Modelling of manufacturing processes by a computational intelligence approach

Garg, Akhil

2015

Garg, A. (2015). Modelling of manufacturing processes by a computational intelligence approach. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/62151>

<https://doi.org/10.32657/10356/62151>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**MODELLING OF MANUFACTURING PROCESSES BY A
COMPUTATIONAL INTELLIGENCE APPROACH**

AKHIL GARG

**SCHOOL OF MECHANICAL AND AEROSPACE
ENGINEERING**

2015

**MODELLING OF MANUFACTURING PROCESSES BY A
COMPUTATIONAL INTELLIGENCE APPROACH**

AKHIL GARG

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University in
partial fulfillment of the requirement for the degree of

Doctor of Philosophy

2015

ACKNOWLEDGEMENT

It was indeed a great pleasure to work with Dr. Tai Kang, my supervisor at Nanyang Technological University, towards my PhD research work. His guidance, understanding, support, and friendliness left an incredible mark on me. He was always around whenever I needed him, and helped me focus in the right direction. His motivation and persistence helped me professionally and I am sure it has helped many others as well.

I would like to express a special mention about the financial assistance provided by the University and the funding support by the Singapore Ministry of Education Academic Research Fund through research grant RG 30/10.

I would like to acknowledge Dr. Lily Rachmawati, Lee Chen Hui, Dr. Goh Chi-Keong, Mr. Kelvin Chan, Dr. Partha Dutta and Ms Anna Tai (Rolls-Royce, Singapore) for their useful discussions on this topic.

I wish to thank my mother, father and closed ones who always motivated me and gave me spirit and were thousands of miles away but never made me feel so.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	i
TABLE OF CONTENTS	ii
ABSTRACT	vii
LIST OF FIGURES	ix
LIST OF TABLES	xiii
LIST OF ABBREVIATIONS	xvii
LIST OF PUBLICATIONS	xx
CHAPTER 1 INTRODUCTION	1
1.1. Background and Motivation	1
1.2. Research Objectives	6
1.3. Scope of the Current Research Work	7
1.4. Organization of the thesis	8
CHAPTER 2 LITERATURE REVIEW ON SYSTEMS, MODELING METHODS AND MODELS	9
2.1. Systems, Models and Modeling Methods	9
2.2. Systems and Modelling Method Chosen for the Study	32
CHAPTER 3 COMPUTATIONAL INTELLIGENCE METHODS INCLUDING PROPOSED METHODOLOGY	39
3.1. Artificial neural network	39
3.2. M5' model tree	41
3.3. Support vector regression	43
3.4. Adaptive neuro-fuzzy inference system	45
3.5. Proposed M5'-AMGGP Approach	48

CHAPTER 4 A MODIFIED MULTI-GENE GENETIC PROGRAMMING

MODEL USING STEPWISE APPROACH.....	60
4.1. Methodology	60
4.2. Problem: Turning Process	64
4.2.1. Results and Discussion	70
4.2.1.1 Parameter Settings of CI Methods.....	70
4.2.1.2 Evaluation and Statistical Comparison of CI Methods	73
4.2.1.3 Sensitivity and Parametric Analysis of the Proposed Model	78
4.3. Problem: Turning of DSA regime of ASS 304 steel	81
4.3.1. Results and Discussion	85
4.3.1.1 Parameter Settings of CI Methods.....	85
4.3.1.2 Evaluation and Statistical Comparison of CI Methods	87
4.3.1.3 Sensitivity and Parametric Analysis of the Proposed Model	92
4.4. Summary	95

CHAPTER 5 ORTHOGONAL BASIS FUNCTIONS AS A COMPLEXITY MEASURE FOR MGGP MODELS IN REGULARIZED FITNESS

FUNCTIONS	96
5.1. Methodology	96
5.2. Problem: Machining Processes such as Turning and Drilling	101
5.2.1. Turning process	104
5.2.2. Drilling process	105
5.2.3. Results and Discussion	106
5.2.3.1 Parameter Settings of MGGP	106
5.2.3.2 Evaluation and Comparison of Fitness Functions for Turning process	107
5.2.3.3 Evaluation and Comparison of Fitness Functions for Drilling process	109
5.3. Problem: Vibratory finishing process.....	111

5.3.1. Results and Discussion	113
5.3.1.1 Parameter Settings of MGGP	113
5.3.1.2 Evaluation and Statistical Comparison of Fitness Functions	114
5.4. Summary	116
 CHAPTER 6 CLASSIFICATION-DRIVEN MODEL SELECTION APPROACH OF MULTI-GENE GENETIC PROGRAMMING	 117
6.1. Methodology	117
6.2. Problem: Fused deposition modeling process (FDM).....	119
6.2.1. Results and Discussion	124
6.2.1.1 Parameter Settings of CI Methods.....	124
6.2.1.2 Evaluation and Statistical Comparison of CI Methods	127
6.2.1.3 Sensitivity and Parametric Analysis of the Proposed Model	130
6.3. Problem: Turning process of AISI 1040 Steel	132
6.3.1. Results and Discussion	134
6.3.1.1 Parameter Settings of CI Methods.....	134
6.3.1.2 Evaluation and Statistical Comparison of CI Methods	138
6.3.1.3 Sensitivity and Parametric Analysis of the Proposed Model	143
6.4. Summary	145
 CHAPTER 7 HYBRID APPROACH OF MULTI-GENE GENETIC PROGRAMMING FOR IMPROVING TRUSTWORTHINESS OF PREDICTION ABILITY OF MODEL ON UNSEEN SAMPLES	 146
7.1. Methodology	146
7.2. Problem: Vibratory finishing process.....	148
7.2.1. Results and Discussion	149
7.2.1.1 Parameter Settings of CI Methods.....	149
7.2.1.2 Evaluation and Statistical Comparison of CI Methods	152
7.2.1.3 Sensitivity and Parametric Analysis of the Proposed Model	157

7.3. Problem: Fused deposition modeling process (FDM).....	163
7.3.1. Results and Discussion	164
7.3.1.1 Parameter Settings of CI Methods.....	164
7.3.1.2 Evaluation and Statistical Comparison of CI Methods	167
7.3.1.3 Sensitivity and Parametric Analysis of the Proposed Model	170
7.4. Summary	172

CHAPTER 8 HYBRID APPROACH OF ADVANCED MULTI-GENE GENETIC PROGRAMMING WITH ANOTHER COMPUTATIONAL INTELLIGENCE METHOD

8.1. Methodology	173
8.2. Problem: Turning Process of AISI H11 Steel	175
8.2.1. Results and Discussion	175
8.2.1.1 Parameter Settings of CI Methods.....	175
8.2.1.2 Evaluation and Statistical Comparison of CI Methods	177
8.3. Problem: Turning of ASS 304 Steel Subjected to DSA Regime	179
8.3.1. Results and Discussion	181
8.3.1.1 Parameter Settings of CI Methods.....	181
8.3.1.2 Evaluation and Statistical Comparison of CI Methods	184
8.4. Problem: Fused deposition modeling process (FDM).....	187
8.4.1. Results and Discussion	187
8.4.1.1 Parameter Settings of CI Methods.....	187
8.4.1.2 Evaluation and Statistical Comparison of CI Methods	189
8.5. Problem: Vibratory finishing process.....	191
8.5.1. Results and Discussion	192
8.5.1.1 Parameter Settings of CI Methods.....	192
8.5.1.2 Evaluation and Statistical Comparison of CI Methods	194
8.6. Summary	200

CHAPTER 9 CONCLUDING REMARKS AND FUTURE WORK.....	201
9.1. Concluding Remarks	201
9.2. Original Contributions Arising from the Work	203
9.3. Recommendation for Future Work.....	206
 REFERENCES.....	 208

ABSTRACT

Modelling is a term widely used in System Identification (SI), which is referred to as the art and science of building mathematical models of systems using some measured data. The systems of interest in this thesis are additive manufacturing processes such as fused deposition modelling, machining processes such as turning, and finishing processes such as vibratory finishing. These processes comprise multiple input and output variables, making their operating mechanisms complex. In addition, it can be costly to obtain the process data and therefore there is a strong need for effective and efficient ways of modelling these systems. The models developed for a system can help to reveal hidden information such as the dominant input variables and their appropriate settings for operating the system in an optimal way. The models formulated must not only predict the values of output variables accurately on the testing samples but should also be able to capture the dynamics of the systems. This is known as a generalization problem in modelling. The generalization of data obtained from manufacturing systems is a capability highly demanded by the industry.

Several modelling methods and types of models were studied by classifying SI in different ways, such as (1) black box, grey box and white box, (2) parametric and non-parametric, and (3) linear SI, non-linear SI and evolutionary SI. A study of the literature also reveals that extensive focus has been paid to computational intelligence (CI) methods such as genetic programming (GP), M5', adaptive neuro fuzzy inference system (ANFIS), artificial neural network (ANN), support vector regression (SVR), etc. for modelling the output variables of the systems because of their ability to formulate the models based only on data obtained from the system. It was also learned that by

embedding the features of several methods from different fields of SI into a given method, it is possible to improve its generalization ability. Popular variants of GP such as multi-gene genetic programming (MGGP), which evolves the model structure and its coefficients automatically, has been applied extensively. However, the full potential of MGGP has not been achieved due to some shortcomings leading to its poor generalization ability.

In the present work, four variants/methods of MGGP are proposed to counter the four shortcomings identified, namely (1) inappropriate procedure of formulation of the MGGP model, (2) inappropriate complexity measure of the MGGP model, (3) difficulty in model selection, and (4) ensuring greater trustworthiness of prediction ability of the model on unseen samples. A robust CI approach was also developed by applying these four variants of MGGP and the M5' method in parallel. These methods are applied in modelling of output variables of various manufacturing systems such as turning, fused deposition modelling and vibratory finishing process. The performance is compared to those of the other methods such as MGGP, SVR, ANFIS and ANN. The statistical comparison conducted reveals that the generalization ability achieved from the four variants of MGGP and robust CI approach is better than those of the other methods. Furthermore, the sensitivity and parametric analysis conducted validates the robustness of the proposed models by unveiling the dominant input variables and hidden non-linear relationships.

LIST OF FIGURES

Fig. 2.1: Illustration of literature review on SI	10
Fig. 2.2: Step-by-Step procedure to formulate models using given data.....	12
Fig. 2.3: Illustration of model selection problem in GP	37
Fig. 3.1: Feed forward neural network of single layer.....	40
Fig. 3.2: Architecture of ANN	41
Fig. 3.3: M5' model tree: Models numbered 1-6 are linear regression models	42
Fig. 3.4: Architecture of SVM	44
Fig. 3.5: Architecture of ANFIS	46
Fig. 3.6: Flowchart showing the mechanism of MGGP	50
Fig. 3.7: Example of the gene $16 + \tan(x) - \left(\frac{8}{y}\right)$	51
Fig. 3.8: Formulation of the MGGP model using OLS method	51
Fig. 3.9: Subtree crossover operation	54
Fig. 3.9: Subtree mutation operation.....	54
Fig. 3.11: Development of AMGGP method.....	56
Fig. 3.12: GUI for implementation of MGGP and AMGGP	57
Fig. 3.13: Formulation of CI approach M5'-AMGGP	58
Fig. 4.1: Step-by-Step implementation of M-MGGP approach.....	63
Fig. 4.2: M-MGGP model formulation using the stepwise regression approach	64
Fig. 4.3: GUI showing the parameter settings chosen for MGGP and M-MGGP	70
Fig. 4.4: RMSE obtained by ANN models by varying the number of neurons in the hidden layer	72
Fig. 4.5: Architecture of ANN determined based on trial-and-error approach.....	72
Fig. 4.6: Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data.....	75
Fig. 4.7: Comparison of simulated M-MGGP model and experimental values on (a) training and (b) testing data.....	76
Fig. 4.8: Comparison of simulated SVR model and experimental values on (a) training and (b) testing data.....	76

Fig. 4.9: Comparison of simulated ANN model and experimental values on (a) training and (b) testing data.....	77
Fig. 4.10: Bar graph showing complexity of models evolved using MGGP method.....	78
Fig. 4.11: Bar graph showing complexity of models evolved using M-MGGP method.....	78
Fig. 4.12: Relative contribution of each input variable to surface roughness.....	80
Fig. 4.13: Variation of surface roughness with respect to each input variable.....	80
Fig. 4.14: GUI showing the parameter settings chosen for MGGP and M-MGGP	85
Fig. 4.15: RMSE obtained by ANN models by varying the number of neurons in the hidden layer.....	87
Fig. 4.16: Architecture of ANN determined based on trial-and-error approach.....	87
Fig. 4.17: Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data.....	89
Fig. 4.18: Comparison of simulated M-MGGP model and experimental values on (a) training and (b) testing data.....	90
Fig. 4.19: Comparison of simulated SVR model and experimental values on (a) training and (b) testing data.....	90
Fig. 4.20: Comparison of simulated ANN model and experimental values on (a) training and (b) testing data.....	91
Fig. 4.21: Bar graph showing complexity of models evolved using MGGP method.....	92
Fig. 4.22: Bar graph showing complexity of models evolved using M-MGGP method.....	92
Fig. 4.23: Relative contribution of each input variable to true stress	93
Fig. 4.24: Variation of true stress with respect to each input variable.....	94
Fig. 5.1: Formulation of new MGGP approach by using new fitness function	101
Fig. 5.2: GUI showing the parameter settings chosen for MGGP and M-MGGP	119
Fig. 5.3: GUI showing the parameter settings chosen for MGGP and M-MGGP	125
Fig. 6.1: Schematic flowchart of the C-MGGP methodology showing the classification methods (inside dashed line).....	119
Fig. 6.2: GUI showing the parameter settings chosen for MGGP and C-MGGP	125
Fig. 6.3: Comparison of simulated MGGP model and experimental values on (a) training (b) validation data and (c) testing data.....	128

Fig. 6.4: Comparison of simulated C-MGGP model and experimental values on (a) training (b) validation data and (c) testing data.....	129
Fig. 6.5: Relative contribution of each input variable to compressive strength	131
Fig. 6.6: Variation of compressive strength with respect to each input variable	131
Fig. 6.7: GUI showing the parameter settings chosen for MGGP and C-MGGP.....	134
Fig. 6.8: RMSE obtained by ANN models by varying the number of neurons in the hidden layer	137
Fig. 6.9: Architecture of ANN determined based on trial-and-error approach.....	137
Fig. 6.10: Comparison of simulated MGGP model and experimental values on (a) training (b) validation data and (c) testing data.....	139
Fig. 6.11: Comparison of simulated C-MGGP model and experimental values on (a) training (b) validation data and (c) testing data.....	140
Fig. 6.12: Comparison of simulated SVR model and experimental values on (a) training (b) validation data and (c) testing data.....	141
Fig. 6.13: Comparison of simulated ANN model and experimental values on (a) training (b) validation data and (c) testing data.....	142
Fig. 6.14: Relative contribution of each input variable to surface roughness.....	144
Fig. 6.15: Variation of surface roughness with respect to each input variable	144
Fig. 7.1: Hybrid M1-M2 approach	146
Fig. 7.2: GUI showing the parameter settings chosen for MGGP and MGGP-ANN.....	150
Fig. 7.3: Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data for the output H	153
Fig. 7.4: Comparison of simulated MGGP-ANN model and experimental values on (a) training and (b) testing data for the output H	154
Fig. 7.5: Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data for the output E	154
Fig. 7.6: Comparison of simulated MGGP-ANN model and experimental values on (a) training and (b) testing data for the output E	155
Fig. 7.7: Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data for the output S	155
Fig. 7.8: Comparison of simulated MGGP-ANN model and experimental values on	

(a) training and (b) testing data for the output S	156
Fig. 7.9: Relative contribution of each input variable to projection height reduction, edge radius and surface finish reduction respectively	159
Fig. 7.10: Variation of output H with respect to each input variable.....	160
Fig. 7.11: Variation of output E with respect to each input variable	161
Fig. 7.12: Variation of output S with respect to each input variable	162
Fig. 7.13: GUI showing the parameter settings chosen for MGGP and M5'-MGGP	165
Fig. 7.14: FIS rules for FDM modelling using ANFIS	166
Fig. 7.15: Comparison of simulated M5'-MGGP model and experimental values on (a) training and (b) testing data	168
Fig. 7.16: Comparison of simulated ANFIS model and experimental values on (a) training and (b) testing data	168
Fig. 7.17: Comparison of simulated SVR model and experimental values on (a) training and (b) testing data	169
Fig. 7.18: Relative contribution of each input variable to compressive strength	171
Fig. 7.19: Variation of compressive strength with respect to each input variable	171
Fig. 8.1: Formulation of CI approach M5'-AMGPP	174
Fig. 8.2: GUI showing the parameter settings chosen for M5'-AMGPP	176
Fig. 8.3: GUI showing the parameter settings chosen for M5'-AMGPP	181
Fig. 8.4: GUI showing the parameter settings chosen for M5'-AMGPP	188
Fig. 8.5: GUI showing the parameter settings chosen for M5'-AMGPP	192

LIST OF TABLES

Table 2.1: Classification of SI into three categories	14
Table 2.2: Classification of SI into two categories	15
Table 2.3: Classification of SI into various fields.....	17
Table 2.4: Classification of SI into various fields.....	26
Table 4.1: AISI H11 steel composition.....	68
Table 4.2: Input variables used in turning process	68
Table 4.3: Experiments showing values of input process variables and surface roughness.....	69
Table 4.4: Parameter settings for the three-layer ANN	72
Table 4.5: Multi-objective error of the four models	77
Table 4.6: Descriptive statistics for relative error (%) of the four models	77
Table 4.7: Hypothesis testing to compare the four prediction models	77
Table 4.8: ASS 304 composition	84
Table 4.9: Input variables used for tensile test of ASS 304 operated at DSA regime	84
Table 4.10: Descriptive statistics of the input and output process variables considered for tensile testing	84
Table 4.11: Parameter settings for the three-layer ANN	86
Table 4.12: Multi-objective error of the four models	91
Table 4.13: Descriptive statistics for relative error (%) of the four models	91
Table 4.14: Hypothesis testing to compare the four prediction models	91
Table 5.1: Parameter settings of MARS	100
Table 5.2: Fitness functions and their mathematical formulae	100
Table 5.3: Summary of applications of CI methods in modelling of machining processes ...	103
Table 5.4: Process input variables of drilling process and their respective values.....	106
Table 5.5: Descriptive statistics of the process variables used in drilling process	106
Table 5.6: Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the MGGP model with minimum training error.....	108

Table 5.7: Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the top 10% MGGP models	109
Table 5.8: Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the MGGP model with minimum training error	110
Table 5.9: Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the top 10% MGGP models	110
Table 5.10: Descriptive statistics of the data set generated using FF experimental design	113
Table 5.11: Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the MGGP model with minimum training error	115
Table 5.12: Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the top 10% MGGP models	115
Table 6.1: Input variables of FDM process and their respective values	123
Table 6.2: Descriptive statistics of the input and output process variables of FDM process .	124
Table 6.3: Class of C-MGGP models classified by three classifiers	125
Table 6.4: Multi-objective error of the two models	129
Table 6.5: Descriptive statistics for relative error (%) of the two models	129
Table 6.6: Hypothesis testing to compare the two prediction models	130
Table 6.7: Input variables of turning process with their low-centre-high values	133
Table 6.8: Cutting tool geometry variables and average measured surface roughness values	133
Table 6.9: Class (best or bad) of C-MGGP models predicted by four classification methods	135
Table 6.10: Parameter settings for the three-layer ANN	137
Table 6.11: Multi-objective error of the four models	142
Table 6.12: Descriptive statistics for relative error (%) of the four models	142
Table 6.13: Hypothesis testing to compare the four prediction models	143

Table 7.1: Descriptive statistics of the input and output process variables	149
Table 7.2: Parameter settings for ANN.....	150
Table 7.3: Multi-objective error of the two models on the three outputs	156
Table 7.4: Descriptive statistics for relative error (%) of the two models for output H	156
Table 7.5: Descriptive statistics for relative error (%) of the two models for output E.....	156
Table 7.6: Descriptive statistics for relative error (%) of the two models for output S.....	157
Table 7.7: Hypothesis testing to compare the two prediction models on the three outputs ...	157
Table 7.8: Parameter settings for M5'	164
Table 7.9: Multi-objective error of the three models	169
Table 7.10: Descriptive statistics for relative error (%) of the three models	169
Table 7.11: Hypothesis testing to compare the three prediction models	169
Table 8.1: Parameter settings for M5'	175
Table 8.2: R^2 , MAPE (%) and RMSE of the six models	178
Table 8.3: Multi-objective error of the six models	179
Table 8.4: Descriptive statistics for relative error (%) of the six models	179
Table 8.5: Hypothesis testing to compare the six prediction models	179
Table 8.6: Parameter settings for M5'	180
Table 8.7: R^2 , MAPE (%) and RMSE of the six models	186
Table 8.8: Multi-objective error of the six models	186
Table 8.9: Descriptive statistics for relative error (%) of the six models	187
Table 8.10: Hypothesis testing to compare the six prediction models	187
Table 8.11: Parameter settings for M5'	188
Table 8.12: R^2 , MAPE (%) and RMSE of the five models.....	190
Table 8.13: Multi-objective error of the five models.....	191
Table 8.14: Descriptive statistics for relative error (%) of the five models.....	191
Table 8.15: Hypothesis testing to compare the five prediction models	191
Table 8.16: Parameter settings for M5'	192
Table 8.17: R^2 , MAPE (%) and RMSE of the six models for output H	196
Table 8.18: R^2 , MAPE (%) and RMSE of the six models for output E	196
Table 8.19: R^2 , MAPE (%) and RMSE of the six models for output S	197
Table 8.20: Multi-objective error of the six models on the three outputs.....	197

Table 8.21: Descriptive statistics for relative error (%) of the six models for the output H ...	197
Table 8.22: Descriptive statistics for relative error (%) of the six models for the output E ...	198
Table 8.23: Descriptive statistics for relative error (%) of the six models for the output S ...	198
Table 8.24: Hypothesis testing to compare the six prediction models for the output H	198
Table 8.25: Hypothesis testing to compare the six prediction models for the output E	198
Table 8.26: Hypothesis testing to compare the six prediction models for the output S	199

LIST OF ABBREVIATIONS

ANFIS	adaptive neuro-fuzzy inference system
ANOVA	analysis of variance
AIC	akaike information criterion
AMGGP	advanced multi-gene genetic programming
ANN	artificial neural network
ARIMA	autoregressive integrated moving average
BIC	Bayesian information criterion
BP	back propagation
BPNN	back propagation neural network
CAD	computer aided design
CART	classification and regression trees
CI	computational intelligence
CGP	cartesian based genetic programming
C-MGGP	classification-driven model selection approach of multi-gene genetic programming
CNC	computer numerical control
CRISP	cross industry standard process for data mining
CSA	coupled simulated annealing
DM	data mining
DSA	dynamic strain aging
ERM	empirical risk minimization
FDM	fused deposition modeling
FFT	fast fourier transform
FF	full factorial
FL	fuzzy logic
FL-ANN	fuzzy logic-artificial neural network
FEM-GP	finite element method-genetic programming
FPE	final prediction error
GARCH	generalized autoregressive conditionally heteroskedastic
GA-FL	genetic algorithm-Fuzzy logic
GA-ANN	genetic algorithm-artificial neural network

GA-GP	genetic algorithm-genetic programming
GEP	gene expression programming
GP	genetic programming
GP-OLS	genetic programming-orthogonal least squares
GP-SA	genetic programming-simulated annealing
GUI	graphical user interface
JC	johnson cook
JEW	jenkins-watt
kNN	k-nearest neighbours
K-S	kennard and stone
LCI	lower confidence interval
LS-SVM	least squares-support vector machines
LPV	linear parameter varying models
MGGP	multi-gene genetic programming
MGGP-ANN	multi-gene genetic programming
M-MGGP	modified multi-gene genetic programming
MAPE	mean absolute percentage error
MARS	multi-adaptive regression splines
MEP	multi-expression programming
ML	machine learning
MO	multiobjective error
MFC	microbial fuel cell
OLS	orthogonal least squares
PART	partition and regression trees
PSO	particle swarm optimization
PRESS	predicted residual error sum of squares
RP	additive manufacturing
RBF	radial basis function
RMSE	root mean square error
RSM	response surface methodology
SA	sensitivity analysis
SE mean	standard error of mean
SVM	support vector machines

SI	system identification
SRM	structural risk minimization
SVR	support vector regression
SVC	support vector classification
STD	standard deviation
UCI	upper confidence interval
UTM	universal testing machine
VC	vapnik-Chervonekis

LIST OF PUBLICATIONS

The following journal publications are related to the present research:

1. **Garg, A.**, Rachmawati, L. and Tai, K. “Orthogonal Basis Functions as a Complexity Measure for genetic programming models in regularized fitness functions”, *IEEE Trans. on Evolutionary Computation*, (under preparation).
2. **Garg, A.**, Tai, K. "Stepwise approach for the evolution of generalized genetic programming model in prediction of surface finish of the turning process ", *Advances in Engineering software*, Vol. 78, pp. 16-27
3. **Garg, A.**, Tai, K. and Gupta, A.K. (2014) “A Modified Multi-Gene Genetic Programming Approach for Modelling True Stress of Dynamic Strain Aging Regime of Austenitic Stainless Steel 304”, *Meccanica*, Vol.49, No.5, pp.1193-1209.
4. **Garg, A.**, Rachmawati, L. and Tai, K. (2013) “Classification-Driven Model Selection Approach of Genetic Programming in Modelling of Turning Process”, *International Journal of Advanced Manufacturing Technology*, Vol.69, No.5-8, pp.1137-1151.
5. **Garg, A.**, Tai, K., Lee, C.H. and Savalani, M.M. “A Hybrid M5’-Genetic Programming Approach for Ensuring Greater Trustworthiness of Prediction Ability in Modelling of FDM Process”, *Journal of Intelligent Manufacturing*, Volume 25, Issue 6, pp. 1349-1365.
6. **Garg, A.**, Garg, Ankit, Tai, K., Sreedeeep S. (2014) “An integrated SRM-multi-gene genetic programming approach for prediction of factor of safety of 3-D soil nailed slopes”, *Engineering Applications of Artificial Intelligence*, Vol.30, No.1-4, pp.30-40.
7. **Garg, A.**, Tai, K. and Savalani, M.M. (2014) “Formulation of Bead Width Model of an SLM Prototype Using Modified Multi-Gene Genetic Programming Approach”, *International Journal of Advanced Manufacturing Technology*, Vol.73, No.1-4, pp.375-388.
8. **Garg, A.**, Vijayaraghavan, V., Tai, K. and Savalani M.M. 2014. (2013) “A novel evolutionary approach in modelling wear depth of laser engineering

titanium coatings”, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture (Imech)*, (In press).

9. **Garg, A.**, Tai, K. and Savalani, M.M. (2014) “State-of-the-Art in Empirical Modelling of Rapid Prototyping Processes”, *Rapid Prototyping Journal*, Vol.20, No.2, pp.164-178
10. **Garg, A.**, Tai, K., Vijayaraghavan, V. and Singru, P.M. (2014) “Mathematical Modelling of Burr Height of the Drilling Process Using a Statistical Based Multi-Gene Genetic Programming Approach”, *International Journal of Advanced Manufacturing Technology*, Vol.73, No.1-4, pp.113-126.
11. **Garg, A.**, Vijayaraghavan, V., Mahapatra, S.S., Tai, K. and Wong, C.H. (2014) "Performance Evaluation of Microbial Fuel Cell by Artificial Intelligence Methods", *Expert Systems with Applications*, Vol.41, No.4, pp.1389-1399.
12. **Garg, A** Vijayaraghavan, V., Wong, C.H., Tai, K. and Mahapatra, S.S. (2014) “Measurement of Properties of Graphene Sheets Subjected to Drilling Operation Using Computer Simulation”, *Measurement*, Vol.50, pp.50-62.
13. **Garg, A.**, Vijayaraghavan, V., Wong, C.H., Tai, K. and Gao L. (2014) "An embedded simulation approach for modeling the thermal conductivity of 2D nanoscale material", *Simulation Modelling Practice and Theory*, Vol.44, pp.1-13.
14. **Garg, A.**, Garg, Ankit., Tai, K. (2013) “A multi-gene genetic programming model for estimating stress dependent soil water retention curves”, *Computational Geosciences*, Vol.18, No.1, pp.45-56.
15. **Garg, A.**, Bhalerao, Y. and Tai, K. (2013) “Review of Empirical Modelling Techniques for Modelling of Turning Process”, *International Journal of Modelling, Identification and Control*, Vol.20, No.2, pp.121-129
16. **Garg, A.**, Vijayaraghavan, V, Wong, C.H, Tai, K, Sumithra K, Gao L. and Mahapatra S.S (2014). "On the Study of machining characteristics of 2-D nanoscale material” *Nanoscience and Nanotechnology letters* Volume 6, No. 12, December 2014, pp. 1079-1086.

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Modelling is a term widely used in the field of System Identification (SI), which is referred to as the art and science of building mathematical models for a system from the given input-output data. Modelling includes the systems, models and modeling methods, which can also be studied under the field of SI. The systems modelled can be manufacturing processes such as turning, vibratory finishing and additive manufacturing, etc. or chemical processes such as fuel cell, reactors or such as those involving the study of mechanical and thermal properties of graphene and carbon nanotubes or the stock market and weather phenomenon, etc. Among these processes, additive manufacturing processes (processes involves the fabrication of products from CAD data automatically), machining processes (material removal processes), vibratory finishing (material removal processes) are the potential ones. The working mechanisms behind these systems are governed by multiple input and output variables, which make these operating mechanisms complex. The cost involved in the execution of such systems is reasonably high, and therefore it can be costly to measure the data. Moreover, there exists some useful information hidden in the system. The information can be in the form of a relation between the system output and input variables, dominant input variables,

etc. Such information is vital for optimizing the performance of the systems. Also, in an era of widespread development of capital intensive systems with their complex operating mechanisms, the need of modelling and optimization has been strengthened [1, 2].

Models such as analysis of variance (ANOVA), hypothesis tests, functional expressions, etc used in various fields of science, namely natural science (physics, biology, earth science and metrology), social sciences (economics, sociology, political science) and engineering disciplines (manufacturing processes) are used to unveil the hidden information for the practical understanding and realization of the system. To formulate these models, a gamut of modelling methods such as regression analysis, response surface methodology (RSM), partial least square regression, genetic programming (GP), artificial neural network (ANN), fuzzy logic (FL), M5- prime (M5'), support vector regression (SVR), adaptive neuro-fuzzy inference systems (ANFIS), etc. can be applied [1, 3-6]. The models formulated must not only predict the output variables accurately on the testing samples but should also be able to capture the dynamics of the systems. This is known as a generalization problem in modelling. The generalization of data obtained from manufacturing systems is a capability highly demanded by the industry. Higher generalization ability of the model indicates that it has rightly captured the physics behind the system.

Systems, models as well as modelling methods can be studied under various classifications of SI [7, 8]. For example, modeling methods can be studied under the three categories of modelling: grey box, white box and black box [9]. Generally, the prior information about the system is not known, and, therefore the systems are modeled using the black box modeling methods such as ANN, GP,

FL, etc. Models and modelling methods can also be studied by the classification of SI into linear, nonlinear and evolutionary [7, 10-14]. Due to advent in development of capital intensive machines, the systems behave non-linearly. Therefore, the methods that fall within the category of non-linear SI are frequently being adapted by researchers to model these systems. However, these methods are based on a prior assumption of a model structure, and the estimation of the large number of coefficients of the model is not reliable. In this perspective, an evolutionary SI method namely GP, is used, since it evolves the model structure and its coefficients automatically [12, 15]. The third route of studying the methods and models is by the classification of SI into various fields such as statistics, econometrics, machine learning (ML), statistical learning theory, statistical process control, chemometrics, etc [8]. This classification is according to the type of systems to be modeled. For example, in chemometrics, the chemical systems such as fuel cell, reactors, etc. are modeled. In econometrics, mainly economic systems such as growth domestic product, stock markets, etc are modeled. The modeling methods can also be studied under two categories: statistical and computational intelligence (CI). Statistical methods consist of regression analysis, RSM, partial least square regression, etc [3, 16-18]. Statistical methods are based on various assumptions such as the structure of the model, the normality of residuals and uncorrelated residuals, etc. CI methods comprises advanced heuristic and optimization methods such as GP, ANN, M5', SVR, ANFIS, etc.

The study of modelling methods categorized under various classifications of SI reveals that the CI methods are extensively being applied by researchers for modeling non-linear systems because these methods have the ability to formulate the models from the given data without the need for incorporation of any other

prior knowledge about the systems. More efficient CI methods have been developed by clustering the features of two or more methods. For example, the hybrid methods: GA-FL, GA-ANN, FL-ANN, particle swarm optimization (PSO)-ANN, etc are able to predict the systems output accurately [17, 19-26]. The M5' method used to build regression trees has the prediction accuracy on par with that of ANN but the researchers have not studied this method comprehensively [27, 28].

Among CI methods, GP, also popularly known as evolutionary SI method, possesses a unique feature of evolving the model structure and its coefficients automatically. Extensive literature of GP in modeling of non-linear systems is found. Researchers have developed hybrid approach of GP such as GA-GP [29], Clustering-GP [30], FEM-GP [31], GP-OLS [32], GP-SA [33], etc for improving its generalization ability. New selection schemes and genetic operators for mutation, crossover and reproduction have been developed [34]. Variants of GP such as linear genetic programming, probabilistic genetic programming, multi-expression genetic programming, Cartesian based genetic programming (C-GP), gene expression programming (GEP), multi-gene genetic programming (MGGP), etc have been formulated [34]. From this, an inference can be drawn that by learning the features of several modeling methods under the various classifications of SI can provide a scope to hybridize the features of GP with others to make it more robust. Among those variants developed, the MGGP method, which uses multiple sets of genes for the formulation of a model, is primarily focused. However, there are important issues in the functioning of MGGP that need to be addressed.

Based on the preliminary applications of MGGP [35-38], it was found that the generalization is the main problem, due to which its applications have not gained much prominence. High generalization refers to the satisfactory performance of the model on the testing (unseen) data samples. High generalization is essential for the prediction of systems behavior in uncertain input process conditions [39]. Since the industrial data is costly to obtain, high generalization of models results in increase in productivity of the systems. Reasons behind poor generalization of MGGP approach can be attributed to: a) inappropriate procedure of formulation of the model, b) inappropriate measure of complexity of the model, c) difficulty in model selection and d) trustworthiness of prediction ability of the model on unseen samples. In MGGP method, the genes are randomly chosen and combined using the least squares method. Since the combination mechanism is random, the genes of lower performance i.e. genes having poor accuracy on the training data, may get combined with the other genes of higher performance, to form a model, which then gives poor generalization ability. It is also well argued that complexity of the MGGP models defined by the number of nodes of the tree and/or depth of the tree is not an appropriate measure. Restrictions on parameter settings such as maximum number of genes to be combined and the depth of the gene are not enough to exert control over the complexity of the model. Difficulty in model selection is another one of the vital issues in MGGP. Since MGGP is a population based method, it generates models of varying fits and sizes. Generally, the best MGGP model is selected based on the lowest training/validation error. However, it is found that there exists other models in the population, whose performance on the testing data is better than that of the best MGGP model with little compromise on the training error. The issue of ensuring greater trustworthiness on the

prediction ability of the model [40] is a big concern, because in practice, the best model may not perform satisfactorily on the testing samples. The approach commonly used for ensuring greater trustworthiness on the prediction ability of model is based on ensembles, i.e. averaging the predictions of best models formed from the given input-output data [4].

In addition, while the author was working closely with a major company in the manufacturing industry (i.e. Rolls Royce), it is learned that the industry is keen to develop functional expressions, which can also be easily optimized analytically or coded into a system for online prediction and monitoring [40]. Moreover, there is also a demand for the development of user friendly graphical user interface (GUI) software for the implementation of GP [34, 41]. Hence, this has indeed motivated the author to work on CI methods, specifically MGGP and develop a robust CI approach on GUI.

The research objectives and the scope of the current work are discussed in the following sections.

1.2 Research Objectives

The objective of this research is to develop a robust CI approach by adopting parallelism mechanism of an advanced multi-gene genetic programming (AMGGP) with another CI approach. The objective can be achieved on the completion of following sub-objectives.

(a) Develop a modified- MGGP model by embedding stepwise regression in its paradigm.

- (b) Develop Orthogonal basis functions as a complexity measure for MGGP models in regularized fitness functions.
- (c) Develop a Classification-driven model selection approach of MGGP.
- (d) Develop a Hybrid approach for ensuring greater trustworthiness of prediction ability of the model on the unseen samples.

1.3 Scope of the Current Research Work

The scope of the current research work is as follows.

For combining the genes in a more efficient way, stepwise regression principle from the field of statistics is integrated in the paradigm of MGGP. By embedding the stepwise approach, only selective genes of higher performance are chosen for combination. In this way, the selection of relevant genes for combination improves the generalization characteristic of the MGGP model.

The complexity measure in MGGP is defined using the two orthogonal basis functions: polynomials and multi-adaptive regression splines (MARS). The minimal order of polynomial or number of basis functions of MARS that best fits the MGGP model is considered as a measure of its complexity. In this way, by incorporating this new measure of complexity in regularized fitness functions, the generalization ability of the MGGP model is improved.

Classification criteria based on validation error is designed to classify a MGGP model into the two categories of “bad” and “best”. The new methodology integrates potential classification methods with MGGP to drive the model selection in MGGP with classifiers formed based on the variables of the model:

training error, validation error and number of nodes. Classifiers are able to classify the class (bad or best) of MGGP and in this way the best model is selected from the pool of models.

The hybridized methods are developed using M1 and M2 (M1 and M2 are two potential CI methods) model in parallel, and where M1 predicts the error of the M2 model. The proposed approach can work effectively on the smaller set of data which otherwise could have resulted in huge consumption of time and resources. The M1 model also ensures greater trustworthiness on the prediction ability of the M2 model on an unseen sample.

1.4 Organization of the thesis

The remainder of this report is organized as follows:

Chapter 2 provides a literature review which gives an idea on the type of systems, models and modelling methods in SI. **Chapter 3** discusses the CI methods including the proposed approach, which is implemented in phases in the subsequent chapters: 4, 5, 6, 7 and 8. **Chapter 4** introduces a modified MGGP approach. **Chapter 5** illustrates two new complexity measures of the MGGP model. **Chapter 6** discusses the classification-driven model selection approach of MGGP. **Chapter 7** introduces the hybrid approach of MGGP. **Chapter 8** introduces the applications of the proposed CI approach. The concluding remarks and the recommended future work are discussed in **Chapter 9**.

