leaf=[]
    for $i = 1, ...., n_{leafs}$ do
        if $PM[i] < PM_{Q25}$ then
            Optimal-Leaf $\leftarrow$ leaf[$i$]
            Draw a route from leaf[$i$] to root node
            Optimal-Intervals[] $\leftarrow$ Append all optimal interval for each factor        // factor
              split threshold value
        else
            Avoidance-Leaf $\leftarrow$ leaf[$i$]
            Draw a route from leaf[$i$] to root node
            Avoidance-Intervals [] $\leftarrow$ Append all avoidance interval for each factor
              // factor split threshold value
        end
    end
end
Optimal-Interval=[]                          // estimate the optimal intervals for every factor
Avoidance-Interval=[]                        // estimate the avoidance intervals for every factor
for $j$=1 to $n$ do
    $f \leftarrow X[j]$
    if $f$ is quantitative then
        Optimal-Interval[$j$]= mean(Optimal-Intervals[$j$])
        Avoidance-Interval=[$j$]= mean(Avoidance-Intervals[$j$])
    else
        Optimal-Interval[$j$]= mode(Optimal-Intervals[$j$])
        Avoidance-Interval=[$j$]= mode(Avoidance-Intervals[$j$])
    end
end
Compare optimal and avoidance intervals
```

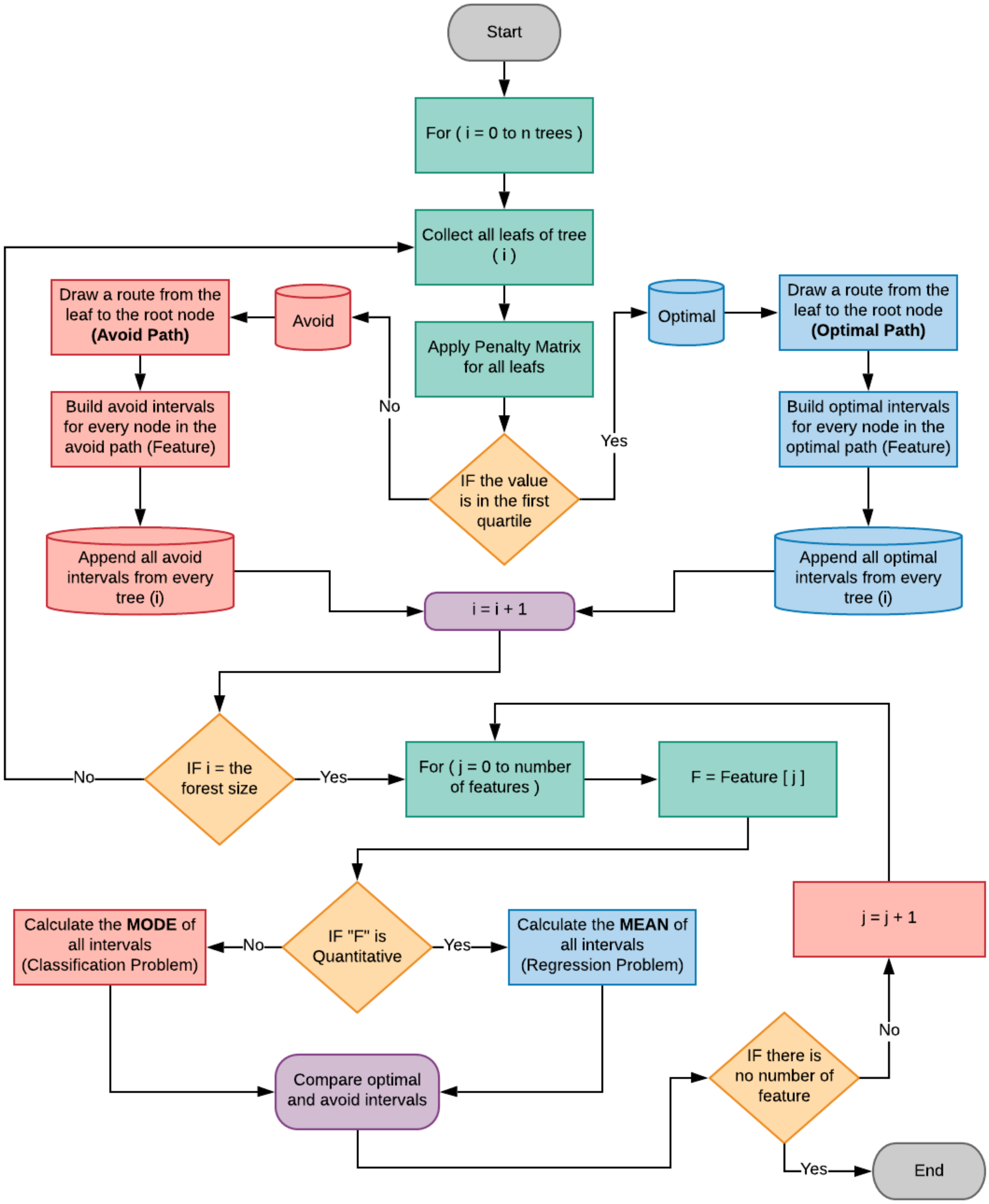

Start
For (i = 0 to n trees)
Collect all leafs of tree (i)
Apply Penalty Matrix for all leafs
IF the value is in the first quartile
No
Yes
Avoid
Optimal
Draw a route from the leaf to the root node **(Avoid Path)**
Build avoid intervals for every node in the avoid path (Feature)
Append all avoid intervals from every tree (i)
Draw a route from the leaf to the root node **(Optimal Path)**
Build optimal intervals for every node in the optimal path (Feature)
Append all optimal intervals from every tree (i)
i = i + 1
IF i = the forest size
No
Yes
For (j = 0 to number of features)
F = Feature [j]
IF "F" is Quantitative
No
Yes
Calculate the **MODE** of all intervals (Classification Problem)
Calculate the **MEAN** of all intervals (Regression Problem)
Compare optimal and avoid intervals
IF there is no number of feature
No
Yes
j = j + 1
End

Figure 5.2: Decision Path Search (DPS) algorithm

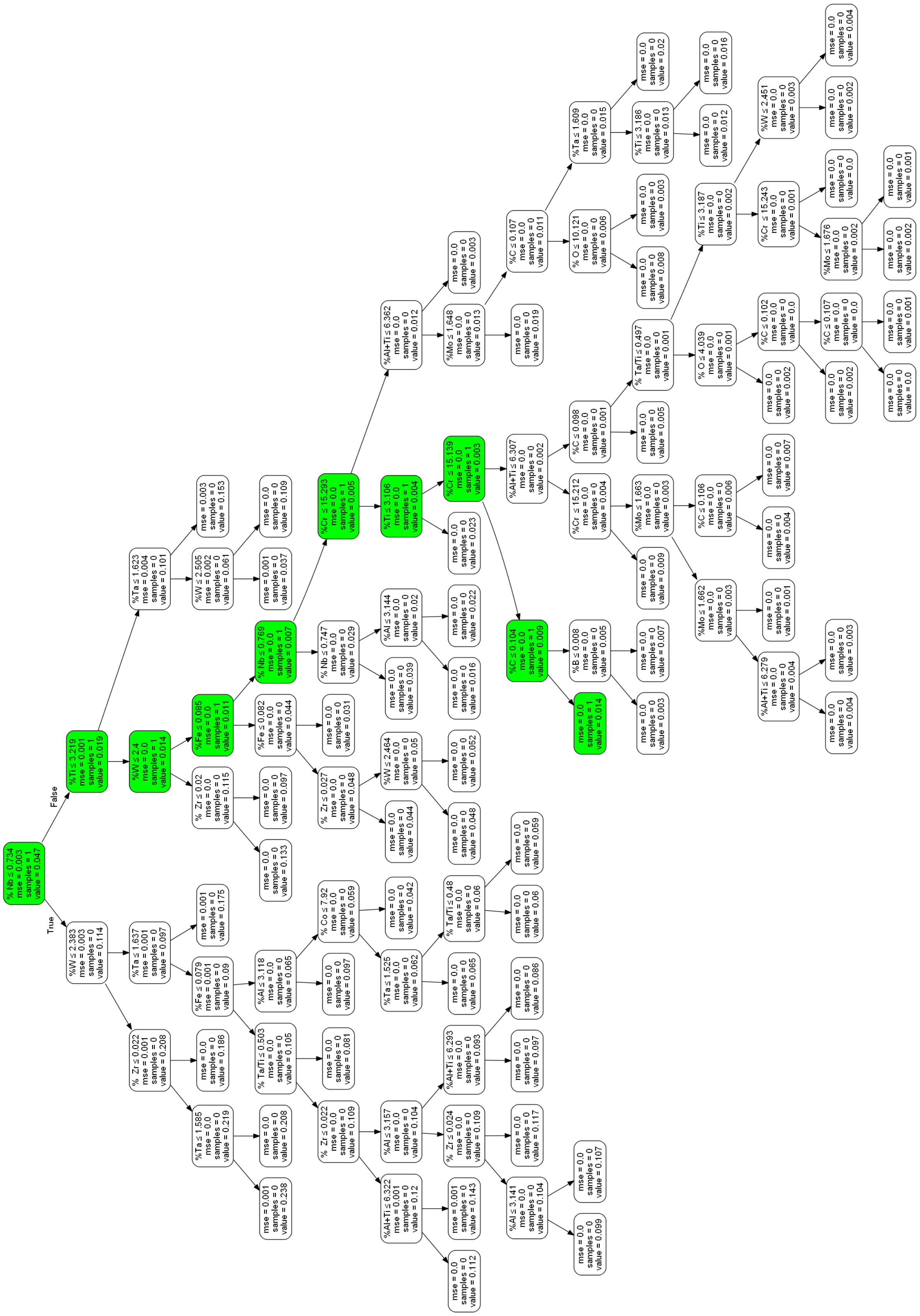

% Nb ≤ 0.734 mse = 0.003 samples = 1 value = 0.047
True
False
%W ≤ 2.383 mse = 0.003 samples = 0 value = 0.114
%Ti ≤ 3.219 mse = 0.001 samples = 1 value = 0.019
%Ta ≤ 1.637 mse = 0.001 samples = 0 value = 0.097
% Zr ≤ 0.022 mse = 0.001 samples = 0 value = 0.208
mse = 0.001 samples = 0 value = 0.175
%Fe ≤ 0.079 mse = 0.001 samples = 0 value = 0.09
mse = 0.0 samples = 0 value = 0.186
%Ta ≤ 1.585 mse = 0.0 samples = 0 value = 0.219
mse = 0.0 samples = 0 value = 0.208
mse = 0.001 samples = 0 value = 0.238
%Al ≤ 3.118 mse = 0.0 samples = 0 value = 0.065
%Co ≤ 7.92 mse = 0.0 samples = 0 value = 0.059
mse = 0.0 samples = 0 value = 0.097
mse = 0.0 samples = 0 value = 0.042
%Ta ≤ 1.525 mse = 0.0 samples = 0 value = 0.062
%Ta/Ti ≤ 0.48 mse = 0.0 samples = 0 value = 0.06
mse = 0.0 samples = 0 value = 0.059
mse = 0.0 samples = 0 value = 0.06
mse = 0.0 samples = 0 value = 0.065
% Ta/Ti ≤ 0.503 mse = 0.0 samples = 0 value = 0.105
mse = 0.0 samples = 0 value = 0.081
% Zr ≤ 0.022 mse = 0.0 samples = 0 value = 0.109
%Al ≤ 3.157 mse = 0.0 samples = 0 value = 0.104
%Al+Ti ≤ 6.293 mse = 0.0 samples = 0 value = 0.093
mse = 0.0 samples = 0 value = 0.086
mse = 0.0 samples = 0 value = 0.097
% Zr ≤ 0.024 mse = 0.0 samples = 0 value = 0.109
mse = 0.0 samples = 0 value = 0.117
%Al ≤ 3.141 mse = 0.0 samples = 0 value = 0.104
mse = 0.0 samples = 0 value = 0.107
mse = 0.0 samples = 0 value = 0.099
%Al+Ti ≤ 6.322 mse = 0.001 samples = 0 value = 0.12
mse = 0.001 samples = 0 value = 0.143
mse = 0.0 samples = 0 value = 0.112
%Ta ≤ 1.623 mse = 0.004 samples = 0 value = 0.101
mse = 0.003 samples = 0 value = 0.153
%W ≤ 2.505 mse = 0.002 samples = 0 value = 0.061
mse = 0.0 samples = 0 value = 0.109
mse = 0.001 samples = 0 value = 0.037
%W ≤ 2.4 mse = 0.0 samples = 1 value = 0.014
% Zr ≤ 0.02 mse = 0.0 samples = 0 value = 0.115
mse = 0.0 samples = 0 value = 0.097
mse = 0.0 samples = 0 value = 0.133
%Fe ≤ 0.085 mse = 0.0 samples = 1 value = 0.011
%Fe ≤ 0.082 mse = 0.0 samples = 0 value = 0.044
mse = 0.0 samples = 0 value = 0.031
% Zr ≤ 0.027 mse = 0.0 samples = 0 value = 0.048
%W ≤ 2.464 mse = 0.0 samples = 0 value = 0.05
mse = 0.0 samples = 0 value = 0.052
mse = 0.0 samples = 0 value = 0.048
mse = 0.0 samples = 0 value = 0.044
% Nb ≤ 0.769 mse = 0.0 samples = 1 value = 0.007
% Nb ≤ 0.747 mse = 0.0 samples = 0 value = 0.029
%Al ≤ 3.144 mse = 0.0 samples = 0 value = 0.02
mse = 0.0 samples = 0 value = 0.022
mse = 0.0 samples = 0 value = 0.016
mse = 0.0 samples = 0 value = 0.039
%Cr ≤ 15.293 mse = 0.0 samples = 1 value = 0.005
%Al+Ti ≤ 6.362 mse = 0.0 samples = 0 value = 0.012
mse = 0.0 samples = 0 value = 0.003
%Mo ≤ 1.648 mse = 0.0 samples = 0 value = 0.013
mse = 0.0 samples = 0 value = 0.019
%C ≤ 0.107 mse = 0.0 samples = 0 value = 0.011
%Ta ≤ 1.609 mse = 0.0 samples = 0 value = 0.015
mse = 0.0 samples = 0 value = 0.02
%Ti ≤ 3.186 mse = 0.0 samples = 0 value = 0.013
mse = 0.0 samples = 0 value = 0.016
mse = 0.0 samples = 0 value = 0.012
% O ≤ 10.121 mse = 0.0 samples = 0 value = 0.006
mse = 0.0 samples = 0 value = 0.003
mse = 0.0 samples = 0 value = 0.008
%Ti ≤ 3.106 mse = 0.0 samples = 1 value = 0.004
mse = 0.0 samples = 0 value = 0.023
%Cr ≤ 15.139 mse = 0.0 samples = 1 value = 0.003
%Al+Ti ≤ 6.307 mse = 0.0 samples = 0 value = 0.002
%C ≤ 0.098 mse = 0.0 samples = 0 value = 0.001
mse = 0.0 samples = 0 value = 0.005
% Ta/Ti ≤ 0.497 mse = 0.0 samples = 0 value = 0.001
%Ti ≤ 3.187 mse = 0.0 samples = 0 value = 0.002
%W ≤ 2.451 mse = 0.0 samples = 0 value = 0.003
mse = 0.0 samples = 0 value = 0.004
mse = 0.0 samples = 0 value = 0.002
%Cr ≤ 15.243 mse = 0.0 samples = 0 value = 0.001
mse = 0.0 samples = 0 value = 0.0
%Mo ≤ 1.676 mse = 0.0 samples = 0 value = 0.002
mse = 0.0 samples = 0 value = 0.001
mse = 0.0 samples = 0 value = 0.002
% O ≤ 4.039 mse = 0.0 samples = 0 value = 0.001
mse = 0.0 samples = 0 value = 0.002
%C ≤ 0.102 mse = 0.0 samples = 0 value = 0.0
%C ≤ 0.107 mse = 0.0 samples = 0 value = 0.0
mse = 0.0 samples = 0 value = 0.002
mse = 0.0 samples = 0 value = 0.001
mse = 0.0 samples = 0 value = 0.0
%Cr ≤ 15.212 mse = 0.0 samples = 0 value = 0.004
mse = 0.0 samples = 0 value = 0.009
%Mo ≤ 1.663 mse = 0.0 samples = 0 value = 0.003
%C ≤ 0.106 mse = 0.0 samples = 0 value = 0.006
mse = 0.0 samples = 0 value = 0.007
mse = 0.0 samples = 0 value = 0.004
%Mo ≤ 1.662 mse = 0.0 samples = 0 value = 0.003
mse = 0.0 samples = 0 value = 0.001
%Al+Ti ≤ 6.279 mse = 0.0 samples = 0 value = 0.004
mse = 0.0 samples = 0 value = 0.003
mse = 0.0 samples = 0 value = 0.004
%C ≤ 0.104 mse = 0.0 samples = 1 value = 0.009
%B ≤ 0.008 mse = 0.0 samples = 0 value = 0.005
mse = 0.0 samples = 0 value = 0.007
mse = 0.0 samples = 0 value = 0.003
mse = 0.0 samples = 1 value = 0.014

Figure 5.3: Regression tree of optimal Decision Path Search (DPS) for a single factor (%*Nb*) in the nickel-based superalloy dataset

5.2.2 Critical Process Factors

The objective of this work is not only to find the most appropriate models for predicting the process response but also to recognise which of the input parameters are the most significant for making predictions, as this will lead to a deeper understanding of the process under study. In this case, typical Random Forests [17] utilise a range of frameworks for determining the importance of the input variable and thereby increasing the overall model accuracy. The most common mechanism used to compute feature importance is the mean decrease in the impurity importance of a feature. This is calculated by measuring how effective the feature is at variance when creating the decision trees that make up the Random Forest. However, the drawback of this conventional mechanism is that it can be affected by noise in the data. In this section, a built-in Random Forest of factor importance features called 'Permutation Importance' is reviewed. Moreover, a robust novel method based on the contribution of each factor to the prediction power and the optimal operating range is introduced.

Permutation Importance

Random Forest can use out-of-bag measurements to assess the sensitivity and importance of each input parameter. This is a Random Forest function that was included in Breiman's [17] original paper and was more recently explored by Strobl [100] and Genuer [101]. The feature needed to define a sample out-of-the-bag (OOB) for a single tree (O_b) for the b_{th} tree is referred to as T_b; i.e., observations that have not been included in the fitting of this single tree. Features are selected based on their impact on the prediction accuracy, while the significant feature is determined through Random Forest by assigning a score to each variable; the sum of all variable scores is unity. The score assigned to a variable is defined as the overall average reduction in the accuracy of the forecasts of all trees as the values of that variable are permuted. The input variable that induces the greatest reduction in accuracy after permutation is defined as the most significant. Once the entire forest has been trained, the permutation importance of variable X_j is measured by comparing the OOB prediction accuracy ($h(T_b)$) of a single tree, i.e. the classification rate (classification trees) or the mean squared error (regression trees) before and after the X_j feature has been permuted. The permutation importance of the j_{th} feature $I_{permute}(j)$ is defined in Equation 5.1 as follows [101]:

$$I_{permute}(j) = \frac{1}{B}\sum_{b=1}^{B} \underbrace{\left\{ \frac{\sum_{i \in O_b}(Y_i - \hat{Y}_i^b)^2}{|O_b|} - \frac{\sum_{i \in O_b}(Y_i - \hat{Y}_{i,\pi_j}^b)^2}{|O_b|} \right\}}_{I_{permute}(j)^b} \tag{5.1}$$

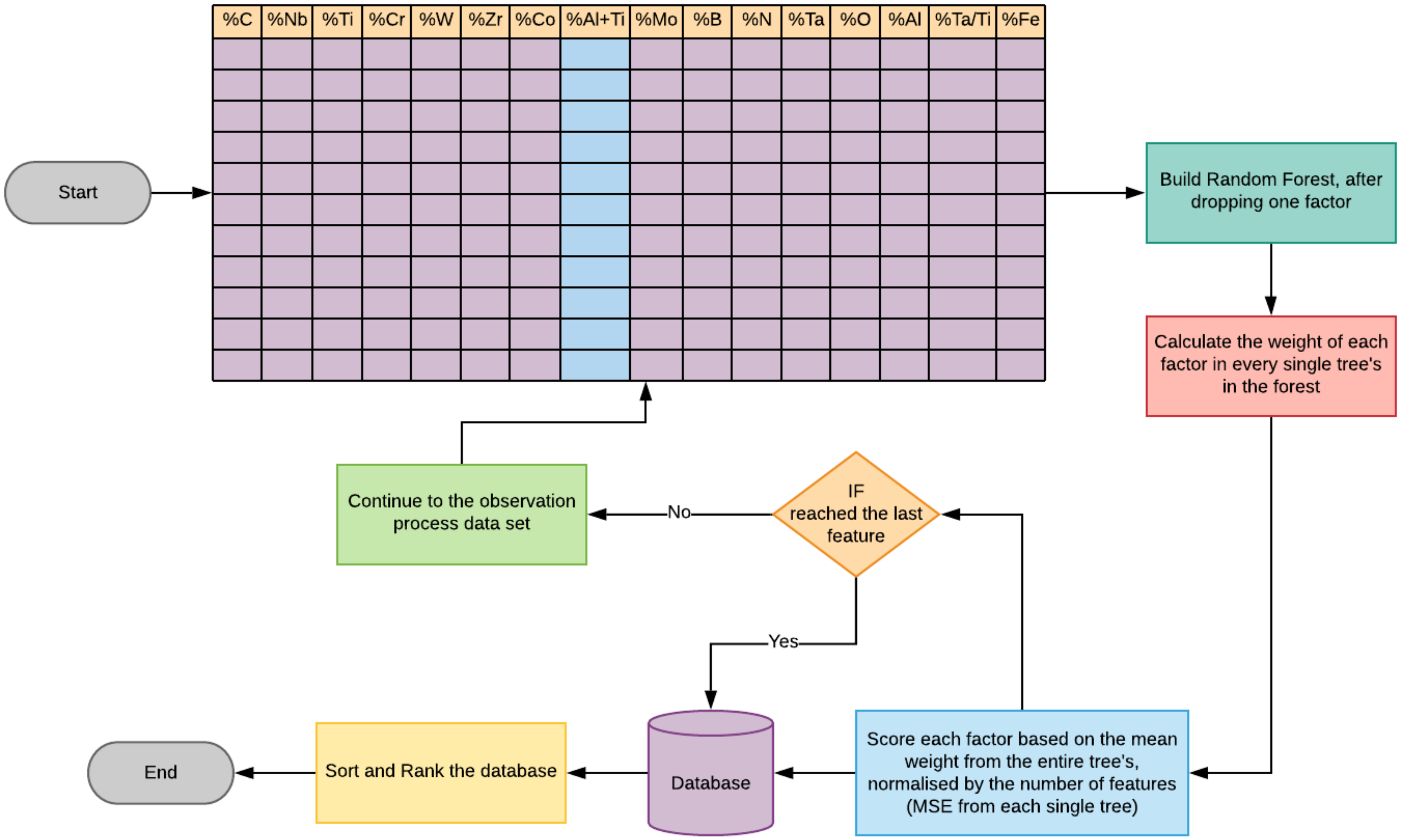

Start
%C
%Nb
%Ti
%Cr
%W
%Zr
%Co
%Al+Ti
%Mo
%B
%N
%Ta
%O
%Al
%Ta/Ti
%Fe
Build Random Forest, after dropping one factor
Calculate the weight of each factor in every single tree's in the forest
Score each factor based on the mean weight from the entire tree's, normalised by the number of features (MSE from each single tree)
IF reached the last feature
No
Continue to the observation process data set
Yes
Database
Sort and Rank the database
End

Figure 5.4: Permutation Importance schematic diagram

The theory behind this is as follows: if this feature is applied to the prediction or has had an effect on the target, the precision would be decreased. Ultimately, the average of each reduction of the total trees yields the permutation importance. Algorithm 14 presents the steps required to quantify the importance of the variables, while Figure 5.4 contains a schematic sample of the permutation importance method.

Algorithm 14: Permutation Importance Algorithm [101]

```
Fit a Random Forest on the training set $L$ using Algorithm 4 presented in chapter three.
for $b = 1$ to $B$ do
    Compute the OOB prediction accuracy of the $b_{th}$ tree $h(T_b)$.
    Permute randomly the observations of the feature $X_j$ in the OOB sample.
    Re-compute the OOB prediction accuracy of the $b_{th}$ tree $h(T_b)$ using the permuted
      input.
    Compute $I_{permute}(j)^b$
end
Compute the average decrease of prediction accuracy over the entire trees i.e. $I_{permute}(j)$
```

The results of the permutation feature importance for the nickel-based superalloy dataset are shown in Figure 5.5. The ranking shown in Figure 5.5 makes sense from a metallurgical point of view. Since the main chemical element in the Nicked-based superalloy is Iron (%Fe), it can be seen that the absence of this element results in the highest penalty (ferrous metals).

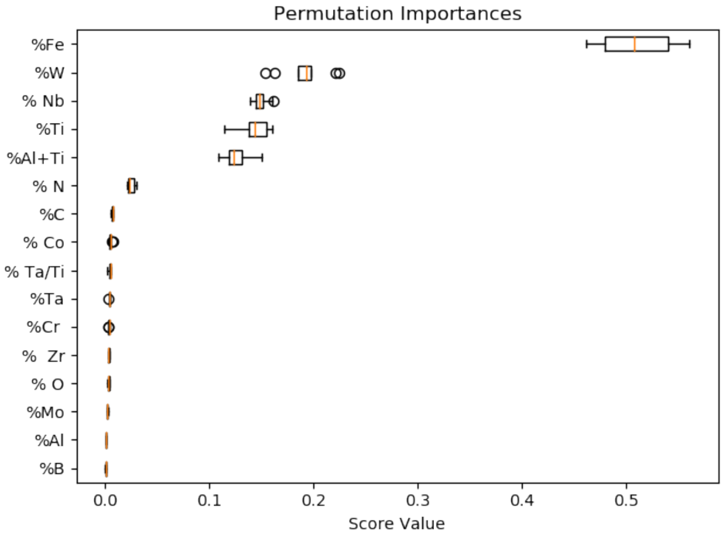


Figure 5.5: Permutation Importance ranking for nickel-based superalloy dataset

Permutation importance gives a score for each factor based on its contribution to the overall prediction power. This can be conceptualised as the ranking of critical process factors involved in casting the nickel-based superalloy. However, ranking the process factors alone is not enough for process optimisation; the optimal limits for each factor that reduce defects and increase the quality of process outputs should also be identified. Accordingly, in the next part, a novel technique used to obtain the optimal process limits based on the trained MRF model is presented.

Novel Technique for Finding Optimal Critical Process Factor

The novel approach to extracting the most significant process factor that contributes to the optimal and the avoidance ranges is presented in this section. From the obtained values of optimal and avoidance intervals found by applying the Decision Path Search (DPS) algorithm, the interval comparison identified four possible cases, as shown in Figure 5.7. The first case, where the avoidance interval is located in the middle of the optimal intervals, indicates that the factor is considered an avoiding factor and the new operating range is kept as the original range. The second case is the opposite: in short, an optimal interval is located in the middle of an avoiding interval, the factor is considered optimal, and the new range is selected as the optimal interval. The third case is where the optimal range is shifted to the right compared with the avoidance range; thus, the chosen factor is optimal and its range is obtained from the optimal interval. The last case is the opposite of the third case, where the factor is to be considered an avoidance factor and its ranges are obtained from the avoidance interval.

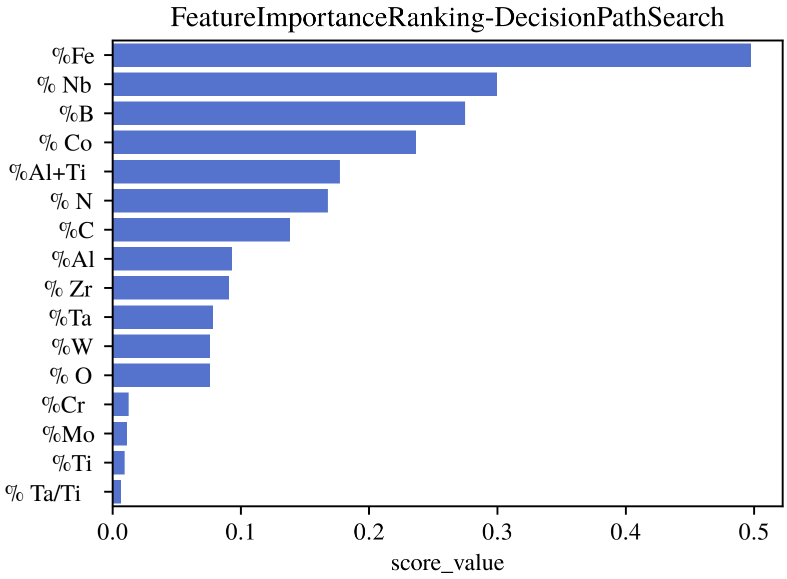


Figure 5.6: Feature importance ranking based on DPS for nickel-based superalloy dataset

The scoring for each factor, as shown in Figure 5.6, is calculated based on the non-overlapping area between the optimal and the avoidance ranges. As shown in Figure 5.7, cases 1 and 2, the factor has a minimal score because the two areas of both intervals overlap. On the opposite side, for cases 3 and 4, the score calculated is proportional to the non-overlapping area. The greater the non-overlap area, the larger the score's contribution.

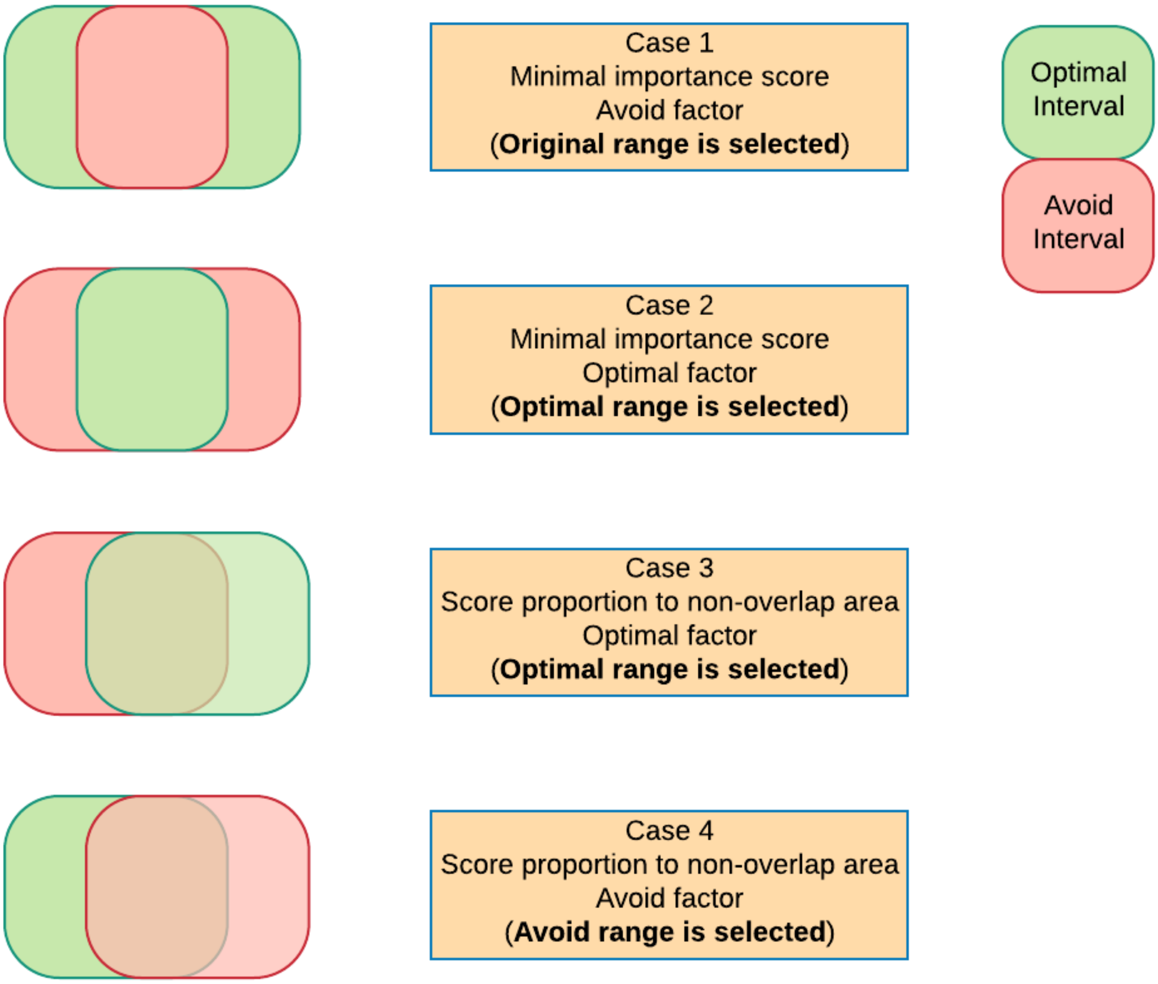

Case 1
Minimal importance score
Avoid factor
(**Original range is selected**)
Optimal Interval
Avoid Interval
Case 2
Minimal importance score
Optimal factor
(**Optimal range is selected**)
Case 3
Score proportion to non-overlap area
Optimal factor
(**Optimal range is selected**)
Case 4
Score proportion to non-overlap area
Avoid factor
(**Avoid range is selected**)

Figure 5.7: All possible cases for optimal and avoidance intervals

5.2.3 Results of a Foundry Case Study for a Nickel-based Superalloy

In general, the MRF model can be applied to the nickel dataset matrix. As discussed in chapter four, a predictive model was developed, after which the optimal decision path search was used to estimate the optimal operating limits. The DPS algorithm was then applied to understand the contribution of each factor to the shrinkage defects vector. Figure 5.8 illustrates the optimal and avoidance configuration for all variables; the optimum range is shown in black, while the avoidance range is shown in cyan. Both new ranges (on the left side of each factor) are plotted against the complete range for all variables (on the right side of each factor). Figure 5.9 represents the optimal and avoidance intervals for the nickel-based superalloy dataset. It also represents the effect of each factor on the response based on visualising the interval. Each factor will be categorised based on the cases shown in Figure 5.7. Here, the optimal range interval is marked in green while the avoidance range interval is marked in red. In addition, the no-effect and non-importance intervals are marked in light brown. If the factor is considered optimal, this means that increasing the range will lead to minimising the shrinkage defect; thus, increasing the range is important to producing a better final product. By contrast, if the factor is considered an avoidance factor, increasing the range will lead to increasing the shrinkage defect; in this case, therefore, increasing the range does not enhance the quality of the final product. The other cases are considered to have minimal importance, meaning that these factors have minimal impact on the process output quality. From Figure 5.9, it can be observed that the factors $\%Fe$, $\%Nb$, $\%Al+Ti$, and $\%C$ make a strong contribution to minimising the shrinkage defect. These factors represent case 3 in Figure 5.7, from which it can be observed that the optimal range will be selected because the green area is increased to further minimise defect. By contrast, factors $\%B$ and $\%Co$ showed an avoidance contribution to minimising the shrinkage defect. These factors represent case 4 in Figure 5.7. It can thus be observed that the avoidance range will be selected, because when the red area increases, the defect also increases. Other factors such as $\%Cr$, $\%Mo$, and $\%Ti$ have a very small effect on the shrinkage defect; these factors represent cases 1 and 2 in Figure 5.7.

In summary, four variables ($\%Fe$, $\%Nb$, $\%Al+Ti$, and $\%C$) showed a high optimal contribution to minimising the shrinkage defects. By contrast, $\%B$, $\%Co$, and $\%N$ show an undesirable (avoidance) contribution to the shrinkage defects. Working within the optimal limits (shown in green in Figure 5.9) will help to ensure that the output product has very little probability of being defective. This conclusion can be backed up by the test data, for which the conclusion holds although the algorithm was not trained on it. This further demonstrates that the proposed MRF algorithm generalises well for this type of datasets.

Note that the MRF algorithm is designed to discover correlations from the observational process data. Correlations are not always expected to lead to causation. These correlations are presented to process engineers who would then use their metallurgical experience to select some of the correlations for a confirmation trial. The analysis is considered as successful if it leads to process engineers choosing at least one or two factors for further monitoring. If none of the factors are chosen for a confirmation trial, it is still considered as a useful study as it would have helped eliminate chemistry parameters as potential causes so that further changes, such as design, can be investigated. However, the analysis of this dataset had led to a change in at least one chemical composition element for the foundry.

