A COMPARATIVE STUDY ON PREDICTIVE MODELING OF ELECTRICAL DISCHARGE MACHINING PROCESSES

By

KUMARESH DEY

B.Tech (Mechanical Engineering)-2015-19
Jalpaiguri Government Engineering College
Maulana Abul Kalam Azad University of Technology

Examination Roll No.: M4PRD22012

Registration No.: 154480

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF MASTER OF PRODUCTION ENGINEERING IN THE FACULTY OF ENGINEERING AND TECHNOLOGY,
JADAVPUR UNIVERSITY

DEPARTMENT OF PRODUCTION ENGINEERING
JADAVPUR UNIVERSITY
KOLKATA-700032

JADAVPUR UNIVERSITY FACULTY OF ENGINEERING AND TECHNOLOGY

CERTIFICATE OF RECOMMEDATION

I HEREBY RECOMMEND THAT THE THESIS ENTITLED "**A COMPARATIVE STUDY ON PREDICTIVE MODELING OF ELECTRICAL DISCHARGE MACHINING PROCESSES**" CARRIED OUT UNDER MY/OUR GUIDANCE BY **MR. KUMARESH DEY** MAY BE ACCEPTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF "MASTER OF PRODUCTION ENGINEERING"

_____

(Dr. Shankar Chakraborty)
Thesis Advisor
Dept. of Production Engineering
Jadavpur University
Kolkata-700032

_____

HEAD, Dept. of Production Engineering
Jadavpur University
Kolkata-700032

_____

DEAN, Faculty of Engineering and Technology
Jadavpur University
Kolkata-700032

JADAVPUR UNIVERSITY
FACULTY OF ENGINEERING AND TECHNOLOGY

CERTIFICATE OF APPROVAL

The foregoing thesis is hereby approved as a creditable study of an engineering subject carried out and presented in a manner of satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed and conclusion drawn therein but thesis only for the purpose for which it has been submitted.

_____

COMMITTEE ON
FINAL EXAMINATION
FOR EVALUATION OF
THE THESIS

_____

ACKNOWLEDGEMENT

I owe a deep debt of gratitude to my project supervisor Dr. Shankar Chakraborty, Department of Production Engineering. Jadavpur University for his invaluable and untiring guidance, encouragement and supervision, throughout this research work. His effective skill, knowledge and experience have made it possible for me to successfully complete this thesis work within the stipulated time. I am very much indebted to him and express my sincere gratitude to him.

I also take this opportunity to my gratitude to all the faculty members of Production Engineering department for their mental support, immense help and co-operation during the course of this thesis work

I express my heartiest thanks to my friends and classmates for their useful assistance, cooperation and support.

I am also thankful to the librarian and research scholars of Production Engineering Department, Jadavpur University for their cordial assistance.

I thankfully acknowledge the support of all those people, who some time or the other directly or indirectly rendered their help at different stages of this work

Finally, special thanks to my beloved parents, as they always stood by me, caring least the prevalent situation.

_____

Kumaresh Dey
Class Roll No: 002011702013

Table of Contents

1 Introduction:

1.1 An overview of Electric discharge machining process:

Electrical discharge machining (EDM), also known as spark machining, spark eroding, die sinking, wire burning or wire erosion, is a metal fabrication process whereby a desired shape is obtained by using electrical discharges (sparks). EDM is one of the most extensively used non-conventional material removal processes. Its unique feature of using thermal energy to machine electrically conductive parts regardless of hardness has been its distinctive advantage in the manufacture of mould, die, automotive, aerospace and surgical components. In addition, EDM does not make direct contact between the electrode and the workpiece eliminating mechanical stresses, chatter and vibration problems during machining.

The material erosion mechanism primarily makes use of electrical energy and turns it into thermal energy through a series of discrete electrical discharges occurring between the electrode and workpiece immersed in a dielectric fluid. The thermal energy generates a channel of plasma between the cathode and anode at a temperature in the range of 8000 to 12,000 °C or as high as 20,000 °C [1] initializing a substantial amount of heating and melting of material at the surface of each pole. When the pulsating direct current supply occurring at the rate of approximately 20,000–30,000 Hz is turned off, the plasma channel breaks down. This causes a sudden reduction in the temperature allowing the circulating dielectric fluid to implore the plasma channel and flush the molten material from the pole surfaces in the form of microscopic debris. This process of melting and evaporating material from the workpiece surface is in complete contrast to the conventional machining processes, as chips are not mechanically produced. The volume of material removed per discharge is typically in the range of 10^{-6}–10^{-4} mm^3 and the material removal rate (MRR) is usually between 2 and 400 mm^3/min depending on specific application. Since the shaped electrode defines the area in which the spark erosion will occur, the accuracy of the part produced after EDM is fairly high. After all, EDM is a reproductive shaping process in which the form of the electrode is mirrored in the workpiece.

The EDM process is most widely used by the mold-making, tool, and die industries, but is becoming a common method of making prototype and production parts, especially in the aerospace, automobile and electronics industries [2]–[5] in which production quantities are relatively low. A number of EDM variations based on this basic configuration have emerged in the industry to cope with the machining of exotic materials or super hard metal alloys used exclusively in the manufacture of aeronautical and aerospace parts. Wire-cut EDM[6] is one of the most favourable variants owing to its ability to machine conductive, exotic and high strength and temperature resistive materials with the scope of generating intricate shapes and profiles. It uses a thin continuously travelling wire feeding through the workpiece by a micro-processor eliminating the need for elaborate pre-shaped electrodes, which are required in the EDM. Other

more specialised variations include electrical discharge texturing [7]used for the texturing of cold rolled steel and aluminium sheets and electrical discharge grinding (EDG) used for the manufacture of polycrystalline diamond cutting tools[8].

1.2 Need for prediction of response parameters in EDM process:

Non-conventional machining processes continue to occupy a dominant fraction of all manufacturing operations. EDM is one of the most popular machining process among the non-conventional machining processes. New advances in machining technologies, along with advanced material development, all aimed at improved manufacturing productivity, product quality and cost reduction, require predictive performance models for use in process planning systems for machining processes. At the discrete part/product level, this includes selection of: (1) optimal cutting conditions; (2) coolant/lubricant type; (3) cutting tools, (3) optimal power consumption etc. The goal is to satisfy functional design requirements with an optimized process plan. To enable this, predictive performance models need to be developed and integrated with process planning. Development of advanced predictive models has accelerated to accommodate demands. These models can be broadly clustered as analytical, numerical, experimental, Artificial Intelligence (AI) based, and hybrid modelling techniques [9]. However, a dichotomy remains whereby predictive models focus on fundamental physical process variables such as stresses, strains, strain-rates, temperatures, and dynamic tool deflection. Alternatively, applications require prediction of process performance measures such as tool-wear/tool-life, surface integrity, cutting forces/power/torque, part accuracy and process stability.

Machining presents challenging and exciting intellectual problem that have fascinated researchers and practitioners for decades. Prediction of the fundamental physical variables by different methods has continued to make significant progress. However, we cannot ignore that the end goal of machining models is to predict industry-relevant outcomes, and thus improve productivity. Model inputs include cutting parameters, tool geometry, workpiece and tool materials. Models can calculate both intermediary fundamental physical variables or machining performance outcomes. Without successful models, expensive experimental testing will continue to dominate practical process development. Thus, the most successful models, in terms of their adoption by industry, are those that successfully that can compute incorporate modelling of both kind of variables, tool-life modelling is one such example. However, in the current state of-the-art, while input variables appear to remain unchanged, process outcomes can still change drastically. This points to areas where there is still some fundamental lack of understanding, and this emphasizes the need for continued fundamental modelling efforts.

There are various types of prediction techniques through which response parameters of EDM process can be predicted. However, we can not rely on one prediction techniques as different

prediction models works better based on some certain parameters. So in this work comparison of performance of different metamodeling techniques and neural networks has been performed on different response parameters like material removal rate, surface roughness and overcut. Through these models a conclusion can be drawn which model works best on that condition.

1.3 Literature review:

Dang[10] used the Kriging model to explore the highly nonlinear relationship between process factors and machining reactions in electric discharge machining (EDM), including material removal rate (MRR), surface roughness (SR), and electrode wear rate (EWR). The ideal machining conditions are then determined using particle swarm, a new multi-objective optimization technique that not only maximises machining speed but also minimises EWR while keeping the SR limitation. P20 steel was used in the experiment, which was done on a CNC EDM machine with a copper electrode. When the discharge current is increased, the MRR increases dramatically, according to the findings. An intelligent process modelling and optimization of EDM machining is a combination of Kriging regression model and particle swarm optimization.

A data-driven strategy to metamodeling manufacturing/machining processes is explored in the study of Kalita et.al.[11]. A non-predefined form-free technique is proposed as an alternative to the commonly used second-order polynomial regression metamodels. Using genetic programming, the highly adaptive metamodeling method known as symbolic regression is carried out. The training and testing data are taken from a central composite design-based experimental dataset on electric discharge machining. To quantify three different responses, four different process parameters, namely (voltage, pulse on time, pulse off time, and current), are employed as independent parameters (material removal rate, electrode wear rate, and surface roughness). Various statistical metrics such as R^2, MAE, and MSE are used to assess the performance of metamodels. For all of the responses, the metamodels' performance on the training and testing data is judged to be satisfactory.

Ghadai et. al. [12] approached a metamodel coupled with global optimization approach to predict suitable combinations of input parameters (current, pulse on-time and pulse off-time) that would effectively increase the material removal rate and minimize the tool wear. The metamodels are built by using a novel symbolic regression approach carried out using Genetic Programming (GP). On comparative evaluation against traditional RSM metamodels, the GP metamodels show much better and accurate estimation. GP metamodels are then coupled with a genetic algorithm to carry out multi-objective optimization of the EDM process. Using genetic programming (GP) metamodeling approach, MRR and TWR are expressed as functions of current, pulse on-time and pulse off-time. As compared to fixed-form polynomial RSM metamodels the genetically searched

form-free GP metamodels were seen to perform about 2% and 21% better for MRR and TWR respectively.

Ulas et. al.[13] in his study, experimented wire cut EDM (WEDM) on Al7075 aluminum alloy with different parameters (voltage, pulse-on-time, dielectric pressure and wire feed) . Each parameter is at 3 levels, so 81 experiments were carried out. The experiments for machining of Al7075 via WEDM were modeled by machine learning methods. Four different models of two different methods were used for the prediction of surface roughness values of machined samples with WEDM. These models were extreme learning machine (ELM), weighted extreme learning machine (W-ELM), support vector regression machine (SVR) and quantum SVR (Q-SVR). All of the models were applied to the data set and the W-ELM model was the best performing model with the value of 0.9720 R^2.

Machine learning techniques such as artificial neural networks (ANN), support vector machines (SVM), and genetic algorithms (GA) are used by Paturi et. al. [14] to model and optimise surface roughness during Inconel 718 wire electrical discharge machining (WEDM). Surface roughness values were acquired using real-time WEDM experiments with various control parameters such as pulse on time, pulse off time, peak current, servo voltage, and wire feed rate. The best ANN model architecture was found to be 5-10-10-1, and SVM parameters were fine-tuned using the grid search method. The predictions of the ANN and SVM models were compared to the predictions of the response surface methodology (RSM), and performance was assessed using the correlation coefficient (R-value) between experimental and model predictions. The R-value of 0.99998 with experimental data and the least mean absolute percentage error (MAPE) of 0.0347 percent indicated that the SVM predictions were accurate among all the models evaluated. Furthermore, utilising the created RSM equation as the fitness function, the GA technique was executed, resulting in a 61.31 percent improvement in surface roughness. The suggested SVM and GA technique would aid in the prediction and optimization of surface roughness during Inconel 718 WEDM in a timely and precise manner.

Rahman [15] implemented an artificial intelligence model for predicting the best machining parameters for Ti-6Al-4V via electrical discharge machining (EDM) with copper as the electrode and positive polarity. Effects of peak current, servo voltage, pulse on- and off-time in EDM on material removal rate (MRR), tool wear rate (TWR), and surface roughness (SR) is studied. The artificial neural network (ANN) modelling of MRR, TWR, and SR is developed using a radial basis function neural network (RBFNN). The response surface methodology (RSM) technique is used to implement the design of experiments (DOE) method. Analysis of variance was used to conduct a validity evaluation of the suggested models' fit and adequacy (ANOVA). The proposed ANN model is used to estimate and verify the best machining settings. The created model is found

to be within the acceptable error range when compared to experimental results. Sensitivity analysis is used to determine the relative impact of various factors on performance measurements. Peak current has been demonstrated to have a significant impact on performance measurements. The obtained findings show that the suggested ANN models can accurately assess the MRR, TWR, and SR in EDM.

A new reduced modeling optimization framework is proposed by Surleraux et. al.[16], whereby the computational optimizer is replaced by an inexpensive surrogate that is trained by examplesMore specifically, given the geometry of the intended workpiece cavity, an artificial neural network (ANN) is trained using a small number of full reverse simulations and then used to directly produce optimal tool designs. A method of data augmentation is proposed to efficiently train the ANN, in which numerous characteristics from completely simulated EDM cavities are employed as distinct instances. Two ANNs are compared, one trained without changing process parameters (gap size and crater shape) and the other trained with a variety of process parameter cases. Two ANNs are compared, one trained without changing process parameters (gap size and crater shape) and the other trained with a variety of process parameter cases. When compared to the whole computational optimization procedure, the ANN can create unseen tool shape geometries with less than 6% variance and at essentially no cost in both circumstances. Our findings show that optimum tool shapes may be created very instantly, paving the way for virtual design and manufacturability evaluation of EDM die-sinking processes.

Majumder[17] adjusted the process parameters of electric discharge machining for optimal material removal rate and minimum wear ratio in their paper. The relationship between process parameters and machining performance was established using a properly trained neural network. The neural network model was then combined with three distinct evolutionary techniques, including simulated annealing, genetic algorithm, and particle swarm optimization, to forecast the optimum process parameters for highest material removal rate and minimum wear ratio. The performance of the evolutionary algorithms utilised thus far has been compared.

Zhang et. al.[18] suggested a new pulse classification approach based on the recurrent neural network (RNN) for high-speed EDM pulse analysis. During the machining process, discharge pulses of high-speed EDM were divided into five categories, which differed from typical EDM: open, spark, arc, partially short, and short. To assess the discharge pulses in the study, models based on three different RNNs with varying activation functions were created, including the standard RNN, LSTM (long short-term memory), and IndRNN (independently recurrent neural network). In the classification approach, a new input data structure based on the minimal signal change period was developed to simplify the model structure while also improving accuracy. The suggested model's highest classification accuracy is up to 97.85% without specifying thresholds,

and it can simultaneously categorise discharge pulses based on 10,000 orders of magnitude including diverse current values. The proposed method was successfully adapted to the complex machining conditions and the high-speed EDM's compound power source. Under varying currents, fluxes, and feeding speeds, the ideal model was utilised to examine the distribution of discharge pulses during the machining process. The proportion of discharge pulses could be anticipated with certainty. The management of discharge under varied machining parameters was disclosed more reliably by evaluating the discharge pulses of lengthy machining time, offering useful information for the enhancement of high-speed EDM servo systems.

Sahu et al[19]. studied the performance of a rapid prototyping (RP) based rapid tool during electrical discharge machining (EDM) of titanium as the work piece utilising EDM 30 oil as the dielectric medium. The tool electrode is constructed of AlSi10Mg and is created by selective laser sintering, an RP process. The quick tool's performance is compared to those of solid copper and graphite tool electrodes. Material removal rate, tool wear rate, and surface integrity of the machined surface measured in terms of average surface roughness (Ra), white layer thickness, surface fracture density, and micro-hardness on white layer are the machining performance parameters considered in this study. The feasibility of using a predictive tool like the least square support vector machine to provide instructions for practitioners to anticipate various machining performance parameters before actual machining has been investigated. The predictive model is deemed to be robust because root mean square error for several performance indicators ranges from 0.11 to 0.34. For simultaneously optimising the performance metrics, a hybrid optimization technique known as desirability based grey relational analysis in tandem with the firefly algorithm is used. Peak current and tool type have been found to have a considerable influence on all performance indicators.

The electrical discharge machining (EDM) of the aforementioned material is carried out with a 32-trial testing plan, and the outputs are predicted using artificial neural networks (ANN) and adaptive neuro-fuzzy inference system (ANFIS) by Pourasl et. al.[20]. The impacts of various key operational parameters, such as pulse on-time (Ton), pulse current (I), and voltage (V), on EDM process performance metrics such material removal rate (MRR), tool wear ratio (TWR), and average surface roughness (Ra), are investigated. Process plans (i.e., parameter–effect correlations) are established to guide the process operators. The MRR, TWR, and Ra estimate models were built using experimental data. To determine the error of training models, the root means the square error was utilised. Furthermore, the models' predicted outcomes have been validated through an unexplored set of tests. The results show that the ANFIS approach is superior to other ML techniques in terms of output parameter needs in electric discharge machining, with reduced RMSE and more dependable and accurate outcomes.

Conde et al.[21] proposed utilising an Elman-based Layer Recurrent Neural Network (LRNN) to forecast the correctness of components produced by WEDM. The average discrepancy between network predictions and real components is less than 6m, indicating that the network is doing exceedingly well. In a subsequent phase, a method for creating wire pathways with variable radius was proposed, allowing defects in machined parts to be remedied via software. Wire pathways of variable radius can be created by combining the predictions of the developed LRNN with the Simulated Annealing (SA) optimization technique, with radial deviations due to wire deformations eliminated. When the part radius is small and the part height is great, the wire's stiffness is lowered and the part's inaccuracy grows dramatically. The average deviation was lowered by as much as 80%, and the Coefficient of Variation (CV) was reduced by 43% in certain circumstances. Any current WEDM machine can be used to implement the solution.

