Prediction performance analysis of neural network models for an electrical discharge turning process

Kumaresh Dey · Kanak Kalita · Shankar Chakraborty

Received: 2 June 2022 / Accepted: 20 July 2022 / Published online: 11 August 2022
© The Author(s), under exclusive licence to Springer-Verlag France SAS, part of Springer Nature 2022

Abstract
In many of the modern-day manufacturing industries, electrical discharge machining (EDM) now appears as an effective non-traditional material removal process for generating intricate shape geometries on various hard-to-cut work materials to meet the ever-increasing demands of higher dimensional accuracy and better surface quality. Development of an appropriate prediction model for any of the EDM processes is quite difficult due to complex material removal mechanism, and dynamic interactions between the input parameters and responses. To address the problem, this paper proposes development and deployment of five neural network models, i.e. feed forward neural network, convolutional neural network, recurrent neural network, general regression neural network and long short term memory-based recurrent neural network as effective prediction tools for an electrical discharge turning (EDT) process. The EDT is a variant of EDM process involving removal of material from cylindrical workpieces. The input parameters for the considered EDT process are magnetic field, pulse current, pulse duration and angular velocity, whereas, the responses are material removal rate and overcut. Several statistical error metrics, like $R^2$, adjusted $R^2$ ($R^2_{adj}$), root mean square error and relative root mean square error are employed to compare the prediction accuracy of all the investigated neural network models. Based on a past experimental dataset, it is observed that long short term memory-based recurrent neural network provides more accurate prediction of both the responses under consideration. On the other hand, general regression neural network is noticed to be extremely robust having highly repetitive prediction performance.

Keywords Electrical discharge turning · Prediction · Neural network · Response · Statistical error metric

1 Introduction
Electrical discharge machining (EDM) is a non-traditional method of material removal based on erosive action of pulsating current which has been extensively utilized in tool and die making, automobile, aerospace and medical industries due to its ability to generate complex shapes and thin-wall configurations on many of the high-strength-temperature-resistant alloys, metal matrix composites and electrically conductive ceramics. This process is mostly suited to machine flat workpieces while creating a predefined hole/cavity which would be a projection of the tool profile [1]. It has several advantages, like suitability to machine very small workpieces, attainment of close tolerances, no burr formation, generation of no residual stress etc. On the other hand, it also suffers from many disadvantages, like low material removal rate (MRR), formation of heat affected zone and recast layer etc.

Several modifications to the EDM process, including rotary EDM, magnetic field-assisted EDM, ultrasonic-assisted EDM and so on, have been proposed in recent decades to enhance its machining performance. Electrical discharge turning (EDT) is another variant of EDM process, mainly employed to remove material from cylindrical workpieces. There are only a few research studies available in the literature leading to performance analysis and characterization of EDT processes [2]. Based on Taguchi methodology, Matoorian et al. [3] attempted to study the influences of intensity, pulse-on time, pulse-off time, voltage, servo and rotational speed on MRR. It was observed that the derived optimal combination of the EDT parameters would result in...
higher MRR value. Gohil and Puri [4] optimized an EDT process using grey relational analysis, and concluded that an optimal parametric intermix of pulse-on time $= 5 \mu s$, peak current $= 5 \text{ A}$, gap voltage $= 40 \text{ V}$, spindle speed $= 40 \text{ rpm}$ and no flushing pressure would result in achieving the most desired values of both MRR and surface roughness (SR). Gohil and Puri [5] also performed statistical study including variance and regression analysis of MRR and SR during EDT of titanium alloy. Srivastava et al. [6] investigated the effect of rotational speed on surface integrity involving various parameters, like SR, morphology of the recast layer, microhardness variation, and formation of residual stresses on the machined surface as well as subsurface during EDT operation. Extensive studies were conducted to explore the effects of several EDT parameters, i.e. magnetic flux density, discharge current, pulse-on time and rotational speed on MRR, overcut (OC), SR and recast layer thickness [7, 8]. Rehman et al. [9] noticed that MRR, OC, SR, recast layer thickness and hardness would be significantly affected by magnetic flux density, current, pulse time and spindle speed during EDT operation of die steel AISI D2 material. With the help of magnetic field assistance, MRR and hardness had increased by 56 and 38%, respectively, whereas, there had been 31, 43 and 47% reductions in OC, SR and recast layer thickness respectively. Other research studies include EDT using strip electrode [10], EDT in presence of ultrasonic vibration [11] and many more.

It has been noticed that EDT has a complex material removal mechanism due to involvement of several input parameters and conflicting responses, and dynamic interactions between them. Development of an appropriate prediction model for EDT would help in studying the process behavior and envisaging the tentative values of the responses for given sets of input parameters. Nowadays, for prediction of responses in any of the machining processes, applications of different techniques of machine learning and deep learning have become quite popular. Neural networks (NNs) serve the basic purpose of such deep learning techniques. Because of their tolerance to modest input errors, NNs are more advantageous than any other prediction tool. Different types of NNs are now available and have been researched about. The most popular NNs include feed forward neural network (FNN), convolutional neural network (CNN), recurrent neural network (RNN), general regression neural network (GRNN), RNN with long short term memory (LSTM) and many more. The FNN is based on simple mathematical calculations and quicker because it does not involve any logical rule [12, 13]. The FNN is commonly used on random dataset where it is hard to perform feature engineering and find patterns in the input–output relation. Examples of FNN applications include electrical load demand forecasting [14], predicting parameters responsible for COVID-19 outbreak [15], fault classification in rotating machines [16], optimization and prediction of responses of different machining processes [17–20] and many more. The CNN is mainly responsible for image processing and computer vision [21–24] revolution in the world. The key benefit of CNN over its predecessors is its capability to automatically recognize significant features without the need for human intervention, making it the most widely utilized NN model [25]. The RNN is one of the prime choices for prediction of sequence-based dataset, such as natural language processing [26–28] and time series prediction [29–31]. However, it is also effective in prediction and analysis of responses of different machining processes [32, 33]. The LSTM is a special kind of RNN, capable of learning long-term dependencies [34, 35]. Along with its natural language processing capability, LSTM has been proven to be an effective tool for acoustic modeling [36], trajectory prediction [37] and correlation analysis [38]. Meanwhile, a GRNN may converge to the optimal regression surface with more sample aggregations, and has non-linear mapping capability and rapid learning rate [39]. The prediction results of GRNN can be satisfactory even if the training samples are small, and can it also deal with unstable data [40].