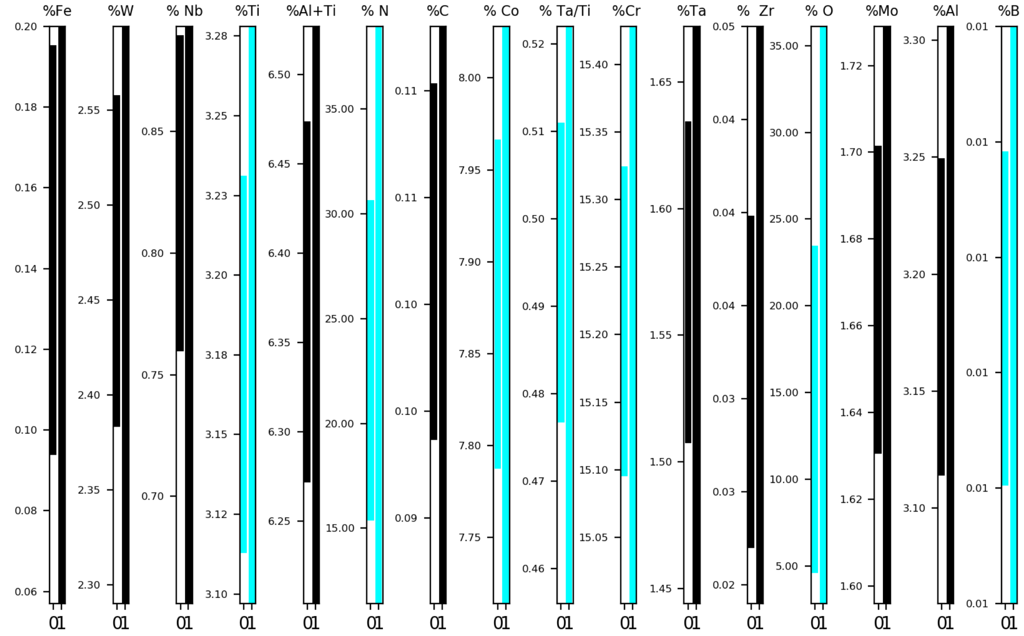


Figure 5.8: Tolerance limits of nickel-based superalloy dataset

The next step is to create a set of new examples within the obtained operating limit range in order to study the consequence of the process-controlled factors, as well as the new models generated using the Bootstrap by replacement method from within the optimal range for each factor. Finally, the odds ratio of the optimal range is compared with the odds ratio of the original range.

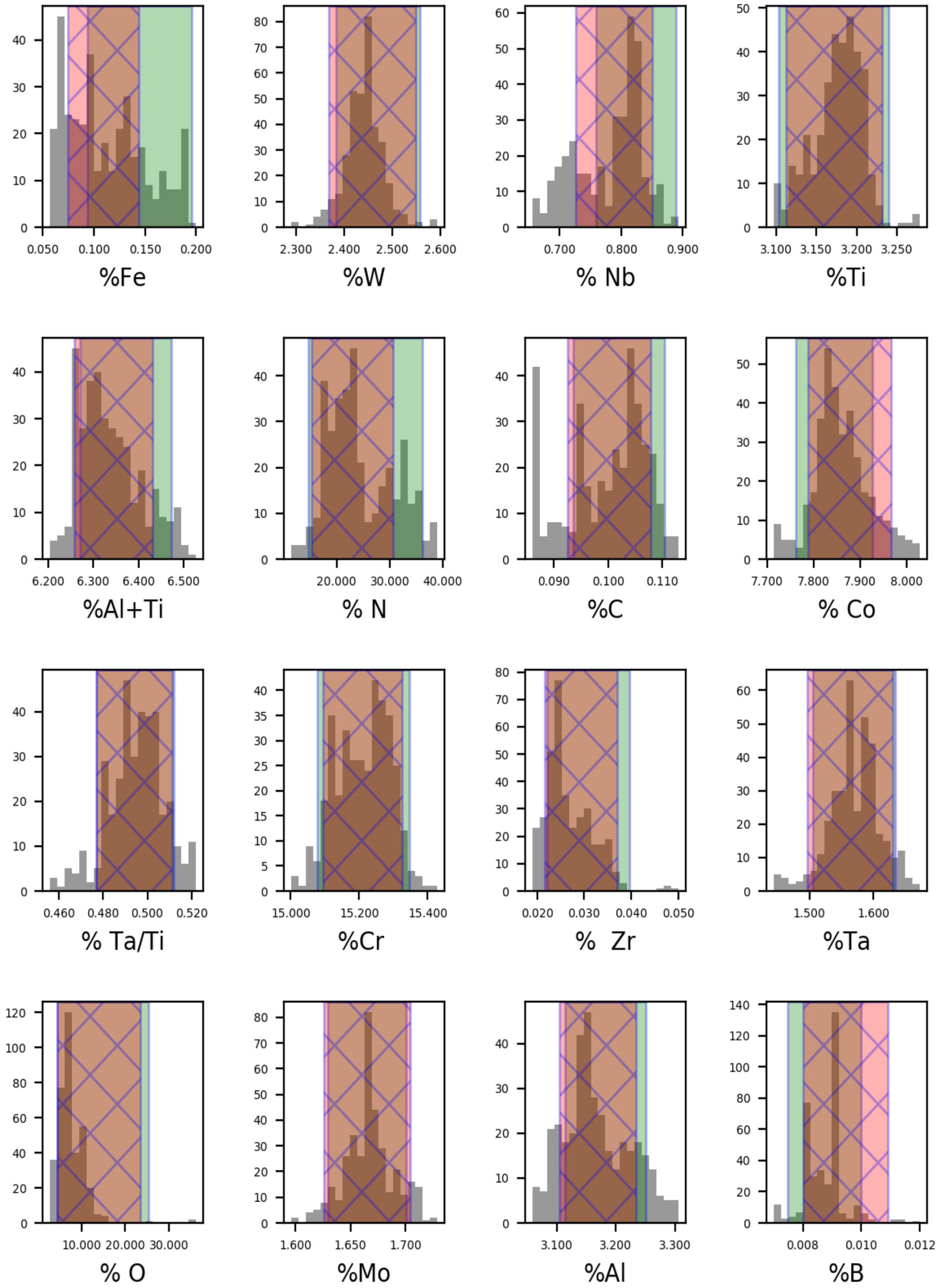


Figure 5.9: Optimal and avoidance intervals for nickel-based superalloy dataset

5.3 Odds Ratio

The odds ratio is considered to evaluate the behaviour of the discovered operating limits range by estimating the values of the corresponding response. After determining the response of the operating limits, this tool can be used to compare the performance of the process before and after applying operating limits. In other words, it enables estimating the probability of occurrence of the desired (optimal) and undesired (avoidance) response values for a confirmation trial plan and the original plan. The comparison of two proportions of occurrence, such as success and failure, can be carried out by calculating the probabilities of these proportions; another test method used is the likelihood ratio test [16]. The odds of success are defined as the ratio of the probability of success to the likelihood of failure [102, 103], and can be calculated as follows:

$$\Omega = \frac{\pi_s}{\pi_f} \tag{5.2}$$

Where:
Ω: The overall odds of success.
π_s: The probability of success (Odd of success).
π_f: The probability of failure (Odd of failure).

In terms of manufacturing defects, success represents the occurrence of desired response values, such as a lower percentage of defects in batches (optimal response values). Moreover, failure is represented by higher rates of defects or the appearance of undesired response values (avoidance response values). As a result, the odds ratio and probabilities of success (odds of success) and failure (odds of failure) will be replaced by the probability of an optimal result (π_p) and the likelihood of an undesired result (π_v) respectively. The odds ratio equation above can thus be re-written as follows:

$$\Omega = \frac{\pi_p}{\pi_v} \tag{5.3}$$

Where,
π_p : The probability of an optimal result, P($Y \leq Th_{op}$).
π_v: The probability of an avoidance result, P($Y > Th_{op}$).
Th_{op}: The optimal threshold.

Confirmation trial simulations were conducted to explore the consequences of applying the recommended process limit to the overall process output. The bootstrap (B) method was used to generate 1000 examples from the combination of the optimal operating limits of factors and then compared with the original range by estimating the odds ratios for the original factors and the odds ratio based on the recommended operating limits. Figures 5.10 and 5.11 represents the odds ratio and the response histogram values for the original range and bootstrapped operating limits, respectively. These two figures show the significant improvements in process output quality achieved by using the recommended operating range, which stems from the DPS algorithm.

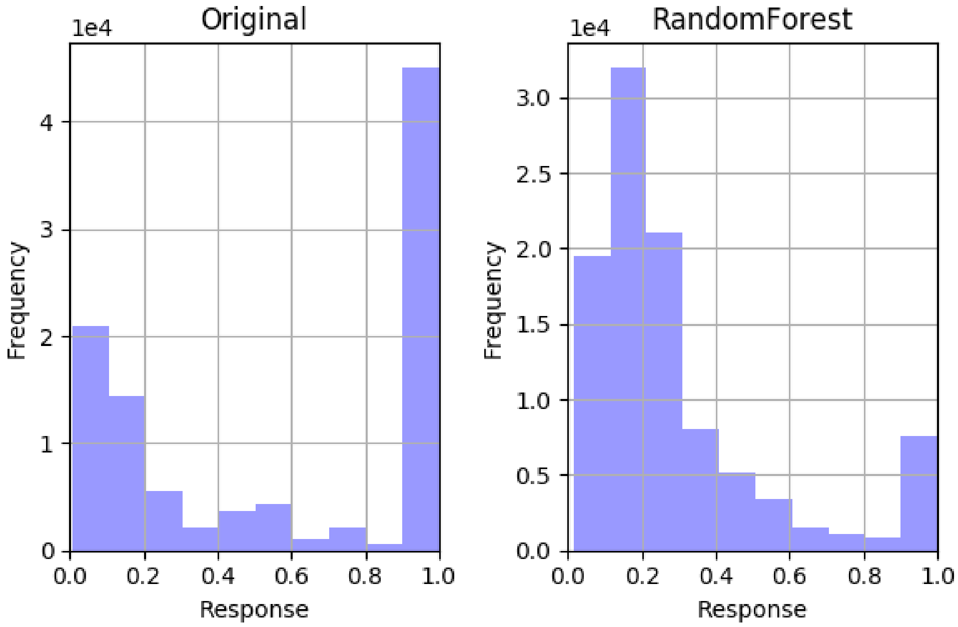


Figure 5.10: Response before and after applying the recommended process limits from MRF to the nickel-based superalloy dataset

Figure 5.12 presents a comparison of the odds ratio of interacted factors with the proposed optimal limit that determines how the interaction of the factor will affect the output response. It shows that when each of the three factors interacts with each other, this will yield a response probability value that might lead to success (if it is equal to or above one) or failure (if it is less than one). If the odds ratio of the three interacted factors shows success, this means that the factors have a strong correlation that will affect the output response. Each value in the figure represents a value of three interacted factors from the optimal limits, while the rest of the factors are based on the original limit. This process reveals the efficiency of the proposed optimal limit in influencing the response output.

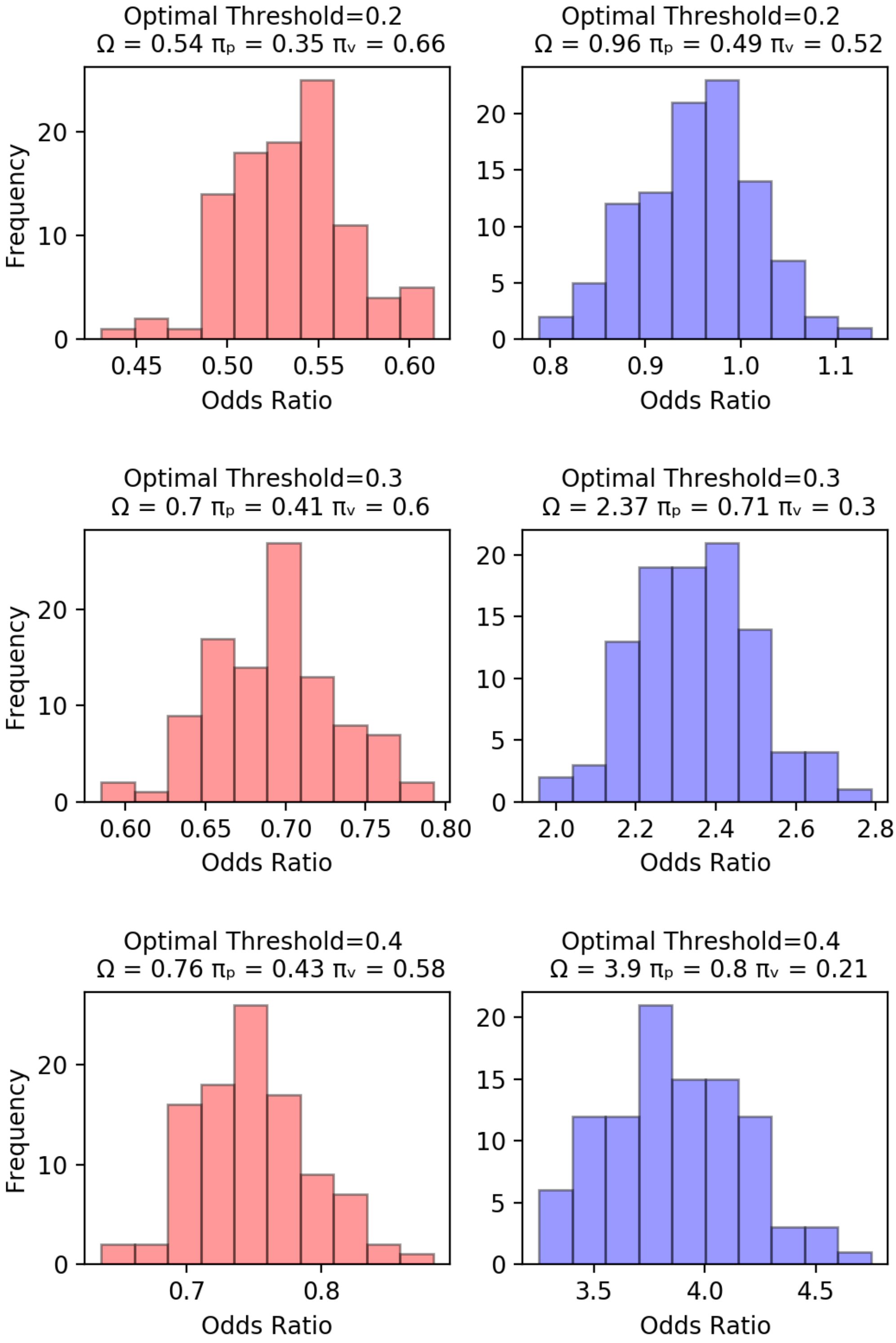


Figure 5.11: Odds ratio of original (red distributions) and optimal recommended process limits (blue distributions) of MRF applied to all factors of the nickel-based superalloy dataset

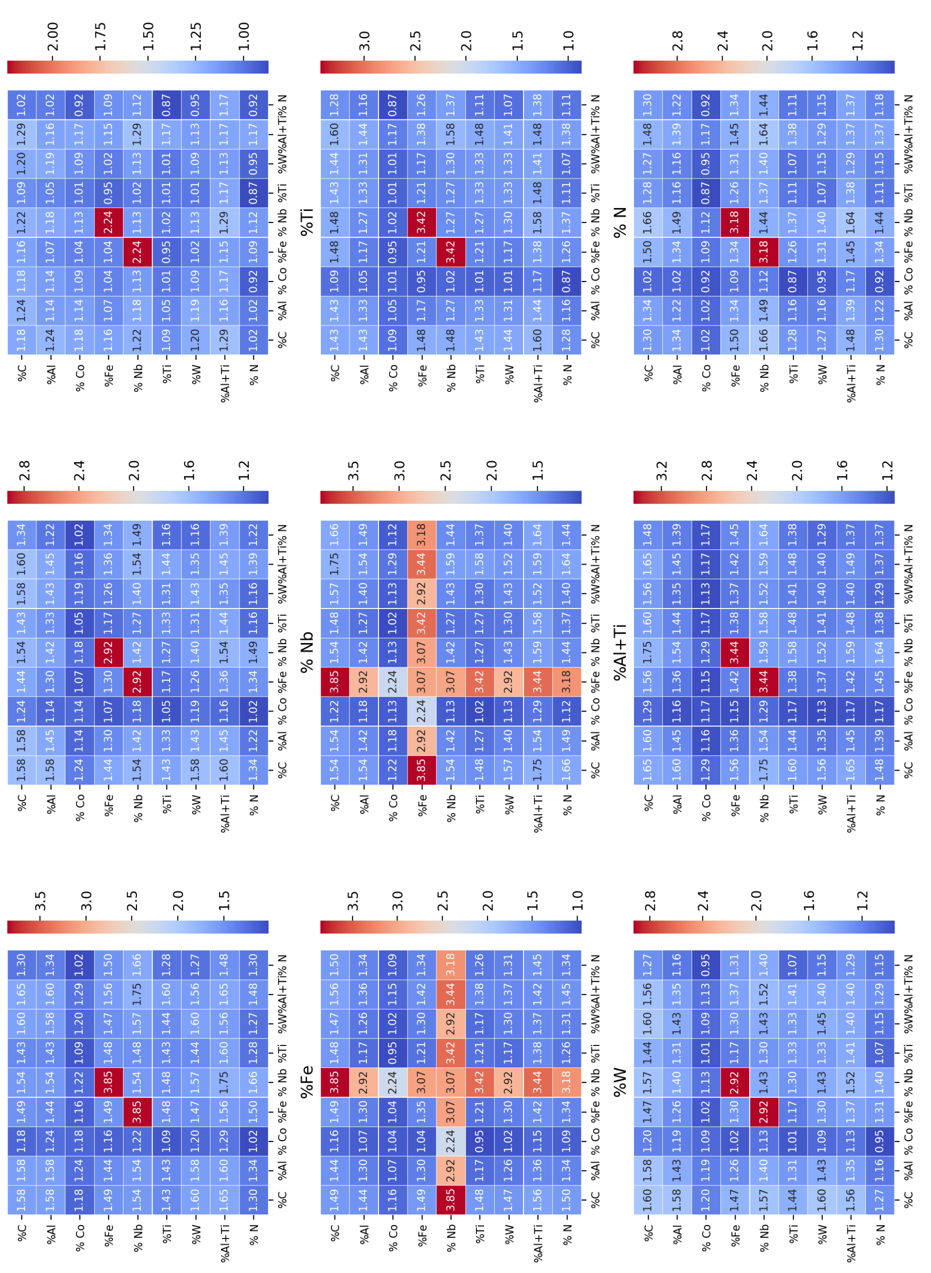


Figure 5.12: Odds ratio of interacted factors with proposed optimal limits from MRF (with 0.2 penalty threshold) applied to the nickel-based superalloy dataset

5.4 Discussion of Results

The nickel-based superalloy dataset used in chapters four and five to estimate the optimal limits is discussed here. The results are compared with the Penalty Matrix (PM) approach [13] and the QCA algorithm [8], as seen in Table 5.1. The optimum regions and avoidance ranges are determined by evaluating the number of observations across a quartile in the Penalty Matrix framework. In the QCA context, the spectrum obtained depends on the estimation projection of load scores. By contrast, in the current work, each factor is scored based on its contribution to the prediction of each tree. To determine which scores contribute to shrinking defects, optimal and avoidance intervals are plotted in Figure 5.8 to score the contribution of each factor to the defects in the process outputs. Because of the proposed algorithm's ability to process mixed data sources, it is now possible to contrast the categorical interaction variable with the corresponding continuous variables on the same feature ranking plots. Through this process, the analysis of the process factors and their optimum operating range provides more information and contributes to a better understanding of the process.

Table 5.1: Comparison of obtained ranges with Penalty Matrix [13], QCA [8] and MRF

Variables	Penalty Matrix Range	QCA Range	Predicted Range by MRF
$\%C$	0.095 - 0.113 (Optimal)	0.093 - 0.112 (Optimal)	0.092 - 0.110 (Optimal)
$\%Fe$	0.114 - 0.200 (Optimal)	0.095 - 0.200 (Optimal)	0.092 - 0.191 (Optimal)
$\%Al$	3.240 - 3.306 (Optimal)	3.145 - 3.306 (Optimal)	3.110 - 3.250 (Optimal)
$\%Nb$	0.770 - 0.827 (Optimal)	0.770 - 0.865 (Optimal)	0.755 - 0.885 (Optimal)
$\%W$	2.451 - 2.594 (Optimal)	2.413 - 2.594 (Optimal)	2.377 - 2.548 (Optimal)
$\%Zr$	0.026 - 0.050 (Optimal)	0.023 - 0.050 (Optimal)	0.020 - 0.040 (Optimal)
$\%Al + Ti$	6.299 - 6.527 (Optimal)	6.299 - 6.498 (Optimal)	6.270 - 6.470 (Optimal)
$\%Co$	7.840 - 8.028 (Avoidance)	7.847 - 8.018 (Avoidance)	7.781 - 7.976 (Avoidance)
$\%Ti$	-	3.154 - 3.278 (Avoidance)	3.113 - 3.232 (Avoidance)
$\%B$	-	-	0.008 - 0.011 (Avoidance)
$\%Ta/Ti$	-	-	0.478 - 0.512 (Avoidance)
$\%N$	-	-	15.244 - 30.870 (Avoidance)
$\%Cr$	-	-	15.093 - 15.324 (Avoidance)
$\%Mo$	-	-	1.631 - 1.702 (Optimal)
$\%Ta$	-	-	1.508 - 1.633 (Optimal)
$\%O$	-	-	4.714 - 25.399 (Optimal)

The simulation conducted in Figure 5.12 can be extended to study the effects of interaction between factors by bootstrapping three factors as (e.g., $\%Fe, \%C, \%Cr$) from optimal limits and bootstrapped values for factors taken from the original range. This procedure is repeated for all factors. An optimal value threshold for the penalty values of 0.2 is chosen to classify an optimal process response. The high values of the odds ratio resulting from each combination of factors indicate the existence of interaction among the factors. The value of the odds ratio in Figure 5.12 is shown in a cell for factor names shown in the corresponding row and column for the given table associated with the identical factor name. Figure 5.13 presents the odds ratio steps taken to construct the results.

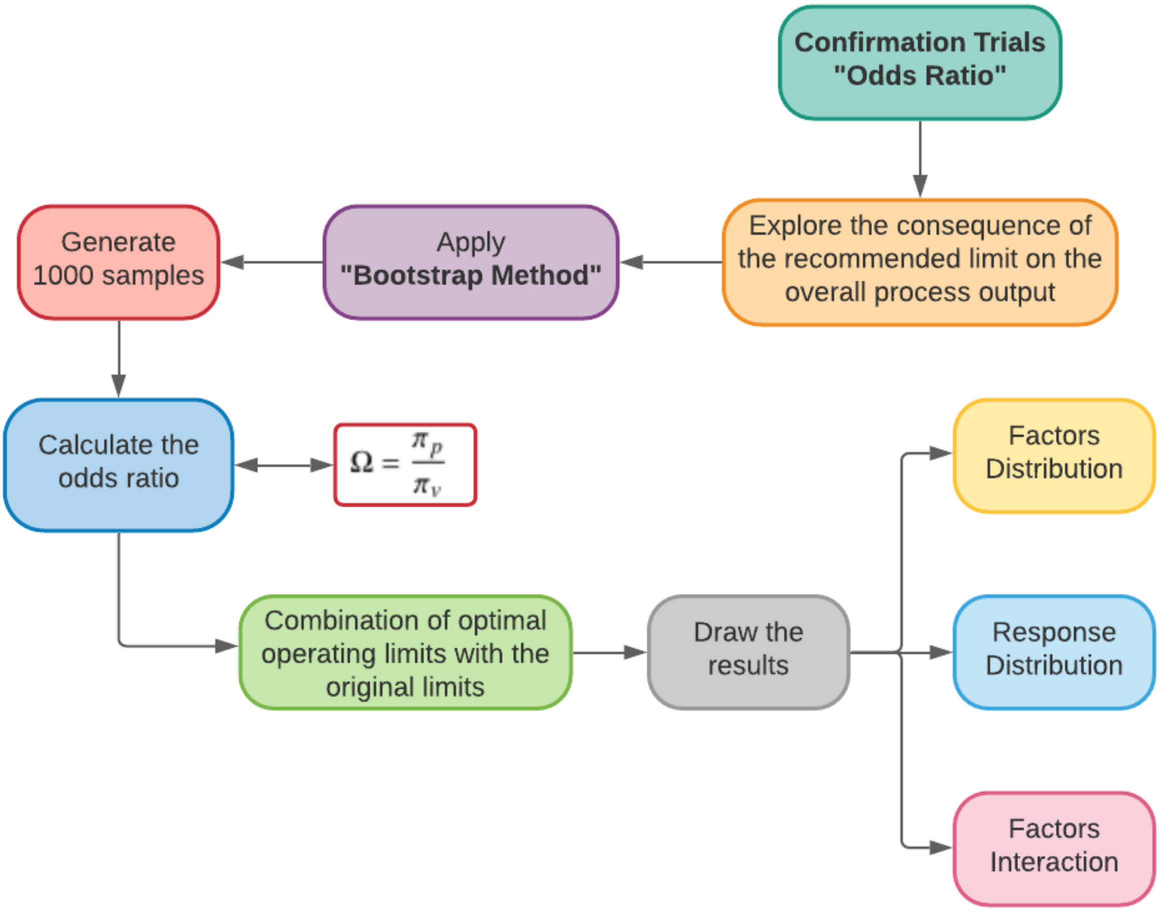

Confirmation Trials
"Odds Ratio"
Explore the consequence of the recommended limit on the overall process output
Apply
"Bootstrap Method"
Generate
1000 samples
Calculate the
odds ratio
$\Omega = \frac{\pi_p}{\pi_v}$
Combination of optimal operating limits with the original limits
Draw the
results
Factors
Distribution
Response
Distribution
Factors
Interaction

Figure 5.13: Confirmation trials steps

5.5 Conclusions

The novel Decision Path Search (DPS) algorithm approach introduced in this research is capable of scoring the critical process factors successfully, even from the confirmation trials conducted, as shown in Figure 5.11. The optimised range results obtained from the MRF algorithm match very closely with the ranges found by previous Swansea researchers [8, 11], as seen in Table 5.1 and Figure 5.14. It was shown that the MRF algorithm could find these ranges using a direct predictive approach. This simplifies the process and avoids the introduction of unnecessary techniques (e.g. uncertainty quantification) for which the accuracy cannot be easily judged. Moreover, this algorithm relies exclusively on the predictions to find the optimal ranges, which match very well with the published values, this further underscores the credibility of the model in the prediction task. In summary, the MRF algorithm achieves good results in obtaining the optimal range intervals for the nickel-based superalloy dataset.

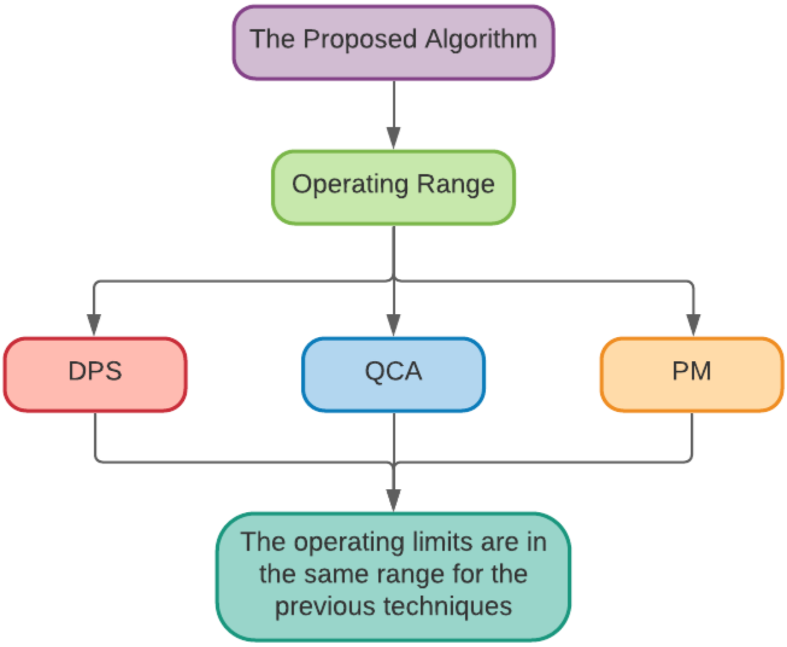

The Proposed Algorithm
Operating Range
DPS
QCA
PM
The operating limits are in the same range for the previous techniques

Figure 5.14: Summary of operating range comparisons with the previous work

Chapter 6

Verification on Published Datasets

6.1 Introduction

After reviewing the many datasets published in the literature, four of them were selected for verifying the proposed algorithm. Training runs, testing runs, and process optimal range analysis were conducted for the selected case studies. The results are reported in this chapter. Moreover, the MRF algorithm was compared with three other published RF models. The comparison highlights the main advantages and limitations of the proposed approach. The proposed algorithm is fully automated, with few user inputs required for internal parameter selection. The table of required input values and the schematic graphical representation of a sample obtained output are presented in Figure 6.1. The results and input values used are shown in Section 6.2; these follow the input and output structure shown in Figure 6.1. In Section 6.3, a comparison of the MRF algorithm with different published Random Forest models is conducted. The discussion of results, the comparison of the prediction accuracy, and recommendations for the process range obtained using the proposed algorithm are described in Section 6.4. A computational cost comparison is shown in Section 6.5. Finally, a comparison between the current and previous works [8] is presented in Section 6.6.

6.2 Case Studies

This section illustrates the results obtained from the proposed algorithm on the following datasets:

1. High Performance Concrete [104].

2. Energy Performance of Residential Buildings [105].

3. Combined Cycle Power Plant [106].

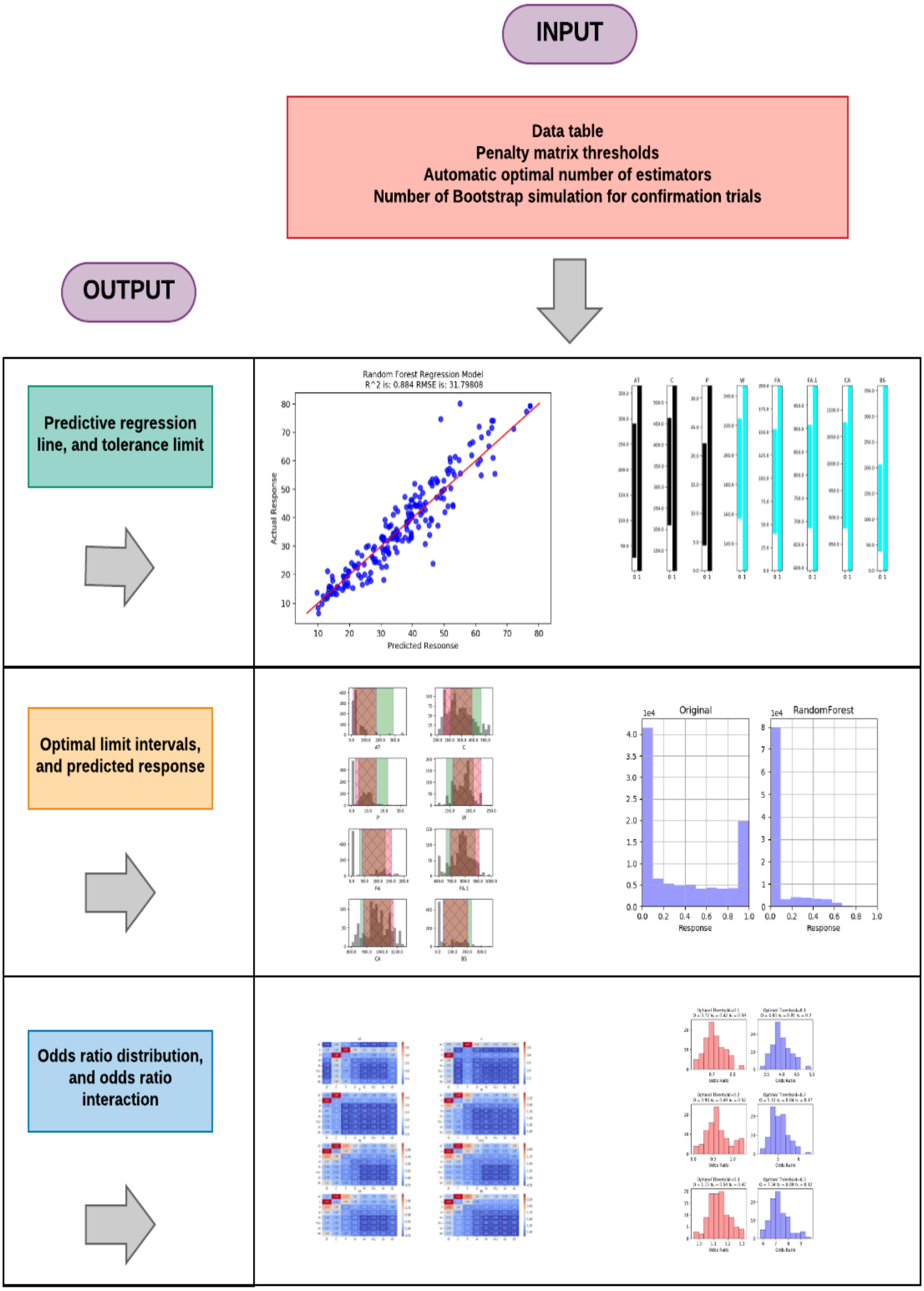

INPUT
Data table
Penalty matrix thresholds
Automatic optimal number of estimators
Number of Bootstrap simulation for confirmation trials
OUTPUT
Predictive regression line, and tolerance limit
Random Forest Regression Model
R^2 is: 0.884 RMSE is: 31.79808
Actual Response
Predicted Response
AT
C
P
W
FA
FA.1
CA
BS
Optimal limit intervals, and predicted response
Original
RandomForest
1e4
Response
Odds ratio distribution, and odds ratio interaction
Odds Ratio

Figure 6.1: The required input parameters and the resulting model output

6.2.1 High Performance Concrete Dataset

The compressive strength of concrete is a critical issue for the construction industry, as low compressive strength can lead to failures. The concrete's composition influences its compressive strength. Yeh [104] collected experimental data from 17 different sources for high-performance concrete (HPC) to model the power of the HPC based on a Neural Network model. The dataset contains 1030 observations consisting of one output response, Compressive Strength (CS), and eight factors: Cement (C), Fly-Ash (F), Blast-Furnace Slag (BS), Water (W), Superplasticiser (P), Coarse Aggregate (CA), Fine Aggregate (FA) and Age of Testing (AT_c). The dataset properties are listed in Table 6.1. In the current work, the MRF algorithm was applied to predict the Compressive Strength (CS) and estimate the optimal limits of all factors. The influence of using optimal and original boundaries on response values is quantified based on odds ratio assessments. Simulation parameters used for the MRF algorithm are listed in Table 6.2. The results of the prediction and optimal process range analysis are depicted in Figures (6.2 - 6.9).

Table 6.1: High performance concrete dataset properties

Variable	Minimum	Maximum	Mean	Data Type	Variable Type
$CS\ (Mpa)$	2.33	82.60	35.82	Quantitative	Response
$C\ (Kg/m^3)$	102.00	540.00	281.17	Quantitative	Factor
$BS\ (Kg/m^3)$	0.00	359.40	73.90	Quantitative	Factor
$F\ (Kg/m^3)$	0.00	200.10	54.19	Quantitative	Factor
$W\ (Kg/m^3)$	121.75	247.00	181.57	Quantitative	Factor
$P\ (Kg/m^3)$	0.00	32.20	6.20	Quantitative	Factor
$CA\ (Kg/m^3)$	801.00	1145.00	972.92	Quantitative	Factor
$FA\ (Kg/m^3)$	594.00	992.60	773.58	Quantitative	Factor
$AT_c\ (days)$	1.00	365.00	45.66	Quantitative	Factor

Table 6.2: User inputs for prediction and confirmation for high performance concrete dataset

MRF		Confirmation Trials	
Type	HB	Bootstrap	1000
Th_{min}	20	Simulation No	100
Th_{max}	40	Total Run's	100000
No of Factors	8	Th_{op}	0.1, 0.2, 0.3

The test was performed on the high performance concrete dataset [104]. From Figure 6.7, the observations indicate that the factors AT_c, C, and P are considered optimal intervals, while the factors W, F, FA, and CA are considered avoidance intervals. The factor BS does not show a significant effect on the process output. The dataset has non-linear interacted factors and the proposed algorithm still performs well in predicting the response with an accuracy of 88.4%, as shown in Figure 6.3. In the current work, it was concluded that the optimal number of trees that produces the best performance and prediction accuracy is 180 (see Appendix B). An optimal value threshold for penalty values of 0.1 is chosen to classify an optimal process response. The process response based on the optimal limit indicates high performance regarding the raw data, leading to fewer defects and enhancing the total output performance, as shown in Figure 6.6. The odds distribution in Figure 6.8 shows improved performance on the proposed optimal limit rather than the raw limit data, even with the minimum threshold in place. This indicates that the proposed limit will perform well in terms of enhancing product quality. The odds ratio interacted plot, presented in Figure 6.9, depicts the efficiency of the process when the interacted factors are chosen from the optimal limit and the remaining factors are based on the original limit. It can be deduced that the high odds ratio value of 4.92 is based on the interaction of three factors (namely AT, P, and C). As a result, the process engineer must analyse these specific interactions further. The proposed algorithm also has a lower computational cost in comparison to other published works (see Appendix B and Section 6.5); moreover, it achieves improved prediction accuracy compared with other research work. A thorough comparison with different published studies is conducted in Section 6.4.