In the research work of Pradhan et. al.[22] two neuro-fuzzy models and a neural network model are presented for predictions of material removal rate (MRR), tool wear rate (TWR), and radial overcut (G) in die sinking electrical discharge machining process for American Iron and Steel Institute D2 tool steel with copper electrode. The network's inputs are the discharge current (I_p), pulse length (T_{on}), duty cycle (τ), and voltage (V). The studies were carried out using a full-factorial design with varying amounts of I_p, T_{on}, τ and V. I_p is the most influencing factor for MRR and G, with the maximum degree of contributions of 87.61 percent and 81.90 percent, respectively, according to the analysis of variance results. Neural network architechtures were made with a different set of data to obtain appropriate number of neurons, epoch, and the fuzzy rule base. The comparison results reveal that the artificial neural network and the neuro-fuzzy models are comparable in terms of accuracy and speed, and further, the proposed models can be employed successfully in prediction of MRR, TWR, and G of the stochastic and complex EDM process.

Fazlollahtabar et al.[23] used the fuzzy theory to evaluate the impact of EDM settings on surface roughness, material removal rate, and electrode corrosion percentage. A fuzzy possibility regression model examines the relationship between the machining parameters and the output process specification, and mathematical relationships between exact inputs and fuzzy outputs of the EDM process are recovered. Interfacing models and fuzzy hypothesis testing are used to assess the effectiveness of the three outputs. A fuzzy adaptive neural network is utilised to identify the ideal levels of each output, and relevant models are generated to be modified with a fitted model of fuzzy possibility regression for comparison reasons. The computational results presented and the efficiency and effectiveness of the fuzzy adaptive neural network indicated the good performance and high accuracy of the proposed method for solving such problems.

Khan et. al.'s[24] research focuses on the creation of an artificial neural network (ANN) model for predicting surface roughness (Ra). Data that is discovered following the design of experiments is used for training and testing. Scanning electronic microscopy was used to examine the surface topography of the machined item. The results show that the ANN model can accurately predict surface roughness. Smaller craters and micro-cracks arise from low discharge energy, resulting in an appropriate surface structure. This method aids in cost-effective EDM machining.

The electrical discharge machining (EDM) process is described and optimised by Moghaddam et al.[25] utilising an artificial neural network and an optimization heuristic method. The EDM process' performance parameters include material removal rate (MRR), tool wear rate (TWR), and surface roughness (SR). The goal of this study is to optimise process parameters in order to identify a combination of process parameters that minimises TWR and SR while maximising MRR. Back propagation neural network (BPNN) was utilised to build the relationships between the input and output process parameters. The particle swarm optimization (PSO) algorithm was used to optimise the various response characteristics in the concluding phase of this study. A series of verification tests are also run to ensure that the optimization approach is accurate in determining the best levels of process parameters. The proposed modelling technique (BPNN) can accurately imitate the actual EDM process with less than 1% inaccuracy, according to the results. Furthermore, PSO algorithm outcomes with less than 4% error are extremely efficient in the optimization process.

Shakeri et al.[26] investigated wire electro-discharge machining of cementation alloy steel 1.7131 and developed a linear regression model and feedforward backpropagation neural network to predict surface roughness and material removal rate for effective machining. For the experiments, the full factorial experiment was adopted. Experiments were carried out with various pulse current, frequency, wire speed, and servo speed cutting circumstances. The optimised neural network with the best prediction performance contained eight neurons in the hidden layer and was capable of 0.773 percent overall mean prediction error, while the regression model indicated 2.547 percent mistakes. Overall, the comparison of findings revealed that the neural network is more reliable and accurate.

An Elman network is used by Pradhan et.al.[27] for the prediction of material removal rate (MRR) in electrical discharge machining (EDM). Training of the models is performed with data from series of EDM experiments on AISI D2 tool steel from finishing, semi-finish to roughing operations. The machining parameters such as discharge current, pulse duration, duty cycle, and voltage were used as model input variables during the development of predictive models. The developed model is validated with a new set of experimental data that was not used

for the training step. The mean percentage error of the model is found to be less than 6 per cent, which shows that the proposed model can satisfactorily predict the MRR in EDM.

The results of an experimental investigation to investigate the effect of parameters that create residual stresses during electric discharge machining of two particulate-reinforced metal matrix composites are reported in the article by Sidhu et al.[28] Several variables were changed to see how they affected the generation of residual stresses, with pulse-off time being the most important. To anticipate residual stresses, an artificial neural network was used. Residual stresses are lower in metal matrix composites with a low coefficient of thermal expansion and a high reinforced particle content. Additionally, when better conductive electrode materials are utilised during machining, residual stress is reduced. The residual stresses are accurately predicted by the artificial neural network model, making it a dependable method for predicting residual stresses. The micrographs reveal that a workpiece with a higher concentration of reinforced particles has a smaller flow line in the matrix material, resulting in less residual stresses. The phase transformation on the machined surface is also revealed by X-ray spectra.

Tzeng et al.[29] investigated the effects of process parameters on material removal rate, electrode wear ratio, and workpiece surface polish during the manufacturing of SKD61 by electrical discharge machining (EDM). To discover the best parameter settings for the EDM process, a hybrid method combining a back-propagation neural network (BPNN), a genetic algorithm (GA), and response surface methodology (RSM) was presented. Specimens were created using a Taguchi orthogonal array table and varied EDM processing conditions. The BPNN was trained to predict the material removal rate (MRR), relative electrode wear ratio (REWR), and roughness average (Ra) attributes using the data of 18 experimental runs. Simultaneously, the RSM and GA techniques were used to find the best possible setting. Furthermore, ANOVA was used to determine significant factors for the EDM process parameters, and the findings from the BPNN with integrated GA were compared to those from the RSM technique. The results reveal that the GA approach's proposed algorithm outperforms the RSM method in terms of prediction and confirmation.

Maji et al.[30] used an adaptive network-based fuzzy inference system to create input–output correlations of an electrical discharge machining process in both forward and backward orientations. The mappings took into account three input parameters: peak current, pulse-on-time, and pulse-duty-factor, as well as two output parameters: material removal rate and surface roughness. For the created adaptive network-based fuzzy inference system, which was designed utilising linear (say triangular) and non-linear (bell-shaped) membership function distributions of the input variables independently, a batch method of training was used with 1000 data. With the help of certain test cases collected from genuine trials, the performance of the proposed models

was tested for both forward and backward translations. The forward and reverse mapping difficulties are effectively addressed by an adaptive network-based fuzzy inference method. For the input variables, the fuzzy inference system using non-linear membership functions performs marginally better than the one using linear membership functions.

Krishnan et al.[31] used an artificial neural network with a feed-forward back-propagation algorithm and an adaptive neuro-fuzzy inference system to represent wire electrical discharge turning (WEDT). To train the neural network and test its performance, the experiments were created using Taguchi design of experiments. The process is adjusted with two output process factors in mind: material removal rate and surface roughness, both of which are critical for boosting product productivity and quality. Because the output parameters are incompatible, the process is optimised using a multi-objective optimization method based on the non-dominated sorting genetic algorithm-II. Using the proposed techniques, a pareto-optimal front leading to a collection of optimal solutions for material removal rate and surface roughness is generated. Experiments back up the findings, which show that it improves the performance of the WEDT process. Required input parameters can be adjusted using this collection of solutions to get a higher material removal rate and an excellent surface finish.

Ghadai et. al. [32] created a data-driven genetic programming based symbolic regression metamodels for electrical discharge machining (EDM). In this study, the data-driven method of metamodeling of production / machine processes is improved by implementing symbolic regression along with gene expression programming. Four different process parameters i.e. (voltage, heart rate, pulse duration, and current) are used as independent parameters to measure three different responses (removal rate, electrode aging rate, and surface hardness). The performance of metamodels is tested using various mathematical metrics such as R2, MAE, MSE. The effectiveness of metamodels in training and assessment data is found to be adequate for all responses.

1.4 Objective and scope of the present research works:

Machine learning models and neural networks are becoming increasingly used in every industry for detecting patterns in unlabelled data. From anticipating next session's sale to predicting stock market health, predicting consumer behaviour for a company, detecting cancer from a health report, facial recognition, and voice commands, machine learning (ML) and deep learning have a solution for every problem (DL). ML and DL have significantly decreased human effort and intervention in pattern recognition and calculation. This study aims to incorporate the benefits of ML and DL into a widely used non-conventional machining method, namely electric discharge machining. ML and DL will aid in reducing the human effort of trial and error in determining the best input parameters for achieving desired results. But the categories of ML and

DL models are not less. Trying and experimenting with each model to get the best forecast results will take up more human time. The goal of this study is to evaluate certain commonly used metamodeling techniques and neural networks on electric discharge machining and electric discharge turning processes. This research work also involves several accuracy and error metrics to study the performances of the prediction model over actual responses. Finally, a conclusion to be drawn declaring the most accurate and consistent metamodel and neural network so that the time can be reduced to find the best model for building prediction models for such type of dataset.

2 Different prediction models:

2.1 PR metamodel

The PR is a simple but useful technique to represent the nonlinear relationship between various input (process parameters) and output (response) variables [33]. In general, an m^{th}-order PR model can be represented in the following matrix form:

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^m \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \ldots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \ldots \\ \varepsilon_n \end{bmatrix} \tag{2.1}$$

where y_1, y_2,...,y_n are the output variables, x_1, x_2,...,x_n are the input variables, β_0, β_1,...,β_n are the coefficients and ε_1, ε_2,...,ε_n are the random error terms.

In this paper, a second-order PR metamodel having the following form is considered to explore the nonlinear relationships between the EDM process parameters and responses.

$$y(x)=\beta_0+\sum_{i=1}^{n}\beta_i x_i+\sum_{i=1}^{n}\beta_{ii}x_i^2+\sum_{i\neq j}^{n}\beta_{ij}x_i x_j+\varepsilon \tag{2.2}$$

where $y(x)$ is the response, β_0 is the intercept, β_i is the coefficient of i^{th} process parameter, x_i is the i^{th} process parameter (i = 1,2,…,n), β_{ii} is the coefficient of x_i^2 term, β_{ij} is the coefficient for process parameters i and j ($i \neq j$), and ε is the random error term.

2.2 RBF metamodel

A RBF is a real-valued function which depends on the Euclidian distance between the layer of inputs and some fixed point which is either origin or any other point. It is one of the classical algorithms of supervised learning which functions as a radial kernel of some other machine learning algorithms, like GPR and SVM [34]. The surrogate RBF model represents the integration function as a linear combination of basic functions, one for each training data. It is so named because the basic functions depend only on the Euclidean range from the predictive location to the primary activity training area. In this model, the coefficients of the basic functions are calculated during the training phase maintaining the best fit. The RBFs are often added with a polynomial function to capture the general trends. The prediction equation of RBF can be represented using the following expression:

$$y=f(x).w_p+\sum_{i}^{nt}\psi(x,x_i)w_r \tag{2.3}$$

where $x \in R^{nx}$ is the prediction input vector, $y \in R$ is the prediction output, $x_i \in R^{nx}$ is the input vector for i^{th} training point, $f(x) \in R^{np}$ is the vector mapping the polynomial coefficients to the

prediction output, $\psi(x, x_i) \in R^{nt}$ is the vector mapping the RBF coefficients to the prediction output, $w_p \in R^{np}$ is the vector of polynomial coefficients, and $w_r \in R^{nt}$ is the vector of RBF coefficients. The coefficients, w_p and w_r, can be computed by solving the following linear system:

$$\begin{bmatrix} \psi(x_1,x_1) & \dots & \psi(x_1,x_{nt}) & f(x_1)^T \\ \dots & \dots & \dots & \dots \\ \psi(x_{nt},x_1) & \dots & \psi(x_{nt},x_{nt}) & f(x_{nt})^T \\ f(x_1) & \dots & f(x_{nt}) & 0 \end{bmatrix} \begin{bmatrix} w_{r1} \\ \dots \\ w_{rnt} \\ w_p \end{bmatrix} = \begin{bmatrix} y_1 \\ \dots \\ y_{nt} \\ 0 \end{bmatrix} \tag{2.4}$$

Some of the most commonly employed vector mapping equations of RBF include thin plate spline, Gaussian basis function, multi-quadratic function and inverse multi-quadratic function. In this paper, Gaussian basis function is considered due to its several advantageous features, like excellent representation of nonlinear relationship between the input and output variables, faster computational speed, exact modeling of the real-time data etc. It can be represented using the following equation [35]:

$$\psi(x_i, x_j) = \left(\frac{\left\| x_i - x_j \right\|^2}{d_0^2} \right) \tag{2.5}$$

where d_0 is the scaling parameter.

The RBF metamodel has already been applied as an effective prediction tool in various machining processes. Leyva-Bravo et al. [36] adopted three soft computing tools, i.e. fuzzy logic, back-propagation neural network (BPNN) and RBF for prediction of MRR while treating gap voltage, peak current and frequency as the input parameters of an electrochemical discharge machining process. It was observed that BPNN and RBF had better prediction accuracy as compared to fuzzy logic. The RBF was also employed by Nguyen et al.[37] to predict power factor, energy consumption and surface roughness during dry milling of stainless steel 304 material. Cutting speed, feed rate, depth of cut and nose radius of the tool were considered as the input parameters. It was observed that there had been a good congruence between the predicted and experimental results.

2.3 Kriging metamodel

Kriging is the most popular interpolation method in statistics, precisely in geostatistics [38]. It is the linear combination of a weighted deterministic function which is added to the realization of a stochastic process. Depending on the stochastic characteristics of the random field and different degrees of conception, it has several forms, like ordinary kriging, simple kriging,

universal kriging, indicator kriging etc. The basic form of an ordinary kriging can be presented as below:

$$\hat{y} = \sum_{i=1}^{k} w_i f_i(x) + Z(x) \tag{2.6}$$

where $f_i(x)$ is the deterministic function which can be constant, linear, quadratic or multi-quadratic and w_i is the weight assigned to $f_i(x)$. In this equation, the deterministic term may be considered as constant. On the other hand, $Z(x)$ is the realization of a stochastic process with mean zero and spatial covariance function, as given in Eq. (2.7).

$$\mathrm{cov}\left[Z\left(x^{(i)}\right), Z\left(x^{(i)}\right)\right] = \sigma^2 R\left(x^{(i)}, x^{(j)}\right) \tag{2.7}$$

where σ^2 is the process variance and R is the correlation function. Several correlation functions, like exponential correlation function (Ornstein-Uhlenbeck process), squared exponential (Gaussian) correlation function, Matérn 5/2 correlation function, Matérn 3/2 correlation function etc. can be considered while developing the corresponding kriging metamodels. In this paper, for prediction of the response values of the considered EDM processes, the Gaussian correlation function is employed (due to its simple representation and straightforward computation) which can be expressed using the following equation:

$$\prod_{l=1}^{nx} \exp\left(-\theta_l \left(x_l^{(i)} - x_l^{(j)}\right)^2\right), \ \forall \theta_l \in R^+ \tag{2.8}$$

where θ_l is the initial hyperparameter.

During dry milling of SKD61 steel material with depth of cut, feed rate, spindle speed and tool nose radius as the input parameters, Nguyen [39] developed the corresponding kriging metamodels for prediction of specific cutting energy, Ra, distance between the highest peak and the deepest valley (Rz) and MRR. It was confirmed that kriging metamodels could be employed instead of PR and ANN to better capture the nonlinear relationships between various milling parameters and responses. Based on simulated data, Meng et al.[40] proposed the deployment of kriging metamodels to predict cost per volume and machining temperature during corner-milling of H62 brass material. The developed metamodels were observed to be quite accurate in predicting the considered responses with mean error of 6% and 3% respectively. Finally, a *K*-means PSO algorithm was employed to solve the multi-objective optimization problem of the said milling process.

2.4 GEP

It belongs to the family of evolutionalry algorithms which overcomes many of the limitations of GA and genetic programming [41]. It consists of a number of operators, like replication, mutation, transportation, recombination etc. In GEP, a chromosome is a fixed-length linear expression representing genetic information encoded. Thereafter, genetic information is

transformed into indirect objects (computer programs or expression trees (ETs)) of varying lengths and sizes. The chromosome may have more than one gene, each gene encoding (genotype) represents small subprograms. Each chromosome in real value is then expressed (phenotype) through a random process. To assess suitability of each chromosome, a set of fitness functions is considered. The chromosomes with the right solutions are selected based on their fitness values, and re-evaluated by genetic modification (mutation, transportation and recombination) and re-testing. This process is repeated until the most suitable solution (chromosome) is achieved having the required level of accuracy.

The GEP has two main components, i.e. chromosome and ET; the latter being the expression of the genetic information encoded (translation) in the former. Translation is the one-to-one relationship between symbols of the gene and nodes they represent in the tree. The rules for translation define the spatial arrangement of the nodes in the ET and types of interactions between their sub-elements (sub-ETs) or phenotype. Figure 2.1 presents an example of ET of an evolved GEP, where d_0, d_1 and d_2 are the input variables. In this tree, there are also several algebraic functions, like +, -, *, /, average and inverse. It also considers other random constants, like c_0, c_1 and c_2 in the model.

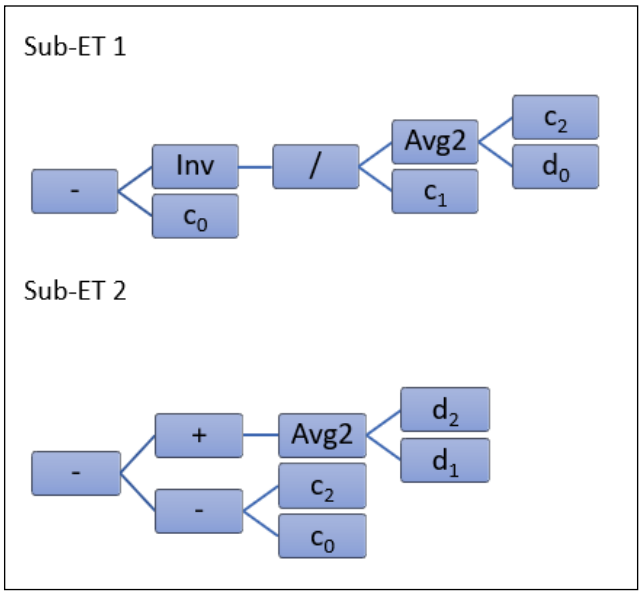

Figure 2.1 Example of an ET in GEP

The corresponding algebraic equation that can be derived from the ET of Figure 2.1 is represented as shown below:

$$y=\left(\frac{1}{\frac{Avg2(c_2,d_0)}{c_1}}-c_0\right)+\left(Avg2(d_2,d_1)-(c_2-c_0)\right) \tag{2.9}$$

During minimum quantity lubrication (MQL)-assisted milling of Inconel 690 alloy, Sen et al. [42] applied two predictive models, i.e. GEP and ANN to monitor tool wear based on cutting speed, feed rate, depth of cut and flow rate of MQL as the input parameters. It was concluded that GEP had better prediction performance than ANN. Shah et al. [43] modelled two important properies of fly ash, i.e. compressive strength and splitting tensile strength using GEP, and later compared its prediction performance against RSM, multiple linear and nonlinear regression models.