It can be noticed that all the above-mentioned NN models have very limited applications as effective prediction tools in machining processes. No application of any of the considered NNs has been found in the domain of EDT process. Thus, this paper focuses on the applications of five different NN models in the form of FNN, CNN, RNN, LSTM and GRNN in accurately predicting the responses of an EDT process. Based on a past experimental data of EDT process, the prediction performance of all these NNs is validated with the help of four statistical error metrics, i.e. $R$-squared ($R^2$), adjusted $R$-squared ($R^2_{\text{adj}}$), root mean square error (RMSE) and relative root mean square error (RRMSE). To the best of the authors’ knowledge, this type of comprehensive comparative analysis of prediction performance of different NN models is unique in the area of EDT process.

Typically, the past researchers in the field of EDT process have relied on simple statistical methods, like response surface methodology (RSM) for its computational modeling, and deriving the functional relationships between the input parameters and process outputs. The application RSM technique is also restricted by the pre-specified fixed form selected a priori to the modeling step. Thus, there remains a major gap in capturing all the nonlinearity and complexity of the EDT process. Applications of NN models can effectively bridge this gap while including a variety of activation functions that can easily deal with high nonlinearity and complexity in the experimental data. Unlike RSM technique, carefully calibrated NN models do not require any secondary statistical step, like analysis of variance and term elimination method to robustify the model. A comprehensive comparison of various NN models would thus be quite useful to the future researches in justifying their usage. Additionally, as compared to other powerful methods, like gene expres-
sion programming, NN models have immense parallelism [41], thereby making them highly efficient in predicting the process characteristics and envisaging the response values. Compared to similar neuron-based methods, like adaptive neuro-fuzzy inference system (ANFIS), NN models are generally faster to train and deploy [42].

This paper is structured as follows: Sect. 2 briefly introduces details of all the considered NN models along with the statistical error metrics. Applications of these NN models to an EDT process are enumerated in Sect. 3. Section 4 deals with prediction performance analysis of the NN models and conclusions are drawn in Sect. 5.

2 Neural network models

2.1 Feed forward neural network (FNN)

An FNN is a mathematical model that is inspired by the biological NNs’ functional features. An NN is usually 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 1 provides the 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 FNN.

2.2 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’s visual cortex [21, 22], 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), 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 exhibits a simple representation of the 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 input features. The next layer is a flattened layer where the layers of convolutional 1D are flattened into one single array. The most important imputed data from the flattened layer is finally obtained in the output layer.

2.3 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
An infinite impulse recurrent network is a directed cyclic graph that cannot be unrolled and replaced with a strictly 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 3 is the flowchart representation of a traditional RNN process, as developed by Zhang et al. [32]. The computational process of the traditional RNN can be explained using the following equation:

\[ h^{<t>} = \sigma \left( W x^{<t>} + U h^{<t-1>} + b \right) \]  

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 matrices, \( b \) is the bias vector, and \( \sigma \) 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.

2.4 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 (like speeches or videos). A typical LSTM model usually consists of a cell, an input gate, an output gate and a forget gate. These three gates control the flow of information into and out of the cell, and the cell remembers values across arbitrary time intervals.

Figure 4 provides the 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 (\( \sigma \)). In the next step, it passes through a forget gate loop where the internal state variable \( s_t \), lagged by one time step (\( s_{t-1} \)), 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 \( h_t \).

2.5 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 feed forward 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, GRNN trains itself in much less time.

The general network flow of a GRNN architecture is exhibited in Fig. 5. 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.

Typically, in NN models, hidden layer(s) is the bridge between the input and output layers. Its main function is to perform nonlinear transformations of the inputs. The hidden layers are responsible for carrying out bulk of the ‘mathematical mapping’. Though hidden layers are extremely common in NN models, their use and architecture vary depending on the use case. In this paper, pilot tests are carried out to select the best architecture for each of the five NN models.
used. Thus, the number of hidden layers is different from one another in FNN, CNN, RNN, LSTM and GRNN models.

2.6 Statistical error metrics

In order to validate the prediction performance of FNN, CNN, RNN, LSTM and GRNN models, four statistical error metrics, i.e. $R^2$, $R^2_{\text{adj}}$, RMSE and RRMSE are considered in this paper. The corresponding mathematical expressions of all these metrics are provided as below:

\[
R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}.
\]

(2)

where $y_i$ represents the actual value of $i^{th}$ observation, $\hat{y}_i$ denotes the predicted value of $i^{th}$ observation, $\bar{y}$ is the mean of all the observations and $n$ is the number of observations. The $R^2_{\text{adj}}$ is the modified version of $R^2$ which considers number of predictors (independent variables) in the model. It has been noticed that increasing number of terms in a model may lead to higher $R^2$ value \[43\]. To overcome this problem, $R^2_{\text{adj}}$ is taken into account to highlight the percentage of dependent variable variation that can be explained by its relationship with one or more predictor variables, considering the number of predictors in the model. It represents fitness of the model and adjusts for the number of terms in the model. Its value should be always less than or equal to $R^2$. The $R^2_{\text{adj}}$ value can be calculated using the following equation:

\[
R^2_{\text{adj}} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}.
\]

(3)

where $p$ denotes the number of independent variables in the model. On the other hand, RMSE and RRMSE can be estimated employing the following expressions:

\[
\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y})^2}{n}}
\]

(4)

\[
\text{RRMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y})^2}{n \times \sum_{i=1}^{n} \hat{y}_i^2}}
\]

(5)

3 Neural network-based prediction of EDT responses

It has already been mentioned that the 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. \[7\] is considered here. On a TEHRAN EKRAM machine modified by a spindle to perform turning operation to produce sharp edge grooves, Jadidi et al. \[7\] performed 81 experiments to study the effects of magnetic field (M) (in T), pulse current (Ip) (in A), pulse duration (Ton) (in $\mu$s) and angular velocity (N) (in rpm) on volumetric MRR (in mm$^3$/min) and OC (in $\mu$m). 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, all the input EDT parameters were varied at three different levels, i.e. pulse current (5, 10 and 15 A), pulse duration (600, 800 and 1000 $\mu$s), magnetic field density (0, 0.2 and 0.4 T) and angular velocity (50, 150 and 250 rpm). Table 1 depicts the experimental dataset and the measured response values. The detailed working principle along with the actual photograph of the EDT process can be available in \[7\].