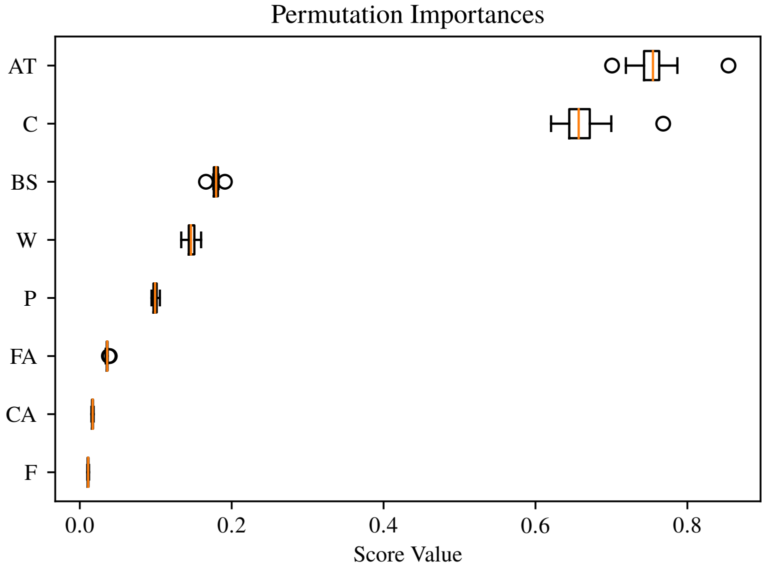


Figure 6.2: Permutation Importance ranking for high performance concrete dataset

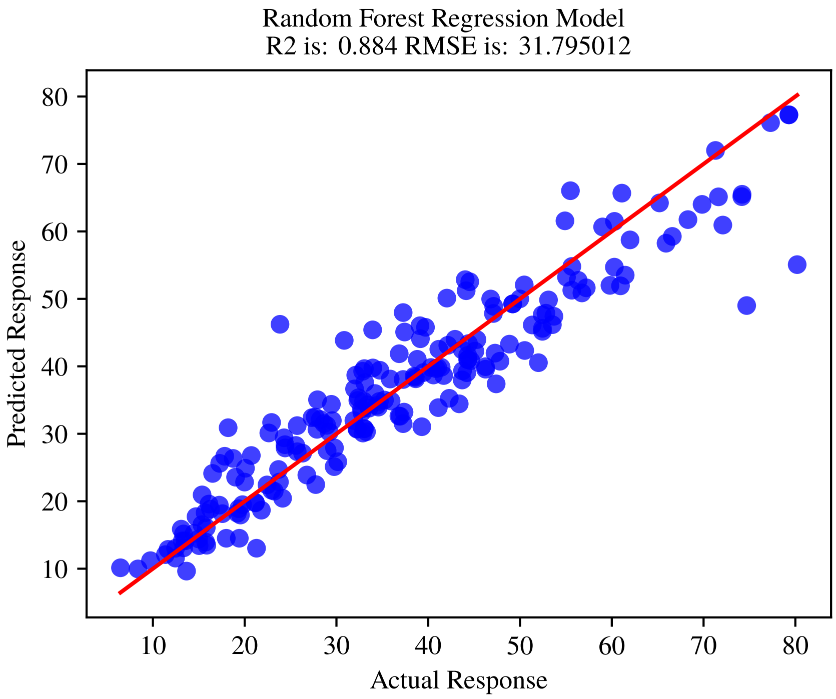


Figure 6.3: Testing on unseen data - Concrete dataset

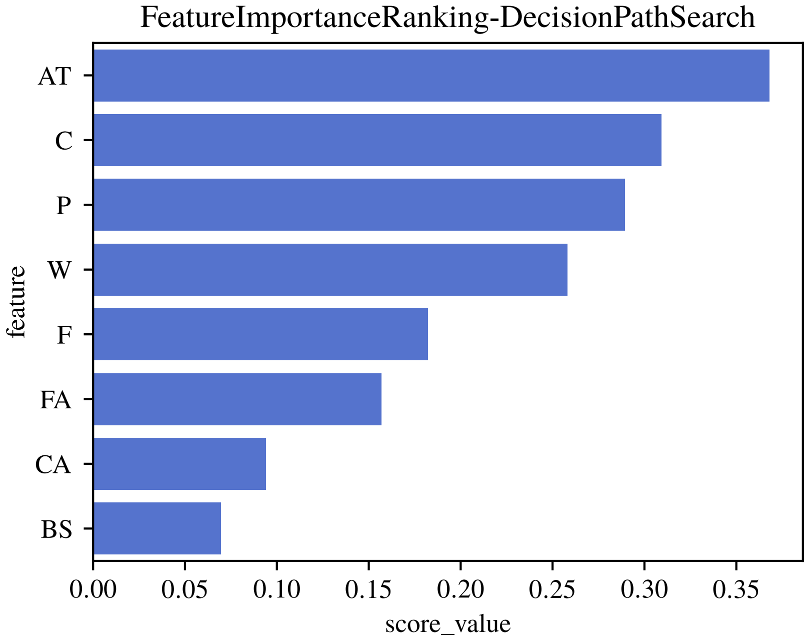


Figure 6.4: Feature importance ranking based on Decision Path Search for high performance concrete dataset

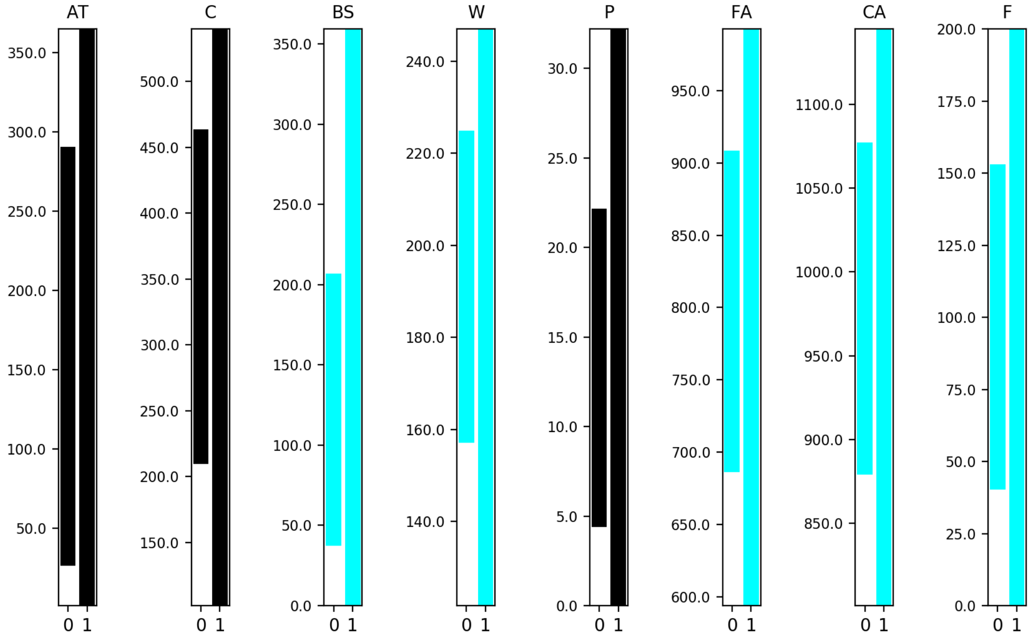


Figure 6.5: Tolerance limits of high performance concrete dataset

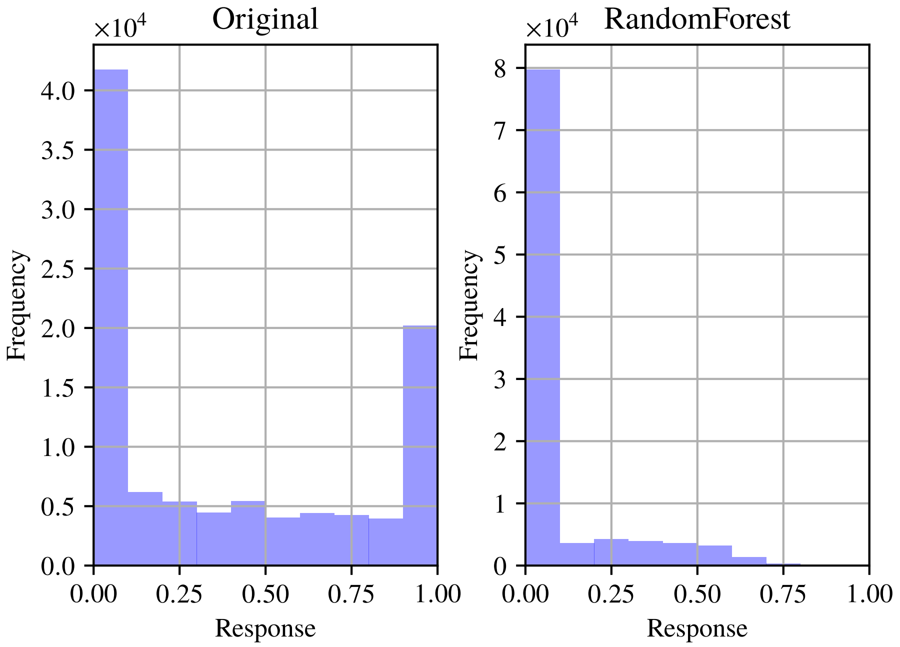


Figure 6.6: Response before and after applying the recommended process limits from MRF to the high performance concrete dataset

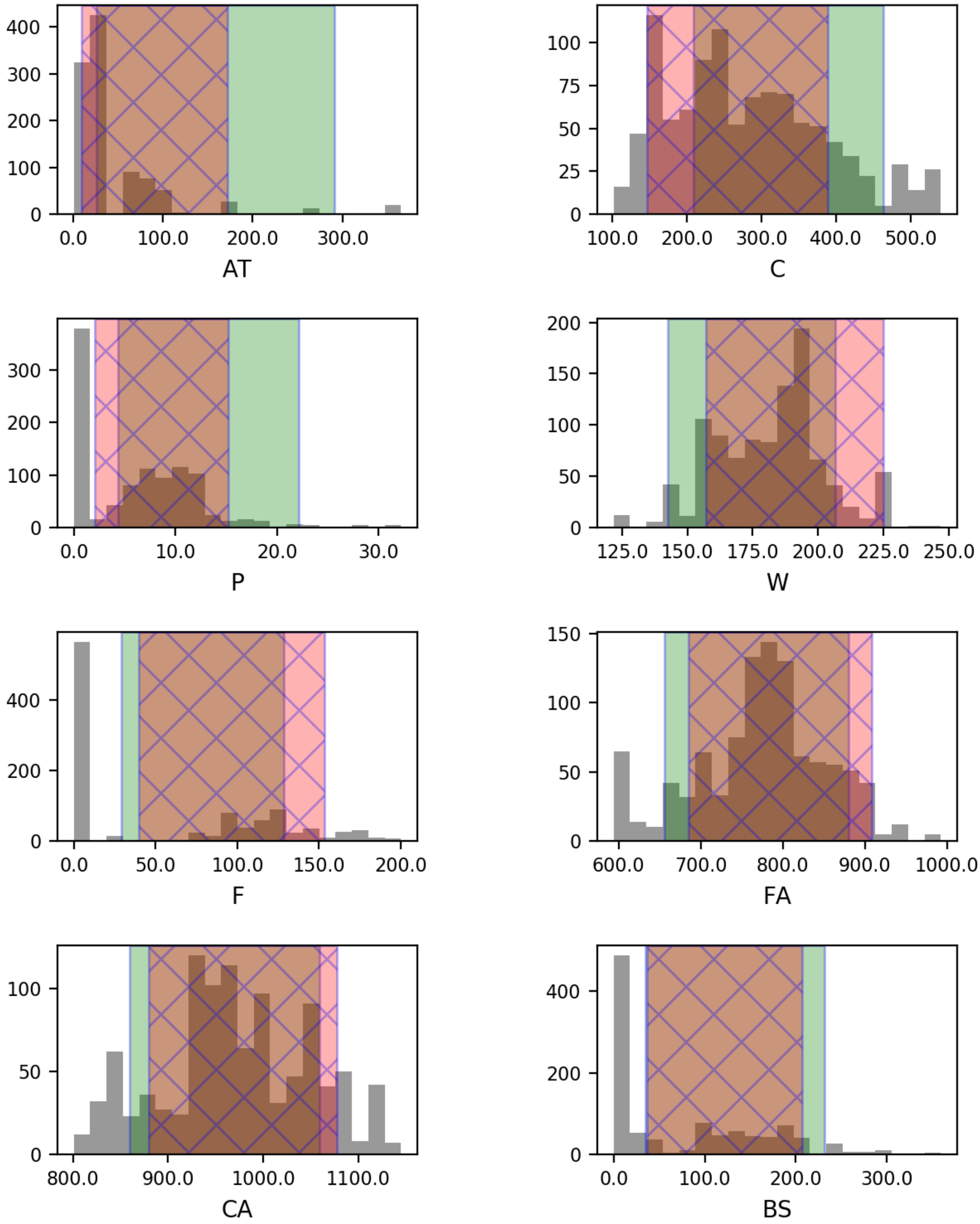


Figure 6.7: Optimal and avoidance intervals for high performance concrete dataset

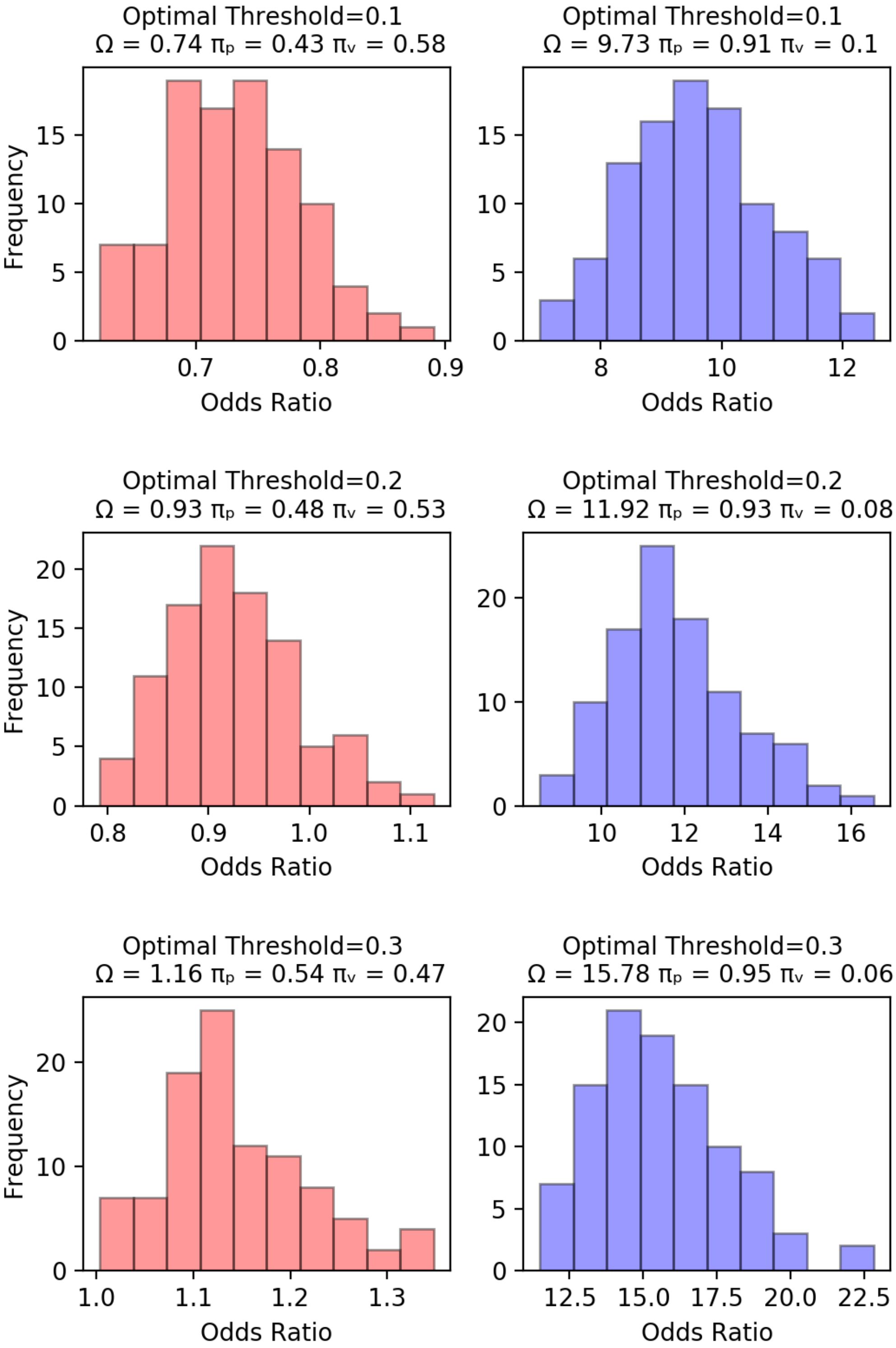
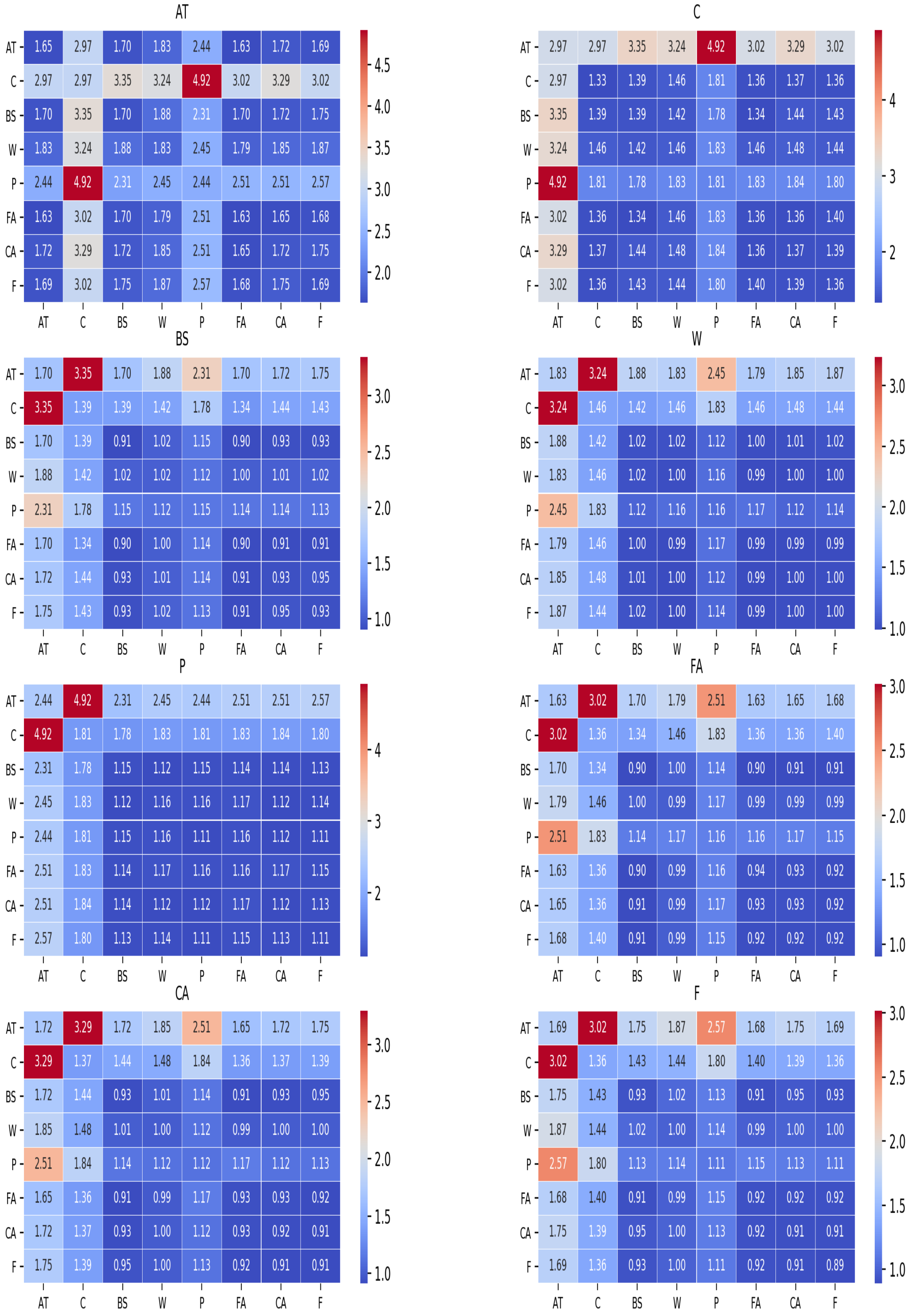


Figure 6.8: Odds ratio of original (red distributions) and optimal recommended process limits (blue distributions) of MRF applied to all factors of the high performance concrete dataset



AT

	AT	C	BS	W	P	FA	CA	F
AT	1.65	2.97	1.70	1.83	2.44	1.63	1.72	1.69
C	2.97	2.97	3.35	3.24	4.92	3.02	3.29	3.02
BS	1.70	3.35	1.70	1.88	2.31	1.70	1.72	1.75
W	1.83	3.24	1.88	1.83	2.45	1.79	1.85	1.87
P	2.44	4.92	2.31	2.45	2.44	2.51	2.51	2.57
FA	1.63	3.02	1.70	1.79	2.51	1.63	1.65	1.68
CA	1.72	3.29	1.72	1.85	2.51	1.65	1.72	1.75
F	1.69	3.02	1.75	1.87	2.57	1.68	1.75	1.69

C

	AT	C	BS	W	P	FA	CA	F
AT	2.97	2.97	3.35	3.24	4.92	3.02	3.29	3.02
C	2.97	1.33	1.39	1.46	1.81	1.36	1.37	1.36
BS	3.35	1.39	1.39	1.42	1.78	1.34	1.44	1.43
W	3.24	1.46	1.42	1.46	1.83	1.46	1.48	1.44
P	4.92	1.81	1.78	1.83	1.81	1.83	1.84	1.80
FA	3.02	1.36	1.34	1.46	1.83	1.36	1.36	1.40
CA	3.29	1.37	1.44	1.48	1.84	1.36	1.37	1.39
F	3.02	1.36	1.43	1.44	1.80	1.40	1.39	1.36

BS

	AT	C	BS	W	P	FA	CA	F
AT	1.70	3.35	1.70	1.88	2.31	1.70	1.72	1.75
C	3.35	1.39	1.39	1.42	1.78	1.34	1.44	1.43
BS	1.70	1.39	0.91	1.02	1.15	0.90	0.93	0.93
W	1.88	1.42	1.02	1.02	1.12	1.00	1.01	1.02
P	2.31	1.78	1.15	1.12	1.15	1.14	1.14	1.13
FA	1.70	1.34	0.90	1.00	1.14	0.90	0.91	0.91
CA	1.72	1.44	0.93	1.01	1.14	0.91	0.93	0.95
F	1.75	1.43	0.93	1.02	1.13	0.91	0.95	0.93

W

	AT	C	BS	W	P	FA	CA	F
AT	1.83	3.24	1.88	1.83	2.45	1.79	1.85	1.87
C	3.24	1.46	1.42	1.46	1.83	1.46	1.48	1.44
BS	1.88	1.42	1.02	1.02	1.12	1.00	1.01	1.02
W	1.83	1.46	1.02	1.00	1.16	0.99	1.00	1.00
P	2.45	1.83	1.12	1.16	1.16	1.17	1.12	1.14
FA	1.79	1.46	1.00	0.99	1.17	0.99	0.99	0.99
CA	1.85	1.48	1.01	1.00	1.12	0.99	1.00	1.00
F	1.87	1.44	1.02	1.00	1.14	0.99	1.00	1.00

P

	AT	C	BS	W	P	FA	CA	F
AT	2.44	4.92	2.31	2.45	2.44	2.51	2.51	2.57
C	4.92	1.81	1.78	1.83	1.81	1.83	1.84	1.80
BS	2.31	1.78	1.15	1.12	1.15	1.14	1.14	1.13
W	2.45	1.83	1.12	1.16	1.16	1.17	1.12	1.14
P	2.44	1.81	1.15	1.16	1.11	1.16	1.12	1.11
FA	2.51	1.83	1.14	1.17	1.16	1.16	1.17	1.15
CA	2.51	1.84	1.14	1.12	1.12	1.17	1.12	1.13
F	2.57	1.80	1.13	1.14	1.11	1.15	1.13	1.11

FA

	AT	C	BS	W	P	FA	CA	F
AT	1.63	3.02	1.70	1.79	2.51	1.63	1.65	1.68
C	3.02	1.36	1.34	1.46	1.83	1.36	1.36	1.40
BS	1.70	1.34	0.90	1.00	1.14	0.90	0.91	0.91
W	1.79	1.46	1.00	0.99	1.17	0.99	0.99	0.99
P	2.51	1.83	1.14	1.17	1.16	1.16	1.17	1.15
FA	1.63	1.36	0.90	0.99	1.16	0.94	0.93	0.92
CA	1.65	1.36	0.91	0.99	1.17	0.93	0.93	0.92
F	1.68	1.40	0.91	0.99	1.15	0.92	0.92	0.92

CA

	AT	C	BS	W	P	FA	CA	F
AT	1.72	3.29	1.72	1.85	2.51	1.65	1.72	1.75
C	3.29	1.37	1.44	1.48	1.84	1.36	1.37	1.39
BS	1.72	1.44	0.93	1.01	1.14	0.91	0.93	0.95
W	1.85	1.48	1.01	1.00	1.12	0.99	1.00	1.00
P	2.51	1.84	1.14	1.12	1.12	1.17	1.12	1.13
FA	1.65	1.36	0.91	0.99	1.17	0.93	0.93	0.92
CA	1.72	1.37	0.93	1.00	1.12	0.93	0.92	0.91
F	1.75	1.39	0.95	1.00	1.13	0.92	0.91	0.91

F

	AT	C	BS	W	P	FA	CA	F
AT	1.69	3.02	1.75	1.87	2.57	1.68	1.75	1.69
C	3.02	1.36	1.43	1.44	1.80	1.40	1.39	1.36
BS	1.75	1.43	0.93	1.02	1.13	0.91	0.95	0.93
W	1.87	1.44	1.02	1.00	1.14	0.99	1.00	1.00
P	2.57	1.80	1.13	1.14	1.11	1.15	1.13	1.11
FA	1.68	1.40	0.91	0.99	1.15	0.92	0.92	0.92
CA	1.75	1.39	0.95	1.00	1.13	0.92	0.91	0.91
F	1.69	1.36	0.93	1.00	1.11	0.92	0.91	0.89

Figure 6.9: Odds ratio of interacted factors with proposed optimal limits from MRF (with 0.1 penalty threshold) applied to the high performance concrete dataset

6.2.2 Energy Performance of Residential Buildings Dataset

The Ecotect energy simulation tool was used to simulate 12 building forms in Athens, Greece by Tsanas and Xifara [105]. All buildings have the same volume and material but different surface areas. The dataset contained 768 observations and consisted of two output responses, Heating Load (HL) and Cooling Load (CL), and eight factors: Relative compactness (Rc), surface Area (As), wall Area (Aw), roof Area (Ar), glazing Area (Ag), glazing Area distribution (Ad), overall Height (H), and Orientation (O). The dataset properties are listed in Table 6.3.

The authors used the traditional Random Forest method to arrive at the importance level for each factor to determine causality. The importance analysis obtained by studying the optimal search and prediction score is presented in Figure 6.12. The objective of this research is to ensure that all input factors always remain in the optimal range. If undesired variation in response values continues, the suggestion is to revisit the cause and effect, as shown in Figure 6.14, and identify the most important factors to monitor.

The MRF algorithm was used to identify related factors and estimate the tolerance limits of the quantitative and categorical factors. The odds ratio of correlated factors predicted for every single element and the interacted factors is based on the procedure in Section 5.3, keeping in mind that the heating load (HL) is the response with all the other factors. The simulation parameters used for the MRF algorithm are shown in Table 6.4. The results of prediction analysis and optimal process range analysis are depicted in Figures (6.10 - 6.17).

Table 6.3: Energy performance of residential buildings dataset properties

Variable	Range/Categories	Data Type	Variable Type
HL	6.01 - 43.10	Quantitative	Response
CL	10.90 - 48.03	Quantitative	Response
Rc	0.62 - 0.98	Quantitative	Factor
As	514.50 - 808.50	Quantitative	Factor
Aw	245.00 - 416.50	Quantitative	Factor
Ar	110.25 - 220.50	Quantitative	Factor
Ag	0 - 0.40	Quantitative	Factor
Ad	0 - 5.00	Quantitative	Factor
H	3.5 and 7	Categorical	Factor
O	2,3,4 and 5	Categorical	Factor

Table 6.4: User inputs for prediction and confirmation for energy performance of residential buildings dataset

MRF		Confirmation Trials	
Type	LB	Bootstrap	1000
Th_{min}	15	Simulation No	100
Th_{max}	35	Total Run's	100000
No of Factors	8	Th_{op}	0.1

The test was performed on the energy performance for residential buildings dataset [105]. From Figure 6.14, the observations indicate that the factors Ar, As, and H-3.5 are considered optimal intervals, while the factors Rc, Aw, Ag, Ad, and H-7 are considered avoidance intervals. The factor O does not show an effect on the process output and is thus considered a non-important factor. Although the data used is a mixed dataset (i.e. contains both regression and classification factors) , the proposed algorithm achieves excellent performance in predicting the response, with an accuracy of 99.6%, as shown in Figure 6.11. In the current work, it was concluded that the optimal number of trees to ensure the best performance and prediction accuracy was 300 (see Appendix B). An optimal value threshold for penalty values of 0.1 is chosen to classify an optimal process response. The process response based on the optimal limit indicates high performance regarding the raw data, leading to fewer defects and enhancing the total output performance as shown in Figure 6.15. The odds distribution in Figure 6.16 shows improved performance on the proposed optimal limit rather than the raw range data, even with the limited threshold value. This indicates that the proposed limit will perform well in terms of enhancing product quality. The odds ratio-interacted plot, shown in Figure 6.17, depicts the efficiency of the process when the interacted factors are chosen from the optimal limit and the remaining factors are based on the original limit. It can therefore be concluded that a high odds ratio value of 1.63 is based on the interaction of three factors (namely Ar, As, and H-7). As a result, the process engineer must analyse and modify these interactions further. The proposed algorithm has a lower computational cost in comparison to other published works (see Appendix B and Section 6.5); moreover, it also achieves improved prediction accuracy compared with other research work. A thorough comparison with different published studies is discussed in Sections 6.4 and 6.5.