2.5. *Feed-forward neural network (FNN):*

An FNN is a mathematical model that is inspired by the biological NNs' functional features. A NN is made up of a group of artificial neurons that work together to interpret data in a connectionist manner. In general, an FNN is an adaptive system that adjusts its structure in response to external or internal data that flow over the network during the learning process. Figure 2.2 provides a general representation of an FNN. In this figure, the NN has one input layer with four neurons, one hidden layer having five neurons and one output layer with one neuron. Information from the input layer combined with appropriate weights move to the hidden layer where the information coming from different neurons are accumulated and the most weighted information is passed to the output layer. This whole process does not send back any information to the previous neurons for feedback, information move only in forward direction. So, this NN is called as FNN.

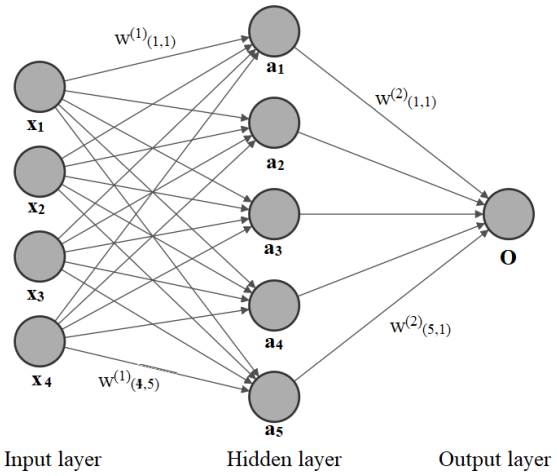

Figure 2.2 Architecture of a typical FNN

2.6 Convolutional neural network (CNN)

The CNN is a deep learning model for data processing with a grid pattern, such as photographs. It is inspired by the organization of animal visual cortex [44], [45], and meant to learn spatial hierarchies of characteristics, from low- to high-level patterns, automatically and

adaptively. A typical CNN is usually made up of three types of layer (or building blocks), i.e. convolution, pooling and fully connected layers. The first two layers (convolution and pooling) extract features, whereas, the third one which is a fully linked layer, transfers those features into final output, such as classification. The convolution layer is an important component of CNN, consisting of a stack of mathematical operations, like convolution, which is a specific sort of linear operation. A CNN can effectively analyze one-dimensional (forecasting, regression models), two-dimensional (picture pattern recognition) or three-dimensional (MRI, CT scan analysis) datasets. In this paper, for predicting the response values of an EDT process, one-dimensional CNN (1D CNN) is employed. Figure 2.3 exhibits a simple representation of a CNN model. The first layer in this model is an input layer represented by 1D arrays. Each array is a representation of one data point. The second layer is a 1D convolutional layer. Each block of the array represents the imputation of several input layers or several input features. The next layer is a flatten layer where the layers of convolutional 1D are flattened into one single array. The most important imputed data from the flatten layer is finally obtained in the output layer.

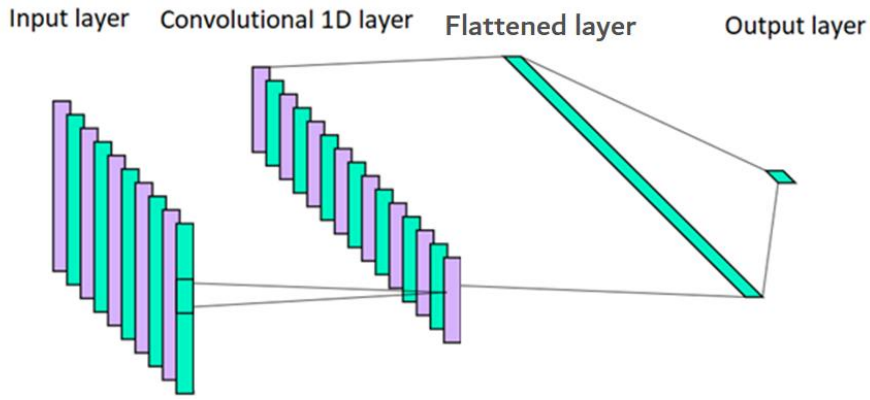

Figure 2.3 Representation of a typical CNN architecture

2.7 Recurrent neural network (RNN)

A RNN is a type of NN in which nodes form a directed or undirected graph along a temporal axis. As a result, it can display temporal dynamic behavior of a given system. The RNN, which is based on FNN, can process variable length sequences of inputs using their internal state (memory). The term 'RNN' is employed to describe a type of network with an infinite impulse response, whereas, 'CNN' is considered to represent a type of network having a limited impulse response. An infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled and replaced with FNN. On the other hand, a finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly FNN. Figure 2.4 is the flowchart representation of a traditional RNN process, as developed by Zhang et. al.[18]. The computational process of the traditional RNN can be explained using the following equation:

$$h^{<t>} = \sigma\,(W\,x^{<t>} + Uh^{<t-1>} + b) \tag{10}$$

where $x^{<t>}$ is the input data at time step t, $h^{<t-1>}$ is the information from the previous cell, W and U are the weight matrixes, b is the bias vector, and σ is the activation function. The computational result $h^{<t>}$ is entered into the next cell. This cycle goes on until all the determined epochs are run.

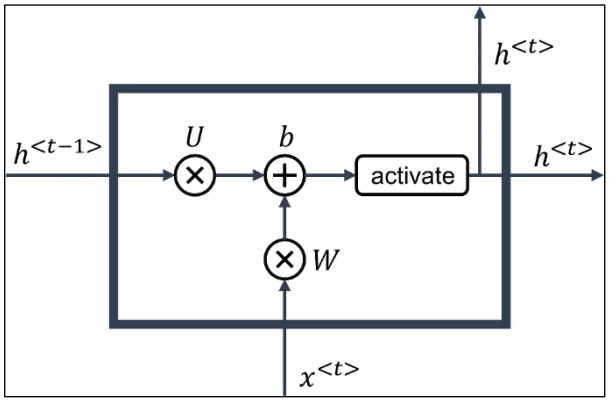

Figure 2.4 Representation of a typical RNN flow [18]

2.8. RNN with long short term memory (LSTM)

The LSTM has a RNN architecture that is artificial in nature. The LSTM has feedback connections, unlike normal FNNs. It can effectively deal with not only individual data points (such as photos), but also complete data streams (such as speeches or videos). A cell, an input gate, an output gate and a forget gate usually make up a typical LSTM model. These three gates control the flow of information into and out of the cell, and the cell remembers values across arbitrary time intervals.

Figure 2.5 provides a simple representation of an LSTM cell. The input value x_t after being concatenated to the previous cell output h_{t-1} first moves through the tanh layer. The input is then passed through an input gate which is activated by sigmoid function (σ). In the next step, it passes through a forget gate loop where the internal state variable s_t and s_{t-1}, is added to the input data to develop an effective layer of recurrence. Through this process, the network learns to decide which state of variables should be remembered or forgotten. Finally, there is a tanh squashing function, whose output is controlled by an output gate. This gate determines which values are actually permitted as cell output h_t.

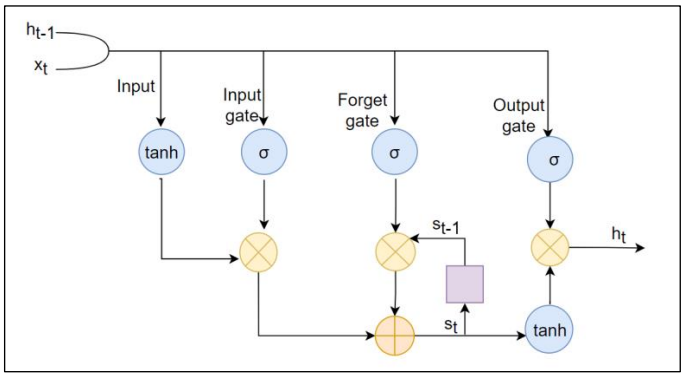

Figure 2.5 Representation of a typical LSTM cell

2.9. General regression neural network (GRNN)

The GRNN is a memory-based FNN which is a combination of radial basis function network (RBFN) and probabilistic neural network (PNN). The GRNN asymptotically converges to the ideal regression surface as the number of training samples increases. The GRNN has a unique property in that it does not require iterative training, in addition to having a solid statistical foundation. The GRNN training is a one pass technique, unlike the most prevalent error-back-propagation (EBP) algorithm, which trains multilayer feedforward networks iteratively. Furthermore, the GRNN formulation has only one free parameter that can be tuned quickly. As a result, when compared to EBP-based training, the GRNN trains itself in much less time.

The general network flow of a GRNN architecture is exhibited in Figure 2.6. This architecture consists of four layers, i.e. input layer, pattern layer, summation layer and output layer. Each pattern unit corresponds to a single training sample. The chance of an input vector fitting into a pattern unit is estimated by each pattern unit. The pattern layer's neurons are organized into K groups (to be decided by the model itself), one for each category. The RBF kernel is employed by i^{th} pattern neuron in k^{th} group to compute its output. The neurons of summation layer compute the approximation of the conditional class probability function through a combination of previously computed densities.

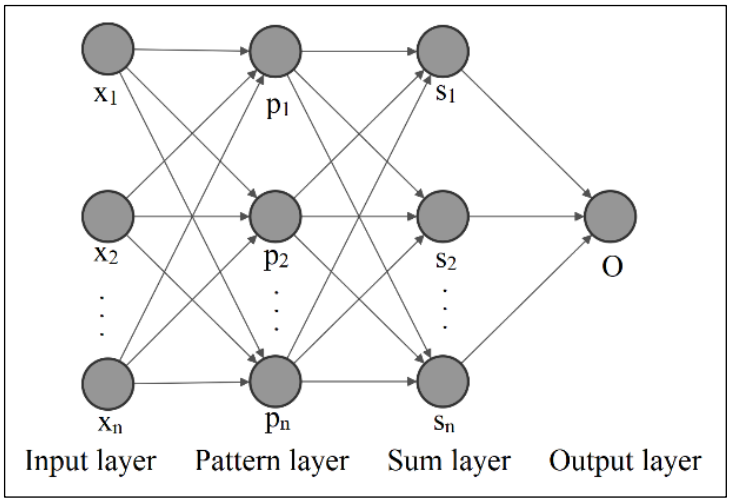

Figure 2.6 Representation of a GRNN architecture

3. Prediction of EDM responses using different metamodeling techniques:

3.1 Experimental data

In order to contrast the prediction performance of PR, RBF, kriging and GEP metamodels with respect to four model accuracy metrics, the experimental data of Anitha et al. [46] is considered in this example. Using a die-sinking EDM setup (Model: Electronica Electraplus PS 50ZNC) and electrolytic copper electrode (30 mm diameter with positive polarity), Anitha et al. [41] performed 30 experiments to study the effects of pulse current (Ip) (in A), pulse duration (Ton) (in μs), duty cycle (DC) and voltage (V) (in V) on volumetric MRR (in mm^3/min) and Ra (in μm). The workpiece material was AISI D2 (DIN 1.2379) tool steel (rectangular in shape having density and thickness of 7.7 g/cc and 4 mm respectively). Commercial grade EDM oil was utilized as the dielectric fluid and a side flushing technique with pressure 0.3 kgf/cm^2 was maintained during the EDM experiments. During the experiments, pulse current was varied between 5 and 15 A, whereas, the range for pulse duration was from 50 to 100 μs. The duty cycle and voltage were varied between 50 and 83, and 40 V and 50 V respectively. Table 3.1 shows the experimental dataset along with the measured response values.

Table 3.1 Experimental dataset [46]

Exp. No.	EDM process parameters				Responses	
	Ip (A)	Ton (μs)	DC	V (V)	MRR (mm^3/min)	Ra (μm)
1	10	75	66.5	45	9.04	5.98
2	5	50	50	50	5.18	5.01
3	5	100	83	40	5.25	5.03
4	5	50	83	40	8.87	4.71
5	15	100	50	50	51.09	8.1
6	10	75	66.5	45	8.95	6.12
7	5	100	50	40	4.35	4.89
8	15	100	50	40	51	10.93
9	5	100	83	50	6.97	5.7
10	15	100	83	40	33.02	12.49
11	5	50	83	50	14.12	5.19
12	10	75	66.5	45	8.42	6.54
13	15	50	83	40	20	12.01

14	10	75	66.5	40	8.94	8.2
15	10	75	83	45	9.36	7.13
16	15	75	66.5	45	33.08	9.68
17	10	50	66.5	45	9.18	5.87
18	5	75	66.5	45	5.36	6.07
19	10	75	66.5	45	10.35	5.55
20	15	50	83	50	29.16	8.43
21	5	50	50	40	4.61	4.59
22	15	50	50	40	29.74	10.49
23	10	75	66.5	45	11.01	6.25
24	10	75	50	45	9.25	5.92
25	15	50	50	50	33.1	7.43
26	5	100	50	50	4.35	5.59
27	15	100	83	50	33.11	9.01
28	10	75	66.5	50	11.01	6.35
29	10	100	66.5	45	10.43	7.27
30	10	75	66.5	45	9.35	6.75

3.2 Data sampling

The application of any of the metamodeling techniques as well as machine learning algorithms starts with a set of training data. The success of a metamodel as an effective and accurate prediction tool thus depends on how well it has been trained using the data provided. Therefore, the training dataset should adequately represent all the features under the design space. It is thus required to build a training database from the design space without any bias. For this purpose, several sampling techniques are available, including random sampling, Latin hypercube sampling, Hammersley sequence sampling etc. In this paper, random sampling technique is considered to generate both the testing and training data points. It has numerous advantages, like less chance of error, equal chance of selection of data points, less knowledge requirement, simplest way of data generation, minimum biasness, applicability to entire population etc. The training data points are employed to train and construct a metamodel, while the testing data points are utilized to validate the metamodel and check its prediction accuracy over a new set of input data points. In this example, from the experimental dataset of Table 2, a randomly chosen set of

80% of the available observations (i.e. 24) is used to train all the four metamodels and the remaining 20% of the observations are considered to test those metamodels. However, for a fair analysis, both the training and testing datasets are considered to be the same for all the metamodels.

3.3 Selection of the tuning parameters

For each of the metamodeling techniques, selection of the most appropriate set of tuning parameters during training of the metamodels is extremely important. Proper combination of those tuning parameters results in development of an accurate metamodel. In this paper, several trials are conducted with different combinations of the tuning parameters and those parameters with the best results are noted. The values of the considered tuning parameters for each of the metamodeling techniques are summarized in Table 3.2.

Table 3.2 Values of different tuning parameters

Metamodel	Tuning parameter	Value
PR	Degree of polynomial	2
RBF	Basis function scaling parameter (d_0)	7
	Polynomial degree	2 (Quadratic)
Kriging	Number of optimization runs	10
	Initial hyperparameters	0.01
GEP	Number of chromosomes	30
	Head size	8
	Number of genes	3
	Linking function	Addition (+)
	Fitness function	RMSE
	Mutation rate	0.00138
	Insertion sequence transposition rate	0.00546
	Root insertion sequence transposition rate	0.00546
	Inversion rate	0.00546
	One-point recombination rate	0.00277
	Two-point recombination rate	0.00277
	Gene recombination rate	0.00277

	Gene transposition	0.00277
	Constants per gene	10
	Data type	Floating-point
	Lower bound	-100
	Upper bound	100

3.4 Results and discussion

As mentioned earlier, this example deals with development of four metamodels in the form of PR, RBF, kriging and GEP to predict MRR and Ra values of an EDM process while machining AISI D2 tool steel material. At first, based on the training dataset, the following two PR-based metamodels are developed with the help of a Python library Scikit-Learn.

$$PR(MRR) = 28.439 - 3.736\times Ip + 0.303\times Ton + 1.194\times DC - 3.207\times V + 0.42\times Ip^2 + 0.035\times Ip\times Ton - 0.058\times Ip\times DC - 0.013\times Ip\times V + 0.001\times Ton^2 - 0.005\times Ton\times DC - 0.009\times Ton\times V - 0.004\times DC^2 + 0.005\times DC\times V + 0.044\times V^2 \quad (3.1)$$

$$PR(Ra) = 29.69 + 1.043\times Ip + 0.003\times Ton + 0.374\times DC - 1.826\times V + 0.048\times Ip^2 + 0.001\times Ip\times Ton + 0.004\times Ip\times DC - 0.041\times Ip\times V - 0.002\times DC^2 - 0.002\times DC\times V - 0.024\times V^2 \quad (3.2)$$

In the similar direction, using appropriate values of the considered tuning parameters, the corresponding RBF and kriging metamodels are subsequently developed for prediction of MRR and Ra responses. In case of RBF metamodel, 20 trials with the basic function scaling parameter (d_0) ranging between 1 and 10, and polynomial degree as 1 and 2 are conducted, and it is noticed that the best performance of RBF with respect to maximum R^2 value is attained at $d_0 = 7$ and polynomial degree = 2. For kriging model, the number of optimization runs is decided based on elbow method, and it can be noted that with 10 optimization runs, the most satisfactory result with respect to maximum R^2 value can be obtained with minimum computational cost. On the other hand, for construction of GEP-based metamodels, a computer software called GenXpro Tools 5.0 is employed. Besides the four basic arithmetic operators (+, -, *, /), some other mathematical functions, like exponential, log, inverse, power of 2, cube root, minimum of 2, maximum of 2, average, arctangent, hyperbolic tangent, complement etc. are also considered for developing the GEP-based models for MRR and Ra. In this paper, '/' is defined as the protected division which returns a value of 10^6 if division by zero is encountered. Addition operator is considered as a linking function to connect the mathematical terms encoded in each gene. While developing these GEP metamodels, all the four genetic operators, i.e. mutation, recombination, inversion and transportation are used. Figure 3.1 exhibits the corresponding ET representing the relationships between the EDM process parameters and MRR. Similar ET can also be developed

for Ra response. These ETs lead to the development of GEP-based metamodels for MRR and Ra, as provided in Eqs. (3.3) and (3.4) respectively.