CHAPTER 2

LITERATURE REVIEW ON SYSTEMS, MODELING METHODS AND MODELS

2.1 Systems, Modeling methods and Models

Fig. 2.1 is an illustration of how the literature review and the subject of modelling studied under the SI community is organized and structured in this chapter. Information about the systems, modelling methods and the models is obtained by studying the SI into the following categories:

- (a) Grey box, White box and Black box
- (b) Parametric and Non-parametric
- (c) Statistics, Econometrics, Machine learning, etc.
- (d) Linear SI, Non-linear SI and Evolutionary SI.

Further, the author listed the important manufacturing systems to be studied and the reason of choosing the evolutionary SI approach of genetic programming. The main issues behind the functioning of genetic programming are addressed. To tackle these issues, author would propose a robust CI approach in his work.

Fig. 2.1 Illustration of literature review on SI

In the literature, the efficiency of many systems such as blast furnace process [42], grinding process [43], drilling process [44, 45], milling process [46], spinning process [9], vibratory finishing process [47], turning process [48], online fault diagnosis (FD) [49-51], software reliability [52], chemical reactions [53] and biological systems [54], etc. have been improved by deploying the mathematical models formulated using various modeling methods. The mathematical models can be represented by an equation, graph, table, block diagrams, decision trees, chart, etc. Generally, a model should represent the functional relationship between the output and input variables of the system. The model is represented by an equation comprising the input variables, the output variable, the constants and the coefficients. The procedure of the formulation of the mathematical model is shown in Fig. 2.2 with the steps as follows:

- (a) Design the experiments and collect the data.
- (b) Do necessary pre-processing such as checking outliers, normalization, eliminating multicollinearity, transformation, aggregation, etc.
- (c) Select the set of model structures.
- (d) Fit each of the model structure and compute its coefficients using the statistical, numerical or optimization methods.
- (e) Validate the models on the testing data, and, if not satisfied, then repeat step 3.

Fig 2.2 Step-by-step procedure to formulate models using given data

Several modelling methods exist that formulate the models by estimating its coefficients. In the following sections, models and modelling methods which have been applied to predict the response of various systems are introduced under various classifications of SI. The purpose of introducing the models and modelling methods is to highlight the features of various modelling methods, the types of systems that have been modelled, technical terms used in modelling and the

understanding on how an improvement in the performance of one method can be made by incorporating the features of other modelling methods. In the end, vital issues arising from the study on classifications of SI are highlighted and discussed in brief.

2.1.1 Classification of SI into White box, Grey box and Black box

The models can be classified into the three categories: white box, grey box and black box based on the kind of modelling phenomenon used [9]. Qualitative differences between these models are shown in Table 2.1. When prior information about a system is available in the form of mechanical, chemical or physical equations, then these equations are known as white box or glass box or clear box models. Such a phenomenon of modelling a system is known as white box modelling. Generally, the processes occurring in nature behave non-linearly and the white box models cannot take into account the complexity of the systems. In this perspective, the grey box models are formulated based on insights of the system and the experimental data. This type of modelling is known as grey box modelling. When prior information about the system is not known, a general class of functions can be used to fit the data. These functions require estimation of many coefficients, which is computationally intensive and sometimes unreliable. In such cases, a few CI methods such as ANN, GP, FL, etc., which assumes no prior assumptions about the structure of the model can be used as an alternative, and this type of modelling is known as black box modelling.

Table 2.1 Classification of SI into three categories

Black Box	Grey Box	White Box
Models: No assumption of model. Also referred as empirical Modeling	Models: Differential equations, Polynomial equations, Weiner series, etc Involve coefficients estimation and is both analytical and empirical.	Models: Newton Law, Pascal law, Gravitational Law Also known as analytical Modeling
Methods: ANN, GP and FL	Methods: optimization methods, statistical Methods, numerical methods.	Derived from first principles.

2.1.2 Classification of SI into Parametric and Nonparametric Methods

Ljung [8] studied the modelling methods by the classification of SI into two categories: parametric and non-parametric methods. The qualitative differences between these methods are shown in Table 2.2. Based on the data obtained from the system, the methods used for estimating the coefficients of the model are known as parametric methods and the models are known as parametric models (for example, differential algebraic equations [55], state space models, smoke-grey models, composite local linear models and linear parameter varying models (LPV) , block oriented system, ANN, FL, SVM , etc). Methods in non-parametric category do not estimate the coefficients of the model but instead used to form a surface by smoothing over the data points in the space (for example, semi-supervised regressions, local polynomial methods, direct weight optimization, kernel methods, etc).

Table 2.2 Classification of SI into two categories

Parametric	Non-parametric
Methods are used to estimate the coefficients of the model	Methods forms a surface and generate a data point
Methods: linear parameter varying models, block oriented system, ANN, FL and SVM	Methods: semi-supervised regressions, local polynomial methods, direct weight optimization and kernel methods.

2.1.3 Classification of SI into Various Fields

Various modelling methods and models can also be studied by the classification of SI into its applications in various fields such as statistics, econometrics, time series, statistical learning theory, ML, chemometrics and data mining. Table 2.3 shows the models and methods used in these fields.

2.1.3.1 Statistics

Statistics is referred to as the parent of SI [2], since it has been rigorously applied in a wide range of disciplines such as data mining, chemometrics, demography, econometrics, image processing, etc. Statistics can be studied under two categories: descriptive and predictive statistics. In descriptive statistics, the data is characterized by means of statistical variables such as mean, standard deviation, variance, skewness, minimum, maximum, range, etc whereas, in predictive statistics, the statistical tools are used to unveil the hidden relationships in the data for the study of process behaviour. The data can be illustrated graphically using box plots. Box plots are very useful for evaluating the performance of the models. Statistical models are usually represented by a set of mathematical equations in

terms of random variables and their associated probability distributions such as z , chi-square, F , t , etc [56]. Several statistical methods that assist in establishing the correlations are ANOVA, chi-square test, correlation, factor analysis, mann-whitney U, mean square weighted deviation, PLS regression, ridge regression, student t-test, and method of least squares [57]. Besides the statistical models, the problem of multicollinearity in the data (the high correlation between the input variables) is discussed explicitly in this field [57]. A study on the comparison of statistical and machine learning methods was conducted by Garg and Tai [18]. In this study, various statistical methods such as stepwise regression analysis, PLS regression, ridge regression, etc. were implemented using the statistical packages such as JMP, MINITAB, SYSTAT, SPSS, etc. These statistical methods were able to select the relevant input process variables by eliminating the highly correlated and redundant input variables based on the p -values. Their performance was compared to those of ML methods: MGGP and ANN. It was found that MGGP was able to unveil the relevant input process variables without the need for use of variable reduction methods. The drawback of using the statistical methods is that they require expertise in statistics to conclude on an inference about the system behaviour. The statistical models are linear, quadratic, cubic, etc., and which have to be pre-assumed for fitting them to a given data. The errors are pre-assumed to be normally independent and distributed with zero mean and constant variance. Therefore, such models may not describe the non-linear and the interactive relationships between the process variables and so may not be reliable for use when there is limited information about the system.

Table 2.3 Classification of SI into various fields

Fields	Models	Modelling methods	Remarks
Statistics	Z ,t, F and chisquare distributions	Regression, correlation, and factor analysis	Pre-assumption of the model structure, not suitable for modelling non-linear systems.
Econometrics	ARIMA, tobit, etc	Mainly statistical methods	Need expertise for making decisions from the statistical models
Time series	ARIMA, GP, SVR and ANN	FFT, ANN, GP and SVR	Modern heuristic methods are mainly considered.
Statistical learning theory	SVR model	Regularization networks and SVM	Includes new measure of performance of model such as ERM and SRM. Well known for providing generalization ability
Machine learning (ML)	Decision trees, ANN, SVR, GP and kNN	SVM, GP, M5, RIPPER, CN2, ANN and kNN	No pre-assumption of model structure. Adapt to the non-linearity of the systems. Implementation of methods require expert knowledge.
Chemometrics	Polynomials, ANN, SVR, GP and kNN	DOE, signal processing PLS, PCA, MDS, ANN, SVR, GP and kNN	Emphasis on pre-processing of the data and validation of the model.
Data Mining (DM)	CRISP, SEMMA and Six-Sigma	Statistical charts, variable reduction methods and visualization methods	Finding hidden patterns in the data. Highly crucial in banks and industries.

2.1.3.2 Econometrics

In this field, the important economic related decisions and measures are taken using the mathematical or econometric models. These models are mainly developed using the statistical methods and represents the key relationship between the factors such as price, demand, quantity, etc. [58]. Central banks and government also used these models for evaluating and guiding economic policy (such as the Federal Reserve Bank [59] and DRI-WEFA [60] model). Some other econometric models are autoregressive integrated moving average (ARIMA), tobit, vector auto-regression, co-integration, etc [58]. Econometric analysis is carried out by various methods such as single equation methods, simultaneous methods, method of moments, Bayesian methods, two stage least squares, three stage least squares, generalized method of moments, etc. Since most of the econometric models are statistical, core expertise is needed in understanding the statistical variables of the model and making critical economic decisions from these models [61].

2.1.3.3 Time series

A time series is a sequence of observations of a random variable which essentially is from a stochastic process. Examples of time series include monthly demand for a product, inflow of immigrants into a country, daily volume of flows in a river, weather data, etc. Forecasting time series data is an important component of operations research because these data often provide the foundation for decision models. An inventory model requires estimates of future demands, a course scheduling and staffing model for a university requires estimates of future student

inflow, and a model for providing warnings to the population in a river basin requires estimates of river flows for the immediate future.

Time series analysis provides tools for selecting a model that can be used for forecast of future events. Modelling the time series is a statistical problem. Forecasts are used in computational procedures to estimate the variables of a model being used to allocate limited resources or to describe random processes such as those mentioned above. Time series models assume that observations vary according to some probability distribution about an underlying function of time.

In time series modelling, stock market prediction is of great challenge because it possesses higher volatility, complexity and dynamics. The methods for predicting stock market index includes classical and modern heuristic methods [62]. The classical methods such as exponential smoothing methods, regression methods, ARIMA, threshold methods and generalized autoregressive conditionally heteroskedastic (GARCH) methods rely on statistical assumptions, choice of model structure and assume that the time series is stationary [62, 63]. The modern heuristic methods such as GP, SVR and ANN have the ability to handle non-static stock markets and build non-linear stock market forecasting models [64-68]. These methods do not require the model to be prescribed as in the case of classical methods.

2.1.3.4 Statistical learning theory

Statistical learning theory provides the theoretical basis for many of today's CI algorithms. In particular, the focus is on the generalization ability of the learning algorithms in terms of how well they perform on the testing data [8].

The training of a learning algorithm is statistical in nature and so the design procedure should take into consideration both the performance of the model and its complexity. The task of the learning machine is to minimize a function:

$$J(w) = E(w) + \lambda C \quad (2.1)$$

where, $E(w)$ is the empirical risk or the standard performance measure resulting from the training data set such as the root mean square error, and the second term C is a complexity term usually specified by the number of coefficients in the model. Examples of such functions include model selection criteria such as Akaike information criterion (AIC) [69], Jenkins-Watt (JEW), Final prediction error (FPE), Bayesian information criterion (BIC) [70], predicted residual sum of squares (PRESS) [71], structural risk minimization (SRM) [72], etc. In equation (2.1), λ is known as the regularization parameter which plays a major role in exerting the generalization ability in the model. When the value of λ is zero, the equation is called the empirical risk minimization (ERM) principle and no capacity control is utilized, which normally leads to *over-fitting* of the training data and results in poor generalization. When λ is increased, more emphasis is placed on the complexity, and the error rate in the training data set increases, but better generalization is achieved. This means that a suitable balance should be struck between the empirical risk and the complexity term of the error. Statistical learning theory has lived with this compromise since its early days.

The statistical learning theory gained wide popularity following the development of the Vapnik-Chervonenkis (VC) theory by Vapnik [72]. Vapnik [72] proposed the SRM principle as an alternate inductive principle for learning, which is able to control the generalization ability of learning machines by minimizing a confidence

interval derived from the capacity of the set of functions implemented by the learning machine (VC dimension), instead of striking the compromise between empirical risk and machine complexity. The same author showed later that a practical way to minimize the VC dimension is to design classifiers that maximize the margin. The margin is defined as the minimum distance between the training set samples and the decision surface. The framework of CI algorithms, namely SVR and support vector classification (SVC), were developed based on statistical learning theory and regularization networks.

2.1.3.5 Machine learning (ML)

ML is one of the important fields of SI, where the algorithms are developed and applied for making the computer predict behaviours based on the measured data [73]. The fields that are associated with this discipline are probability theory, statistics, data mining, pattern recognition, adaptive control, theoretical computer science, computational neuroscience, etc. ML algorithms can be classified into different types depending on the outcome of algorithm. The literature identifies five typical classifications of ML based on learning, namely supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning and manifold learning. Among these, supervised and unsupervised learning have been an intense focus of researchers [74, 75].

In the case of supervised learning, the training data consist of a set of training samples and each sample is a pair consisting of an input object (typically a vector) and a desired output value. A supervised learning method analyses the training data and generates a function, which is called a classifier (if the output is discrete) or a regression function (if the output is continuous). The function should predict

the correct output value for any valid input object and this requires the learning algorithm to generalize from the training data to unseen (testing data) situations.

Kotsiantis et al., [75] has rigourously discussed the advantages, disadvantages and issues relating to the modelling methods falling in the supervised learning category. Three categories of methods for the supervised learning are as follows:

(a) Logic based (symbolic) algorithms

The modelling methods includes decision trees models such as FICUS, C4.5, EC4.5, rainforest, PUBLIC, etc. The advantage of using a decision tree model is its comprehensibility and easy interpretation by humans.

(b) Data-driven based supervised learning

GP and ANN are the commonly used ML methods for supervised learning. GP, based on the evolution of a population of models, possesses the ability to evolve the model structure and coefficients automatically based on only the given data. In ANN, the multilayer perceptron uses back propagation neural network (BPNN) for updating the weights of the architecture of ANN. BPNN is based on gradient descending process and may get stuck in local minima. Hence, for determining the optimal neural network structure, powerful optimization methods such as GA and PSO are used [3, 26, 76, 77]. Other variants of ANN used widely are radial basis function (RBF) neural networks. RBF neural network is a three layer neural network in which each hidden unit implements a radial activation function. Applications of GP and ANN are found in forecasting, intrusion detection, image reconstruction, modeling of monthly traffic accidents, electrostatic field modeling, etc.

(c) Statistical learning algorithms

Statistical methods are characterized by having an explicit underlying probability model, which provides a probability that an instance belongs in each class, rather than simply a classification. Statistical learning methods such as Bayesian networks and instance-based methods are described in the following paragraphs.

Naive Bayes classifier is a probabilistic model (classifier) based on Bayes theorem. It assumes the strong assumption of independency of features and requires a small amount of data for the estimation of coefficients of the model (classifier) for classification. A Bayesian network is represented by a directed acyclic graph which could define the probabilistic relationship between the input and output variables. The advantage is that it takes into account the prior information about the given problem. However, it is not suitable for the data sets with multiple features, since the network constructed is not feasible in terms of time and space.

The instance based or lazy learning modelling method includes k-nearest neighbour (kNN), which is used to predict an unseen instance by observing the class of its labelled k -neighbours. It requires a large storage space for storing labelled features and is sensitive to the choice of k . The choice of parameter k is an important consideration and can be estimated using heuristic method such as cross-validation and evolutionary algorithms.

In ML, an unsupervised learning refers to the problem of finding the hidden pattern in unlabelled data. The methods that fall in this category are clustering methods, self-organizing maps and adaptive resonance theory [74].

2.1.3.6 Chemometrics and Statistical process controls

Chemometrics is one field of SI where the modelling methods are applied to the data set obtained from chemical systems [78]. One recent example is the application of CI methods in the performance evaluation of a microbial fuel cell (MFC) [37]. Three potential CI methods such as MGGP, SVR and ANN were applied for identifying the voltage output of the MFC. Chemometrics study deals with the collection and pre-processing of the data sets followed by the formulation and validation of the model. Data should be collected in such a way that it represents the domain of the system. For this purpose, the experimental design methods such as full factorial and latin hypercube sampling are used. Data is costly to obtain and therefore sampling efficiency of these designs is important in formulating the model that represents the system characteristics. Recently, a few studies on the comparison of performance of experimental design methods can be found [79, 80] which provide information on the optimum design to be used. After the data collection, the data is separated from noise using the resolution and signal processing methods such as least squares peak, curve resolution and fourier spectral de-convolution , kalman filtering, etc. [81]. Since the data collected in chemical systems possesses higher dimensions, the characteristics of such variables cannot be interpreted from high dimensional plots such as 4D, 5D, etc. Hence, the projection or data (variable) reduction methods such as PLS, principal component analysis, multidimensional scaling, local linear methods, etc. are used to reduce the original dimensional space to lower dimensional space [82]. These methods are also used to eliminate multicollinearity in the data set when there is a strong relationship between the variables.

2.1.3.7 Data mining (DM)

DM is the SI field that deals with extraction of information from the highly complex data set [83]. In contrast to the field of ML, the emphasis here lies in discovering unknown patterns as opposed to generalizing known existing patterns to the new data. DM methods are applied in major fields of statistics, association rule learning, ML and visualization, etc [84]. Visualization methods include interactive displays such as, e.g., scatter plots, coplots, projection data sets, parallel coordinates, projection based methods, etc for extracting information from the given data. Some of DM models used are cross-industry standard process for data mining, Six –sigma and SEMMA [84]. The purpose of using these models is to convert the data into valuable information and delivering this information to a business analyst for making strategic decisions.

2.1.4 Classification of SI into Linear and Non-linear and Evolutionary SI

SI can also be classified into three phases: linear [7], non-linear [7] and evolutionary SI [12]. Important qualitative differences of methods among these phases are shown in Table 2.4. The brief description of each are discussed as follows:

Table 2.4 Classification of SI into various fields

Type of SI	Models	Modelling methods	Remarks
Linear	Linear polynomial	Gradient, steepest descent. Newton method, etc.	Pre-assumption of model structure. Inability to capture dynamics of the system. Convergence proof of estimation methods not available.
Nonlinear	Volterra , Weiner , block oriented systems, bilinear systems, etc.	Kernel method, linear regression, FFT, etc.	Pre-assumption of model structure. Estimation methods restricted to a few coefficients and thus not reliable.
Evolutionary	GP Tree	GP	Evolve model structure and coefficients automatically. Generate variety of forms of models.

2.1.4.1 Linear SI

Zadeh [14] formulated the SI problem as an optimization problem by defining a loss function as sum of squares of difference between the actual and predicted values. The loss function can be minimized using ordinary least squares [85], correlation methods [86], etc. Such methods suffer from the violation of statistical assumptions such as uncorrelated, independent, and constant variance errors. In order to counter this problem, there exist more generalized ways of defining loss functions such as generalized least squares [87], maximum likelihood [88], instrumental variables [89] and weighted least squares [90]. The loss functions can

also be minimized by various traditional optimization methods such as gradient methods, steepest descent method, Newton's method, conjugate gradient method, Rosenbrock pattern search, Hessian matrix and standard errors, Hooke-Jeeves pattern search, etc [91, 92]. The model selective criteria based on cross-validation such as gross cross-validation, AIC, BIC, FPE, Mallow's C_p and PRESS are used for the discrimination of model generalization capability [93]. The model selective criteria based on experimental designs such as A-Optimality and D-Optimality [94] are also used. From the literature on linear SI methods, several shortcomings are reported as follows.

- (a) The convergence proof of the traditional optimization methods required to minimize the loss functions are not available, and moreover they require the users to specify the initial start values and step size, which is a kind of uncertainty.
- (b) Estimation of large number of coefficients of the assumed model involves hard computation and is time consuming.
- (c) The choice of a model structure determines the computational effort required to solve the problem, the possibility of getting an optimal solution (variables evaluation) and the representation of the results. Hence the pre-assumption of the model structure is also an important assumption.

The area of linear SI has become quite matured since the system comprising of the linear relationships between the variables can be easily interpreted and visualized using the linear regression model. However, the processes occurring in nature are inherited with complexities and non-linearity.

2.1.4.2 Non-linear SI

The processes occurring in nature are non-linear and can be recognized as linear processes combined with perturbations. Billings [11] reviewed the modelling methods used for modelling the non-linear systems. Ku and Wolf [13] have discussed the applications of Volterra-Wiener functional series for the analysis of non-linear systems. Haber and Unbehauen [95] conducted a survey on structure identification methods for the modelling of non-linear dynamic systems. Hong et al., [10] in his work has discussed about the model selection approaches for non-linear SI. The Literature from these peer researchers is summarized and described in the following paragraphs.

The functional series such as Volterra series, Weiner series, block oriented system [96], bilinear systems [97] are represented as non-linear models. Voleterra series is a functional representation of dynamic non-linear and time invariant functional. The model is determined by the estimation of Voleterra kernels. Several estimation methods such as cross correlation, linear regression, exact orthogonal theorem, kernel method, differential sampling are used for estimating these kernels. Wiener series are the orthogonal expansions of non-linear functional and is used to approximate the response based on the past input data. The frequency domain methods such as FFT and Laplace functions are used for estimating Weiner kernels.

The block oriented system consists of linear dynamic and non-linear static elements. It is a cascade system in which a linear system followed by a nonlinear element in cascade with linear system. Several methods for identifying the stochastic and deterministic bilinear system are proposed by researchers.

Balakrishnan [98] and Bruni et al., [99] used maximum likelihood criterion for identifying the coefficients (variables). Baheti and Mohler [100] applied correlation analysis combined with least squares. The coefficient estimation methods are classified according to the model structures which are based on linear in variables or non-linear in variables. For the structure of the system known a priori, the variables estimation algorithms can be applied directly to estimate the variables of the model. If there is no information about the structure then the several classes of model such as differential equation, polynomial equation, Wiener series, and Voleterra series could be used to represent the input-output data and then the parameter estimation or kernel identification methods could be applied to estimate the variables [101]. Garg and Boziuk [102] assumed that the systems states are measured without noise and used least square algorithm to estimate the variables. Ivakhnenko [103] proposed the group method of data handling for identifying the variables using the principles of heuristic self-organization with large dimensionality and short data sequences. This parameter estimation method effectively reduces the computations associated with fitting polynomial models and appears to work well in non-linear system prediction applications.

Hong [10] illustrated an overview of the model selection approaches for the identification of the non-linear system. The most fundamental problem in the non-linear SI is to find the minimal order with the best generalization performance of the model from the data. An attempt has been made to construct the model using the forward/backward subset selection algorithms that reduce the number of possible model candidate evaluation by removing one regressor (order) at a time based on the consequent improvement in the model fit [104]. Various subset

selection algorithms such as backward elimination, forward selection, stepwise regression analysis and forward orthogonal least square (OLS) algorithm are used [105].

From the study on non-linear SI methods, several shortcomings are reported as follows.

(a) The choice of a model to be used depends on the system to be modelled. This is a kind of uncertainty in the assumption of a model.

(b) The number of coefficients and kernels to be specified is a difficult task for the user. The methods proposed are restricted to measuring up to 2nd order kernel.

(c) Large number of kernel and parameter estimation is time consuming.

2.1.4.3 Evolutionary SI

The shortcoming of methods in the category of linear and non-linear SI reveals that there is a need to adopt a methodology that can avoid model assumptions. Hence, researchers applied evolutionary approaches such as GP, which evolves model structures and its coefficients automatically.

GP has found numerous applications in the field of engineering, especially for formulating models from the given data. It has been applied successfully to the modelling of laboratory scale process involving the coupled water tank system and to the identification of a helicopter rotor speed controller and engine from the flight test data [106]. The structure of the model and its coefficients was identified using GP, and furthermore the coefficients were optimized using Nelder-simplex search and simulated annealing. The results show that the values predicted from the GP

model are well in agreement with the experimental data. GP is applied to formulate models for the prediction of three variables, namely the melting rate, specific required amount of oxygen, and the carbon content in the hot metal of blast furnace process. Performance of SVR and GP was found to be better than linear regression. Another application of the GP in industrial process involves the case study of fibre to yarn production process [9]. It involves the prediction of the most important characteristics of the yarn, spinnability and its tenacity which depends on the quality of fibre and the machine variables setting. The predictions obtained from the GP model was found to be on par with ANN. Yang et al. [107] described the application of GP for the force identification of dynamic system. Various examples described prove that the GP could be effectively used to identify the excitation force of the single degree, three degree and more complex frame dynamics systems. Flores and Graff [108] in their work proposed evolutionary computation based system identification approach using GP and gene expression programming (GEP). An attempt has been made to identify the non-linear structures using GP and the simultaneous parameter optimization of model using concepts of evolutionary strategies [109]. Madár et al., [32] proposed GP-OLS to generate the nonlinear input-output models. The OLS is used to exert control over the formation of overly-sized models that may result in their poor generalisation ability. GP-OLS is found to perform better in terms of accuracy when compared to standardized GP. From the study of applications of GP in modelling of various processes, the important characteristics of GP are as follows [110]:

(a) GP does not assume any structure of the model, i.e. it evolves the model structure and its coefficients by optimizing the global search space. Thus, GP is considered to be a structure optimization methodology.

(b) Unlike ANN, it provides the exact model visualization by generating the tree structure.

(c) The flexible way of defining the functional and terminal sets allow GP to generate various forms of models such as, for example, logistics models, probabilistic models, etc.

2.2 Systems and Modelling Method Chosen for the Study

2.2.1 Systems Chosen for Study

The study of systems, models and modelling methods under various classifications of SI was undertaken. The survey on applications of modelling methods in modelling of manufacturing systems (machining, additive manufacturing processes) undertaken by Garg et al. [16, 17, 34] suggests that the systems (machining processes such as turning, drilling, etc. and additive manufacturing processes such as fused deposition modelling (FDM) process) involve multiple number of input-output process variables, which makes their operating mechanism complex. Therefore, data is the only information that can be extracted from the system. Since the execution of these systems is costly, only a limited amount of data (in the range of twenty and thirty data samples) can be obtained. Generalization of the data obtained from the manufacturing systems is highly demandable because these systems are considered as the heart of engineering industry.

2.2.2 Reasons for Choosing CI Method of Multi-gene genetic programming (MGGP)

In various classifications of SI, the models formulated are represented by trees, if-then-rules, functional expressions, etc depending upon the type of method used. Methods categorised under the classifications of SI: statistics, linear and non-linear SI are based on assumptions of the model structure and requires estimation of large number of coefficients of the model. Based on the complexity of system involving the number of input and output process variables, there is a high degree of uncertainty in the selection of an optimum model structure. In addition, these modelling methods also require data pre-processing before their implementation, which makes their performance highly dependent on the kind of data. In this perspective, researchers have shifted their focus from the statistical or methods classified under linear and non-linear SI to the methods categorised under ML/CI or black box such as GP, SVR and ANN in analysing the system behaviour.

There exists several works [111-116] which have shown the ability of CI methods to build generalized models on the data samples obtained from the machining and additive manufacturing processes. Among the CI methods, GP also popularly known as evolutionary SI approach, evolves the model structure and its coefficients automatically. In addition, the GP method automates the selection of relevant input process variables and eliminate multicollinearity in the data. Thus, GP does not require pre-processing methods for the data dimension reduction. Extensive literature on applications of GP in symbolic regression problems indicates that the accuracy of the model was improved by developing hybrid approaches such as GA-GP, Bayesian-GP, Clustering-GP, FEM-GP, GP-OLS, GP-SA, etc. Each hybrid approach is a combination of two or more potential CI

methods. Since the evolutionary search in GP is dependent on its genetic operators such as reproduction, mutation and crossover, and the selection methods, researchers have therefore developed new genetic operators and selection schemes by incorporating a probability theory. Several variants of GP such as linear genetic programming, probabilistic based genetic programming, multi-expression genetic programming, cartesian based genetic programming (CGP), GEP, MGGP, etc were also developed using the methods of statistics and probability theory. This suggests that learning the features of modeling methods under various classifications of SI can provide some scope to improve the performance of GP by hybridizing its features with other potential modeling methods. Among these variants, MGGP method is extensively applied and considered as one of the popular methods for generating the functional expressions with greater accuracy. Unlike GP, in MGGP, every model in the population is constituted from the multiple sets of genes.

2.2.3 Main Issues Related to Generalization of MGGP

Based on preliminary applications of MGGP, it was found that the generalization is the most important issue emerging in MGGP. The main reasons behind poor generalization of MGGP are as follows:

(a) Inappropriate procedure of formulation of the model

In MGGP method, the genes are randomly chosen and combined using the least squares method. Since the combination mechanism is random, the genes of lower performance i.e. genes having poor accuracy on the training data may get combined with other genes of higher performance and degrades the performance of the MGGP model. If a way can be found to eliminate the genes of lower

performance and allow only the genes of higher performance to take part in the mechanism, the generalization of the MGGP model can be improved.

(b) Inappropriate complexity measure of the model

Generalization ability is also closely linked to model complexity in statistical learning area, mostly due to Occam's razor [117]. Loosely speaking, one should choose the simplest model that yields sufficient explanatory power for the available data. Identification of the right model complexity for a given set of data, however, is non trivial. Too low and the model would be *underfitted*, too high and the model would be *overfitted*. *Underfitting* produces large errors in training and test data, and is usually easily avoided. *Overfitting* is not quite as simple – because it usually implies limiting the fitting error on training data or trading it off for simplicity.

There are two main components of any approach to the generalization problem: first, finding an accurate measure of simplicity/complexity, and second, finding a compromise between the measured complexity and the fitting accuracy on training data. While the latter is a widely researched subject with considerable results in statistical learning, the former has not received as much attention particularly for the class of nonlinear models. While measuring the complexity of mathematical models in the absolute sense is difficult, when the space of models compared belong to the same class (e.g. polynomials, sinusoids, splines), relative complexity can be quite easily measured. L_1 and L_2 norms are in most cases adequate to represent model complexity in classical regression techniques [118].

MGGP, however, is unique in that the model space is not confined to that of the

same class. This contributes to the high approximation power of MGGP models, but it also means that preventing *overfitting* is less straightforward. Researchers in the area have discovered that regularized fitness functions that take into account fitting accuracy, model complexity, and/or amount of data produce significant improvement over a simple minimization of fitting error [10, 119]. Fitness functions such as AIC, BIC, JEW, SRM, FPE and PRESS comprise of a term known as complexity measure which takes into account the size of the model (number of nodes of the MGGP tree) and penalizes the accuracy of the model as the size of the model increases. In all these work, however, the number of nodes in a MGGP expression (effectively signifying the number of functions and variables) defines complexity. This definition would imply that, e.g. $\sin(x)$ and $-x$, are of equal complexity. The issue has been raised before [120], but the author is not aware of methods to address this problem in MGGP.

(c) Difficulty in model selection

In the context of symbolic regression problems, extensive literature has been published, but less focus has been paid to the difficulty in model selection in MGGP. Models evolved are of varying size and fitness values, and the best model is usually selected based on minimum training error. However, this best model may perform poorly on the testing data and cannot be used in extreme input process conditions. By running the preliminary trials on MGGP and as shown in Fig. 2.3, it was observed that there are a number of models in the population whose performance on the testing data is better than this model with little compromise on training error. The selection of the generalized MGGP model is highly important for ensuring the higher productivity of the process. Researchers have adopted

various procedures to tackle this problem, such as by applying a boosting method i.e. averaging the predictions of the best models obtained from the different combinations of training and testing data. In some cases, the best models are selected from the Pareto front through human intervention based on the minimum training error and are linearly combined to provide an accurate prediction. Several heuristics and methods for the selection of appropriate models have also been discussed in [121-125]. To the best of the authors' knowledge, little research has been done to resolve the issue of selection of a best model in MGGP.

Fig 2.3 Illustration of model selection problem in MGGP

(d) Trustworthiness of the model on unseen (testing) samples

CI methods: MGGP, SVR and ANN produce models from the training data. The appropriate training and testing data can be selected using the methods such as K-fold validation [126] and Kennard and Stone (K-S) algorithm [127]. Preliminary analysis suggests that data sets generated using K-fold validation and K-S algorithm gives somewhat similar performance of MGGP on the testing data [128-

132]. K-S algorithm takes less computational time and is therefore can be more appropriate for use. The testing data is used to test the generalization ability of the model. Whenever a new data sample corresponding to the testing range is to be evaluated, ensuring greater trustworthiness on the prediction ability of the model on this sample is of great concern. Researchers use the concept of boosting i.e. averaging the predictions of the best models obtained from the different combinations of training and testing data [122]. The best models are selected from the Pareto front and are linearly combined to provide an accurate prediction. Such method may improve the overall accuracy of the model but does not ensure the trustworthiness of the prediction ability of the model whenever a new sample is to be evaluated. This issue demands special attention as it directly links to the real world problem.

Modelling studied under the fields of SI is a vast area and the author in his present work tries to introduce the notion about the systems, the modelling methods and the models. The appropriate exchange of features and the emerging gaps among the various modelling methods can broaden the scope for researchers to contribute significantly in the field of SI. Thus, the present work shall attempt to resolve the major issues in MGGP and aims to develop a robust CI approach. The following chapter will discuss about the various CI methods.

CHAPTER 3

COMPUTATIONAL INTELLIGENCE METHODS INCLUDING PROPOSED METHODOLOGY

In this chapter, the five CI methods including the proposed methodology are discussed. For the simplicity of understanding of the proposed methodology, the four standardized CI methods ANN, M5', SVR and ANFIS are firstly discussed.

3.1 Artificial neural network

ANN is inspired by the operation of biological neural networks. ANN has no statistical assumption, therefore it can be used for a wide range of applications including classification, signal processing, and process control and regression problems [133, 134]. Recent applications of ANN are conducted by Ray et al [135-138], in which the ability of ANN in capturing the dynamics of underwater vehicles is explored. Other applications of ANN by Ray et al. [139, 140] include solving process-design-optimization problems using an evolutionary optimization framework. ANN is able to model the system with complex non-linear relationships between the dependent and independent variables with high prediction accuracy.

An ANN model consists of a highly interconnected set of artificial neurons that are usually organized in layers. Each artificial neuron aggregates and maps its input

according to a specified set of weights and non-linear mapping function. Common architectures are feed forward network and recurrent network. The feed forward network generally used is of single layer due to its simplicity (Fig 3.1). In fully connected feed forward network, neurons of one layer are fully connected to the next layer; whereas in a fully connected recurrent network, each neuron is connected to all other neurons.

Fig. 3.1 Feed forward neural network of Single-layer

A neuron takes in n number of inputs, which are modified by their respective weights, before passing the modified inputs through the activation function. The output of the activation function is taken as the output of the neuron (Fig. 3.2). Common activation functions include the sigmoid function, hyperbolic tangent, piecewise linear functions and identity function. The identity function is normally used for the output neuron. An identity function takes in real-valued inputs, and gives out the same value for output. For the hidden layer, the sigmoid function is used and it is given by:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

Fig. 3.2 Architecture of ANN

3.2 M5' Model Tree

M5' (M5-Prime) [28] is a model tree algorithm and a modified form of M5 [27]. A model tree (Fig. 3.3) is constructed with the leaf nodes and internal nodes. The internal nodes are the splitting criterion and leaves nodes are the linear regression models. M5' builds model trees by a divide-and-conquer approach. The model trees are first built, and then pruned to improve its predictive ability. Then, the smoothing process described in [27] is employed to remove the discontinuity between the adjacent linear regression models.

Fig. 3.3 M5' model tree: Models numbered 1-6 are linear regression models [20]

Given a collection of training cases T , the set T is divided into subsets in a way such that the sum of standard deviation within each subset is minimized. The algorithm used for splitting of set T is described in Quinlan [27]. The algorithm treats the standard deviation as a measure of error, and the expected reduction in error of a split is computed using the Equation 3.2. All potential splits are tested with this equation before the split that has maximum expected error reduction is chosen.

$$\Delta error = sd(T) - \sum_i \frac{|T_i|}{T} * sd(T_i) \quad (3.2)$$

After the splitting process is terminated, each subset is fitted with a linear regression model which will form the leaves nodes.

A large model tree tends to over-fit the training cases, and is likely to have poor predictive ability. Therefore, it might be necessary to reduce the size of the model tree in order to improve its prediction accuracy. During the pruning stage, a linear model is computed for each of the internal nodes based on the subsets of training cases that fulfil the splitting criterion of the respective internal nodes. Then an estimate of expected error of test data is computed for each node. It is computed by multiplying the average of absolute error in training cases at each node with the multiplication factor of $(n + v)/(n - v)$, where n is the number of training cases that reach that node, and v is the number of variables in the model that represent the internal node. The model tree is then pruned backwards from the leaves nodes to minimize the estimate of expected error.

3.3 Support vector regression

The most famous method in the field of CI specifically for imparting generalisation ability to the models is SVR [141-143]. The SVR method is derived from the SVMs, which is often applied to solve classification problems. The architecture of SVM is shown in Fig. 3.4. SVM when applied to regression problems is named as SVR. SVR is not based on statistical assumptions (model structure, error dependency, etc) and formulates models based only on the given data.

SVR originated from the community of statistical learning, and its framework is developed based on the principle of SRM. The input process variables space is transformed into the higher dimensional space T using the non-linear hyperspace

function. For example, the regression problem of higher non-linearity is converted to a linear regression problem in a higher dimensional space. Several non-linear/hyperspace functions exist to do the transformation.