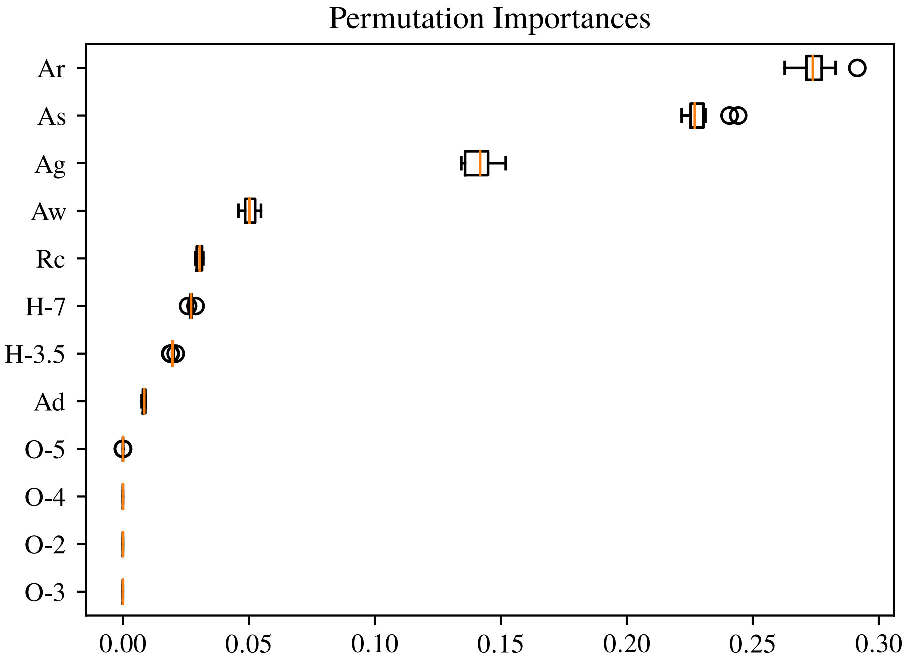


Figure 6.10: Permutation Importance ranking for energy performance of residential buildings dataset

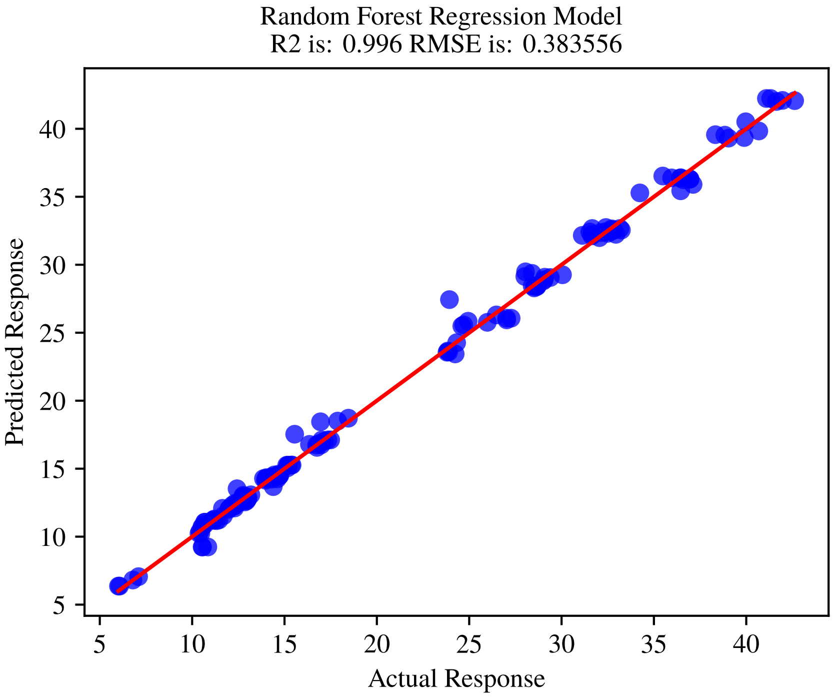


Figure 6.11: Testing on unseen data - Energy performance for residential buildings dataset

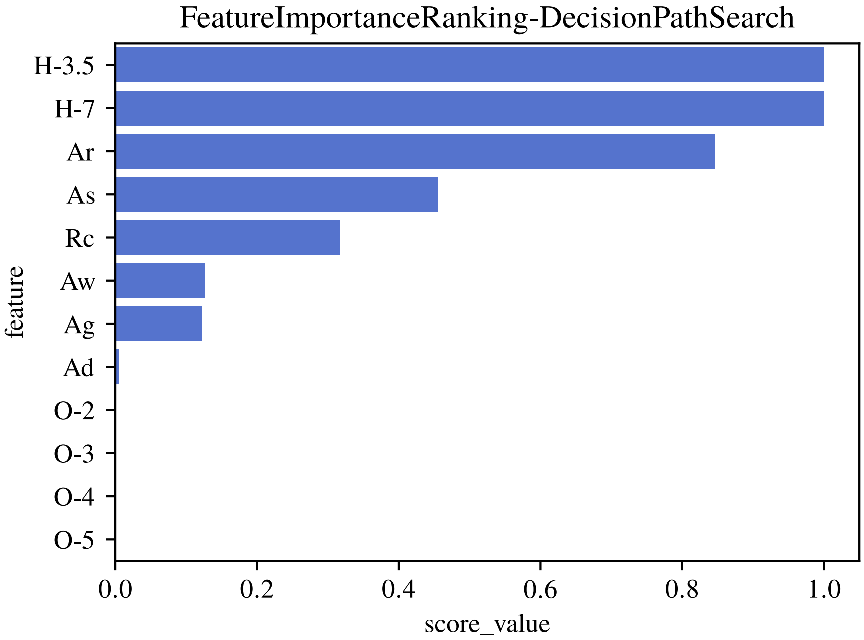


Figure 6.12: Feature importance ranking based on Decision Path Search for energy performance of residential buildings dataset

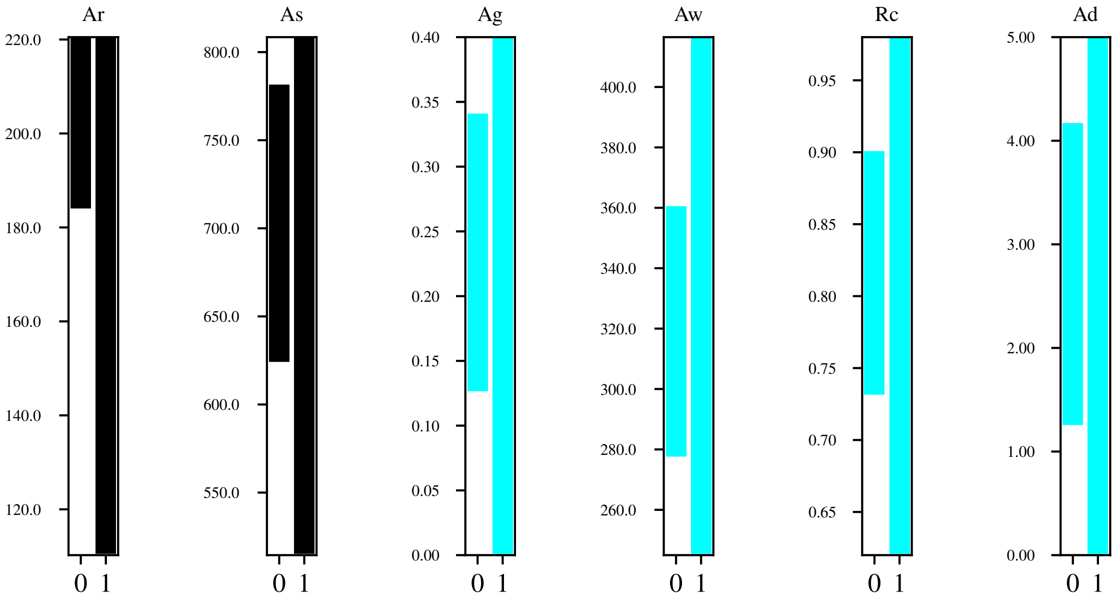


Figure 6.13: Tolerance limits of energy performance for residential buildings dataset

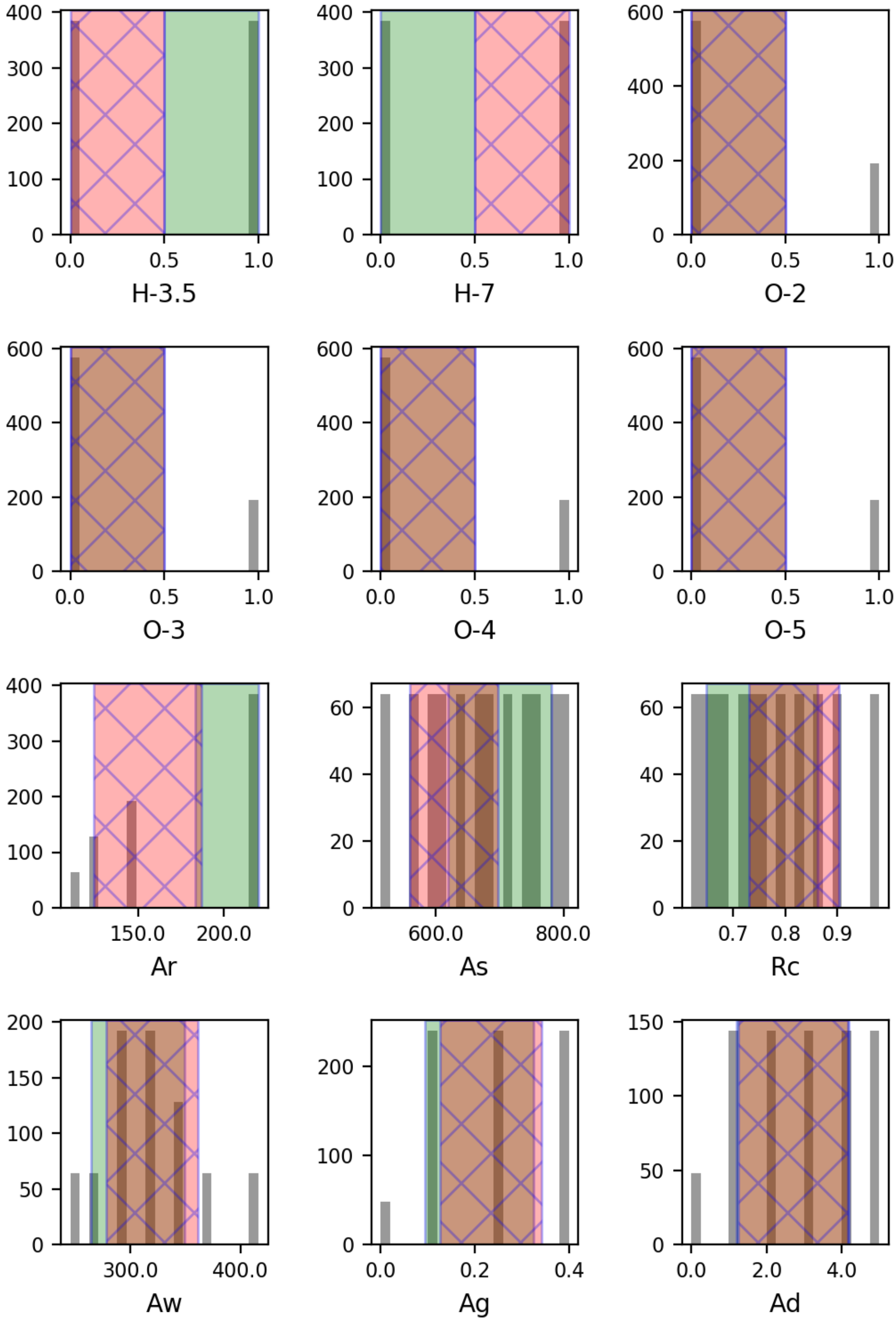


Figure 6.14: Optimal and avoidance intervals for energy performance of residential buildings dataset

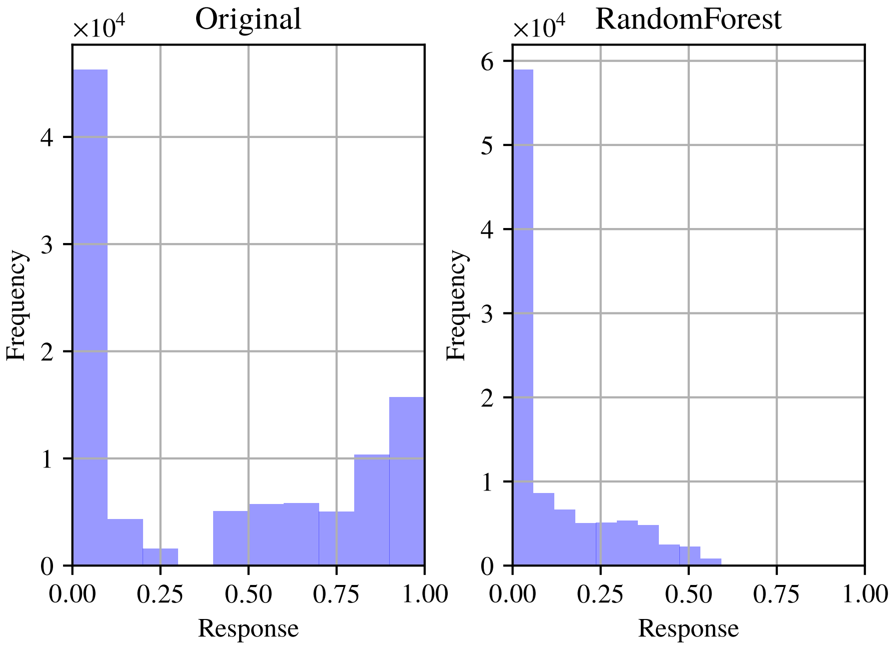


Figure 6.15: Response before and after applying the recommended process limits from MRF to the energy performance for residential buildings dataset

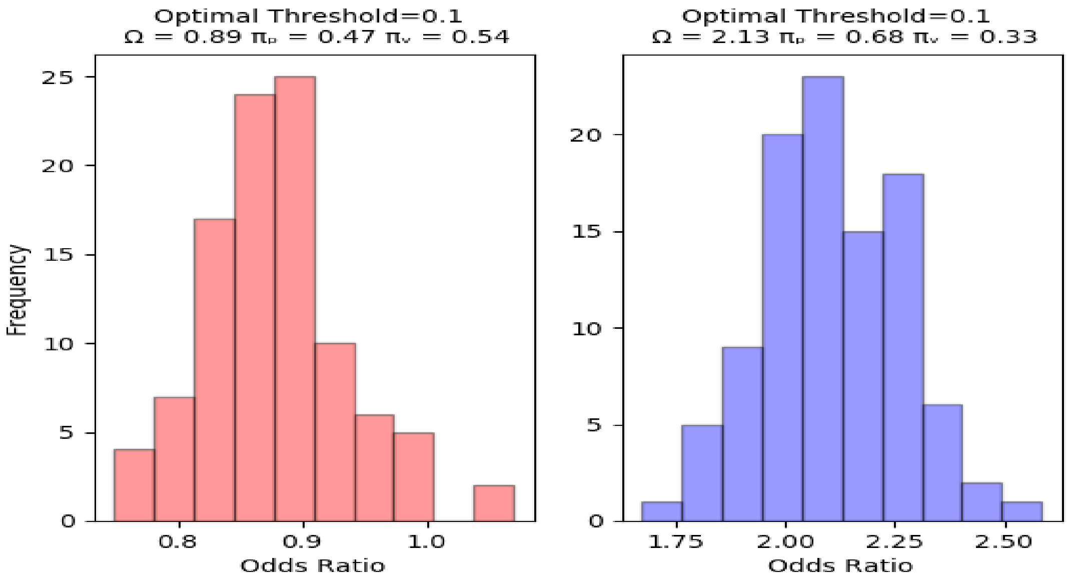


Figure 6.16: Odds ratio of original (red distributions) and optimal recommended process limits (blue distributions) of MRF applied to all factors of the energy performance of residential buildings dataset

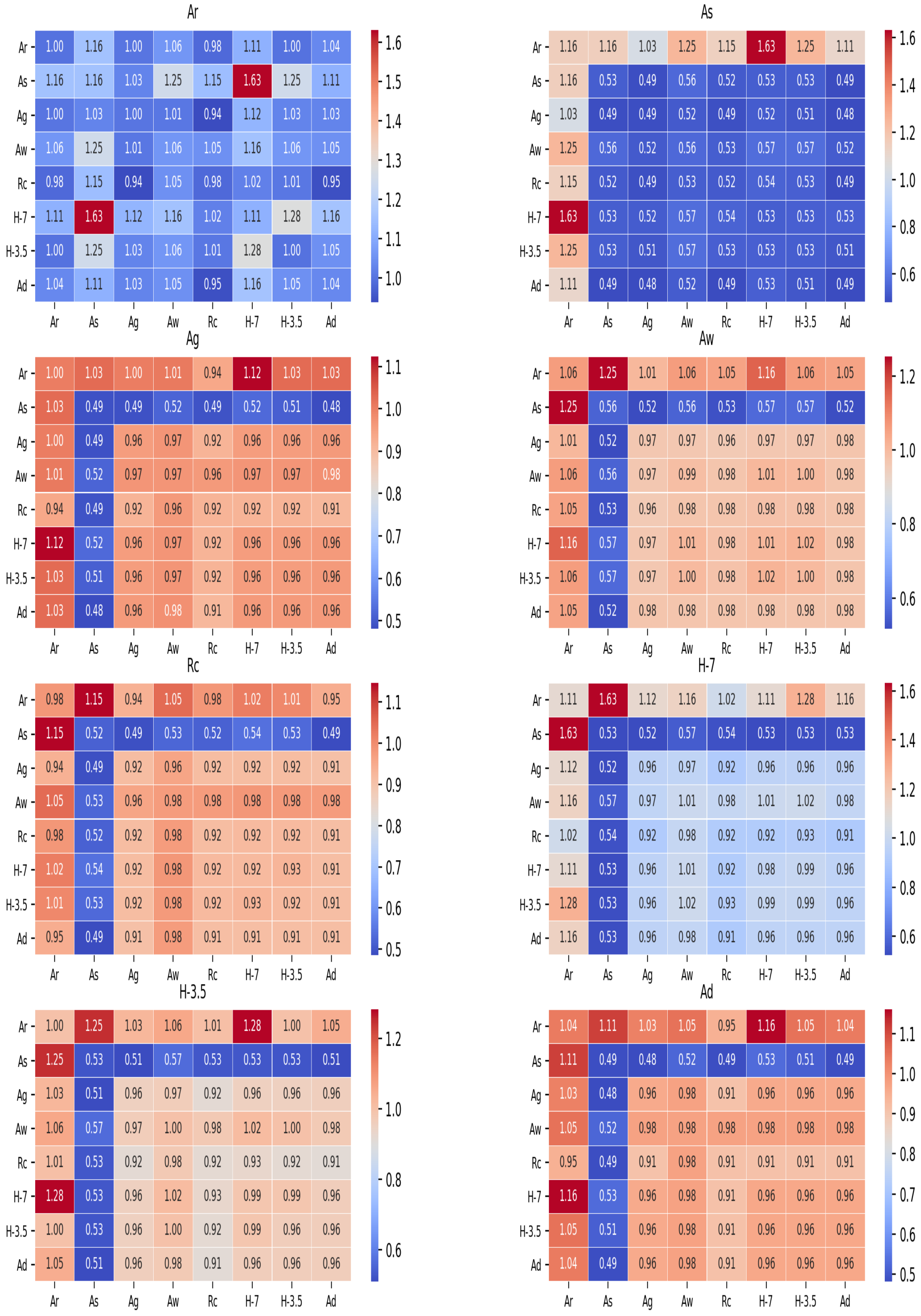


Figure 6.17: Odds ratio of interacted factors with proposed optimal limits from MRF applied to the energy performance of residential buildings dataset

6.2.3 Combined Cycle Power Plant Dataset

The combined cycle power plant (CCPP) is a power cycle that consists of gas turbines (GT), steam turbines (ST), and heat recovery steam generators (HRSG), where the electricity is generated by the GT and ST [106, 107]. The CCPP dataset used in the current study was designed with the following specifications: Nominal generation capacity of 480 MW, two 160 MW ABB 13E2 of GT, two dual-pressure HRSGs, and 160 MW ABB ST [106], as illustrated in Figure 6.18. The electrical power (PE) generated by both gas and steam turbines is influenced by four quantitative factors: Ambient Temperature (AT_p), Atmospheric Pressure (AP), Relative Humidity (RH), and the Exhaust Steam Pressure (or vacuum, V). The dataset properties are listed in Table 6.5. This dataset contains 9568 observations collected over six years (2006 till 2011).

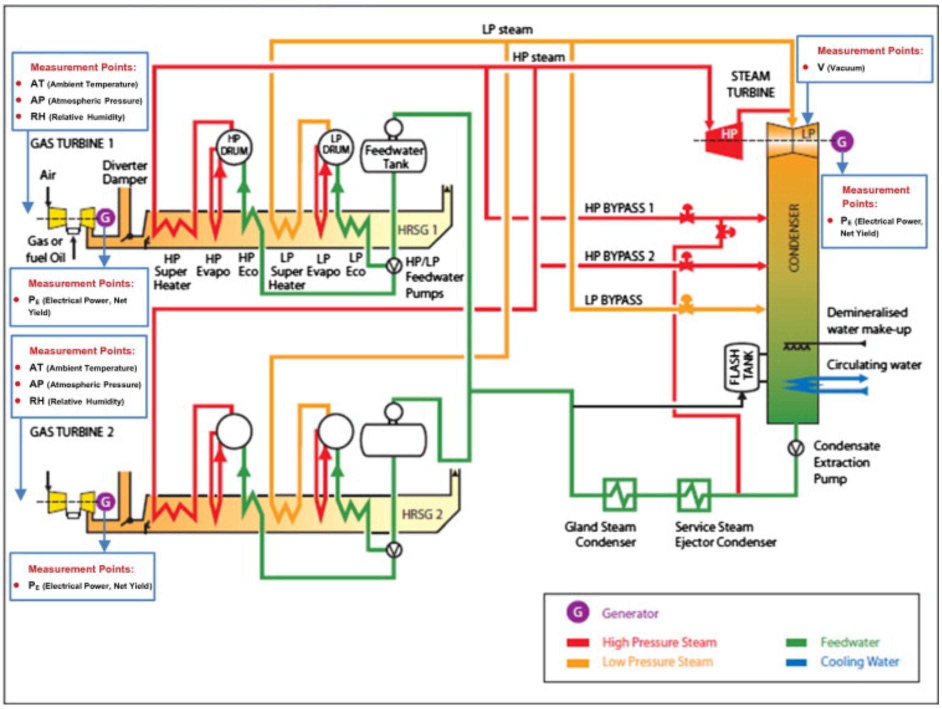


Figure 6.18: The combined cycle power plant layout [106]

The MRF algorithm was used to estimate the prediction performance and extract the optimal tolerance limits of the factors to maximise efficiency. The simulation parameters used for the MRF algorithm are listed in Table 6.6. The interaction of the odds ratio, as shown in Figure 6.26, is displayed for an optimal threshold value of 0.1, with the highest odds ratio of 6.52 for the MRF algorithm. The optimal combination of factors leading to a high odds ratio value was identified as the interaction between V, AP and AT for the MRF algorithm, as shown in Figure 6.26. The results of prediction analysis and optimal process range analysis are provided in Figures (6.19 - 6.26).

Table 6.5: Combined cycle power plant dataset properties

Variable	Minimum	Maximum	Mean	Data Type	Variable Type
$PE\ (MW)$	420.26	495.76	454.37	Quantitative	Response
AT_p (°C)	1.81	37.11	19.65	Quantitative	Factor
$V(cm.Hg)$	25.36	81.56	54.31	Quantitative	Factor
$AP\ (mbar)$	992.89	1033.30	1013.26	Quantitative	Factor
RH (%)	25.56	100.16	73.31	Quantitative	Factor

Table 6.6: User inputs for prediction and confirmation for combined cycle power plant dataset

MRF		Confirmation Trials	
Type	HB	Bootstrap	10000
Th_{min}	425	Simulation No	100
Th_{max}	470	Total Run's	1000000
No of Factors	4	Th_{op}	0.1

Testing was performed on the combined cycle power plant dataset [106]. From Figure 6.23, the observations indicate that the factors RH and AP are considered optimal intervals, while the factors AT_p and V are considered avoidance intervals. The proposed algorithm performs admirably in predicting the response, with an accuracy of 95.9%, as shown in Figure 6.21. In the current work, it was concluded that the optimal number of trees to ensure the best performance and prediction accuracy was 260 (see Appendix B). An optimal value threshold for penalty values of 0.1 was chosen to classify an optimal process response. The process response based on the optimal limit indicates high performance regarding the raw data, leading to fewer defects and enhancing the total output performance, as shown in Figure 6.24. The odds distribution in Figure 6.25 shows improved performance on the proposed optimal limit rather than the raw range data, even with the limited threshold value. This indicates that the proposed limit will perform well in terms of enhancing product quality. The odds ratio-interacted plot, as shown in Figure 6.26, depicts the efficiency of the process when the interacted factors are chosen from the optimal limit and the remaining factors are based on the original limit. The proposed algorithm has a lower computational cost in comparison to other published works (see Appendix B and Section 6.5); moreover, it also achieves improved prediction accuracy compared with other research work. A thorough comparison with different published studies is conducted in Sections 6.4 and 6.5.

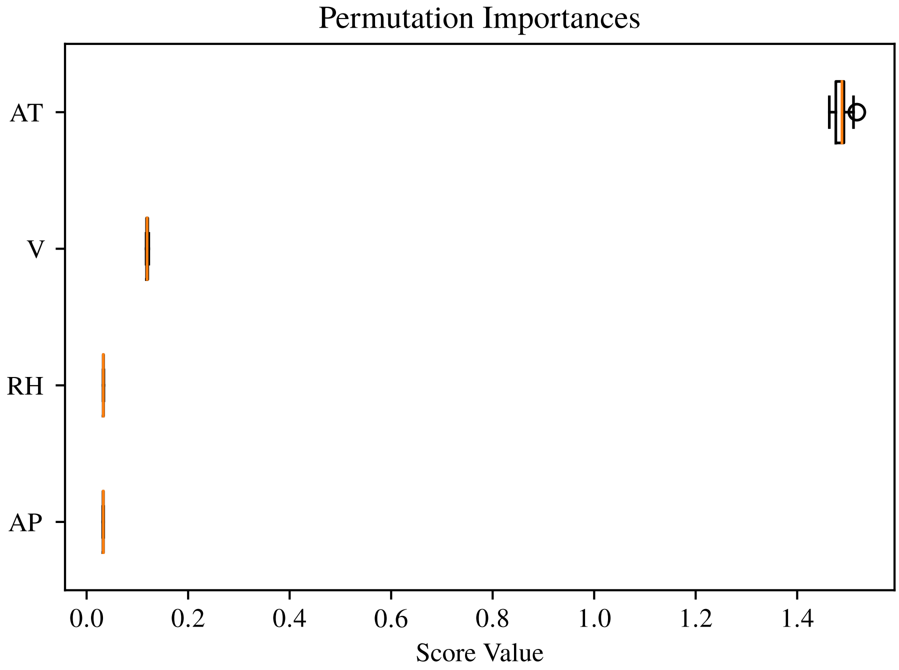


Figure 6.19: Permutation Importance ranking for combined cycle power plant dataset

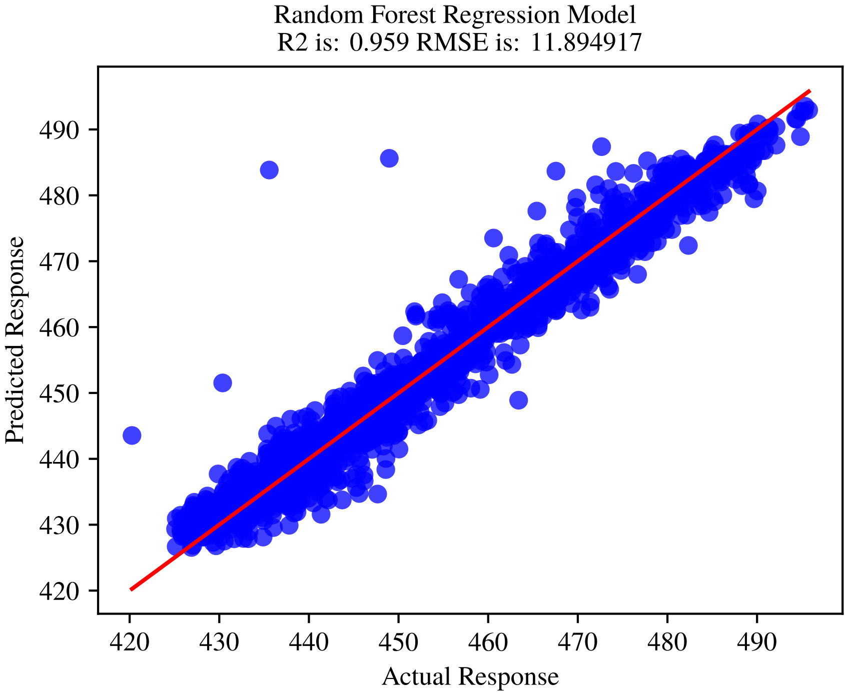


Figure 6.20: Testing on unseen data - Combined cycle power plant dataset

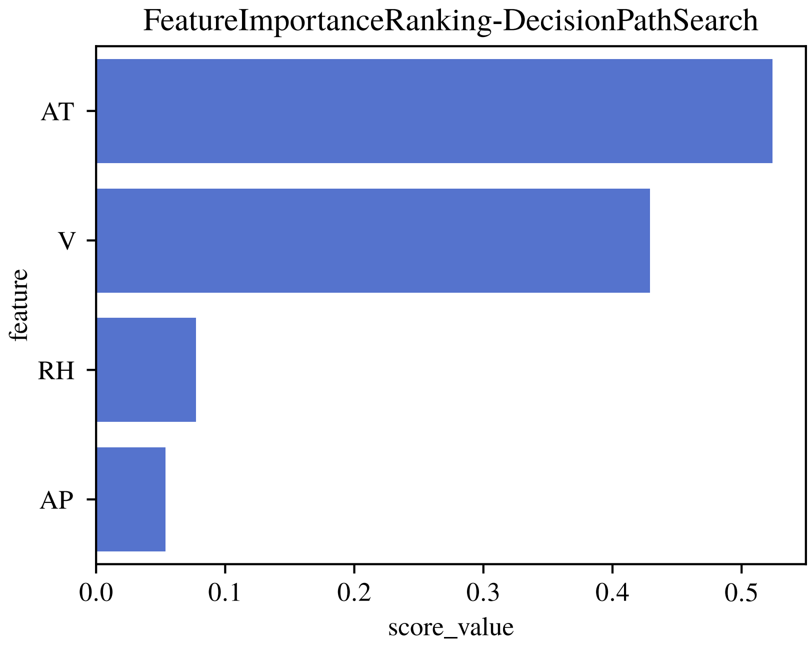


Figure 6.21: Feature importance ranking based on Decision Path Search for combined cycle power plant dataset

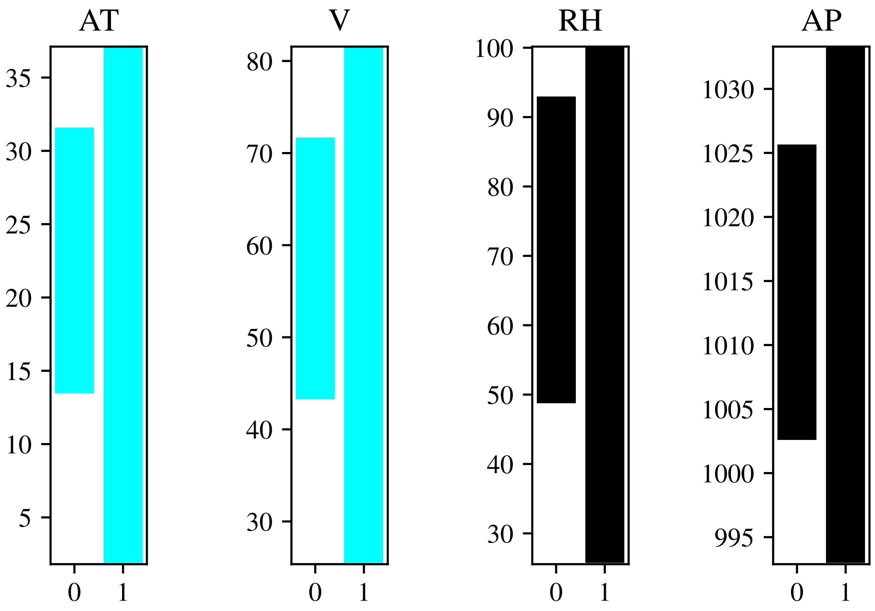


Figure 6.22: Tolerance limits of combined cycle power plant dataset

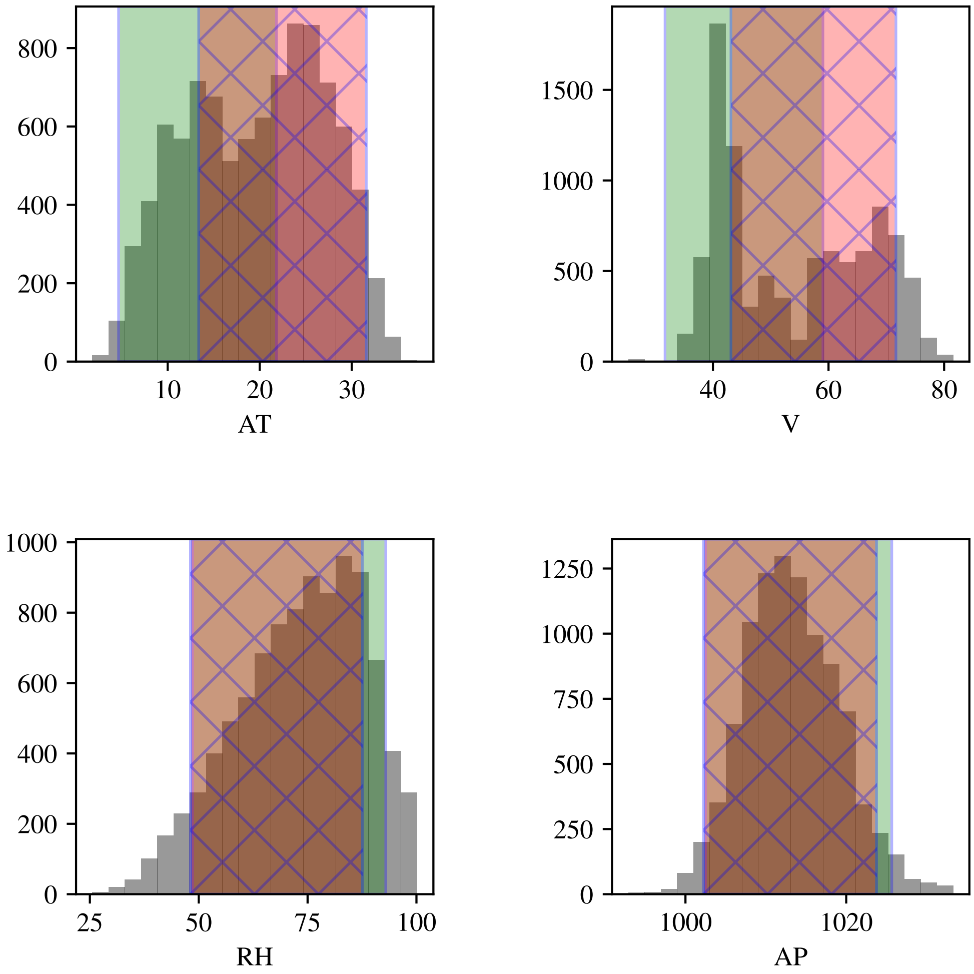


Figure 6.23: Optimal and avoidance intervals for combined cycle power plant dataset

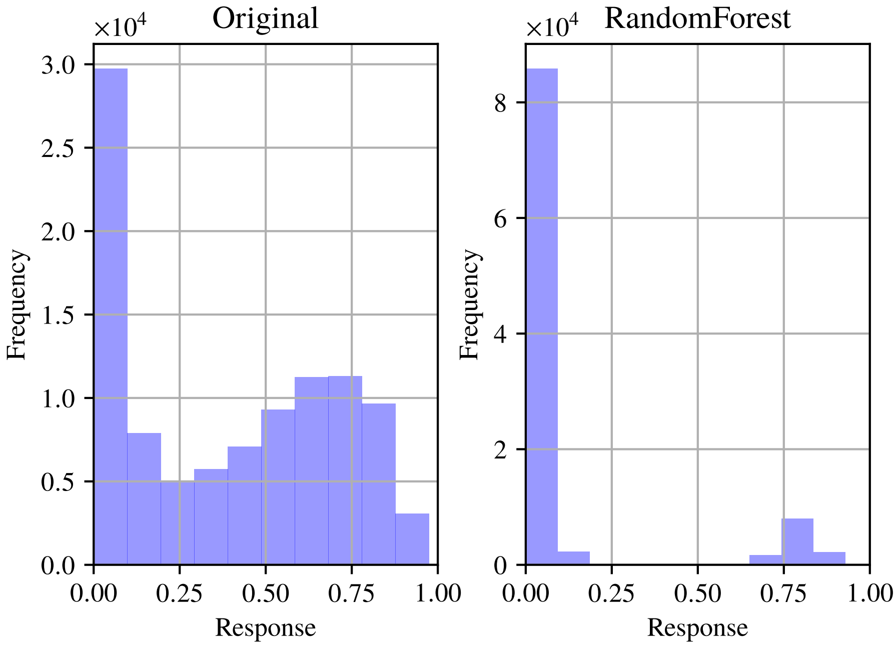


Figure 6.24: Response before and after applying the recommended process limits from MRF to the combined cycle power plant dataset

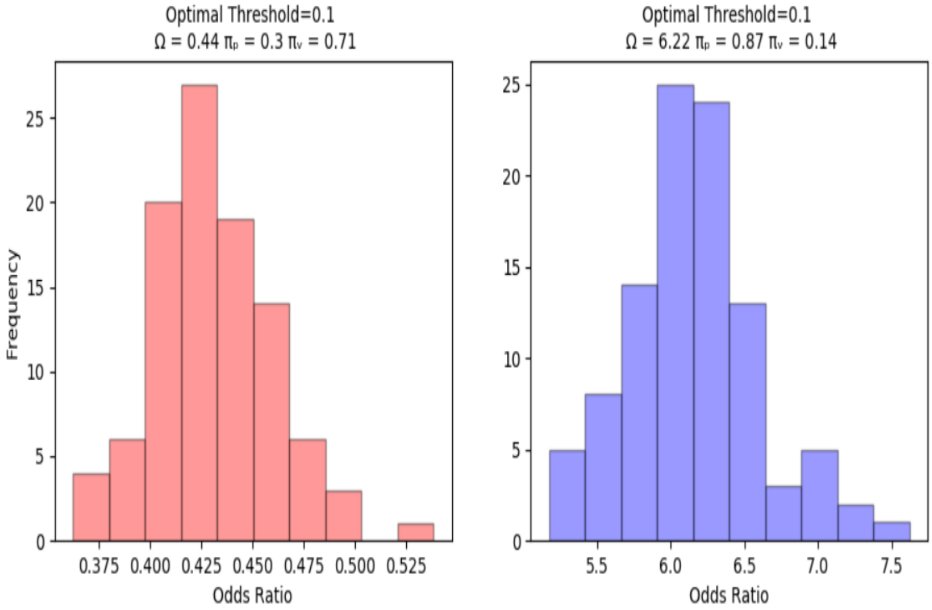


Figure 6.25: Odds ratio of original (red distributions) and optimal recommended process limits (blue distributions) of MRF applied to all factors of the combined cycle power plant dataset

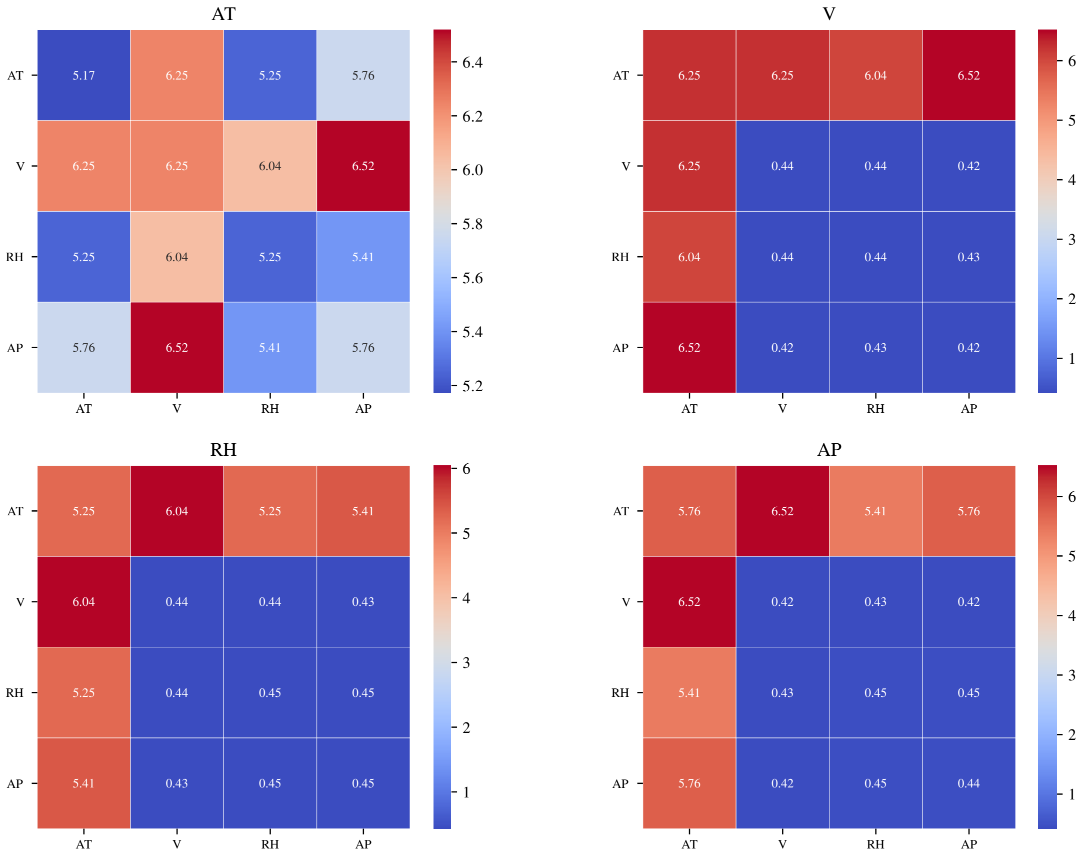


Figure 6.26: Odds ratio of interacted factors with proposed optimal limits from MRF applied to the combined cycle power plant dataset

6.3 Comparison with Published Random Forest Models

In this section, a comparison of the proposed algorithm (MRF) with published Random Forest models is discussed. As shown in Figure 6.27, three different models and datasets were compared with the proposed algorithm for further analysis and validation. The selected models are JigSaw [108] and CausalNex by McKinsey, while a comparison with the Student Performance dataset [64] is also conducted.

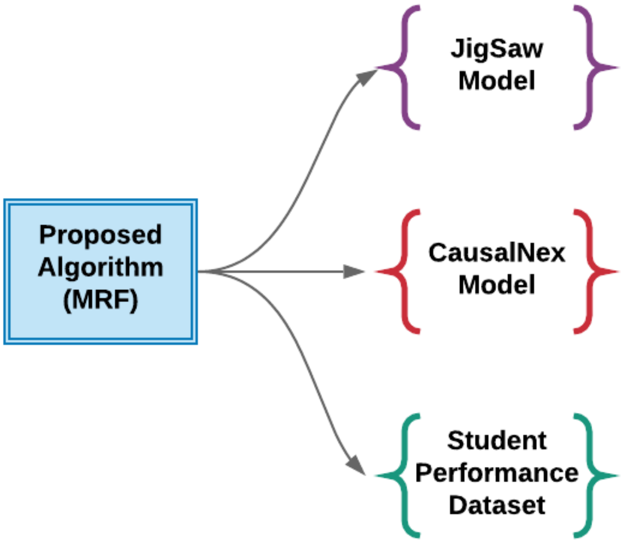


Figure 6.27: Published Random Forest models and dataset selected for comparison

6.3.1 JigSaw

Machine learning makes it possible to process big data to discern complicated trends and patterns, an outcome that has revolutionised the study of biology [108]. For example, JigSaw [108], an emerging model, was designed to predict patterns that could characterise the forest based on the Random Forest structure. Through laboratory trials, the model was verified to be effective in recovering various ground truth structures, regardless of the noise levels. Both MRF and JigSaw are based on the RF algorithm, require minimal domain expertise and are able to deal with non-linear interactions. Table 6.7 presents a comparison between the proposed algorithm and JigSaw. Although the two algorithms have different application scopes, Table 6.7 shows that MRF has certain advantages over the JigSaw algorithm.