3.1 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 randomly chosen for training and developing all the NN models, and the remaining observations are treated as testing data points to validate the prediction perfor-
| Exp. No. | M | Ip | Ton | N | MRR | OC | Exp. No. | M | Ip | Ton | N | MRR | OC |
|---------|---|----|-----|---|-----|----|---------|---|----|-----|---|-----|----|
| 1       | 0 | 5  | 600 | 50| 231 | 113| 0.2     | 10| 800| 250 | 433| 92  |    |
| 2       | 0 | 5  | 600 | 150| 241 | 103| 0.2     | 10| 1000| 50 | 405 | 143 |    |
| 3       | 0 | 5  | 600 | 250| 245 | 88 | 0.2     | 10| 1000| 150| 411 | 131 |    |
| 4       | 0 | 800| 50  | 308| 135 | 45 | 0.2     | 10| 1000| 250| 415 | 112 |    |
| 5       | 0 | 800| 150 | 315| 123 | 46 | 0.2     | 15| 600 | 50 | 381 | 109 |    |
| 6       | 0 | 800| 250 | 303| 105 | 47 | 0.2     | 15| 600 | 150| 397 | 99  |    |
| 7       | 0 | 1000| 50 | 283 | 163| 48 | 0.2     | 15| 600 | 50 | 405 | 85  |    |
| 8       | 0 | 1000| 150| 288 | 149| 49 | 0.2     | 15| 800 | 50 | 508 | 130 |    |
| 9       | 0 | 1000| 250| 303 | 105| 47 | 0.2     | 15| 800 | 150| 520 | 118 |    |
| 10      | 0 | 1000| 300| 124 | 51 | 0.2 | 15     | 800| 150| 500 | 101 |    |
| 11      | 0 | 1000| 150| 113 | 52 | 0.2 | 15     | 800| 150| 468 | 157 |    |
| 12      | 0 | 1000| 318| 97  | 53 | 0.2 | 15     | 1000| 50| 475 | 143 |    |
| 13      | 0 | 1000| 400| 148 | 54 | 0.2 | 15     | 1000| 150| 479 | 122 |    |
| 14      | 0 | 1000| 410| 135 | 55 | 0.4 | 5      | 600| 50 | 323 | 68  |    |
| 15      | 0 | 1000| 394| 115 | 56 | 0.4 | 5      | 600| 150| 373 | 62  |    |
| 16      | 0 | 1000| 368| 179 | 57 | 0.4 | 5      | 600| 250| 343 | 53  |    |
| 17      | 0 | 1000| 374| 164 | 58 | 0.4 | 5      | 800| 50 | 431 | 81  |    |
| 18      | 0 | 1000| 377| 140 | 59 | 0.4 | 5      | 800| 150| 441 | 74  |    |
| 19      | 0 | 1500| 50 | 346 | 136| 60 | 0.4     | 5 | 800| 250| 424 | 63  |    |
| 20      | 0 | 1500| 361| 124 | 61 | 0.4 | 5      | 1000| 50| 396 | 98  |    |
| 21      | 0 | 1500| 250| 368 | 106| 62 | 0.4     | 5 | 1000| 150| 403 | 89  |    |
| 22      | 0 | 1500| 462| 162 | 63 | 0.4 | 5      | 1000| 250| 406 | 76  |    |
| 23      | 0 | 1500| 473| 148 | 64 | 0.4 | 5      | 1000| 50 | 420 | 74  |    |
| 24      | 0 | 1500| 455| 126 | 65 | 0.4 | 10     | 600| 150| 438 | 68  |    |
| 25      | 0 | 1500| 425| 196 | 66 | 0.4 | 10     | 600| 250| 445 | 58  |    |
| 26      | 0 | 1500| 432| 179 | 67 | 0.4 | 10     | 800| 50 | 560 | 89  |    |
| 27      | 0 | 1500| 435| 152 | 68 | 0.4 | 10     | 800| 150| 574 | 81  |    |
| 28      | 0.2| 600| 50 | 254 | 90 | 69 | 0.4     | 10| 800| 250| 552 | 69  |    |
| 29      | 0.2| 600| 150| 265 | 82 | 70 | 0.4     | 10| 1000| 50| 515 | 107 |    |
| 30      | 0.2| 600| 250| 270 | 70 | 71 | 0.4     | 10| 1000| 150| 524 | 98  |    |
| 31      | 0.2| 800| 50 | 339 | 108| 72 | 0.4     | 10| 1000| 250| 528 | 84  |    |
| 32      | 0.2| 800| 150| 347 | 98 | 73 | 0.4     | 15| 600| 50 | 484 | 82  |    |
| 33      | 0.2| 800| 250| 333 | 84 | 74 | 0.4     | 15| 600| 150| 505 | 74  |    |
| 34      | 0.2| 1000| 50 | 311 | 130| 75 | 0.4     | 15| 600| 250| 515 | 84  |    |
| 35      | 0.2| 1000| 150| 317 | 119| 76 | 0.4     | 15| 800| 50 | 647 | 97  |    |
| 36      | 0.2| 1000| 250| 319 | 102| 77 | 0.4     | 15| 800| 150| 662 | 89  |    |
| 37      | 0.2| 600| 50 | 330 | 99 | 78 | 0.4     | 15| 800| 250| 637 | 76  |    |
| 38      | 0.2| 600| 150| 344 | 90 | 79 | 0.4     | 15| 1000| 50| 595 | 118 |    |
| 39      | 0.2| 600| 250| 350 | 78 | 80 | 0.4     | 15| 1000| 150| 605 | 107 |    |
| 40      | 0.2| 800| 50 | 440 | 118| 81 | 0.4     | 15| 1000| 250| 609 | 91  |    |
| 41      | 0.2| 800| 150| 451 | 108|    |        |    |    |    |    |     |    |

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.