Fig. 3.4 Architecture of SVM

The training data is used to formulate the SVR model, where x_i is the input process variable and y_i is the actual value/response of the process. The SVR model is given by:

$$y = f(x) = \sum_{i=1}^N w_i \rho_i(x) + b = w^T \rho(x) + b \quad (3.3)$$

where the function $\rho_i(x)$ is a space that has been converted into a higher dimensional space, and $w = [w_1 \ w_2 \ \dots \ w_N]^T$ and $\rho = [\rho_1 \ \rho_2 \ \dots \ \rho_N]^T$

Nonlinear hyper-surface given by Equation (3.3) is projected in the higher dimensional space T . The regression model $\rho(x)$ is the transformed linear model

in the higher dimensional space. Given the data, the kernel function chosen learns and minimises the regularised risk function (L_r). The optimisation of L_r determined the variables weight (w) and bias (b) as given in Equation (3.4).

$$L_r(w) = \frac{1}{2} w^T w + \lambda \sum_{i=1}^N |y_i - f(x)|_e \quad (3.4)$$

$$\text{where } |y_i - f(x)|_e = \begin{cases} 0, & \text{if } |y_i - f(x)| < \varepsilon \\ |y_i - f(x)| - \varepsilon, & \text{otherwise} \end{cases}$$

The regularisation parameter λ is a trade-off between the weight vector norm and the approximation error. The approximation error is decreased by increasing λ or weight vector norm, but this may not ensure the high generalization ability of the model and may cause *over-fitting*. The user defines the values of λ and ε where ε is the tolerance level. The definition of ε -insensitive loss function ($|y_i - f(x)|_e$) is given in Equation 3.4. If the predicted values of the SVR model $f(x)$ lies within the defined tolerance level ε , the loss function is zero, and for the points outside ε , the loss function is the absolute of the difference between the values predicted by the SVR model and tolerance level ε . The points on the margin lines ($y = f(x) \pm \varepsilon$) are called support vectors, whereas those outside are known as error set.

3.4 Adaptive neuro-fuzzy inference system

ANFIS is a neuro-fuzzy system, comprising a feed forward neural network where each layer is a neuro-fuzzy system component (Fig. 3.5) and simulates Takagi – Sugeno-Kang fuzzy rule of type-3 [144]. The consequent part of the rule is a linear

combination of input variables and a constant [145, 146]. The type-3 rule is of the following form as follows:

IF x_1 is A_1 **AND** x_2 is A_2 **AND** ... **AND** x_p is A_p **THEN** $y = c_0 + c_1x_1 + c_2x_2 + \dots + c_px_p$

where $x_1, x_2, \dots, x_p$ are the input variables, $A_1, A_2, \dots, A_p$ are the membership functions, y is the output variable and $c_0, c_1, c_2, \dots, c_p$ are the consequent variables.

The neural network structure consists of six layers (Fig. 3.5) as follows:

Fig. 3.5 Architecture of ANFIS [20]

Layer 0 is the *input* layer. It has n nodes, where n is the number of inputs to the systems

Layer 1 incorporates membership functions (MFs). A MF $\mu_{A_i}(x)$ can be piecewise and continuous differentiable functions that transform the input variable x into a membership degree between 0 and 1. Some of the most commonly used MFs are generalized bell, Gaussian, trapezoidal and triangular. Layer 1 is the fuzzification layer in which each node represents a membership value to a linguistic term as a Gaussian function with a mean;

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i} \right)^2 \right]^{b_i}} \quad (3.5)$$

where a_i , b_i and c_i are the variables of the membership function and their values are determined using BP-NN or least squares method or a hybrid of BP-NN and least squares method.

Layer 2 comprises nodes which strengthen the rules by means of multiplication operator. It performs AND operation given by

$$w_i = \mu_{A_i}(x_0) * \mu_{B_i}(x_1) \quad (3.6)$$

Every node in this layer computes the multiplication of the input values and gives the product as the output as in Equation (3.6).

Layer 3 is the *normalization* layer which normalizes the strength of all the rules according to the equation given by

$$\overline{w}_i = \frac{w_i}{\sum_{j=1}^R w_j} \quad (3.7)$$

where w_i is the firing strength of the i th rule which is computed in Layer 2. Node i computes the ratio of the i th rule firing strength to the sum of rule firing strengths.

Layer 4 is a layer of adaptive nodes. Every node computes a linear function where the function coefficients are adapted by using the error function of the multilayer feed-forward neural network.

$$\overline{w}_i f_i = \overline{w}_i (p_0 x_0 + p_1 x_1 + p_2) \quad (3.8)$$

where p_i are the variables, $i = n+1$ and n is the number of inputs to the system (i.e., number of nodes in layer 0). In the present work, there are five input variables and so there are six variables in Layer 4 and $\overline{w}_i$ is the output of layer 3. The variables are updated by Kalman filtering based on least squares approximation and back-propagation algorithm.

Layer 5 and 6 is the *output* layer whose function is the summation of the net outputs of the nodes in Layer 4. The output is computed by

$$\sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (3.9)$$

where w_i is the output of node i in Layer 4. ANFIS uses a hybrid learning algorithm to train the network. For the variables in Layer 1, the back-propagation algorithm is used. For the training of variables in Layer 4, a variation of least – squares approximation or back-propagation algorithm is used.

3.5 M5'-AMGGP

M5'-AMGGP is proposed in this work to tackle the four important issues related to generalisation (issues discussed in chapter 2, Section 2.2.3): (a) inappropriate procedure of formulation of the MGGP model, (b) inappropriate complexity measure of the MGGP model, (c) model selection and (d) ensuring greater

trustworthiness of prediction ability of the model on unseen samples. The formulation of CI approach is completed in two phases as follows:

- 1) Development of Advanced MGGP (AMGGP) approach.
- 2) Integration of the two methods AMGGP and M5' in parallel.

3.5.1 Phase 1: Development of AMGGP Approach

In order to understand the notion of the AMGGP approach, the MGGP method is firstly discussed. The MGGP algorithm is based on Darwinian principle of “Survival of the fittest” [147]. MGGP was formulated by Hinchliffe et al., [148]. The mechanism of MGGP is the same as that of GA but the only difference between them is that the latter evolves solutions represented by a binary or real number of fixed length whereas the former evolves solutions represented by tree structures/non-linear equations of varying sizes. Implementation of MGGP as shown in Fig. 3.6 involves a few steps, which are discussed as follows:

Fig. 3.6 Flowchart showing the mechanism of MGGP

(a) Initialization

MGGP method starts by generating multiple sets of genes randomly. An example of a gene generated is shown in Fig. 3.7. A gene is generated by combining the elements randomly from the functional and terminal set. Generally, the function set F consists of elements such as basic arithmetic operations (+, -, $\times$, /, etc.), Boolean operators (AND, OR, etc.), or other operators as defined by the user. The terminal set T consists of elements such as input process variables and numerical constants. The constants are specified so as to take into account the experimental errors in the data. The algorithm uses ramped half-and-half growth method for the generation of genes. This method ensures that the genes generated of different sizes are uniform in number. The size of the gene refers to the number of nodes in the gene. Checks are maintained regularly during the initialization stage to ensure

that no duplicate genes are created. Several sets of genes are then combined randomly using the statistical method of orthogonal least squares (OLS) to form a model (Fig. 3.8) comprising of the coefficients W_0 , W_1 , W_2 and W_3 . The maximum number of genes (G_{max}) that can be combined is set by the user. The output value is predicted based on the two input process variables such as x_1 and x_2 .

Fig. 3.7 Example of the gene $16 + \tan(x) - \left(\frac{8}{y}\right)$

Fig. 3.8 Formulation of the MGGP model using OLS method

(b) Evaluation of performance of models using fitness function

After the initialization of models, the performance of each model is evaluated using a user defined fitness function. The most frequently used fitness function is the root mean square error (RMSE) given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N |G_i - A_i|^2}{N}} \quad (3.10)$$

where G_i is the valued predicted of the i th data sample by the MGGP model, A_i is the actual value of the i th data sample and N is the number of training samples. It is argued that the models having larger size (higher number of nodes) are vulnerable to *over-fitting*. Therefore, other regularized fitness functions such as AIC, BIC, JEW, FPE, PRESS and SRM that take into account the empirical error as well as the size of the model can also be used in place of RMSE to avoid *over-fitting*.

(c) Genetic operations for evolution of new population

The models are ranked based on their fitness value. Tournament selection method, well-known for maintaining the diversity in the population, is applied to select the models for genetic operations such as crossover, mutation and reproduction. It ensures that not all good and not all bad models are selected for genetic operations. Since, the model consists of a set of genes, the method employs an operation known as two-point high-level crossover in addition to the standard subtree crossover and mutation operation. The two-point high-level crossover promotes the exchange of genes between the two selected models (two parents). For example, suppose the selected first parent consists of four genes (G_1, G_2, G_3, G_4),

and the second parent also consists of four genes (G_5, G_6, G_7, G_8). Points are randomly selected in each of these parents and the genes are swapped. Resulting offspring from these parents are ($G_1, \mathbf{G_6}, \mathbf{G_7}, G_3, G_4$) and ($G_5, \mathbf{G_2}, G_8$). However, the method ensures that during the reshuffling of the genes, none of the formed offspring consists of more genes than the specified G_{max} . In the case where the resulting offspring have more than the specified G_{max} genes, the genes are then selected randomly from that offspring and deleted. In addition to the reshuffling of genes between the models, standard subtree crossover and mutation operation are also applied. In MGGP, the standard subtree crossover is also referred to as low point crossover. In the crossover operation as shown in Fig. 3.9, a branch of tree is randomly selected from two models and swapped. MGGP is facilitated with mixed choice of mutation operations: (a) sub-tree mutation (b) mutation of constants using Gaussian function (c) Substitution and replacement of randomly selected nodes (d) assign randomly selected constant to zero (e) assign randomly selected constant to one (f) substitute randomly selected constant with another randomly generated constant. Generally, the highest weightage is given to the subtree mutation (Fig. 3.10), in which a branch of a model is randomly selected and replaced by the newly created gene. The settings of probabilities of crossover and mutation are set by the user.

Fig. 3.9 Subtree crossover operation

Fig. 3.10 Subtree mutation operation

(d) Model selection

This iterative phenomenon of generating new populations continues as long as the termination criterion is not met. Termination criterion is the maximum number of generations or the threshold error of the model, whichever is achieved earlier. For the model selection, there may be many models having different complexity (number of nodes) but with the same training error. The model which has minimum training error with lowest complexity, is chosen as the best model. The performance of this model is then evaluated on the testing data.

To improve on the MGGP approach, an advanced MGGP method is formulated by the author in an attempt to resolve the issues related to generalization. Fig 3.11 shows the working procedure of the AMGGP method. The difference between the proposed AMGGP and the MGGP approach can be seen in Fig 3.11 where procedures (1) and (2) shown by dashed lines are embedded to tackle the issue of inappropriate procedure of formulation of the model and measure of complexity of the model respectively, and (3) for tackling the difficulty in model selection. The new procedures incorporated in the approach are discussed in chapters 4, 5 and 6.

Fig. 3.11 Development of AMGPP method

3.5.2 Development of Graphical User Interface (GUI) and Settings of Variables for MGPP and AMGPP

Parameter settings play a major role in the effective functioning of the MGPP and AMGPP methods. Based on a trial-and-error approach as well as the studies undertaken by researchers, the parameter settings for the implementation of MGPP and AMGPP are set. For generating numerous forms of non-linear mathematical models, the broader set of functional elements is used. The values chosen for population size and number of generations depend on the nature of the

problem. Recommendations by Garg and Tai (2012) in their work on applications of GP in modelling of machining process suggest that these values should be kept high for the smaller sets of data in the size range of 10 to 20. The size of the models evolved is influenced by the maximum number of genes and maximum depth of the gene. For exerting control over the complexity (size) of the models, these two variables should be carefully chosen. A GUI for software GPTIPS as shown in Fig. 3.12 is developed by the author to facilitate easier settings of the variables and thus making it user friendly. Flexibility of using different fitness functions, testing generalization ability (extrapolation and interpolation) of model on testing (unseen) data, creating full factorial design of experiment from the user defined data, etc are some of the key features of the developed GUI.

The screenshot displays the GPTIPS GUI with the following components:

- Buttons:** Data, Settings, Start, Set, Clear All, Default, Refresh.
- Input Fields:**
 - No. of Generation: 100
 - Crossover Rate: 0.85
 - Size of Population: 100
 - Mutation rate: 0.1
 - No. of max. runs: 160
 - Reproduction rate: 0.05
 - range of random constant: [-10 10]
 - Fitness Function: RMSE (dropdown)
 - No. of max. gene: 3
- Settings Chosen Panel:**
 - generation = 100
 - population = 100
 - mutation rate = 0.1
 - crossover rate = 0.85
 - no. of run = 160
 - random constant range = [-10 10]
 - no. of gene = 3
 - reproduction rate = 0.05
 - fitness function = RMSE
 - threshold fitness = 10e-16
 - no. of runs required to reach threshold(%) = 100
- Tick to select functions:**
 - ☒ times ☒ minus ☒ plus ☒ divide
 - ☐ protected square root ☐ protected natural log
 - ☐ square ☐ exp ☐ ifLTE ☐ p. divide
 - ☐ tanh ☐ sin ☐ cos ☐ tan
 - ☐ Power
- Criteria for stopping algorithm:**
 - Threshold Fitness: 10e-16
 - % of Runs to reach threshold: 100

Fig. 3.12 GUI for implementation of MGGP and AMGGP

3.5.3 Phase 2: Integration of Two Approaches: M5' and AMGGP in Parallel

The CI approach is formulated by integrating two methods: M5' and AMGGP as shown in Fig. 3.13. The term error is defined as the difference between the experimental and predicted values of the M5' model. The sign of error (i.e. positive or negative) is taken into account. The error model is formulated using AMGGP, where the input fed into the AMGGP are the input and output process variables corresponding to the given system as well as the predicted values of the M5' model, while the output variable is the error of the predictions of the M5' model on training/validation data. In this way, there is one AMGGP model running in parallel with the M5' model and at every instance, the M5' model predicts the output, and the AMGGP model predict the error of the M5' model. By predicting the error of the M5' model, the AMGGP model ensures greater trustworthiness of the prediction ability of the M5' model on testing (unseen) data. Predictions obtained from the proposed CI approach are given by

$$Y_{Model} = Y_{M5'} + E_{AMGGP} \quad (3.11)$$

where, $Y_{M5'}$ is the output value obtained from the M5' model tree, E_{AMGGP} is the prediction of the error of the M5' model made by AMGGP model and Y_{Model} is the actual value predicted by an integrated approach.

Fig. 3.13 Formulation of CI approach M5'-AMGGP

In the following chapters, the AMGGP method as shown in Fig. 3.11 is implemented in stages, with Chapter 4 and 5 discussing the new procedures (1) and (2) adopted for tackling the issue of inappropriate formulation and complexity measure of the MGGP model respectively, and Chapter 6 discussing the procedure (3) adopted for tackling the problem of difficulty in model selection. Chapter 7 deals with the issue of ensuring greater trustworthiness of the prediction ability of the model on the testing samples by adopting the two CI methods of M5' and AMGGP in parallel. Chapter 8 shows the applications of proposed CI approach in modelling of various manufacturing processes. Chapter 9 concludes with recommendations for future work.

CHAPTER 4

A MODIFIED MULTI-GENE GENETIC PROGRAMMING MODEL USING STEPWISE APPROACH ORTHOGONAL BASIS FUNCTIONS

4.1 Methodology

The first step towards improving the generalization ability of the MGGP model includes the induction of a new procedure of the combination of genes in its paradigm. In MGGP method, the genes of the poor or the better performance on the training data are combined using the OLS method. With this mechanism, there is a possibility that a gene having the poor accuracy on training data can get combined with the other genes of higher performance and results in poor generalization ability of the MGGP model. There is a need for an approach that can selectively remove the poor performing genes and combine only the better performing ones. In this chapter, a modified MGGP (M-MGGP) approach is proposed in which, the statistical approach of stepwise regression is embedded in its paradigm. Step-by-step implementation of the M-MGGP approach is shown by the flowchart in Fig. 4.1, where, the dashed lines indicated by digit (1) represent the procedure of stepwise regression embedded to improve the mechanism of

combination of genes. Stepwise regression is a typical multiple regression method [149] that can select the best-fitted combination of independent genes for the dependent output value with forward-adding and backward-deleting genes. The idea of introducing the stepwise approach came from the statistical algorithms, where it is used to eliminate multicollinearity in data by removing the redundant or correlated input process variables. Stepwise procedure of combining the genes in formulation of the M-MGGP model is described in three steps as follows:

First, the model with no genes is chosen. Then, at each step along the way, the genes are entered or removed based on the partial F -test (t -tests). The significance level for deciding whether to enter a gene and remove a gene from the model is set based on a trial-and-error approach. In this work, the significance level value for entering as well as removing a gene is set at 0.05.

Step 1:

- (a) Let us assume that a total of 3 genes (G_1 , G_2 and G_3) are generated.
Regress output (y) on each generated gene i.e. y on G_1 , y on G_2 and y on G_3 .
- (b) Of those genes whose t -test P -value is less than 0.05, the first gene that enters the model is the one that has the smallest t -test P -value.
- (c) If none of the genes has the t -test P -value less than 0.05, then a new set of genes is generated.

Step 2:

- (a) Let us suppose that a gene G_1 has a t -test P -value below 0.05 and therefore it is the first gene to enter the model.

- (b) Now, perform regression on combination of first gene and others i.e. y on G_1 and G_2 and y on G_1 and G_3 .
- (c) Of those genes whose t -test P -value is less than 0.05, the second gene that is considered is the gene that has the smallest t -test P -value.
- (d) If none of the genes has the t -test P -value less than 0.05, then the model with only one gene is the final M-MGGP model.
- (e) But, suppose instead gene G_2 has the smallest t -test P -value, it enters the model consisting of gene G_1 .
- (f) Now, since gene G_1 was the first best gene in the model, step back and see if entering G_2 into the model has affected the significance level of the G_1 gene. if the t -test P -value is still less than 0.05, then the gene G_1 is retained otherwise G_1 has to be removed from the model.

Step 3:

- (a) Let us suppose that both genes (G_1 and G_2) have entered the model.
- (b) Now, perform regression on combination of the entered genes and the remaining gene i.e. regress y on G_1 , G_2 and G_3 .
- (c) If the gene G_3 has the t -test P -value value less than 0.05, then it enter the model.
- (d) Now, since genes G_1 and G_2 were the best genes in the model, step back and see if entering G_3 into the model has affected the significance level of the genes (G_1 and G_2). Suppose, if the t -test P -value is still less than 0.05, then the genes (G_1 and G_2) are retained.
- (e) Suppose, the gene G_3 has the t -test P -value value greater than 0.10 then it is rejected and the model with two genes (G_1 and G_2) is the

final M-MGGP model with estimated coefficients Q_1 and Q_2 (Fig. 4.2).

The stepwise procedure of combining the three genes can be applied for any number of genes. In MGGP, the genes are generated randomly and therefore for every set of genes, the stepwise procedure is adopted in the formulation of M-MGGP model.

Fig. 4.1 Step-by-Step implementation of M-MGGP approach

Fig. 4.2 M-MGGP model formulation using the stepwise regression approach

4.2 Problem: Turning Process of AISI H11 Steel

A primary operation in many production processes in industry is turning. Components produced during the turning operation have critical features that require specific surface finish [5]. Surface roughness is a widely used index for the measure of product quality. Hence, achieving the desired surface roughness is of prime importance for the functional behaviour of the component. Past studies reveal that the surface roughness depends on process variables such as tool geometry, cutting conditions and the properties of work piece. Economic factors such as cost and time could be regulated by appropriately selecting the input process variables. The identification of an appropriate input parameter setting requires an accurate computation of the surface roughness values. Therefore, the mathematical modelling of the turning process is a potential problem and a motivation for the researchers in reducing the total manufacturing cost of the engineering component [3]. With the extensive use of numerically controlled

equipment, the demand for optimizing the performance characteristics of the machining process has grown tremendously.

Considerable amount of research has been done in the prediction of surface roughness of the turning process [150-157]. Researchers have developed physics-based models to understand the behaviour of turning, but this may be a challenging task with the availability of only partial information about the process [3, 158-160]. In this perspective, several CI methods such as regression analysis, ANN, SVR and FL have been extensively applied in the prediction of surface roughness [161-172].

The regression analysis is based on statistical assumptions, and thereby induces uncertainty in the prediction ability of the model [170, 173, 174]. These models cannot be used to generalise the process data. ANN is well known for capturing the dynamics of the process. However, the optimal architecture of ANN is mostly determined either through a trial-and-error approach or by hybridising it with heuristic optimisation methods such as genetic algorithm (GA), particle swarm optimization, etc. This implies that one requires thorough knowledge in selecting the best architecture of ANN for its faster training [48, 157, 175-177]. SVR based on the principle of SRM, is known for providing the generalisation ability in the models. Least square-support vector machines (LS-SVM) variant of SVR has been mostly used for predicting the performance of the turning process [162, 178]. However, it does not provide explicit formulation between the important process variables, and gives output values in crisp form. FL models mostly used for modelling turning process are Mamdani and ANFIS [179-182]. The formulation of the FL model requires an expert knowledge to formulate the fuzzy rules. For the validation of methods, an additional set of experiments were conducted. This

involves high labour costs, which consequently increases the manufacturing cost of the product [183, 184].

The applications for the explicit formulation of the performance characteristics of the various process using the evolutionary approach, namely, GP have been on the rise [113, 114, 185-188]. The main advantage of GP over the regression analysis and other statistical modelling methods is that it generates functional expressions without assuming any prior form of the existing relationships. GP and its variants have been successfully applied for modelling the performance of various non-conventional production processes. The performance attributes measured were cutting force, surface roughness, tool wear, etc. Other variants of GP that uses the set of genes for the formulation of model is MGGP.

Despite remarkable capabilities of the MGGP method in the field of manufacturing [34, 79], it is found that during the combination of genes using the least squares method in its evolutionary stage, the occurrence of genes of lower performance can degrade the performance of the MGGP model. This limitation of the MGGP method has motivated the author to develop a M-MGGP method by sensibly selecting the relevant genes of higher performance for the combination mechanism.

In this section, the M-MGGP method is proposed and applied to the modelling of surface roughness of turning process. The objective of the present work is to compare the performance of the M-MGGP method to those of the standardized MGGP approach, SVR and ANN. The data on the turning process to be modelled is obtained from an earlier study conducted by Aouici et al. [189] on modelling and optimisation of hard turning of AISI H11 steel using RSM. The experiments

were performed in dry conditions using lathe type SN40C with a spindle power of 6.6 KW. The sample material used was AISI H11 hot work steel, which is often used for the manufacture of diecasting moulds, dies and helicopter rotor blades. The composition of the AISI H11 steel is shown in Table 4.1. The dimensions of the material is 80mm in diameter and is hardened to 50 HRC. The input variables considered were cutting conditions such as cutting speed, feed rate and cutting time and the output variable was surface roughness. The input variables with its low-centre-high levels used are shown in Table 4.2 [189].

Twenty-seven sets of data samples [189] as shown in Table 4.3 are considered in this study in the turning of AISI H11 steel. These data samples were collected based on L_{27} standard full factorial (FF) 3-level experimental design. Selection of an appropriate training and testing data set do affect the learning capacity of the CI methods. For the appropriate selection of the training and testing data set, K-S algorithm is used. This algorithm selects the training and testing data in such a way that the whole data set is distributed uniformly throughout the domain. In these applications, we conducted the preliminary analysis and found that it's worth using K-S algorithm, since its computational time for its implementation is less than that of cross-validation procedure. Table 4.3 shows the set of data samples after the K-S algorithm has been applied. The first twenty-one samples is the set of training data with the remaining as the set of six test samples. The test data samples are chosen for testing the generalization ability of a model, whereas, only the training data is used for formulating the models.

Table 4.1 AISI H11 steel composition

Composition	(Wt %)
C	0.35
Cr	5.26
Mo	1.19
V	0.50
Si	1.01
Mn	0.32
S	0.002
P	0.016
Other components	1.042
Fe	90.31

Table 4.2 Input variables used in turning process

Process input variable	Low (-1)	Centre (0)	High (+1)
Cutting speed (m/min)	120	180	240
Feed rate (mm/rev)	0.08	0.12	0.16
Cutting time (min)	7	14	21

Table 4.3 Experiments showing values of input process variables and surface roughness

Trial no.	Cutting speed (x_1)	Feed rate (x_2)	Cutting time (x_3)	Surface roughness (y)
1	120	0.08	7	0.263
2	120	0.08	14	0.295
3	120	0.08	21	0.315
4	120	0.12	7	0.820
5	120	0.12	14	0.838
6	120	0.12	21	0.888
7	120	0.16	7	0.572
8	120	0.16	14	0.748
9	120	0.16	21	0.874
10	180	0.08	7	0.218
11	180	0.08	14	0.274
12	180	0.08	21	0.347
13	180	0.12	7	0.395
14	180	0.12	14	0.432
15	180	0.12	21	0.494
16	180	0.16	7	0.751
17	180	0.16	14	0.813
18	180	0.16	21	0.901
19	240	0.08	7	0.249
20	240	0.08	14	0.222
21	240	0.08	21	0.280
22	240	0.12	7	0.976
23	240	0.12	14	1.200
24	240	0.12	21	1.440
25	240	0.16	7	0.830
26	240	0.16	14	1.359
27	240	0.16	21	2.240

4.2.1 Results and Discussion

4.2.1.1 Parameter Settings of CI Methods

The proposed M-MGGP method is implemented on MATLAB R2010b. After applying a trial-and-error approach, the parameter settings chosen is shown in Fig. 4.3. For fair comparison with the MGGP method, the parameter settings of MGGP are kept same as that of M-MGGP. Equations (4.1) and (4.2) represent the MGGP and M-MGGP models obtained respectively, where x_1 , x_2 and x_3 are the three input variables of the process and plog is protective natural log.

Fig. 4.3 GUI showing the parameter settings chosen for MGGP and M-MGGP

$$\begin{aligned} \text{Surface roughness}_{MGGP} = & 160.7691 + (-0.00034312) * (((x1) * ((x1) * ((x2) + \\ & (x2)))) * (((x2) * (\tan(x1))) * (x2))) + (-0.0038645) * (\tan((x1) * ((x2) + \\ & (x3)))) + (51.7764) * ((\tanh(\tanh(\tanh(x3)))) + (\exp(\tanh(x2)))) + \\ & (-532.8593) * (\exp(\tanh(\tanh(\text{plog}(\sin(x2)))))) + (-0.00057995) * ((x3) * \end{aligned}$$

$$\begin{aligned}
& (((\exp(\exp(x2))) * (x1)) * (\sin(x2)))) + (0.06718) * ((x3) * (((x2) + (x2)) * \\
& (\tan(x1)))) * ((\log(x3)) * ((\log(x3)) * (x2)))) + (0.19042) * (\log(x3)) + \\
& (0.13004) * (\tan((x1) * ((x2) + (x2)))) \tag{4.1}
\end{aligned}$$

$$\begin{aligned}
\text{Surface roughness}_{M-MGGP} = & 1.65 + (-0.083461) * ((\exp(\exp(\exp((x1) - \\
& (x1)))) + (\log(\cos(\exp((x1) - (x2)))))) + (1.9612e - 007) * (((\sin(x1)) - \\
& (((x1) * (x1)) * (x3)) * (\sin(x1)))) + (\sin(x1))) + (2.0915e - 006) * \\
& (((((\log(x1)) * ((x3) * (x2))) - (\cos((x3) * ((5.997739)))))) * ((x2) * (x1))) * \\
& (((x1) * (x2)) * (x3))) + (0.065342) * (((x2) * (x1)) * (x2)) + (\cos(x1))) + \\
& (1.221e - 006) * ((((\cos((x2) * ((5.997739)))) * (x1)) * ((x1) - (x2))) + \\
& (\tanh(\cos(x2)))) + (-0.33006) * (\cos((x1) - ((\sin(x1)) - (((x2) * (x1)) * \\
& (\sin(x1)))))) + (-0.031921) * (((x2) * ((x1) * (x2))) * (((x3) * ((x1) * (x2))) * \\
& (\sin(x1)))) * (((x2) * (x2)) * (x2))) + (7.1069e - 005) * (((x2) * (x3)) * (((x2) * \\
& ((x1) * (x2))) * (\sin(x1)))) * (((x1) * (x2)) * (x3))) \tag{4.2}
\end{aligned}$$

SVR is implemented using LS-SVM tool box [190] and applied on the data set shown in Table 4.3. Choice of the kernel function in SVR has a significant impact on its generalisation ability. In this chapter, a radial basis function (RBF) kernel function is selected for the analysis, since it is best capable for faster training and been widely used. The parameters λ and σ of RBF are computed by a hybrid approach of coupled simulated annealing (CSA) and a grid search method. The values λ and σ are estimated using CSA, and then these are passed to the grid search method which uses the cross-validation procedure to determine the final values. A SVR model with optimal variables $\lambda = 141901.95$ and $\sigma = 0.031$ are found at the second iteration.

For the implementation of ANN, a three-layer feed forward ANN is used with back propagation (BP) learning and implemented in MATLAB R2010b. The parameter settings used for ANN is shown in Table 4.4. The architecture of ANN is selected based on a trial-and-error approach. Fig. 4.4 shows that for the number

of neurons 9, the value of RMSE is minimum on the training data, and therefore the network of 3-9-1 as shown in Fig. 4.5 for the ANN model is selected.

Table 4.4 Parameter settings for the three-layer ANN

Variables	Values assigned
Number of training data samples	21
Number of testing data samples	6
Number of hidden layer	1
Number of neurons in hidden layer	2-9
Activation function	Sigmoid
Number of epochs	3000
Learning rate	0.70
Architecture selection	Trial-and-error
Target goal mean square error	10^{-5}
Minimum performance gradient	10^{-5}

Fig. 4.4 RMSE obtained by ANN models by varying the number of neurons in the hidden layer

Fig. 4.5 Architecture of ANN determined based on trial-and-error approach

4.2.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the four CI methods (MGGP, M-MGGP, SVR and ANN) are illustrated in Figs. 4.6-4.9 on training and testing data respectively. The best prediction method that gives good generalization ability is determined by comparing these methods using the five metrics: Coefficient of determination (R^2), mean absolute percentage error (MAPE), RMSE, relative error (%) and multi-objective (MO) error. The MO function was proposed by Gandomi et al. [191], in which it was pointed out that R^2 value doesn't change significantly by shifting the output value of the model equally. In addition, the functions such as MAPE and RMSE only show the error and no correlation. Therefore, the criteria should also be a combination of R^2 , MAPE and RMSE, which was then formulated as a MO function. The mathematical representation of the metrics is as follows:

$$R^2 = \left(\frac{\sum_{i=1}^n (A_i - \bar{A}_i)(M_i - \bar{M}_i)}{\sqrt{\sum_{i=1}^n (A_i - \bar{A}_i)^2 \sum_{i=1}^n (M_i - \bar{M}_i)^2}} \right)^2 \quad (4.3)$$

$$MAPE(\%) = \frac{1}{n} \sum_i \left| \frac{A_i - M_i}{A_i} \right| \times 100 \quad (4.4)$$

$$\text{Relative error} = \frac{|M_i - A_i|}{A_i} \times 100 \quad (4.5)$$

$$\text{Multiobjective error (\%)} = \frac{MAPE + RMSE}{R^2} \quad (4.6)$$

where M_i and A_i are predicted and actual values respectively, $\overline{M_i}$ and $\overline{A_i}$ are the average values of predicted and actual respectively and n is the number of training samples.

The results of training phase shown in Fig. 4.6a, Fig. 4.7a, Fig. 4.8a and Fig. 4.9a indicate that all the four models have impressively learned the non-linear relationship between the input variables and surface roughness with high correlation values and relatively low error values. The results of testing phase shown in Fig. 4.6b, Fig. 4.7b, Fig. 4.8b and Fig. 4.9b indicate that the proposed M-MGGP model outperformed the other three with the value of R^2 achieved is as high as 0.95. Among MGGP, SVR and ANN, ANN has shown better performance. Between MGGP and SVR, MGGP has shown better performance.

MO error values for the four models are computed on training and testing data and shown in Table 4.5. It shows that the proposed M-MGGP method outperforms the other three methods. The descriptive statistics of the relative error (%) of the four models are shown in Table 4.6, which illustrates error mean, std dev (standard deviation), SE mean (standard error of mean), lower confidence interval (LCI) of mean at 95%, upper confidence interval (UCI) of mean at 95%, median, maximum and minimum. The lower value of range (range = UCI-LCI) of the confidence intervals and remaining statistical metrics of the M-MGGP model indicates that its performance is better than those of the other three models. Among MGGP, SVR and ANN, ANN has shown better performance. Between MGGP and SVR, MGGP has shown better performance.

Hypothesis tests are also used to compare the goodness of fit of the four models. These tests are conducted for the model formation process and the results are

shown in Table 4.7. These are t -tests to test the mean and f -tests for variance. For the t -test and f -test, the p -values of all the four models are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these four models differ. Therefore, all four prediction models have statistically satisfactory goodness of fit from the modelling point of view.

In addition to the comparison between M-MGGP and MGGP based on the generalisation criterion, the complexity of the best model for M-MGGP and MGGP methods evolved in each run is measured by estimating the number of nodes of the model. The bar graph (see Fig. 4.10 and Fig. 4.11) plotted for 20 best models (1 model evolved in each run based on minimum RMSE) shows that the models evolved by the M-MGGP method possesses lower complexity when compared to the MGGP models.

Thus from the statistical comparison presented, it can be concluded that the proposed M-MGGP model has outperformed the other three models.

Fig. 4.6 Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data

Fig. 4.7 Comparison of simulated M-MGGP model and experimental values on (a) training and (b) testing data

Fig. 4.8 Comparison of simulated SVR model and experimental values on (a) training and (b) testing data

Fig. 4.9 Comparison of simulated ANN model and experimental values on (a) training and (b) testing data

Table 4.5 Multi-objective error of the four models

Methods	Training data	Testing data
MGGP	9.68	16.79
M-MGGP	3.31	7.37
SVR	7.38	17.67
ANN	9.16	14.12

Table 4.6 Descriptive statistics for relative error (%) of the four models

Model s	Coun t	Mea n	Std dev	LCI 95 %	UCI 95%	SE mea n	Media n	Maximu m	Minimu m
MGGP	27	10.18	15.25	4.14	16.22	2.93	0.44	59.72	0
M-MGGP	27	3.49	4.49	1.72	5.27	0.86	1.65	17.85	0.05
SVR	27	8.84	7.22	5.98	11.70	1.39	8.14	24.94	0.23
ANN	27	9.41	10.73	5.17	13.66	2.06	6.76	33.55	0.00

Table 4.7 Hypothesis testing to compare the four prediction models

95%CI	p-value			
	MGGP	M-MGGP	SVR	ANN
Mean paired <i>t</i> -tests	0.1151	0.1129	0.2208	0.2140
Variance <i>f</i> -test	0.2421	0.3926	0.0637	0.0921

Fig. 4.10 Bar graph showing complexity of models evolved using MGGP method

Fig. 4.11 Bar graph showing complexity of models evolved using M-MGGP method

4.2.1.3 Sensitivity and Parametric Analysis of the Proposed Model

The results discussed reveal that the M-MGGP outperforms the other three methods in prediction of surface roughness. To validate the robustness of the M-MGGP method, sensitivity and parametric analysis about its mean is conducted. The sensitivity analysis investigates the contribution of the input variables to the

surface roughness values. The sensitivity analysis (SA) percentage of the surface roughness to each input variable is determined using the following formulas [192]:

$$L_i = f_{\max}(x_i) - f_{\min}(x_i) \quad (4.7)$$

$$SA_i = \frac{L_i}{\sum_{j=1}^n L_j} \times 100 \quad (4.8)$$

where $f_{\max}(x_i)$ and $f_{\min}(x_i)$ are, respectively, the maximum and minimum values of the predicted output over the i th input domain, where the other variables are equal to their mean values.

Fig. 4.12 shows the plots of the sensitivity result of input variables in prediction of surface roughness. Obviously, the higher the sensitivity, the greater is the impact of given input variable on the output. From Fig. 4.12, it is obvious that the process feed rate has the highest impact on the surface roughness followed by cutting speed and time. Furthermore, the parametric analysis provides a measure of the relative importance among the chosen inputs of the M-MGGP model, and illustrates how the surface roughness varies in response to the variation in an input. On the M-MGGP model, the value of the first input is varied between its mean $\pm$ standard deviation value of 0.2, and the output value from the M-MGGP model is computed, while the other inputs are fixed at their respective means. This analysis is then repeated for the other inputs. Fig. 4.13 displays the plots generated illustrating the variation of each input variable and the surface roughness. These plots reveal that, the surface roughness is directly proportional to the cutting speed and feed rate. The results of this analysis are well in agreement with those mentioned in the study conducted by Aouici et al. [189]. The sensitivity and

parametric analysis confirms that the proposed M-MGGP model is capable of capturing the important characteristics and hidden non-linear relationships in the turning process.

Fig 4.12 Relative contribution of each input variable to surface roughness

Fig. 4.13 Variation of surface roughness with respect to each input variable

4.3 Problem: Turning of ASS 304 Steel Subjected to DSA Regime

In the field of defence and nuclear science, Austenitic Stainless Steel (ASS) 304 has found numerous applications because of its excellent mechanical properties such as corrosion resistance, improved friction and wear properties, etc. ASS Steel is widely used in nuclear reactors for the purpose of cladding of the fuel rods. Since the temperatures in reactors reach very high levels, it is essential to study the properties of the steel at such elevated temperatures. It is found that at the elevated temperatures, steel exhibits a peculiar behaviour with low strain rates. The mobility of the solute atoms can become large enough that they can follow a dislocation during its motion and segregate to its core while it has to wait in front of an obstacle. The repeated segregation and detachment process causes the flow stress to oscillate and as a result, wavy pattern like saw tooth in the stress-strain plot is observed. This effect is Dynamic Strain Aging (DSA) or Portevin-Le Chatelier effect. In addition, DSA is clearly indicated by the negative strain sensitivity, which results in unstable, jerky flow [193-199].

Constitutive models such as those based on phenomenological and physics were formulated to study the flow stress behaviour of a variety of materials for the different strain rates and ranges of temperature [200]. The phenomenological-based models such as Johnson Cook (JC) and modified Arrhenius type equations have been used to incorporate the negative strain sensitivity and an adiabatic rise in the temperature [201-205]. A physics-based model such as the modified Zerilli-Armstrong (m-ZA) has been used for different face-centered cubic and body-centered cubic materials over different strain rates for a particular range of temperature [206-208]. The m-ZA model is preferred over the JC model because

it takes into account the coupled effect of strain rate and temperature. However, it cannot be used to predict the flow behaviour of material at lower strain rates and beyond a particular temperature $0.6T_m$, where T_m is the melting point of the material [202, 204]. The phenomenological as well as physics-based models require the estimation of material constants in the equations [200]. The models were then used in finite element software to simulate the material behaviour under specific loading conditions [209]. The accuracy of these models largely depends on how accurately the deformation behaviour of material is being manifested by the model. However, these models hold a major assumption of its structure and its estimation involves tedious computations.