Table 6.7: Comparison between JigSaw and MRF

Algorithm	MRF	JigSaw
Application	Developed for the discovery of causal relationships between the process factor and the target output. MRF can predict the process quality output for any production process. Also, it is used to extract the tolerance limit of the process factor.	Developed to facilitate identification of the structures (patterns) that could describe the predictions created by the forest. Moreover, it can utilise metabolites from blood measurements to assist in the determination of signs that could explain presence of breast tumours.
Application Type	Manufacturing process.	Biological science.
Data Type	Deals with both regression and classification problems.	Deals with classification problems only.
Feature Importance	Ranks the process factors based on its contribution to the prediction score (DPS).	Ranks the process factor based on their highest frequency.
Computational Cost	High performance and computationally efficient; more suitable for real-time defect prediction and heterogeneous datasets.	Computationally expensive.
Causal Relationship	Uses the PM to identify the optimal leaves and then searches for the whole paths through the forest in order to find the optimal process operating settings (optimal category/range).	Uses Euclidean distance to identify the correlations for the observed outcomes. Then, it retraces the routes through the forest to estimate the corresponding optimal limits for the feature with highest score.
Optimal Operating Range	Produces the optimal operating range for continuous data and optimal categories for the categorical data. It uses the power of ensemble method to aggregate the optimal and avoidance intervals based on the feature type (Categorical/ Regression).	Suitable for classification problems only, by finding the upper and lower decision boundaries.

6.3.2 CausalNex

CausalNex [109], is a hybrid learning technique with data and domain expertise that helps encode substantial domain knowledge in models to ensure the correct causal relationship is found while avoiding spurious relationships. CausalNex, which uses Bayesian analysis, converts continuous data into ordered categories (e.g. very low, low, medium, high, very high). Table 6.8 presents a comparison between the proposed algorithm and CausalNex. Because the deliverables of each algorithm are different (e.g. regarding whether or not it provides an optimal range), a numerical comparison between the two could not be conducted. Nonetheless, Table 6.8 highlights some differences between the two approaches and identifies the areas where the MRF algorithm has an advantage over CausalNex.

Table 6.8: Comparison between CausalNex and MRF

Algorithm	**MRF**	**CausalNex**
Domain Expertise	Minimal domain knowledge requirements.	High dependency on domain knowledge as a result of hybrid learning with data and domain expertise.
Non-Linear Interaction Within Features	Point of strength: dealing with the non-linearity in data.	Can handle non-linearity in the data.
Causal Relationships	Generates importance ranking and causal relationships for provided response range.	Provides a generic cause for the overall process.
Optimal Operating Range	Provides the optimal operating range for the given thresholds.	No optimal range is given.
Computational Cost	High performance and computationally efficient; more suitable for real-time defect prediction and heterogeneous datasets.	Computationally expensive.
Minimum Dataset Size	According to testing on several datasets, it is suggested to use at least 350 samples in order to achieve reasonable accuracy.	Based on the performed benchmarking, it is suggested to use at least 1000 samples in order to achieve reasonable accuracy.

6.3.3 Student Performance Dataset

The proposed algorithm (MRF) code was verified on the published Student Performance dataset [64] that McKinsey used for their CausalNex code. Cortez and Silva [64] used an Random Forest algorithm on the published dataset, while the provided dataset was used to calculate the regression line based on the RMSE value. The Student Performance dataset contains mixed data (both continuous and categorical features) with a total of 32 factors. The factors consist of 17 categorical features and 15 quantitative factors. In this work, 10-fold were used for the k-fold cross-validation, as seen in Table 6.9, to check the RMSE and compare it with the traditional RF, enabling an exploration of the efficiency of the proposed approach. The test was applied and compared with the traditional RF on the regression problems only, as illustrated in Figures 6.28 and 6.29. A list of comparisons between different algorithms, including the traditional RF and the MRF algorithm is presented in Table 6.10. The tables and figures below show the results of the performance on the given Student dataset for both Mathematics and Portuguese language class. Based on predictions, the proposed algorithm dominates other competitors' algorithms across both tests. Obviously, the MRF algorithm also achieved better results than the traditional Random Forest method.

Table 6.9: RMSE for 10-fold cross-validation on Student Performance dataset

k-Fold	1	2	3	4	5	6	7	8	9	10	**Mean**
RMSE$_{Math}$	1.76	1.12	1.65	2.30	1.13	1.89	2.08	1.79	1.44	1.35	**1.65**
RMSE$_{Port}$	1.29	1.52	0.99	0.92	1.85	1.52	0.72	1.11	0.99	1.18	**1.21**

Table 6.10: RMSE comparison for different algorithms using Student Performance dataset

Algorithm	RMSE$_{Math}$	RMSE$_{Port}$
Support Vector Machine (SVM)	2.09	1.35
Neural Network (NN)	2.05	1.36
Naive Predictor (NV)	2.01	1.32
Decision Tree (DT)	1.94	1.46
Random Forest (RF)	1.75	1.32
Modified Random Forest (MRF)	**1.65**	**1.21**

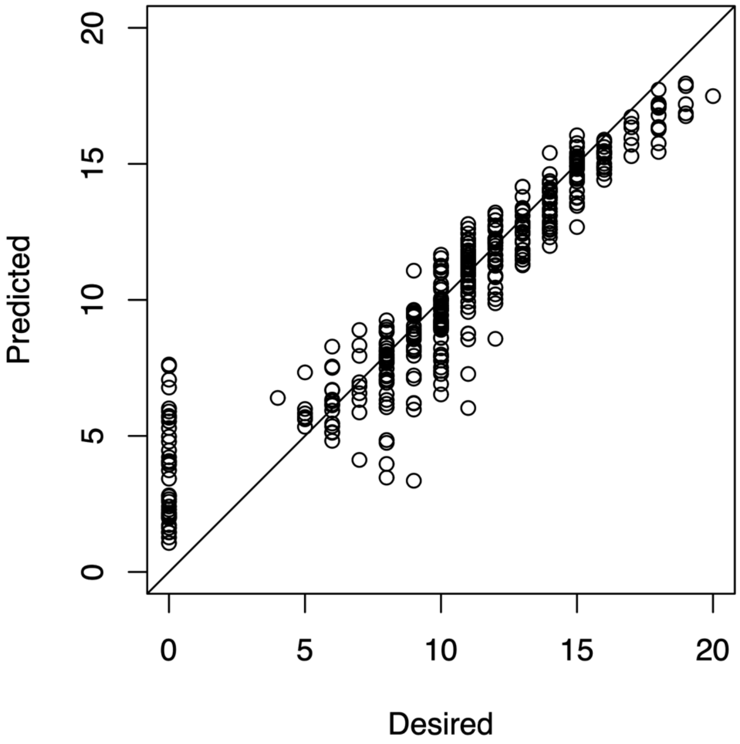


Figure 6.28: Regression model on Student Performance dataset for mathematics - Prediction by traditional Random Forest (RF) [64]

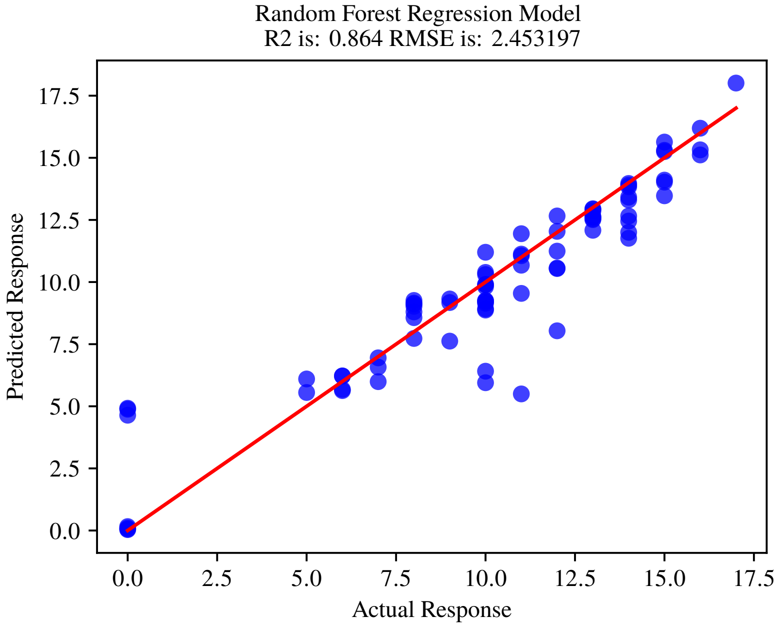


Figure 6.29: Regression model on Student Performance dataset for mathematics - Prediction by the proposed algorithm (MRF)

6.4 Discussion of Results

For the high performance concrete dataset [104], the comparison in Figure 6.30 shows that the regression coefficient R^2 value obtained by the proposed algorithm is 0.884, which is higher than that obtained by the QCA algorithm ($R^2 = 0.67$) in [8] but slightly lower than that obtained by the best model (Neural Network), which is $R^2 = 0.94$, in [104]. To compare the speed of the MRF algorithm to the Neural Network architecture proposed in [104], the same architecture was implemented with the following parameters:

Table 6.11: The parameters of Neural Network implementation

Number of Hidden Layers	1
Number of Hidden Neurons	8
Hidden Layer Activation	ReLU
Number of Neurons in Output Layers	1
Learning Rate	1
Momentum	0.5
Number of Epochs	3000

The data used to train the model is based on experiment R1 in Table 3 in [104]. The instances in the training, validation, and testing sets were sampled randomly from the original dataset (without replacement), as seen in Table 6.12. The Neural Network was implemented in Keras [110] with TensorFlow frontend [111]. Data scaling was carried out using the StandardScaler class [93] from Scikit-Learn by removing the mean and scaling to unit variance.

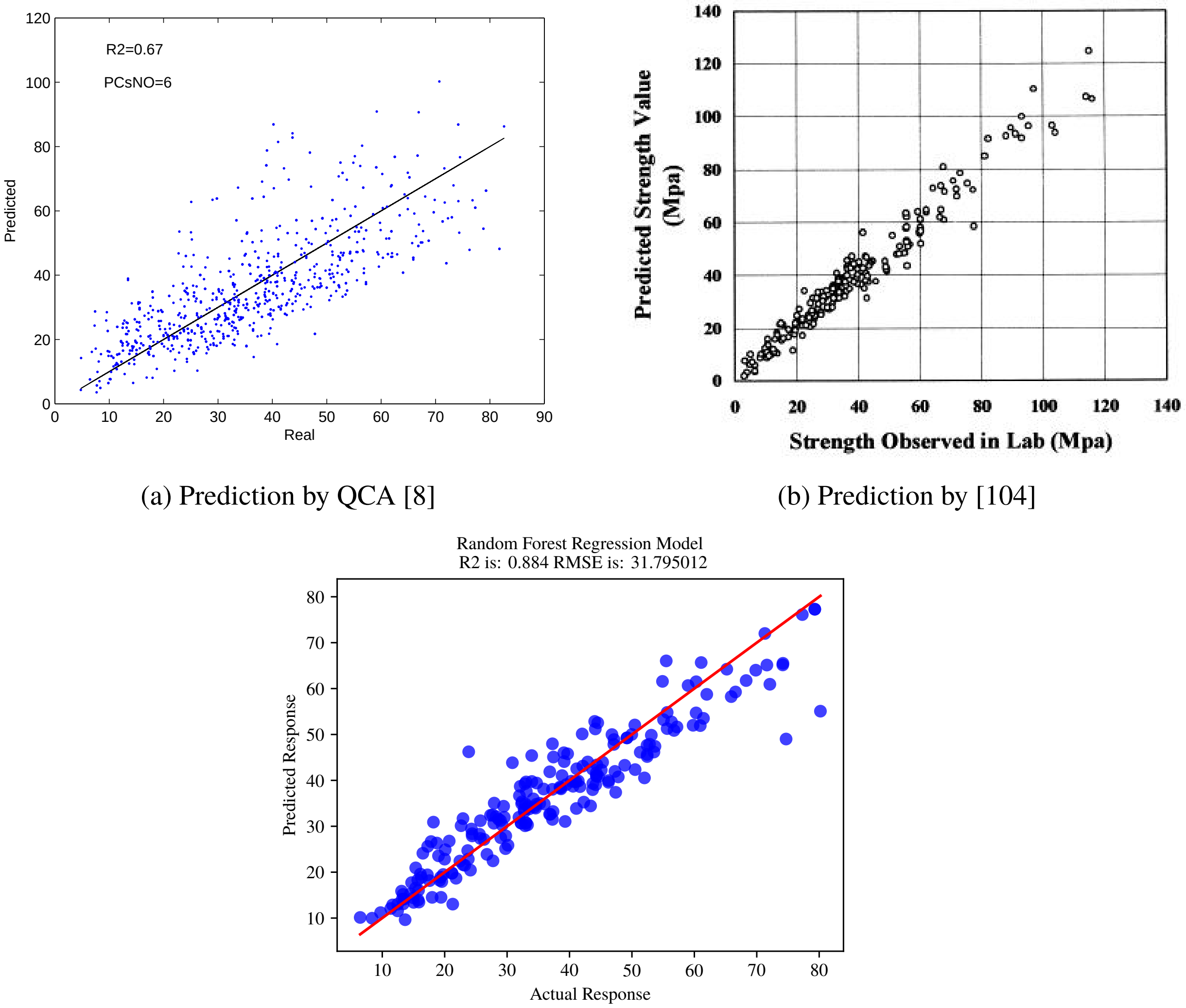
Table 6.12: Training, validation and testing instances used in [104]

Training Instances	545
Validation Instances	182
Testing Instances	182

Finally, it may be concluded that the regression capability of the proposed algorithm yields the closest performance to the Neural Network model in [104], albeit with lower computational cost (see Table 6.13), as it has a smaller number of internal parameters compared to the Neural Network model [104].

Table 6.13: Computational cost of NN and MRF code processing for high performance concrete

NN Process	Time (second)	MRF Process	Time (second)
Data Loading	0.009	Data Loading	0.21
Data Scaling	0.019	Feature Selection (FS)	8.74
Model Construction	0.032	Optimal No. of Tree	37.19
Model Training	155.914	Model Training	1.60
Model Evaluation	0.047	Model Evaluation	0.02
Total Duration	**156.021**	**Total Duration**	**47.76**



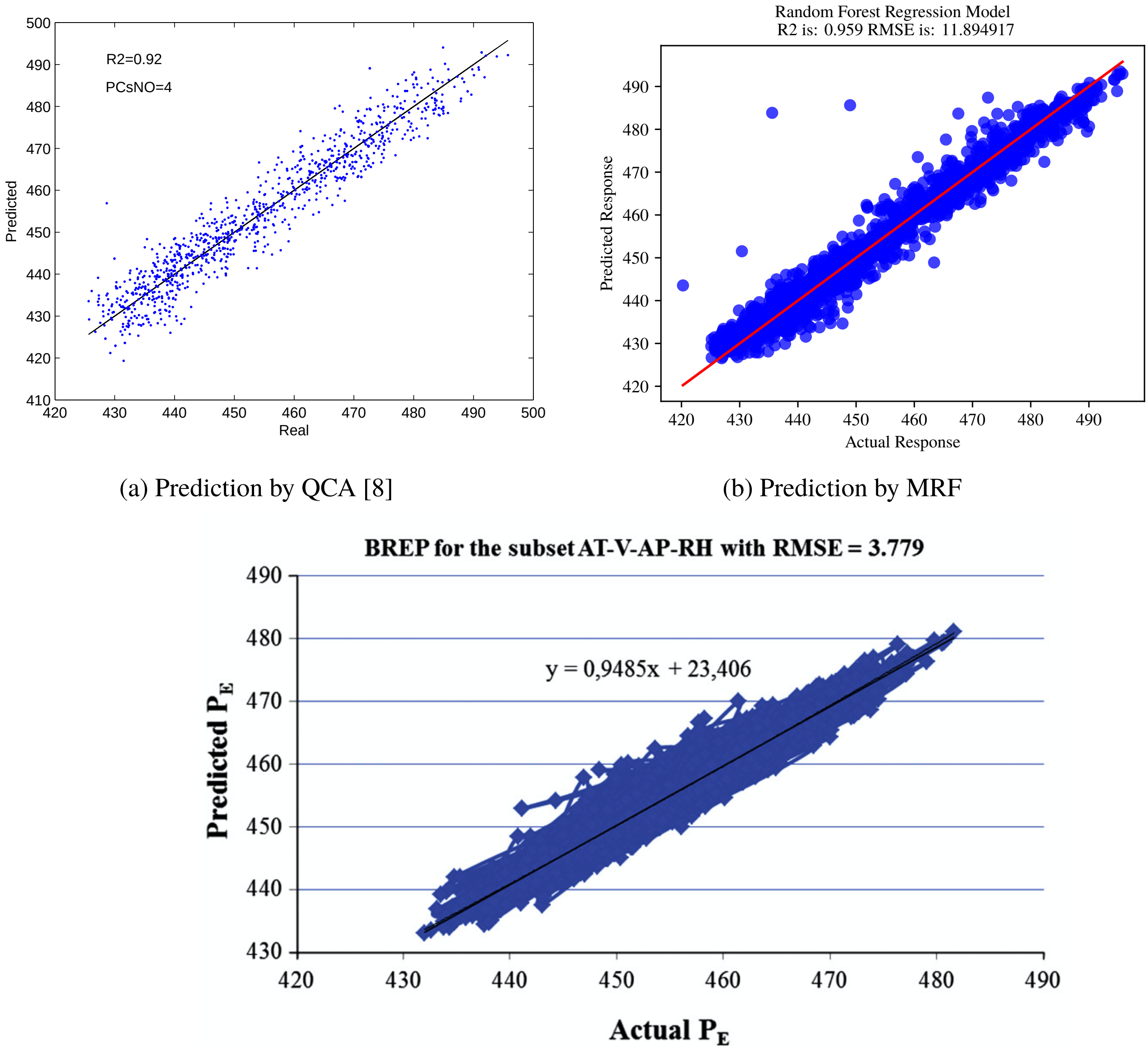
(a) Prediction by QCA [8]

(b) Prediction by [104]

(c) Prediction by MRF

Figure 6.30: High performance concrete dataset regression model

For the combined cycle power plant dataset [106], the comparison in Figure 6.31 shows that the regression coefficient R^2 value obtained by the proposed algorithm is better than that obtained by QCA and the Bagging with REPTree algorithm; the latter two obtained $R^2 = 0.92$ [8] and $R^2 = 0.9485$ [106], whereas the proposed algorithm achieved a coefficient regression value $R^2 = 0.959$. This shows that the proposed algorithm (MRF) achieve better performance than both the QCA and Bagging with REPTree algorithms.



(a) Prediction by QCA [8]

(b) Prediction by MRF

(c) Prediction by [106]

Figure 6.31: Combined cycle power plant dataset regression model

To test the MRF against the traditional Random Forest algorithm, the energy performance for residential buildings dataset was used. The MSE was calculated based on a 10-fold cross-validation approach, as seen in Table 6.14. The comparison in Table 6.15 further shows that the MSE based on the proposed algorithm is 0.42, which is lower than that obtained by the traditional Random Forest algorithm (with a value of 1.03). The proposed algorithm achieves better results in terms of prediction performance compared with the algorithm in [105].

Table 6.14: MSE for 10-fold cross-validation on energy performance for residential buildings

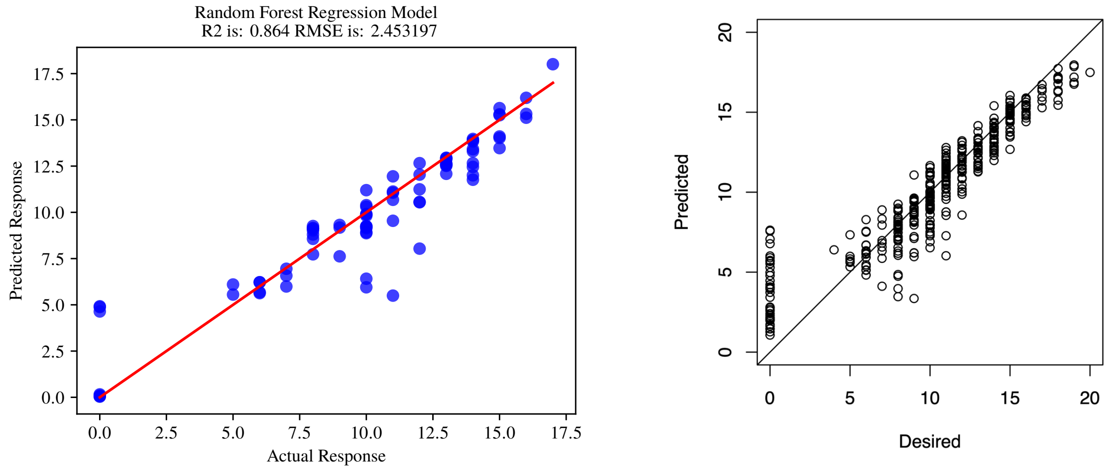
k-Fold	1	2	3	4	5	6	7	8	9	10	**Mean**
MSE	0.72	0.42	0.56	0.21	0.42	0.17	0.25	0.47	0.68	0.25	**0.42**

Table 6.15: MSE comparison for different algorithms using the energy performance dataset

Algorithm	MSE
Iterative Reweighted Least Squares (IRLS)	9.87
Random Forest (RF)	1.03
Modified Random Forest (MRF)	**0.42**

A comparison between the predictive power of MRF and JigSaw [108] cannot be conducted because each has a different application area. One aspect that makes MRF particularly suitable for tackling industrial problems is that it can handle both classification and regression (unlike JigSaw, which can only deal with classification). The computational cost of the JigSaw algorithm also could not be compared with MRF because it is unable to be applied to the problem under investigation. From the optimal range perspective, the MRF optimal range has no constraints regarding the factors, and can trace all the factors in each branch of all trees with no pre-set conditions; hence, it can provide the full optimal range for each factor. On the other hand, JigSaw has constraints on the optimal range limit, since it focuses only on the highest-ranking (frequency) factors, which is not a good strategy for the type of industrial problems considered in the present research. The process engineer needs the full optimal/avoidance ranges for each factor to conduct analysis and modifications. Based on these ranges, the process engineer can then select the best ranges to work with (given the engineer's extensive domain expertise). This is another key advantage of the MRF algorithm compared with JigSaw from the point of view of the optimal/avoidance range limit. Therefore, and looking at the selected case study, it can be seen that MRF has more advantages than JigSaw in the context of the problem investigated in this thesis.

CausalNex's [109] predictive capability could not be tested against MRF. However, given the problem at hand and its aforementioned constraints, MRF is a better alternative overall. This conclusion was reached by weighing the pros and cons of each approach for this specific case (not for the general case). It is stated on CausalNex's website [109] that the algorithm requires some requisite user experience in the area of application and that it is also not fully automated. On the other hand, the MRF algorithm requires minimal (if any) domain expertise and the process is fully automated. Other drawbacks of CausalNex are also stated on its website [109]. The model has a limit on the minimum size of the dataset required for efficient performance, and also has a high computational cost (since the model is graphical and based on Bayesian networks). Thus, the MRF algorithm is a good alternative for the investigated class of industrial problems, since it is able to overcome these limitations. It can further be seen that the MRF algorithm has advantages over CausalNex in terms of requiring minimal domain expertise, its low computational cost, and its ability to deal with small sample sizes.



(a) Prediction by MRF

(b) Prediction by [64]

Figure 6.32: Student performance dataset for mathematics regression model

Regarding the Student Performance dataset, the Mathematics and Portuguese language subsections were selected. In both cases, the proposed algorithm was found to perform better than the traditional Random Forest model used in [64]. The comparison shows that the root mean squared errors (RMSEs) obtained using the proposed algorithm were 1.65 for Mathematics and 1.21 for the Portuguese language class. These results are better than those obtained by the traditional RF model, which yielded RMSEs of 1.75 and 1.32 for Mathematics and Portuguese language, respectively. The regression model of both algorithms (MRF and RF) based on the Mathematics class is illustrated in Figure 6.32.

Conclusion

The proposed algorithm was developed to achieve the following:

- Estimate missing values for non-linear mixed data.
- Predict the process output for any given choice of operating limits.
- Determine the critical process factors.
- Predict potential quality improvement.
- Apply odds ratio by conducting a simulated confirmation trial.
- Produce high computational performance considering on high-dimensional datasets.

The backbone of this research was the development of a generalised fit predictive model to map the process input to the process response, then extract the process causal knowledge from the underlying model structure. The proposed algorithm was verified through comparison with published results on the three chosen datasets. A comparison with other published RF models was also conducted. The objective was to assess the predictive performance and find optimal tolerance limits for the process. The tolerance limits of factors that best explain the variation in response values were estimated. The impact of choosing optimal limits on response values has also been quantified. The previous work [8] was conducted on the same datasets, albeit with different goals: in this work, the high performance concrete and combined cycle power plant datasets were studied using machine learning models to predict the response [104, 106]. By contrast, in the present work, an efficient predictive model was developed. In addition, a causal knowledge technique for finding the optimal setting that leads to optimal performance was also implemented. Moreover, as a result of intensive hyperparameter tuning, the present model can give accurate predictions with minimal risk of overfitting the dataset. The regression assessment of the present model is compared with QCA, traditional RF, Bagging with REPTree, and Neural Network algorithms for the high performance concrete, energy performance of residential buildings, and combined cycle power plant datasets. Moreover, a comparison with the Student Performance dataset is also conducted based on the regression assessments of the proposed model. Figures (6.30, 6.31, 6.32) and Table 6.15 illustrate the results comparison between the proposed algorithm and the other models. Overall, following comparison of the proposed algorithm (MRF) with different datasets and models, it is evident that the proposed algorithm achieves better results in terms of both performance and prediction capability. Moreover, MRF has a lower computational cost relative to algorithms such as QCA, Neural Network, and Bayesian Networks (CausalNex).

6.5 Computational Cost Comparison between QCA and MRF

6.5.1 Nickel-based Superalloy

In this section, a comparison between QCA [8] and MRF on the nickel-based superalloy dataset is conducted to explore the efficiency of both methods in terms of computational cost.

Table 6.16: Computational cost of QCA and MRF code processing for nickel-based superalloy

QCA Process	Time (second)	MRF Process	Time (second)
Data Loading	0.036	Data Loading	0.04
Data Transformation	0.500	Feature Selection (FS)	6.00
FS and Process Limit	13.863	Optimal No. of Trees	25.53
–	–	Model Training	0.61
–	–	Model Evaluation	0.01
–	–	Process Limit	12.42
Odds Ratio	825.478	Odds Ratio	46.78
Total Duration	**839.877**	**Total Duration**	**89.38**

6.5.2 High Performance Concrete

In this section, a comparison between QCA [8] and MRF on the high performance concrete dataset is conducted to explore the efficiency of both methods in terms of computational cost.

Table 6.17: Computational cost of QCA and MRF code processing for high performance concrete

QCA Process	Time (second)	MRF Process	Time (second)
Data Loading	0.105	Data Loading	0.21
Data Transformation	1.000	Feature Selection (FS)	8.74
FS and Process Limit	124.290	Optimal No. of Trees	37.19
–	–	Model Training	1.60
–	–	Model Evaluation	0.02
–	–	Process Limit	15.41
Odds Ratio	310.223	Odds Ratio	27.71
Total Duration	**435.618**	**Total Duration**	**90.88**

6.5.3 Energy Performance on Residential Buildings

In this section, a comparison between QCA [8] and MRF on the energy performance in a residential buildings dataset is conducted to explore the efficiency of both methods in terms of computational cost.

Table 6.18: Computational cost of QCA and MRF code processing for energy performance

QCA Process	Time (second)	MRF Process	Time (second)
Data Loading	0.13	Data Loading	0.178
Data Transformation	0.009	Feature Selection (FS)	22.7
FS and Process Limit	250	Optimal No. of Trees	47.43
–	–	Model Training	1.72
–	–	Model Evaluation	0.018
–	–	Process Limit	15.7
Odds Ratio	650	Odds Ratio	47.3
Total Duration	**900.139**	**Total Duration**	**135.046**

6.5.4 Combined Cycle Power Plant

In this section, a comparison between QCA [8] and MRF on the combined cycle power plant dataset is conducted to explore the efficiency of both methods in terms of computational cost.

Table 6.19: Computational cost of QCA and MRF code processing for combined cycle power plant

QCA Process	Time (second)	MRF Process	Time (second)
Data Loading	0.38	Data Loading	0.42
Data Transformation	0.047	Feature Selection (FS)	9.56
FS and Process Limit	250	Optimal No. of Tree	281.77
–	–	Model Training	16.89
–	–	Model Evaluation	0.07
–	–	Process Limit	68.33
Odds Ratio	1131	Odds Ratio	41.81
Total Duration	**1381.427**	**Total Duration**	**418.86**

Summary and Results

From the previous tables, a comparison of the associated computational cost of the QCA [8] and the MRF algorithms reveals that the MRF has a lower computational cost. It has been observed that computational effort is concentrated on the odds ratio that results in reduced cost for the MRF procedure. Nonetheless, the QCA algorithm is useful for calculating other parts of the process, as it takes a long time to make odds ratio predictions. The QCA algorithm makes use of an iterative technique; hence, the prediction is regarded as a sequential algorithm process, which is associated with very high computational cost. On the other hand, MRF is a deterministic algorithm and can be easily parallelised. Moreover, the MRF algorithm is based on the Decision Path Search (DPS), which is primarily a prediction model, and there is no need for any iterative steps except during the construction of the decision tree. As a result, the MRF algorithm is more computationally efficient compared with sequential implementations. Therefore, the computational cost of MRF is very low, which represents another benefit in addition to its robust predictions. Figure 6.33 presents the QCA algorithm code processing flowchart.

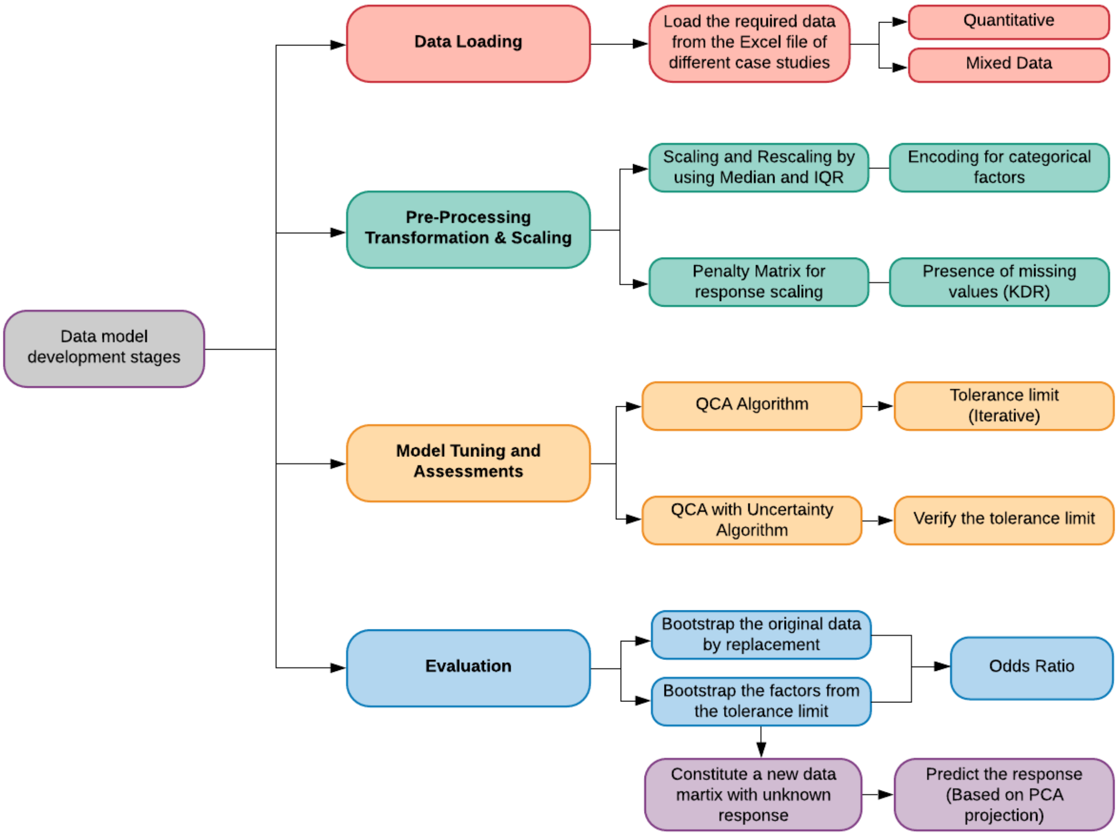

Data model development stages
Data Loading
Load the required data from the Excel file of different case studies
Quantitative
Mixed Data
Pre-Processing Transformation & Scaling
Scaling and Rescaling by using Median and IQR
Encoding for categorical factors
Penalty Matrix for response scaling
Presence of missing values (KDR)
Model Tuning and Assessments
QCA Algorithm
Tolerance limit (Iterative)
QCA with Uncertainty Algorithm
Verify the tolerance limit
Evaluation
Bootstrap the original data by replacement
Bootstrap the factors from the tolerance limit
Odds Ratio
Constitute a new data martix with unknown response
Predict the response (Based on PCA projection)

Figure 6.33: QCA processing code flowchart

6.6 Conclusions

On the published datasets, MRF was found to achieve better prediction results compared with the previous work [8] using QCA. Table 6.20 contrasts MRF with the QCA algorithm.

Table 6.20: Comparison between the present and previous works (MRF vs QCA)

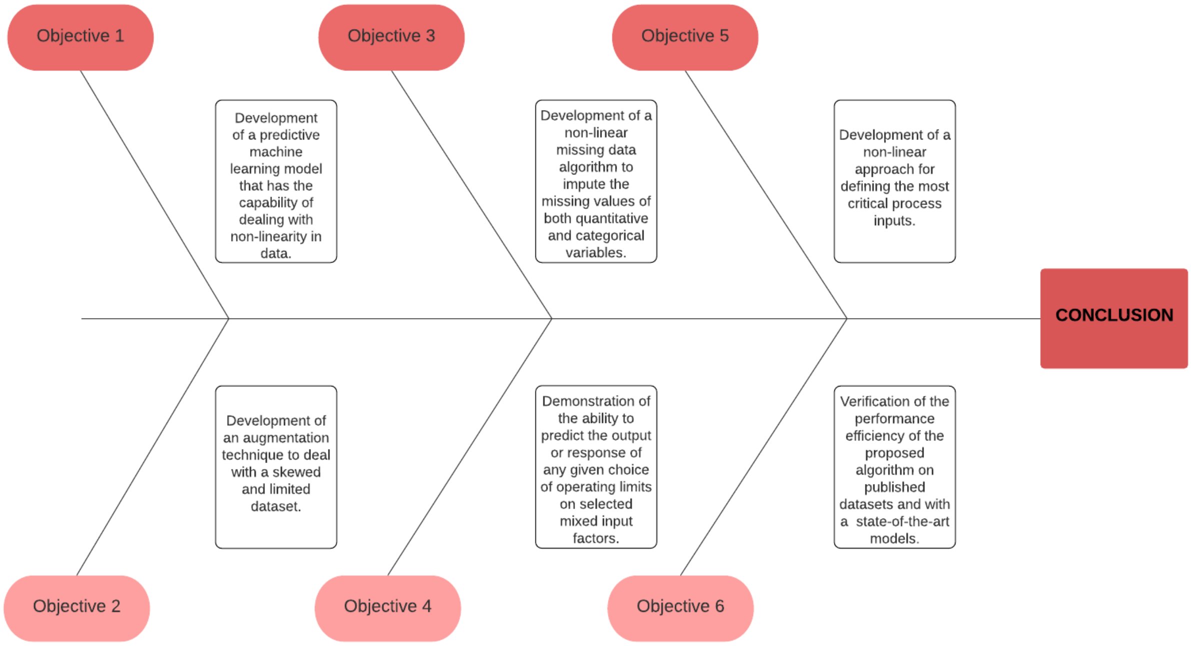
Feature	Current Work (MRF)	Previous Work (QCA)
Prediction	Predicts non-linear relationships in data.	Limited to linear process factors.
Missing Data Algorithm	missForest, shows better prediction of the missing values than KDR used in QCA.	KDR, compared with TSR and FAMD algorithms.
Feature Selection / Feature Importance	DPS, based on extracting the cause and effect knowledge from the data by scoring the critical factor based on contribution to prediction.	CLI, based on linear assumption, by highlighting factors that present opportunities for process fine-tuning.
Computational Cost	High performance and computationally efficient; more suitable for prediction and heterogeneous data.	High computational cost.
Optimal Limit	Extracts the causal knowledge from non-linear process factors using the Decision Path Search algorithm.	Estimates the optimal and avoidance limits that can improve response using the co-linearity index approach.
Confirmation Trials	Utilise odds ratio calculations to quantify the impact of the confirmation trials.	The influence of the tolerance limit is assessed by odds ratio; also identifies the non-linearity interaction in the data.
Factor Effect on Response	Various simulations are applied to assess the factor effects on the output response by applying automated tuning.	Many simulations conducted to estimate the effect of each factor on the response(s), as well as the interaction between factors.
Uncertainty	Random Forest deals naturally with the uncertainty.	Bootstrap utilised for uncertainty evaluation of the QCA.

Chapter 7

Conclusions

7.1 Main Research Contributions

The work presented in this thesis contributes to the current research into cyber-physical manufacturing plants by developing a new data-driven model and associated algorithms. A summary of the main contributions of this work concerning the original research objectives is provided in the following paragraphs.