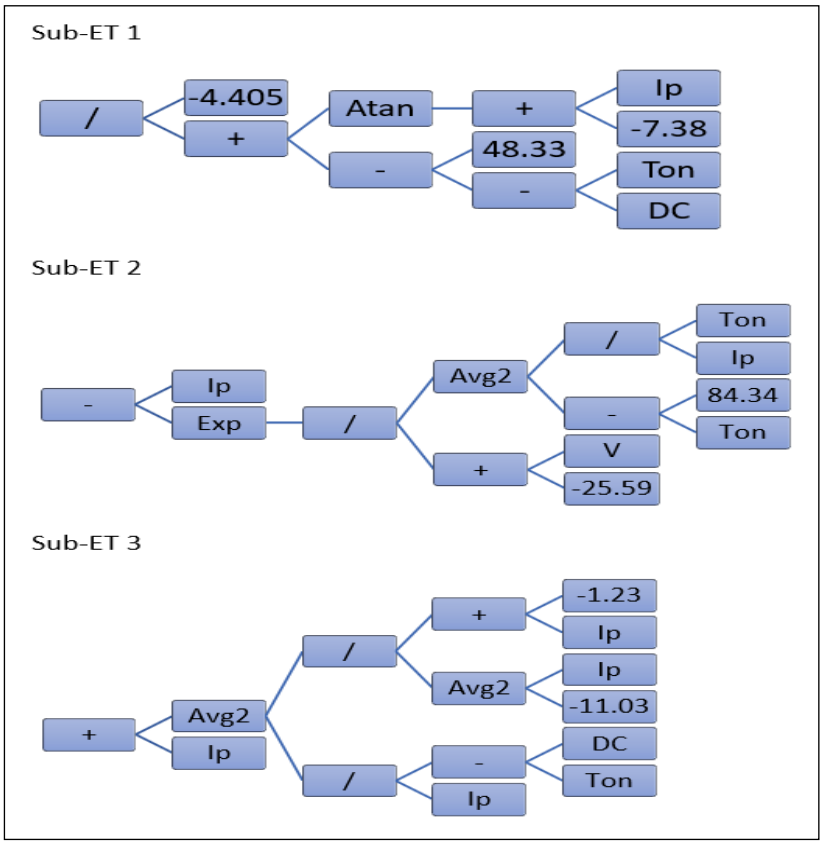

Figure 3.1 Expression tree for MRR

$$GEP(MRR) = \left(\frac{-4.405}{tan^{-1}(Ip-7.38)+48.33-(Ton-DC)}\right) + \left(Ip - \exp\left(\frac{Avg2\left(\frac{Ton}{Ip},(84.34-Ton)\right)}{V-25.59}\right)\right) + \left(Avg2\left(\frac{(-1.23+Ip)}{Avg2(Ip,-11.03)},\frac{DC-Ton}{Ip}\right) + Ip\right) \quad (3.3)$$

$$GEP(Ra) = \left(1 - tan^{-1}\left(\max 2\left(Avg2\left(V,(V-86.75)\right),1-\sqrt[3]{Ip}\right)\right)\right) + \left(tan^{-1}\left(tan^{-1}(Avg2(-75.96,V)\times Ip + 39.07 + V)\right)\right) + Avg2\left(\frac{1}{tan^{-1}(V^2)},\min 2(V,\sqrt[3]{DC} + Ip)\right) \quad (3.4)$$

After proper training of all the developed metamodels with the considered dataset, an attempt is now put forward to predict both the values of MRR and Ra for the EDM process. Table 3.1 provides the target and predicted values for both the responses (including training as well as

testing dataset) for all these metamodels. To have better visualization of the prediction performance of the metamodels, both the target and predicted values for MRR and Ra are plotted in Figures 3 and 4 respectively. It can be revealed from these figures that all the developed metamodels are quite capable of predicting both these responses within a ±20% error band for the considered EDM process. In fact, it is evident from Figure 3.2 that for MRR prediction on training data, all the metamodels have near ideal estimation with most of the data points lying on the diagonal identity line or hugging it. However, in case of RBF and GEP metamodels, one training data point is found to be beyond the ±20% error band. Nevertheless, GEP metamodel shows superior performance on the testing dataset, with almost all the test data predictions being in the ±20% error threshold. This indicates that there is no overtraining in the GEP metamodel and it has excellent generalization. On the contrary, the RBF metamodel, despite having a near ideal performance on the training data, has three large error outliers in the testing data. Interestingly, the performance of PR metamodel is observed to be quite similar to kriging metamodel for both training and testing data.

In case of prediction of Ra by the metamodels (Figure 3.3), a trend similar to MRR is observed. The PR and kriging metamodels have quite similar prediction behaviour, where most of the relatively poor predictions are noticed towards the lower Ra values. The RBF has more extreme error data points as compared to other metamodels. It should be pointed out here that in the experimental dataset (Table 3.3), the central point is replicated six times to estimate pure error for the lack of fit test. All the metamodels seem to have dealt with this repeated point (with different response values) by approximating it around the mean of the six replicates. This is why a small horizontal cluster of data points is observed in the lower Ra region.

The deviations of the predicted response values from the target ones, i.e. residuals for each of the developed metamodels are portrayed in Figure 3.4. It should be noted here that the zero line in Figure 3.4 represents zero error in prediction, whereas, points lying above and below it indicate underprediction (i.e. target value greater than predicted value) and overprediction (i.e. target value lower than predicted value) respectively. In general, the performance of all the metamodels (except RBF) on test data is similar to their corresponding performance on training data. This indicates that the training is adequate. There is also no observable pattern in the scatter of the residuals indicating lack of biasness.

Table 3.3 Target and predicted values of MRR and Ra for the developed metamodels

Exp. No.	MRR					Ra				
	Target	PR	RBF	Kriging	GEP	Target	PR	RBF	Kriging	GEP

1	9.04	9.199	9.520	9.161	9.384	5.98	6.474	6.242	6.486	6.325
2*	5.18	5.374	5.180	5.380	6.802	5.01	5.581	3.491	5.611	5.231
3	5.25	4.766	5.250	4.752	6.367	5.03	5.260	5.030	5.243	5.641
4	8.87	9.766	8.870	9.738	7.964	4.71	4.802	4.710	4.794	4.641
5	51.09	49.935	51.090	49.960	50.980	8.1	8.008	8.100	8.016	8.238
6	8.95	9.200	9.520	9.161	9.384	6.12	6.474	6.242	6.486	6.325
7	4.35	4.095	4.350	4.079	4.767	4.89	4.890	4.890	4.885	5.302
8	51	51.095	51.000	51.068	51.080	10.93	11.112	10.930	11.110	10.630
9	6.97	6.715	6.970	6.731	6.436	5.7	5.700	5.700	5.689	5.570
10	33.02	32.571	33.020	32.561	32.031	12.49	12.655	10.665	12.669	11.969
11*	14.12	16.349	17.060	16.344	10.140	5.19	5.104	5.190	5.127	5.570
12	8.42	9.200	9.520	9.161	9.384	6.54	6.474	6.242	6.486	6.325
13	20	20.194	20.000	20.194	20.816	12.01	11.800	12.010	11.807	11.969
14	8.94	8.940	8.940	8.949	9.133	8.2	7.783	8.200	7.790	8.200
15*	9.36	6.115	14.433	6.045	10.240	7.13	6.067	7.822	6.068	6.481
16	33.08	33.720	33.080	33.702	31.627	9.68	9.959	9.680	9.948	9.823
17	9.18	7.312	9.180	7.365	9.463	5.87	6.085	5.870	6.092	6.325
18*	5.36	5.665	5.802	5.639	6.539	6.07	5.405	6.070	5.417	5.783
19	10.35	9.200	9.520	9.161	9.384	5.55	6.474	6.242	6.486	6.325
20	29.16	25.459	18.510	25.370	32.348	8.43	7.991	9.561	7.986	8.577
21*	4.61	0.589	16.585	0.654	4.625	4.59	4.713	4.590	4.717	5.302
22*	29.74	30.209	20.110	30.263	29.681	10.49	10.539	9.160	10.529	10.630
23	11.01	9.200	9.520	9.161	9.384	6.25	6.474	6.240	6.486	6.325
24	9.25	9.890	9.250	9.872	8.487	5.92	5.534	5.920	5.550	6.142
25	33.1	33.679	33.100	33.698	21.213	7.43	7.296	7.430	7.299	7.238
26	4.35	4.250	4.350	4.234	4.837	5.59	5.896	5.590	5.900	5.231
27	33.11	33.205	33.110	33.194	31.931	9.01	8.984	9.010	8.984	8.577
28	11.01	11.650	11.010	11.573	9.515	6.35	6.381	6.350	6.394	5.926
29	10.43	12.938	10.430	12.877	8.645	7.27	6.669	7.270	6.667	7.325
30	9.35	9.200	9.520	9.133	9.384	6.75	6.474	6.240	6.516	6.325

*Testing data

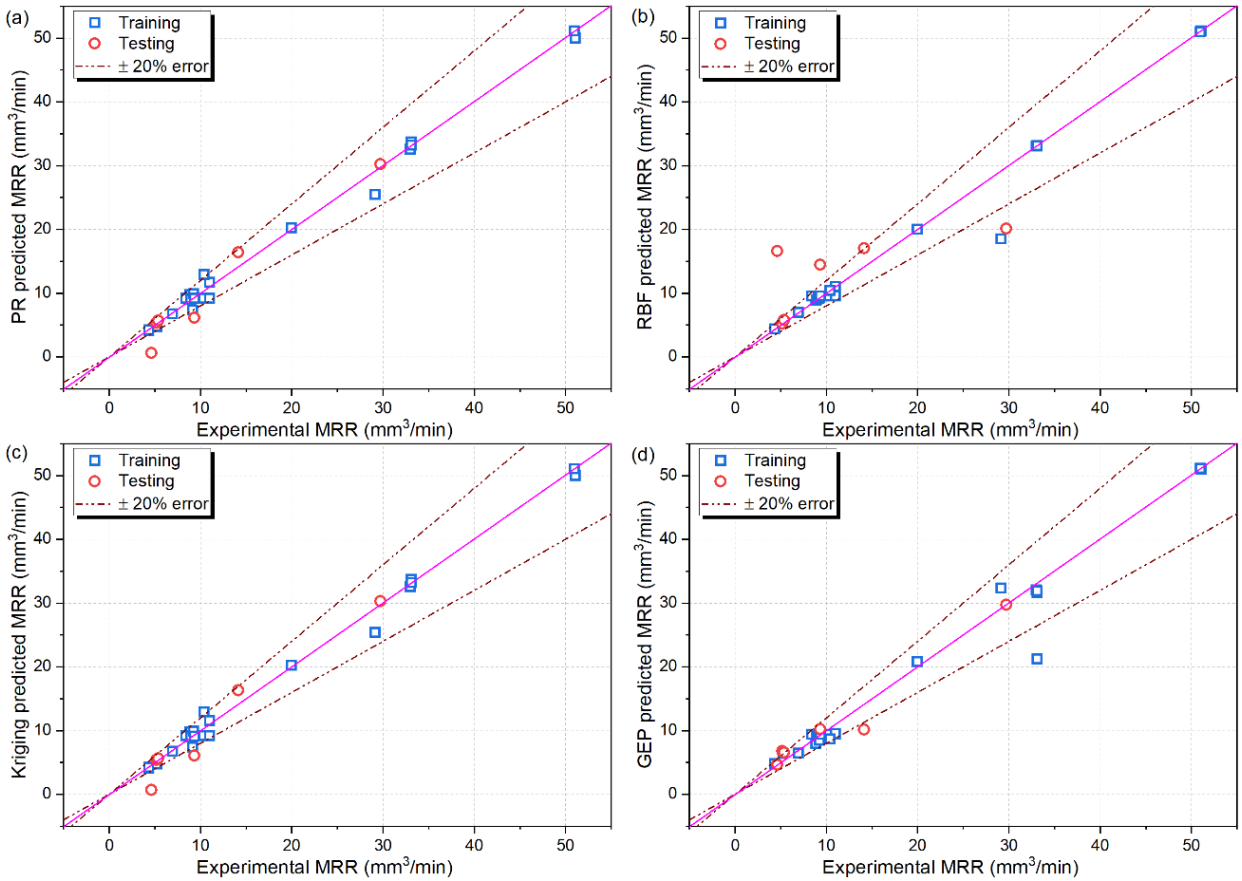

Figure 3.2 Comparison between target and predicted MRR values for a) PR, b) RBF, c) kriging and d) GEP

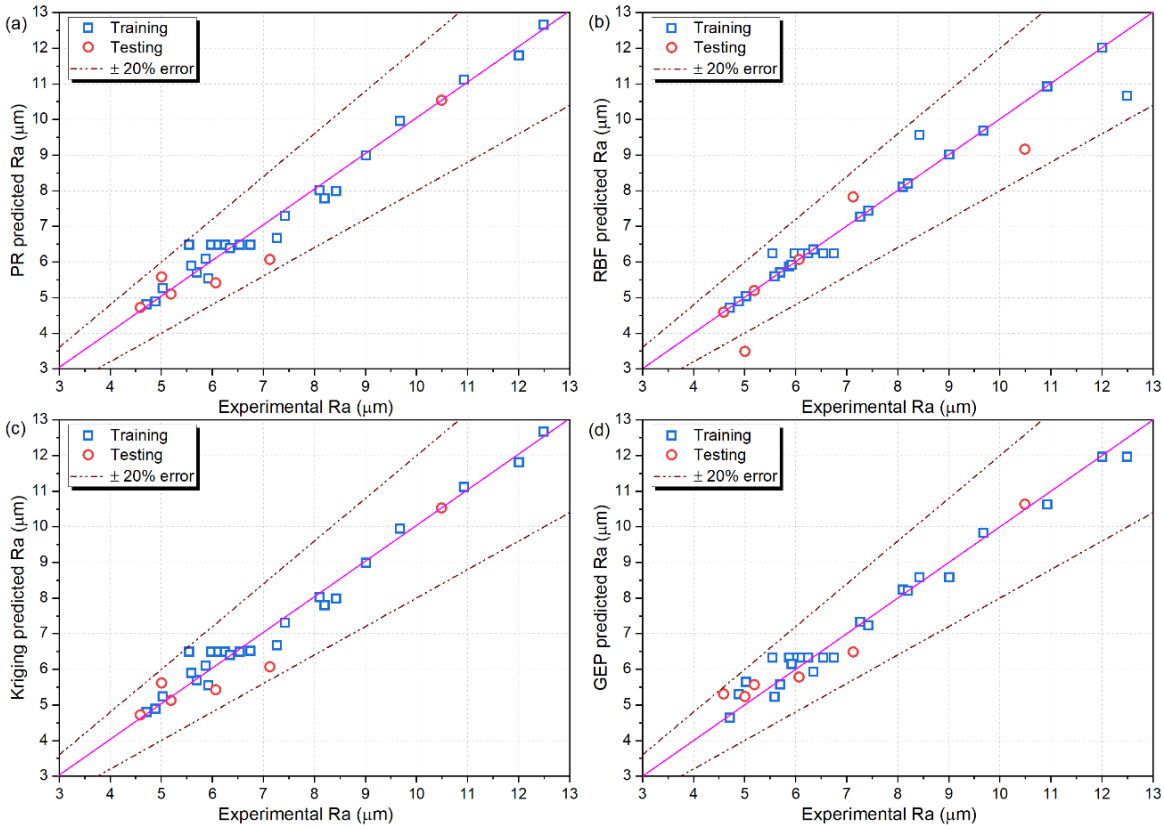

Figure 3.3 Comparison between target and predicted Ra values for a) PR, b) RBF, c) kriging and d) GEP

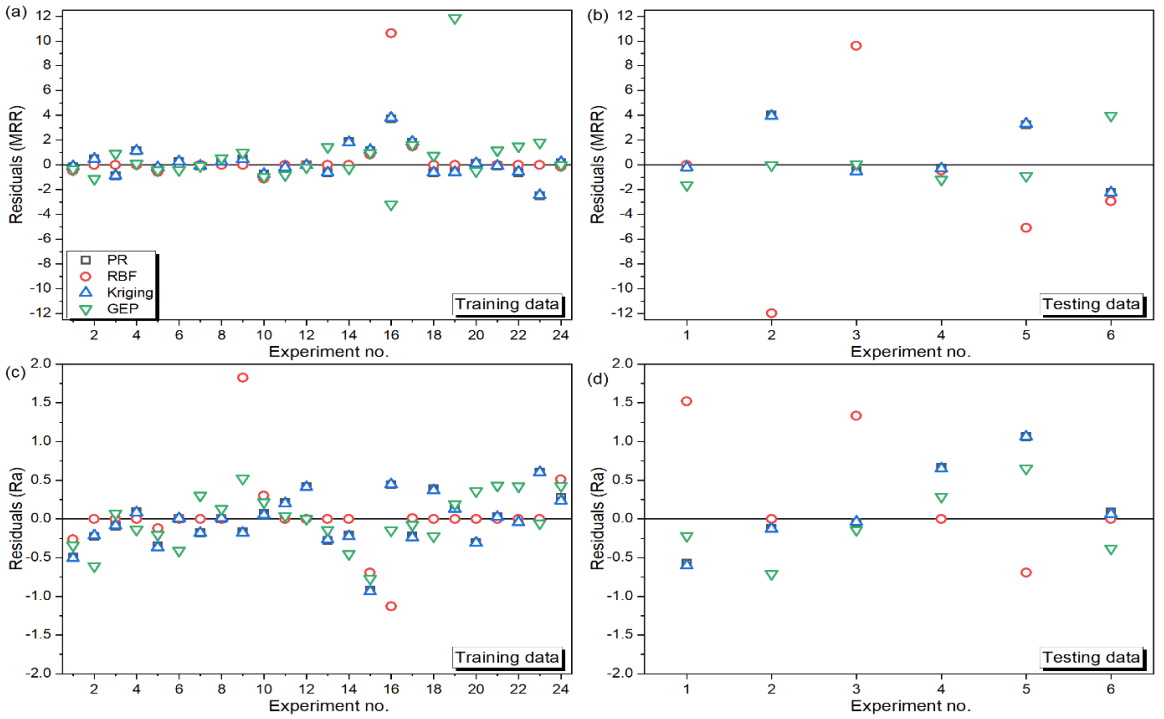

Figure 3.4 Residuals of predicted MRR for (a) training dataset, (b) testing dataset, and residuals of predicted Ra for (c) training dataset, (d) testing dataset.

