



Simulating the response of ionization chamber system to ^{137}Cs irradiator using the artificial neural network modeling algorithm

A. M. Sadek¹ 

Received: 9 December 2019 / Accepted: 19 June 2020 / Published online: 3 July 2020
© Springer Nature Switzerland AG 2020

Abstract

The artificial neural network algorithm has been used to simulate the response of an ionization chamber dosimetric system to the ^{137}Cs irradiator facility at the National Institute for Standards, Egypt. The performance of the designed model has been investigated over different processes functions and a wide range of parameters till the optimum performance has been approached. The standard uncertainty of the designed model has been evaluated for different dose-rate levels. It has been found that the dose-rate can be evaluated using the designed model with an uncertainty $< 2.0\%$ for the dose-rate level $< 10^2$ mGy/h and with an uncertainty $< 0.5\%$ for the dose-rate level $[10^2 - 10^3]$ mGy/h. The bias of the designed model has been evaluated and compared with the usual interpolation algorithms. The errors in the evaluated dose-rate using these methods have exceeded the 100% error level and approached the 20% error level for the dose rates less and greater than 10 mGy/h, respectively. While, the error in the evaluated dose-rate using the designed model did not exceed 7% and 2% for the same dose-rate levels, respectively.

Keywords Artificial neural network · Ionization chamber system · Interpolation fitting methods

1 Introduction

In radiation dosimetry, it is sometimes difficult to avoid the theoretical calculation for the absorbed dose. However, the accuracy of the estimated dose is always required. Therefore, theoretical methods such as the Monte Carlo (e.g. [8, 25]), have been used in numerous applications. However, the using of the Monte Carlo algorithm in radiation dosimetry requires a fully computational design for the experimental system including the geometrical properties of the radiation source, the geometrical properties of the irradiator facility, considering all the possible physical interactions of the radiation particles with the surrounding medium, and having the required library for the interaction cross-section information [55]. Thereby, the complexity and difficulties of using this method restrict its use especially in the routine work [60]. On the other

hand, the artificial neural network (ANN) is an intelligence method that is based on establishing a connection between the input and the output data for nonlinear system without the necessity to know how the system works internally [26]. Therefore, the ANN can lead to much lower computational time [53].

The interpolation fitting algorithms have been widely used in the field of radiation dosimetry. Baltas et al. [4] recommended a simple linear interpolation fitting. On the other hand, others [19, 33] argued that the cubic spline interpolation method is more accurate than the linear interpolation method. Nevertheless, the simplicity of the linear interpolation fitting method is always a point of strength [39]. In fact, the interpolation methods in most cases are limited to estimate the data within the measured values. Therefore, Baltas et al. [4] advised that special care

✉ A. M. Sadek, dr_amrsadek@hotmail.com | ¹ Ionizing Radiation Metrology Department, National Institute for Standards, El-Haram, Giza, Egypt.



shall be taken into accounting for evaluating the dose rate values for positions outside the range of the original data.

The artificial neural network (ANN) modeling technique aims to find a relationship between the input and the target. This technique has been applied to predict the output response of certain system e.g. [1, 20, 42, 54, 57]. A comparison between the ANN and the regression models has been also investigated [30]. However, there is a different between the two techniques. The ANN does not produce a mathematical expression that related the input to the output, but rather a black-box of numerous mathematical processes that attempt to capture the way the output changes with changing the input [9, 31]. Thus, ANN model consists of connected input layers, hidden layer(s) and output layer. The dimension of each layer is described by the number of neurons nodes. The connection strength between the nodes is modeled as synaptic weight. The main idea of the ANN modeling technique is to adjust the values of the synaptic weight so that the error of the network output (i.e. the deviation of the output from the target) is minimized [23]. The optimization process is established based on certain “learning rules”.