#### 3.2 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).

3.2.1 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. (6), 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. The 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 implements the concept of decaying or exponential moving average of partial gradients. The FNN model is run over 5000 epochs. Figure 6 shows the architectural diagram of the developed FNN model.

\[
\frac{\partial f}{\partial x} = \begin{cases} 
1 & \text{if } x \geq 0; \\
0 & \text{else}
\end{cases}
\] (6)

3.2.2 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 (Conv1D) layer with 100 nodes. The Conv1D layer is activated using ReLU function. Another hidden layer is a flattened layer, which converts the 100 output arrays coming from the nodes of Conv1D layer into a single one-dimensional array. The AdaM optimizer is adopted here with MSE as the loss function during the compilation process. This model is run over 5000 epochs. Figure 7 depicts the architectural diagram of the CNN model.

3.2.3 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. (7), 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
Fig. 8 RNN architecture for model training

is also trained over 5000 epochs. The architectural diagram of RNN model is portrayed in Fig. 8.

\[ S(x) = \frac{1}{1 + e^{-x}} \]  

(7)

3.2.4 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 9 shows the developed LSTM architecture. The hidden layer is activated by tanh function. During compilation, AdaM optimization process is taken into account along with MSE as the loss function. This model is also run over 5000 epochs.

3.2.5 GRNN

The developed GRNN model consists of four layers, i.e. one input layer, one pattern layer, one summation layer and an output layer. 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.

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 Fig. 10, the developed GRNN architecture is presented.