Now, after critically analyzing Figures 3.2-3.4, it can be unveiled that while predicting the values of MRR, the relative performance of PR, kriging and GEP metamodels is quite satisfactory. For all these three metamodels, the predicted MRR values have extremely low deviations from the corresponding target values. However, the application of RBF metamodel provides average prediction results, with moderate deviations of the predicted MRR values from the target. Similar observations are also noticed for Ra response. However, just by visualizing the above figures, it is quite difficult to determine which of the developed metamodels has the best prediction performance for the considered example. For this purpose, values of four model accuracy metrics in the form of R^2, R^2_{adj}, RMSE and RRMSE are computed, as exhibited in Table 3.4. It is worthwhile to mention here that for any of the predictive models, higher values of R^2 and R^2_{adj}, and lower values of RMSE and RRMSE are always desirable [47].

Table 3.4 Values of model accuracy metrics for both the responses

Response	Metamodel	Dataset	R^2	R^2_{adj}	RMSE	RRMSE
MRR	PR	Testing	0.9298	0.915	1.661	0.1392
		Training	0.9958	0.995	0.9491	0.0663
		Overall	0.9878	0.9859	1.2139	0.0886
	RBF	Testing	0.3299	0.1888	2.9193	0.5625
		Training	0.999	0.9988	0.6653	0.0464
		Overall	0.9129	0.899	1.9874	0.1369
	Kriging	Testing	0.9296	0.91481	1.663	0.1396
		Training	0.996	0.9951	0.9423	0.0658
		Overall	0.9879	0.986	1.2143	0.0886
	GEP	Testing	0.9821	0.9783	1.1261	0.1122
		Training	0.9901	0.9881	1.1185	0.0833
		Overall	0.9894	0.9877	1.1741	0.0878
Ra	PR	Testing	0.9518	0.9417	0.7523	0.2666
		Training	0.9698	0.9635	0.576	0.2995
		Overall	0.9667	0.9614	0.6245	0.2904
	RBF	Testing	0.7687	0.72	1.1137	0.4261
		Training	0.9904	0.9883	0.4326	0.2226
		Overall	0.9267	0.915	0.7612	0.3654

	Kriging	Testing	0.9511	0.9408	0.7572	0.2693
		Training	0.9703	0.964	0.5739	0.2986
		Overall	0.9666	0.9613	0.6251	0.2908
	GEP	Testing	0.9717	0.9657	0.5610	0.2585
		Training	0.9706	0.9644	0.6136	0.3
		Overall	0.9709	0.9662	0.6040	0.295

It can be noticed from Table 3.4 that in case of MRR, RBF metamodel has the best values of R^2 and R^2_{adj} as 0.999 and 0.9988 respectively on training dataset. The corresponding values of RMSE and RRMSE as 0.6653 and 0.0464 respectively also validate the superior performance of RBF metamodel based on the training dataset. However, considering the performance with respect to the test dataset, GEP has the maximum values of R^2 and R^2_{adj} as 0.9821 and 0.9783 respectively. For the same test data, it has also the minimum RMSE and RRMSE values as 1.1261 and 0.1122 respectively. Prediction of MRR based on the overall dataset (training as well as testing) reveals that GEP is the best metamodel having the maximum R^2 (0.9894) and R^2_{adj} (0.9877), and minimum RMSE (1.1741) and RRMSE (0.0878) values. Furthermore, based on the overall dataset, RBF metamodel exhibits the least R^2 (0.9129) and R^2_{adj} (0.899), and the worst RMSE (1.9874) and RRMSE (0.1369) values. It can be concluded that kriging and PR metamodels rank second and third respectively in predicting MRR values on the overall dataset.

While predicting Ra values for the considered EDM process, similar results can also be noticed. The RBF has the best accuracy on the training dataset having the maximum R^2 and R^2_{adj}, and minimum RMSE and RRMSE values. But for prediction of Ra values using the test data, GEP emerges out as the most accurate metamodel with the maximum R^2 (0.9717) and R^2_{adj} (0.9657), and minimum RMSE (0.561) and RRMSE (0.2585) values. On the overall dataset, GEP also shows the best performance with respect to all the model accuracy metrics, followed by PR and kriging metamodels. The RBF shows the worst values of all the four metrics on overall dataset. Thus, the performnace of RBF is noticed to be quite inconsistent for both MRR and Ra responses. Interestingly, it exhibits good accuracy with training datasets, but its prediction accuracy deteriorates with the testing and overall datasets. In contrast, GEP has consistent accuracy level on the overall datasets for both the responses under consideration.

For intuitive assessment of the metamodels, Taylor diagram depicting the standard deviation, correlation coefficient and root mean square deviation (RMSD) of the metamodels is presented, and contrasted against their experimental results in Figure 6. It can be revealed from Figure 3.5(a) that the performance of all the metamodels is quite similar on training data.

However, on testing data (Figure 3.5(b)), the RMSD of RBF metamodel is observed to be more than twice that of the other metamodels. The performance of PR and kriging metamodels is noticed to be almost similar. In Figure 3.5(c), for Ra, all the metamodels show very high correlation with the experimental data, but the RMSD of RBF is slightly higher than the rest. For Ra testing data (Figure 3.5(d)), the lowest RMSD is observed for GEP metamodel. Thus, based on all the model accuracy metrics, and both testing and overall datasets, GEP evolves out as the best metamodel for almost accurately predicting MRR as well as Ra values for the EDM process under consideration.

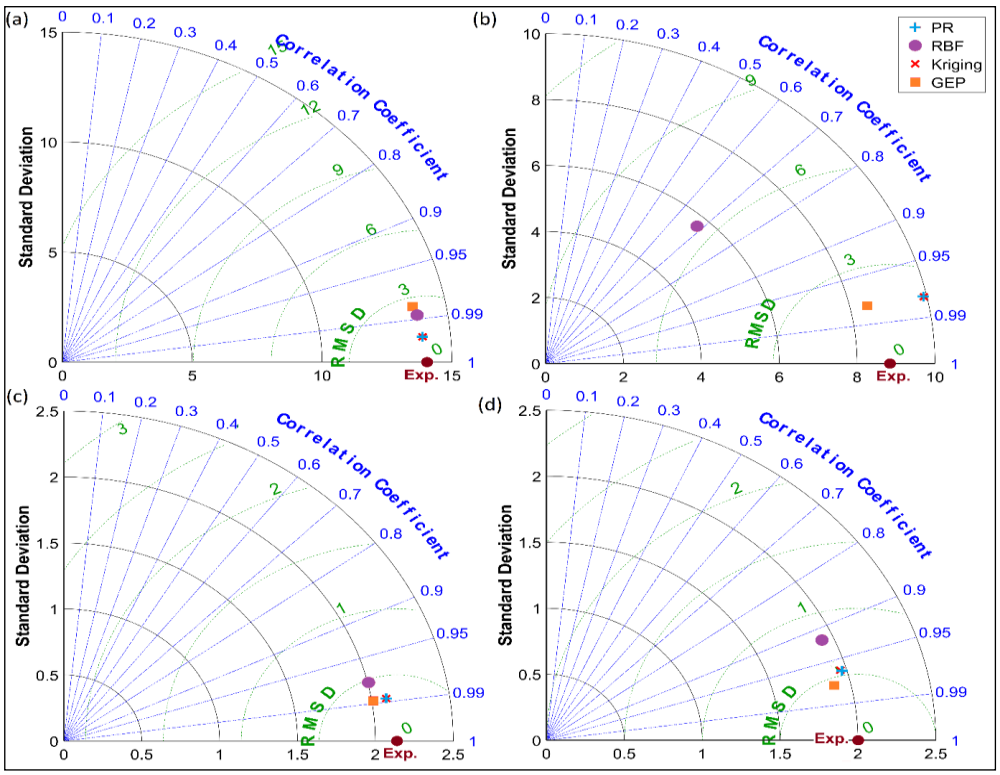

Figure 3.5 Taylor diagram for standard deviation, correlation coefficient and RMSD of the metamodels for MRR (a) training dataset (b) testing dataset, and for Ra (c) training dataset (d) testing dataset

4. **Prediction of electric discharge turning (EDT) responses using different neural networks:**

It has already been mentioned that the basic objective of this paper focuses on applications of five NN models, i.e. FNN, CNN, RNN, LSTM and GRNN for predicting the responses of an EDT process, and comparing their prediction performance using four statistical error metrics. For this purpose, the experimental dataset of Jadidi et al. [5] is considered here. On a TEHRAN EKRAM machine modified by a spindle to perform turning operation to produce sharp edge grooves, Jadidi et al.[48] performed 81 experiments to study the effects of magnetic field (M) (in T), pulse current (Ip) (in A), pulse duration (Ton) (in μs) and angular velocity (N) (in rpm) on volumetric MRR (in mm^3/min) and OC (in μm). The EDT operation was performed on round bars of AISI D2 alloy steel with dimensions of 200 mm in length and 20 mm in diameter. A rectangular-shaped pure copper tool (6 mm thickness, and 30 mm width and length) was employed as the electrode. During the experiments, pulse current was varied between 5 and 15 A, whereas, the range for pulse duration was between 600 and 1000 μs. On the other hand, the magnetic field and angular velocity were varied between 0 and 0.4 T, and 50 and 250 rpm respectively. Table 4.1 depicts the experimental dataset and the measured response values.

Table 4.1 Experimental dataset and measured responses [48]

Exp. No.	M	I	Ton	N	MRR	OC
1	0	5	600	50	231	113
2	0	5	600	150	241	103
3	0	5	600	250	245	88
4	0	5	800	50	308	135
5	0	5	800	150	315	123
6	0	5	800	250	303	105
7	0	5	1000	50	283	163
8	0	5	1000	150	288	149
9	0	5	1000	250	290	127
10	0	10	600	50	300	124
11	0	10	600	150	313	113
12	0	10	600	250	318	97

13	0	10	800	50	400	148
14	0	10	800	150	410	135
15	0	10	800	250	394	115
16	0	10	1000	50	368	179
17	0	10	1000	150	374	164
18	0	10	1000	250	377	140
19	0	15	600	50	346	136
20	0	15	600	150	361	124
21	0	15	600	250	368	106
22	0	15	800	50	462	162
23	0	15	800	150	473	148
24	0	15	800	250	455	126
25	0	15	1000	50	425	196
26	0	15	1000	150	432	179
27	0	15	1000	250	435	152
28	0.2	5	600	50	254	90
29	0.2	5	600	150	265	82
30	0.2	5	600	250	270	70
31	0.2	5	800	50	339	108
32	0.2	5	800	150	347	98
33	0.2	5	800	250	333	84
34	0.2	5	1000	50	311	130
35	0.2	5	1000	150	317	119
36	0.2	5	1000	250	319	102
37	0.2	10	600	50	330	99
38	0.2	10	600	150	344	90
39	0.2	10	600	250	350	78
40	0.2	10	800	50	440	118
41	0.2	10	800	150	451	108
42	0.2	10	800	250	433	92

43	0.2	10	1000	50	405	143
44	0.2	10	1000	150	411	131
45	0.2	10	1000	250	415	112
46	0.2	15	600	50	381	109
47	0.2	15	600	150	397	99
48	0.2	15	600	250	405	85
49	0.2	15	800	50	508	130
50	0.2	15	800	150	520	118
51	0.2	15	800	250	500	101
52	0.2	15	1000	50	468	157
53	0.2	15	1000	150	475	143
54	0.2	15	1000	250	479	122
55	0.4	5	600	50	323	68
56	0.4	5	600	150	337	62
57	0.4	5	600	250	343	53
58	0.4	5	800	50	431	81
59	0.4	5	800	150	441	74
60	0.4	5	800	250	424	63
61	0.4	5	1000	50	396	98
62	0.4	5	1000	150	403	89
63	0.4	5	1000	250	406	76
64	0.4	10	600	50	420	74
65	0.4	10	600	150	438	68
66	0.4	10	600	250	445	58
67	0.4	10	800	50	560	89
68	0.4	10	800	150	574	81
69	0.4	10	800	250	552	69
70	0.4	10	1000	50	515	107
71	0.4	10	1000	150	524	98
72	0.4	10	1000	250	528	84

73	0.4	15	600	50	484	82
74	0.4	15	600	150	505	74
75	0.4	15	600	250	515	84
76	0.4	15	800	50	647	97
77	0.4	15	800	150	662	89
78	0.4	15	800	250	637	76
79	0.4	15	1000	50	595	118
80	0.4	15	1000	150	605	107
81	0.4	15	1000	250	609	91

4.1 Data sampling

The application of any of the NN models starts with a set of training data. The efficiency and effectiveness of an NN model entirely depends on how well it has been trained with the appropriate dataset. The training dataset should be so selected that it would adequately highlight all the features of the design space under consideration. Therefore, there should be no bias in selecting a particular dataset. In this paper, among the 81 experimental runs of the EDT process, 65 observations are considered for training and developing all the NN models, and the remaining observations are treated as testing data points to validate the prediction performance of the NN models. It is worthwhile to mention here that the same sets of training and testing data are considered for comparative analysis of the prediction performance of all the NN models.

4.2 Model architecture

For each NN model under consideration, the corresponding model architecture is built with different types and number of layers having varying number of nodes. Although it is quite obvious that a greater number of layers and more nodes in each layer would eventually increase the model accuracy, but every NN architecture is developed in this paper keeping in mind the optimal computational effort. For having an unbiased comparison among the NN models, their architectures are kept the same for both the responses (MRR and OC).

4.2.1 FNN

For developing a predictive FNN model, a sequence of one input layer, two dense layers and one output layer is taken. Dense layers are the hidden layers with 100 and 30 nodes respectively. Each layer is activated by the rectified linear activation unit (ReLU) function, which can be mathematically expressed using Eq. (4.1) where y is the output of ReLU and x is the input

to that function. During compilation, adaptive moment estimation (AdaM) optimization process is considered with mean square error (MSE) as the loss function. Adam optimizer involves a combination of two gradient descent methodologies, one is momentum, which takes into consideration the 'exponentially weighted average' and accelerates the gradient descent, and another one is root mean square propagation, which implement the concept of decaying or exponential moving average of partial gradients . The FNN model is run over 5000 epochs. Figure 4.1 shows the architectural diagram of the developed FNN model.

$$y = \max(0, x) \rightarrow \frac{\partial f}{\partial x} = 1 \text{ if } x \geq 0; \text{ else } 0 \tag{4.1}$$

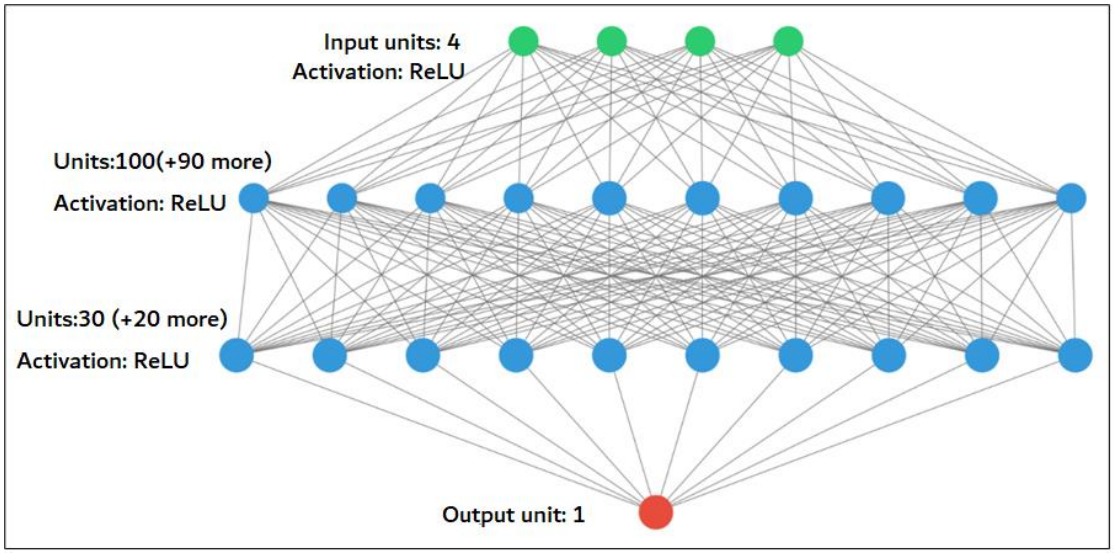

Figure 4.1 Developed FNN architecture for training

4.2.2 CNN

The CNN architecture also consists of one input layer, two hidden layers and one output layer. Between the two hidden layers, one is convolutional one-dimensional (Conv-1D) layer with 100 nodes. The Conv-1D layer is activated using ReLU function. Another hidden layer is a flattened layer, which converts the 100 output arrays coming from the nodes of Conv-1D layer into a single one-dimensional array. The AdaM optimizer is adopted with MSE as the loss function during the compilation process. This model is run over 5000 epochs. Figure 4.2 depicts the architectural diagram of the CNN model.