With this perspective, CI methods such as GP, ANN and SVR can be an alternative for formulating the models since they do not hold any assumption of model structure nor do they require any expert knowledge about the material behaviour [210-212]. CI methods produce models based on only the given data and possess the ability to capture the dynamics of the material behaviour under extreme conditions of temperature and strain rates. In particular, the GP method automatically evolves the model structure and its coefficients.

One of the most advanced variants of GP, namely the MGGP is considered to be a popular choice among researchers because unlike standard GP, in this method, each model participating in the evolutionary stage is a combination of several sets of genes/trees combined randomly using the least squares method. A number of applications of MGGP have been reported in the literature. Data obtained from the systems often suffer from the multicollinearity problem. Garg and Tai [18] compared the performance of MGGP to those of statistical methods such as partial least square regression, ridge regression and ANN in solving the multicollinearity

problems. It was found that MGGP possesses the ability to eliminate the redundant variables automatically. Empirical analyses were carried out to unearth the best experimental design and fitness function in modeling of time series system [79]. Despite recent advancements in MGGP, the issue of procedure of formation of the MGGP model has not yet been addressed [17, 210].

In this section, M-MGGP method is proposed and applied to the modelling of true stress of ASS 304 steel subjected to DSA regime. The performance of the proposed M-MGGP method is then compared with those of standardized MGGP, SVR and ANN methods. The experimental data on tensile testing of ASS 304 is obtained from Gupta et al. [197]. The composition of ASS304 used is given in Table 4.8. The experiments were performed on the flat specimens of thickness 0.9 mm. The dimensions of the specimen considered were as per the reduced version of ASTM: E8/E8M- 11 standards. Wire-cutting electro-discharge machining process was used for providing the high surface finish to the samples. Isothermal tensile tests were conducted on a computerized universal testing machine (UTM) having maximum load capacity of 100 kN in order to determine the flow stress behaviour as well as to study the DSA phenomenon. The UTM is equipped with a controlled system and resistance heating 3-zone split furnace to obtain constant true strain rates and to conduct the tensile tests at elevated temperatures up to 1000 °C respectively. The pull rods of UTM for the elevated temperature testing are made of Nickel base super alloy CM-247.

The experiments were conducted at four different strain rates (0.0001, 0.001, 0.01 and 0.1 s⁻¹) and various temperatures ranging from 623 K to 923 K at intervals of 50 K. The load vs. displacement data was converted into true stress and true plastic strain data.

For the data analysis part, the three input process variables of strain rates (x_1), temperature (x_2), and the true plastic strains (x_3) were considered and their values used are shown in Table 4.9. The output process variable considered is the measured true stress values (y). 225 sets of data samples are generated from the experiment. The nature of the data is shown by descriptive statistics in Table 4.10. The K-S algorithm is used to select first 80% (180 data samples) of the total samples as training data with the remaining forty-five used as a set of test samples

Table 4.8 ASS 304 composition

Composition	(Wt %)
Fe	70.78
C	0.025
Mn	1.140
Si	0.410
Mo	0.360
Ni	8.190
Co	0.210
Cr	18.40
Cu	0.180
Other components	0.305

Table 4.9 Input variables used for tensile test of ASS 304 operated at DSA regime

Process input variable	Values
Strain rates (s^{-1})	0.0001, 0.001, 0.01
Temperature (K)	723, 773, 823, 873, 923
Strain	0.02 to 0.30

Table 4.10 Descriptive statistics of the input and output process variables considered for tensile testing

Parameter	Strain rates (x_1)	Temperature (x_2)	Strain (x_3)	True stress(y)
Mean	0.0037	823.00	0.16000	0.37885
Standard error	0.0002	4.7200	0.00577	0.00879
Median	0.0010	823.00	0.16000	0.36572
Standard deviation	0.0048	70.860	0.08660	0.13192
Variance	0.00002	5022.3	0.00750	0.01740
Kurtosis	-1.5060	-1.3022	-1.21091	-0.83805
Skewness	0.6902	0	0.00000	0.42557
Range	0.0099	200.00	0.28000	0.54881
Minimum	0.0001	723	0.02000	0.16057
Maximum	0.0100	923	0.30000	0.36572

4.3.1 Results and Discussion

4.3.1.1 Parameter Settings of CI Methods

The proposed M-MGGP method is implemented on MATLAB R2010b. After applying a trial-and-error approach, the parameter settings chosen is shown in Fig. 4.14. For fair comparison with the MGGP method, the parameter settings are kept same as that of M-MGGP. Equations (4.7) and (4.8) represent the MGGP and M-MGGP models obtained respectively, where x_1 , x_2 and x_3 are the three input variables of the process.

Fig. 4.14 GUI showing the parameter settings chosen for MGGP and M-MGGP

$$\begin{aligned}
 \text{True stress}_{MGGP} = & -16.6642 + (11.7754) * (\sin(\cos((x_2) + (x_3)))) + \\
 & (5.9936) * (\cos(x_3)) + (-8.9697) * (\cos((x_2) + (x_3))) + (0.0066637) * (x_2) + \\
 & (-5.3956) * (\sin(((plog(\tan(x_2))) * (x_1)) - (\exp(\sin(x_2))))) + (101.5029) * \\
 & (((x_3) * (\tanh(\sin((-3.331954) + (x_2))))) * (x_1)) + (-0.70143) * \\
 & (\sin(\sin(((x_2) + (x_3)) + (x_2)))) + (2.1259) * (x_3)
 \end{aligned} \tag{4.7}$$

$$\begin{aligned} \text{True stress}_{M-MGGP} = & -58.8316 + (1.0074) * (\cos(\cos(\cos(\exp((x1) * \\ & (x2)))))) + (-77.3635) * (\cos(\tan(\cos(\text{plog}(x2)))))) + (0.25111) * \\ & (\cos((\tan((x2) - (x3))) * (\text{plog}(\tan(x2)))))) + (0.054003) * (\cos(\exp((x1) * \\ & (x2)))) + (0.0044348) * (((x1) * (x2)) - (x2)) * (x3)) + (4.9814) * \\ & (\tanh(\sin(\tanh(x3)))) + (0.097164) * (x2) + (0.003381) * (\tan((\cos((x1) * \\ & (x2)))) - (x2))) \end{aligned} \quad (4.8)$$

For the implementation of the SVR method, the kernel function chosen is the standard “RBF”. The parameters λ and σ of RBF are determined using a hybrid of CSA and a grid search method. Firstly, the CSA determines the good initial values of λ and σ , and then these are passed to the grid search method which uses cross-validation procedure to fine tune the variables. The final values of $\lambda = 2451557562.4$ and $\sigma = 57.31$ are estimated.

For the implementation of ANN, a three-layer feed forward neural network is used with BP learning and implemented in MATLAB R2010b. The parameter settings used for ANN is shown in Table 4.11. The optimal network of ANN is determined based on a trial-and-error approach adopted to select the number of neurons in the hidden layers. Fig. 4.15 shows that for the number of neurons 8, the value of RMSE is minimum, and therefore the network of 3-8-1 as shown in Fig. 4.16 for the ANN model is selected.

Table 4.11 Parameter settings for the three-layer ANN

Variables	Values assigned
Number of training data samples	80%
Number of testing data samples	20%
Number of hidden layer	1-2
Number of neurons in hidden layer	2-9
Activation function	Sigmoid
Number of epochs	3000
Learning rate	0.70
Architecture selection	Trial-and-error
Target goal mean square error	10^{-5}
Minimum performance gradient	10^{-5}

Fig. 4.15 RMSE obtained by ANN models by varying the number of neurons in the hidden layer

Fig. 4.16 Architecture of ANN determined based on trial-and-error approach

4.3.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the four CI methods (MGGP, M-MGGP, SVR and ANN) are illustrated in Figs. 4.17-4.20 on the training and testing data respectively. The best prediction method that gives good generalization ability is

determined by comparing these modelling methods using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The result of the training phase shown in Fig. 4.17a, Fig. 4.18a, Fig. 4.19a and Fig. 4.20a indicates that all the four models have impressively learned the non-linear relationship between the input variables and true stress with high correlation values and relatively low error values. The result of the testing phase shown in Fig. 4.17b, Fig. 4.18b, Fig. 4.19b and Fig. 4.20b indicates that the proposed M-MGGP model outperformed the other three models with the value of R^2 achieved as high as 0.93. Among MGGP, SVR and ANN, MGGP has shown better performance. Between SVR and ANN, both have shown comparable performance.

MO error values for the four models are computed on the training and testing data and shown in Table 4.12. It shows that the proposed M-MGGP model outperforms the other three models. The descriptive statistics of the relative error (%) of the four models are shown in Table 4.13. The lower value of range of the confidence intervals and remaining statistical metrics of the M-MGGP model indicates that its performance is better than those of the other three models. Among MGGP, SVR and ANN, MGGP has shown better performance. Between SVR and ANN, both have shown comparable performance.

Hypothesis tests are also used to compare the goodness of fit of the four models. These tests are conducted for the model formation process and the results are shown in Table 4.14. For the t -test, the p -values of the MGGP and M-MGGP models are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these models differ, whereas, for the ANN and SVR models, the p -values are less than 0.05, which indicates that there is

enough evidence to conclude that the predicted and actual values differ. For the f -test, the p -value of the M-MGGP is greater than 0.05 so there is not enough evidence to conclude that the actual values and predicted values from this model differ, whereas for the remaining three models, the p -values are less than 0.05, which indicates that there is enough evidence to conclude that the predicted and actual values differ.

In addition to the comparison between M-MGGP and MGGP methods based on generalisation criterion, the complexity of the best model for M-MGGP and MGGP methods evolved in each run is measured by counting its number of nodes. The bar graph (see Fig. 4.21 and Fig. 4.22) plotted for 20 best models (1 model evolved in each run based on minimum RMSE) shows that the models evolved by the M-MGGP method possesses lower complexity when compared to the MGGP models.

Thus from the statistical comparison presented, it can be concluded that the proposed M-MGGP model has outperformed the other three models

Fig. 4.17 Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data

Fig. 4.18 Comparison of simulated M-MGGP model and experimental values on (a) training and (b) testing data

Fig. 4.19 Comparison of simulated SVR model and experimental values on (a) training and (b) testing data

Fig. 4.20 Comparison of simulated ANN model and experimental values on (a) training and (b) testing data

Table 4.12 Multi-objective error of the four models

Methods	Training data	Testing data
MGGP	5.33	7.21
M-MGGP	6.08	5.37
SVR	7.45	9.67
ANN	5.55	9.99

Table 4.13 Descriptive statistics for relative error (%) of the four models

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
MGGP	225	7.95	6.76	4.65	7.24	0.45	3.44	22.79	0.03
M-MGGP	225	5.34	5.52	5.22	6.67	0.36	3.44	28.68	0.02
SVR	225	6.23	0.41	6.04	8.67	0.41	5.15	29.11	0.02
ANN	225	5.75	5.19	5.07	6.84	0.34	4.33	28.69	0.05

Table 4.14 Hypothesis testing to compare the four prediction models

95%CI	p-value			
	MGGP	M-MGGP	SVR	ANN
Mean paired <i>t</i> -tests	0.1457	0.1229	0.0114	0.0235
Variance <i>f</i> -test	0.0322	0.2954	0.0293	0.0468

Fig. 4.21 Bar graph showing complexity of models evolved using MGGP method

Fig. 4.22 Bar graph showing complexity of models evolved using M-MGGP method

4.3.1.3 Sensitivity and Parametric Analysis of the Proposed Model

The results discussed reveal that the M-MGGP outperforms the other three methods in prediction of true stress values. To validate the robustness of the M-MGGP method, sensitivity and parametric analysis about its mean is conducted. The sensitivity analysis investigates the contribution of the input variables to the true stress values. Fig. 4.23 shows the plots of the sensitivity result of input variables in prediction of true stress. It is obvious that the process temperature has the highest impact on the true stress followed by strain rates and strain. Furthermore, the parametric analysis provides a measure of the relative

importance among the chosen inputs of the M-MGGP model and illustrates how the true stress varies in response to the variation in an input. On the M-MGGP model, the value of the first input is varied between its mean $\pm$ standard deviation value of 0.2, and the output value from the M-MGGP model is computed, while the other inputs are fixed at their respective means. This analysis is then repeated for the other inputs. Fig. 4.24 displays the plots generated illustrating the variation of true stress with respect to each input variable. These plots reveal that the true stress is directly proportional to the strain rates. The results of this analysis generally comply with the perspective of materials science engineering and are in agreement with those mentioned in the study by Gupta et al. [197]. The sensitivity and parametric analysis confirms that the proposed M-MGGP model is capable of capturing the important characteristics of the turning phenomenon of dynamic aging regime of ASS 304.

Fig. 4.23 Relative contribution of each input variable to the true stress

Fig. 4.24 Variation of true stress with respect to each input variable

4.4 Summary

This chapter establishes the motive behind modelling of the two processes: turning of AISI H11 steel and turning of dynamic aging regime of ASS steel. In addition, the issue of procedure of formulation of the MGGP model is addressed. To counter this, the M-MGGP approach is proposed and its performance is compared to those of MGGP, SVR and ANN. The statistical comparison of the four models conducted concludes that the M-MGGP model has outperformed the other three models. This implies that it is able to generalise the surface roughness and true stress values accurately at different input process conditions. The sensitivity and parametric analysis of the M-MGGP model conducted to study the amount of impact and variation of output with respect to each input variable is very useful for monitoring and optimising the performance characteristics of the processes. Among MGGP, SVR and ANN, no conclusion can be drawn since their performance entirely depends on the nature of data as well as the parameter settings chosen for their implementation. The M-MGGP approach bears some similarity with the semantic crossover genetic operation. The similarities are that both these methods used MSE as a statistical measure to remove the insignificant gene from the population (in case of M-MGGP) and swapped the branch between genes (in case of semantic crossover). However, the difference lies in the fact that in case of M-MGGP method, a whole gene (made of several branches of a tree) is removed from population, whereas in case of semantic crossovers, the branch of trees are swapped based on MSE. It would be really interesting to investigate the comparisons of these methods and also there is a possibility that M-MGGP method can further incorporate the semantic crossover phenomenon to generate a simplified functional expression.

CHAPTER 5

ORTHOGONAL BASIS FUNCTIONS AS A COMPLEXITY MEASURE FOR MGGP MODELS IN REGULARIZED FITNESS FUNCTIONS

5.1 *Methodology*

As discussed in Chapter 2 (section 2.2.3), the complexity measure of the model used in the fitness functions of MGGP has a significant impact on the generalization ability of the formed models. The literature reveals that in fitness functions, complexity of the MGGP model is quantified by the number of nodes of the MGGP model. It is well argued that this measure is not an appropriate one. In this chapter, we propose two new complexity measures of the MGGP model built on orthogonal basis functions (a) polynomials and (b) multi-adaptive regression splines (MARS). These functions have well defined complexity and are useful as approximating functions to unknown nonlinear relationships. The brief description about these orthogonal basis functions is as follows:

(a) A polynomial is a function which can be written in a form [213]

$$p(x) = c_0 + c_1x + c_2x^2 \dots + c_nx^n, \quad (5.1)$$

where $c_0, \dots, c_n$ are coefficients determined using least squares method and x is the independent variable. If $c_n \neq 0$, then the polynomial is of order n .

- (b) The MARS algorithm splits the data into many splines on an equivalent interval basis. Furthermore, in every spline MARS splits the data into many subgroups and knots are created. To separate the subgroups, the knots can occur between different input variables or different intervals in the same input process variable. The data point of each subgroup is denoted by a basis function [214]. MARS fits the data in the following form [214]:

$$f(x) = \beta_0 + \sum_{i=1}^P \sum_{b=1}^B [\beta_{ib} (+) \text{Max}(0, x_i - H_{bi}) + \beta_{ib} (-) \text{Max}(0, H_{bi} - x_i)] \quad (5.2)$$

where x_i is the input variable, $f(x)$ is the predicted output value, P is the number of input variables and B is the basis function. $\text{Max}(0, x_i - H_{bi})$ and $\text{Max}(0, H_{bi} - x_i)$ are basis functions and would vanish if the value of their respective coefficients $\beta_{ib} (+)$ and $\beta_{ib} (-)$ is 0. H values are known as knots. MARS algorithm is implemented in three phases (a) a forward selection method for choosing the spline basis functions, (b) a backward elimination method to remove the basis functions until the best combination is found, (c) a smoothing criterion which gives the final MARS approximation. Basis functions are deleted based on the least contribution using the generalized cross-validation criterion (GCV).

$$GCV = \frac{\frac{1}{N} \sum_{i=1}^N [y_i - f(x_i)]^2}{\left[1 - \frac{C(B)}{N}\right]^2} \quad (5.3)$$

Where N is the number of data samples and $C(B)$ is a complexity penalty that increases with an increase in the number of basis function in the model. $C(B)$ is defined as:

$$C(B) = (B + 1) + dB \quad (5.4)$$

Where d is the penalty (smoothing parameter) for each basis function entering the model. More details about the origin and functioning of MARS can be found in [215, 216].

The complexity of the MGGP model is measured as the order of polynomial or the number of basis functions in MARS required for approximating it. The steps introduced in the evolutionary stage for measuring the complexity of the MGGP model are as follows:

Step 1: From the given data (x_i, y_i) , generate y_{mggp} , where x_i are the input variables values, y_i is the actual output value and y_{mggp} is the output value generated from the MGGP model.

Step 2: Fit the data (x_i, y_{mggp}) into orthogonal basis functions: polynomials and MARS of a given range of order.

Step 3: The order of polynomial or number of basis functions of MARS that best fits the data (x_i, y_{mggp}) is considered as the complexity of the model.

Thus:

Complexity of the MGGP model = Optimum order of polynomial/number of basis functions.

However, for appropriately measuring the complexity of the model, the range of order of polynomial and number of basis functions should be carefully selected. Based on the trial-and-error approach, the polynomial is chosen of the order in the range of 1 to 6, and the number of basis functions in the range of 1 to 21 including the intercept term. The rest of the parameter settings used for implementing the MARS is shown in Table 5.1. The proposed complexity measure is then incorporated into the widely used regularized fitness functions. By the incorporation of the complexity measure in these functions, new fitness functions (F_{new}) are thus formulated. Table 5.2 shows the old and formulated (new) fitness functions, F_{old} and F_{new} , which uses the three criteria of complexity measure: nodes of the MGGP tree (k), optimum order of polynomial (r) and number of basis functions (b). In Table 5.2, j equals to r or b depending upon the complexity measuring criteria used, E is the mean square error, SSE is the sum of square of error, m is the MGGP model, n is the size of the training set and σ is the standard deviation of the additive noise.

The present chapter conducts quantitative analysis to study the impact of fitness functions categorised under F_{old} and F_{new} on the performance of MGGP in modelling of the data obtained from the processes: machining (turning and drilling) and vibratory finishing process. The objective of the chapter is to unveil the fitness function that gives the best generalization ability of the MGGP models. The findings will result in development of a new MGGP approach as shown in Fig. 5.1, where the new fitness function is the function that is obtained from the current analysis.

Table 5.1 Parameter settings of MARS

Variables	Values assigned
Maximum number of basis functions to be included in forward building phase	21
Type of modelling	Piecewise-cubic
Self-interactions	1
Cubic fast level	2
Pruning	Yes
Maximum Interactions	2
Threshold error	1E-4

Table 5.2 Fitness functions and their mathematical formulae

Fitness function	F_{old}	F_{new}
AIC	$E(m) + \frac{2k}{n}\sigma^2$	$E(m) + \frac{2j}{n}\sigma^2$
BIC	$E(m) + \frac{\ln(n)k}{n}\sigma^2$	$E(m) + \frac{\ln(n)j}{n}\sigma^2$
JEW	$\frac{E(n-k)}{(n-2k-1)}$	$\frac{E(n-j)}{(n-2j-1)}$
FPE	$E(m)\left(\frac{n+k}{n-k}\right)$	$E(m)\left(\frac{n+j}{n-j}\right)$
PRESS	$\frac{SSE}{n}\left(1 + \frac{2k}{N}\right)$	$\frac{SSE}{n}\left(1 + \frac{2j}{n}\right)$
SRM	$\frac{SSE}{n} \left(1 - \left(\sqrt{\left(\frac{k}{n} - \frac{k}{n} \text{Log}\left(\frac{k}{n}\right) + \frac{\text{Log}\left(\frac{k}{n}\right)}{2n} \right)} \right) \right)^{-1}$	$\frac{SSE}{n} \left(1 - \left(\sqrt{\left(\frac{j}{n} - \frac{j}{n} \text{Log}\left(\frac{j}{n}\right) + \frac{\text{Log}\left(\frac{j}{n}\right)}{2n} \right)} \right) \right)^{-1}$

Fig. 5.1 Formulation of new MGGP approach by using new fitness function

5.2 *Problem: Machining Processes such as Turning and Drilling*

The problem of modeling of turning process is discussed in Chapter 4. In this section, the motivation of choosing the drilling process is discussed. Furthermore, the author summarizes the literature study on both the processes, turning and drilling.

Drilling is used in producing cylindrical holes in work piece. The input variables of the drilling process usually considered are depth of cut, cutting speed, feed, etc. The output process variables can include tool-wear, tool-life, chip-form, burr height, burr thickness and part accuracy [217]. These process variables may be

optimized for obtaining minimum cost and minimum production time. However, for the selection of appropriate input variables settings, the output process variable values must be estimated accurately. This has indeed been a motivation and a potential problem for researchers in purpose of reduction of total cost of the product [3]. With the development and extensive usage of numerically controlled CNC machines, the demand for process modeling and optimization has grown tremendously.

Similar to turning process, considerable amount of research has also been done in the prediction of output process variables values of the drilling process [46, 157, 218-221]. Researchers have developed physics-based models, but this may be a challenging task in the availability of partial information about the drilling process [159, 222-224]. Focus was shifted towards the applications of CI methods, since these methods can formulate models based on only the given data. Several CI methods such as regression analysis, ANN, SVR, FL and GP have been extensively applied.

After extensively reviewing the literature studies on modelling strategies for machining processes [3, 16, 26, 34, 152, 225-227], the author have summarized the characteristics of modelling methods applied to model the machining processes such as turning and drilling in Table 5.3. From this table, it is obvious that GP possesses the ability to evolve model structure and coefficients automatically.

Table 5.3 Summary of applications of CI methods in modelling of machining processes

CI methods	GP	ANN	FL	SVR	Regression
Parameter Selection	Trial and error	Trial and error or using hybrid GA-NN, FL-NN, methods, etc.	Expert knowledge involvement or using hybrid GA-FL method.	Trial and error or optimization method (simplex, SA or GA).	N.A
Interpretation	No	No	Yes	No	Yes
Statistical assumption	No	No	No	No	Yes
Modelling category	Black box	Black box	White box	Black box	Grey box
Reproducibility	No	Yes	Yes	Yes	Yes
Generalization ability	Yes	Yes	Yes	Yes	No
Significant application	Milling	Turning, drilling	Turning, Milling	Turning	Turning, drilling

Significant amount of research has been done in developing variants of GP such as GEP, multi expression programming (MEP), MGGP, probabilistic genetic programming, methods of ensembles, and so on to improve the efficacy and efficiency with which GP represents models and traverses the space of models and parameter values. Among these methods, MGGP, which uses multiple set of genes for the formulation of models, has been used widely. MGGP have also been hybridized with other potential supervised learning methods such as linear least squares and artificial neural networks. While this means that more and more processes can be represented and potentially found with MGGP, the potential for *overfitting* also increases.

Overfitting occurs when a model approximates the training data accurately but produces large errors when applied to new data, usually due to the modelling of noise in training data. An overfit model is said to have poor generalization ability,

i.e. it does not represent the general underlying process. Generalization ability is also crucial for process monitoring and control under extreme process conditions. Given the flexibility and representation and search power of MGGP, it is relatively trivial to construct a model that approximates training data arbitrarily closely. Such models will be misleading and/or useless in practice unless the entire possible input space is captured in the training data.

In this chapter, MGGP is employed for the modelling of surface roughness and burr height of the turning and drilling process respectively. This work proposes the use of orthogonal basis functions in measuring the relative complexity of the population of models evolved in a MGGP. Polynomial basis functions such as polynomials and MARS in particular are employed because of their ability to approximate a continuous function with arbitrary accuracy. Orthogonality is introduced as it allows an unambiguous measure of complexity. The resulting complexity measure is incorporated into well-known regularized forms, and tested on two machining processes with the demonstration of performance improvements.

5.2.1 Turning process

Twenty-seven data points are obtained from an earlier study conducted on modelling of hard turning process for the prediction of surface roughness using regression analysis by Aouici et al. [189]. Experimental details of the process are discussed in Section 4.2. The input process variables considered are cutting speed, feed rate and cutting time. The output parameter considered is surface roughness. The K-S algorithm is used to divide the data set into training and testing samples.

Purposefully, to test the generalization ability of the models, first fourteen samples have been chosen as the set of training data with the remaining thirteen used as the set of test samples.

5.2.2 Drilling process

The experimental details of the drilling process are obtained from an earlier study conducted by Karnik and Gaitonde [220] on the effect of process variables on the burr size of the drilling. The drilling experiments were performed on three-axis ‘YCMV116B’ CNC vertical machining center (Make: Yeong Chin Machinery Industries Co., Taiwan). The machine is equipped with a maximum feed of 5,000 mm/min and a variable spindle speed from 45–4,000 rpm with a 15-kW drive motor. The maximum table travel along X-axis is 1,100 mm and along the Z-axis is 630 mm. The maximum saddle travel along the Y-axis is 600 mm. AISI 316L stainless-steel workpieces 25 mm thick were used for all the drilling experiments. Five input process variables, namely cutting speed, feed, drill diameter, point angle and lip clearance angle were considered. The output parameter considered is burr height. Box-Behnken quadratic design was used to perform experiments. Three levels are defined for each of the five process variables as shown in Table 5.4. The upper level of a process parameter is coded as +1, the center level as 0, and the lower level as –1. Descriptive statistics showing the nature of the data is given in Table 5.5. The K-S algorithm is used to divide the data set into training and testing samples. Purposefully, to test the generalization ability of the models, 60% of the samples (first twenty-eight samples) have been chosen as the set of training data with the remainder eighteen used as the set of test samples.

Table 5.4 Process input variables of drilling process and their respective values

Process input Parameter	Low (-1)	Centre (0)	High(+1)
Cutting speed (m/min)	8	12	16
Feed (mm/rev)	0.04	0.08	0.12
Drill diameter (mm)	10	16	22
Point angle (degree)	118	126	134
Lip clearance angle (degree)	8	10	112

Table 5.5 Descriptive statistics of the process variables used in drilling process

Parameter	Cutting speed	Feed	Drill diameter	Point angle	Lip clearance angle	Burr height
Mean	12	0.083	15.97	126	10	0.26
Standard deviation	2.38	0.02	3.54	4.77	1.19	0.13
Median	12	0.08	16	126	10	0.229
Standard error	0.35	0.004	0.522	0.70	0.17	0.01
Variance	5.68	7.6E-4	12.55	22.75	1.42	0.017
Kurtosis	0.002	0.245	0.020	0.002	0.002	-0.61
Skewness	0	0.3008	-0.028	0	0	0.679
Minimum	8	0.04	10	118	8	0.056
Maximum	16	0.16	22	134	12	0.538

5.2.3 Results and Discussion

5.2.3.1 Parameter Settings of MGGP

After applying a trial-and-error approach, the parameter settings chosen is shown in Fig. 5.2. A broader set of functions is chosen in the functional set since this can provide numerous forms of non-linear mathematical models. The population size and number of generations fairly depends on the nature of the problem. The population size and number of generations should be set to the high values for the data with fewer samples. Based on the current study, the population size is set to a value of 1000 and number of generations is set at 300. The size of search space and number of models searched within the space are directly influenced by the maximum number of runs and maximum depth of the gene respectively. Therefore, the maximum number of runs and maximum depth of gene is kept at 30 and 8 respectively.

The impact of fitness functions categorised under F_{old} and F_{new} on the performance of MGGP in modelling of the data obtained from the machining processes (turning and drilling) is studied. The performance of the MGGP model is computed across different fitness functions in terms of RMSE. The model selected based on minimum training error and the top 10% models with best performance on training data are used as the basis for evaluating the performance among the fitness functions.

Fig. 5.2 GUI showing the parameter settings chosen for MGGP

5.2.3.2 Evaluation and Comparison of Fitness Functions for Turning process

From Table 5.6 and 5.7 (values in bold in Tables indicates the best performance among three complexity measures for each fitness function), it is obvious that the FPE fitness function using optimum order of polynomial as a complexity measure

of the model gives the best generalization ability of the MGGP model when compared to other fitness functions. Among the remaining fitness functions the SRM fitness function using optimum order of polynomial as a complexity measure gives better generalization ability. For fitness functions FPE and PRESS, the models using optimum order of polynomial and number of basis functions of MARS as a complexity measures gives comparable performance. For the PRESS fitness function, the number of nodes used as complexity measure has given better generalization ability when compared to the fitness functions using the other two complexity measures.

Overall, the poor performance of MGGP models is shown by the fitness functions using number of nodes as a complexity measure of the model, whereas, the best performance is shown by fitness functions using optimum order of polynomial as a complexity measure.

Table 5.6 Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the MGGP model with minimum training error

No	$F_{old}(k)$	RMSE for $F_{old}(k)$	RMSE for $F_{new}(r)$	RMSE for $F_{new}(b)$
1	AIC	10.66	2.14	4.50
2	BIC	5.78	1.54	1.83
3	JEW	1.47	1.05	1.30
4	FPE	2.97	0.54	0.59
5	PRESS	1.02	1.58	1.55
6	SRM	1.89	0.98	1.10

Table 5.7 Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the top 10% MGGP models

No	$F_{old}(k)$	RMSE for $F_{old}(k)$	RMSE for F_{new} (r)	RMSE for $F_{new}(b)$
1	AIC	3.77	2.13	5.43
2	BIC	8.41	5.57	6.78
3	JEW	3.58	2.72	6.56
4	FPE	3.71	3.14	3.00
5	PRESS	2.87	2.55	2.81
6	SRM	5.44	2.22	3.51

5.2.3.3 Evaluation and Comparison of Fitness Functions for Drilling process

From Table 5.8 and 5.9, it is obvious that the SRM fitness function using optimum order of polynomial as a complexity measure of model gives the best generalization ability of the MGGP model when compared to the other fitness functions. Among the remaining fitness functions the FPE (b) and JEW (r) fitness functions give better generalization ability. For fitness functions FPE and PRESS, models using optimum order of polynomial and number of basis functions of MARS as a complexity measures gives comparable performance. For the PRESS fitness function, the number of nodes used as complexity measure has given better generalization ability when compared to the fitness functions using the other two complexity measures.

Overall, the poor performance of MGGP models is shown by the fitness functions using number of nodes as complexity measure of the model, whereas, the best

performance is shown by the fitness functions using optimum order of polynomial as a complexity measure.

Table 5.8 Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the MGGP model with minimum training error

No	$F_{old}(k)$	RMSE for $F_{old}(k)$	RMSE for F_{new} (r)	RMSE for F_{new} (b)
1	AIC	0.51	0.37	0.39
2	BIC	0.30	0.14	0.20
3	JEW	0.29	0.13	0.21
4	FPE	0.31	0.16	0.13
5	PRESS	0.18	0.21	0.19
6	SRM	0.14	0.11	0.22

Table 5.9 Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the top 10% MGGP models

No	$F_{old}(k)$	RMSE for $F_{old}(k)$	RMSE for $F_{new}(r)$	RMSE for $F_{new}(b)$
1	AIC	0.44	0.24	0.25
2	BIC	0.29	0.25	0.41
3	JEW	0.28	0.26	1.55
4	FPE	0.29	0.23	0.23
5	PRESS	0.28	0.23	0.25
6	SRM	0.15	0.15	0.25

5.3 Problem: Vibratory finishing process

Vibratory finishing is an important component of a group of industrial processes referred to as mass finishing or mass media finishing. It is an abrasive industrial process that could economically process metal or other materials in bulk to achieve surface effects such as deburring, descaling, surface smoothing, etc. The process utilizes a loose abrasive material referred to as media within a vibratory bowl. The vibratory bowl imparts motion to the media which causes it to rub or wear away at part surfaces. The vibratory finishing process is widely used because of its versatility, consistency and efficiency. This process gained its popularity in 1960 in the metal working sector [228].

Despite the availability of empirical information on the vibratory finishing process, its fundamentals are still mostly developed through trial-and-error approach [229]. Very few models exist that were developed to study the mechanism of vibratory finishing process [180, 230-232]. Hashimoto [47] formulated an empirical relationship between the output (surface roughness, stock removal rate and optimum polishing time) and input process variables (time and initial surface roughness) based on the fundamentals of the process. Sofronas and Taraman [233] used RSM to formulate the polynomial model that explicitly describes the relationship between five input (Brinell hardness, projection width, processing time, media size and frequency of bowl) and three output process variables (surface roughness reduction, projection height reduction, edge radiusing). In another application of RSM, relationship between five input (burnishing speed, feed, ball force, frequency and amplitude of vibration) and two output variables (surface roughness and micro hardness) was established [234].

With years, the main focus was on characterization of vibratory finishing process via experimental and numerical based studies [235-238]. Advanced CI methods such as MGGP, ANN, FL, etc., can also be used as an alternative since these methods are void of statistical assumptions and have the ability to capture the dynamics of the process.

Among these methods, the MGGP method evolves the model structure and its coefficients automatically. In the context of symbolic regression problems, extensive literature has been published but very few have paid attention towards defining the complexity measure of the MGGP model. Regularized fitness functions incorporate complexity of the MGGP model, which significantly influences the generalization ability of the model.

In the present chapter, a quantitative analysis is conducted for studying the impact of fitness functions on the performance of MGGP in modeling of vibratory finishing process. Surface finish reduction model is obtained from the study conducted by Sofronas and Taraman [233]. Three input process variables considered are processing time, media size and vibratory frequencies, and the output considered is surface finish reduction. Twenty-seven data points are generated from the model using the FF experimental design method. In FF design, the sample size of data points generated is of the order n^m , where m is the number of input process variables and n is the number of levels (low, medium, high, etc). In this study, the value of m and n is three, and default settings in MATLAB R2010b are used for its implementation. The nature of the data set is shown by descriptive statistics in Table 5.10. Purposefully, to test the generalization ability of the models, fourteen samples have been chosen as the set of training data with the remaining thirteen as the set of test samples.

Table 5.10 Descriptive statistics of the data set generated using FF experimental design

Parameter	Processing time	Media size	Vibratory frequencies	Surface finish reduction
Mean	70	0.875	1743.3	22.88
Standard error	7.33	0.04	40.07	1.77
Median	60	0.875	1730	21.57
Standard deviation	38.12	0.20	208.23	9.23
Kurtosis	-1.56	-1.56	-1.50	-0.33
Skewness	0.404	0	0.103	0.66
Minimum	30	0.625	1500	9.94
Maximum	120	1.125	2000	44.20

5.3.1 Results and Discussion

5.3.1.1 Parameter Settings of MGGP

After applying a trial-and-error approach, the parameter settings chosen is shown in Fig. 5.3. The performance of the MGGP model is measured across different fitness functions in terms of RMSE. The model with minimum training error and the top 10% models are used as the basis for evaluating the performance among the fitness functions.

Fig. 5.3 GUI showing the parameter settings chosen for MGGP

5.3.1.2 Evaluation and Comparison of Fitness Functions

From Table 5.11 and 5.12, it is obvious that the SRM fitness function using optimum order of polynomial as a complexity measure of model gives best generalization ability of the MGGP model when compared to other fitness functions. Among the remaining fitness functions the AIC fitness function using optimum order of polynomial as a complexity measure gives better generalization ability. For fitness function FPE, the models using optimum order of polynomial and number of basis functions of MARS as a complexity measures give comparable performance. For the PRESS fitness function, the number of nodes used as complexity measure has given better generalization ability when compared to the fitness functions using the other two complexity measures.

Overall, the poor performance of MGGP models is shown by fitness functions using number of nodes as complexity measure of the model and best performance

is shown by fitness functions using optimum order of polynomial as a complexity measure.

Table 5.11 Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the MGGP model with minimum training error

No	$F_{old}(k)$	RMSE for $F_{old}(k)$	RMSE for $F_{new}(r)$	RMSE for $F_{new}(b)$
1	AIC	2.56	1.78	3.65
2	BIC	25.74	8.15	7.35
3	JEW	25.74	6.74	15.65
4	FPE	24.19	2.76	2.04
5	PRESS	4.50	4.75	8.19
6	SRM	2.03	0.78	2.88

Table 5.12 Comparison of fitness functions using number of nodes, optimum order of polynomial and number of basis functions of MARS as a complexity measure of the top 10% MGGP models

No	$F_{old}(k)$	RMSE for $F_{old}(k)$	RMSE for $F_{new}(r)$	RMSE for $F_{new}(b)$
1	AIC	4.85	3.80	6.93
2	BIC	27.77	11.96	12.36
3	JEW	26.24	9.00	24.41
4	FPE	28.24	16.61	11.71
5	PRESS	6.20	11.08	8.66
6	SRM	3.95	3.83	5.41

5.4 Summary

This chapter establishes the need and importance of generalized models in optimizing the performance of manufacturing, machining and vibratory finishing processes. The MGGP method which evolves the model structure and coefficients automatically is a potential alternative. However, the generalization of MGGP models exhibit high dependence on its complexity measure. In this perspective, the vital issue of defining the complexity measure of the MGGP model is addressed. Two new measures of complexity have been proposed using the two orthogonal basis functions: polynomials and MARS. An application of MGGP in modelling of two processes: machining (turning and drilling) and vibratory finishing is introduced to study the impact of the fitness functions using the three complexity measures: number of nodes, polynomials and MARS on its performance. The overall results conclude that the fitness function SRM using optimum order of polynomial have shown better performance of the MGGP models than those of the other fitness functions.

CHAPTER 6

CLASSIFICATION-DRIVEN MODEL SELECTION

APPROACH OF MULTI-GENE GENETIC PROGRAMMING

6.1 *Methodology*

As discussed in Chapter 2 (section 2.2.3), the difficulty in model selection is one of the important issues in MGGP. With this perspective, a classification-driven model selection approach of genetic programming (C-MGGP) is proposed as shown in Fig. 6.1. The classification scheme embedded in the paradigm of MGGP to drive the selection of the best model is implemented in six steps as follows:

Step 1: The MGGP algorithm is implemented with chosen parameter settings.