Objective 1
Objective 3
Objective 5
Development of a predictive machine learning model that has the capability of dealing with non-linearity in data.
Development of a non-linear missing data algorithm to impute the missing values of both quantitative and categorical variables.
Development of a non-linear approach for defining the most critical process inputs.
CONCLUSION
Development of an augmentation technique to deal with a skewed and limited dataset.
Demonstration of the ability to predict the output or response of any given choice of operating limits on selected mixed input factors.
Verification of the performance efficiency of the proposed algorithm on published datasets and with a state-of-the-art models.
Objective 2
Objective 4
Objective 6

Figure 7.1: The objectives of the current work

Objective 1: Development of a predictive machine learning model that has the capability of dealing with non-linearity in data.

The MRF algorithm was developed based on the traditional RF algorithm and novel techniques for automatic tuning. The proposed model inherits the non-linear capability of RF and the prediction performance of real foundry data. Good performance was achieved on the high performance concrete dataset, especially when dealing with non-linearity in the data.

Objective 2: Development of an augmentation technique to deal with a skewed and limited dataset.

A suitable data augmentation technique was developed. The combination of SMOTE and missForest robustly augments the non-linear mixed data. The effectiveness of the proposed approach was demonstrated by analysing the in-process nickel-based superalloy foundry data.

Objective 3: Development of a non-linear missing data algorithm to impute the missing values of both quantitative and categorical variables.

An imputation method based on the Random Forest algorithm, referred to as 'missForest' [15], was used, in the augmentation technique (SMOTE-FOREST), to impute the missing values for mixed data. The results obtained from testing the algorithm with a real foundry dataset showed significantly higher accuracy than the KDR method [8].

Objective 4: Demonstration of the ability to predict the output or response of any given choice of operating limits on selected mixed input factors.

A method for the prediction of unknown responses with known factors is proposed for mixed non-linear data in the present research. The developed method is based on a tuned RF algorithm, which makes it possible to verify the recommended process limits. The verification was conducted by comparing the odds ratio of the response values with the original range, as well as with new ranges obtained via the Decision Path Search algorithm (referred to as optimal operating limits). A large number of simulations were performed to simulate different possible factor interactions. In general, the odds ratio increased significantly compared to the original value after the recommended operating limits were applied. This supports the hypothesis that the proposed optimal limits lead to improved quality and a reduction in defects.

Objective 5: Development of a non-linear approach for defining the most critical process inputs.

A novel machine learning algorithm was developed to minimise the deficiencies of the optimal tolerance limits. This algorithm, referred to as Modified Random Forest (MRF), is presented in chapters four and five. The main advantage of the proposed algorithm is that can work with a small number of observations due to the application of a novel data augmentation technique and is also capable of dealing with non-linear datasets. The algorithm is in accordance with ISO9001 requirements, as it supports continual process improvement by eliminating undesired ranges of critical factors. The essential factors that lead to desired or undesired response ranges are identified by using the novel Decision Path Search (DPS) algorithm described in chapter five. The proposed algorithm is subsequently validated by analysing historical process data of a real industrial case study, after which the results are compared with those of the QCA algorithm and PM approach. The analysis included developing a method for discovering interactions among variables by creating new variables from the interacted optimal ranges of original variables. The MRF algorithm was proven to be able to identify the factors that are responsible for defects among batches of components, or any other equivalent response variation.

Objective 6: Verification of the performance efficiency of the proposed algorithm on published datasets and conduct comparisons with a number of published state-of-the-art models.

Three datasets were analysed in chapter six to test the ability of the Modified Random Forest algorithm to analyse different data-based problems. The datasets are high performance concrete [104], energy performance of residential buildings [105] and combined cycle power plant datasets [106]. The analysis included finding the optimal limits, plus a revision of the obtained limit and the odds ratio of single/interacted factors. Finally, a comparison of results was conducted with published state-of-the-art models and dataset such as: JigSaw [108], CausalNex [109], and Student Performance dataset [64]. The proposed algorithm (MRF) shows a superior result compared to the previous models.

7.2 Future Work

The reduction of manufacturing defects is a complex task. Addressing this issue requires a multidisciplinary approach that spans the production environment, production line, machine technology, and data-driven models. This work has focused on model development challenges concerning the optimal system settings and their effects (cause and effect). However, more interdisciplinary research is required to enhance the ability of the present algorithm to predict the behaviour of the process. Future research efforts to extend the current work could include the following:

1. Investigating deep learning approaches that may prove better at dealing with non-linear data.

2. Applying either JigSaw [108] and/or CausalNex by McKinsey [109] to test the categorical variables.

3. Implementing the proposed algorithm (MRF) on real case studies with different application areas.

4. Capturing and reusing organisational knowledge from meetings, emails, papers, customer feedback, etc., then applying machine learning to these new data types, which are derived from manufacturing, accounting, sales, customer relationships, warranty, after-sales, etc.

    - Applying natural language processing to capture domain knowledge from emails, recorded meeting audio, and external PDF files. Knowledge representation for storing organisational knowledge, the computer will become able to apply this knowledge to discover causation from both correlations and from completely unstructured data that is available across the industry (emails, audio, video speech, paper, etc.) instead of predetermined cause and effect relationships.

Bibliography

[1] F. Forsyth. The future of manufacturing: A new era of opportunity and challenge for the uk. *Summary Report, The Government Office for Science, UK Foresight, London*, 2013.

[2] D. C. Montgomery. *Introduction to Statistical Quality Control*. Wiley, 2009.

[3] M. C. Staff. 50^{th} census of world casting production. *Modern Casting; American Foundry Society: Schaumburg, IL, USA*, pages 25–29, 2016.

[4] C. Seville. The european foundry association, production ferrous/non ferrous documents. *The European Foundry Association, CAEF*, 2019.

[5] C. Giannetti, R. S. Ransing, M. R. Ransing, D. C. Bould, D. T. Gethin, and J. Sienz. A novel variable selection approach based on co-linearity index to discover optimal process settings by analysing mixed data. *Computers and Industrial Engineering, Elsevier*, 72:217 – 229, 2014.

[6] M. L. George, J. Maxey, D. T. Rowlands, and M. Upton. *The Lean Six Sigma Pocket Toolbook: A Quick Reference Guide to Nearly 100 Tools for Improving Quality and Speed*. McGraw-Hill Education, 2004.

[7] S. Steiner and J. MacKay. Reducing process variation with statistical engineering: A case study. *Quality Progress Magazine*, pages 33–39, 2006.

[8] R. S. Batbooti. Data based model for defects reduction in foundry manufacturing data. *Swansea University, Mechanical Engineering Department*, 2018.

[9] A. P. Mouritz. *Introduction to aerospace materials*. Elsevier, 2012.

[10] G. M. Crankovic. Asm handbook: Volume 15 casting. *Materials Park: ASM International*, pages 416–522, 2008.

[11] C. Giannetti. Knowledge driven approaches to defect reduction and in-process quality improvement. *Swansea University, Mechanical Engineering Department*, 2015.

[12] R. S. Batbooti, R. S. Ransing, and M. R. Ransing. A bootstrap method for uncertainty estimation in quality correlation algorithm for risk based tolerance synthesis. *Computers and Industrial Engineering, Elsevier*, 112:654–662, 2017.

[13] R. S. Ransing, C. Giannetti, M. R. Ransing, and M. W. James. A coupled penalty matrix approach and principal component based co-linearity index technique to discover product specific foundry process knowledge from in-process data in order to reduce defects. *Computers in Industry*, 64(5):514 – 523, 2013.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[15] D. J. Stekhoven and P. Bühlmann. missforest non-parametric missing value imputation for mixed-type data. *Bioinformatics, Oxford University Press*, 28(1):112–118, 2012.

[16] C. Giannetti and R. S. Ransing. Risk based uncertainty quantification to improve robustness of manufacturing operations. *Computers and Industrial Engineering, Elsevier*, 101:70–80, 2016.

[17] L. Breiman. Random forests. *Machine learning, Springer*, 45(1):5–32, 2001.

[18] H. Hotteling. Multivariate quality control, illustrated by the air testing of sample bomb-sights. *Techniques of statistical analysis, McGraw-Hill New York*, pages 111–184, 1947.

[19] R. L. Mason, N. D. Tracy, and J. C Young. Decomposition of t^2 for multivariate control chart interpretation. *Journal of quality technology, Taylor and Francis*, 27(2):99–108, 1995.

[20] W. A. Shewhart. *Economic control of quality of manufactured product*. ASQ Quality Press, 1931.

[21] J. Li, J. Jin, and J. Shi. Causation-based t^2 decomposition for multivariate process monitoring and diagnosis. *Journal of Quality Technology, Taylor and Francis*, 40(1):46–58, 2008.

[22] M. J. Greenacre. *Theory and Applications of Correspondence Analysis*. Academic Press, 1984.

[23] T. Kourti and J. F. MacGregor. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and intelligent laboratory systems, Elsevier*, 28(1):3–21, 1995.

[24] D. Ceglarek, W. Huang, S. Zhou, Y. Ding, R. Kumar, and Y. Zhou. Time-based competition in multistage manufacturing: Stream-of-variation analysis (sova) methodology. *International Journal of Flexible Manufacturing Systems, Springer*, 16(1):11–44, 2004.

[25] C. Tong, Y. Song, and X. Yan. Distributed statistical process monitoring based on four-subspace construction and bayesian inference. *Industrial and Engineering Chemistry Research, ACS Publications*, 52(29):9897–9907, 2013.

[26] D. Lin, D. Banjevic, and A. Jardine. Using principal components in a proportional hazards model with applications in condition-based maintenance. *Journal of the Operational Research Society, Taylor and Francis*, 57(8):910–919, 2006.

[27] T. Wuryandari. The cox proportional hazard model on duration of birth process. *Journal of Physics: Conference Series*, pages 012–121, 2018.

[28] C. F. Slama. Multivariate statistical analysis of data obtained from an industrial fluidised catalytic process using pca and pls. In *M. Eng. Thesis*. Department of Chemical Engineering, McMaster University Hamilton, Ontario, 1991.

[29] J. E. Jackson. *A user's guide to principal components*, volume 587. John Wiley and Sons, 2005.

[30] P. Saha, N. Roy, D. Mukherjee, and A. K. Sarkar. Application of principal component analysis for outlier detection in heterogeneous traffic data. *Procedia Computer Science, Elsevier*, 83:107–114, 2016.

[31] S. Thennadil, M. Dewar, C. Herdsman, A. Nordon, and E. Becker. Automated weighted outlier detection technique for multivariate data. *Control Engineering Practice*, pages 40–49, 2017.

[32] M. Daszykowski, K. Kaczmarek, Y. Vander, and B. Walczak. Robust statistics in data analysis—a review: Basic concepts. *Chemometrics and intelligent laboratory systems, Elsevier*, 85(2):203–219, 2007.

[33] T. Y. Thanoon, R. Adnan, , and S. E. Saffari. Multiple factor analysis with continuous and dichotomous variables. *AIP Conference Proceedings 1635*, 2014.

[34] D. Ceglarek and A. S. Wu. Diagnosis approach for the launch of the auto-body assembly process. *Transformation of ASME Journal of Engineering for Industry*, 116(4), 1994.

[35] J. Liu. Variation reduction for multistage manufacturing processes: a comparison survey of statistical-process-control vs stream-of-variation methodologies. *Quality and Reliability Engineering International, Wiley Online Library*, 26(7):645–661, 2010.

[36] M. Shu and H. Wu. Measuring the manufacturing process yield based on fuzzy data. *International Journal of Production Research, Taylor and Francis*, 48(6):1627–1638, 2010.

[37] B. D. Youn, K. M. Park, C. Hu, J. T. Yoon, H. S. Kim, B. C. Jang, and Y. C. Bae. Statistical health reasoning of water-cooled power generator stator bars against moisture absorption. *IEEE Transactions on Energy Conversion, IEEE*, 30(4):1376–1385, 2015.

[38] L. Yang and J. Lee. Bayesian belief network-based approach for diagnostics and prognostics of semiconductor manufacturing systems. *Robotics and Computer-Integrated Manufacturing, Elsevier*, 28(1):66–74, 2012.

[39] Y. Liu and S. Jin. Application of bayesian networks for diagnostics in the assembly process by considering small measurement data sets. *The International Journal of Advanced Manufacturing Technology, Springer*, 65(9-12):1229–1237, 2013.

[40] P. J. Brown, M. Vannucci, and T. Fearn. Multivariate bayesian variable selection and prediction. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), Wiley Online Library*, 60(3):627–641, 1998.

[41] A. Vehtari and J. Ojanen. A survey of bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys, The author, under a Creative Commons Attribution License*, 6:142–228, 2012.

[42] S. Dey and J. A. Stori. A bayesian network approach to root cause diagnosis of process variations. *International Journal of Machine Tools and Manufacture, Elsevier*, 45(1):75–91, 2005.

[43] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. *Proceedings of the 13th International Conference on Neural Information Processing System*, pages 388–394, 2000.

[44] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research 6*, pages 1579–1619, 2000.

[45] D. T. Pham. Control chart pattern recognition using neural networks. *Journal of Systems Engineering*, 2:256–262, 1992.

[46] D. T. Pham and E. Oztemel. Control chart pattern recognition using learning vector quantisation networks. *The International Journal of Production Research, Taylor and Francis*, 32(3):721–729, 1994.

[47] C. S. Cheng. A multi-layer neural network model for detecting changes in the process mean. *Computers and Industrial Engineering, Elsevier*, 28(1):51–61, 1995.

[48] R. S. Guh and J. D. Tannock. A neural network approach to characterize pattern parameters in process control charts. *Journal of Intelligent Manufacturing, Springer*, 10(5):449–462, 1999.

[49] T. Y. Wang and L. H. Chen. Mean shifts detection and classification in multivariate process: a neural-fuzzy approach. *Journal of Intelligent Manufacturing, Springer*, 13(3):211–221, 2002.

[50] L. H. Chen and T. Y. Wang. Artificial neural networks to classify mean shifts from multivariate χ^2 chart signals. *Computers and Industrial Engineering, Elsevier*, 47(2-3):195–205, 2004.

[51] S. T. Niaki and B. Abbasi. Detection and classification mean-shifts in multi-attribute processes by artificial neural networks. *International Journal of Production Research, Taylor and Francis*, 46(11):2945–2963, 2008.

[52] S. Psarakis. The use of neural networks in statistical process control charts. *Quality and Reliability Engineering International, Wiley Online Library*, 27(5):641–650, 2011.

[53] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 01 1992.

[54] K. Singh and M. Xie. Bootstrap: A statistical method. *College of Engineering, Rutgers University*, 2008.

[55] P. Linares. Using multiple linear regression based on principal component analysis (factor analysis of mixed data famd) for predicting the final score of secondary students from portugal. *Nestor Post-Graduate in Data Science in the Technological University Dublin, Ireland*, 2019.

[56] M. Arowolo, M. O. Adebiyi, and A. A. Adebiyi. An efficient pca ensemble learning approach for prediction of rna-seq malaria vector gene expression data classification. *International Journal of Engineering Research and Technology, International Research Publication House.*, 13(1):163–169, 2020.

[57] T. K. Akino and Y. Wang. Stochastic bottleneck: Rateless auto-encoder for flexible dimensionality reduction. *Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA*, 1(2):28–70, 2020.

[58] P. A. Ferrari, P. Annoni, A. Barbiero, and G. Manzi. An imputation method for categorical variables with application to non-linear principal component analysis. *Computational Statistics and Data Analysis, Elsevier*, 55(7):2410–2420, 2011.

[59] M. E. Paoletti, J. M. Haut, X. Tao, J. P. Miguel, and A. Plaza. A new gpu implementation of support vector machines for fast hyperspectral image classification. *Hyperspectral Computing Laboratory (HyperComp), Department of Computer Technology and Communications. Escuela Politecnica de Caceres, University of Extremadura, Spain, Remote Sensing Publications, MDPI*, 12:12–57, 2020.

[60] B. Verlinden, J. R. Duflou, P. Collin, and D. Cattrysse. Cost estimation for sheet metal parts using multiple regression and artificial neural networks: A case study. *International Journal of Production Economics, Elsevier*, 111(2):484–492, 2008.

[61] I. Mohanty, D. Bhattacharjee, and S. Datta. Designing cold rolled if steel sheets with optimized tensile properties using ann and ga. *Computational materials science, Elsevier*, 50(8):2331–2337, 2011.

[62] A. Doroshenko. Applying artificial neural networks in construction. *Moscow state university of civil engineering, Web of Conferences 143*, 1:10–29, 2020.

[63] A. Mollalo, K. M. Rivera, and B. Vahedi. Artificial neural network modelling of novel coronavirus (covid-19) incidence rates across the continental united states. *International Journal of Environmental Research and Public Health, MDPI*, 17:42–55, 2020.

[64] P. Cortez and A. M. Silva. Using data mining to predict secondary school student performance. *EUROSIS-ETI*, 2008.

[65] T. K. Ho. Random decision forests. *Proceedings of the Third International Conference on Document Analysis and Recognition*, pages 278–282, 1995.

[66] Y. Amit and D. Geman. Shape quantisation and recognition with randomized trees. *Neural Computation*, pages 1545–1588, 1997.

[67] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 832–844, 1998.

[68] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomisation. *Machine Learning*, pages 139–157, 2000.

[69] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. *Wadsworth, Belmont, CA*, pages 412–418, 1984.

[70] R. J. Little and D. B. Rubin. *Statistical analysis with missing data*, volume 793. John Wiley and Sons, 2019.

[71] S. Laaksonen. Regression-based nearest neighbour hot decking. In *International workshop on household survey nonresponse*, volume 4, pages 285–298. DEU, 1998.

[72] G. E. Batista and M. C. Monard. An analysis of four missing data treatment methods for supervised learning. *Applied artificial intelligence, Taylor and Francis*, 17(5-6):519–533, 2003.

[73] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics, Oxford University Press*, 17(6):520–525, 2001.

[74] D. Li, J. Deogun, W. Spaulding, and B. Shuart. Towards missing data imputation: a study of fuzzy k-means clustering method. In *International conference on rough sets and current trends in computing*, pages 573–579. Springer, 2004.

[75] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[76] S. W. Looney. Practical issues in sample size determination for correlation coefficient inference. *SM Journal of Biometrics Biostatistics*, 3(1):1–4, 2018.

[77] P. Probst, M. Wright, and A. Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2019.

[78] R. W. Johnson. An introduction to the bootstrap. *Teaching Statistics, Wiley Online Library*, 23(2):49–54, 2001.

[79] A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*, volume 1. Cambridge university press, 1997.

[80] M. Kuhn and K. Johnson. *Applied predictive modeling*, volume 26. Springer, 2013.

[81] Y. P. Chaubey. Resampling methods: A practical guide to data analysis. *Taylor and Francis*, 2000.

[82] D. P. Doane and L. E. Seward. Measuring skewness: A forgotten statistic? *Journal of Statistics Education Volume 19, Number 2*, 2011.

[83] M. Faria, F. Oliveira, and G. Pimentel-Junior. Acoustic emission tests on the analysis of cracked shafts of different crack depths. *ABCM International Congress of Mechanical Engineering, RJ, Brasil*, 2015.

[84] P. H. Westfall. Kurtosis as peakedness. *Author manuscript PMC*, pages 191–195, 2014.

[85] Z. Liang, J. Wei, J. Zhao, H. Liu, B. Li, J. Shen, and C. Zheng. The statistical meaning of kurtosis and its new application to identification of persons based on seismic signals. *Shanghai Institute of Micro-system and Information Technology, Chinese Academy of Sciences*, 2008.

[86] F. Falah, S. G. Nejad, O. Rahmati, M. Daneshfar, and H. Zeinivand. Applicability of generalised additive model in groundwater potential modelling and comparison its performance by bivariate statistical methods. *Geocarto International*, 2016.

[87] H. Jabbar and R. Khan. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). *Computer Science, Communication Instrumentation Devices*, 2002.

[88] R. S. Ransing, R. S. Batbooti, C. Giannetti, and M. R. Ransing. A quality correlation algorithm for tolerance synthesis in manufacturing operations. *Computers and Industrial Engineering, Elsevier*, 93:1–11, 2016.

[89] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[90] J. R. Quinlan. Induction of decision trees - machine learning. *Kluwer Academic Publishers, Boston*, 1986.

[91] J. R. Quinlan. *C4.5: Programs for machine learning*. Elsevier - Machine Learning (ML), 2014.

[92] S. S. Shwartz and S. B. David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[93] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[94] C. Giannetti and R. S. Ransing. Risk-based tolerance synthesis and uncertainty quantification of in-process data to predict robustness of industry 4.0 processes. *College of Engineering, Swansea University*, 2015.

[95] S. Fortmann. Understanding the bias-variance tradeoff. *Bias-Variance ROE*, 2012.

[96] L. Breiman. Bagging predictors. *Machine learning, Springer*, 24(2):123–140, 1996.

[97] M. R. Chernick. *Bootstrap methods: A guide for practitioners and researchers*, volume 619. John Wiley and Sons, 2011.

[98] S. Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *University of Wisconsin Madison, Department of Statistics*, 2018.

[99] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics, Oxford University Press*, 19(16):2088–2096, 2003.

[100] C. Strobl, A. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC bioinformatics, Springer*, 9(1):307, 2008.

[101] R. Genuer, J. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern recognition letters, Elsevier*, 31(14):2225–2236, 2010.

[102] A. Agresti. *Categorical data analysis*, volume 482. John Wiley and Sons, 2003.

[103] A. M. Liberman. How much more likely? the implications of odds ratios for probabilities. *American Journal of Evaluation, Publications Sage CA: Thousand Oaks, CA*, 26(2):253–266, 2005.

[104] I. C. Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research, Elsevier*, 28(12):1797–1808, 1998.

[105] A. Tsanas and A. Xifara. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings, Elsevier*, 49:560–567, 2012.

[106] P. Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power and Energy Systems, Elsevier*, 60:126–140, 2014.

[107] L. X. Niu and X. J. Liu. Multivariable generalised predictive scheme for gas turbine control in combined cycle power plant. In *2008 IEEE Conference on Cybernetics and Intelligent Systems*, pages 791–796. IEEE, 2008.

[108] D. DiMucci. Jigsaw: A tool for discovering explanatory high-order interactions from random forests. *Department of Microbiology, The Forsyth Institute, Cambridge, MA, United States*, 2020.

[109] McKinsey. A first causalnex tutorial — causalnex 0.7.0 documentation, 2020.

[110] F. Chollet. Keras: The python deep learning application, api, $url = https://www.keras.io$, 2020.

[111] G. Brain. The tensor-flow frontend group, pyhton library, $url = https://www.tensorflow.org$, 2020.

Appendices

Appendix A: The Investment Casting

A Nickel-based Superalloy Dataset

The Investment Casting - A Nickel-based Superalloy Dataset

An aerospace foundry's cast components are made from a dataset of a nickel-based superalloy. Each dataset is composed of 16 factors as illustrated in Figure A.1. These factors (chemical compositions) affect the responses of defective components manufactured in a particular batch due to a shrinkage defect. Chemical compositions are input factors, which are quantitative variables. The process data consists of 60 batches or observations, while each batch is composed of 16 factors. In addition, each batch contains the percentage of shrinkage defects in the observed batch. The process data also relate to the percentage of castings rejected as a result of shrinkage and other defects. Furthermore, the final product of investment casting is represented by a single batch. The remainder of the processing variables, such as temperature, moulding criteria, cementing, etc, are fixed in all factor cases. Tables A.1 - A.3 list the process inputs in terms of chemical composition, along with the 16 factors contained in each batch for the nickel-based superalloy.

%C
Carbon
%Al
Aluminium
%B
Boron
%Co
Cobalt
%Cr
Chromium
%Fe
Iron
%Mo
Molybdenum
%N
Nitrogen
%Nb
Niobium
%O
Oxygen
%Ta
Tantalum
%Ti
Titanium
%W
Tungsten
%Zr
Zirconium
%Al+Ti
Aluminium and Titanium
%Ta/Ti
Tantalum over Titanium

Figure A.1: Chemical composition factors

A small number of components for a given product that are processed, and inspected together form a batch. The key process variables are recorded in the beginning of each process step and are associated with the batch. At every inspection point, rejection rates are recorded and the corresponding data is associated with the batch. For this particular case study, the rejection rates became high only for one product type. The process parameters were within the given tolerance limits and all other product types were within acceptable rejection rates. As a result the process was considered as robust and it was felt that the root-cause could either be material or design specific for the said product type. Before investigating potential design changes (e.g. feeding or gate design), it was decided to investigate the effect of variation in chemical composition. The chemistry was later optimised for this product type however the dataset became useful as a benchmark dataset in this as well as the previous research in the research group. The reasons for choosing the nickel-based superalloy dataset in this thesis, illustrated in Figure A.2, are as follows:

- The dataset contains limited observation.
- Non-linear interactions are observed.
- It is highly skewed dataset.
- The dataset is based on a real-world example and used in previous Swansea research.

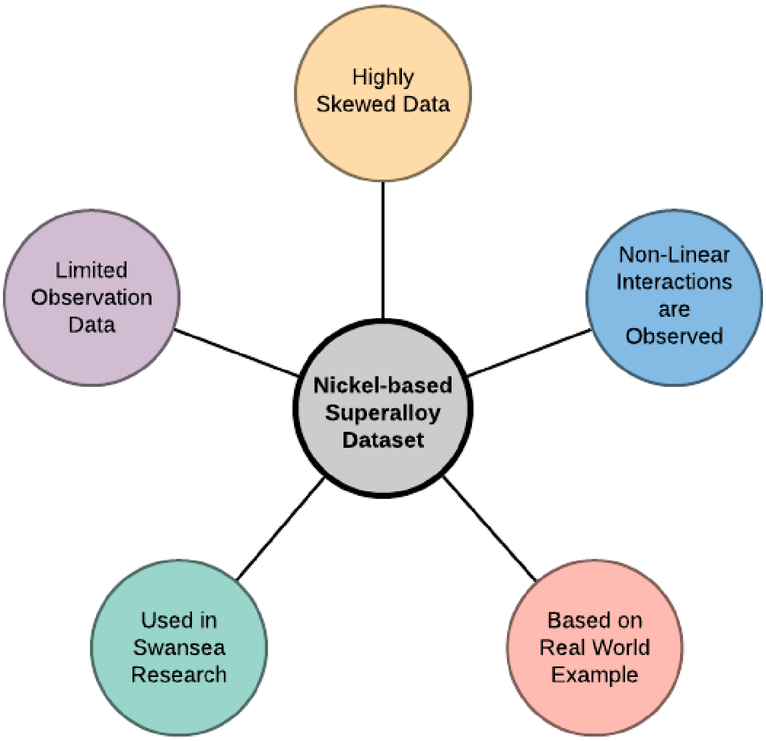

Highly Skewed Data
Limited Observation Data
Non-Linear Interactions are Observed
Nickel-based Superalloy Dataset
Used in Swansea Research
Based on Real World Example

Figure A.2: Advantages of using the nickel-based superalloy dataset

Table A.1: Nickel-based superalloy dataset properties - Part 1

Shrink Penalty	$\%C$	$\%Al$	$\%B$	$\%\ Co$	$\%Cr$	$\%Fe$	$\%Mo$	$\%\ Nb$	$\%Ta$	$\%Ti$	$\%W$	$\%\ Zr$	$\%Al+Ti$	$\%\ N$	$\%\ O$	$\%\ Ta/Ti$
0.12	0.101	3.230	0.009	7.857	15.200	0.086	1.663	0.846	1.587	3.230	2.556	0.037	6.460	33.250	6.650	0.492
0	0.093	3.145	0.009	7.971	15.295	0.086	1.644	0.798	1.558	3.211	2.594	0.050	6.365	11.400	6.650	0.486
0.15	0.107	3.249	0.009	7.781	15.248	0.152	1.691	0.893	1.653	3.278	2.423	0.031	6.527	38.000	10.450	0.505
0	0.103	3.249	0.008	8.028	15.096	0.105	1.653	0.865	1.568	3.211	2.489	0.035	6.460	22.800	7.600	0.489
0	0.105	3.183	0.008	7.781	15.001	0.124	1.682	0.808	1.520	3.107	2.423	0.032	6.289	21.850	5.700	0.490
0	0.107	3.107	0.008	7.800	15.295	0.190	1.663	0.808	1.615	3.145	2.442	0.022	6.251	20.900	8.550	0.514
0	0.109	3.145	0.010	7.866	15.267	0.095	1.691	0.770	1.653	3.192	2.451	0.024	6.337	20.900	9.500	0.518
0	0.112	3.287	0.009	7.743	15.305	0.190	1.663	0.817	1.672	3.211	2.461	0.023	6.498	26.600	5.700	0.521
0.02	0.106	3.145	0.009	7.838	15.352	0.095	1.644	0.808	1.596	3.164	2.461	0.023	6.308	38.950	11.400	0.505
0	0.106	3.249	0.008	7.809	15.276	0.095	1.634	0.817	1.558	3.173	2.480	0.024	6.422	30.400	3.800	0.492
0	0.108	3.097	0.008	7.781	15.286	0.095	1.653	0.836	1.625	3.202	2.432	0.023	6.299	29.450	7.600	0.508
0	0.108	3.183	0.008	7.828	15.286	0.095	1.634	0.798	1.577	3.145	2.480	0.026	6.327	20.900	7.600	0.502
0	0.106	3.240	0.008	7.857	15.020	0.143	1.663	0.865	1.615	3.173	2.518	0.030	6.413	32.300	5.700	0.509
0	0.108	3.268	0.009	7.895	15.267	0.171	1.672	0.865	1.568	3.211	2.470	0.030	6.489	32.300	7.600	0.489
0	0.102	3.306	0.008	7.885	15.248	0.114	1.634	0.817	1.492	3.145	2.489	0.027	6.451	36.100	11.400	0.475
0.07	0.102	3.306	0.009	7.942	15.229	0.067	1.663	0.865	1.549	3.183	2.470	0.031	6.489	20.900	7.600	0.487
0	0.102	3.306	0.009	7.828	15.276	0.114	1.691	0.846	1.530	3.154	2.470	0.030	6.460	33.250	12.350	0.485
0	0.102	3.268	0.009	7.876	15.143	0.114	1.672	0.855	1.520	3.145	2.508	0.033	6.422	30.400	9.500	0.484
0	0.096	3.230	0.007	7.942	15.333	0.133	1.629	0.808	1.587	3.116	2.413	0.035	6.356	36.100	36.100	0.510
0	0.104	3.221	0.008	7.876	15.162	0.143	1.653	0.828	1.539	3.211	2.442	0.032	6.432	24.700	11.400	0.480

Table A.2: Nickel-based superalloy dataset properties - Part 2

Shrink Penalty	$\%C$	$\%Al$	$\%B$	$\% Co$	$\%Cr$	$\%Fe$	$\%Mo$	$\% Nb$	$\%Ta$	$\%Ti$	$\%W$	$\% Zr$	$\%Al+Ti$	$\% N$	$\% O$	$\% Ta/Ti$
0	0.104	3.240	0.007	7.847	15.143	0.152	1.682	0.817	1.568	3.183	2.442	0.034	6.413	19.000	7.600	0.493
0	0.103	3.240	0.009	7.923	15.172	0.133	1.663	0.817	1.539	3.135	2.461	0.034	6.375	20.900	10.450	0.491
0	0.097	3.230	0.009	7.714	15.238	0.114	1.672	0.846	1.634	3.183	2.413	0.032	6.431	28.500	5.700	0.514
0.02	0.108	3.107	0.009	7.838	15.295	0.190	1.663	0.817	1.587	3.192	2.442	0.024	6.299	23.750	5.700	0.498
0.02	0.110	3.088	0.009	7.714	15.314	0.190	1.663	0.827	1.615	3.164	2.451	0.024	6.251	18.050	7.600	0.511
0	0.108	3.164	0.009	7.819	15.428	0.095	1.691	0.827	1.644	3.192	2.470	0.025	6.356	23.750	6.650	0.515
0	0.104	3.097	0.010	7.828	15.428	0.190	1.663	0.836	1.577	3.211	2.404	0.024	6.308	22.800	5.700	0.492
0	0.106	3.088	0.009	7.838	15.286	0.190	1.663	0.798	1.587	3.116	2.442	0.026	6.204	14.250	7.600	0.510
0	0.105	3.145	0.008	7.838	15.162	0.133	1.710	0.808	1.558	3.154	2.423	0.032	6.299	32.300	7.600	0.494
0	0.099	3.240	0.007	7.885	15.248	0.124	1.663	0.808	1.587	3.192	2.451	0.038	6.441	31.350	11.400	0.498
0	0.112	3.211	0.008	7.828	15.020	0.124	1.701	0.817	1.606	3.116	2.480	0.037	6.318	20.900	11.400	0.516
0	0.104	3.240	0.008	7.857	15.153	0.114	1.653	0.817	1.606	3.221	2.489	0.038	6.460	25.650	8.550	0.499
0.02	0.104	3.230	0.008	7.819	15.115	0.133	1.691	0.827	1.596	3.173	2.480	0.035	6.403	25.650	9.500	0.503
0	0.104	3.249	0.009	7.790	15.248	0.143	1.691	0.827	1.596	3.183	2.413	0.035	6.432	36.100	4.750	0.502
0	0.102	3.268	0.008	7.790	15.153	0.133	1.644	0.808	1.539	3.116	2.518	0.032	6.384	31.350	10.450	0.494
0.03	0.103	3.164	0.008	7.800	15.162	0.133	1.663	0.827	1.539	3.097	2.423	0.036	6.261	31.350	10.450	0.497
0.21	0.113	3.135	0.010	7.828	15.314	0.095	1.663	0.827	1.625	3.221	2.480	0.023	6.356	35.150	9.500	0.505
0	0.109	3.145	0.009	7.828	15.286	0.133	1.653	0.798	1.568	3.173	2.451	0.024	6.318	32.300	6.650	0.495
0	0.107	3.107	0.009	7.847	15.371	0.095	1.663	0.798	1.596	3.202	2.527	0.024	6.308	16.150	10.450	0.499
0	0.100	3.107	0.011	7.771	15.229	0.190	1.663	0.798	1.549	3.154	2.432	0.024	6.261	34.200	7.600	0.491

Table A.3: Nickel-based superalloy dataset properties - Part 3

Shrink Penalty	$\%C$	$\%Al$	$\%B$	$\%\,Co$	$\%Cr$	$\%Fe$	$\%Mo$	$\%\,Nb$	$\%Ta$	$\%Ti$	$\%W$	$\%\,Zr$	$\%Al+Ti$	$\%\,N$	$\%\,O$	$\%\,Ta/Ti$
0	0.102	3.192	0.009	7.781	15.276	0.095	1.644	0.808	1.568	3.173	2.461	0.022	6.365	30.400	4.750	0.495
0	0.104	3.278	0.008	7.917	15.238	0.152	1.701	0.789	1.539	3.211	2.518	0.034	6.489	21.850	4.750	0.480
0	0.102	3.202	0.008	7.847	15.181	0.133	1.663	0.798	1.530	3.192	2.480	0.035	6.394	21.850	5.700	0.480
0	0.099	3.183	0.008	7.895	15.381	0.200	1.653	0.789	1.577	3.135	2.499	0.031	6.318	34.200	11.400	0.504
0	0.104	3.173	0.010	7.809	15.276	0.095	1.663	0.808	1.549	3.164	2.404	0.023	6.337	27.550	2.850	0.490
0.13	0.086	3.078	0.008	7.847	15.257	0.095	1.701	0.770	1.539	3.183	2.337	0.019	6.261	22.800	9.500	0.484
0.26	0.095	3.097	0.012	8.009	15.371	0.076	1.644	0.703	1.558	3.173	2.290	0.019	6.270	27.550	4.750	0.492
0.26	0.095	3.059	0.008	7.885	15.295	0.067	1.672	0.722	1.587	3.202	2.366	0.019	6.261	29.450	6.650	0.496
0.13	0.086	3.192	0.009	7.847	15.067	0.067	1.653	0.656	1.444	3.097	2.451	0.021	6.289	18.050	4.750	0.467
0.04	0.095	3.154	0.007	7.999	15.124	0.076	1.596	0.684	1.596	3.154	2.404	0.025	6.308	21.850	16.150	0.507
0	0.095	3.135	0.010	7.809	15.096	0.067	1.710	0.713	1.501	3.221	2.394	0.023	6.356	19.000	7.600	0.467
0.22	0.086	3.135	0.009	8.018	15.153	0.076	1.615	0.722	1.644	3.154	2.442	0.026	6.289	20.900	3.800	0.522
0.09	0.086	3.088	0.008	7.971	15.200	0.086	1.625	0.713	1.634	3.135	2.432	0.025	6.223	15.200	4.750	0.522
0.15	0.086	3.164	0.009	7.857	15.172	0.057	1.701	0.684	1.577	3.192	2.385	0.021	6.356	23.750	5.700	0.495
0	0.086	3.183	0.008	7.847	15.162	0.057	1.672	0.675	1.473	3.135	2.451	0.019	6.318	19.000	7.600	0.470
0.27	0.086	3.211	0.008	7.762	15.048	0.067	1.672	0.684	1.463	3.211	2.375	0.022	6.261	19.000	8.550	0.456
0.18	0.095	3.221	0.009	7.990	15.229	0.067	1.720	0.713	1.492	3.230	2.423	0.022	6.451	21.850	7.600	0.462
0.13	0.095	3.154	0.009	7.857	15.105	0.057	1.710	0.703	1.568	3.211	2.489	0.024	6.261	22.800	7.600	0.489
0.06	0.086	3.173	0.009	7.885	15.143	0.076	1.710	0.732	1.539	3.202	2.394	0.022	6.375	19.000	4.750	0.481
0.06	0.095	3.135	0.009	7.904	15.105	0.095	1.634	0.713	1.520	3.173	2.423	0.027	6.308	17.100	2.850	0.480

Appendix B: Optimal Number of Estimators

1. High Performance Concrete

2. Energy Performance on Residential Buildings

3. Combined Cycle Power Plant

Summary and Results

Optimal Number of Estimators

1. High Performance Concrete

In this section, an optimal forest sizing for the high performance concrete dataset [104] is determined in order to explore the efficiency in terms of computational cost for better prediction accuracy.