The back-propagation (BP) learning rules have shown a quite robust algorithm to minimize the error function [34]. In this algorithm, the learning process is conducted on two sequential processes. In the first process (called *forward* process), the error in the network output is computed using initial values of the synaptic weight. The initial values are being selected using certain random-based initialization function. The second process is the “backward pass” in which the error of the output is passed backward to the network to adjust and update the synaptic weight values so that the error of the network output is minimized.

The ANN modelling algorithm has been used to evaluate the dose distribution and the dose uniformity in the Tunisian gamma irradiator [37] where the measurements were carried out using polymethyl methacrylate dosimeters. However, the ionization chamber dosimetric system is a benchmark of dosimeter, and had been known to provide the most accurate and reliable results [41]. Therefore, in the current work, the ionization chamber system has been used to measure the radiation dose-rate. The aim of the current work is to:

1. Build a black-box model using the ANN algorithm to simulate the response of an ionization chamber system in measuring the radiation dose-rate of ^{137}Cs .
2. Compare between the accuracy of the ANN algorithm and the accuracy other usual interpolation methods.

2 Experimental data

The dose rate of ^{137}Cs irradiator manufactured by Atomic Energy of Canada LTD (Model No. G.B. 150) was experimentally measured using the secondary standard dosimetry system of the National Institute for Standards (NIS)—Egypt. The dosimetry system consists of TM-30013 ionization chamber manufactured by PTW and calibrated at the International Atomic Energy Agency (IAEA). The ionization-chamber is connected to PTW electrometer calibrated at the Bureau International des Poids et Mesure (BIPM).

The dose-rate was recorded for different positions from the radiation source using the ionization-chamber system. Each position is designated by x - y coordinates as illustrated in Fig. 1. The replicate number of each dose-rate measured by the ionization chamber is 10.

The radiation dose-rate was recorded at 190 different positions. The dose-rates at 146 different positions have been used to build-up the computational model (data for training) and the dose rates measured at the other 25 positions have been used to examine the accuracy of the model (data for validating). The data used in the validation process were randomly selected to cover the whole range of the experimental results as illustrated in Fig. 2.

3 Computational algorithms

3.1 Classical interpolation methods

In the current section, the curve interpolation methods using linear, cubic spline, nearest-neighbor and thin-plate

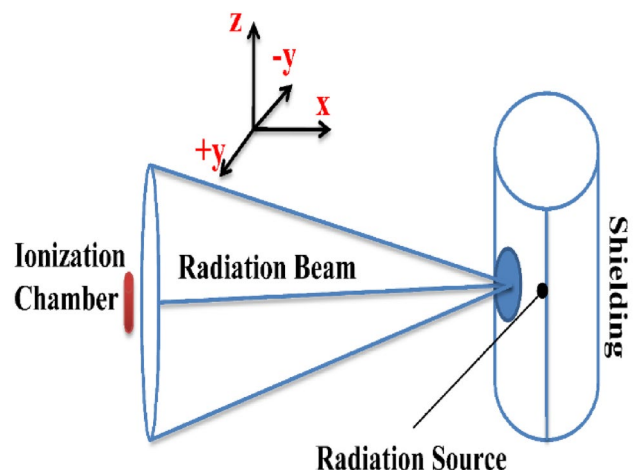


Fig. 1 Schematic representation describing the x - y coordinates of the ionization chamber from the radiation source

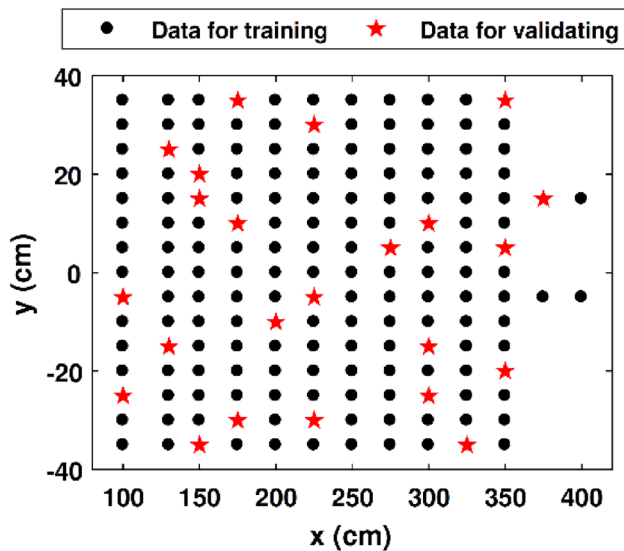


Fig. 2 Selection of the data used in validation process to cover the range of the experimental results