Step 2: For every run, the best model is selected from the last population based on the minimum validation error. Attributes (training, validation error and number of nodes of the models) of the best model for every run is obtained.

Step 3: Based on validation error of the model, the category “bad” or “best” to the model for every run is assigned. Top 15% models based on the minimum validation error are labelled as “best”, with the remaining labelled as “bad”.

Step 4: With the given attributes (training, validation error and number of nodes of the models) and output (class of the models), the classification methods are applied and classifiers are formed.

Step 5: Implement the MGGP algorithm with embedded classifiers.

Step 6: The model classified as “best” by the most number of classifiers is selected and its performance is evaluated on the testing data.

In this proposed methodology, several classification methods such as SVM, ANN, Naïve Bayes classifier, partition and regression trees (PART), classification and regression trees (CART), decision trees, etc can be used. Details about these classification algorithms can be found in [75]. The purpose of using a set of classification methods instead of just one method is to alleviate the uncertainty in accuracy of any one particular classification method. The embedded classification methods make use of valuable information such as training, validation error and the number of nodes of the evolved MGGP models. The number of nodes (i.e. complexity of the model) has a significant impact on the performance of the model and therefore is considered. The classification criteria based on the validation error is formulated to assign these models into the two categories of “best” or “bad”. The top 15% of the models with least training/validation error are categorised as “best” models with the others categorised as “bad”. The classification methods are trained on this set of information (training error, validation error, number of nodes and categories of models) to form the classifiers. In this way, the set of classifiers formed classifies the models evolved in MGGP into the two categories of “best” and “bad”. The model classified as “best” by the most number of classifiers is selected and its performance is evaluated on the testing data. In the following sections, the application of the proposed C-MGGP approach is shown in modelling the data obtained from two processes: FDM and turning process.

Fig. 6.1 Schematic flowchart of the C-MGGP methodology showing the classification methods (inside dashed line)

6.2 Problem: Fused deposition modeling process (FDM)

The demand for shorter product development time has resulted in the emergence of a new paradigm called rapid prototyping (RP). RP relates to the rapidly growing number of additive manufacturing processes such as stereolithography, selective laser sintering, FDM, laminated object manufacturing, ballistic particle manufacturing and three dimensional printing which can fabricate 3-D solid models from the computer-aided design data automatically without involving tooling and human intervention [239-243]. Additive manufacturing processes disperse a 3D CAD model into a series of two-and-half dimensional slice models

with the corresponding software. The details about various types of additive manufacturing processes and the comparison of their characteristics are discussed in Yan and Gu [244] and Pham and Gault [245].

Simplicity of operation, inexpensive machinery, and durability of parts has resulted in its widespread applications not only in prototyping but also for making functional parts. However, additive manufacturing processes have their own drawbacks related to the properties of a prototype such as poor dimensional accuracy, low strength, poor surface finish, long build time, etc. One possible route may be to develop the materials having superior characteristics than the conventional materials and its compatibility with the technology. Another possible route may be to properly adjust the process variables during the fabrication stage so as to improve the properties of a prototype. The literature reveals that the properties of an additive manufacturing prototype exhibit high dependence on the process variables and are improved by its appropriate adjustment [116]. However, for optimization, the output variables of the additive manufacturing process need to be predicted accurately. Hence, the modelling of additive manufacturing processes has attracted a number of researchers for the purpose of reduction of overall cost of the product. With the advent of capital intensive additive manufacturing machines, the need for modelling and optimization of the process is strengthened.

Researchers have made several attempts to improve the performance of additive manufacturing fabricated prototypes via formulation of the physics-based models followed by an optimization [246-250]. The formulation of the physics-based models requires in-depth understanding of the additive manufacturing process, which poses a great challenge. In the circumstance of incomplete or partial

information of the additive manufacturing processes, CI methods that formulate models based only on the empirical data can be applied. CI methods such as ANN, regression analysis and FL are extensively applied to formulate the mathematical relationship between the output and input variables of the various additive manufacturing processes [251-265].

CI formulated models are used to predict the behaviour of properties of an additive manufacturing fabricated prototype such as dimensional accuracy, surface roughness, strength (tensile, impact, flexural and compressive), and build cost based on the given input process variables. Regression models used are of linear and quadratic functions [251, 259]. In addition, the ANOVA model is used to identify the significant additive manufacturing process variables that affect the properties of a prototype [250, 266-272]. In this way, those process variables with minimum influence could be suitably modified to suit the economic considerations. The significant input process variables identified using an ANOVA model may not be the same for all the properties of a prototype. Since each property is vital so therefore there is a need for the formulation of a model that can predict the behaviour of all the properties based on the given input process variables. Thereby optimization of this model can optimize the entire additive manufacturing process successfully. Very few researchers have made an attempt to form a model that relates all the additive manufacturing fabricated prototype properties with the process input variables [21, 269]. Regression analysis and the model ANOVA exhibit high dependence on statistical assumptions such as the strong assumption on model structure (linear or nonlinear). Thus, the choice of structure of a model and its dependency on statistical assumptions induce uncertainty in the prediction ability of the model. In most of the applications of

ANN, the optimal settings are determined using the trial-and-error approach [273-276]. The optimal settings used for ANN is single hidden layer with number of neurons varying from three to eight. However there is no general agreement for it. The implementation of FL method requires the incorporation of expert knowledge to formulate fuzzy rules, which is difficult. Very few researchers have compared the performance of CI methods for modelling additive manufacturing processes [277], so the appropriate choice of CI method is still unknown.

Given the nonlinearity and the complexity of the additive manufacturing processes, an evolutionary approach such as MGGP, which evolves the functional expressions without assuming any prior form of the existing relationships, can be applied. Since MGGP uses multiple sets of genes for the formulation of model and is population based, it suffers from the problem of difficulty in selecting the final optimal model.

In this chapter, a C-MGGP method is proposed and applied to model the FDM process. In this approach, three classifiers: SVM, ANN and naïve Bayesian are embedded in the procedure. The performance of the C-MGGP approach is compared to that of the standardized MGGP approach. The data on the FDM process is obtained from the study by Sood et al. [116] conducted on the experimental investigation of the compressive strength of FDM fabricated prototype. The process input variables considered are as follows:

- (a) Layer thickness (x_1): It is the thickness that is deposited by the nozzle and which is also influenced by a kind of nozzle used.
- (b) Orientation (x_2): It is the inclination of a FDM part in a build platform with respect to axis X, Y and Z axis.

- (c) Raster angle (x_3): It refers to the direction of raster relative to the X-axis.
- (d) Raster width (x_4): It is the width of the raster patter.
- (e) Air gap (x_5): It refers to the space between two adjacent rasters on the same layer.

For each of the input process variables, values at three levels (low, centre and high) are considered. The respective values and levels of the input variables used are shown in Table 6.1. Other process variables such as part fill style, contour width, part interior style and visible surface are kept constant. The output process parameter considered is compressive strength (y) of the FDM-built 3D square prism of dimensions 10×10×30. Thirty-two data points have been generated using 2^5 half factorial unblocked design, 10 axial runs and 6 centre run [116]. Nature of the data set collected is shown by the descriptive statistics in Table 6.2. The K-S algorithm is used to divide the data set into training, validation and testing data. The training set comprises the first twenty samples, whereas, the remaining twelve are split equally between validation and testing data.

Table 6.1 Input variables of FDM process and their respective values

Input variable	Low (-1)	Centre (0)	High(+1)
Layer thickness (mm)	0.127	0.178	0.254
Orientation (degree)	0	15	30
Raster angle (degree)	0	30	60
Raster width (mm)	0.4064	0.4564	0.5064
Air gap (mm)	0	0.004	0.008

Table 6.2 Descriptive statistics of the input and output process variables of FDM process

Statistical parameter s	Layer thickness (mm)	Orientation (degree)	Raster angle (degree)	Raster width (mm)	Air gap (mm)	Compressive strength
Mean	0.1850	15	30	0.4548	0.004	12.54
Standard error	0.008	2.02	4.04	0.006	0.005	0.364
Median	0.178	15	30	0.4564	0.004	12.27
Standard deviation	0.0488	11.43	22.86	0.039	0.003	2.063
Kurtosis	-1.202	-1.223	-1.22	-1.328	-1.22	0.453
Skewness	0.352	0	0	0.056	-0.0002	0.328
Minimum	0.127	0	0	0.4064	0	7.44
Maximum	0.254	30	60	0.5064	0.008	16.98

6.2.1 Results and Discussion

6.2.1.1 Parameter Settings of CI Methods

After applying a trial-and-error approach, the parameter settings chosen for both the methods (MGGP and C-MGGP) is shown in Fig. 6.2. The fitness function used is SRM with optimum order of polynomial as the complexity measure of the model. The classifiers formed from the attributes (training error, validation error and number of nodes) of the models predict the class of the unlabelled C-MGGP models. From Table 6.3, the model corresponding to run 28 is classified as “best” by all three classifiers from the classification methods. Equations (6.1) and (6.2) represent the MGGP and C-MGGP models obtained respectively. The performance of these models is further evaluated on the testing data.

Fig. 6.2 GUI showing the parameter settings chosen for MGGP and C-MGGP

Table 6.3 Class of C-MGGP models classified by three classifiers

Run	Training RMSE	Validation RMSE	ANN	SVM	naïve Bayesian
1	0.60	1.88	Bad	Bad	Bad
2	0.32	2.44	Bad	Best	Bad
.	0.33	4.08			
.	0.31	3.96			
.	0.39	4.05			
28	0.34	0.20	Best	Best	Best
.	0.33	3.51			
.	0.40	1.73			
.	0.32	4.83			
60	0.36	3.39	Best	Bad	Best

$$\begin{aligned}
\text{Compressive strength}_{MGGP} = & 17.2629 + (2.62) * (\tan(\exp(\tan(\tanh((x5) - \\
& (x3)))))) + (2.7228) * ((((\log(x2)) * (\exp(x5)) * (\tanh(x5)))) * (((x3) + \\
& (x2)) * (\log(x4)))) - (\sin(\exp(\log((x5) - (x3)))))) + (0.0044826) * \\
& (((((x3) + (x2)) * (\log(x4))) * (\exp(x5)) * (x1))) * ((\log(\tan(x1))) + \\
& (\log(\tan(x1)))) + (((x3) + (x2)) * (\log(x4))) - (\log((-9.219389)))))) + \\
& (0.0007562) * ((((\log(x2)) * (\exp(x5)) * (\log(x4)))) * (((\log(x2)) * \\
& (\exp(x5)) * (\tanh(x5)))) + (((x3) + (x3)) * (\log(x5))) - (\log((x3) + \\
& (x2)))))) + (-0.37647) * ((((\tan(x1)) * (\tan(x5))) * (((x4) + (x2)) * \\
& (\log(x4))) - (\log((x3) + (x2)))) - (\sin(((x3) + (\tanh(x3))) * \\
& (\log(x4)))) + (3.3399) * ((((\tan(x2)) * (\tan(x5))) * (\log((x2) + ((x4) + \\
& (x2)))) - (\tanh(x2))) + (-9.1946) * ((((\tan(x1)) * (\tan(x5))) * (((x2) + \\
& (x4)) * (\log(x4))) - (\log((x4) + (x2))))))
\end{aligned} \tag{6.1}$$

$$\begin{aligned}
\text{Compressive strength}_{C-MGGP} = & 15.1525 + (0.031875) * ((\tan(((x2) - ((x2) - \\
& (x4))) * (((x2) - (x5)) * (x3)))) + (((\log((x1) * (x2))) - ((\tan(x2)) - \\
& (\tan((-4.660807)))) - ((\tanh((x2) - (x3))) * (\cos((x1) * (x3)))))) + \\
& (0.0021736) * (((((x2) - ((x2) - (x1))) * (((x2) - (x5)) * (x3))) + (((\cos(x5)) * \\
& ((x2) * (x3))) + (((x1) + (x5)) * ((x1) * (x5)))) * (\tan(\exp(\exp(x1)))) + \\
& (0.015577) * ((\tan(((\cos(x2)) * ((x2) * (x1))) + (\exp(x3)))) + \\
& (((\log((3.837083))) - ((x5) - (\tan(x2)))) - ((\tanh((x5) - (x3))) * \\
& (\cos(\sin((-8.028761)))))) + (2.6258) * (\sin((\log(\log((x2) - (x5)))) * \\
& ((\exp(\exp((-4.660807)))) - (((8.213626)) - (x1)))) + (0.097641) * \\
& ((\tan(((\cos(x5)) * ((x2) * (x3))) + (\exp(x1)))) * (\log(\exp((x1) * (x3)))) + \\
& (154.4125) * ((\tan(((x4) * (x3)) - (((8.738797)) * (x3))) * ((\tan(x5)) * \\
& (\tan(x5)))) * (\cos(((x2) - ((x2) - (x5))) - (\tanh(\cos(x2)))))) + (-0.49136) * \\
& ((\tan(((\cos(x5)) * ((x2) * (x3))) + (\exp(x1)))) + (((\log((x1) * (x3))) - \\
& ((\tan(x2)) - (\tan(x2))))))
\end{aligned} \tag{6.2}$$

6.2.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the two methods MGGP and C-MGGP are illustrated in Figs. 6.3-6.4 on the training, validation and testing data respectively. The best prediction method that gives good generalization ability is determined by comparing these methods using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The results of the training and validation phase shown in Figs. 6.3a-6.4a and Figs. 6.3b-6.4b respectively indicate that both the models have impressively learned the non-linear relationship between the input variables and compressive strength with high correlation values and relatively low error values. The results of the testing phase shown in Figs. 6.3c -6.4c indicate that the proposed C-MGGP model outperformed the standardized MGGP with value of R^2 achieved as high as 0.97.

MO error values for the two models are computed on training, validation and testing data and shown in Table 6.4. It shows that the proposed C-MGGP model outperforms the standardized MGGP. The descriptive statistics of the relative error (%) of the two models are shown in Table 6.5. The lower value of range of the confidence intervals and remaining statistical metrics of the C-MGGP model indicates that its performance is better than that of the MGGP.

Hypothesis tests are also used to compare the goodness of fit of the two models. These tests are conducted for the model formation process and the results are shown in Table 6.6. For the t -test and f -test, the p -values of both the models are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these models differ. Therefore, both the models have statistically satisfactory goodness of fit from the modelling point of view.

Thus from the statistical comparison presented, it can be concluded that the proposed C-MGGP model has outperformed the standardized MGGP.

Fig. 6.3 Comparison of simulated MGGP model and experimental values on (a) training (b) validation data and (c) testing data

Fig.6.4 Comparison of simulated C-MGGP model and experimental values on (a) training (b) validation data and (c) testing data

Table 6.4 Multi-objective error of the two models

Methods	Training data	Validation data	Testing data
MGGP	2.93	2.60	2.61
C-MGGP	2.28	1.82	1.07

Table 6.5 Descriptive statistics for relative error (%) of the two models

Model s	Coun t	Mea n	Std dev	LCI 95 %	UCI 95 %	SE mean	Media n	Maximu m	Minimu m
MGGP	32	2.36	1.74	1.43	2.99	0.3088	1.79	5.85	0.05
C-MGGP	32	1.65	1.52	1.10	2.20	0.26	0.99	0.25	4.83

Table 6.6 Hypothesis testing to compare the two prediction models

95%CI	<i>p</i> -value	
	MGGP	C-MGGP
Mean paired <i>t</i> -tests	0.5274	0.4164
Variance <i>f</i> -test	0.8867	0.9728

6.2.1.3 Sensitivity and Parametric Analysis of the Proposed Model

The results discussed reveal that the C-MGGP outperforms the MGGP in prediction of compressive strength values. To validate the robustness of the C-MGGP method, sensitivity and parametric analysis about its mean is conducted. The sensitivity analysis investigates the contribution of the input variables to the compressive strength values. Fig. 6.5 shows the plots of the sensitivity result of input variables in prediction of compressive strength. It is obvious that the process orientation has the highest impact on the compressive strength followed by layer thickness, raster width, air gap and raster angle. Furthermore, the parametric analysis provides a measure of the relative importance among the chosen inputs of the C-MGGP model, and illustrates how the compressive strength varies in response to the variation in an input. On the C-MGGP model, the value of the first input is varied between its mean $\pm$ standard deviation value of 0.2, and the output value from the C-MGGP model is computed, while the other inputs are fixed at their respective means. This analysis is then repeated for the other inputs. Fig. 6.6 displays the plots generated illustrating the variation of compressive strength with respect to each input variable. These plots reveal that the compressive strength is directly proportional to the layer thickness. The results of this analysis generally comply with the perspective of additive manufacturing theory and are in agreement with those mentioned in the study conducted by Sood et al [116]. The

sensitivity and parametric analysis confirms that the proposed C-MGGP model is capable of capturing the important characteristics of the FDM process.

Fig. 6.5 Relative contribution of each input variable to compressive strength

Fig. 6.6 Variation of compressive strength with respect to each input variable

6.3 Problem: Turning process of AISI 1040 Steel

The motivation behind choosing this problem is same as discussed in Section 4.1.1. Despite being extensively applied, the MGGP method has the problem of difficulty in model selection. In the present chapter, a new methodology referred to as C-MGGP is proposed to tackle the problem of selecting the best model in MGGP. In this approach, four classifiers: Bayes, PART, CART and decision trees are embedded in its evolutionary stage. The proposed approach is applied to the data set obtained from the turning operations and its performance is compared to those of the standardized MGGP, SVR and ANN. The data on the turning process to be modelled is obtained from an earlier study by [278] conducted on the optimization of tool geometry variables for turning operations via response surface methodology. The experiments on turning were carried out in dry conditions on Harrison M300 lathe with maximum spindle speed of 2500 RPM and maximum power of 2.2 kW. The sample material used was AISI 1040 steel in the form of round bars with 40 mm diameter and 250 mm cutting length. The input process variables considered were tool geometry variables (nose radius, approach angle, rake angle) and the output variable was surface roughness. The input process variables with their low-centre-high levels used are shown in Table 6.7.

Twenty-seven sets of data samples [278] are considered in this study in the turning of AISI 1040 steel. These data samples were collected based on L_{27} standard Taguchi orthogonal experimental design. The K-S algorithm is used to divide the data set into training, validation and testing data. Table 6.8 shows the set of data samples after the K-S algorithm has been applied. The first nineteen samples is the

set of training data with the remainder as the set of three validation and the five test samples.

Table 6.7 Input variables of turning process with their low-centre-high values

Process Input Variable	Low (-1)	Centre (0)	High(+1)
Nose radius (mm)	0.4	0.8	1.2
Approach angle (degree)	60	75	90
Rake angle (degree)	-9	-6	-3

Table 6.8 Cutting tool geometry variables and average measured surface roughness values

Trial No.	Nose radius (x_1)	Approach angle (x_2)	Rake angle (x_3)	Surface roughness (y)
1	0.400	60	-3	2.025
2	0.400	75	-3	2.358
3	0.400	75	-6	2.850
4	0.400	90	-3	3.509
5	0.800	60	-6	5.142
6	0.800	75	-6	6.066
7	0.800	90	-6	7.944
8	1.200	90	-3	5.874
9	1.200	90	-9	8.951
10	0.400	60	-6	2.283
11	0.800	75	-9	6.563
12	0.400	60	-9	2.892
13	0.400	75	-9	3.962
14	0.800	60	-3	4.225
15	0.800	75	-3	4.308
16	0.800	90	-9	7.990
17	1.200	60	-3	4.475
18	1.200	75	-9	8.000
19	1.200	90	-6	8.665
20	0.400	90	-9	4.099
21	0.800	60	-9	4.876
22	0.800	90	-3	5.692
23	1.200	60	-6	5.010
24	1.200	60	-9	5.650
25	1.200	75	-3	5.967
26	1.200	75	-6	4.796
27	0.400	90	-6	7.662

6.3.1 Results and Discussion

6.3.1.1 Parameter Settings of CI Methods

The proposed C-MGGP method is implemented on MATLAB R2010b. After applying a trial-and-error approach, the parameter settings chosen for both the methods is shown in Fig. 6.7. The fitness function used is SRM with optimum order of polynomial as complexity measure of the model. The classifiers formed from the attributes such as training error, validation error and number of nodes of the models predicts the class of the unlabelled C-MGGP models. From Table 6.9, we can observe that the model corresponding to run 38 is classified as “best” by all four classification methods. Equations (6.3) and (6.4) represent the C-MGGP and MGGP models obtained respectively. The performance of these models is further evaluated on the testing data.

Fig. 6.7 GUI showing the parameter settings chosen for MGGP and C-MGG

Table 6.9 Class (best or bad) of C-MGGP models predicted by four classification methods

Run	Training error	Validation error	Number of nodes	Bayes	PART	CART	Decision tree
1	1.400	0.242	42	bad	bad	bad	bad
2	0.875	0.110	104	bad	bad	bad	best
3	1.383	0.237	106	bad	bad	bad	bad
4	0.695	0.202	84	bad	bad	bad	bad
5	1.347	0.250	64	bad	bad	bad	bad
6	0.227	0.107	92	bad	best	best	best
7	0.809	0.218	146	best	bad	best	best
8	0.433	0.851	66	bad	bad	best	bad
⋮	⋮	0.112	⋮				
16	0.452	0.207	102	bad	bad	bad	bad
17	0.502	0.225	114	best	bad	bad	bad
18	0.486	0.261	106	best	bad	bad	best
19	1.089	0.378	108	best	bad	bad	bad
⋮	⋮	⋮	⋮				
30	0.914	0.306	62	bad	best	bad	best
	0.372	0.271	54	bad	best	best	best
31							
32	0.838	0.421	114	best	bad	best	best
36	0.701	0.515	162	bad	bad	bad	best
38	0.341	0.120	121	best	best	best	best
⋮	⋮	⋮	⋮				
48	0.529	0.144	54	bad	best	best	best
49	0.752	0.179	38	bad	best	best	best
50	0.480	0.203	116	best	bad	best	best

$$\begin{aligned}
\text{Surface roughness}_{C-MGGP} = & -23.0621 + (4.6268) * ((\cos((\tan(\exp(x3)))) * \\
& ((\exp(x3)) - (\tanh(x3)))) * (\text{plog}(\sin((\exp(x2)) - (\exp(x3))))) + (-0.19039) * \\
& ((\cos((\tan(\exp(x3)))) * ((\exp(x2)) - (\exp(x3)))) * (\text{plog}(\sin(\sin(\exp(x1))))) + \\
& (11.3098) * ((\cos((\tan(\exp(x3)))) * ((\exp(x3)) - (\tanh(x3)))) * \\
& (\exp(\tanh(\text{plog}((x1) * ((9.896637)))))) + (0.24278) * ((\tan(\cos(\tanh((x3) - \\
& (x2)))) * ((\cos(((7.885747)) + (x3)) * (((-3.696601)) * (x2)))) * ((\text{plog}((x1) * \\
& (x3))) * (\tan(\text{plog}(x2)))) + (0.19647) * ((\tanh((\tan(\exp(x3))) * ((\exp(x3)) - \\
& (\tanh(x3)))) + (\sin((\text{plog}((x1) * ((9.923039)))) * (((7.885747)) + (x3)) * \\
& (((9.923039)) * (x2)))) + (-0.59341) * ((\tanh((\tan(\exp(x3))) * (((x1) * \\
& ((9.923039)) - (\tanh(x3)))) + (\sin((\text{plog}((x1) * ((9.923039)))) * (\text{plog}((x3) + \\
& (x1))))) \quad (6.3)
\end{aligned}$$

$$\begin{aligned}
\text{Surface roughness}_{MGGP} = & 5.8964 + (0.079586) * (((\tan(x2)) * (((-6.619773)) + \\
& (x3))) + ((\text{plog}(\cos(\exp(x3)))) * (((\text{plog}(x2)) * ((x1) * (x1))) + (\sin((x2) + \\
& (x3))))) + (-0.58198) * (((\text{plog}(\cos(\exp(x3)))) * (((\text{plog}(x2)) * ((x1) * \\
& (x1))) + (\sin((x1) + (x3))))) * ((\tanh(((x3) + ((0.735306))) - ((x3) * (x1)))) * \\
& ((\sin((x2) + ((-6.976700)))) - (((x2) + (x1)) - ((x3) * (x2))))) +
\end{aligned}$$

$$\begin{aligned}
& (-0.00023921) * ((plog(tan((-6.619773)) + (x3)))) * ((tanh(((x1) - (x3)) - \\
& ((x1) * (x1)))) * (((x2) - (x3)) - ((x3) * (x3))) - ((plog(x3)) - ((x3) * \\
& (x2)))))) + (-5.9186e - 007) * ((plog(tan(plog(plog(x1)))))) * (((sin((x2) + \\
& ((0.735306)))) - (((x2) + (x2)) - ((8.628579) * (x2)))) * ((sin((x2) + \\
& ((0.735306)))) - (((x2) + (x2)) - ((x3) * (x2)))))) + (1.1645) * \\
& (plog((plog(((0.735306) * (x3))) * (sin(((x3) - (x1)) - ((-6.619773)) + \\
& (x3)))))) + (0.027635) * (tan((plog(((x2) + ((0.735306)))) + (x3))) * \\
& (((plog(x3))))
\end{aligned} \tag{6.4}$$

SVR implemented using LS-SVM tool box is applied on the data set shown in Table 6.8. In this chapter, RBF kernel function is chosen. The parameters λ and σ of RBF are estimated using a hybrid approach of CSA and a grid search method. A SVR model with optimal variables $\lambda = 4120.80$ and $\sigma = 0.056$ are found at the second iteration.

For the implementation of ANN, a three-layer feed forward neural network is used with BP learning in MATLAB R2010b. The parameter settings used for ANN is shown in Table 6.10. The optimal network of ANN is determined based on a trial-and-error approach. Fig. 6.8 shows that for the number of neurons 7, the value of RMSE is minimum, and therefore the network of 3-7-1 as shown in Fig. 6.9 for the ANN model is selected.

Table 6.10 Parameter settings for the three-layer ANN

Variables	Values assigned
Number of training data samples	19
Number of validation data samples	3
Number of testing data samples	5
Number of hidden layer	1
Activation function	Sigmoid
Number of epochs	3000
Learning rate	0.001
Architecture selection	Random
Target goal Mean Square Error (MSE)	10^{-5}
Minimum performance gradient	10^{-5}

Fig. 6.8 RMSE obtained by ANN models by varying the number of neurons in the hidden layer

Fig. 6.9 Architecture of ANN determined based on trial-and-error approach

6.3.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the four CI methods (MGGP, C-MGGP, SVR and ANN) are illustrated in Figs. 6.10-6.13 on training, validation and testing data respectively. The best prediction method that gives good generalization ability is determined by comparing these methods using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The results of the training and validation phase shown in Figs. 6.10a-b, Figs. 6.11a-b, Figs. 6.12a-b and Figs. 6.13a-b indicate that all the four models have impressively learned the non-linear relationship between the input variables and surface roughness with high correlation values and relatively low error values. The results of the testing phase shown in Fig. 6.10c, Fig. 6.11c, Fig. 6.12c and Fig. 6.13c indicate that both the proposed C-MGGP and ANN models outperformed the other two models (MGGP and SVR) with value of R^2 achieved as high as 0.96. Between C-MGGP and ANN, both have shown comparable performance. Between MGGP and SVR, SVR has shown better performance.

MO error values for the four models are computed on the training and testing data (Table 6.11). It shows that both the proposed C-MGGP and ANN models outperformed the other two models (MGGP and SVR). The descriptive statistics of the relative error (%) of the four models are shown in Table 6.12. The lower value of range of the confidence intervals and remaining statistical metrics of the ANN and C-MGGP models indicate that their performance is better than those of the other two models (MGGP and SVR). Between C-MGGP and ANN, both have shown comparable performance. Between MGGP and SVR, SVR has shown better performance.

Hypothesis tests are also used to compare the goodness of fit of four models. These tests are conducted for the model formation process and the results are shown in Table 6.13. These are t -tests to test the mean and f -tests for variance. For the t -test and f -test, the p -values of all the four models are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these four models differ. Therefore, all four prediction models have statistically satisfactory goodness of fit from the modelling point of view.

Thus from the statistical comparison presented, it can be concluded that the proposed C-MGGP model has outperformed the two models (MGGP and SVR) and its performance is on par with ANN.

Fig. 6.10 Comparison of simulated MGGP model and experimental values on (a) training (b) validation data and (c) testing data

Fig. 6.11 Comparison of simulated C-MGGP model and experimental values on (a) training (b) validation data and (c) testing data

Fig. 6.12 Comparison of simulated SVR model and experimental values on (a) training (b) validation data and (c) testing data

Fig. 6.13 Comparison of simulated ANN model and experimental values on (a) training (b) validation data and (c) testing data

Table 6.11 Multi-objective error of the four models

Methods	Training data	Validation data	Testing data
MGGP	5.82	6.51	6.34
C-MGGP	4.21	1.87	3.48
SVR	3.46	4.38	4.76
ANN	1.28	3.16	3.16

Table 6.12 Descriptive statistics for relative error (%) of the four models

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
MGGP	27	5.17	6.31	2.67	7.67	1.21	2.46	24.26	0.06
C-MGGP	27	3.42	4.30	1.72	5.12	0.82	1.31	16.17	0.09
SVR	27	3.47	3.99	1.49	5.05	0.76	1.72	14.23	0.12
ANN	27	1.55	3.01	0.36	2.74	0.58	0.11	14.30	0.03

Table 6.13 Hypothesis testing to compare the four prediction models

95%CI	<i>p</i> -value			
	MGGP	C-MGGP	SVR	ANN
Mean paired <i>t</i> -tests	0.2198	0.4484	0.6661	0.7133
Variance <i>f</i> -test	0.8953	0.9397	0.9736	0.8953

6.3.1.3 Sensitivity and Parametric analysis of the Proposed Model

The results discussed reveal that the C-MGGP and ANN outperform the other two methods in prediction of surface roughness values. To validate the robustness of the C-MGGP method, sensitivity and parametric analysis about its mean is conducted. The sensitivity analysis investigates the contribution of the input variables to the surface roughness values. Fig. 6.14 shows the plots of the sensitivity result of input variables in prediction of surface roughness. It is obvious that the process nose radius has the highest impact on the surface roughness followed by rake angle and approach angle. Furthermore, the parametric analysis provides a measure of the relative importance among the chosen inputs of the C-MGGP model, and illustrates how the surface roughness varies in response to the variation in an input. On the C-MGGP model, the value of the first input is varied between its mean $\pm$ standard deviation value of 0.2, and the output value from the C-MGGP model is computed, while the other inputs are fixed at their respective means. This analysis is then repeated for the other inputs. Fig. 6.15 displays the plots generated illustrating the variation of surface roughness with respect to each input variable. These plots reveal that the surface roughness is directly proportional to the nose radius and approach angle. The results of this analysis are in agreement with those mentioned in the study conducted by Neseli et al. [278]. The sensitivity and parametric analysis confirms that the proposed C-MGGP

model is capable of capturing the important characteristics of the turning process of AISI 1040 steel.

Fig. 6.14 Relative contribution of each input variable to surface roughness

Fig. 6.15 Variation of surface roughness with respect to each input variable

6.4 Summary

This present chapter addresses the two important issues of 1) The need and importance of formulating the models for studying the FDM and Turning process 2) Difficulty in model selection in MGGP. To counter these issues, a C-MGGP methodology to select the best model from the population of models is proposed. The proposed methodology is applied on the data obtained from the two processes. The performance of the model selected using the C-MGGP is compared to those of the standardized MGGP, SVR and ANN models. The results discussed conclude that the performance of the C-MGGP model is better than those of the standardized MGGP and SVR, and on par with the ANN model. The conducted sensitivity and parametric analyses validate the robustness of model by unveiling dominant input parameters and hidden non-linear relationships of the FDM and turning process.

CHAPTER 7

HYBRID APPROACH OF MULTI-GENE GENETIC PROGRAMMING FOR IMPROVING TRUSTWORTHINESS OF PREDICTION ABILITY OF MODEL ON UNSEEN SAMPLES

7.1 Hybrid M1-M2 Methodology

As discussed in Chapter 2 (section 2.2.3), the issue of ensuring greater trustworthiness of the prediction ability of the model on testing samples needs attention. To address this, a hybridized M1-M2 approach is proposed for the formulation of a mathematical model, where M1 and M2 are two different CI methods. The hybrid approach comprises of the M1 model and the M2 error model in parallel presented as shown in Fig. 7.1.

Fig.7.1 Hybrid M1-M2 approach

The mathematical representation of the adopted approach is given by Equation (7.1):

$$Y_{Model} = Y_{M1} + E_{M2} \quad (7.1)$$

where, Y_{M1} is the output value obtained from the M1 model tree, E_{M2} is the prediction of the error of the M1 model made by M2 model and Y_{Model} is the actual value predicted by hybridized M1-M2 approach.

The error term E_{M2} is defined as the difference between the actual and predicted values of the M1 model. The sign of error (positive or negative) is taken into account. The error model is formulated using M2, where the inputs into the M2 are the input and output process variables corresponding to the given system as well as the predicted output process variable values from the M1 model, while the output variable is the error of the predictions of the M1 model on the training and/or validation data. In this way, there is one M2 model running in parallel with the M1 model and at every instance, the M1 model predicts, and the M2 model predict the error of the M1 model. By predicting the error of the M1 model, the M2 model ensures greater trustworthiness of the prediction ability of the M1 model on testing (unseen) data. In the following sections, two proposed hybrid approaches MGGP-ANN and M5'-MGGP are applied on the data obtained from the vibratory finishing and FDM processes, respectively.

7.2 Problem: Vibratory finishing process

The motivation behind solving this problem is the same as that discussed in Section 5.3. Advanced CI methods such as MGGP, ANN, FL, etc., can be used for modelling of data obtained from the vibratory finishing process because these methods are void of statistical assumptions and possess the ability to capture the dynamics of the process. Among these methods, MGGP automatically evolve the model structure and its coefficients. In the context of symbolic regression problems, extensive literature has been published that point towards improvement in generalization ability of MGGP, but less focus has been paid to the issue of ensuring greater trustworthiness of the prediction ability of the model on unseen samples.

In this section, the hybrid approach of MGGP-ANN (In reference to Fig. 7.1, MGGP is M1 and ANN is M2) is proposed to model the three performance characteristics of the vibratory finishing process. The hybrid approach is based on the error compensation parallel approach using the ANN model in parallel with the MGGP model. The objective is to compare the performance of the proposed approach MGGP-ANN to that of the standardized MGGP. Twenty-two data points are obtained from an earlier study conducted by Sofronos and Taraman [233] on model development and optimization of vibratory finishing process. Descriptive statistics of the data is shown in Table 7.1. The input variables considered are Brinell hardness (x_1), projection width (x_2), processing time (x_3), media size (x_4) and vibratory frequencies (x_5). The output variables considered are projection height reduction (H), edge radius (E) and surface finish reduction (S). The K-S algorithm is used to select the appropriate training and testing data sets. First

seventeen samples are used as the set of training data with the remainder as the set of test data.

Table 7.1 Descriptive statistics of the input and output process variables

Parameter	x_1	x_2	x_3	x_4	x_5	H	E	S
Mean	120.36	14.72	70.90	0.875	1744.54	25.54	9.54	9.63
Standard error	6.42	0.93	8.5	0.04	46.56	2.87	3.26	0.94
Median	120	14	60	0.875	1730	23.5	4	9
Standard deviation	30.11	4.38	39.86	0.21	218.40	13.47	15.31	4.43
Kurtosis	-1.73	-1.74	-1.75	-1.73	-1.73	0.22	14.17	1.15
Skewness	0.01	0.20	0.32	0	0.08	0.89	3.54	1.04
Minimum	86	10	30	0.625	1500	7	1	3
Maximum	155	20	120	1.125	2000	56	72	20

7.2.1 Results and Discussion

7.2.1.1 Parameter Settings of CI Methods

The two methods of MGGP and MGGP-ANN are implemented on MATLAB R2010b. After applying a trial-and-error approach, the parameter settings chosen for both the methods is shown in Fig. 7.2. The fitness function used is SRM with optimum order of polynomial as the complexity measure of the model. A three-layer feed forward neural network with BP learning is used for the prediction of errors of the MGGP model. The parameter settings used for ANN are shown in Table 7.2. The optimal architecture of ANN is determined based on a trial-and-

error approach. It was found that for the number of neurons 8, the value of RMSE is minimum, and therefore the network of 3-8-1 for the ANN model is selected.