4 Prediction performance analysis

4.1 Performance analysis using experimental dataset

All the developed NN models with the specific architectures and parameters, mentioned in Sub-Sect. 3.2, 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 2, 3 respectively. In Fig. 11, the 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 Fig. 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 NN models, the normality of residuals is analyzed in Fig. 12. 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 (\( \mu \)) of the residuals. The GRNN
| Exp. No. | Target FNN | CNN | RNN | GRNN | LSTM | Exp. No. | Target FNN | CNN | RNN | GRNN | LSTM |
|---------|------------|-----|-----|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|------|---------|------------|-----|-----|------|------|-------
| Exp. No. | Target | FNN    | CNN    | RNN    | GRNN   | LSTM  | Exp. No. | Target | FNN    | CNN    | RNN    | GRNN   | LSTM  |
|---------|--------|--------|--------|--------|--------|-------|---------|--------|--------|--------|--------|--------|-------|
| 23      | 473    | 441.6089 | 445.2768 | 472.1294 | 473    | 465.158 | 64      | 420    | 463.0616 | 490.8338 | 433.0352 | 403.5 | 415.0911 |
| 24      | 455    | 443.2077 | 454.0099 | 458.4851 | 455    | 462.3222 | 65      | 438    | 471.1336 | 490.4475 | 451.2133 | 437.9998 | 426.8229 |
| 25      | 425    | 459.8849 | 440.5154 | 434.6253 | 425    | 422.8334 | 66      | 445    | 453.5775 | 437.8876 | 447.585 | 444.9998 | 434.9218 |
| 26      | 432    | 461.4838 | 440.8075 | 443.1791 | 432    | 432.9521 | 67      | 560    | 517.6382 | 545.4392 | 560.531 | 559.9999 | 554.272 |
| 27      | 435    | 463.0826 | 441.0996 | 438.8657 | 434.9999 | 434.8688 | 68      | 574    | 515.0923 | 545.8732 | 569.3662 | 574    | 571.9506 |
| 28      | 254    | 289.4057 | 298.6721 | 278.1001 | 277    | 256.4423 | 69      | 552    | 514.7343 | 559.6365 | 559.3177 | 551.9997 | 556.8716 |
| 29      | 265    | 291.0046 | 298.1467 | 290.4476 | 265.0002 | 266.3912 | 70      | 515    | 518.4727 | 527.3013 | 541.0726 | 546.9464 | 505.5234 |
| 30      | 270    | 272.2703 | 294.3777 | 300.5333 | 270.0001 | 268.824 | 71      | 524    | 515.9268 | 527.7189 | 546.7112 | 523.9997 | 524.5939 |
| 31      | 339    | 309.2806 | 346.4592 | 377.103 | 339.0001 | 337.0804 | 72      | 528    | 513.3809 | 528.1364 | 542.8898 | 527.9999 | 521.7757 |
| 32      | 347    | 310.8795 | 346.4688 | 385.7106 | 315.0007 | 341.7346 | 73      | 484    | 479.7213 | 505.6108 | 498.6324 | 483.9998 | 479.0403 |
| 33      | 333    | 328.4267 | 350.8888 | 373.2826 | 333.0003 | 338.134 | 74      | 505    | 487.7932 | 505.2244 | 519.2001 | 504.9999 | 502.5916 |
| 34      | 311    | 329.1555 | 334.6184 | 331.4263 | 311.0003 | 312.8942 | 75      | 515    | 484.2672 | 458.06 | 511.4446 | 445    | 509.8736 |
| 35      | 317    | 330.7544 | 335.3602 | 342.7874 | 367.0552 | 319.8819 | 76      | 647    | 599.0868 | 622.4588 | 649.5006 | 646.9996 | 627.1984 |
| 36      | 319    | 332.3532 | 336.1021 | 337.1532 | 319.0001 | 317.415 | 77      | 662    | 607.1587 | 622.9064 | 656.4434 | 546.7412 | 629.381 |
| 37      | 330    | 377.9568 | 372.0195 | 332.7023 | 340.1118 | 329.098 | 78      | 637    | 614.5477 | 635.8134 | 638.0488 | 636.9996 | 620.2736 |
| 38      | 344    | 386.0287 | 371.4942 | 348.2987 | 344    | 343.7427 | 79      | 595    | 626.921 | 618.8296 | 603.0941 | 594.9998 | 603.1805 |
| 39      | 350    | 356.5101 | 367.9689 | 357.0338 | 344.8068 | 349.3522 | 80      | 605    | 624.3751 | 619.2471 | 610.6132 | 604.9996 | 615.7647 |
| 40      | 440    | 398.5565 | 425.6302 | 456.0275 | 452.0208 | 430.1285 | 81      | 609    | 621.8291 | 619.6647 | 600.5006 | 591.342 | 604.7985 |
| 41      | 451    | 400.1553 | 426.372 | 459.2618 | 501.2143 | 443.736 |         |         |        |        |        |        |       |
| Exp. No. | Target | FNN      | CNN      | RNN      | GRNN     | LSTM     | Exp. No. | Target | FNN      | CNN      | RNN      | GRNN     | LSTM     |
|----------|--------|----------|----------|----------|----------|----------|----------|--------|----------|----------|----------|----------|----------|
| 1        | 113    | 119.1732 | 117.1817 | 118.7867 | 113      | 113.3572 | 42       | 92     | 94.5168  | 93.43893 | 94.20787 | 92       | 91.58872 |
| 2        | 103    | 105.4071 | 102.9344 | 103.5098 | 103      | 103.6932 | 43       | 143    | 142.0636 | 140.6329 | 148.8281 | 143.0001 | 141.5845 |
| 3        | 88     | 91.64101 | 93.2215  | 94.12621 | 87.9996  | 88.23135 | 44       | 131    | 128.2975 | 129.4192 | 133.5659 | 131      | 129.6714 |
| 4        | 135    | 139.1878 | 136.5541 | 142.8677 | 134.9999 | 134.5466 | 45       | 112    | 114.5314 | 113.4675 | 118.3068 | 112      | 110.5894 |
| 5        | 123    | 125.4217 | 123.9434 | 127.6086 | 123      | 123.2369 | 46       | 109    | 111.9031 | 110.6315 | 112.4175 | 109      | 108.7342 |
| 6        | 105    | 111.6556 | 108.8373 | 112.4053 | 105      | 104.6459 | 47       | 99     | 98.13701 | 96.38415 | 97.15841 | 99       | 99.78037 |
| 7        | 163    | 159.2023 | 155.9231 | 166.9667 | 162.9999 | 163.1688 | 48       | 85     | 84.3709  | 86.67123 | 83.98773 | 106      | 84.92549 |
| 8        | 149    | 145.4362 | 144.7093 | 151.7076 | 149      | 147.9738 | 49       | 130    | 131.9177 | 131.1046 | 136.5165 | 130      | 129.5996 |
| 9        | 127    | 131.6701 | 128.7576 | 136.4485 | 126.9999 | 126.6402 | 50       | 118    | 118.1516 | 118.4938 | 121.2573 | 118.0001 | 118.6681 |
| 10       | 124    | 129.0418 | 128.8575 | 130.5592 | 124      | 124.2958 | 51       | 101    | 104.3855 | 103.3877 | 105.9983 | 101      | 100.6499 |
| 11       | 113    | 115.2757 | 114.6102 | 115.3001 | 90       | 114.0981 | 52       | 157    | 151.9322 | 151.7542 | 160.6207 | 157      | 157.5365 |
| 12       | 97     | 101.5096 | 104.8973 | 101.7745 | 97       | 96.91842 | 53       | 143    | 138.1661 | 140.5404 | 145.3563 | 143      | 142.584 |
| 13       | 148    | 149.0564 | 149.4253 | 154.6581 | 148      | 148.2501 | 54       | 122    | 124.4    | 124.5888 | 130.0972 | 121.5    | 122.0401 |
| 14       | 135    | 135.2902 | 136.8145 | 139.3991 | 135      | 135.4655 | 55       | 68     | 65.1587  | 64.97232 | 63.2821  | 68       | 68.77152 |
| 15       | 115    | 121.5241 | 121.7084 | 124.14   | 115      | 116.2149 | 56       | 62     | 51.39259 | 50.72502 | 55.3672  | 62.00004 | 64.75293 |
| 16       | 179    | 169.0709 | 168.9024 | 178.7593 | 178.9999 | 175.2614 | 57       | 53     | 37.62647 | 41.0121  | 51.83048 | 53.00003 | 56.00575 |
| 17       | 164    | 155.3048 | 157.6887 | 163.498  | 163.9999 | 164.646  | 58       | 81     | 85.17326 | 84.3447  | 83.00337 | 81.00005 | 81.20598 |
| 18       | 140    | 141.5387 | 141.737  | 148.2388 | 139.9999 | 139.8717 | 59       | 74     | 71.40714 | 71.73396 | 74.88757 | 83.84985 | 75.80059 |
| 19       | 136    | 138.9104 | 138.9019 | 142.3496 | 135.9999 | 136.4606 | 60       | 63     | 57.64103 | 56.62787 | 68.40468 | 63.00004 | 65.40933 |
| 20       | 124    | 125.1443 | 124.6546 | 127.0905 | 124      | 124.7663 | 61       | 98     | 105.1878 | 103.7137 | 107.1042 | 98.00006 | 98.43254 |
| 21       | 106    | 111.3782 | 114.9416 | 111.8516 | 106      | 106.9143 | 62       | 89     | 91.42169 | 92.49994 | 94.40794 | 89       | 90.66702 |
| 22       | 162    | 158.9249 | 159.375  | 166.4484 | 161.9999 | 162.7376 | 63       | 76     | 77.65557 | 76.54825 | 86.49302 | 102      | 77.99039 |
| Exp. No. | Target | FNN       | CNN       | RNN       | GRNN      | LSTM      | Exp. No. | Target | FNN       | CNN       | RNN       | GRNN      | LSTM      |
|---------|--------|-----------|-----------|-----------|-----------|-----------|---------|--------|-----------|-----------|-----------|-----------|-----------|---------|
| 23      | 148    | 145.1588  | 146.7642  | 151.1895  | 147.9999  | 147.8719  | 64      | 74     | 75.0278   | 72.74953  | 70.69492  | 70.69492  | 76.72082  | 73.79083 |
| 24      | 126    | 131.3927  | 131.6581  | 135.9305  | 125.9999  | 127.0482  | 65      | 68     | 61.26116  | 58.50222  | 60.12483  | 68.00004  | 69.19202  |
| 25      | 196    | 178.9395  | 180.0246  | 190.5519  | 195.9999  | 179.142   | 66      | 58     | 47.49505  | 48.7893   | 55.12465  | 58        | 59.20315  |
| 26      | 179    | 165.1734  | 168.8108  | 175.2884  | 178.9999  | 175.6008  | 67      | 89     | 95.04182  | 93.31722  | 94.79379  | 89        | 88.2756   |
| 27      | 152    | 151.4073  | 152.8592  | 160.0293  | 152       | 152.4515  | 68      | 81     | 81.27572  | 80.70649  | 79.6452   | 81        | 81.74551  |
| 28      | 90     | 92.16597  | 89.01103  | 88.83662  | 90.5      | 89.6367   | 69      | 69     | 67.5096   | 65.6004   | 71.82796  | 69.00005  | 70.6964   |
| 29      | 82     | 78.39986  | 74.76373  | 73.57761  | 82        | 83.08086  | 70      | 107    | 115.0564  | 112.7944  | 118.8968  | 143       | 107.473   |
| 30      | 70     | 64.63374  | 65.05081  | 68.69128  | 70        | 72.38254  | 71      | 98     | 101.2903  | 101.5807  | 103.6338  | 98.00007  | 98.45018  |
| 31      | 108    | 112.1805  | 108.3834  | 112.9356  | 108       | 107.5831  | 72      | 84     | 87.52415  | 85.629    | 91.25066  | 84.00006  | 83.30798  |
| 32      | 98     | 98.4144   | 95.77267  | 97.67657  | 123       | 98.93671  | 73      | 82     | 84.89585  | 82.36105  | 82.48532  | 82.00005  | 81.37911  |
| 33      | 84     | 84.64829  | 80.66657  | 83.15662  | 84        | 84.7815   | 74      | 74     | 71.12974  | 68.11375  | 67.26267  | 74        | 75.45749  |
| 34      | 130    | 132.1951  | 127.7524  | 137.0354  | 130       | 129.0288  | 75      | 84     | 57.36362  | 58.40083  | 58.55273  | 61.25697  | 64.85317  |
| 35      | 119    | 118.429   | 116.5387  | 121.7754  | 119       | 118.8686  | 76      | 97     | 104.9104  | 102.8342  | 106.5853  | 97.00007  | 96.9202   |
| 36      | 102    | 104.6628  | 100.587   | 106.5164  | 102.0001  | 100.5181  | 77      | 89     | 91.14429  | 90.22343  | 91.32515  | 118       | 89.96214  |
| 37      | 99     | 102.0345  | 100.5881  | 100.627   | 124       | 98.36054  | 78      | 76     | 77.37817  | 75.11734  | 76.48792  | 76.00005  | 77.01233  |
| 38      | 90     | 88.26843  | 86.34076  | 85.36806  | 89.99996  | 90.67656  | 79      | 118    | 124.925   | 123.4838  | 130.6894  | 118.0001  | 117.432   |
| 39      | 78     | 74.50232  | 76.62785  | 76.33949  | 77.5      | 78.28259  | 80      | 107    | 111.1588  | 112.27    | 115.4242  | 107.0001  | 108.1045  |
| 40      | 118    | 122.0491  | 121.1558  | 124.726   | 118.5     | 117.8336  | 81      | 91     | 97.39272  | 96.31836  | 100.1652  | 91        | 90.71896  |
| 41      | 108    | 108.283   | 108.545   | 109.4669  | 108       | 108.085   | 108     |        |          |           |           |           |           |         |
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 large deviation of the residuals from normality.