Table B.1: R^2 scores of optimal number of estimators for a high performance concrete dataset

Trees	R^2 Score	$2 \times \sigma$	Trees	R^2 Score	$2 \times \sigma$
10	73.05	(+/- 0.2774)	100	75.32	(+/- 0.2462)
20	74.06	(+/- 0.2740)	140	75.21	(+/- 0.2483)
30	74.22	(+/- 0.2562)	**180**	**75.34**	**(+/- 0.2440)**
40	74.48	(+/- 0.2586)	220	75.02	(+/- 0.2483)
50	74.60	(+/- 0.2611)	260	75.12	(+/- 0.2465)
60	75.06	(+/- 0.2528)	300	75.15	(+/- 0.2463)
70	75.15	(+/- 0.2454)	340	75.32	(+/- 0.2441)
80	75.24	(+/- 0.2483)	380	75.32	(+/- 0.2457)
90	75.25	(+/- 0.2500)	420	75.32	(+/- 0.2442)

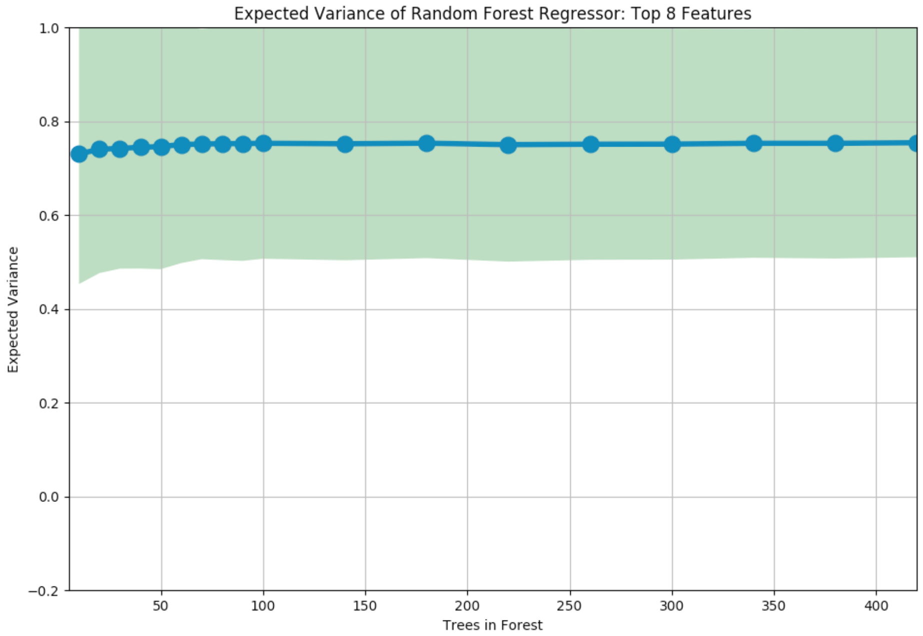


Figure B.1: The scores of different numbers of estimators for the high performance concrete dataset

2. Energy Performance on Residential Buildings

In this section, an optimal forest sizing for the energy performance in residential buildings dataset [105] is determined in order to explore the efficiency in terms of computational cost for better prediction accuracy.

Table B.2: R^2 scores of optimal number of estimators for energy performance on residential buildings dataset

Trees	R^2 Score	$2 \times \sigma$	Trees	R^2 Score	$2 \times \sigma$
10	98.57	(+/- 0.0701)	100	98.58	(+/- 0.0701)
20	98.58	(+/- 0.0701)	140	98.58	(+/- 0.0701)
30	98.58	(+/- 0.0702)	180	98.58	(+/- 0.0701)
40	98.58	(+/- 0.0701)	220	98.58	(+/- 0.0701)
50	98.58	(+/- 0.0701)	260	98.58	(+/- 0.0701)
60	98.58	(+/- 0.0701)	**300**	**98.59**	**(+/- 0.0701)**
70	98.58	(+/- 0.0701)	340	98.58	(+/- 0.0701)
80	98.58	(+/- 0.0701)	380	98.58	(+/- 0.0701)
90	98.58	(+/- 0.0701)	420	98.58	(+/- 0.0701)

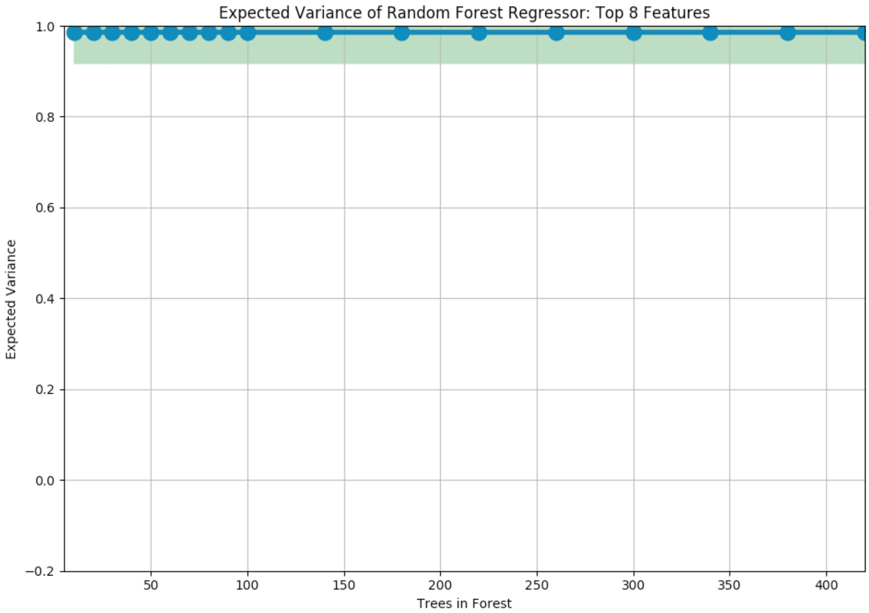


Figure B.2: The scores of different numbers of estimators for the energy performance on residential buildings dataset

3. Combined Cycle Power Plant

In this section, an optimal forest sizing for the combined cycle power plant dataset [106] is determined in order to explore the efficiency in terms of computational cost for better prediction accuracy.

Table B.3: R^2 scores of optimal number of estimators for combined cycle power plant dataset

Trees	R^2 Score	$2 \times \sigma$	Trees	R^2 Score	$2 \times \sigma$
10	95.96	(+/- 0.0125)	100	96.17	(+/- 0.0118)
20	96.06	(+/- 0.0120)	140	96.17	(+/- 0.0117)
30	96.10	(+/- 0.0122)	180	96.17	(+/- 0.0117)
40	96.12	(+/- 0.0122)	220	96.17	(+/- 0.0117)
50	96.14	(+/- 0.0120)	**260**	**96.18**	**(+/- 0.0115)**
60	96.15	(+/- 0.0118)	300	96.18	(+/- 0.0117)
70	96.16	(+/- 0.0117)	340	96.18	(+/- 0.0116)
80	96.16	(+/- 0.0117)	380	96.18	(+/- 0.0116)
90	96.17	(+/- 0.0118)	420	96.18	(+/- 0.0116)

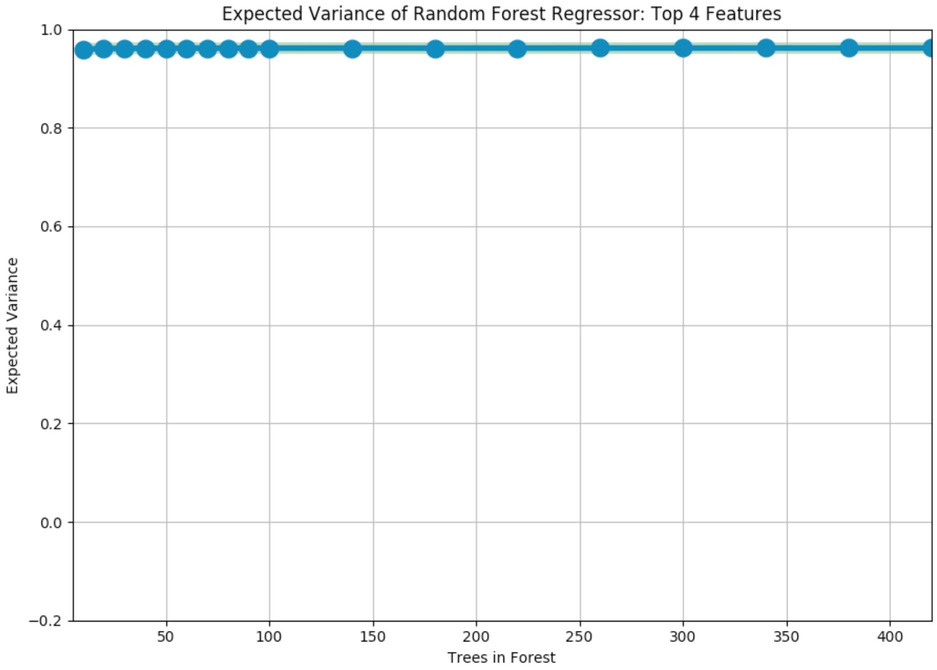


Figure B.3: The scores of different numbers of estimators for the combined cycle power plant dataset

Summary and Results

From the previous tables, the proposed algorithm (MRF) is seen to excel at minimising computational cost in terms of the optimal number of trees. In fact, finding the optimal number of trees will enhance the prediction accuracy with minimum associated computational cost. Therefore, MRF can achieve a higher prediction accuracy at lower computational cost, representing another benefit of this algorithm in addition to its robust predictions. Figure B.4 illustrates the different numbers of estimators for each case study presented in this work.

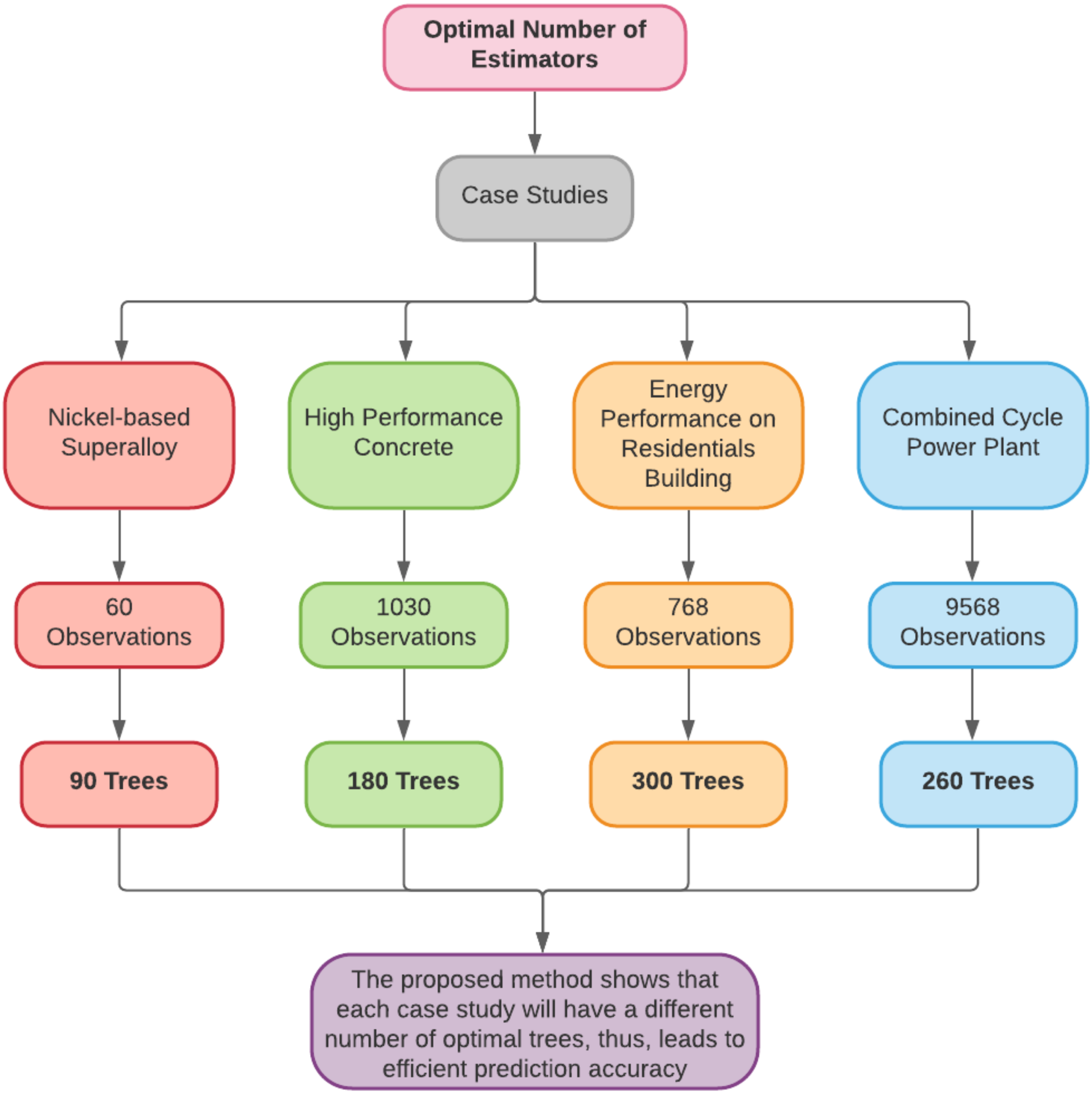

Optimal Number of Estimators
Case Studies
Nickel-based Superalloy
High Performance Concrete
Energy Performance on Residentials Building
Combined Cycle Power Plant
60 Observations
1030 Observations
768 Observations
9568 Observations
90 Trees
180 Trees
300 Trees
260 Trees
The proposed method shows that each case study will have a different number of optimal trees, thus, leads to efficient prediction accuracy

Figure B.4: Optimal number of trees for each case study

Appendix C: QCA Refactoring using Python

1. **Verification on Nickel-based Superalloy Dataset**
2. **Verification on High Performance Concrete Dataset**
3. **Verification on Energy Performance on Residential Buildings**
4. **Verification on Combined Cycle Power Plant Dataset**

QCA Refactoring using Python

1. Verification on Nickel-based Superalloy Dataset

Table C.1: QCA and Prediction simulation table for nickel-based superalloy dataset

QCA		Predication	
Type	LB	B	1000
Th_{min}	0	No of PCs	6
Th_{max}	0.03	Simulations No	100
No of PCs	6	Th_{op}	0.2,0.3,0.4
B	100	n_c	6

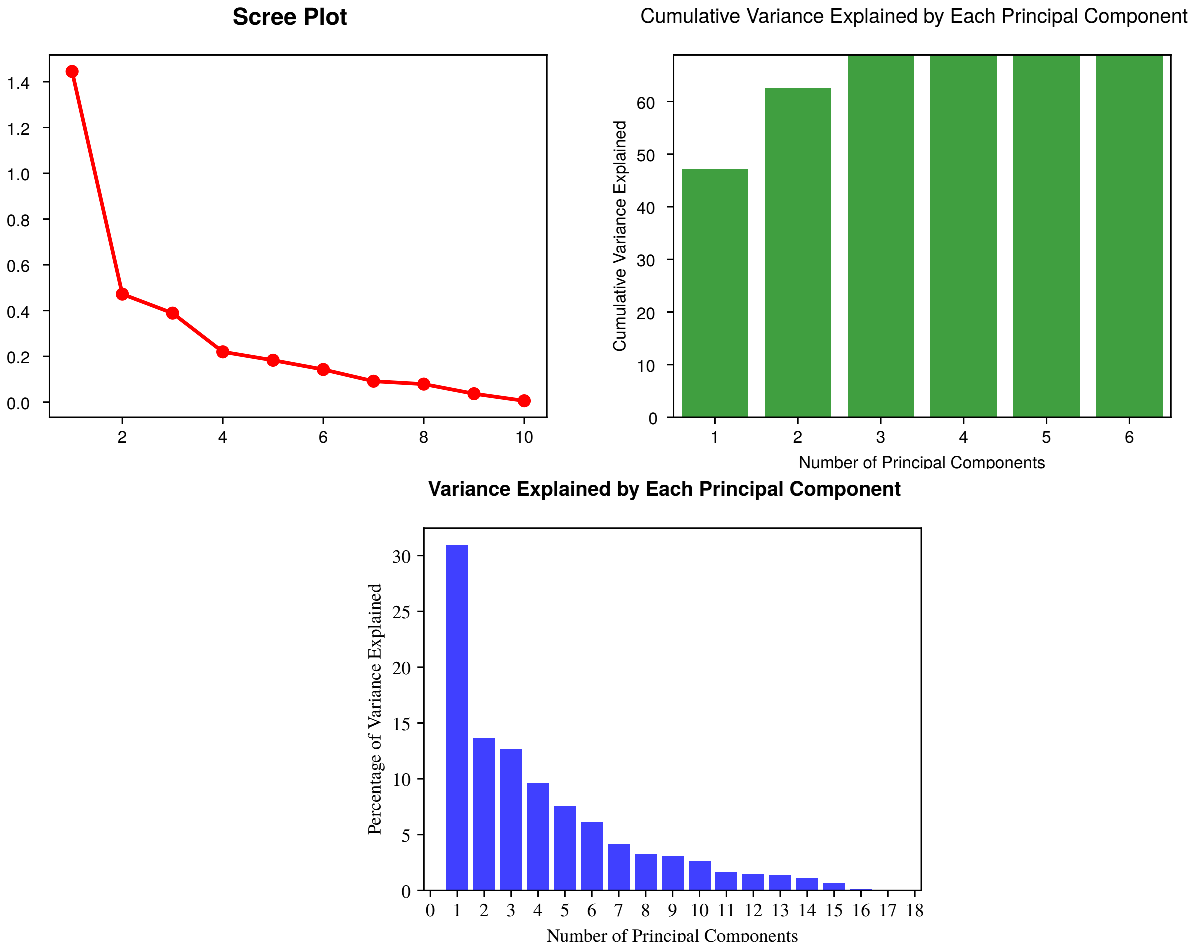


Figure C.1: Scree plot and principal component for nickel-based superalloy dataset

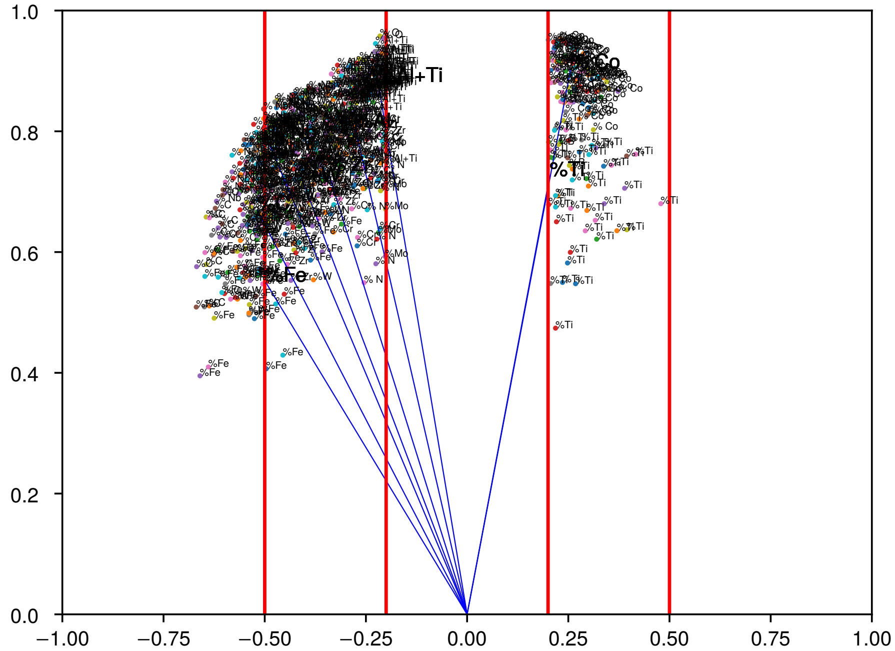