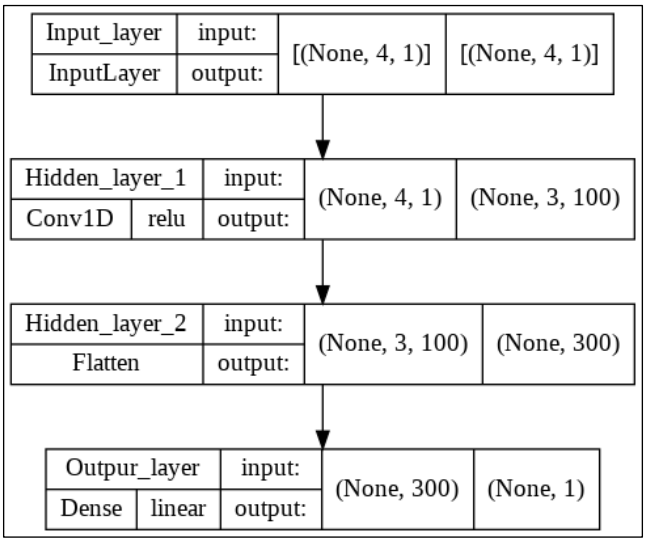

Figure 4.2 Developed CNN architecture

4.2.3 RNN

The RNN architecture is built with one input layer, one hidden layer and one output layer. The hidden layer is a simple RNN layer with 100 nodes. The RNN layer is activated by sigmoid activation function which can be represented using Eq. (4.2) where $S(x)$ denotes the sigmoid function and x is the input to that function. During model training, MSE is treated as the loss function and AdaM as the optimizer. This model is also trained over 5000 epochs. The architectural diagram of RNN model is portrayed in Figure 4.3.

$$S(x)=\frac{1}{1+e^{-x}} \tag{4.2}$$

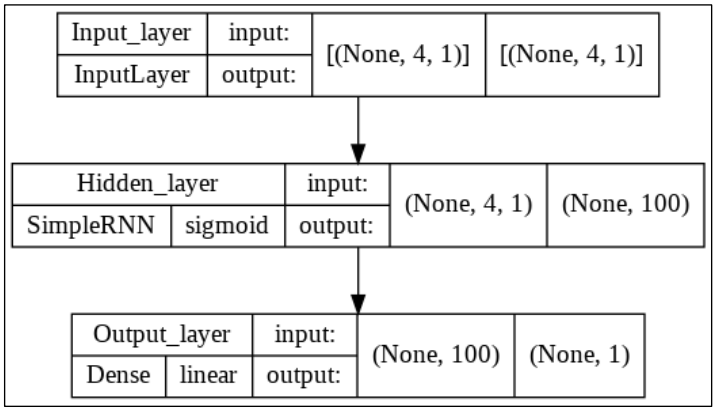

Figure 4.3 RNN architecture for model training

4.2.4 LSTM

For developing the corresponding LSTM model, a sequence of one input, one hidden and one output layers is considered in this paper. The hidden layer is an LSTM layer with 100 nodes. Figure 4.4 shows the developed LSTM architecture. The hidden layer is activated by tanh function. During compilation, AdaM optimization process is taken into account along with mean square error (MSE) as the loss function. This model is also run over 5000 epochs.

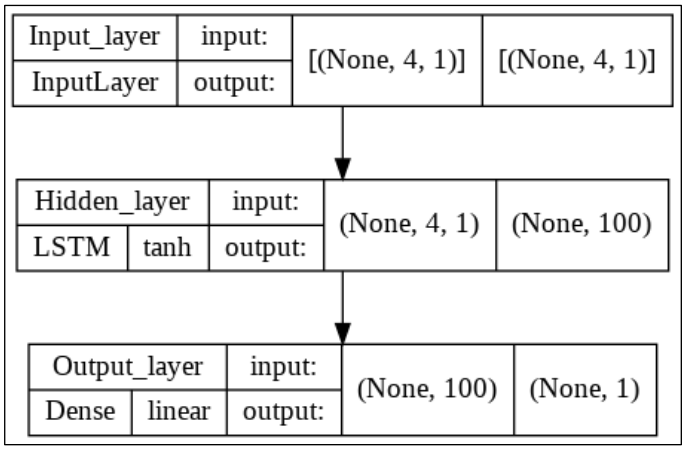

Figure 4.4 Developed LSTM architecture

4.2.5 GRNN

The developed GRNN model consists of four layers, i.e. one input layer, one pattern layer and one summation layer which is treated as an output layer also. In the pattern layer, which is based on RBF kernel, 100 nodes are taken. The bandwidth standard deviation parameter for the kernel is treated as 5. Summation layer has 100 neurons. Gradient search approach is employed to minimize the loss function and to find out the local minimum of the cost function, limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS-B) algorithm is adopted. In Figure 4.5, the developed GRNN architecture is presented.

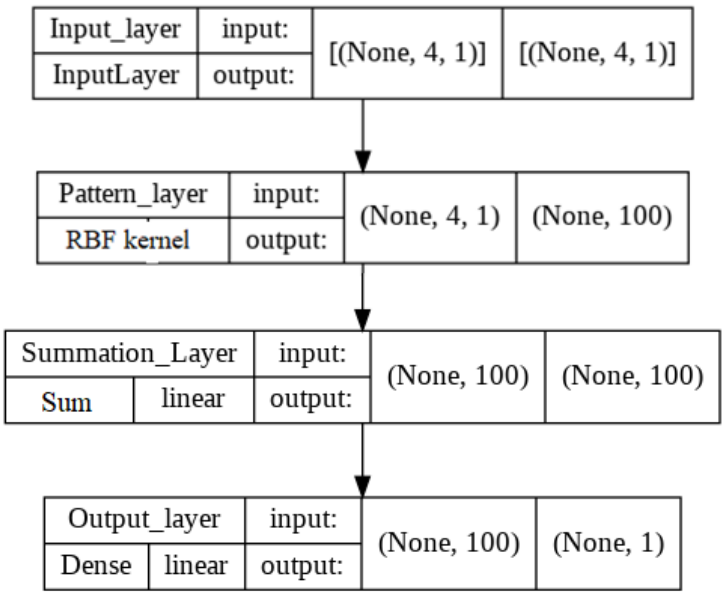

Figure 4.5 GRNN architecture for training

4.3 Prediction performance analysis

4.3.1 Performance using the experimental dataset

All the developed NN models with the specific architectures and parameters, mentioned in Sub-section 4.2, are now trained on the past experimental dataset over the pre-defined epochs. After training, the corresponding predicted values of MRR and OC responses are obtained, as shown in Tables 4.2-4.3 respectively. In Figure 4.6, the corresponding scatter plots between the target and predicted MRR values for both the training and testing datasets are provided. This figure reveals that for MRR, there are excellent agreements between the target and predicted values for both the training and testing data. Based on Figure 11, it can be unveiled that for prediction of MRR, LSTM has the best results as all the target versus predicted data points are nearly positioned along the diagonal identity line. Almost all the MRR values predicted by different NN models are within the ±20% error bounds with respect to the actual. To further analyze the prediction pattern of various NNs, the normality of residuals is analyzed in Figure 4.7. The residuals of FNN, CNN and LSTM appear to follow normality assumption better than the other two NN models. The residuals of RNN are left skewed indicating that it is more likely to overpredict the response. This is also evident from the comparison of the mean (μ) of the residuals. The GRNN has unusually higher number of residuals in the near zero zone indicating that it has most probably 'memorized' some of the data. This has led to extremely high residuals in the 'non-memorized' data points causing extreme deviation of the residuals from normality.

Nevertheless, for MRR, FNN appears to have the worst prediction performance among all the considered NN models.

In Figure 4.8, the target versus predicted values of OC are presented for both the training and testing data. Here too, except GRNN, the OC values predicted by all other NN models are found to be within the ±20% error bounds. Further, analysis of the NN models is carried out by inspecting the normality of residuals plots in Figure 4.9. Although presence of outliers is detected in all the NN models, RNN is the most left-skewed. The GRNN model is found to have extreme deviation from the normality assumption. On the other hand, although, the residual pattern for LSTM is observed to be non-normal, it is interesting to note that by removing only one outlier, the model can achieve normality. It is worth mentioning here that the classical texts [47], [49] on normality of residuals of prediction models state that 5% extreme outliers can be removed from the data to evaluate approximate adherence to normality assumption. However, just by visualizing these scatter plots and normal probability plots of residuals, it is quite unjustifiable to single out the best NN model for predicting the EDT responses. For this reason, values of the corresponding statistical error metrics are computed and analyzed.

Table 4.2 Target and predicted MRR values

	Target	FNN	CNN	RNN	GRNN	LSTM
1	231	241.5835	256.1749	243.1462	231.0001	240.6476
2	241	243.2697	255.6496	250.2661	241	245.4606
3	245	241.2878	251.8766	258.6545	245.0002	243.8911
4	308	261.4585	301.3728	315.7292	308.0002	304.009
5*	315	263.0573	301.5704	324.214	315.0002	302.8106
6	303	289.2599	307.0701	321.3434	303.0002	303.0361
7	283	282.3422	289.3399	292.0892	283.0002	282.7749
8	288	282.9322	289.632	300.4781	288.0002	284.9379
9	290	297.2757	289.9241	301.8713	290.0002	286.7987
10*	300	315.8103	329.5225	307.056	300	301.5501
11	313	325.6764	328.9971	320.1007	315.5171	315.037
12	318	308.837	325.2262	329.1774	318	323.7809
13	400	350.7343	379.6528	426.1958	399.9999	395.796
14	410	352.3331	379.9449	431.6697	409.9999	407.1128
15*	394	365.5308	389.6679	416.9841	394	404.1015

16	368	370.6091	371.1419	406.6893	368	364.8444
17	374	372.2079	371.434	418.5018	374	375.2183
18	377	373.8069	371.7261	412.6671	377	371.9635
19	346	337.5731	350.2021	342.6192	346	345.3936
20*	361	334.0365	349.6768	359.1476	361	359.4715
21	368	362.1772	345.9062	367.8444	367.9999	361.6707
22	462	440.0101	444.9847	469.2689	462	451.8853
23	473	441.6089	445.2768	472.1294	473	465.158
24	455	443.2077	454.0099	458.4851	455	462.3222
25*	425	459.8849	440.5154	434.6253	425	422.8334
26	432	461.4838	440.8075	443.1791	432	432.9521
27	435	463.0826	441.0996	438.8657	434.9999	434.8688
28	254	289.4057	298.6721	278.1001	277	256.4423
29	265	291.0046	298.1467	290.4476	265.0002	266.3912
30*	270	272.2703	294.377	300.5333	270.0001	268.824
31	339	309.2806	346.4592	377.103	339.0001	337.0804
32	347	310.8795	346.4688	385.7106	315.0007	341.7346
33	333	328.4267	350.8888	373.2826	333.0003	338.134
34	311	329.1555	334.6184	331.4263	311.0003	312.8942
35*	317	330.7544	335.3602	342.7874	367.0552	319.8819
36	319	332.3532	336.1021	337.1532	319.0001	317.415
37	330	377.9568	372.0195	332.7023	340.1118	329.098
38	344	386.0287	371.4942	348.2987	344	343.7427
39	350	356.5101	367.9689	357.0338	344.8068	349.3522
40*	440	398.5565	425.6302	456.0275	452.0208	430.1285
41	451	400.1553	426.372	459.2618	501.2143	443.736
42	433	403.8371	435.0461	444.7231	433.0001	440.165
43	405	418.4314	419.5776	427.6633	404.9999	396.7093
44	411	420.0302	420.3194	436.9309	411.0003	407.1244
45*	415	421.629	421.0613	431.9099	415	405.2103

46	381	396.0037	393.2519	382.4686	381.0001	378.2838
47	397	402.6884	392.7401	399.4798	403.7114	388.0857
48	405	408.2534	389.2522	403.3632	368.0001	392.5266
49	508	506.5971	496.1371	514.2295	508.0002	497.24
50*	520	504.0511	496.4657	518.168	519.9999	514.4553
51	500	501.5051	510.0148	507.356	500.0001	506.8848
52	468	507.7071	492.6698	486.2296	468	464.4308
53	475	509.306	493.0873	490.7864	475	476.0628
54	479	510.9048	493.5048	488.2872	486.7877	480.8539
55*	323	408.3556	378.6997	329.9644	323	331.9796
56	337	405.8097	378.2998	345.209	337.0001	337.9277
57	343	366.3757	336.2568	352.0139	343.0001	341.9521
58	431	409.1899	421.2308	448.0037	431.0001	420.8405
59	441	406.6441	421.5127	451.2734	574	435.2151
60*	424	408.406	426.3244	436.1522	424.0001	442.2969
61	396	410.0244	411.8985	426.0116	395.9998	387.9447
62	403	407.4785	412.6403	434.9962	403.0002	399.2752
63	406	404.9326	413.3821	429.1181	422.4982	404.8684
64	420	463.0616	490.8338	433.0352	403.5	415.0911
65*	438	471.1336	490.4475	451.2133	437.9998	426.8229
66	445	453.5775	437.8876	447.585	444.9998	434.9218
67	560	517.6382	545.4392	560.531	559.9999	554.272
68	574	515.0923	545.8732	569.3662	574	571.9506
69	552	514.7343	559.6365	559.3177	551.9997	556.8716
70*	515	518.4727	527.3013	541.0726	464.9464	505.5234
71	524	515.9268	527.7189	546.7112	523.9997	524.5939
72	528	513.3809	528.1364	542.8898	527.9999	521.7757
73	484	479.7213	505.6108	498.6324	483.9998	479.0403
74	505	487.7932	505.2244	519.2001	504.9999	502.5916
75*	515	484.2672	458.06	511.4446	445	509.8736

76	647	599.0868	622.4588	649.5006	646.9996	627.1984
77	662	607.1587	622.9064	656.4434	546.7412	629.381
78	637	614.5477	635.8134	638.0488	636.9996	620.2736
79	595	626.921	618.8296	603.0941	594.9998	603.1805
80*	605	624.3751	619.2471	610.6132	604.9996	615.7647
81	596.5643	621.8291	619.6647	600.5006	591.342	604.7985

Table 4.3 Target and predicted OC values

	Target	FNN	CNN	RNN	GRNN	LSTM
1	113	119.1732	117.1817	118.7687	113	113.3572
2	103	105.4071	102.9344	103.5098	103	103.6932
3	88	91.64101	93.2215	94.12621	87.99996	88.23135
4	135	139.1878	136.5541	142.8677	134.9999	134.5466
5*	123	125.4217	123.9434	127.6086	123	123.2369
6	105	111.6556	108.8373	112.4053	105	104.6459
7	163	159.2023	155.9231	166.9667	162.9999	163.1688
8	149	145.4362	144.7093	151.7076	149	147.9738
9	127	131.6701	128.7576	136.4485	126.9999	126.6402
10*	124	129.0418	128.8575	130.5592	124	124.2958
11	113	115.2757	114.6102	115.3001	90	114.0981
12	97	101.5096	104.8973	101.7745	97	96.91842
13	148	149.0564	149.4253	154.6581	148	148.2501
14	135	135.2902	136.8145	139.3991	135	135.4655
15*	115	121.5241	121.7084	124.14	115	116.2149
16	179	169.0709	168.9024	178.7593	178.9999	175.2614
17	164	155.3048	157.6887	163.498	163.9999	164.646
18	140	141.5387	141.737	148.2388	139.9999	139.8717
19	136	138.9104	138.9019	142.3496	135.9999	136.4606
20*	124	125.1443	124.6546	127.0905	124	124.7663

21	106	111.3782	114.9416	111.8516	106	106.9143
22	162	158.9249	159.375	166.4484	161.9999	162.7376
23	148	145.1588	146.7642	151.1895	147.9999	147.8719
24	126	131.3927	131.6581	135.9305	125.9999	127.0482
25*	196	178.9395	180.0246	190.5519	195.9999	179.142
26	179	165.1734	168.8108	175.2884	178.9999	175.6008
27	152	151.4073	152.8592	160.0293	152	152.4515
28	90	92.16597	89.01103	88.83662	90.5	89.6367
29	82	78.39986	74.76373	73.57761	82	83.08086
30*	70	64.63374	65.05081	68.69128	70	72.38254
31	108	112.1805	108.3834	112.9356	108	107.5831
32	98	98.4144	95.77267	97.67657	123	98.93671
33	84	84.64829	80.66657	83.15662	84	84.47815
34	130	132.1951	127.7524	137.0354	130	129.0288
35*	119	118.429	116.5387	121.7754	119	118.8686
36	102	104.6628	100.587	106.5164	102.0001	100.5181
37	99	102.0345	100.5881	100.627	124	98.36054
38	90	88.26843	86.34076	85.36806	89.99996	90.67656
39	78	74.50232	76.62785	76.33949	77.5	78.28259
40*	118	122.0491	121.1558	124.726	118.5	117.8336
41	108	108.283	108.545	109.4669	108	108.085
42	92	94.51688	93.43893	94.20787	92	91.58872
43	143	142.0636	140.6329	148.8281	143.0001	141.5845
44	131	128.2975	129.4192	133.5659	131	129.6714
45*	112	114.5314	113.4675	118.3068	112	110.5894
46	109	111.9031	110.6315	112.4175	109	108.7342
47	99	98.13701	96.38415	97.15841	99	99.78037
48	85	84.3709	86.67123	83.98773	106	84.92549
49	130	131.9177	131.1046	136.5165	130	129.5996
50*	118	118.1516	118.4938	121.2573	118.0001	118.6681