In Fig. 13, 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 Fig. 14. Although presence of outliers is detected in all the NN models, RNN is the most left-skewed. The GRNN 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 inter-
Table 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 |
|          | 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 |

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.

As observed from Table 4 and Fig. 15, 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 2.7162 and 0.0270 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 as the best performing NN model with the corresponding $R^2$, $R^2_{adj}$, RMSE and RRMSE values as 0.9905, 0.9871, 2.9825 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 on MRR, as clearly noticed from Figs. 15 and 16.

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 performance 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 different from that obtained for MRR. On the basis of $R^2$ values for prediction of OC on 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 0.0836, 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.

To assess robustness of the derived results, each NN model is trained independently (using different seed values) for \( n \) times and tested \( n \) times. It should be noted that in Table 4 as well as Figs. 15 and 16, the prediction performance measures of only the best models are shown. Figure 17 exhibits the variation in the prediction performance measure (overall \( R^2 \)) after \( n = 10 \) independent trials. The box plots in Fig. 17 provide visually intuitive measurement of the models’ performance on repeated trials. Thus, instead of reporting only standard deviations, these box plots are more powerful tools in visualizing the repeatability characteristics of the models. Naturally, an NN model with high model accuracy as well as low variation spread in repeated trials is better than a model with high model accuracy as well as high variation spread. It can be noticed that in general for both the responses, FNN and LSTM have the highest variations in the predicted results. The variation in repeated trials is found to be the lowest for CNN models. In case of GRNN models for MRR modeling (Fig. 17a), though the variation of GRNN is more than CNN models, the mean and median of \( n \) trials of GRNN are found to be 0.9785 and 0.9829 respectively, which are higher than the best model values of all other NN models. This indicates an extremely high likelihood of GRNN outperforming any randomly chosen NN model among the other considered architectures. Similarly, in case of GRNN models for OC modeling (Fig. 17b), the median of \( n \) trials is estimated to be 0.9773, which is higher than the best model values of all other NN models. It simply means that at least 50% of the developed GRNN models outperform all other NN models.