Figure C.2: Co-linearity index plot for nickel-based superalloy dataset

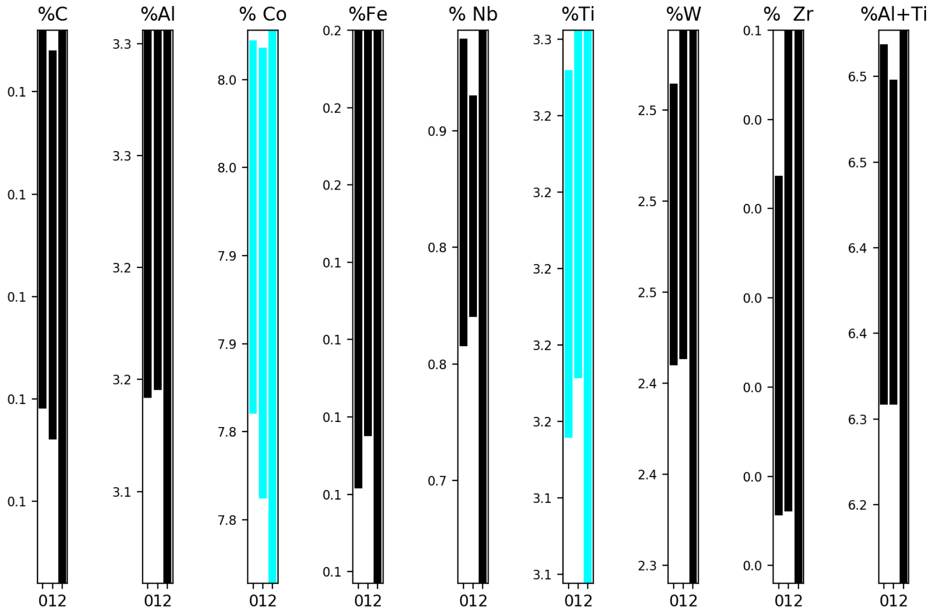


Figure C.3: Tolerance limits for nickel-based superalloy dataset

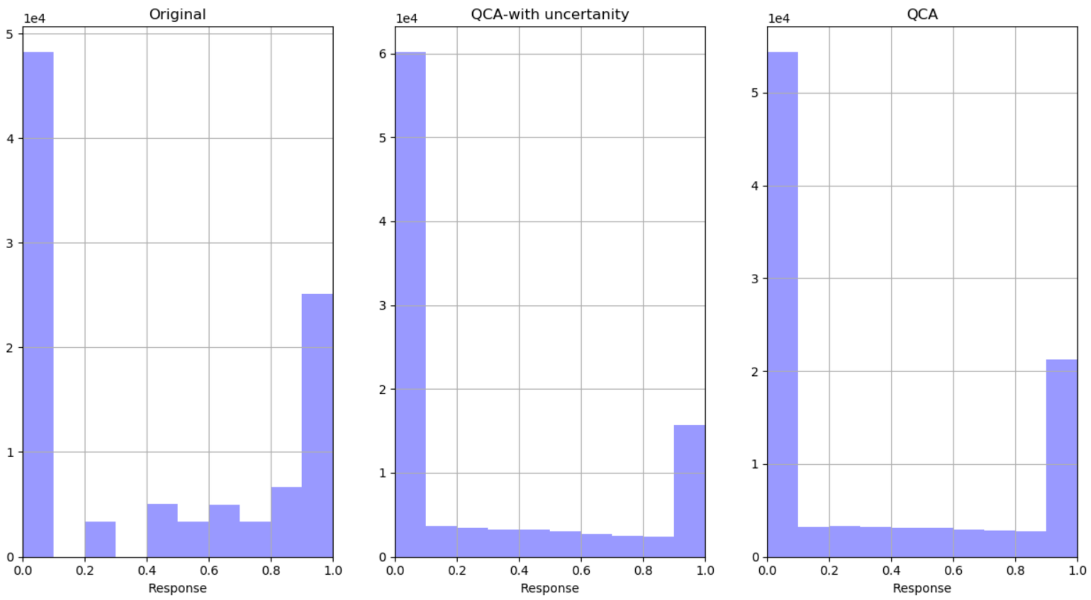


Figure C.4: Predicted response distribution for nickel-based superalloy dataset

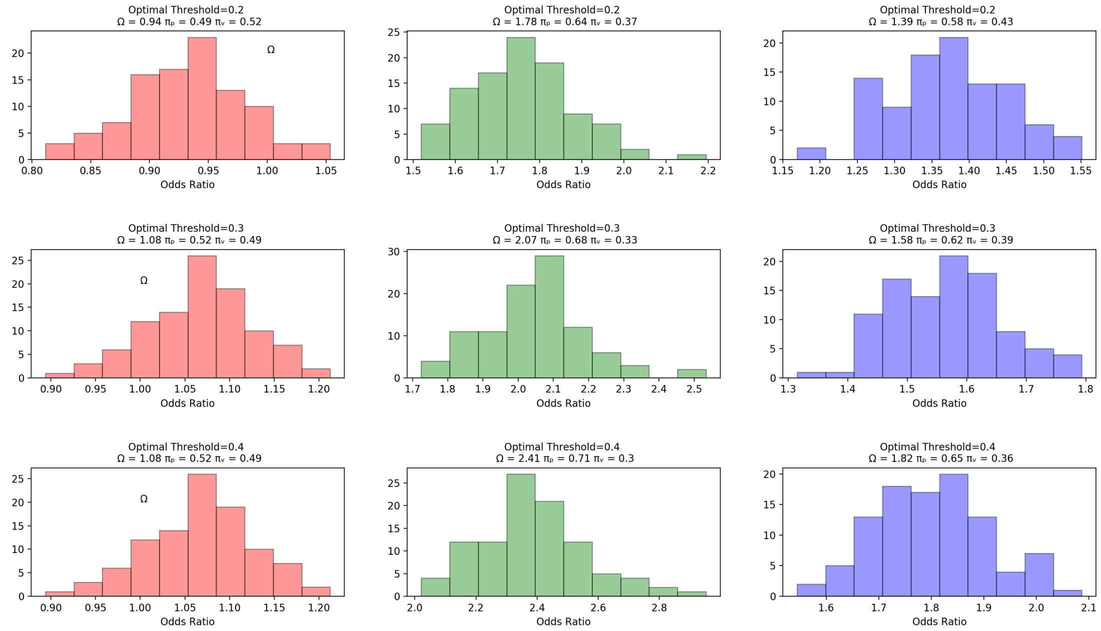


Figure C.5: Odds ratio distribution of all factors for nickel-based superalloy dataset

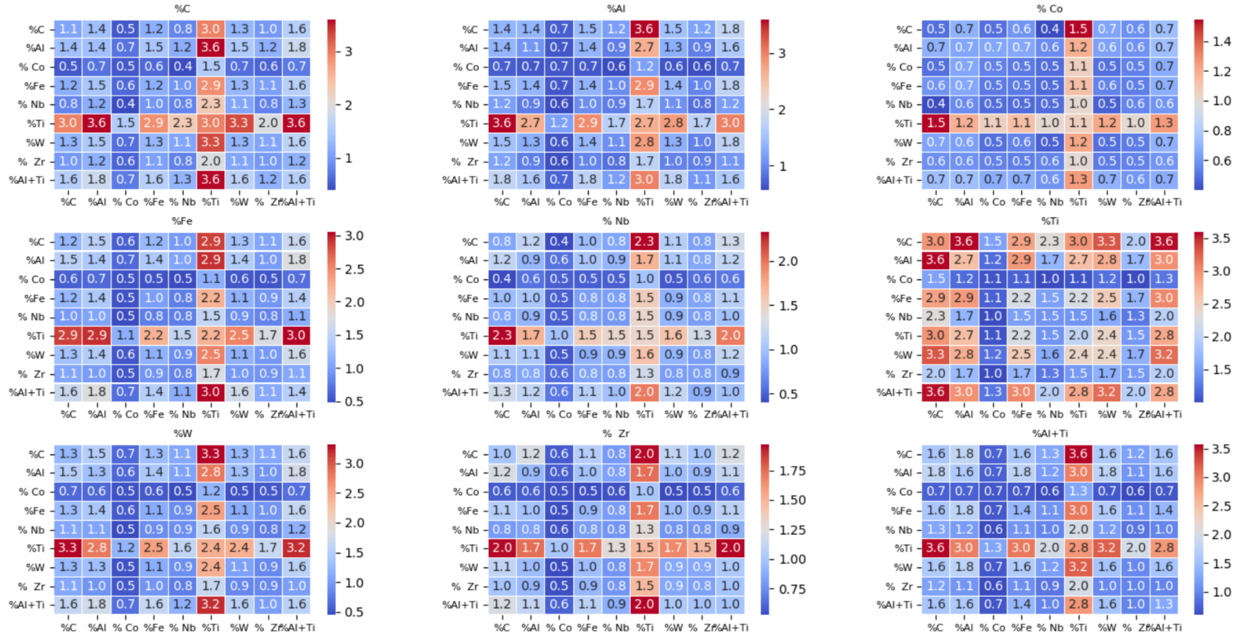


Figure C.6: Odds ratio of interacted factors with optimal limits for nickel-based superalloy dataset

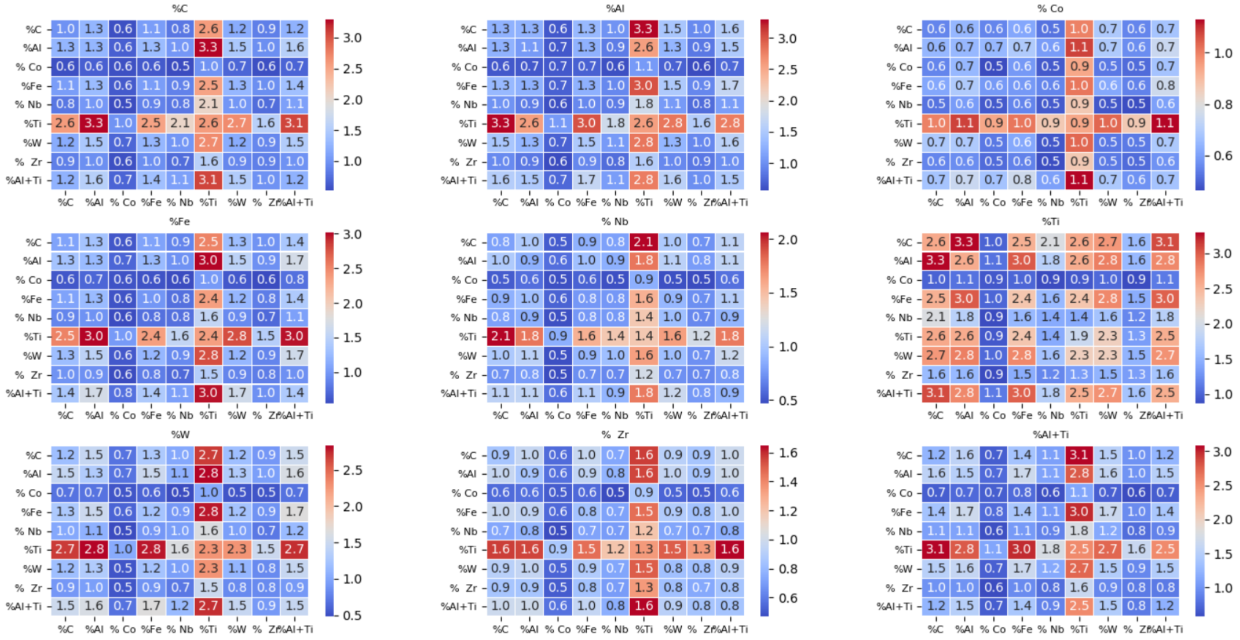


Figure C.7: Odds ratio of interacted factors with uncertainty optimal limits for nickel-based superalloy dataset

2. Verification on High Performance Concrete Dataset

Table C.2: QCA and Prediction simulation table for high performance concrete dataset

QCA		Predication	
Type	HB	B	1000
Th_{min}	20	No of PCs	2
Th_{max}	40	Simulations No	100
No of PCs	3	Th_{op}	0.1,0.2,0.3
B	100	n_c	6

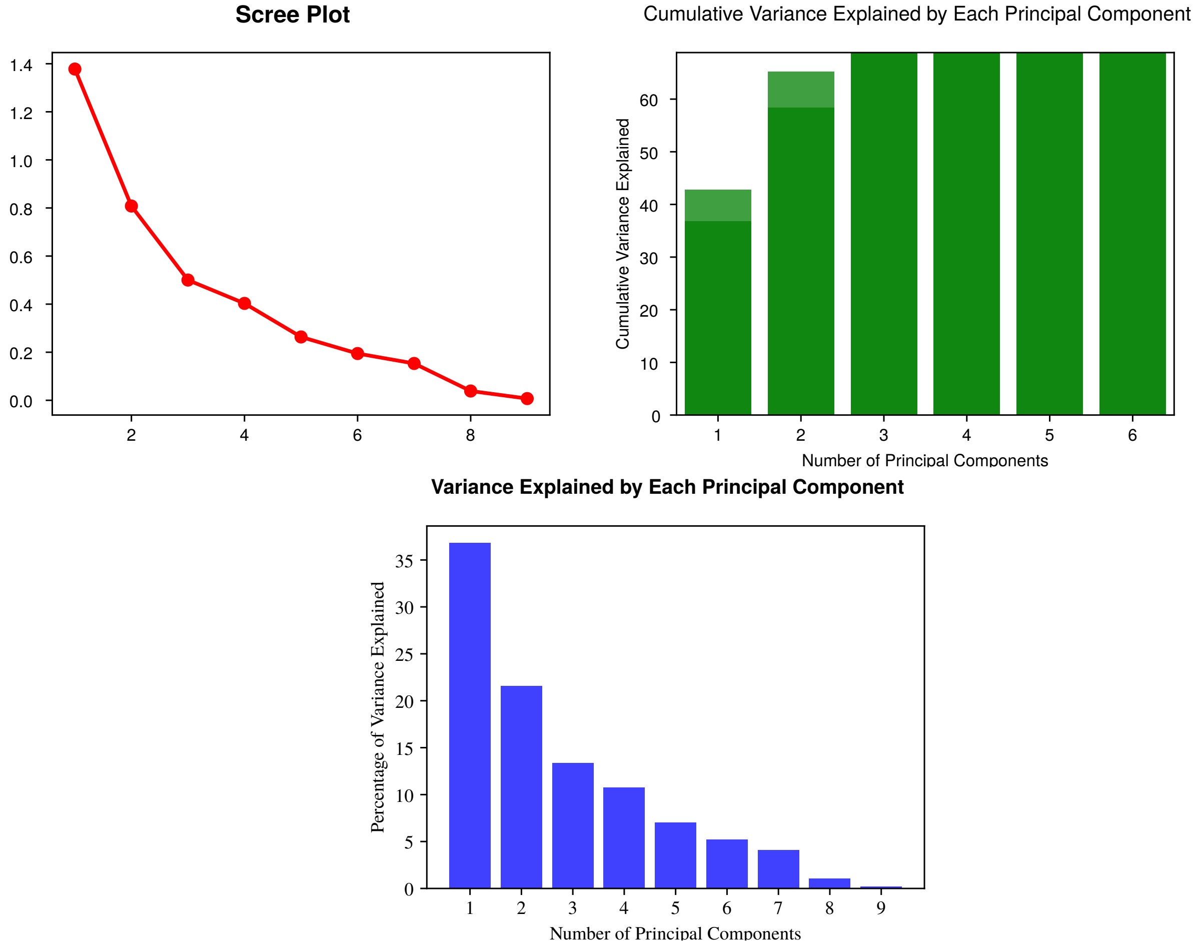


Figure C.8: Scree plot and principal component for high performance concrete dataset

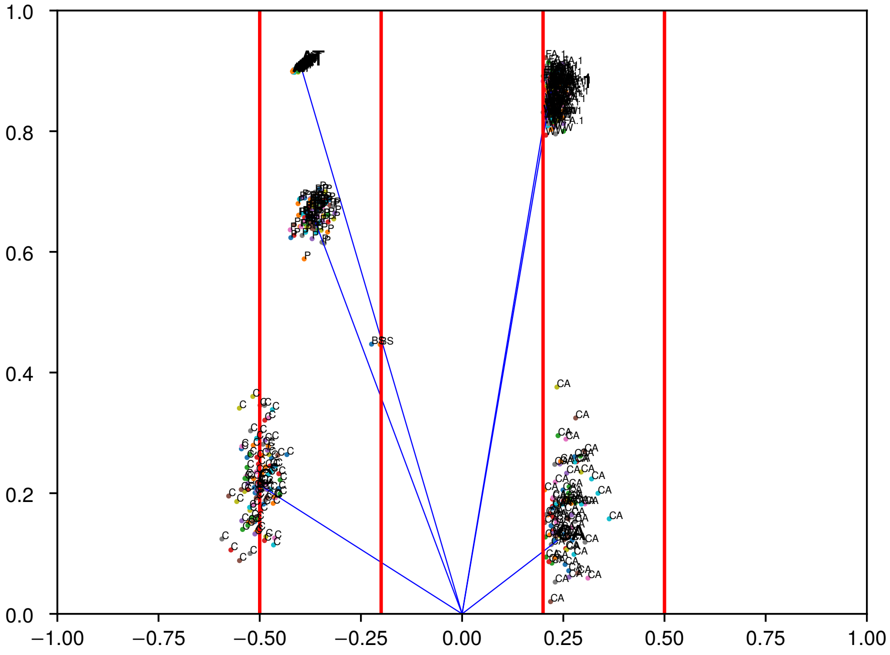


Figure C.9: Co-linearity index plot for high performance concrete dataset

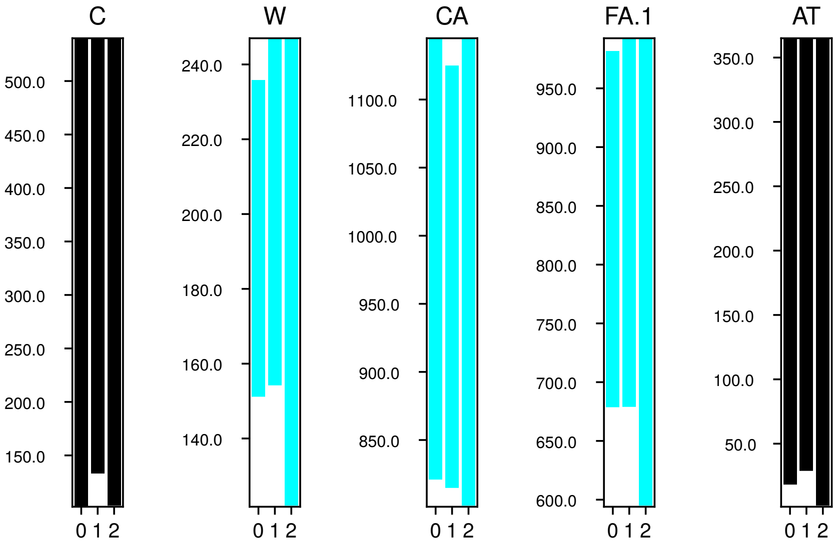


Figure C.10: Tolerance limits for high performance concrete dataset

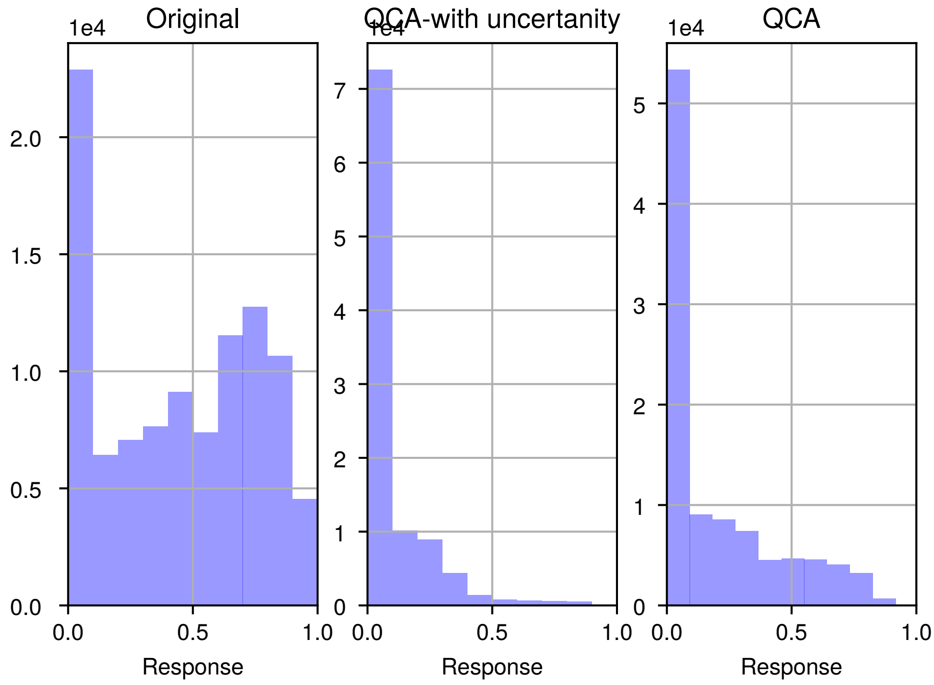


Figure C.11: Predicted response distribution for high performance concrete dataset

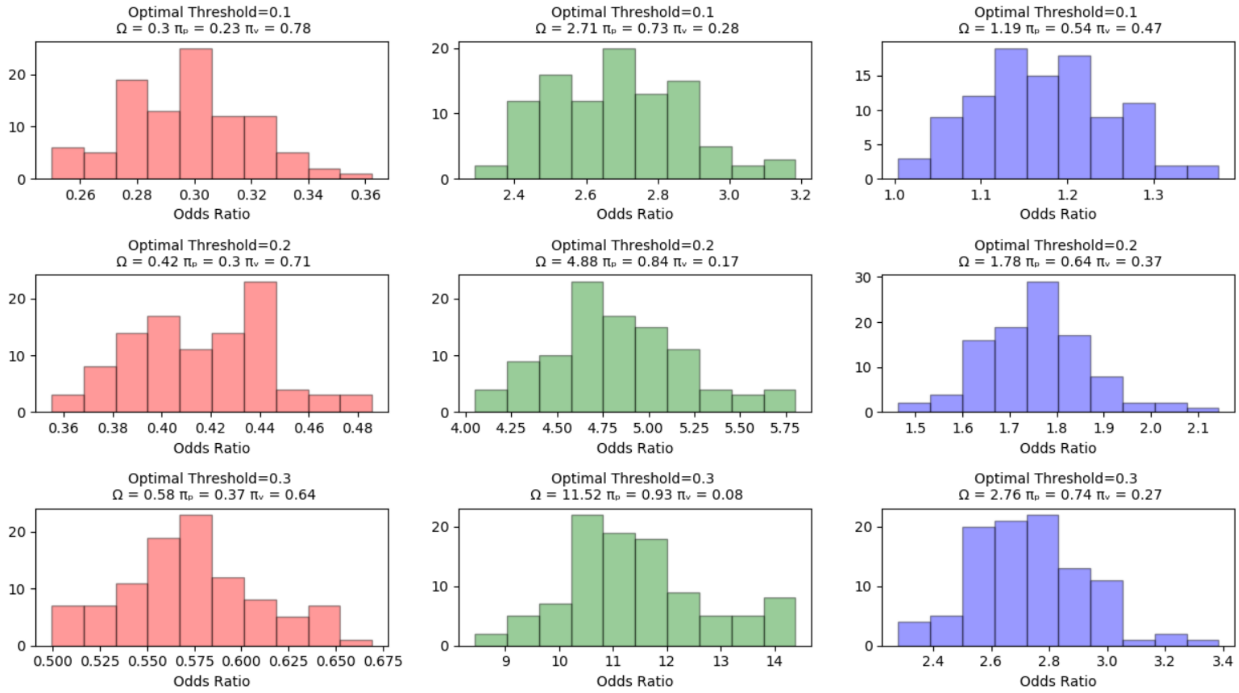


Figure C.12: Odds ratio distribution of all factors for high performance concrete dataset

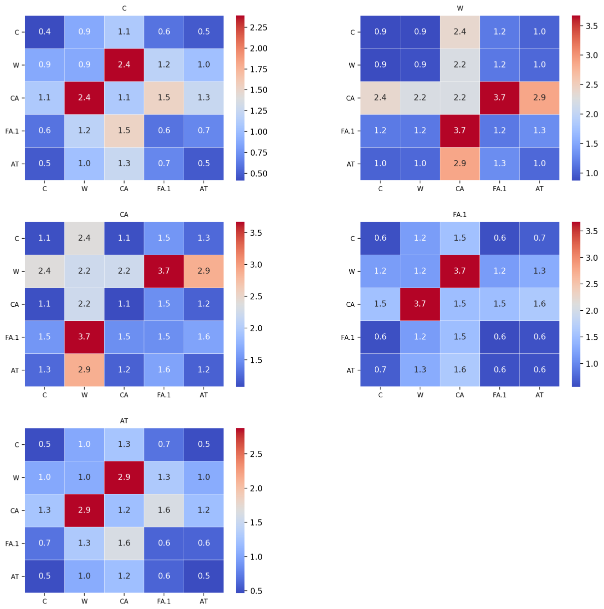


Figure C.13: Odds ratio of interacted factors with optimal limits for high performance concrete dataset

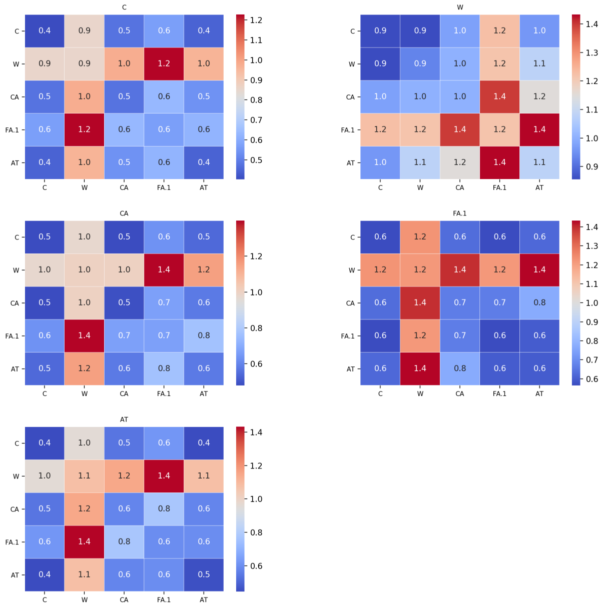


Figure C.14: Odds ratio of interacted factors with uncertainty optimal limits for high performance concrete dataset

3. Verification on Energy Performance of Residential Buildings Dataset

Table C.3: QCA and Prediction simulation table for energy performance of residential buildings dataset

QCA		Predication	
Type	LB	B	1000
Th_{min}	15	No of PCs	1
Th_{max}	35	Simulations No	100
No of PCs	5	Th_{op}	0.1
B	100	n_c	5

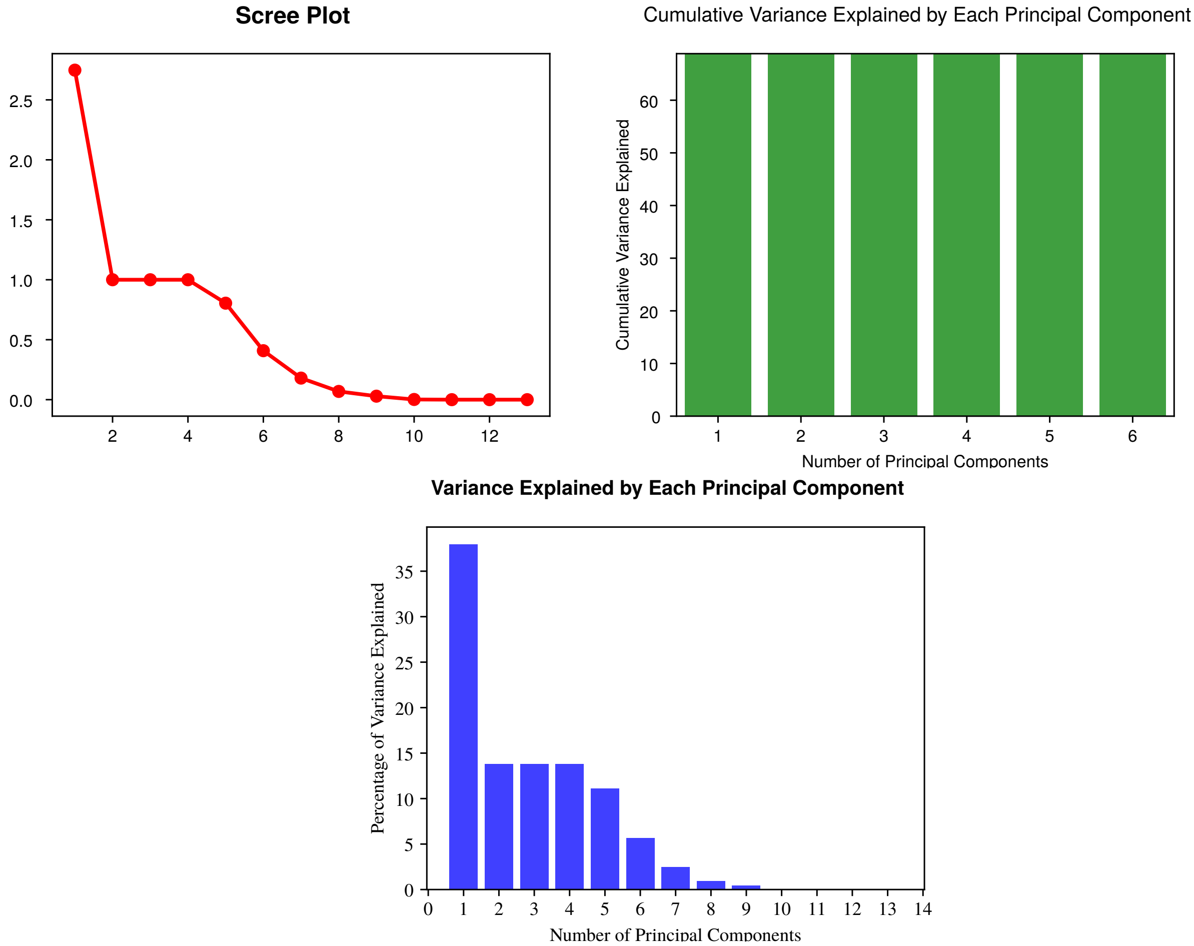


Figure C.15: Scree plot and principal component for energy performance of residential buildings dataset

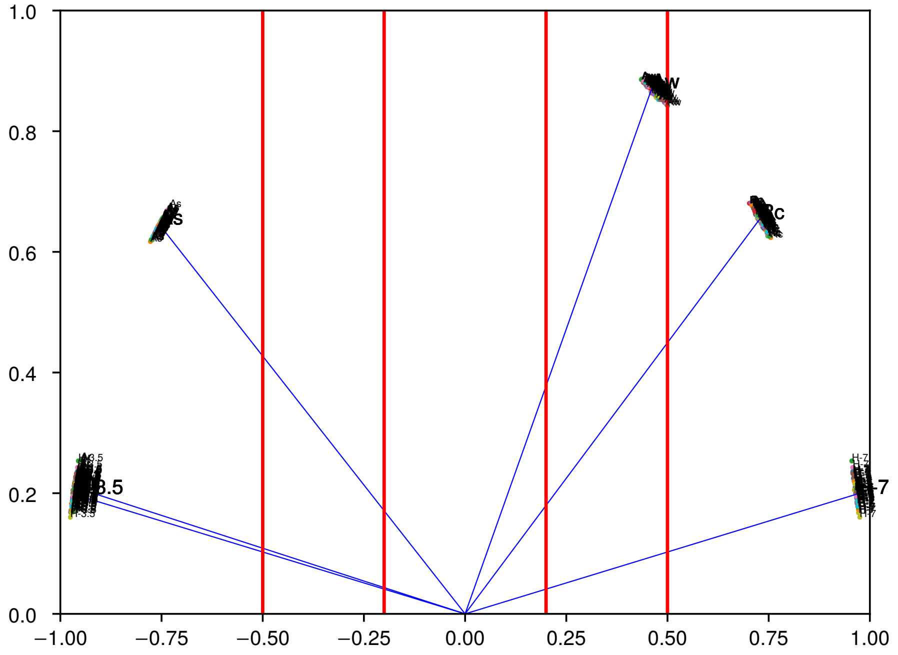


Figure C.16: Co-linearity index plot for energy performance of residential buildings dataset

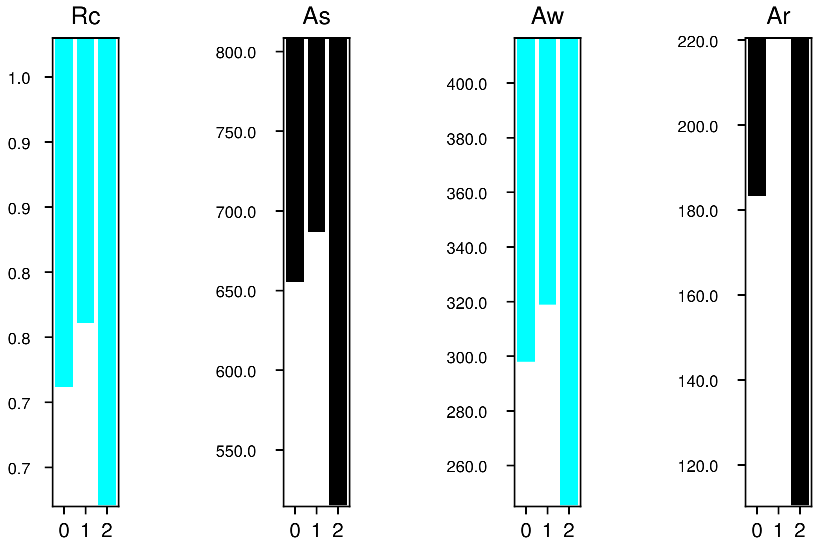


Figure C.17: Tolerance limits for energy performance of residential buildings dataset

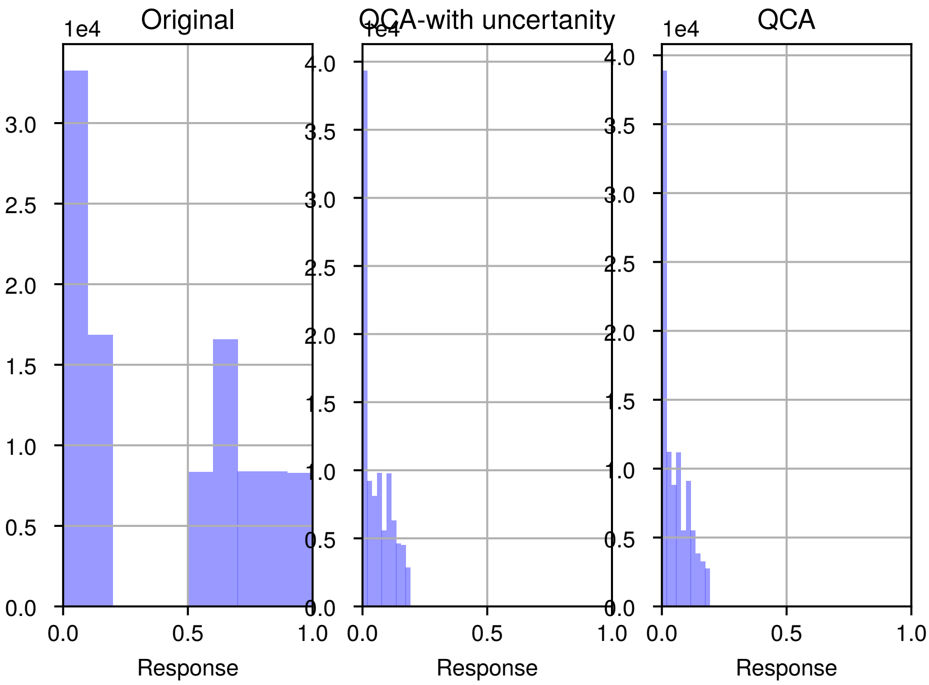


Figure C.18: Predicted response distribution for energy performance of residential buildings dataset

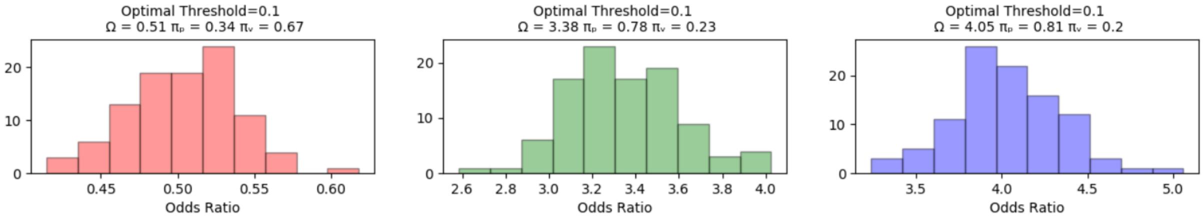


Figure C.19: Odds ratio distribution of all factors for energy performance of residential buildings dataset

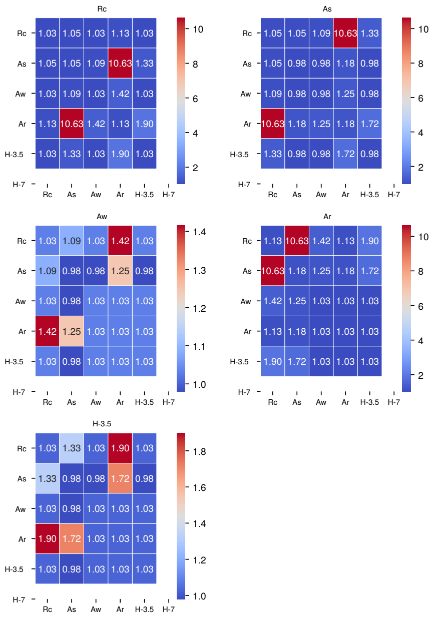


Figure C.20: Odds ratio of interacted factors with optimal limits for energy performance of residential buildings dataset

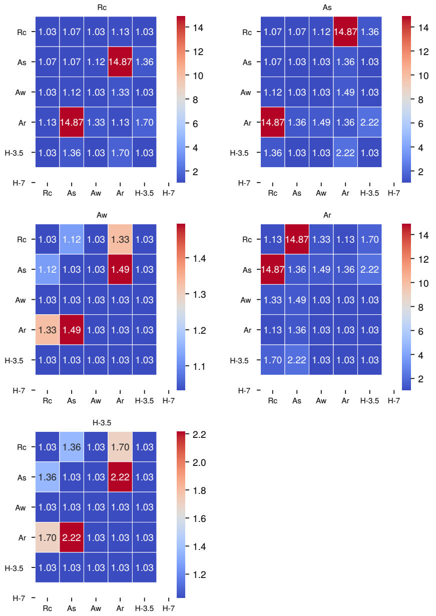


Figure C.21: Odds ratio of interacted factors with uncertainty optimal limits for energy performance of residential buildings dataset

4. Verification on Combined Cycle Power Plant Dataset

Table C.4: QCA and Prediction simulation table for combined cycle power plant dataset

QCA		Predication	
Type	HB	B	10000
Th_{min}	425	No of PCs	2
Th_{max}	470	Simulations No	100
No of PCs	2	Th_{op}	0.1
B	100	n_c	4

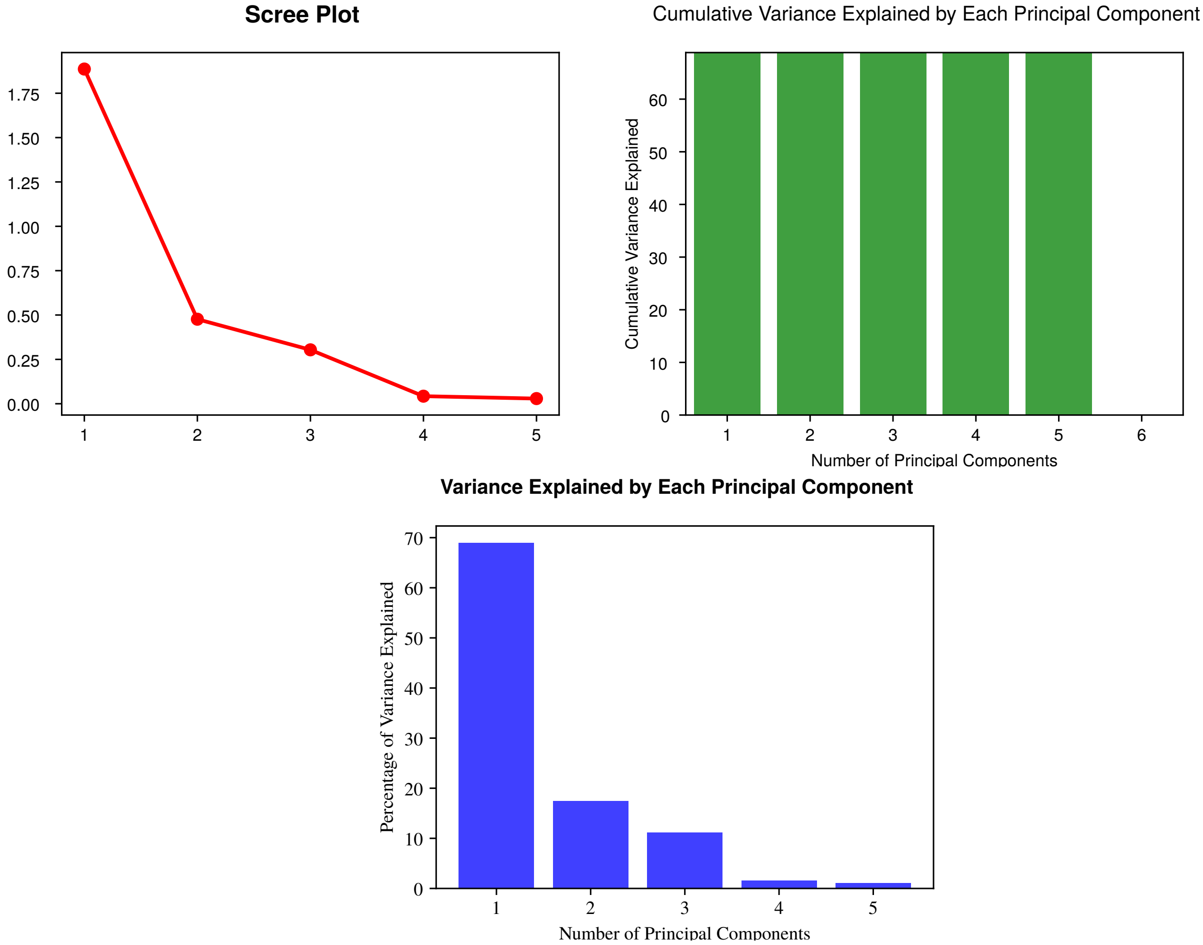


Figure C.22: Scree plot and principal component for combined cycle power plant dataset

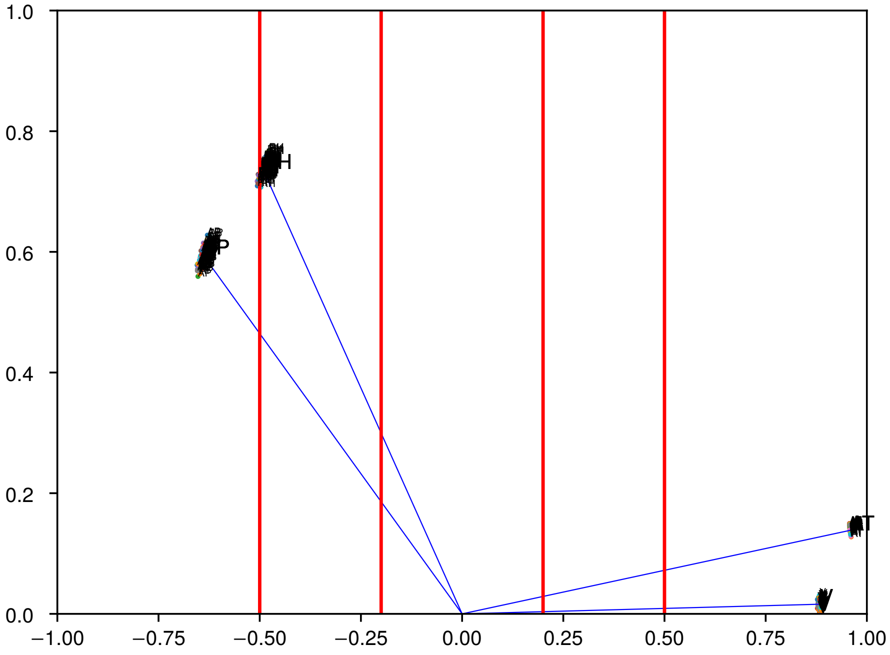


Figure C.23: Co-linearity index plot for combined cycle power plant dataset

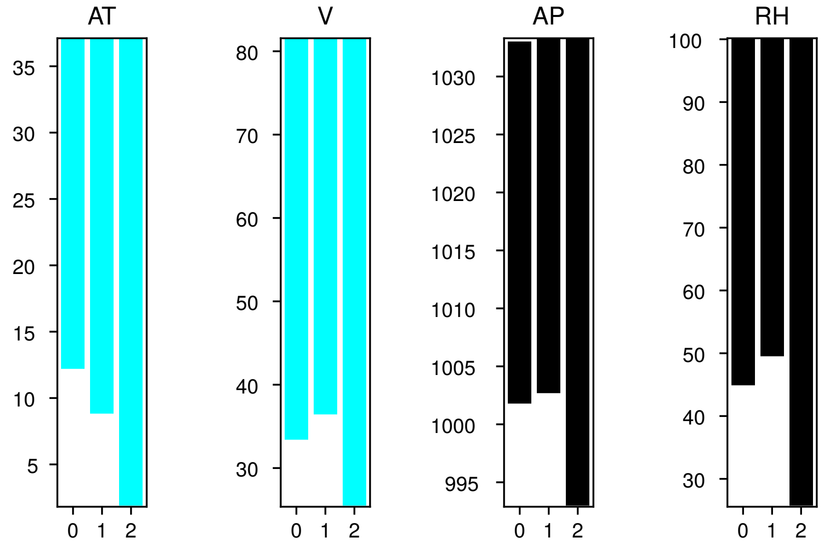


Figure C.24: Tolerance limits for combined cycle power plant dataset

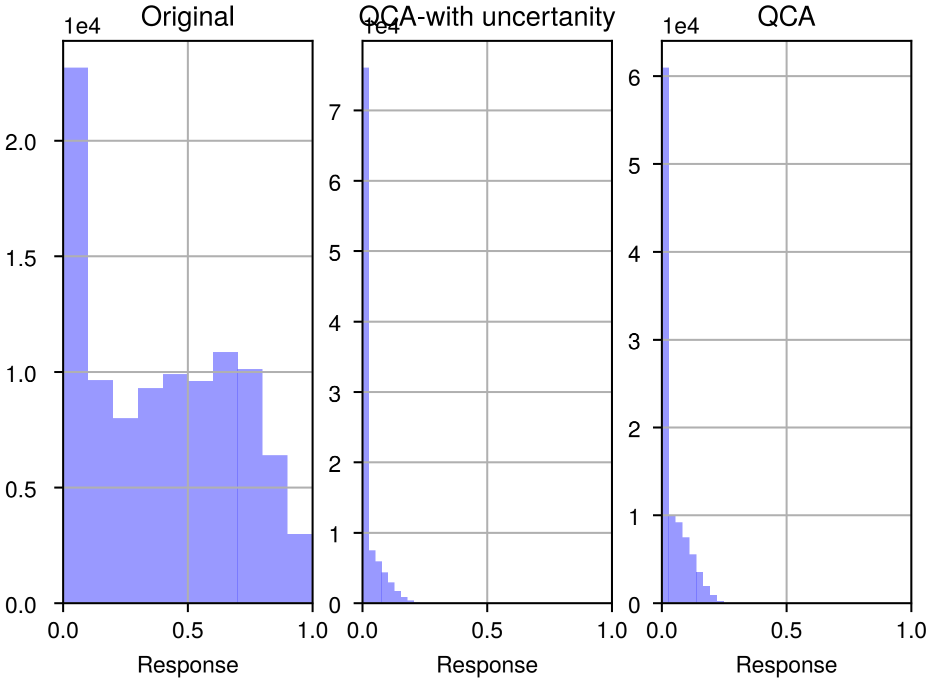


Figure C.25: Predicted response distribution for combined cycle power plant dataset

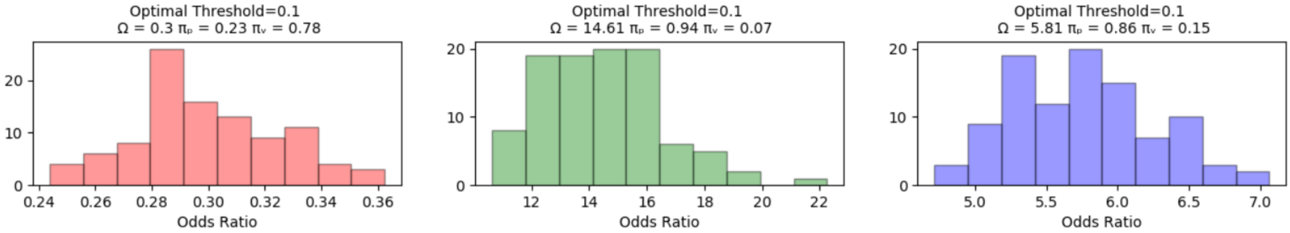


Figure C.26: Odds ratio distribution of all factors for combined cycle power plant dataset

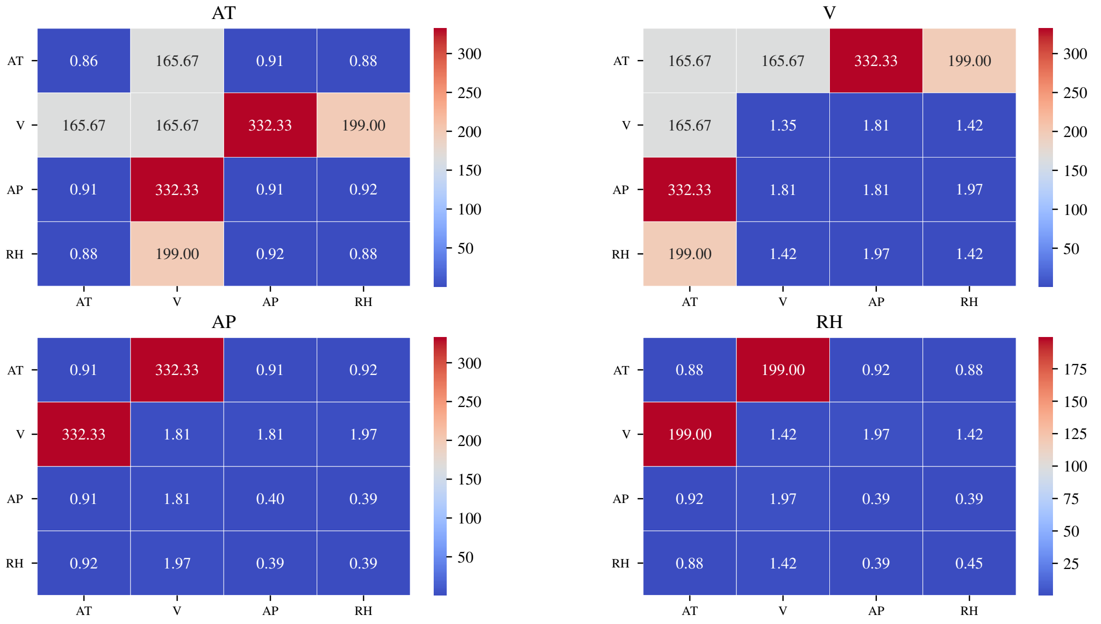


Figure C.27: Odds ratio of interacted factors with optimal limits for combined cycle power plant dataset

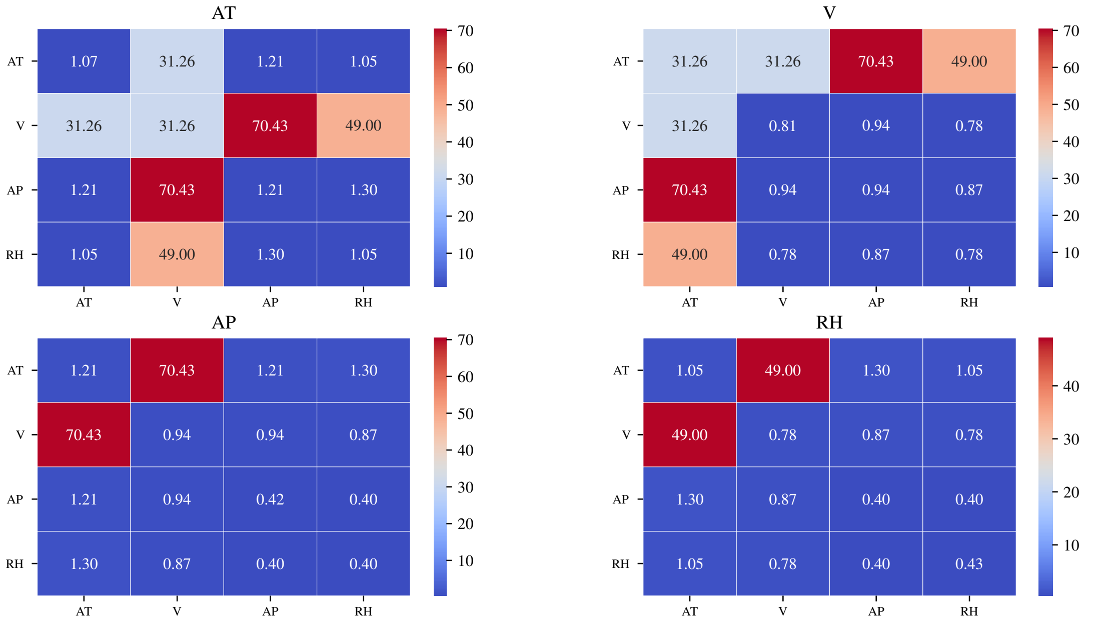


Figure C.28: Odds ratio of interacted factors with uncertainty optimal limits for combined cycle power plant dataset