Fig. 7.2 GUI showing the parameter settings chosen for MGGP and MGGP-ANN

TABLE 7.2 Parameter settings for ANN

Variables	Values assigned
Number of training data samples	17
Number of testing data samples	5
Number of hidden layers	1
Number of neurons in hidden layer	8
Activation function	Sigmoid
Number of epochs	1000
Learning rate	0.001
Architecture selection	Random
Target goal MSE	10^{-5}
Minimum performance gradient	10^{-5}
Optimization Algorithm	Levenberg-Marquardt Algorithm

The three MGGP-ANN models for projection height reduction (H), edge radius (E) and surface finishing reduction (S) respectively are as follows:

$$\begin{aligned}
H_{MGGP-ANN} = & -75.6555 + (105.9905) * (\exp(\tanh((\text{plog}(\sin(x1))) + (x4)))) + \\
& (-5.4815) * ((\cos((\sin((x1) - (x1))) - ((\tan(x5)) - (\tan((3.168126)))))) * \\
& ((\tan((-1.665528)) + (x5))) + ((\sin((-1.665528)) + (x5))) - ((\tan(x5)) - \\
& (\tan(x2)))))) + (170.5642) * ((\sin(\exp(x4))) - ((\sin((x1) - (x1))) - \\
& (\tan(\text{plog}(\sin(x4)))))) + (26.6675) * ((\text{plog}((\tan((x1) + (x5))) + (\cos((x2) + \\
& (-1.622109)))))) + (\cos((\tan((-1.665528)) + (x5))) + (\exp(\tan((-3.207200)))))) + \\
& (1.3595) * (((\text{plog}(\sin((-1.665528)) + (x5))) + (x2)) + (\sin((\sin(x3)) - (x4)))) + \\
& (3.46) * (((\exp(\tan((x2) * (x4)))) - ((\sin((x1) - (x5))) * ((\cos((-1.622109)) - \\
& (\sin(x5)))))) * (\sin((\tan((x1) + (-1.665528)))) - (\cos((x5) + (x3)))))) + (-0.07894) * \\
& ((\cos((\sin((-1.665528)) + (x5))) - ((x1) - (x1)) - (\tan((3.168126)))))) * \\
& ((\tan(\sin(\exp(x4)))) + (x5))) + (1.006) * (((\tan(\sin(x5))) * (\cos((x5) + (x5)))) * \\
& ((\tan(\text{plog}(x5))) * ((\exp(x4)) + (\text{plog}((5.275107)))))) * ((\exp(\tan((x2) * (x4)))) - \\
& ((\sin((x1) - (x1))) * ((x4) - ((x5) + (x5)))))); + E_{ANN} \tag{7.2}
\end{aligned}$$

$$\begin{aligned}
\text{Edge radius}_{MGGP-ANN} = & -11.0274 + (-0.026524) * ((\text{plog}((((x4) + (x5)) * (x4)) + \\
& ((x5) - ((x5) - (x1)))))) * (((\tan(x3)) - ((x2) * (x2))) + ((x3) - (\sin(x5)))) * \\
& (\cos(x5))) + (-30.6784) * (\tan(\sin(x4))) + (0.028814) * ((x3) - (\sin((((x3) - (x4)) - \\
& (x5)) + (x2)))) + (-0.23709) * (\tan(\tan((x3) * (((x4) + (-9.169185))) - ((x5) - \\
& (x1)))))) + (0.026357) * (((x5) - ((\cos((x5) - (x1))) - ((x3) - (x4)) - (\cos(x5)))))) * \\
& (x4)) + (0.12538) * (((\tan((x5) - (x1))) - ((x2) * (x2))) + (\sin((x3) - (x5)))) * \\
& (\cos(x5))) + (10.8983) * (\cos(\cos((x3) - (\sin((x3) + (x2)))))) + (-3.6334) * \\
& (((\tanh(\exp((x1) + (x4)))) + \cos(\cos((x5) - (x1))))); + E_{ANN} \tag{7.3}
\end{aligned}$$

$$\begin{aligned}
S_{MGGP-ANN} = & -6.5597 + (7.2994) * ((\tanh((\tan(\sin(x4))) + ((\sin(x2)) + (\cos(x2)))))) + \\
& ((\sin(\exp(\sin(x4)))) + (\cos(\tanh(\exp(x3)))))) + (-0.044247) * ((x3) - ((\tanh(((x3) - \\
& (x2)) * ((x3) - (x2)))) - ((\exp((x4) * (x4))) * (\exp((x4) * (x4)))))) + (-1.1019) * \\
& ((\sin(\exp(\tan((x5) - (x2)))))) + ((\sin(\sin(x4))) + (\tanh(((3.884574)) + ((x3) +
\end{aligned}$$

$$\begin{aligned}
& (x2)))))) + (0.085035) * ((\exp(((((-7.973680)) * (x5)) * (x5)) + (((x3) - (x1)) * \\
& ((x3) - (x2)))))) - (x1)) + (-23.1029) * ((\tanh(\text{plog}(x4))) * \\
& (\tanh(\sin(\exp(\sin(x4)))))) + (7.0602) * (\cos(((\tan(\tanh(x1))) + ((\tanh(x4)) - (x2))) + \\
& ((\cos(\tanh(x4))) - (\tan((x5) - ((7.842407))))))) + (4.8911) * (\sin((\tanh((x5) - \\
& ((7.842407))) * ((x3) - (x1)))) - ((\exp((x4) * (x4))) * (\tanh(\exp(x4)))))) + (8.4562) * \\
& ((\sin(\sin(((x3) - (x1)) * ((x3) - (x2)))))) * (\tanh(x4)))
\end{aligned} \tag{7.4}$$

7.2.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the two methods (MGGP and MGGP-ANN) are illustrated in Figs. 7.3-7.8 on the training and testing data for the three outputs respectively. The best prediction method that gives good generalization ability is determined by comparing these methods using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The results of the training phase shown in Figs. 7.3a-7.8a indicate that both the models have impressively learned the non-linear relationship between the input variables and the three outputs with high correlation values and relatively low error values. The results of the testing phase shown in Figs. 7.3b-7.8b indicate that the proposed MGGP-ANN model outperformed the standardized MGGP with value of R^2 achieved as high as 0.98 for the output E .

MO error values for the two models are computed on the training and testing data and shown in Table 7.3. It shows that the proposed MGGP-ANN model outperforms the standardized MGGP on all the three outputs. The descriptive statistics of the relative error (%) of the two models are shown in Tables 7.4-7.6 for the three outputs respectively. The lower value of range of the confidence

intervals and remaining statistical metrics of the MGGP-ANN models indicate that its performance is better than that of the MGGP for all the three outputs.

Hypothesis tests are also used to compare the goodness of fit of the two models. These tests are conducted for the model formation process and the results are shown in Table 7.8 for all the three outputs. For the t -test and f -test, the p -values of both the models are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these models differ. Therefore, both the models have statistically satisfactory goodness of fit from the modelling point of view.

Thus from the statistical comparison presented, it can be concluded that the MGGP-ANN model has outperformed the standardized MGGP.

Fig. 7.3 Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data for the output H

Fig.7.4 Comparison of simulated MGGP-ANN model and experimental values on (a) training and (b) testing data for the output H

Fig.7.5 Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data for the output E

Fig.7.6 Comparison of simulated MGGP-ANN model and experimental values on (a) training and (b) testing data for the output E

Fig.7.7 Comparison of simulated MGGP model and experimental values on (a) training and (b) testing data for the output S

Fig.7.8 Comparison of simulated MGGP-ANN model and experimental values on (a) training and (b) testing data for the output *S*

Table 7.3 Multi-objective error of the two models on the three outputs

Outputs	H		E		S	
Methods	Training data	Testing data	Training data	Testing data	Training data	Testing data
MGGP	24.89	22.82	58.43	38.92	35.14	25.90
MGGP-ANN	6.14	11.14	17.45	4.94	16.41	10.70

Table 7.4 Descriptive statistics for relative error (%) of the two models for output *H*

Model s	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
MGGP	22	17.38	14.73	10.85	23.91	3.14	15.22	56.16	0.48
MGGP-ANN	22	5.22	7.00	2.11	8.32	1.49	2.02	25.14	0.05

Table 7.5 Descriptive statistics for relative error (%) of the two models for output *E*

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
MGGP	22	50.41	53.61	26.64	74.48	11.43	40.60	235.77	1.25
MGGP-ANN	22	13.72	31.54	-0.26	27.70	6.72	5.58	151.57	0.06

Table 7.6 Descriptive statistics for relative error (%) of the two models for output S

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
MGGP	22	24.22	22.24	14.36	34.09	4.74	17.50	114	5.37
MGGP-ANN	22	12.74	17.44	5.006	20.47	3.72	5.98	63.69	1.61

Table 7.7 Hypothesis testing to compare the two prediction models on the three outputs

95%CI	H		E		S	
	MGGP	MGGP-ANN	MGGP	MGGP-ANN	MGGP	MGGP-ANN
Mean paired t -tests	0.0511	0.6956	0.1331	0.0723	0.3269	0.5883
Variance f -test	0.9724	0.9743	0.9464	0.9955	0.2705	0.9728

7.2.1.3 Sensitivity and Parametric Analysis of the Proposed Model

The results discussed reveal that the MGGP-ANN outperforms the MGGP in the prediction of the three performance characteristics of the vibratory finishing process. To validate the robustness of the MGGP-ANN method, the sensitivity and parametric analysis about its mean is conducted. The sensitivity analysis investigates the contribution of the input variables to the three output values. Fig. 7.9 shows the plots of the sensitivity result of input variables in prediction of the three outputs. It is obvious that the process brinell hardness has the highest impact on the output H followed by media size, vibratory frequencies, projection width and processing time. For output E , processing time has the highest impact followed by vibratory frequencies, projection width, media size and Brinell hardness. For output S , processing time has the highest impact followed by media size, projection width, Brinell hardness and vibratory frequencies. This implies that by regulating the most significant parameter, the greater amount of variation in the three outputs can be achieved. Furthermore, the parametric analysis provides a

measure of the relative importance among the chosen inputs of the MGGP-ANN model, and illustrates how the three outputs vary in response to the variation in an input. On the MGGP-ANN model, the value of the first input is varied between its mean $\pm$ standard deviation value of 0.2, and the output value from the MGGP-ANN model is computed, while the other inputs are fixed at their respective means. This analysis is then repeated for the other inputs. Figs. 7.10-7.12 displays the plots generated showing the variation of the three outputs, H , E and S , with respect to the each input variable. From Fig. 7.10, it is clear that the output H has the non-linear relation with respect to all inputs except the processing time. From Fig. 7.11, it is clear that the input variables (processing time, media size and frequencies) have a positive impact on edge radius whereas the projection width has a negative impact. From Fig. 7.12, it is clear that the input variables (Brinell hardness, processing time and media size) have a negative impact on surface finish reduction. The results of this analysis generally comply with the perspective of surface finish theory and are in agreement with those mentioned in the study by Sofronas and Taraman [233]. The sensitivity and parametric analysis confirms that the proposed MGGP-ANN model is capable of capturing the important characteristics of the vibratory finishing process.

Fig 7.9 Relative contribution of each input variable to projection height reduction, edge radius and surface finish reduction respectively

Fig. 7.10 Variation of output H with respect to each input variable

Fig. 7.11 Variation of output E with respect to each input variable

Fig. 7.12 Variation of output S with respect to each input variable

7.3 Problem: Fused deposition modeling process (FDM)

The motivation behind solving this problem is the same as that discussed in Section 6.2. For modeling the response characteristics of the FDM process, several CI methods such as MGGP, regression analysis, ANN, M5' and ANFIS can be used. The M5 algorithm used to build regression trees has prediction accuracy on par with ANN but was later modified as M5' [28, 279-282]. The ANFIS model combines BP-NN and fuzzy inference system to generate the rules-based models. Several applications of ANFIS in the field of manufacturing have been reported [163, 283, 284].

Despite advanced characteristics of CI methods, the issues concerning the trustworthiness of the model on the unseen data or test data still remain. Although the hybrid approach MGGP-ANN produces generalized models, there exists a limitation in implementing the approach. The performance of the best model produced for every run of MGGP differs a lot, so the ANN model used to compensate the MGGP can only be applied on a best run of MGGP. Apart from the best run of MGGP, it is highly likely that the ANN model when compensated may not improve the prediction accuracy. Therefore, the MGGP which was a M1 method in Section 7.2 is now made as M2 method. A hybridized approach of M5'-Multi-gene genetic programming (M5'-MGGP) is proposed for the formulation of the functional expression of compressive strength of the FDM fabricated part. With reference to Fig. 7.1, M5' is M1 method and MGGP is M2 method. The objective is to compare the performance of the M5'-MGGP to those of SVM and ANFIS and to resolve the issue of ensuring greater trustworthiness of prediction ability of the model on the test samples. The data on the FDM process is obtained

from the study conducted by Sood et al [116] on the experimental investigation and modelling of the compressive strength of FDM built parts. The process input and output variables and their values used are the same as those discussed in Section 6.2. The K-S algorithm is used to select the appropriate training and testing data set. The first twenty-five samples is the set of training data with the remainder seven as the set of test data.

7.3.1 Results and Discussion

7.3.1.1 Parameter Settings of CI Methods

The implementation of the M5'-MGGP method is carried out in two phases. In the first phase, the M5' algorithm is executed using the M5' software built in MATLAB platform [285]. In the second phase, MGGP is implemented in MATLAB R2010b. After applying a trial-and-error approach, the parameter settings chosen for M5'-MGGP is shown in Table 7.8 and Fig. 7.13. The fitness function used is SRM with optimum order of polynomial as the complexity measure of the model.

Table 7.8 Parameter settings for M5'

Variables	Values assigned
Minimum number of cases	3
Smoothing	Yes
Pruning	Yes
Split threshold	0.05

Fig. 7.13 GUI showing the parameter settings chosen for MGGP and M5'-MGGP

The error predicted by the MGGP model is used to compensate the M5' model for the formulation of the M5'-MGGP model given by:

$$\begin{aligned}
 &\text{Compressive strength}_{M5'-MGGP} = \\
 &\quad (if \ x_1 > 0.254 \\
 &\quad y = 0 \\
 &\quad else \ y = 20.35 \ x_1 - 0.1202 \ x_2 - 206.31 \ x_5 + 11.98) \\
 &\quad + E_{MGGP}
 \end{aligned} \tag{7.5}$$

ANFIS tool box [286] built in MATLAB is used for the implementation of ANFIS. The ANFIS implements Sugeno-style first order fuzzy system. Since the number of input variables is more than four, this can generate an excessive number of rules, if the FIS structure with fixed number of membership functions is used. Therefore, in order to avoid the curse of dimensionality, the subtractive clustering method proposed by Chiu [287] is used before the generation of FIS. This method

partitions the data into groups called clusters, and generates a FIS with the minimum number of rules required to distinguish the fuzzy qualities associated with each of the clusters. The Gaussian membership function is used for each input and a total of 20 rules are generated as shown in Fig. 7.14. A hybrid method of least squares and BP-NN is used to train the FIS. For the stopping criterion, the error tolerance is kept at zero and the number of epochs is set at 50.

Fig. 7.14 FIS rules for FDM modelling using ANFIS

SVR is implemented using the LS-SVM tool box. In this chapter, RBF kernel function is chosen. The parameters λ and σ of RBF are determined using hybrid approach of CSA and a standard simplex. Optimal parameters $\lambda = 13.72$ and $\sigma = 129.82$ are obtained at the 12th iteration.

7.3.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the three CI methods (M5'-MGGP, ANFIS and SVR) are illustrated in Figs. 7.15-7.17 on training and testing data respectively. The best prediction method that gives good generalization ability is determined by comparing these methods using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The results of the training phase shown in Figs. 7.15a-7.17a indicate that all the three methods have impressively learned the non-linear relationship between the input variables and compressive strength with high correlation values and relatively low error values. The results of the testing phase shown in Figs. 7.15b-7.17b indicate that both the proposed M5'-MGGP and SVR models outperformed the ANFIS model with value of R^2 achieved as high as 0.94. Between M5'-MGGP and SVR, both have shown comparable performance.

MO error values for the three models are computed on the training and testing data and shown in Table 7.9. It shows that both the proposed M5'-MGGP and SVR models outperformed the ANFIS model. The descriptive statistics of the relative error (%) of the three models are shown in Table 7.10. The lower value of range of the confidence intervals and remaining statistical metrics of the M5'-MGGP and SVR models indicates that their performance is better than that of the ANFIS. Between M5'-MGGP and SVR, both have shown comparable performance.

Hypothesis tests are also used to compare the goodness of fit of the three models. These tests are conducted for the model formation process and the results are shown in Table 7.11. These are t -tests to test the mean and f -tests for variance. For the t -test and f -test, the p -values of all the three models are greater than 0.05, so

there is not enough evidence to conclude that the actual values and predicted values from these three models differ. Therefore, all three prediction models have statistically satisfactory goodness of fit from the modelling point of view.

Thus, from the statistical comparison presented, it can be concluded that the proposed M5'-MGGP model has outperformed ANFIS model. Between M5'-MGGP and SVR, both have shown comparable performance.

Fig. 7.15 Comparison of simulated M5'-MGGP model and experimental values on (a) training and (b) testing data

Fig. 7.16 Comparison of simulated ANFIS model and experimental values on (a) training and (b) testing data

Fig. 7.17 Comparison of simulated SVR model and experimental values on (a) training and (b) testing data

Table 7.9 Multi-objective error of the three models

Methods	Training data	Testing data
M5'-MGGP	3.16	3.71
ANFIS	3.44	7.98
SVR	5.11	3.90

Table 7.10 Descriptive statistics for relative error (%) of the three models

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
M5'-MGGP	32	2.71	2.45	1.82	3.59	0.43	2.22	9.61	0.04
ANFIS	32	3.36	4.73	1.65	5.06	0.83	1.47	18.72	0.03
SVR	32	3.85	3.92	2.44	5.26	0.69	2.55	16.91	0.06

Table 7.11 Hypothesis testing to compare the three prediction models

95%CI	p-value		
	M5'-MGGP	ANFIS	SVR
Mean paired <i>t</i> -tests	0.2177	0.0513	0.0931
Variance <i>f</i> -test	0.9928	0.7624	0.4262

7.3.1.3 Sensitivity and Parametric Analysis of the Proposed Model

The results discussed reveal that the proposed M5'-MGGP outperforms in the prediction of the compressive strength of FDM fabricated part. To validate the robustness of the M5'-MGGP method, sensitivity and parametric analysis about its mean is conducted. Fig. 7.18 shows the plots of the sensitivity result of input variables in prediction of compressive strength. It is obvious that the process orientation has the highest impact on the compressive strength followed by layer thickness, raster width, air gap and raster angle. This implies that by regulating the orientation of the build part, the greater amount of variation in the compressive strength can be achieved. Furthermore, the parametric analysis provides a measure of the relative importance among the chosen inputs of the M5'-MGGP model, and illustrates how the compressive strength varies in response to the variation in an input. On the M5'-MGGP model, the value of the first input is varied between its mean $\pm$ standard deviation value of 0.2, and the output value from the model is computed, while the other inputs are fixed at their respective means. This analysis is then repeated for the other inputs. Fig. 7.19 displays the plots generated showing the variation of the compressive strength with respect to each input variable. These plots reveal that the input variables (raster angle and raster width) have a positive impact on compressive strength whereas the remaining has a negative impact. The results of this analysis are in good agreement with those of the study conducted by Sood et al. [116]. The sensitivity and parametric analysis confirms that the proposed M5'-MGGP model is capable of capturing the important characteristics of the FDM process.

Fig 7.18 Relative contribution of each input variable to compressive strength

Fig. 7.19 Variation of compressive strength with respect to each input variable

7.4 Summary

The major issues related to the empirical modelling of the two manufacturing processes of vibratory finishing and FDM are discussed. In addition, the need for ensuring the greater trustworthiness in the prediction ability of the model is also addressed. To tackle it, two hybrid methods of MGGP (MGG—ANN and M5'-MGGP) are proposed and applied to model the performance characteristics of the two processes. The results show that for both processes, the hybrid models have performed better than those of the other models. In other words, a hybrid network of two potential methods is formed, wherein one of them acts as an error model to predict the error of the other model and thus provides greater trustworthiness of the prediction ability for the unseen data samples. The conducted sensitivity and parametric analyses validate the robustness of model by unveiling dominant input parameters and hidden non-linear relationships of the vibratory finishing and FDM process.

CHAPTER 8

HYBRID APPROACH OF ADVANCED MULTI-GENE GENETIC PROGRAMMING WITH ANOTHER COMPUTATIONAL INTELLIGENCE METHOD

8.1 *Methodology*

Based on the study of methods in Chapter 4-7, the CI approach of M5'-AMGGP is formulated, which consists of the integration of two methods (M5' and AMGGP) as shown in Fig. 8.1. Details about the complete steps for the formulation of the approach are discussed in Chapter 3. In the proposed approach, the error term is defined as the difference between the actual and predicted values of the M5' model. The sign of the error (positive or negative) is taken into account. The error model is formulated using AMGGP, where the inputs fed into the AMGGP are the input and output process variables corresponding to the given system as well as the predicted values of the M5' model, while the output parameter is the error of the predictions of the M5' model on training data. In this way, there is one AMGGP model running in parallel with the M5' model and at every instance, the M5' model predicts the output, and the AMGGP model predict the error of the M5' model. By predicting the error of the M5' model, the AMGGP model ensures greater

trustworthiness of the prediction ability of the M5' model on testing (unseen) data.

Predictions obtained from the proposed CI approach are given by

$$Y_{Model} = Y_{M5'} + E_{AMGGP} \quad (8.1)$$

Fig. 8.1 Formulation of CI approach M5'-AMGGP

where, $Y_{M5'}$ is the output value obtained from the M5' model tree, E_{AMGGP} is the prediction of the error of the M5' model made by AMGGP model and Y_{Model} is the actual value predicted by hybridized M5'-AMGGP approach.

In the following sections, the proposed approach M5'-AMGGP is applied on the data obtained from two turning processes, one vibratory finishing process and one FDM process. The performance is compared to those of other standardized methods: MGGP, SVR, ANFIS and ANN.

8.2 Problem: Turning Process of AISI H11 Steel

The motivation behind solving this problem and the details about the process variables considered have been discussed in Section 4.2. In this section, the hybridized M5'-AMGGP approach is applied to the modelling of surface roughness of turning process. The objective is to compare the performance of the M5'-AMGGP approach to those of the standardized MGGP, M-MGGP, SVR, ANFIS and ANN. The first twenty-one samples is the set of training data with the remaining as the set of six test data.

8.2.1 Results and Discussion

8.2.1.1 Parameter Settings of CI Methods

The implementation of the M5'-AMGGP method is carried out in two phases. In the first phase, M5' algorithm is executed. In the second phase, AMGGP is implemented in MATLAB R2010b. The parameter settings for M5'-AMGGP are set based on a trial-and-error approach (Table 8.1 and Fig. 8.2). The fitness function used is SRM with optimum order of polynomial as the complexity measure of the model.

Table 8.1 Parameter settings for M5'

Variables	Values assigned
Minimum number of cases	5
Smoothing	No
Pruning	Yes
Split threshold	0.05

Fig. 8.2 GUI showing the parameter settings chosen for M5'-AMGGP

The error predicted using the AMGGP model is used to compensate the M5' model for the formulation of the M5'-AMGGP model given by:

Surface roughness $M5'-AMGGP =$

if $x_2 \leq 0.1$

$$y = 0.2737$$

else

if $x_1 \leq 210$

$$y = 0.7105$$

else

$$y = 1.341 + E_{GP} \quad (8.2)$$

The parameter settings for the methods (M-MGGP, MGGP, SVR and ANN) are the same as those discussed in Section 4.2.1.1. ANFIS is implemented using ANFIS tool box built in MATLAB R2010b. Since, the number of input variables

is less than four, the subtractive clustering is not used as a pre-processing tool for obtaining clusters. The Gaussian membership function is used for each input and the set of rules are generated. A hybrid method of least squares and BP-NN is used to train the FIS. For the stopping criterion, the error tolerance is kept at zero and the number of epochs is set at 50.

8.2.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the six CI methods (MGGP, M5'-AMGGP, M-MGGP, SVR, ANFIS and ANN) are illustrated in Tables 8.2 and 8.3 on training and testing data. The best prediction method that gives good generalization ability is determined by comparing these methods using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The results of the training phase shown in Table 8.2 indicate that all the six models have impressively learned the non-linear relationship between the input variables and surface roughness with high correlation values and relatively low error values. The result of the testing phase (Table 8.2) indicates that the models (M5'-AMGGP and M-MGGP) outperformed the other four with value of R^2 achieved as high as 0.97. Between M5'-AMGGP and M-MGGP, both have shown comparable performance. Among MGGP, SVR, ANFIS and ANN, ANN has shown better performance followed by MGGP, SVR and ANFIS.

MO error values for the six models are computed on the training and testing data and shown in Table 8.3. It shows that the two models (M5'-AMGGP and M-MGGP) outperform the other four models. The descriptive statistics of the relative error (%) of the six models are shown in Table 8.4. The lower value of range of confidence intervals and remaining statistical metrics of the M5'-AMGGP and M-

MGGP models indicate that their performance is better than those of the other four models. Between M5'-AMGGP and M-MGGP, both have shown comparable performance. Among MGGP, SVR, ANFIS and ANN, ANN has shown better performance followed by MGGP, SVR and ANFIS.

Hypothesis tests are also used to compare the goodness of fit of the six models. These tests are conducted for the model formation process and the results are shown in Table 8.5. For the *t*-test and *f*-test, the *p*-values of all the models are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these models differ. Therefore, all six prediction models have statistically satisfactory goodness of fit from the modelling point of view.

Thus from the statistical comparison presented, it can be concluded that the proposed M5'-AMGGP and M-MGGP models have outperformed the other four models. Between M5'-AMGGP and M-MGGP, both have shown comparable performance.

Table 8.2 R^2 , MAPE (%) and RMSE of the six models

Error measure metrics	R^2		MAPE (%)		RMSE	
	Training data	Testing data	Training data	Testing data	Training data	Testing data
M5'- AMGGP	0.98	0.97	2.30	6.06	0.017	0.12
MGGP	0.91	0.92	8.74	15.20	0.07	0.25
M-MGGP	0.96	0.95	3.15	6.9	0.031	0.11
SVR	0.94	0.90	6.88	15.7	0.06	0.21
ANFIS	0.92	0.87	12.40	22.10	0.07	0.33
ANN	0.92	0.94	8.36	13.10	0.07	0.18

Table 8.3 Multi-objective error of the six models

Methods	Training data	Testing data
M5'-AMGGP	2.36	6.37
MGGP	9.68	16.79
M-MGGP	3.31	7.37
SVR	7.38	17.67
ANFIS	13.55	25.78
ANN	9.16	14.12

Table 8.4 Descriptive statistics for relative error (%) of the six models

Models	Co unt	Me an	Std dev	LCI 95%	UCI 95%	SE mean	Medi an	Maxi mum	Minimu m
M5'-AMGGP	27	3.12	2.99	1.94	4.31	0.57	2.31	11.60	0.12
MGGP	27	10.18	15.25	4.14	16.22	2.93	0.44	59.72	0
M-MGGP	27	3.49	4.49	1.72	5.27	0.86	1.65	17.85	0.05
SVR	27	8.84	7.22	5.98	11.70	1.39	8.14	24.94	0.23
ANFIS	27	14.56	15.99	8.23	20.89	3.07	8.43	59.72	0.02
ANN	27	9.41	10.73	5.17	13.66	2.06	6.76	33.55	0.00

Table 8.5 Hypothesis testing to compare the six prediction models

95%CI	<i>p</i> -value					
	M5'-AMGGP	MGGP	M-MGGP	SVR	ANFIS	ANN
Mean paired <i>t</i> -tests	0.9861	0.1151	0.1129	0.2208	0.1162	0.2140
Variance <i>f</i> -test	0.7308	0.2421	0.3926	0.0637	0.2241	0.0921

8.3 Problem: Turning of ASS 304 Steel Subjected to DSA Regime

The motivation behind solving this problem and the details about the process variables considered have been discussed in Section 4.3. In this section, the hybridized M5'-AMGGP is applied to the modelling of true stress of ASS 304 steel subjected to DSA regime. The performance of the M5'-AMGGP approach is compared to those of the standardized MGGP, M-MGGP, SVR, ANFIS and ANN. 225 sets of data samples are generated from the experiment. The K-S algorithm is

applied to select 80% (180 samples) of the total samples as training data with the remaining as the set of forty five test data.

8.3.1 Results and Discussion

8.3.1.1 Parameter Settings of CI Methods

The implementation of the M5'-AMGGP method is carried out in two phases. In the first phase, M5' algorithm is executed. In the second phase, AMGGP is implemented in MATLAB R2010b. The parameter settings for M5'-AMGGP are set based on a trial-and-error approach (Table 8.6 and Fig. 8.3). The fitness function used is SRM with optimum order of polynomial as the complexity measure of the model.

Table 8.6 Parameter settings for M5'

Variables	Values assigned
Minimum number of cases	6
Smoothing	No
Pruning	Yes
Split threshold	0.05

Fig. 8.3 GUI showing the parameter settings chosen for M5'-AMGGP

The error predicted using the AMGGP model is used to compensate the M5' model for the formulation of the M5'-AMGGP model given by:

True stress $M5'-AMGGP =$

if $x_2 \leq 798$

if $x_3 \leq 0.13$

$y = 0.2136 + 1.925 * x_3$

else

if $x_3 \leq 0.21$

if $x_3 \leq 0.17$

if $x_1 \leq 0.00055$

$y = 0.5036$

else

```

    if  $x_3 \leq 0.15$ 

         $y = 0.4562$ 

    else

         $y = 0.4835$ 

    else

        if  $x_1 \leq 0.00055$ 

             $y = 0.5638$ 

        else

            if  $x_3 \leq 0.19$ 

                 $y = 0.5106$ 

            else

                 $y = 0.5304$ 

        else

            if  $x_1 \leq 0.00055$ 

                 $y = 0.3566 + 1.117 * x_3$ 

            else

                if  $x_3 \leq 0.25$ 

                    if  $x_3 \leq 0.23$ 

                         $y = 0.553$ 

                    else

                         $y = 0.5752$ 

                else

                     $y = 0.5965$ 

            else

                if  $x_1 \leq 0.00055$ 

```

```

if  $x_3 \leq 0.07$ 
 $y = 0.197$  (9)
else
if  $x_2 \leq 848$ 
 $y = 0.2278 + 0.3027 * x_3$ 
else
 $y = -0.4649 + 0.0007524 * x_2 + 0.1616 * x_3$ 
else
if  $x_3 \leq 0.13$ 
if  $x_3 \leq 0.07$ 
if  $x_2 \leq 898$ 
if  $x_1 \leq 0.0055$ 
 $y = 0.1963$ 
else
 $y = 0.2563$ 
else
 $y = 0.2902$ 
else
if  $x_2 \leq 898$ 
if  $x_1 \leq 0.0055$ 
 $y = 0.2492$ 
else
 $y = 0.3354$ 
else
 $y = 0.3581$ 

```

$$\begin{aligned}
& \text{else} \\
& \text{if } x_1 \leq 0.0055 \\
& \text{if } x_2 \leq 898 \\
& \text{if } x_2 \leq 848 \\
& y = 0.2488 + 0.4974 * x_3 \\
& \text{else} \\
& y = 0.2003 + 0.2537 * x_3 \\
& \text{else} \\
& y = 0.323 + 0.5593 * x_3 \\
& \text{else} \\
& \text{if } x_2 \leq 848 \\
& y = 0.328 + 0.6709 * x_3 \\
& \text{else} \\
& y = -0.299 + 0.0006486 * x_2 + 0.5909 * x_3 + E_{GP} \quad (8.3)
\end{aligned}$$

The parameter settings for the methods (M-MGGP, MGGP, SVR and ANN) are the same as those discussed in Section 4.3.1.1. ANFIS is implemented using ANFIS tool box built in MATLAB R2010b. The Gaussian membership function is used for each input and the set of rules are generated. A hybrid method of least squares and BP-NN is used to train the FIS. For the stopping criterion, the error tolerance is kept at zero and the number of epochs is set at 50.

8.3.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the six CI methods (MGGP, M5'-AMGGP, M-MGGP, SVR, ANFIS and ANN) are illustrated in Tables 8.7 and 8.8 on training and testing data. The best prediction method that gives good generalization ability is

determined by comparing these methods using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The results of the training phase (Table 8.7) indicate that all the six models have impressively learned the non-linear relationship between the input variables and true stress with high correlation values and relatively low error values. The results of the testing phase (Table 8.7) indicate that the proposed M5'-AMGGP model outperformed the other five models with value of R^2 achieved as high as 0.96. Among M-MGGP, MGGP, SVR, ANFIS and ANN, M-MGGP is the first best and MGGP is the second best. Among ANFIS, SVR and ANN, all three have shown comparable performance.

MO error values for the six models are computed on the training and testing data and shown in Table 8.8. It shows that the M5'-AMGGP model outperforms the other five models. The descriptive statistics of the relative error (%) of the six models are shown in Table 8.9. The lower value of range of confidence intervals and remaining statistical metrics of the M5'-AMGGP model indicate that its performance is better than those of the other five models. Among MGGP, M-MGGP, SVR, ANFIS and ANN, M-MGGP is the first best and MGGP is second best. Among ANFIS, SVR and ANN, all three have shown comparable performance.

Hypothesis tests are also used to compare the goodness of fit of the six models. For the t -test, the p -values of the models (MGGP, M-MGGP and M5'-AMGGP) are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these models differ, whereas, for the remaining three models (ANFIS, ANN and SVR), the p -values are less than 0.05, which

indicates that there is enough evidence to conclude that the predicted and actual values differ. For the f -test, the p -values of the models (M-MGGP and M5'-AMGGP) are greater than 0.05 so there is not enough evidence to conclude that the actual values and predicted values from these models differ, whereas for the remaining four models, the p -values are less than 0.05, which indicates that there is enough evidence to conclude that the predicted and actual values differ.

Thus from the statistical comparison presented, it can be concluded that the proposed M5'-AMGGP model has outperformed the other five models. Among M-MGGP, MGGP, SVR, ANFIS and ANN, M-MGGP is first best and MGGP is second best. Among ANFIS, SVR and ANN, all three have shown comparable performance

Table 8.7 R^2 , MAPE (%) and RMSE of the six models

Error measure metrics	R^2		MAPE (%)		RMSE	
	Training data	Testing data	Training data	Testing data	Training data	Testing data
M5'-AMGGP	0.97	0.96	4.35	5.77	0.019	0.03
MGGP	0.95	0.85	5.05	6.97	0.022	0.034
M-MGGP	0.95	0.93	5.76	4.98	0.024	0.023
SVR	0.94	0.81	6.68	7.80	0.029	0.039
ANFIS	0.95	0.84	5.17	7.88	0.027	0.038
ANN	0.96	0.82	5.31	8.16	0.024	0.034

Table 8.8 Multi-objective error of the six models

Methods	Training data	Testing data
M5'-AMGGP	4.40	4.98
MGGP	5.33	7.21
M-MGGP	6.08	5.37
SVR	7.45	9.67
ANFIS	5.47	9.42
ANN	5.55	9.99

Table 8.9 Descriptive statistics for relative error (%) of the six models

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
M5'-AMGGP	225	4.33	4.18	3.78	4.88	0.27	2.78	19.35	0.006
MGGP	225	7.95	6.76	4.65	7.24	0.45	3.44	22.79	0.03
M-MGGP	225	5.34	5.52	5.22	6.67	0.36	3.44	28.68	0.02
SVR	225	6.23	0.41	6.04	8.67	0.41	5.15	29.11	0.02
ANFIS	225	5.59	5.23	4.90	6.28	0.34	4.48	36.61	0.02
ANN	225	5.75	5.19	5.07	6.84	0.34	4.33	28.69	0.05

Table 8.10 Hypothesis testing to compare the six prediction models

95%CI	<i>p</i> -value					
	M5'-AMGGP	MGGP	M-MGGP	SVR	ANFIS	ANN
Mean paired <i>t</i> -tests	0.9098	0.1457	0.1229	0.0114	0.0364	0.0235
Variance <i>f</i> -test	0.8780	0.0322	0.2954	0.0293	0.0493	0.0468

8.4 Problem: Fused deposition modeling process (FDM)

The motivation behind solving this problem and the details about the process variables considered have been discussed in Section 7.3. In this section, the hybridized approach of M5'-AMGGP is applied to the modelling of compressive strength of the FDM process. The performance of the M5'-AMGGP method is compared to those of the M5'-MGGP, SVR, ANFIS and ANN. The K-S algorithm is applied to select twenty-five samples as training data with the remaining seven as a set of test data.

8.4.1 Results and Discussion

8.4.1.1 Parameter Settings of CI Methods

The implementation of the M5'-AMGGP method is carried out in two phases. In the first phase, the M5' algorithm is executed. In the second phase, AMGGP is implemented in MATLAB R2010b. The parameter settings for M5'-AMGGP are set based on a trial-and-error approach and shown in Table 8.11 and Fig. 8.4. The

fitness function used is SRM with optimum order of polynomial as the complexity measure of the model.

Table 8.11 Parameter settings for M5'

Variables	Values assigned
Minimum number of cases	3
Smoothing	Yes
Pruning	Yes
Split threshold	0.05

Fig. 8.4 GUI showing the parameter settings chosen for M5'-AMGGP

The error predicted using the AMGGP model is used to compensate the M5' model for the formulation of the M5'-AMGGP model given by:

Compressive strength_{M5'-AMGGP} =

(if $x_1 > 0.254$

$y = 0$

else

$$y = 20.35 x_1 - 0.1202 x_2 - 206.31 x_5 + 11.98) + E_{GP} \quad (8.4)$$

The parameter settings for the methods (M5'-MGGP, SVR and ANFIS) are the same as those discussed in Section 7.3.1.1. ANN is implemented in a tool box built in MATLAB R2010b. The optimal architecture of ANN is determined based on a trial-and-error approach. For the number of neurons 7, the value of RMSE is minimum, and therefore the network of 3-7-1 for the ANN model is selected. The remaining parameter settings are same as those mentioned in Section 7.3.1.1.

8.4.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the five CI methods (M5'-MGGP, M5'-AMGGP, SVR, ANFIS and ANN) are illustrated in Tables 8.12 and 8.13 on training and testing data respectively. The best prediction method that gives good generalization ability is determined by comparing these methods using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The results of the training phase (Table 8.12) indicate that all the five models have impressively learned the non-linear relationship between the input variables and compressive strength with high correlation values and relatively low error values. The results of the testing phase (Table 8.12) indicate that the models (M5'-AMGGP and ANN) outperformed the other three with value of R^2 achieved as high as 0.96. Between M5'-AMGGP and ANN, both have shown comparable performance. Among M5'-MGGP, SVR and ANFIS, M5'-MGGP has shown better performance followed by SVR and ANFIS.

MO error values for the five models are computed on the training and testing data and shown in Table 8.13. It shows that the two models (M5'-AMGGP and ANN)

outperforms the other three models. The descriptive statistics of the relative error (%) of the five models are shown in Table 8.14. The lower value of range of confidence intervals and remaining statistical metrics of the M5'-AMGGP and ANN models indicate that their performance is better than those of the other three models. Between M5'-AMGGP and ANN, both have shown comparable performance. Among M5'-MGGP, SVR and ANFIS, M5'-MGGP has shown better performance followed by SVR and ANFIS.

Hypothesis tests are also used to compare the goodness of fit of the five models. These tests are conducted for the model formation process and the results are shown in Table 8.15. For the *t*-test and *f*-test, the *p*-values of all the models are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these models differ. Therefore, all five prediction models have statistically satisfactory goodness of fit from the modelling point of view.

Thus, from the statistical comparison presented, it can be concluded that the proposed M5'-AMGGP and ANN models have outperformed the other three models. Between M5'-AMGGP and ANN, both have shown comparable performance.