5 Conclusions

This paper deals with identifying the most suitable NN model for prediction of two responses, i.e. MRR and OC, during EDT operation of AISI D2 alloy based on four input parameters, i.e. magnetic field, pulse current, pulse duration and angular velocity. Five popular NN models, i.e. FNN, CNN, RNN, GRNN and LSTM are developed for both the considered responses, and a detailed comparative analysis of their prediction performance is carried out using four statistical error metrics, i.e. \( R^2 \), \( R^2_{\text{adj}} \), RMSE and RRMSE. It is observed that for both MRR and OC, LSTM emerges out as the best performing NN model with the maximum \( R^2 \) and \( R^2_{\text{adj}} \), and minimum RMSE and RRMSE values. The LSTM incorporates both long term and short term memories to store the most repetitive characteristics during training. This adds more value to back-propagation and can provide more accurate results than simple RNN process. The LSTM helps to overcome vanishing and exploding gradient problems during training, thus providing the most reliable prediction results on both the training and testing data. Furthermore, the prediction performance of CNN and RNN is almost comparable. The FNN, being a basic and random learning NN model, provides moderately satisfactory results. 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. Additionally, based on repeatability of the prediction performance, GRNN is found to be extremely robust. In this paper, a past experimental dataset containing 81 experimental runs is considered.
to train and test the considered NN models. A better picture may be obtained if a large data repository is developed for the same purpose. As a future scope of this paper, other NN models, like RNN with gated recurrent unit (GRU), RBFN, modular neural network (MNN) or CNN with LSTM etc. may be applied for prediction of responses of the considered EDT process, and their performance may be contrasted with that of the present NN models.