51	101	104.3855	103.3877	105.9983	101	100.6499
52	157	151.9322	151.7542	160.6207	157	157.5365
53	143	138.1661	140.5404	145.3563	143	142.584
54	122	124.4	124.5888	130.0972	121.5	122.0401
55*	68	65.1587	64.97232	63.2821	68	68.77152
56	62	51.39259	50.72502	55.3672	62.00004	64.75293
57	53	37.62647	41.0121	51.83048	53.00003	56.00575
58	81	85.17326	84.3447	83.00337	81.00005	81.20598
59	74	71.40714	71.73396	74.88757	83.84985	75.80059
60*	63	57.64103	56.62787	68.40468	63.00004	65.40933
61	98	105.1878	103.7137	107.1042	98.00006	98.43254
62	89	91.42169	92.49994	94.40794	89	90.66702
63	76	77.65557	76.54825	86.49302	102	77.99039
64	74	75.02728	72.74953	70.69492	76.72082	73.79083
65*	68	61.26116	58.50222	60.12483	68.00004	69.19202
66	58	47.49505	48.7893	55.12465	58	59.20315
67	89	95.04182	93.31722	94.79379	89	88.2756
68	81	81.27572	80.70649	79.6452	81	81.74551
69	69	67.5096	65.6004	71.82796	69.00005	70.6964
70*	107	115.0564	112.7944	118.8968	143	107.473
71	98	101.2903	101.5807	103.6338	98.00007	98.45018
72	84	87.52415	85.629	91.25066	84.00006	83.30798
73	82	84.89585	82.36105	82.48532	82.00005	81.37911
74	74	71.12974	68.11375	67.22626	74	75.45749
75*	84	57.36362	58.40083	58.55273	61.25697	64.85317
76	97	104.9104	102.8342	106.5853	97.00007	96.9202
77	89	91.14429	90.22343	91.32515	118	89.96214
78	76	77.37817	75.11734	76.48792	76.00005	77.01233
79	118	124.925	123.4838	130.6894	118.0001	117.432
80*	107	111.1588	112.27	115.4242	107.0001	108.1045

81	91	97.39272	96.31836	100.1652	91	90.71896

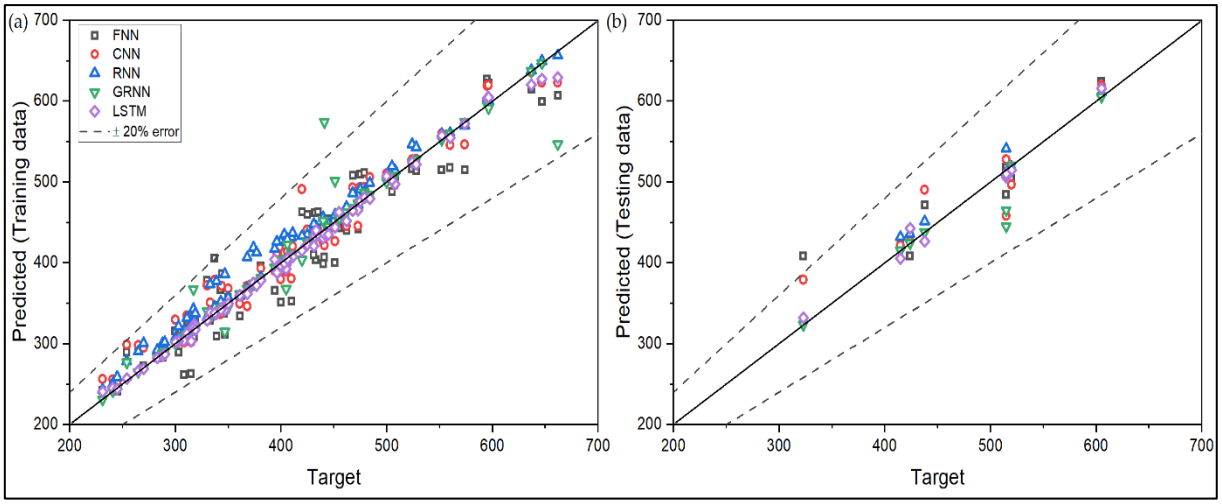

Figure 4.6 Scatter plots between target and predicted MRR values

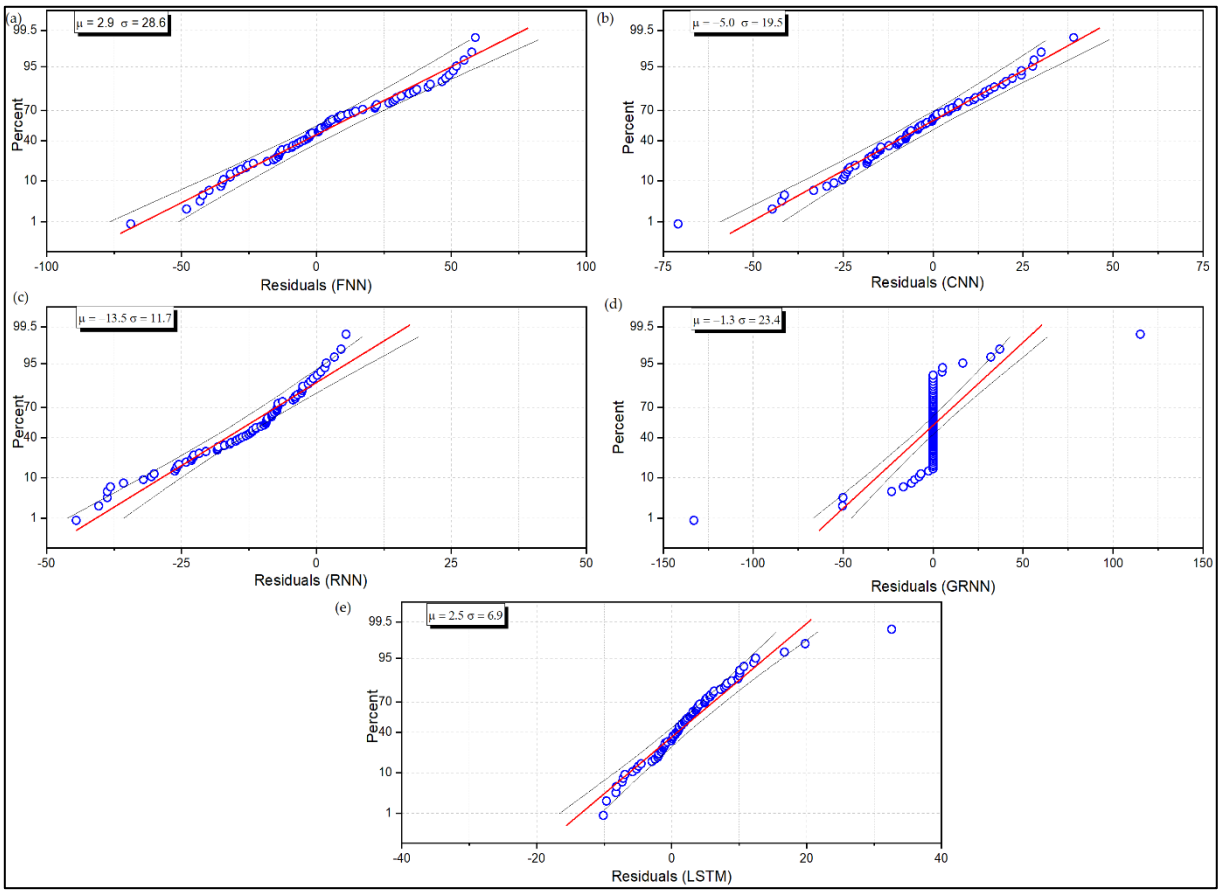

Figure 4.7 Normal probability plots of residuals for MRR

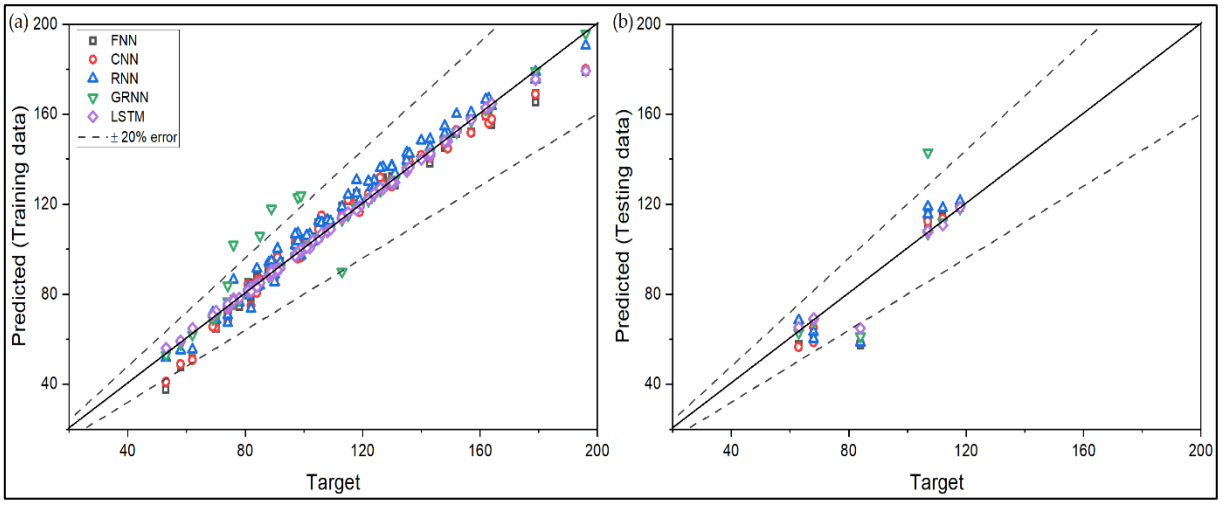

Figure 4.8 Scatter plots the target and predicted OC values

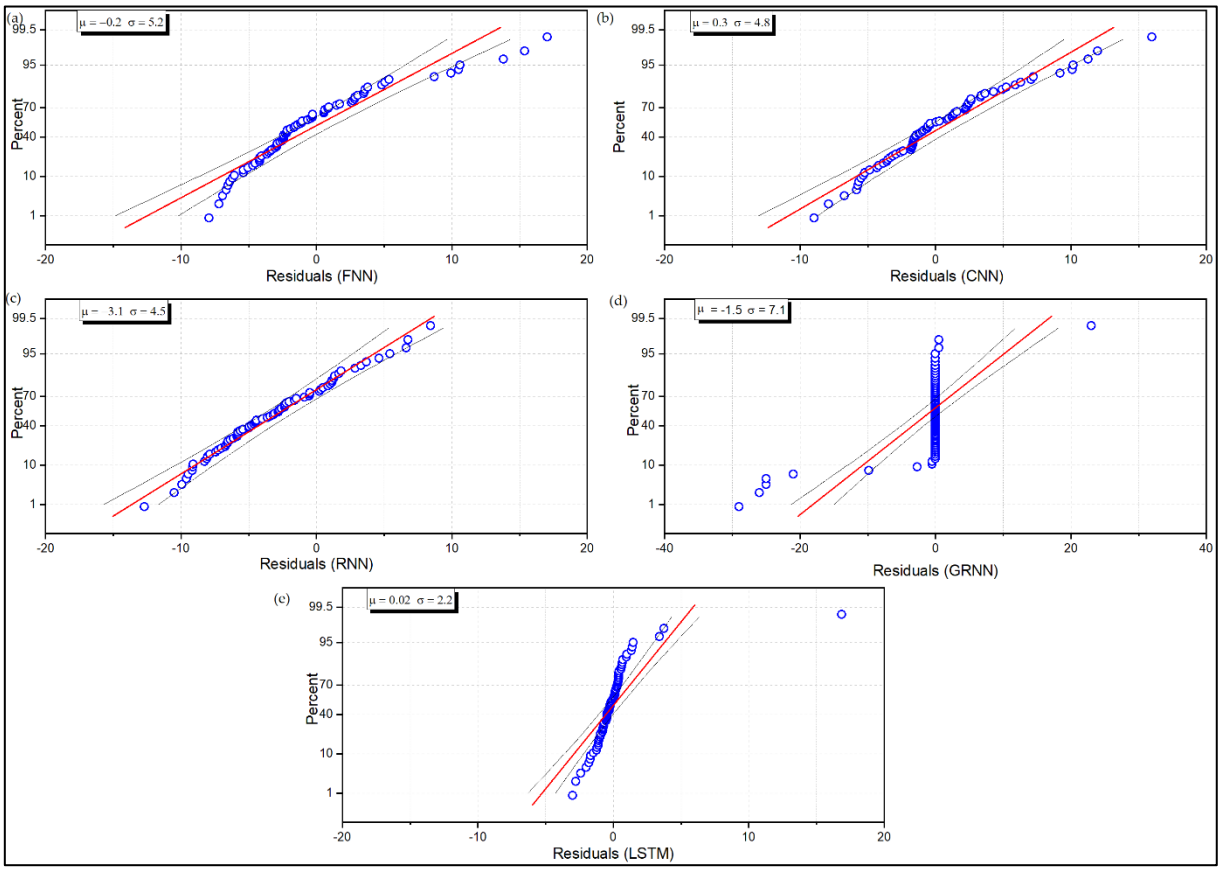

Figure 4.9 Normal probability plots of residuals for OC

4.3.2 Performance analysis based on statistical error metrics

In order to further validate the prediction performance of the five NN models, R^2, R^2_{adj}, RMSE and RRMSE values for the training, testing and overall datasets are computed, as provided

in Table 4.4 and Figure 4.10. It is worthwhile to mention here that for the best NN model, maximum R^2 and R^2_{adj}, and minimum RMSE and RRMSE values are always recommended.

Table 4.4 Calculated values of different statistical error metrics for MRR and OC

Response	Model	Dataset	R^2	R^2_{adj}	RMSE	RRMSE
MRR	FNN	Test	0.86652	0.817982	5.781077	0.059413545
		Train	0.923498	0.918397	5.324524	0.05442662
		Overall	0.914158	0.90964	5.424152	0.055849182
	CNN	Test	0.905718	0.871433	5.299829	0.060108586
		Train	0.960906	0.9583	4.501822	0.046077734
		Overall	0.95186	0.949326	4.693894	0.049220299
	RNN	Test	0.970145	0.959289	3.975655	0.04343048
		Train	0.969533	0.967501	4.229813	0.042370989
		Overall	0.969633	0.968035	4.18319	0.042819783
	GRNN	Test	0.924896	0.897586	5.00691	0.058453518
		Train	0.945483	0.941849	4.892095	0.047865163
		Overall	0.942109	0.939062	4.915418	0.049818196
	LSTM	Test	0.990544	0.987105	2.98253	0.031526761
		Train	0.994819	0.994474	2.716194	0.026974513
		Overall	0.994118	0.993809	2.775115	0.028046963
OC	FNN	Test	0.919772	0.890598	2.997723	0.087378542
		Train	0.973504	0.971738	2.225383	0.073491219
		Overall	0.962171	0.96018	2.443214	0.079003216
	CNN	Test	0.923956	0.896304	2.957849	0.085732075
		Train	0.977724	0.976239	2.130928	0.069605591
		Overall	0.966382	0.964613	2.372183	0.075974076
	RNN	Test	0.921414	0.892837	2.982266	0.083574451
		Train	0.96744	0.96527	2.343048	0.072460485
		Overall	0.957736	0.955511	2.511881	0.076549424
	GRNN	Test	0.887396	0.846448	3.262865	0.091782641
		Train	0.936141	0.931884	2.772783	0.092196109

		Overall	0.925873	0.921971	2.890678	0.093227675
	LSTM	Test	0.958372	0.943234	2.544235	0.082863211
		Train	0.99863	0.998538	1.061218	0.035312806
		Overall	0.990132	0.989613	1.746054	0.058172876

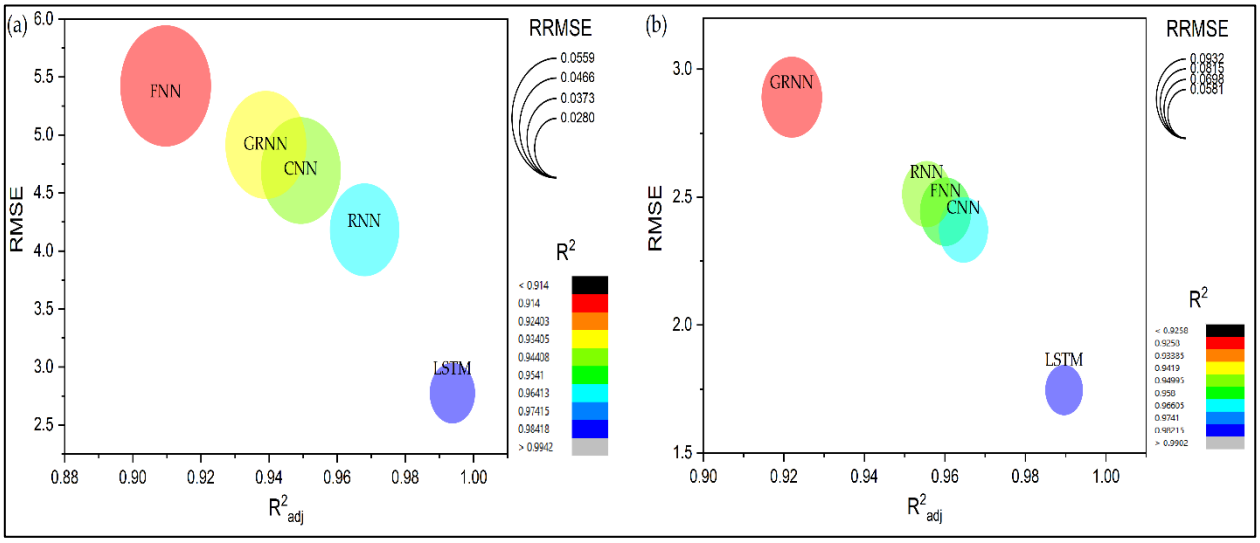

Figure 4.10 Prediction performance of the neural networks for (a) MRR (b) OC [Color and size of the bubbles represent R^2 and RRMSE respectively]

As observed from Table 4.4 and Figure 4.10, for MRR, LSTM has the best prediction performance with the highest R^2 and R^2_{adj} values of 0.9948 and 0.9945 respectively based on the training dataset. The corresponding lowest values of RMSE and RRMSE as 2.7161 and 0.028 respectively also reassure the same observation. The LSTM is also proven to be the best for prediction of MRR using the testing dataset with R^2, R^2_{adj}, RMSE and RRMSE values as 0.9905, 0.9871, 2.9825 and 0.0315 respectively. When both the training and testing data sets are considered together, LSTM again emerges outs as the best performing NN model with the corresponding R^2, R^2_{adj}, RMSE and RRMSE values as 0.9941, 0.9938, 2.7751 and 0.0280 respectively. Thus, it performs excellently for all the datasets. Based on R^2 for prediction of MRR, RNN occupies the second position having a value of 0.9701 for the testing dataset, followed by GRNN, CNN and FNN. This same ranking of the NN models can also be validated with respect to R^2_{adj} and RMSE values. But when RRMSE values are considered, FNN performs marginally better that CNN, although the prediction performance of RNN and GRNN remains unaltered. Thus, the NN models can be ranked as LSTM-RNN-CNN-GRNN-FNN with respect to their overall prediction performance of MRR, as clearly noticed from Figure 4.10(a).