spline algorithms have been used to fit the experimental data. The fitting results are presented in Fig. 3.

It is obvious from the figure that the interpolation fitting using “linear” and “cubic-spline” can only fit the data points the lie within experimental results. In the “nearest-neighbor”, no new data points are generated, the method sets the fitted value to the nearest experimental value. However, it is obvious that the interpolation fitting using “thin-plate” method could fit all the experimental data.

3.2 Artificial neural network (ANN) model

3.2.1 Assigning the functions and the parameters of the ANN

The processes functions and parameters of the ANN are dependent on the nature of the problem [44]. Hence, there is no universal design for the ANN that is applicable for any (or even a certain) type of experimental data [49]. Consequently, approaching the optimum design for the ANN is mainly a trial and error procedure [45] and also on the examples and experiences provided by the user [53]. Thus, the optimum design for the ANN can be proposed by varying the number of hidden layers, transfer function and the number of neurons in each hidden layer [17, 56]. The efficiency of the ANN can be validated throughout several validation methods. Most of these methods aim to assess the difference between the ANN output and the experimental results, i.e. the error function. The linear regression analysis is one of the efficient methods that can measure the performance of the network [48]. However, monitoring the performance of the ANN during the learning processes

is also a crucial factor [44]. These subjects have been discussed in details in Sects. 4.1 and 4.2.

In order to approach the optimum ANN design, the performance of the network has been examined for different neurons number and hidden layers. Furthermore, the performance of the network has been monitored for different processes functions and wide range of parameters in such that a generalized optimum performance of the ANN is approached with the simplest design [7]. It has been found that the optimum performance of the ANN has been approached with three hidden layers and a total number of 80 neurons. The processes functions and parameters of each layer is illustrated in Table 1. A graphical representation for the design of the ANN is presented in Fig. 4.

3.2.2 Input, target and output data

The learning process in ANN is performed mainly throughout two stages. In the first stage, the ANN trains on how the input is correlated with the output. While, in the second stage the ANN validates the generality of the training process. Therefore, the experimental data are divided into two groups; one group for the training stage and one group for the validation stage. It is common to divide the data into 70% and 30% for the training and the validation processes, respectively [59]. The validating data are used by the network to ensure the generality of the network and preventing from falling into a local optimum [22]. However, occasionally, the 30% of the data is further divided into 15% for validation and 15% for testing. The testing process examines the performance of the network for new different data, i.e. data were not included in the training process.

In the current work, the input of the ANN is the x–y position coordinates (in cm) of the ionization chamber with respect to the radiation source. While, the output is the dose rate in mGy/h. The input and output data have been transformed into scaled-range data before the ANN uses them in the training stage. Indeed, using scaled-range data significantly improves the performance of the ANN [2]. The transfer functions used in the current work have been further discussed in Sect. 3.2.5.

3.2.3 Hidden layers and neurons size

The design of any ANN consists of one input layer, one or several hidden layers, and one output layer. For each layer, the dimension of the layer (number of neurons), initial function, net input function and the transfer function should be assigned by the user in accordance with the general criteria discussed in Sect. 3.2.1.