Table 8.12 R^2 , MAPE (%) and RMSE of the five models

Error measure metrics	R^2		MAPE (%)		RMSE	
	Training data	Testing data	Training data	Testing data	Training data	Testing data
M5'- AMGGP	0.97	0.96	1.86	2.30	0.35	0.37
M5'-MGGP	0.96	0.94	2.61	3.04	0.43	0.45
SVR	0.92	0.93	4.04	3.16	0.67	0.47
ANFIS	0.92	0.88	2.57	6.15	0.60	0.88
ANN	0.97	0.96	2.01	2.43	0.36	0.36

Table 8.13 Multi-objective error of the five models

Methods	Training data	Testing data
M5'-AMGGP	2.27	2.78
M5'-MGGP	3.16	3.71
SVR	5.11	3.90
ANFIS	3.44	7.98
ANN	2.44	2.90

Table 8.14 Descriptive statistics for relative error (%) of the five models

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
M5'-AMGGP	32	1.95	1.99	1.23	2.67	0.38	1.34	9.61	0.01
M5'-MGGP	32	2.71	2.45	1.82	3.59	0.43	2.22	9.61	0.04
SVR	32	3.85	3.92	2.44	5.26	0.69	2.55	16.91	0.06
ANFIS	32	3.36	4.73	1.65	5.06	0.83	1.47	18.72	0.03
ANN	32	2.13	2.16	1.35	2.91	0.38	1.34	9.61	0.01

Table 8.15 Hypothesis testing to compare the five prediction models

95%CI	<i>p</i> -value				
	M5'-AMGGP	M5'-MGGP	SVR	ANFIS	ANN
Mean paired <i>t</i> -tests	0.6546	0.2177	0.0931	0.0513	0.8875
Variance <i>f</i> -test	0.8814	0.9928	0.4262	0.7624	0.9542

8.5 Problem: Vibratory finishing process

The motivation behind solving this problem and the details about the process variables considered have been discussed in Section 7.2. In this section, the hybridized approach of M5'-AMGGP is applied to model the three performance characteristics of the vibratory finishing process. The performance of the M5'-AMGGP method is compared to those of the standardized MGGP approach, MGGP-ANN, SVR, ANFIS and ANN. The K-S algorithm is applied to select seventeen samples as training data with the remaining five as a set of test data.

8.5.1 Results and Discussion

8.5.1.1 Parameter Settings of CI Methods

The implementation of the M5'-AMGGP method is carried out in two phases. In the first phase, M5' algorithm is executed. In the second phase, AMGGP is implemented in MATLAB R2010b. The parameter settings for M5'-AMGGP are set based on a trial-and-error approach (Table 8.16 and Fig. 8.5). The fitness function used is SRM with optimum order of polynomial as the complexity measure of the model.

Table 8.16 Parameter settings for M5'

Variables	Values assigned
Minimum number of cases	3
Smoothing	Yes
Pruning	Yes
Split threshold	0.05

Fig. 8.5 GUI showing the parameter settings chosen for M5'-AMGGP

The error predicted using the AMGGP model is used compensate the M5' model for the formulation of the M5'-AMGGP model given by:

$H_{M5'-AMGGP} =$

(if $x_3 \leq 90$

$y = 47.71 - 28 * x_4$

else

$$y = 29.63 + E_{MGGP} \quad (8.5)$$

Edge radius $M5'-AMGGP =$

(if $x_5 \leq 1865$

if $x_3 \leq 45$

$y = 1.5$

else

$y = 4.3$

else

$$y = 20.13 + E_{GP} \quad (8.6)$$

$S_{M5'-AMGGP} =$

$$y = 17.55 - 10.4 * x_4 + E_{GP} \quad (8.7)$$

The parameter settings for the methods (MGGP and MGGP-ANN) are the same as those discussed in Section 7.2.1.1. ANFIS is implemented using ANFIS tool box built in MATLAB R2010b. Since, the number of inputs is more than four, the subtractive clustering is used as a pre-processing tool for obtaining clusters. The Gaussian membership function is used for each input and the set of rules are

generated. A hybrid method of least squares and BP-NN is used to train the FIS. For the stopping criterion, the error tolerance is kept at zero and the number of epochs is set at 50. ANN is implemented in a tool box built in MATLAB R2010b. The optimal architecture of ANN is determined based on a trial-and-error approach. Based on minimum RMSE on training data, the network of single hidden layer with number of neurons eight is selected for the analysis. The remaining parameter settings are same as those mentioned in Section 7.1.2.1. SVR is implemented using LS-SVM tool box built in MATLAB. The kernel function chosen is RBF. The parameters λ and σ of RBF are estimated using a hybrid approach of CSA and a grid search method. A SVR model with optimal variables $\lambda = 1256.84$ and $\sigma = 1.42$ are found at the second iteration.

8.5.1.2 Evaluation and Statistical Comparison of CI Methods

The results obtained from the six CI methods (MGGP, M5'-AMGGP, MGGP-ANN, SVR, ANFIS and ANN) for the three outputs are illustrated in Tables 8.17-8.20 on training and testing data respectively. The best prediction method that gives good generalization ability is determined by comparing these methods on each output using the five metrics: R^2 , MAPE (%), RMSE, relative error (%) and MO error.

The results of the training phase (Tables 8.17-8.19) for the three outputs respectively, indicate that all the six models have impressively learned the non-linear relationship between the input variables and the three outputs with high correlation values and relatively low error values. The result of the testing phase (Tables 8.17-8.19) for the three outputs respectively, indicate that the models M5'-AMGGP and MGGP-ANN outperformed the other four. For the output H ,

between M5'-AMGGP and MGGP-ANN, both have shown comparable performance. Among MGGP, SVR, ANFIS and ANN, SVR has shown better performance followed by MGGP, ANN and ANFIS. For the output E , between M5'-AMGGP and MGGP-ANN, MGGP-ANN has shown better performance. Among MGGP, SVR, ANFIS and ANN, SVR has shown better performance followed by ANN, ANFIS and MGGP. For the output S , between M5'-AMGGP and MGGP-ANN, M5'-AMGGP has shown better performance. Among MGGP, SVR, ANFIS and ANN, SVR has shown better performance followed by ANN, ANFIS and MGGP.

MO error values for the six models are computed on training and testing data of three outputs and shown in Table 8.20. It shows that the two models (M5'-AMGGP and MGGP-ANN) outperform the other four models. The descriptive statistics of the relative error (%) of the six models for the three outputs are shown in Tables 8.21-8.23 respectively. The lower value of range of confidence intervals and remaining statistical metrics of the M5'-AMGGP and MGGP-ANN models indicate that their performance is better than those of the other four models. Between M5'-AMGGP and MGGP-ANN, both have shown comparable performance.

Hypothesis tests are also used to compare the goodness of fit of the six models for the three outputs. These tests are conducted for the model formation process and the results are shown in Tables 8.24-8.26. For the t -test and f -test, the p -values of all the models are greater than 0.05, so there is not enough evidence to conclude that the actual values and predicted values from these models differ. Therefore, all six prediction models have statistically satisfactory goodness of fit from the modelling point of view.

Thus, from the statistical comparison presented, it can be concluded that the proposed M5'-AMGGP and MGGP-ANN models have outperformed the other four models. Between M5'-AMGGP and MGGP-ANN, both have shown comparable performance. Among MGGP, SVR, ANFIS and ANN, SVR has shown better performance followed by ANN, ANFIS and MGGP.

Table 8.17 R^2 , MAPE (%) and RMSE of the six models for output H

Error measure metrics	R^2		MAPE (%)		RMSE	
	Training data	Testing data	Training data	Testing data	Training data	Testing data
M5'-AMGGP	0.99	0.96	2.35	8.30	1.06	6.74
MGGP	0.89	0.92	17.63	16.54	4.53	4.46
MGGP-ANN	0.98	0.96	4.60	7.31	1.42	3.39
SVR	0.98	0.96	3.75	14.61	1.17	7.13
ANFIS	0.95	0.90	8.00	14.70	2.55	8.58
ANN	0.99	0.93	3.01	14.62	1.09	9.73

Table 8.18 R^2 , MAPE (%) and RMSE of the six models for output E

Error measure metrics	R^2		MAPE (%)		RMSE	
	Training data	Testing data	Training data	Testing data	Training data	Testing data
M5'-AMGGP	0.99	0.97	5.77	7.56	0.27	0.82
MGGP	0.99	0.85	56.04	31.24	1.81	1.85
MGGP-ANN	0.99	0.98	16.41	4.54	0.83	0.31
SVR	0.98	0.95	13.76	15.64	4.19	1.08
ANFIS	0.99	0.93	9.80	17.50	0.29	1.25
ANN	0.99	0.92	6.39	25.50	0.29	1.87

Table 8.19 R^2 , MAPE (%) and RMSE of the six models for output S

Error measure metrics	R^2		MAPE (%)		RMSE	
	Training data	Testing data	Training data	Testing data	Training data	Testing data
M5'-AMGGP	0.96	0.97	5.10	7.84	0.93	1.26
MGGP	0.81	0.77	26.30	17.15	2.17	2.80
MGGP-ANN	0.92	0.96	13.82	9.07	1.28	1.21
SVR	0.93	0.94	7.52	7.94	1.34	1.14
ANFIS	0.90	0.84	7.46	12.36	1.49	2.59
ANN	0.95	0.92	4.23	10.37	0.90	1.32

Table 8.20 Multi-objective error of the six models on the three outputs

Outputs	H		E		S	
Methods	Training data	Testing data	Training data	Testing data	Training data	Testing data
M5'-AMGGP	3.44	15.66	6.10	8.63	6.28	9.38
MGGP	24.89	22.82	58.43	38.92	35.14	25.90
MGGP-ANN	6.14	11.14	17.45	4.94	16.41	10.70
SVR	5.02	22.64	18.31	17.6	9.52	9.65
ANFIS	11.10	25.86	10.19	20.16	9.94	17.79
ANN	4.14	26.18	6.74	29.75	5.40	12.70

Table 8.21 Descriptive statistics for relative error (%) of the six models for the output H

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
M5'-AMGGP	22	3.70	6.55	0.80	6.61	1.39	0.81	26.51	0.04
MGGP	22	17.38	14.73	10.85	23.91	3.14	15.22	56.16	0.48
MGGP-ANN	22	5.22	7.00	2.11	8.32	1.49	2.02	25.14	0.05
SVR	22	6.21	8.12	2.61	9.81	1.73	2.52	33.43	0.53
ANFIS	22	9.52	15.12	2.81	16.23	3.22	2.24	56	0.09
ANN	22	5.65	9.21	1.56	9.73	1.96	1.94	35.66	0.03

Table 8.22 Descriptive statistics for relative error (%) of the six models for the output E

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
M5'-AMGGP	22	6.18	7.35	2.92	9.44	1.56	2.33	22.5	0.0004
MGGP	22	50.41	53.61	26.64	74.48	11.43	40.60	235.77	1.25
MGGP-ANN	22	13.72	31.54	-0.26	27.70	6.72	5.58	151.57	0.06
SVR	22	14.19	14.48	7.76	20.61	3.08	10.13	57.25	1.422
ANFIS	22	11.55	14.89	4.95	18.16	3.17	9.37	67.5	0.022
ANN	22	10.73	14.27	4.40	17.06	3.04	6.33	60.5	0.056

Table 8.23 Descriptive statistics for relative error (%) of the six models for the output S

Models	Count	Mean	Std dev	LCI 95%	UCI 95%	SE mean	Median	Maximum	Minimum
M5'-AMGGP	22	5.72	6.98	2.63	8.82	1.48	1.37	22.55	0.01
MGGP	22	24.22	22.24	14.36	34.09	4.74	17.50	114	5.37
MGGP-ANN	22	12.74	17.44	5.006	20.47	3.72	5.98	63.69	1.61
SVR	22	7.57	10.92	2.72	12.41	2.32	1.47	42	0.03
ANFIS	22	8.57	15.44	1.73	15.42	3.29	0.55	58.88	0.06
ANN	22	5.63	9.22	1.54	9.72	1.96	0.95	37.77	0.006

Table 8.24 Hypothesis testing to compare the six prediction models for the output H

95%CI	H					
	M5'-AMGGP	MGGP-ANN	MGGP	SVR	ANFIS	ANN
Mean paired t -tests	0.3288	0.6956	0.0511	0.2218	0.2126	0.2309
Variance f -test	0.6220	0.9743	0.9724	0.6684	0.5895	0.5341

Table 8.25 Hypothesis testing to compare the six prediction models for the output E

95%CI	E					
	M5'-AMGGP	MGGP-ANN	MGGP	SVR	ANFIS	ANN
Mean paired t -tests	0.5645	0.0723	0.1331	0.2336	0.2773	0.1292
Variance f -test	0.9990	0.9955	0.9464	0.3216	0.9918	0.9946

Table 8.26 Hypothesis testing to compare the six prediction models for the output S

95%CI	S					
	M5'-AMGGP	MGGP-ANN	MGGP	SVR	ANFIS	ANN
Mean paired t -tests	0.7758	0.5883	0.3269	0.0705	0.8100	0.4820
Variance f -test	0.9938	0.9728	0.2705	0.8458	0.6892	0.8818

8.6 Summary

This chapter shows the applications of the proposed CI approach of M5'-AMGGP in modelling of various manufacturing systems such as turning, FDM and vibratory finishing processes. Its performance is compared to those of the methods M-MGGP, MGGP-ANN, M5'-MGGP, SVR, ANFIS and ANN. In most of the applications, the generalization ability of M5'-AMGGP approach is found to be better than those of the other proposed methods (M-MGGP, MGGP-ANN and M5'-MGGP), whereas for all the others (MGGP, SVR, ANFIS and ANN) their performance were found to be comparable. The sensitivity and parametric analysis conducted of the M5'-AMGGP model on the four processes were somewhat found to be same as those shown in previous chapters.

CHAPTER 9

CONCLUDING REMARKS AND FUTURE WORK

9.1 *Concluding Remarks*

Chapters 4, 5, 6, 7 and 8 discussed the performance of the proposed CI methods in modeling of turning, FDM and vibratory finishing processes. The methods were proposed with a special focus on improving the generalization ability of MGGP. The chapters compare the performance of the proposed methods to those of standardized methods such as SVR, ANFIS and ANN. The concluding remarks on these respective systems modeling are as follows:

- (a) The M-MGGP approach for modelling the two processes (turning of AISI H11 steel and turning of dynamic aging regime of ASS steel process) results in accurate models. Thus, we have successfully formed a robust version of MGGP, by selecting and combining only the relevant genes of higher performance. Although the implementation of M-MGGP method is time consuming, that can be compensated by the better generalization ability of the evolved model. The higher generalization ability of the M-MGGP model is beneficial for industry experts who are currently looking for high fidelity models that predict the response characteristics of the processes under uncertain input process conditions, and therefore the cost of having to run additional experiments can be avoided.

- (b) Quantitative analysis conducted for studying the impact of several fitness functions using the three measures of complexity of the model (number of nodes, polynomials and MARS) revealed that the fitness function SRM using polynomial measure as complexity of the model has resulted in evolution of higher generalized MGGP models. The reason for such good performance is because the SRM principle is able to strike a good balance between the empirical error and size of the model. It was expected, since the same SRM principle is responsible for inducing the generalization capability in the SVR approach by putting upper and lower bounds on the empirical error. The high generalization ability of the SRM-based-MGGP model is beneficial for production experts who are currently looking for high fidelity models that predict the process behaviour under uncertain input process conditions and therefore the cost of having to run additional experiments can be avoided. Limitation of the analysis is that the complexity is measured only as a function of size of the tree, whereas the computational time has been ignored.
- (c) The classification scheme introduced in the proposed C-MGGP method has improved the selection of a best model from the population comprising of optimum solutions. Although models can be found whose performance are better than the one selected by the C-MGGP method, the number of such models is very few when compared to those in standardized MGGP.
- (d) The hybrid M1-M2 approach for modelling vibratory finishing and FDM process produces generalized models. By using M2 method in parallel it

provides compensation to the M1 model and the sense of trustworthiness in the prediction ability of the model on the unseen samples. Since it is a parallel approach, the computational time is the limitation of this method.

- (e) The overall methodology of M5'-AMGGP has resulted in evolution of highly generalized models. Quantitative comparison conducted between the different sets of methods reveals that the performance of M5'-AMGGP is better than those of other methods. It is not possible to compare the performance among the remaining methods, since the performance of each varies significantly depending on the nature of the data. The improved performance of the M5'-AMGGP approach is due to weighted effects of several factors such as the new procedure of combination of multiple sets of genes, application of the new complexity measure in the fitness function, improving the model selection by classification and ensuring greater trustworthiness in the prediction ability of the model. The effect of each factor on the generalization ability of the approach varies depending on the nature of the data and parameter settings.

9.2 Original Contributions Arising from the Work

- (a) For the first time, the study addresses the need of formulating the explicit and generalized model using MGGP for understanding the mechanism and hidden non-linear relationships of a few manufacturing processes. In addition, despite the recent advancements in MGGP, the issue of procedure of formation of the MGGP model has not yet been addressed. In this perspective, M-MGGP approach is proposed that inducts the stepwise approach in its evolutionary paradigm. Unlike standardized MGGP, M-MGGP approach wisely selects the

genes and combines them to form a compact and generalized model. The analysis of M-MGGP model on two processes proved to be able to capture the dynamics of the process, which comply well with the previous study on turning operation.

- (b) The literature focussed extensively on using the number of nodes of the tree as a complexity measure of the MGGP model. Choosing an appropriate complexity measure is essential for preventing the *over-fitting* of the evolved models in MGGP. This study proposed two new complexity measures based on orthogonal basis functions: polynomials and MARS. These measures were incorporated in the regularized fitness functions for the formation of new fitness functions. Quantitative analysis conducted proves that the fitness function “SRM” using polynomial as complexity measure results in evolution of higher generalized MGGP models.
- (c) Numerous applications and preliminary studies of MGGP suggest that there are many models in the population whose performance on the testing data is better than that of the best model. Based on the author’s knowledge, very little work has been focussed on this. For choosing the best model from the population of models, a classification-based-MGGP approach (C-MGGP) is proposed. The approach inducts the classifications strategies in the evolutionary stage and classifies models into either of two categories “bad” or “good”. This requires the study and implementation of classification methods existing in ML field of SI. The analysis of the proposed approach on the data sets reveals that there is a significant improvement in model selection.

- (d) The literature reveals that ensuring a greater trustworthiness of the prediction ability of the model on unseen samples is a vital issue and has not been paid much attention. For the first time, a hybrid parallel approach of two methods is proposed, wherein one of them acts as an error model predicting the error of the other model, and thus provides a sense of trustworthiness to the prediction ability of the other model on the unseen data samples. Analysis of the proposed model on the data sets reveals that the approach works well on fewer data samples and limit the necessity of tuning of one of the methods.
- (e) Four issues on generalization such as inappropriate procedure of formulation of the MGGP model, inappropriate complexity measure of the MGGP model, model selection and ensuring greater trustworthiness of prediction ability of the model on unseen samples are collectively tackled by the formulation of an overall CI approach. This CI approach was basically formulated by incorporating principles from several fields of SI such as statistics, statistical learning theory, data mining, ML, econometrics, etc. This certainly requires the extensive study on methods, systems and models in various fields of SI. The performance of the proposed approach was compared to those of the other methods such as MGGP, SVR, ANFIS and ANN. Applications of the proposed approach reveal that its performance is better than those of the other standardized methods. However, computational time of the proposed method is more than that of the standardized methods, which can be compensated by the higher accuracy achieved.

9.3 Recommendation for Future Work

There are still some gaps in the proposed methods which should be further resolved for achieving the required objectives as follows:

- (a) Induction of the stepwise approach in M-MGGP selectively combines the genes and removes the irrelevant ones based on the p -value. The p -value was set at standardized 0.05. A lower p -value results in maximum removal of genes from the population and thereby increasing the computational time of the proposed approach. The M-MGGP approach needs to be further investigated by varying the p -value gradually and observing the size and accuracy of the models on different sets of data.
- (b) The study presents two new measures for measuring the complexity of the MGGP model: 1) Polynomials 2) MARS. Choosing optimum order of polynomial to fit a given MGGP model takes time, but it gives better accuracy. On the other hand, fitting MARS to the MGGP model gives lower accuracy but takes less time in its evaluation. Future work may be done to investigate the hybrid of these two measures and including computational time as an additional objective in evaluating the performance of the models.
- (c) The C-MGGP methodology comprises of a two way procedure: first forming the classifiers and then implementing it in MGGP. The implementation consumes a significant amount of time. Future work to be done includes the development of a framework for the model selection in MGGP that is less

computationally expensive and works faster than that of the standardized MGGP.

- (d) More sophisticated CI methods such as variants of GP have been developed. Future work will include the comparison of the proposed hybrid M1-M2 approach to those of the advanced variants in GP such as GEP, C-GP, etc.
- (e) Future work will include the comparison of the proposed CI approach to those of the several ensemble methods of CI for the empirical study of various RP and machining processes. In this way, a guide can be developed for assisting industry experts and researchers in choosing an appropriate method for modelling any given RP or machining process.

Reference

- [1] M. Willis, H. Hiden, M. Hinchliffe, B. McKay, and G. W. Barton, "Systems modelling using genetic programming," *Computers & chemical engineering*, vol. 21, pp. S1161-S1166, 1997.
- [2] M. Deistler, "System identification and time series analysis: Past, present, and future," *Stochastic Theory and Control*, pp. 97-109, 2002.
- [3] M. Chandrasekaran, M. Muralidhar, C. M. Krishna, and U. Dixit, "Application of soft computing techniques in machining performance prediction and optimization: a literature review," *The International Journal of Advanced Manufacturing Technology*, vol. 46, pp. 445-464, 2010.
- [4] E. Vladislavleva and G. Smits, "Symbolic regression via genetic programming," *Final Thesis for Dow Benelux BV*.
- [5] U. Çaydaş and S. Ekici, "Support vector machines models for surface roughness prediction in CNC turning of AISI 304 austenitic stainless steel," *Journal of Intelligent Manufacturing*, vol. 23, pp. 639-650, 2012.
- [6] S. N. Patra, R. J. T. Lin, and D. Bhattacharyya, "Regression analysis of manufacturing electrospun nonwoven nanotextiles," *Journal of Materials Science*, vol. 45, pp. 3938-3946, // 2010.
- [7] K. J. Astrom and P. Eykhoff, "System identification--A survey," *Automatica*, vol. 7, pp. 123-162, 1971.
- [8] L. Ljung, "Perspectives on system identification," *Annual Reviews in Control*, vol. 34, pp. 1-12, 2010.
- [9] S. Sette and L. Boullart, "Genetic programming: principles and applications," *Engineering applications of artificial intelligence*, vol. 14, pp. 727-736, 2001.
- [10] X. Hong, R. Mitchell, S. Chen, C. J. Harris, K. Li, and G. Irwin, "Model selection approaches for nonlinear system identification: a review," *International Journal of Systems Science*, vol. 39, pp. 925-946, 2008.
- [11] S. Billings, "Identification of nonlinear systems--A survey," 1980, pp. 272-285.
- [12] M. Affenzeller and S. Winkler, *Genetic algorithms and genetic programming: modern concepts and practical applications* vol. 6: Chapman & Hall/CRC, 2009.
- [13] Y. Ku and A. A. Wolf, "Volterra-Wiener functionals for the analysis of nonlinear systems* 1," *Journal of The Franklin Institute*, vol. 281, pp. 9-26, 1966.
- [14] L. A. Zadeh, "From circuit theory to system theory," *Proceedings of the IRE*, vol. 50, pp. 856-865, 1962.
- [15] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, vol. 4, pp. 87-112, 1994.
- [16] A. Garg, Y. Bhalerao, and K. Tai, "Review of empirical modelling techniques for modelling of turning process," *International Journal of Modelling, Identification and Control*, vol. 20, pp. 121-129, 2013.
- [17] M. M. Savalani, A. Garg, and K. Tai, "State-of-the-art in empirical modelling of rapid prototyping processes," *Rapid Prototyping Journal*, vol. 20, pp. 9-9, 2014.

- [18] A. Garg and K. Tai, "Comparison of statistical and machine learning methods in modelling of data with multicollinearity," *International Journal of Modelling, Identification and Control*, vol. 18, pp. 295-312, 2013.
- [19] Z. Yang, X. S. Gu, X. Y. Liang, and L. C. Ling, "Genetic algorithm-least squares support vector regression based predicting and optimizing model on carbon fiber composite integrated conductivity," *Materials & Design*, vol. 31, pp. 1042-1049, 3// 2010.
- [20] A. Garg, K. Tai, C. Lee, and M. Savalani, "A hybrid\ text {M} 5^\ prime-genetic programming approach for ensuring greater trustworthiness of prediction ability in modelling of FDM process," *Journal of Intelligent Manufacturing*, pp. 1-17, 2013.
- [21] D. Umbrello, G. Ambrogio, L. Filice, and R. Shivpuri, "A hybrid finite element method–artificial neural network approach for predicting residual stresses and the optimal cutting conditions during hard turning of AISI 52100 bearing steel," *Materials & Design*, vol. 29, pp. 873-883, // 2008.
- [22] B. Wang, X. Wang, and Z. Chen, "A hybrid framework for reservoir characterization using fuzzy ranking and an artificial neural network," *Computers and Geosciences*, vol. 57, pp. 1-10, // 2013.
- [23] Y. G. Liu, J. Luo, and M. Q. Li, "The fuzzy neural network model of flow stress in the isothermal compression of 300M steel," *Materials and Design*, vol. 41, pp. 83-88, 2012.
- [24] W. Li, Y. Yang, Z. Yang, and C. Zhang, "Fuzzy system identification based on support vector regression and genetic algorithm," *International Journal of Modelling, Identification and Control*, vol. 12, pp. 50-55, 2011.
- [25] A. Garg, "A Fuzzy-Taguchi Approach for Improving Dimensional Accuracy of Fused Deposition Modelling (FDM) Built Parts," 2010.
- [26] I. Mukherjee and P. K. Ray, "A review of optimization techniques in metal cutting processes," *Computers & Industrial Engineering*, vol. 50, pp. 15-34, 2006.
- [27] J. R. Quinlan, "Learning with continuous classes," 1992, pp. 343-348.
- [28] Y. Wang and I. H. Witten, "Induction of model trees for predicting continuous classes," 1996.
- [29] L. Wei-Po, J. Hallam, and H. H. Lund, "A hybrid GP/GA approach for co-evolving controllers and robot bodies to achieve fitness-specified tasks," in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, 1996, pp. 384-389.
- [30] H. Xie, M. Zhang, and P. Andreae, "Population Clustering in Genetic Programming," in *Genetic Programming*. vol. 3905, P. Collet, M. Tomassini, M. Ebner, S. Gustafson, and A. Ekárt, Eds., ed: Springer Berlin Heidelberg, 2006, pp. 190-201.
- [31] K. Kumarci, P. Dehkordi, and I. Mahmodi, "Calculation of Plate Natural Frequency by Genetic Programming," *Journal of Applied Sciences*, vol. 10, pp. 451-461, 2010.
- [32] J. Madár, J. Abonyi, and F. Szeifert, "Genetic programming for the identification of nonlinear input-output models," *Industrial & engineering chemistry research*, vol. 44, pp. 3178-3186, 2005.
- [33] G. Folino, C. Pizzuti, and G. Spezzano, "Genetic Programming and Simulated Annealing: A Hybrid Method to Evolve Decision Trees," in *Genetic Programming*. vol. 1802, R. Poli, W. Banzhaf, W. Langdon, J.

- Miller, P. Nordin, and T. Fogarty, Eds., ed: Springer Berlin Heidelberg, 2000, pp. 294-303.
- [34] A. Garg and K. Tai, "Review of genetic programming in modeling of machining processes," in *Modelling, Identification & Control (ICMIC), 2012 Proceedings of International Conference on*, 2012, pp. 653-658.
 - [35] A. Garg and K. Tai, "Comparison of regression analysis, Artificial Neural Network and genetic programming in Handling the multicollinearity problem," 2012, pp. 353-358.
 - [36] A. Garg and K. Tai, "Selection of a robust experimental design for the effective modeling of the nonlinear systems using genetic programming," in *Proceedings of 2013 IEEE Symposium on Computational Intelligence and Data mining (CIDM), Singapore*, 2013, pp. 293-298.
 - [37] A. Garg, V. Vijayaraghavan, S. Mahapatra, K. Tai, and C. Wong, "Performance evaluation of microbial fuel cell by artificial intelligence methods," *Expert Systems with Applications*, vol. 41, pp. 1389-1399, 2014.
 - [38] A. Garg, S. Sriram, and K. Tai, "Empirical analysis of model selection criteria for genetic programming in modeling of time series system," in *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2013 IEEE Conference on*, 2013, pp. 90-94.
 - [39] I. Gonçalves, S. Silva, J. Melo, and J. B. Carreiras, "Random Sampling Technique for Overfitting Control in Genetic Programming," in *Genetic Programming*. vol. 7244, A. Moraglio, S. Silva, K. Krawiec, P. Machado, and C. Cotta, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 218-229.
 - [40] U. M. O'Reilly, *Genetic programming theory and practice II* vol. 2: Springer-Verlag New York Inc, 2005.
 - [41] A. Kordon, F. Castillo, G. Smits, and M. Kotanchek, "Application issues of genetic programming in industry," *Genetic Programming Theory and Practice III*, pp. 241-258, 2006.
 - [42] G. Kronberger, C. Feilmayr, M. Kommenda, S. Winkler, M. Affenzeller, and T. Burgler, "System Identification of Blast Furnace Processes with Genetic Programming," pp. 1-6.
 - [43] A. Nandi and D. Pratihari, "An expert system based on FBFN using a GA to predict surface finish in ultra-precision turning," *Journal of materials processing technology*, vol. 155, pp. 1150-1156, 2004.
 - [44] G. C. Onwubolu and S. Kumar, "Response surface methodology-based approach to CNC drilling operations," *Journal of materials processing technology*, vol. 171, pp. 41-47, 2006.
 - [45] T.-I. Liu, G. Liu, and Z. Gao, "In-process and post-process measurements of drill wear for control of the drilling process," in *Seventh International Symposium on Precision Engineering Measurements and Instrumentation*, 2011, pp. 83213T-83213T-8.
 - [46] P. Benardos and G. C. Vosniakos, "Predicting surface roughness in machining: a review," *International Journal of Machine Tools and Manufacture*, vol. 43, pp. 833-844, 2003.
 - [47] F. Hashimoto and D. DeBra, "Modelling and optimization of vibratory finishing process," *CIRP Annals-Manufacturing Technology*, vol. 45, pp. 303-306, 1996.
 - [48] K. Risbood, U. Dixit, and A. Sahasrabudhe, "Prediction of surface roughness and dimensional deviation by measuring cutting forces and

- vibrations in turning process," *Journal of materials processing technology*, vol. 132, pp. 203-214, 2003.
- [49] S. Winkler, H. Efendic, L. Del Re, and M. Affenzeller, "Online modelling based on Genetic Programming," *International Journal of Intelligent Systems Technologies and Applications*, vol. 2, pp. 255-270, 2007.
 - [50] T.-I. Liu, J. Lee, P. Singh, and G. Liu, "Using acceleration measurements and neuro-fuzzy systems for monitoring and diagnosis of bearings," in *Sixth International Symposium on Precision Mechanical Measurements*, 2013, pp. 89160B-89160B-7.
 - [51] T.-I. Liu, J. Lee, and P. Singh, "On-line detection of ball bearing failures by an intelligent technique," in *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, 2008, pp. 346-350.
 - [52] B. Andersson and M. Persson, "Software Reliability Prediction," 2004.
 - [53] B. McKay, M. Willis, and G. Barton, "Steady-state modelling of chemical process systems using genetic programming," *Computers & chemical engineering*, vol. 21, pp. 981-996, 1997.
 - [54] R. E. Keller and W. Banzhaf, "Genetic programming using genotype-phenotype mapping from linear genomes into linear phenotypes," 1996, pp. 116-122.
 - [55] P. A. Fritzson, *Principles of object-oriented modeling and simulation with Modelica 2.1*: Wiley-IEEE Press, 2004.
 - [56] D. R. Anderson, D. J. Sweeney, and T. A. Williams, *Statistics for business and economics*: South-Western Pub, 2010.
 - [57] J. Neter, W. Wasserman, M. H. Kutner, and W. Li, *Applied linear statistical models*: Irwin, 1996.
 - [58] D. Card, "The causal effect of education on earnings," *Handbook of labor economics*, vol. 3, pp. 1801-1863, 1999.
 - [59] F. De Leeuw and E. Gramlich, *The Federal Reserve-MIT Econometric Model:(Staff Economic Study)*: Board of Governors of the Federal Reserve System, 1968.
 - [60] W. L. Megginson, "Toward a global model of venture capital?," *Journal of Applied Corporate Finance*, vol. 16, pp. 89-107, 2004.
 - [61] B. M. Pötscher and I. R. Prucha, "Contributions to econometrics, time-series analysis, and systems identification: a Festschrift in honor of Manfred Deistler," *Journal of Econometrics*, vol. 118, pp. 1-5, 2004.
 - [62] A. Garg, S. Sriram, and K. Tai, "Empirical analysis of model selection criteria for genetic programming in modeling of time series system," in *Proceedings of 2013 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, Singapore, 2013, pp. 84-88.
 - [63] C. Grosan and A. Abraham, "Stock market modeling using genetic programming ensembles," *Genetic Systems Programming*, pp. 131-146, 2006.
 - [64] N. Wagner, Z. Michalewicz, M. Khouja, and R. R. McGregor, "Time series forecasting for dynamic environments: the DyFor genetic program model," *Evolutionary Computation, IEEE Transactions on*, vol. 11, pp. 433-452, 2007.
 - [65] G. Zhang, X. Xiang, and H. Tang, "Time Series Prediction of Chimney Foundation Settlement by Neural Networks," *International Journal of Geomechanics*, vol. 11, pp. 154-158, // 2011.

- [66] M. Kaboudan, "Genetic programming prediction of stock prices," *Computational Economics*, vol. 16, pp. 207-236, 2000.
- [67] H. Iba and N. Nikolaev, "Genetic programming polynomial models of financial data series," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 2000, pp. 1459-1466.
- [68] M. Santini and A. Tettamanzi, "Genetic programming for financial time series prediction," *Genetic Programming*, pp. 361-370, 2001.
- [69] H. Akaike, "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol. 19, pp. 716-723, 1974.
- [70] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, pp. 461-464, 1978.
- [71] A. R. Barron, "Predicted squared error: a criterion for automatic model selection," 1984.
- [72] V. Vapnik, "The Nature of Statistical Learning," ed: Springer, NY, 1995.
- [73] C. M. Bishop and SpringerLink, *Pattern recognition and machine learning* vol. 4: Springer New York, 2006.
- [74] G. E. Hinton and T. J. Sejnowski, *Unsupervised learning: foundations of neural computation*: MIT press, 1999.
- [75] S. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering: real word AI systems with applications in eHealth, HCI, information retrieval and pervasive technologies*, vol. 160, p. 3, 2007.
- [76] Y. Maheshkumar, V. Ravi, and A. Abraham, "A particle swarm optimization-threshold accepting hybrid algorithm for unconstrained optimization," *Neural Network World*, vol. 23, pp. 191-222, 2013.
- [77] L. Wang, B. Yang, Y. Chen, A. Abraham, H. Sun, Z. Chen, *et al.*, "Improvement of neural network classifier using floating centroids," *Knowledge and information systems*, vol. 31, pp. 433-454, 2012.
- [78] M. A. Sharaf, D. L. Illman, and B. R. Kowalski, *Chemometrics* vol. 82: Wiley-Interscience, 1986.
- [79] A. Garg and K. Tai, "Selection of a robust experimental design for the effective modeling of nonlinear systems using genetic programming," in *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on*, 2013, pp. 287-292.
- [80] B. Can and C. Heavey, "Comparison of experimental designs for simulation-based symbolic regression of manufacturing systems," *Computers & Industrial Engineering*, 2011.
- [81] D. D. Wolff and M. L. Parsons, "Pattern recognition approach to data interpretation," *New York: Plenum Press, 1983*, vol. 1, 1983.
- [82] E. MOLINOWSKI, "Factor analysis in chemistry," 2002.
- [83] M. Kantardzic and I. Press, *Data mining: concepts, models, methods, and algorithms*: Wiley-Interscience, 2003.
- [84] S. J. Lee and K. Siau, "A review of data mining techniques," *Industrial Management & Data Systems*, vol. 101, pp. 41-46, 2001.
- [85] J. R. Westlake, *A handbook of numerical matrix inversion and solution of linear equations* vol. 767: Wiley New York, 1968.
- [86] M. Ezekiel, "Methods of correlation analysis," 1930.
- [87] T. Kariya and H. Kurata, *Generalized least squares* vol. 560: John Wiley & Sons Inc, 2004.

- [88] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1-38, 1977.
- [89] D. O. Staiger and J. H. Stock, "Instrumental variables regression with weak instruments," ed: National Bureau of Economic Research Cambridge, Mass., USA, 1994.
- [90] D. Ruppert and M. Wand, "Multivariate locally weighted least squares regression," *The annals of statistics*, pp. 1346-1370, 1994.
- [91] D. J. Wilde, *Optimum seeking methods* vol. 14: Prentice-Hall Englewood Cliffs, NJ;, 1964.
- [92] G. A. Bekey and R. B. McGhee, "Gradient methods for the optimization of dynamic system parameters by hybrid computation," 1964, p. 305.
- [93] J. Mills and K. Prasad, "A comparison of model selection criteria," *Econometric reviews*, vol. 11, pp. 201-234, 1992.
- [94] X. Hong and C. Harris, "Nonlinear model structure design and construction using orthogonal least squares and D-optimality design," *Neural Networks, IEEE Transactions on*, vol. 13, pp. 1245-1250, 2002.
- [95] R. Haber and H. Unbehauen, "Structure identification of nonlinear dynamic systems--a survey on input/output approaches," *Automatica*, vol. 26, pp. 651-677, 1990.
- [96] R. K. Pearson and M. Pottmann, "Gray-box identification of block-oriented nonlinear models," *Journal of Process Control*, vol. 10, pp. 301-315, 2000.
- [97] R. R. Mohler and W. J. Kolodziej, "An overview of bilinear system theory and applications," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 10, pp. 683-688, 1980.
- [98] A. Balakrishnan, "Modelling and identification theory- A flight control application," *Theory and applications of variable structure systems*, pp. 1-24, 1972.
- [99] C. Bruni, G. KOCH, and G. DI PILLO, "Mathematical models and identification of bilinear systems," *Theory and applications of variable structure systems*, pp. 137-152, 1972.
- [100] R. Baheti, R. Mohler, and H. Spang III, "Second-order correlation method for bilinear system identification," *Automatic Control, IEEE Transactions on*, vol. 25, pp. 1141-1146, 1980.
- [101] J. H. Seinfeld, "Nonlinear estimation theory," *Industrial & Engineering Chemistry*, vol. 62, pp. 32-42, 1970.
- [102] D. Garg and J. Boziuk, "Parameter estimation of non-linear dynamical systems†," *International Journal of Control*, vol. 15, pp. 1121-1127, 1972.
- [103] A. Ivakhnenko, "Heuristic self-organization in problems of engineering cybernetics," *Automatica*, vol. 6, pp. 207-219, 1970.
- [104] S. Derksen and H. Keselman, "Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables," *British Journal of Mathematical and Statistical Psychology*, 1992.
- [105] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, pp. 1873-1896, 1989.