For prediction of OC, similar results are also observed. The LSTM has the best prediction accuracy on training data having the maximum R^2 (0.9986), R^2_{adj} (0.9985), and minimum RMSE (1.0612) and RRMSE (0.0353) values. On testing data, LSTM also shows the best prediction performance having R^2, R^2_{adj}, RMSE and RRMSE values as 0.9584, 0.9432, 2.5442 and 0.0829 respectively. With respect to the overall data, its perfomance is also the best with R^2, R^2_{adj}, RMSE and RRMSE values as 0.9901, 0.9896, 1.7460 and 0.0582 respectively. However, the ranking order of the other NN models is differenet from that obtained for MRR. On the basis of R^2 values for prediction of OC on the testing data, CNN occupies the second position with a R^2 value of 0.9239, followed by RNN, FNN and GRNN having R^2 values as 0.9214, 0.9198 and 0.8874 respectively. The values of R^2_{adj} and RMSE also validate the same results. But, based on RRMSE, RNN occupies the second position with a value of 0.0835, followed by CNN, FNN and GRNN having RRMSE values as 0.0857, 0.0874 and 0.0918 respectively. However, when the prediction performance of all the five NN models is evaluated using the overall data, their ranking order can be derived as LSTM-CNN-RNN-FNN_GRNN. For both MRR and OC responses, LSTM thus has the best prediction performance based on the considered EDT experimental dataset.

Conclusion:

This thesis attempts to develop four metamodels (PR, RBF, kriging and GEP) and five neural networks (FNN, CNN, RNN, RNN-LSTM, GRNN) for accurately predicting different responses values of EDM and EDT processes respectively. For the first EDM process, pulse current, pulse rate, duty cycle and voltage are treated as the input parameters. On the other hand, magnetic field, pulse current, pulse duration and angular velocity are the EDT parameters for the second analysis. Based on these experimental datasets, the prediction performance of the four developed metamodels and five neural network architectures are contrasted, i.e. R^2, R^2_{adj}, RMSE and RRMSE. The detailed comparative analysis on overall performance of the metamodels draws the following conclusions:

a) For metamodeling of EDM processes, GEP emerges out as the best metamodel with maximum R^2 and R^2_{adj}, and minimum RMSE and RRMSE values in almost accurately envisaging all the responses under consideration. GEP provides consistent results for prediction on both the training and testing datasets. Furthermore, although RBF has excellent prediction performance on training datasets, but it has poor accuracy on the test data, leading to overfitting of the model.

b) Among the neural network models of EDT for both MRR and OC, LSTM emerges out as the best performing NN model with the maximum R_2 and R_2^{adj}, and minimum RMSE and RRMSE values. LSTM incorporates both long term and short term memory to store most repetitive characteristics during training. This adds more value to backpropagation and can provide more accurate result than simple RNN process. LSTM helps the model to overcome vanishing and exploding gradient problems during training, thus it provides the most reliable prediction results. Furthermore, the prediction performances of CNN and RNN are almost comparable. The FNN, being a basic and random learning NN model, provides a moderately satisfactory result. However, although GRNN provides average prediction result, it is noticed that during training, based in its architecture, GRNN learns the data pattern very quickly. Thus, for achieving quicker prediction results, GRNN may be recommended.

As a future scope, the application potentiality of other metamodels, including random forest regressor, support vector regressor, Naïve Bias regressor etc. and other NN models, like, RNN with gated recurrent unit (GRU), RBFN, modular neural network (MNN) or CNN with LSTM etc. may be explored and their prediction performance may be compared for accurately envisaging the response values of EDM processes. In the two illustrative examples, small datasets consisting of only 30 and 81 experimental observations are respectively considered for validating the prediction performance of all the metamodels. To achieve a better picture, a data repository

containing large set of experimental data may be developed for training and testing of those metamodels.

References:

[1] J. E. Abu Qudeiri, A. Saleh, A. Ziout, A. H. I. Mourad, M. H. Abidi, and A. Elkaseer, "Advanced electric discharge machining of stainless steels: Assessment of the state of the art, gaps and future prospect," *Materials*, vol. 16, no. 6. 2019. doi: 10.3390/ma12060907.

[2] F. Khan, J. Kumar, and T. Soota, "Optimization of EDM process parameter for stainless steel D3," in *Materials Today: Proceedings*, 2019, vol. 25. doi: 10.1016/j.matpr.2019.07.529.

[3] J. E. A. Qudeiri, A. Zaiout, A. H. I. Mourad, M. H. Abidi, and A. Elkaseer, "Principles and Characteristics of Different EDM Processes in Machining Tool and Die Steels," *Applied Sciences (Switzerland)*, vol. 10, no. 6, 2020, doi: 10.3390/app10062082.

[4] D. Gouda, A. Panda, B. K. Nanda, R. Kumar, A. K. Sahoo, and B. C. Routara, "Recently evaluated Electrical Discharge Machining (EDM) process performances: A research perspective," in *Materials Today: Proceedings*, 2021, vol. 44. doi: 10.1016/j.matpr.2020.12.180.

[5] K. H. Ho and S. T. Newman, "State of the art electrical discharge machining (EDM)," *International Journal of Machine Tools and Manufacture*, vol. 43, no. 13, 2003, doi: 10.1016/S0890-6955(03)00162-7.

[6] K. Rajmohan and A. S. Kumar, "Experimental investigation and prediction of optimum process parameters of micro-wire-cut EDM of 2205 DSS," *International Journal of Advanced Manufacturing Technology*, vol. 93, no. 1–4, 2017, doi: 10.1007/s00170-016-8615-3.

[7] E. Evin, M. Tomáš, and J. Kmec, "Optimization of electro-discharge texturing parameters for steel sheets' finishing rollers," *Materials*, vol. 13, no. 5, 2020, doi: 10.3390/ma13051223.

[8] Y. Jia and J. Li, "Impact Analysis of Electrode Material on Electrical Discharge Grinding Polycrystalline Diamond Cutting Tools," in *Procedia CIRP*, 2018, vol. 68. doi: 10.1016/j.procir.2017.12.147.

[9] P. J. Arrazola, T. Özel, D. Umbrello, M. Davies, and I. S. Jawahir, "Recent advances in modelling of metal machining processes," *CIRP Annals - Manufacturing Technology*, vol. 62, no. 2, 2013, doi: 10.1016/j.cirp.2013.05.006.

[10] X. P. Dang, "Constrained multi-objective optimization of EDM process parameters using kriging model and particle swarm algorithm," *Materials and Manufacturing Processes*, vol. 33, no. 4, 2018, doi: 10.1080/10426914.2017.1292037.

[11] K. Kalita, R. K. Ghadai, D. S. Shinde, and X.-Z. Gao, "Data-Driven Genetic Programming-Based Symbolic Regression Metamodels for EDM Process," 2020. doi: 10.4018/978-1-7998-7206-1.ch009.

[12] R. K. Ghadai, K. Kalita, and X. Z. Gao, "Symbolic regression metamodel based multi-response optimization of EDM process," *FME Transactions*, vol. 48, no. 2, 2020, doi: 10.5937/FME2002404G.

[13] M. Ulas, O. Aydur, T. Gurgenc, and C. Ozel, "Surface roughness prediction of machined aluminum alloy with wire electrical discharge machining by different machine learning algorithms," *Journal of Materials Research and Technology*, vol. 9, no. 6, 2020, doi: 10.1016/j.jmrt.2020.08.098.

[14] U. M. R. Paturi, S. Cheruku, V. P. K. Pasunuri, S. Salike, N. S. Reddy, and S. Cheruku, “Machine learning and statistical approach in modeling and optimization of surface roughness in wire electrical discharge machining,” *Machine Learning with Applications*, vol. 6, 2021, doi: 10.1016/j.mlwa.2021.100099.

[15] M.M. Rahman, “Modeling of machining parameters of Ti-6Al-4V for electric discharge machining: A neural network approach,” *Scientific Research and Essays*, vol. 7, no. 8, 2012, doi: 10.5897/sre10.1116.

[16] A. Surleraux, R. Lepert, J. P. Pernot, P. Kerfriden, and S. Bigot, “Machine Learning-based reverse modeling approach for rapid tool shape optimization in die-sinking micro electro discharge machining,” *Journal of Computing and Information Science in Engineering*, vol. 20, no. 3, 2020, doi: 10.1115/1.4045956.

[17] A. Majumder, “Comparative study of three evolutionary algorithms coupled with neural network model for optimization of electric discharge machining process parameters,” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 229, no. 9, 2015, doi: 10.1177/0954405414538960.

[18] X. Zhang, Y. Liu, X. Wu, and Z. Niu, “Intelligent pulse analysis of high-speed electrical discharge machining using different RNNs,” *Journal of Intelligent Manufacturing*, vol. 31, no. 4, 2020, doi: 10.1007/s10845-019-01487-8.

[19] A. K. Sahu and S. S. Mahapatra, “Prediction and optimization of performance measures in electrical discharge machining using rapid prototyping tool electrodes,” *Journal of Intelligent Manufacturing*, vol. 32, no. 8, 2021, doi: 10.1007/s10845-020-01624-8.

[20] H. H. Pourasl, M. Javidani, V. M. Khojastehnezhad, and R. Vatankhah Barenji, “The Performance Prediction of Electrical Discharge Machining of AISI D6 Tool Steel Using ANN and ANFIS Techniques: A Comparative Study,” *Crystals (Basel)*, vol. 12, no. 3, 2022, doi: 10.3390/cryst12030343.

[21] A. Conde, A. Arriandiaga, J. A. Sanchez, E. Portillo, S. Plaza, and I. Cabanes, “High-accuracy wire electrical discharge machining using artificial neural networks and optimization techniques,” *Robotics and Computer-Integrated Manufacturing*, vol. 49, 2018, doi: 10.1016/j.rcim.2017.05.010.

[22] M. K. Pradhan and C. K. Biswas, “Neuro-fuzzy and neural network-based prediction of various responses in electrical discharge machining of AISID2 steel "NF and NN based prediction of responses in EDM of D2 steel,” *International Journal of Advanced Manufacturing Technology*, vol. 50, no. 5–8, 2010, doi: 10.1007/s00170-010-2531-8.

[23] H. Fazlollahtabar and H. Gholizadeh, “Fuzzy possibility regression integrated with fuzzy adaptive neural network for predicting and optimizing electrical discharge machining parameters,” *Computers and Industrial Engineering*, vol. 140, 2020, doi: 10.1016/j.cie.2019.106225.

[24] M. A. Rahman Khan, M. M. Rahman, and K. Kadirgama, “Neural network modeling and analysis for surface characteristics in electrical discharge machining,” in *Procedia Engineering*, 2014, vol. 90. doi: 10.1016/j.proeng.2014.11.783.

[25] M. A. Moghaddam and F. Kolahan, “Modeling and optimization of the electrical discharge machining process based on a combined arti cial neural network and particle swarm optimization algorithm,” *Scientia Iranica*, vol. 27, no. 3 B, 2020, doi: 10.24200/SCI.2019.5152.1123.

[26] S. Shakeri, A. Ghassemi, M. Hassani, and A. Hajian, “Investigation of material removal rate and surface roughness in wire electrical discharge machining process for cementation alloy steel using artificial neural network,” *International Journal of*

Advanced Manufacturing Technology, vol. 82, no. 1–4, 2016, doi: 10.1007/s00170-015-7349-y.

[27] M. K. Pradhan and R. Das, "Recurrent neural network estimation of material removal rate in electrical discharge machining of AISI D2 tool steel," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 225, no. 3, 2011, doi: 10.1177/2041297510394083.

[28] S. S. Sidhu, A. Batish, and S. Kumar, "Neural network-based modeling to predict residual stresses during electric discharge machining of Al/SiC metal matrix composites," in *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2013, vol. 227, no. 11. doi: 10.1177/0954405413492505.

[29] C. J. Tzeng and R. Y. Chen, "Optimization of electric discharge machining process using the response surface methodology and genetic algorithm approach," *International Journal of Precision Engineering and Manufacturing*, vol. 14, no. 5, 2013, doi: 10.1007/s12541-013-0095-x.

[30] K. Maji and D. K. Pratihar, "Forward and reverse mappings of electrical discharge machining process using adaptive network-based fuzzy inference system," *Expert Systems with Applications*, vol. 37, no. 12, 2010, doi: 10.1016/j.eswa.2010.05.019.

[31] S. Aravind Krishnan and G. L. Samuel, "Multi-objective optimization of material removal rate and surface roughness in wire electrical discharge turning," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 9–12, 2013, doi: 10.1007/s00170-012-4628-8.

[32] K. Kalita, R. K. Ghadai, D. S. Shinde, and X.-Z. Gao, "Data-Driven Genetic Programming-Based Symbolic Regression Metamodels for EDM Process," 2020. doi: 10.4018/978-1-7998-7206-1.ch009.

[33] E. Ostertagová, "Modelling using polynomial regression," in *Procedia Engineering*, 2012, vol. 48. doi: 10.1016/j.proeng.2012.09.545.

[34] Q. Que and M. Belkin, "Back to the Future: Radial Basis Function Network Revisited," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, 2020, doi: 10.1109/TPAMI.2019.2906594.

[35] G. E. Fasshauer and M. J. McCourt, "Stable evaluation of gaussian radial basis function interpolants," *SIAM Journal on Scientific Computing*, vol. 34, no. 2, 2012, doi: 10.1137/110824784.

[36] J. Leyva-Bravo, P. Chiñas-Sanchez, A. Hernandez-Rodriguez, and G. G. Hernandez-Alba, "Electrochemical discharge machining modeling through different soft computing approaches," *International Journal of Advanced Manufacturing Technology*, vol. 106, no. 7–8, 2020, doi: 10.1007/s00170-019-04766-z.

[37] T. T. Nguyen, M. Mia, X. P. Dang, C. H. Le, and M. S. Packianather, "Green machining for the dry milling process of stainless steel 304," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 234, no. 5, 2020, doi: 10.1177/0954405419888126.

[38] N. Cressie, "The origins of kriging," *Mathematical Geology*, vol. 22, no. 3, 1990, doi: 10.1007/BF00889887.

[39] T. T. Nguyen, "Prediction and optimization of machining energy, surface roughness, and production rate in SKD61 milling," *Measurement: Journal of the International Measurement Confederation*, vol. 136, 2019, doi: 10.1016/j.measurement.2019.01.009.

[40] J. Meng, Y. Wang, Q. Liao, and Y. Yang, “Corner-milling process parameter optimization regarding H62 brass using Kriging model and improved particle swarm optimization algorithm,” *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, no. 4, 2020, doi: 10.1007/s40430-020-2260-3.

[41] C. Ferreira, “Gene Expression Programming: A New Adaptive Algorithm for Solving Problems Cândida,” *Complex Systems*, vol. 13, no. 2, 2011.

[42] B. Sen, M. Mia, U. K. Mandal, and S. P. Mondal, “GEP- and ANN-based tool wear monitoring: a virtually sensing predictive platform for MQL-assisted milling of Inconel 690,” *International Journal of Advanced Manufacturing Technology*, vol. 105, no. 1–4, 2019, doi: 10.1007/s00170-019-04187-y.

[43] H. A. Shah, S. K. U. Rehman, M. F. Javed, and Y. Iftikhar, “Prediction of compressive and splitting tensile strength of concrete with fly ash by using gene expression programming,” *Structural Concrete*, 2021, doi: 10.1002/suco.202100213.

[44] R. Avanzato and F. Beritelli, “A CNN-based differential image processing approach for rainfall classification,” *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 4, 2020, doi: 10.25046/aj050452.

[45] T. Islam, “Plant Disease Detection using CNN Model and Image Processing,” *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, vol. 9, no. 10, 2020.

[46] J. Anitha, R. Das, and M. K. Pradhan, “Multi-objective optimization of Electrical Discharge Machining processes using artificial neural network,” *Jordan Journal of Mechanical and Industrial Engineering*, vol. 10, no. 1, 2016.

[47] S. Bhattacharya, P. Protim Das, P. Chatterjee, and S. Chakraborty, “Prediction of Reponses in a Sustainable Dry Turning Operation: A Comparative Analysis,” *Mathematical Problems in Engineering*, vol. 2021, 2021, doi: 10.1155/2021/9967970.

[48] A. Jadidi, R. B. Azhiri, and R. Teimouri, “Electrical discharge turning by assistance of external magnetic field, part I: Study of MRR and dimensional accuracy,” *International Journal of Lightweight Materials and Manufacture*, vol. 3, no. 3, pp. 265–276, Sep. 2020, doi: 10.1016/j.ijlmm.2020.02.004.

[49] H. Majumder and K. Maity, “Prediction and optimization of surface roughness and micro-hardness using grnn and MOORA-fuzzy-a MCDM approach for nitinol in WEDM,” *Measurement: Journal of the International Measurement Confederation*, vol. 118, 2018, doi: 10.1016/j.measurement.2018.01.003.