**Declarations**

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**References**

1. Kumar, V., Diyaley, S., Chakraborty, S.: Teaching-learning-based parametric optimization of an electrical discharge machining process. Facta Univ. Ser. Mech. Eng. **18**(2), 281–300 (2020)
2. Jozić, S., Bajić, D., Dumanić, I., Bagavac, Z.: Optimization for an efficient and highly productive turning process. Rep. Mech. Eng. **2**(1), 212–221 (2021)
3. Matoorian, P., Sulaiman, S., Ahmad, M.M.H.M.: An experimental study for optimization of electrical discharge turning (EDT) process. J. Mater. Process. Technol. **204**, 350–356 (2008)
4. Gohil, V., Puri, Y.M.: Optimization of electrical discharge turning process using Taguchi-grey relational approach. Proc. CIRP **68**, 70–75 (2018)
5. Gohil, V., Puri, Y.M.: Statistical analysis of material removal rate and surface roughness in electrical discharge turning of titanium alloy (Ti-6Al-4V). Proc. Inst. Mech. Eng. Part B J. Eng. Manuf. **232**(9), 1603–1614 (2018)
6. Srivastava, A.K., Nag, A., Dixit, A.R., Hloch, S., Tiwari, S., Scucka, J., Pachauri, P.: Surface integrity in wire-EDM tangential turning of in situ hybrid metal matrix composite A359/B4C/Al2O3. Sci. Eng. Compos. Mater. **26**(1), 122–133 (2019)
7. Jadidi, A., Azhiri, R.B., Teimouri, R.: Electrical discharge turning by assistance of external magnetic field, part I: Study of MRR and dimensional accuracy. Int. J. Lightweight Mater. Manuf. **3**(3), 265–276 (2020)
8. Azhiri, R.B., Jadidi, A., Teimouri, R.: Electrical discharge turning by assistance of external magnetic field, part II: Study of surface integrity. Int. J. Lightweight Mater. Manuf. **3**(3), 305–315 (2020)
9. Rehman, S., Alam, M.M., Alherns, L.M., Alimoradi, A.: Experimental modeling and optimization of magnetic field assisted electrical discharge turning: Applicable for wind power turbine elements. Alex. Eng. J. **60**, 2209–2223 (2021)
10. Song, K.Y., Chung, D.K., Park, M.S., Chu, C.N.: EDM turning using a strip electrode. J. Mater. Process. Technol. **213**(9), 1495–1500 (2013)
11. Azhiri, R.B., Bideskan, A.S., Javidpour, F., Tekiyeh, R.M.: Study on material removal rate, surface quality, and residual stress of AISI D2 tool steel in electrical discharge machining in presence of ultrasonic vibration effect. Int. J. Adv. Manuf. Technol. **101**(9–12), 2849–2860 (2019)
12. Natarajan, K.K., Gokulachandran, J.: Artificial neural network based machining operation selection for prismatic components. Int. J. Adv. Sci. Eng. Inf. Technol. **10**(2), 618–628 (2020)
13. Hinton, G.E., Osindero, S., Teh, Y.-H.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
14. Machado, E., Pinto, T., Guedes, V., Morais, H.: Electrical load demand forecasting using feed-forward neural networks. Energies **14**, 7644 (2021)
15. Kumar, A., Sinwar, D., Saini, M.: Study of several key parameters responsible for COVID-19 outbreak using multiple regression analysis and multi-layer feed forward neural network. J. Interdiscip. Math. **24**(1), 53–75 (2021)
16. Alghoul, A., Jamdal, A., Alysouf, I., Bingamil, A., Ali, M., Al Baiti, S.: On the usefulness of pre-processing methods in rotating machines faults classification using artificial neural network. J. Appl. Comput. Mech. **7**(1), 254–261 (2021)
17. Chalisaonkar, R., Kumar, J., Pant, P.: Prediction of machining characteristics of finish cut WEDM process for pure titanium using feed forward back propagation neural network. Mater. Today Proc. **4**(2), 203–212 (2019)
18. Karthikeyan, S., Subbarayan, M.R., Mathan Kumar, P., Beemaraj, R.K., Sivakandan, C.: Computer vision-based surface roughness measurement using artificial neural network. Mater. Today Proc. **60**, 1325–1328 (2022)
19. Sangwan, K.S., Saxena, S., Kant, G.: Optimization of machining parameters to minimize surface roughness using integrated ANN-GA approach. Proc. CIRP **29**, 305–310 (2015)
20. Jampana, V.N.R., Ramana Rao, P.S.V., Sampathkumar, A.: Experimental and thermal investigation on powder mixed EDM using FEM and artificial neural networks. Adv. Mater. Sci. Eng. Article ID 8138294 (2021)
21. Avanzato, R., Bertelli, F.: A CNN-based differential image processing approach for rainfall classification. Adv. Sci. Technol. Eng.Syst. **5**(4), 438–444 (2020)
22. Islam, M.T.: Plant disease detection using CNN model and image processing. Int. J. Eng. Res. Technol. **9**(10), 291–297 (2020)
23. Mahewaradedy, M.T., Vamsi, B.K., Nagamalleswari, T.Y.J.: CNN for image processing to detect weeds using IOT. Int. J. Psychosoc. Rehabil. **24**(8), 1080–1087 (2020)
24. Siddiqua, R., Rahman, S., Uddin, J.: A deep learning-based dengue mosquito detection method using faster R-CNN and image processing techniques. Ann. Emerg. Technol. Comput. **5**(3), 11–23 (2021)
25. Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L.: Review of deep learning: concepts CNN architectures, challenges, applications, future directions. J. Big Data **8**, 53 (2021)
26. Lee, H., Song, J.: Understanding recurrent neural network for texts using English-Korean corpora. Commun. Stat. Appl. Methods **27**(3), 313–326 (2020)
27. Al-Shaher, M.A.: A hybrid deep learning and NLP based system to predict the spread of Covid-19 and unexpected side effects on people. Period. Eng. Nat. Sci. **4**(2), 2232–2341 (2020)
28. Gurunath, R., Alahmadi, A.H., Samanta, D., Khan, M.Z., Alahmadi, A.: A novel approach for linguistic steganography evaluation based on artificial neural networks. IEEE Access **9** (2021). https://doi.org/10.1109/ACCESS.2021.3108183
29. Yi, D., Bu, S., Kim, I.: An enhanced algorithm of RNN using trend based on artificial neural networks. IEEE Access **8** (2021). doi.org/10.1109/ACCESS.2021.3108183
30. Zhang, X., Liu, Y., Wu, X., Niu, Z.: Intelligent pulse analysis of meteorological data. Big Data Res. **27**, 11 (2021)
31. Yu, W., Kim, I.Y., Mechefske, C.: Analysis of different RNN autoencoder variants for time series classification and machine prognostics. Mech. Syst. Signal Process. **149**, 107322 (2021)
32. Zhang, X., Liu, Y., Wu, X., Niu, Z.: Intelligent pulse analysis of high-speed electrical discharge machining using different RNNs. J. Intell. Manuf. **31**(4), 937–951 (2020)
33. Lee, W.K., Abdullah, M.D., Ong, P., Abdullah, H., Teo, W.K.: Prediction of flank wear and surface roughness by recurrent neural network in turning process. J. Adv. Manuf. Technol. **15**(1), 55–67 (2021)
34. Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., Zhang, H.: Deep learning with long short-term memory for time series prediction. IEEE Commun. Mag. 57(6), 114–119 (2019)
35. Hochreiter, S., Schmidhuber, J.: Long short term memory. Neural Comput. 9(8), 1735–1780 (1997)
36. Karita, S., Ogawa, A., Delcroix, M., Nakatani, T.: Forward-backward convolutional LSTM for acoustic modeling. In: Proceedings of the Annual Conference of the International Speech Communication Association, Stockholm, Sweden, pp. 1601–1605 (2017)
37. Dai, S., Li, L., Li, Z.: Modeling vehicle interactions via modified LSTM models for trajectory prediction. IEEE Access 7,(2019). https://doi.org/10.1109/ACCESS.2019.2907000
38. Yongsheng, D., Fengshun, J., Jie, Z., Zhikeng, L.: A short-term power output forecasting model based on correlation analysis and ELM-LSTM for distributed PV system. J. Electrical Comput. Eng. Article ID 2051232 (2020)
39. Li, Q., Zhou, S., Wang, Z.: Quantitative risk assessment of explosion rescue by integrating CFD modeling with GRNN. Process Saf. Environ. Prot. 154, 291–305 (2021)
40. Chen, Y., Shen, L., Li, R., Xu, X., Hong, H., Lin, H., Chen, C.: Quantification of interfacial energies associated with membrane fouling in a membrane bioreactor by using BP and GRNN artificial neural networks. J. Colloid Interface Sci. 565, 1–10 (2020)
41. Chen, H.M., Tsai, K.H., Qi, G.Z., Yang, J.C.S., Amini, F.: Neural network for structure control. J. Comput. Civ. Eng. 9(2), 168–176 (1995)
42. Mucha, W.: Comparison of machine learning algorithms for structure state prediction in operational load monitoring. Sensors 20(24), 7087 (2020)
43. Bhattacharya, S., Das, P.P., Chatterjee, P., Chakraborty, S.: Prediction of responses in a sustainable dry turning operation: A comparative analysis. Math. Prob. Eng. (2021). https://doi.org/10.1155/2021/9967970
44. Iglewicz, B., Martinez, J.: Outlier detection using robust measure of scale. J. Stat. Comput. Simul. 15(4), 285–294 (1982)
45. Kalita, K., Dey, P., Halder, S.: Search for accurate RSM metamodels for structural engineering. J. Reinf. Plast. Compos. 38(21–22), 995–1013 (2019)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.