- [106] G. J. Gray, D. J. Murray-Smith, Y. Li, K. C. Sharman, and T. Weinbrenner, "Nonlinear model structure identification using genetic programming," *Control Engineering Practice*, vol. 6, pp. 1341-1352, 1998.
- [107] Y. Yang, C. Wang, and C. Soh, "Force identification of dynamic systems using genetic programming," *International journal for numerical methods in engineering*, vol. 63, pp. 1288-1312, 2005.
- [108] J. Flores and M. Graff, "System identification using genetic programming and gene expression programming," *Computer and Information Sciences-ISCIS 2005*, pp. 503-511, 2005.
- [109] S. Winkler, M. Affenzeller, and S. Wagner, "New methods for the identification of nonlinear model structures based upon genetic programming techniques," *SYSTEMS SCIENCE-WROCLAW*, vol. 31, p. 5, 2005.
- [110] R. Poli, W. B. Langdon, and N. F. McPhee, *A field guide to genetic programming*: Lulu Enterprises UK Ltd, 2008.
- [111] C. Gölo lu and Y. Arslan, "Zigzag machining surface roughness modelling using evolutionary approach," *Journal of Intelligent Manufacturing*, vol. 20, pp. 203-210, 2009.
- [112] C. GÖLO LU and Y. ARSLAN, "GENET K PROGRAMLAMA LE MALAT Ç N YÜZEY PÜRÜZLÜLÜK MODEL GEL T R LMES."
- [113] M. Kovacic and M. Brezocnik, "Genetic programming approach for surface quality prediction," *Tehnicki Vjesnik*, vol. 10, pp. 19-24, 2003.
- [114] M. Kovacic, J. Balic, and M. Brezocnik, "Evolutionary approach for cutting forces prediction in milling," *Journal of materials processing technology*, vol. 155, pp. 1647-1652, 2004.
- [115] A. K. Sood, A. Equbal, V. Toppo, R. K. Ohdar, and S. S. Mahapatra, "An investigation on sliding wear of FDM built parts," *CIRP Journal of Manufacturing Science and Technology*, vol. 5, pp. 48-54, // 2012.
- [116] A. K. Sood, R. K. Ohdar, and S. S. Mahapatra, "Experimental investigation and empirical modelling of FDM process for compressive strength improvement," *Journal of Advanced Research*, vol. 3, pp. 81-90, 2012.
- [117] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning* vol. 2: Springer, 2009.
- [118] F. Yan, K. Mikolajczyk, J. Kittler, and M. Tahir, "A Comparison of L₁ Norm and L₂ Norm Multiple Kernel SVMs in Image and Video Classification," in *Content-Based Multimedia Indexing, 2009. CBMI'09. Seventh International Workshop on*, 2009, pp. 7-12.
- [119] V. N. Vapnik, "Statistical learning theory," 1998.
- [120] L. Vanneschi, M. Castelli, and S. Silva, "Measuring bloat, overfitting and functional complexity in genetic programming," presented at the Proceedings of the 12th annual conference on Genetic and evolutionary computation, Portland, Oregon, USA, 2010.
- [121] J. Rowland, "Model selection methodology in supervised learning with evolutionary computation," *Biosystems*, vol. 72, pp. 187-196, 2003.
- [122] M. Kotanchek, G. Smits, and E. Vladislavleva, "Trustable symbolic regression models: using ensembles, interval arithmetic and pareto fronts to develop robust and trust-aware models," *Genetic Programming Theory and Practice V*, pp. 201-220, 2008.
- [123] C. E. Borges, C. L. Alonso, and J. L. Montaña, "Model selection in genetic programming," 2010, pp. 985-986.

- [124] K. Y. Chan, C. K. Kwong, T. S. Dillon, and Y. C. Tsim, "Reducing overfitting in manufacturing process modeling using a backward elimination based genetic programming," *Applied Soft Computing*, vol. 11, pp. 1648-1656, 3// 2011.
- [125] I. Kushchu, "Genetic programming and evolutionary generalization," *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 431-442, 2002.
- [126] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," 1995, pp. 1137-1145.
- [127] M. Daszykowski, B. Walczak, and D. Massart, "Representative subset selection," *Analytica chimica acta*, vol. 468, pp. 91-103, 2002.
- [128] A. Garg and K. Tai, "A hybrid genetic programming-artificial neural network approach for modeling of vibratory finishing process," *International Proceedings of Computer Science and Information Technology (IPCSIT)*, vol. 18, pp. 14-19, 2011.
- [129] A. Saptoro, M. O. Tadé, and H. Vuthaluru, "A Modified Kennard-Stone Algorithm for Optimal Division of Data for Developing Artificial Neural Network Models," *Chemical Product and Process Modeling*, vol. 7, p. 13, 2012.
- [130] K. Roy, "On some aspects of validation of predictive quantitative structure-activity relationship models," 2007.
- [131] R. May, H. R. Maier, and G. C. Dandy, "Data splitting for artificial neural networks using SOM-based stratified sampling," *Neural Networks*, vol. 23, pp. 283-294, 2010.
- [132] R. W. Kennard and L. A. Stone, "Computer aided design of experiments," *Technometrics*, vol. 11, pp. 137-148, 1969.
- [133] J. Kang, W. Meng, A. Abraham, and H. Liu, "An adaptive PID neural network for complex nonlinear system control," *Neurocomputing*, vol. 135, pp. 79-85, 2014.
- [134] S. Sulaiman, S. Mariyam Shamsuddin, A. Abraham, and S. Sulaiman, "Intelligent web caching using machine learning methods," *Neural Network World*, vol. 21, p. 429, 2011.
- [135] O. Hassanein, G. Sreenatha, and T. Ray, "Hybrid Neuro-Fuzzy Network Identification for Autonomous Underwater Vehicles," in *Swarm, Evolutionary, and Memetic Computing*, ed: Springer, 2013, pp. 287-297.
- [136] A. Isaacs, V. Puttige, T. Ray, W. Smith, and S. Anavatti, "Development of a memetic algorithm for dynamic multi-objective optimization and its applications for online neural network modeling of uavs," in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, 2008, pp. 548-554.
- [137] O. Hassanein, G. Sreenatha, and T. Ray, "Improved Fuzzy Neural Modeling for Underwater Vehicles," *Int. J. World Academy of Science, Engineering and Technology*, vol. 71, pp. 1208-1215, 2012.
- [138] V. Puttige, S. Anavatti, and T. Ray, "Comparative analysis of multiple neural networks for online identification of a UAV," in *AI 2007: Advances in Artificial Intelligence*, ed: Springer, 2007, pp. 120-129.
- [139] T. Ray, "A Neural-Network-Assisted Optimization Framework and Its Use for Optimum-Parameter Identification," *Artificial Neural Networks in Finance and Manufacturing*, p. 221, 2006.

- [140] K. Liew, T. Ray, H. Tan, and M. Tan, "Evolutionary optimization and use of neural network for optimum stamping process design for minimum springback," *Journal of Computing and Information Science in Engineering*, vol. 2, pp. 38-44, 2002.
- [141] E. Byvatov and G. Schneider, "Support vector machine applications in bioinformatics," *Applied bioinformatics*, vol. 2, p. 67, 2003.
- [142] M. A. Hearst, S. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, pp. 18-28, 1998.
- [143] V. Kecman, *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*: MIT press, 2001.
- [144] L.-X. Wang, "Stable adaptive fuzzy control of nonlinear systems," *Fuzzy Systems, IEEE Transactions on*, vol. 1, pp. 146-155, 1993.
- [145] T. Takagi and M. Sugeno, "Fuzzy identification of system and its applications to modelling and control," *IEEE Trans. Syst., Man, and Cyber*, vol. 1, p. 5, 1985.
- [146] E. Buyukbingol, A. Sisman, M. Akyildiz, F. N. Alparslan, and A. Adejare, "Adaptive neuro-fuzzy inference system (ANFIS): a new approach to predictive modeling in QSAR applications: a study of neuro-fuzzy modeling of PCP-based NMDA receptor antagonists," *Bioorganic & medicinal chemistry*, vol. 15, pp. 4265-4282, 2007.
- [147] J. R. Koza, *On the programming of computers by means of natural selection* vol. 1: MIT press, 1996.
- [148] M. Hinchliffe, H. Hiden, B. McKay, M. Willis, M. Tham, and G. Barton, "Modelling chemical process systems using a multi-gene genetic programming algorithm," 1996, pp. 28-31.
- [149] Y. Chen, R. Shi, S. Shu, and W. Gao, "Ensemble and enhanced PM< sub>10</sub> concentration forecast model based on stepwise regression and wavelet analysis," *Atmospheric Environment*, vol. 74, pp. 346-359, 2013.
- [150] E. Bagci and B. Işık, "Investigation of surface roughness in turning unidirectional GFRP composites by using RS methodology and ANN," *The International Journal of Advanced Manufacturing Technology*, vol. 31, pp. 10-17, 2006.
- [151] A. C. Basheer, U. A. Dabade, S. S. Joshi, V. Bhanuprasad, and V. Gadre, "Modeling of surface roughness in precision machining of metal matrix composites using ANN," *Journal of Materials Processing Technology*, vol. 197, pp. 439-444, 2008.
- [152] P. Benardos and G.-C. Vosniakos, "Predicting surface roughness in machining: a review," *International Journal of Machine Tools and Manufacture*, vol. 43, pp. 833-844, 2003.
- [153] U. Çaydaş, A. Haşçalk, O. Buytoz, and A. Meyveci, "Performance evaluation of different twist drills in dry drilling of AISI 304 austenitic stainless steel," *Materials and Manufacturing Processes*, vol. 26, pp. 951-960, // 2011.
- [154] J. P. Davim, V. Gaitonde, and S. Karnik, "Investigations into the effect of cutting conditions on surface roughness in turning of free machining steel by ANN models," *Journal of Materials Processing Technology*, vol. 205, pp. 16-23, 2008.

- [155] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, pp. 121-167, 1998.
- [156] W. Grzesik and S. Brol, "Hybrid approach to surface roughness evaluation in multistage machining processes," *Journal of materials processing technology*, vol. 134, pp. 265-272, 2003.
- [157] E. Ezugwu, D. Fadare, J. Bonney, R. Da Silva, and W. Sales, "Modelling the correlation between cutting and process parameters in high-speed machining of Inconel 718 alloy using an artificial neural network," *International Journal of Machine Tools and Manufacture*, vol. 45, pp. 1375-1385, 2005.
- [158] Y. T. Ic, M. Yurdakul, and E. Eraslan, "Development of a component-based machining centre selection model using AHP," *International journal of production research*, vol. 50, pp. 6489-6498, 2012.
- [159] X. Chu, C. Tang, D. Xue, Y. Su, and X. Sun, "Identification of machining features based on available resources of cutting tools," *International journal of production research*, vol. 50, pp. 4141-4157, 2012.
- [160] X. Sun, X. Chu, D. Xue, Y. Su, and C. Tang, "An integrated setup/fixture planning approach for machining prismatic parts," *International journal of production research*, vol. 50, pp. 1009-1027, 2012.
- [161] P. Suresh, P. Venkateswara Rao, and S. Deshmukh, "A genetic algorithmic approach for optimization of surface roughness prediction model," *International Journal of Machine Tools and Manufacture*, vol. 42, pp. 675-680, 2002.
- [162] D. Shi and N. N. Gindy, "Tool wear predictive model based on least squares support vector machines," *Mechanical Systems and Signal Processing*, vol. 21, pp. 1799-1814, 2007.
- [163] V. S. Sharma, S. K. Sharma, and A. K. Sharma, "Cutting tool wear estimation for turning," *Journal of Intelligent Manufacturing*, vol. 19, pp. 99-108, 2008.
- [164] S. S. Rangwala and D. A. Dornfeld, "Learning and optimization of machining operations using computing abilities of neural networks," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, pp. 299-314, 1989.
- [165] T. Rajasekaran, B. Vinayagam, K. Palanikumar, and S. Prakash, "Influence of machining parameters on surface roughness and material removal rate in machining carbon fiber reinforced polymer material," in *Frontiers in Automobile and Mechanical Engineering (FAME)*, 2010, 2010, pp. 75-80.
- [166] T. Rajasekaran, K. Palanikumar, and B. Vinayagam, "Application of fuzzy logic for modeling surface roughness in turning CFRP composites using CBN tool," *Production Engineering*, vol. 5, pp. 191-199, 2011.
- [167] I. Korkut, M. Kasap, I. Ciftci, and U. Seker, "Determination of optimum cutting parameters during machining of AISI 304 austenitic stainless steel," *Materials & Design*, vol. 25, pp. 303-305, 2004.
- [168] A. Kohli and U. Dixit, "A neural-network-based methodology for the prediction of surface roughness in a turning process," *The International Journal of Advanced Manufacturing Technology*, vol. 25, pp. 118-129, 2005.
- [169] R. Azouzi and M. Guillot, "On-line prediction of surface finish and dimensional deviation in turning using neural network based sensor

- fusion," *International Journal of Machine Tools and Manufacture*, vol. 37, pp. 1201-1217, 1997.
- [170] Y. Hadi and S. G. Ahmed, "Assessment of Surface Roughness Model for Turning Process," in *Knowledge Enterprise: Intelligent Strategies in Product Design, Manufacturing, and Management*, ed: Springer, 2006, pp. 152-158.
 - [171] S. Yaldiz, F. Unsacar, and H. Saglam, "Comparison of experimental results obtained by designed dynamometer to fuzzy model for predicting cutting forces in turning," *Materials & Design*, vol. 27, pp. 1139-1147, // 2006.
 - [172] S. K. Pal and D. Chakraborty, "Surface roughness prediction in turning using artificial neural network," *Neural Computing & Applications*, vol. 14, pp. 319-324, 2005.
 - [173] J. Y. Zhang, S. Y. Liang, G. Zhang, and D. Yen, "Modeling of residual stress profile in finish hard turning," *Materials and manufacturing processes*, vol. 21, pp. 39-45, 2006.
 - [174] S. Kaewkuekool, K. Jirapattarasilp, and K. Pechkong, "A Study of Influence Factors Affecting to Surface Roughness in Stainless Steel Turning," in *Computer Engineering and Technology, 2009. ICCET'09. International Conference on*, 2009, pp. 299-302.
 - [175] N. Tosun and L. Özler, "A study of tool life in hot machining using artificial neural networks and regression analysis method," *Journal of materials processing technology*, vol. 124, pp. 99-104, 2002.
 - [176] R. Dutta, S. Paul, and A. Chattopadhyay, "Applicability of the modified back-propagation algorithm in tool condition monitoring for faster convergence," *Journal of materials processing technology*, vol. 98, pp. 299-309, 2000.
 - [177] E. Ezugwu, S. Arthur, and E. Hines, "Tool-wear prediction using artificial neural networks," *Journal of materials processing technology*, vol. 49, pp. 255-264, 1995.
 - [178] Y. Qian, J. Tian, L. Liu, Y. Zhang, and Y. Chen, "A tool wear predictive model based on SVM," in *Control and Decision Conference (CCDC), 2010 Chinese*, 2010, pp. 1213-1217.
 - [179] A. K. Nandi and D. K. Pratihari, "Design of a genetic-fuzzy system to predict surface finish and power requirement in grinding," *Fuzzy sets and systems*, vol. 148, pp. 487-504, 2004.
 - [180] S. S. Gill, R. Singh, J. Singh, and H. Singh, "Adaptive neuro-fuzzy inference system modeling of cryogenically treated AISI M2 HSS turning tool for estimation of flank wear," *Expert Systems with Applications*, vol. 39, pp. 4171-4180, 2012.
 - [181] A. Gajate, R. Haber, R. del Toro, P. Vega, and A. Bustillo, "Tool wear monitoring using neuro-fuzzy techniques: a comparative study in a turning process," *Journal of intelligent Manufacturing*, vol. 23, pp. 869-882, 2012.
 - [182] X. Fang and I. Jawahir, "Predicting total machining performance in finish turning using integrated fuzzy-set models of the machinability parameters," *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, vol. 32, pp. 833-849, 1994.
 - [183] T.-S. Lan, "Fuzzy parametric deduction for material removal rate optimization," *Journal of mathematics and statistics*, vol. 7, pp. 51-56, 2011.

- [184] D. Thakur, B. Ramamoorthy, and L. Vijayaraghavan, "An experimental analysis of effective high speed turning of superalloy Inconel 718," *Journal of materials science*, vol. 44, pp. 3296-3304, 2009.
- [185] M. Milfelner, U. Zuperl, and F. Cus, "Generation of a Model for Cutting Forces Using Artificial Intelligence," *Strojniski Vestnik*, vol. 51, pp. 41-54, 2005.
- [186] M. Milfelner, J. Kopac, F. Cus, and U. Zuperl, "Genetic equation for the cutting force in ball-end milling," *Journal of materials processing technology*, vol. 164, pp. 1554-1560, 2005.
- [187] D. Kondayya and A. G. Krishna, "An integrated evolutionary approach for modelling and optimization of wire electrical discharge machining," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 225, p. 549, 2011.
- [188] M. Kk, E. Kanca, and O. Eyerciglu, "Prediction of surface roughness in abrasive waterjet machining of particle reinforced MMCs using genetic expression programming," *International Journal of Advanced Manufacturing Technology*, vol. 55, pp. 955-968, 2011.
- [189] H. Aouici, M. A. Yallese, B. Fnides, K. Chaoui, and T. Mabrouki, "Modeling and optimization of hard turning of X38CrMoV5-1 steel with CBN tool: Machining parameters effects on flank wear and surface roughness," *Journal of mechanical science and technology*, vol. 25, pp. 2843-2851, 2011.
- [190] K. Pelckmans, J. A. K. Suykens, T. Van Gestel, J. De Brabanter, L. Lukas, B. Hamers, *et al.*, "LS-SVMLab: a matlab/c toolbox for least squares support vector machines," *Tutorial. KULeuven-ESAT. Leuven, Belgium*, 2002.
- [191] A. H. Gandomi, A. H. Alavi, M. Mousavi, and S. M. Tabatabaei, "A hybrid computational approach to derive new ground-motion prediction equations," *Engineering Applications of Artificial Intelligence*, vol. 24, pp. 717-732, 6// 2011.
- [192] A. Gandomi, G. Yun, and A. Alavi, "An evolutionary approach for modeling of shear strength of RC deep beams," *Materials and Structures*, vol. 46, pp. 2109-2119, 2013/12/01 2013.
- [193] A. Armas, O. Bettin, I. Alvarez-Armas, and G. Rubiolo, "Strain aging effects on the cyclic behavior of austenitic stainless steels," *Journal of Nuclear Materials*, vol. 155, pp. 644-649, 1988.
- [194] S.-G. Hong and S.-B. Lee, "Dynamic strain aging under tensile and LCF loading conditions, and their comparison in cold worked 316L stainless steel," *Journal of nuclear materials*, vol. 328, pp. 232-242, 2004.
- [195] X. Wang and D. Li, "Mechanical, electrochemical and tribological properties of nano-crystalline surface of 304 stainless steel," *Wear*, vol. 255, pp. 836-845, 2003.
- [196] A. Armas, S. Here, I. Alvarez-Armas, S. Degallaix, A. Cond, and F. Lovey, "The influence of temperature on the cyclic behavior of aged and unaged super duplex stainless steels," *Materials Science and Engineering: A*, vol. 491, pp. 434-439, 2008.
- [197] A. K. Gupta, H. N. Krishnamurthy, Y. Singh, K. M. Prasad, and S. K. Singh, "Development of constitutive models for dynamic strain aging regime in Austenitic stainless steel 304," *Materials & Design*, 2012.

- [198] K. Peng, K. Qian, and W. Chen, "Effect of dynamic strain aging on high temperature properties of austenitic stainless steel," *Materials Science and Engineering: A*, vol. 379, pp. 372-377, 2004.
- [199] L. Llanes, A. Mateo, L. Iturgoyen, and M. Anglada, "Aging effects on the cyclic deformation mechanisms of a duplex stainless steel," *Acta materialia*, vol. 44, pp. 3967-3978, 1996.
- [200] Y. Lin and X.-M. Chen, "A critical review of experimental results and constitutive descriptions for metals and alloys in hot working," *Materials & Design*, vol. 32, pp. 1733-1759, 2011.
- [201] G. R. Johnson and W. H. Cook, "A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures," in *Proceedings of the 7th International Symposium on Ballistics*, 1983, pp. 541-547.
- [202] D. Samantaray, S. Mandal, and A. Bhaduri, "A comparative study on Johnson Cook, modified Zerilli–Armstrong and Arrhenius-type constitutive models to predict elevated temperature flow behaviour in modified 9Cr–1Mo steel," *Computational Materials Science*, vol. 47, pp. 568-576, 2009.
- [203] D. Samantaray, S. Mandal, and A. Bhaduri, "A critical comparison of various data processing methods in simple uni-axial compression testing," *Materials & Design*, vol. 32, pp. 2797-2802, 2011.
- [204] D. Samantaray, S. Mandal, U. Borah, A. Bhaduri, and P. Sivaprasad, "A thermo-viscoplastic constitutive model to predict elevated-temperature flow behaviour in a titanium-modified austenitic stainless steel," *Materials Science and Engineering: A*, vol. 526, pp. 1-6, 2009.
- [205] C. Zener and J. Hollomon, "Effect of strain rate upon plastic flow of steel," *Journal of Applied Physics*, vol. 15, pp. 22-32, 1944.
- [206] F. J. Zerilli and R. W. Armstrong, "Dislocation-mechanics-based constitutive relations for material dynamics calculations," *Journal of Applied Physics*, vol. 61, pp. 1816-1825, 1987.
- [207] Y.-H. Xiao and C. Guo, "Constitutive modelling for high temperature behavior of 1Cr12Ni3Mo2VNbN martensitic steel," *Materials Science and Engineering: A*, vol. 528, pp. 5081-5087, 2011.
- [208] S.-T. Chiou, W.-C. Cheng, and W.-S. Lee, "Strain rate effects on the mechanical properties of a Fe–Mn–Al alloy under dynamic impact deformations," *Materials Science and Engineering: A*, vol. 392, pp. 156-162, 2005.
- [209] X. He, Z. Yu, and X. Lai, "A method to predict flow stress considering dynamic recrystallization during hot deformation," *Computational Materials Science*, vol. 44, pp. 760-764, 2008.
- [210] A. Garg, A. Garg, and K. Tai, "A multi-gene genetic programming model for estimating stress-dependent soil water retention curves," *Computational Geosciences*, pp. 1-12, 2014.
- [211] A. H. Gandomi and A. H. Alavi, "Multi-stage genetic programming: a new strategy to nonlinear system modeling," *Information Sciences*, vol. 181, pp. 5227-5239, 2011.
- [212] R. García, E. Verdú, L. M. Regueras, J. P. De Castro, and M. J. Verdú, "A neural network based intelligent system for tile prefetching in web map services," *Expert Systems with Applications*, vol. 40, pp. 4096-4105, // 2013.

- [213] G. K. Smyth, "Polynomial approximation," *Encyclopedia of Biostatistics*, 1998.
- [214] P. Samui and D. Kothari, "Application of multivariate adaptive regression splines to evaporation losses in reservoirs," *Earth Sci. India*, vol. 4, pp. 15-20, 2011.
- [215] J. H. Friedman, "Multivariate adaptive regression splines," *The annals of statistics*, pp. 1-67, 1991.
- [216] S. Milborrow, "Earth: multivariate adaptive regression spline models," *R Software Package* (<http://cran.r-project.org/web/packages/earth/index.html>), 2009.
- [217] I. Jawahir, A. Balaji, K. Rouch, and J. Baker, "Towards integration of hybrid models for optimized machining performance in intelligent manufacturing systems," *Journal of materials processing technology*, vol. 139, pp. 488-498, 2003.
- [218] K. Patra, S. K. Pal, and K. Bhattacharyya, "Artificial neural network based prediction of drill flank wear from motor current signals," *Applied Soft Computing*, vol. 7, pp. 929-935, 2007.
- [219] A. K. Nandi and J. Paulo Davim, "A study of drilling performances with minimum quantity of lubricant using fuzzy logic rules," *Mechatronics*, vol. 19, pp. 218-232, 2009.
- [220] S. Karnik and V. Gaitonde, "Development of artificial neural network models to study the effect of process parameters on burr size in drilling," *The International Journal of Advanced Manufacturing Technology*, vol. 39, pp. 439-453, 2008.
- [221] U. Çaydaş and A. Haşçalık, "A study on surface roughness in abrasive waterjet machining process using artificial neural networks and regression analysis method," *Journal of materials processing technology*, vol. 202, pp. 574-582, 2008.
- [222] S. Kumar, S. R. Chauhan, P. K. Rakesh, I. Singh, and J. P. Davim, "Drilling of glass fiber/vinyl ester composites with fillers," *Materials and Manufacturing Processes*, vol. 27, pp. 314-319, // 2012.
- [223] L. Ozler and N. Dogru, "An experimental investigation of hole geometry in friction drilling," *Materials and Manufacturing Processes*, vol. 28, pp. 470-475, // 2013.
- [224] T. Rajmohan and K. Palanikumar, "Experimental investigation and analysis of thrust force in drilling hybrid metal matrix composites by coated carbide drills," *Materials and Manufacturing Processes*, vol. 26, pp. 961-968, // 2011.
- [225] A. M. Zain, H. Haron, and S. Sharif, "Review of ANN Technique for Modeling Surface Roughness Performance Measure in Machining Process," 2009, pp. 35-39.
- [226] K. S. Park and S. H. Kim, "Artificial intelligence approaches to determination of CNC machining parameters in manufacturing: a review," *Artificial intelligence in engineering*, vol. 12, pp. 127-134, 1998.
- [227] A. Aggarwal and H. Singh, "Optimization of machining techniques—a retrospective and literature review," *Sadhana*, vol. 30, pp. 699-711, 2005.
- [228] D. A. Davidson, "Vibratory finishing: Versatile, effective, and reliable," *Metal Finishing*, vol. 106, 2008.

- [229] J. Domblesky, R. Evans, and V. Cariapa, "Material removal model for vibratory finishing," *International journal of production research*, vol. 42, pp. 1029-1041, 2004.
- [230] S. Wang, R. Timsit, and J. Spelt, "Experimental investigation of vibratory finishing of aluminum," *Wear*, vol. 243, pp. 147-156, 2000.
- [231] D. A. Davidson, "MECHANICAL SURFACE PREPARATION-Mass Finishing Processes," *Metal Finishing*, vol. 1, pp. 75-87, 2006.
- [232] P. Brust, "Surface Improvement by Vibratory Cascade Finishing Process," *TRANSACTIONS-NORTH AMERICAN MANUFACTURING RESEARCH INSTITUTION OF SME*, pp. 93-98, 1997.
- [233] A. Sofronas and S. Taraman, "Model development and optimization of vibratory finishing process," *International journal of production research*, vol. 17, pp. 23-31, 1979.
- [234] S. Pande and S. Patel, "Investigations on vibratory burnishing process," *International Journal of Machine Tool Design and Research*, vol. 24, pp. 195-206, 1984.
- [235] S. Wanga, R. S. Timsit, and J. K. Spelt, "Experimental investigation of vibratory finishing of aluminum," *Wear*, vol. 243, pp. 147-156, // 2000.
- [236] K. Ogino, A. Hida, S. Kishima, and S. Kumanomido, "Susceptibility of type 431 stainless steel to erosion-corrosion by vibratory cavitation in corrosive media," *Wear*, vol. 116, pp. 299-307, // 1987.
- [237] A. Mohajerani and J. K. Spelt, "Numerical modeling of the edge rounding of brittle materials by vibratory finishing," *Wear*, vol. 268, pp. 1002-1012, // 2010.
- [238] D. Ciampini, M. Papini, and J. K. Spelt, "Characterization of vibratory finishing using the Almen system," *Wear*, vol. 264, pp. 671-678, // 2008.
- [239] A. Bernard and A. Fischer, "New trends in rapid product development," *CIRP Annals-Manufacturing Technology*, vol. 51, pp. 635-652, 2002.
- [240] N. Hopkinson, R. Hague, and P. Dickens, *Rapid manufacturing*: Wiley Online Library, 2006.
- [241] S. Mansour and R. Hague, "Impact of rapid manufacturing on design for manufacture for injection moulding," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 217, p. 453, 2003.
- [242] S. Upcraft and R. Fletcher, "The rapid prototyping technologies," *Assembly Automation*, vol. 23, pp. 318-330, 2003.
- [243] B. Wiedemann and H. A. Jantzen, "Strategies and applications for rapid product and process development in Daimler-Benz AG," *Computers in Industry*, vol. 39, pp. 11-25, 1999.
- [244] X. Yan and P. Gu, "A review of rapid prototyping technologies and systems," *Computer-Aided Design*, vol. 28, pp. 307-318, 1996.
- [245] D. Pham and R. Gault, "A comparison of rapid prototyping technologies," *International Journal of Machine Tools and Manufacture*, vol. 38, pp. 1257-1287, 1998.
- [246] D. Ahn, J.-H. Kweon, S. Kwon, J. Song, and S. Lee, "Representation of surface roughness in fused deposition modeling," *Journal of Materials Processing Technology*, vol. 209, pp. 5593-5600, 2009.
- [247] K. Thrimurthulu, P. M. Pandey, and N. Venkata Reddy, "Optimum part deposition orientation in fused deposition modeling," *International Journal of Machine Tools and Manufacture*, vol. 44, pp. 585-594, 2004.

- [248] L. Aijun, Z. Zhuohui, W. Daoming, and Y. Jinyong, "Optimization method to fabrication orientation of parts in fused deposition modeling rapid prototyping," in *2010 International Conference on Mechanic Automation and Control Engineering*, 2010, pp. 416-419.
- [249] H.-S. Byun and K. H. Lee, "Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making," *Robotics and Computer-Integrated Manufacturing*, vol. 22, pp. 69-80, 2006.
- [250] D.-Y. Chang and B.-H. Huang, "Studies on profile error and extruding aperture for the RP parts using the fused deposition modeling process," *The International Journal of Advanced Manufacturing Technology*, vol. 53, pp. 1027-1037, 2011.
- [251] W. Wang, C. Cheah, J. Fuh, and L. Lu, "Influence of process parameters on stereolithography part shrinkage," *Materials & Design*, vol. 17, pp. 205-213, 1996.
- [252] G. Casalino, L. De Filippis, A. Ludovico, and L. Tricarico, "An investigation of rapid prototyping of sand casting molds by selective laser sintering," *Journal of Laser Applications*, vol. 14, pp. 100-106, 2002.
- [253] P. M. Pandey, N. Venkata Reddy, and S. G. Dhande, "Improvement of surface finish by staircase machining in fused deposition modeling," *Journal of materials processing technology*, vol. 132, pp. 323-331, 2003.
- [254] M. Mahesh, J. Y. Fuh, Y. Wong, and H. T. Loh, "Benchmarking for decision making in rapid prototyping systems," in *Automation Science and Engineering, 2005. IEEE International Conference on*, 2005, pp. 19-24.
- [255] A. Chatterjee, S. Kumar, P. Saha, P. Mishra, and A. R. Choudhury, "An experimental design approach to selective laser sintering of low carbon steel," *Journal of Materials Processing Technology*, vol. 136, pp. 151-157, 2003.
- [256] T. Reddy, Y. R. Kumar, and C. Rao, "Determination of optimum process parameters using Taguchi's approach to improve the quality of SLS parts," in *Proceedings of the 17th IASTED international conference on Modelling and simulation*, 2006, pp. 228-233.
- [257] G. Vosniakos, T. Maroulis, and D. Pantelis, "A method for optimizing process parameters in layer-based rapid prototyping," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 221, pp. 1329-1340, 2007.
- [258] P. M. Dixit and U. S. Dixit, *Modeling of metal forming and machining processes: by finite element and soft computing methods*: Springer, 2008.
- [259] G. P. Kumar and S. P. Regalla, "Optimization of support material and build time in fused deposition modeling (FDM)," *Applied Mechanics and Materials*, vol. 110, pp. 2245-2251, 2012.
- [260] A. K. Sood, R. Ohdar, and S. Mahapatra, "A hybrid ANN-BFOA approach for optimization of FDM process parameters," in *Swarm, Evolutionary, and Memetic Computing*, ed: Springer, 2010, pp. 396-403.
- [261] A. Sood, R. Ohdar, and S. Mahapatra, "Parametric appraisal of fused deposition modelling process using the grey Taguchi method," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 224, pp. 135-145, 2010.

- [262] A. K. Sood, R. Ohdar, and S. Mahapatra, "Improving dimensional accuracy of Fused Deposition Modelling processed part using grey Taguchi method," *Materials & Design*, vol. 30, pp. 4243-4252, 2009.
- [263] B. Duan, W. L. Cheung, and M. Wang, "Optimized fabrication of Ca-P/PHBV nanocomposite scaffolds via selective laser sintering for bone tissue engineering," *Biofabrication*, vol. 3, p. 015001, 2011.
- [264] R. Quintana, J.-W. Choi, K. Puebla, and R. Wicker, "Effects of build orientation on tensile strength for stereolithography-manufactured ASTM D-638 type I specimens," *The International Journal of Advanced Manufacturing Technology*, vol. 46, pp. 201-215, 2010.
- [265] J.-W. Choi, R. Quintana, and R. B. Wicker, "Fabrication and characterization of embedded horizontal micro-channels using line-scan stereolithography," *Rapid Prototyping Journal*, vol. 17, pp. 351-361, 2011.
- [266] R. Anitha, S. Arunachalam, and P. Radhakrishnan, "Critical parameters influencing the quality of prototypes in fused deposition modelling," *Journal of Materials Processing Technology*, vol. 118, pp. 385-388, 2001.
- [267] J. P. Kruth and S. Kumar, "Statistical analysis of experimental parameters in selective laser sintering," *Advanced Engineering Materials*, vol. 7, pp. 750-755, 2005.
- [268] J. Laeng, Z. A. Khan, and S. Khu, "Optimizing flexible behaviour of bow prototype using Taguchi approach," *Journal of Applied Sciences*, vol. 6, pp. 622-630, 2006.
- [269] M. Monzón, P. M. Hernández, A. N. Benítez, M. D. Marrero, and Á. Fernández, "Predictability of plastic parts behaviour made from rapid manufacturing," *Tsinghua Science & Technology*, vol. 14, pp. 100-107, 2009.
- [270] S. Campanelli, G. Cardano, R. Giannoccaro, A. Ludovico, and E. L. Bohez, "Statistical analysis of the stereolithographic process to improve the accuracy," *Computer-Aided Design*, vol. 39, pp. 80-86, 2007.
- [271] R. Cheng, X. Wu, and J. Zheng, "Improving dimensional accuracy of SLS processed part using Taguchi method," in *Fourth International Seminar on Modern Cutting and Measuring Engineering*, 2010, pp. 799715-799715-5.
- [272] W.-Y. Yeong, C.-K. Chua, K.-F. Leong, and M. Chandrasekaran, "Rapid prototyping in tissue engineering: challenges and potential," *TRENDS in Biotechnology*, vol. 22, pp. 643-652, 2004.
- [273] L. Di Angelo and P. Di Stefano, "A neural network-based build time estimator for layer manufactured objects," *The International Journal of Advanced Manufacturing Technology*, vol. 57, pp. 215-224, 2011.
- [274] S. Lee, W. Park, H. Cho, W. Zhang, and M. Leu, "A neural network approach to the modelling and analysis of stereolithography processes," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 215, pp. 1719-1733, 2001.
- [275] J. Munguía, J. Ciurana, and C. Riba, "Neural-network-based model for build-time estimation in selective laser sintering," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 223, pp. 995-1003, 2009.
- [276] X. Shen, J. Yao, Y. Wang, and J. Yang, "Density prediction of selective laser sintering parts based on artificial neural network," in *Advances in neural networks-ISBN 2004*, ed: Springer, 2004, pp. 832-840.

- [277] A. Equbal, A. K. Sood, and S. Mahapatra, "Prediction of dimensional accuracy in fused deposition modelling: a fuzzy logic approach," *International Journal of Productivity and Quality Management*, vol. 7, pp. 22-43, 2011.
- [278] S. Neşeli, S. Yaldız, and E. Türkeş, "Optimization of tool geometry parameters for turning operations based on the response surface methodology," *Measurement*, vol. 44, pp. 580-587, 2011.
- [279] D. P. Solomatine and Y. Xue, "M5 model trees and neural networks: application to flood forecasting in the upper reach of the Huai River in China," *Journal of Hydrologic Engineering*, vol. 9, p. 491, 2004.
- [280] D. P. Solomatine and M. Siek, "Flexible and optimal M5 model trees with applications to flow predictions," in *Proc. 6th Int. Conf. on Hydroinformatics. World Scientific, Singapore*, 2004.
- [281] A. Etemad-Shahidi and J. Mahjoobi, "Comparison between M5' model tree and neural networks for prediction of significant wave height in Lake Superior," *Ocean Engineering*, vol. 36, pp. 1175-1181, 2009.
- [282] B. Bhattacharya and D. P. Solomatine, "Neural networks and M5 model trees in modelling water level–discharge relationship," *Neurocomputing*, vol. 63, pp. 381-396, 2005.
- [283] N. Erginel, "Modeling and analysis of packing properties through a fuzzy inference system," *Journal of Intelligent Manufacturing*, vol. 21, pp. 869-874, 2010.
- [284] A. Kumagai, T. I. Liu, and P. Hozian, "Control of shape memory alloy actuators with a neuro-fuzzy feedforward model element," *Journal of Intelligent Manufacturing*, vol. 17, pp. 45-56, 2006.
- [285] G. Jekabsons, "M5primelab: M5'regression tree and model tree toolbox for Matlab," *Octave, Institute of Applied Computer Systems*, 2010.
- [286] M. U. s. Guide, "The mathworks," *Inc., Natick, MA*, vol. 5, 1998.
- [287] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of intelligent and Fuzzy systems*, vol. 2, pp. 267-278, 1994.