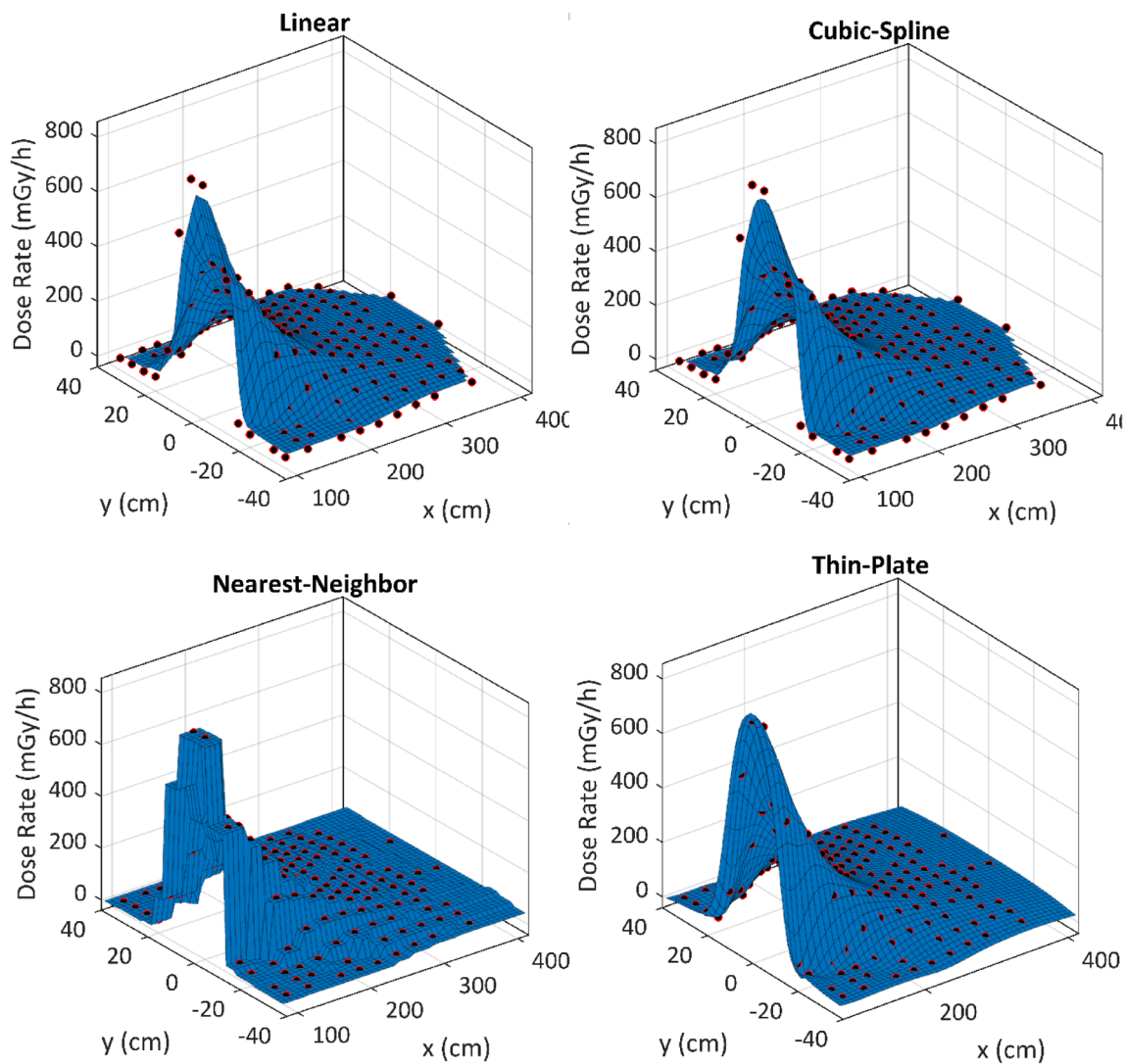


Fig. 3 Interpolation fitting using linear, cubic-spline, nearest-neighbor and thin-plate algorithms for the experimental data results obtained using ionization chamber system

In multiple hidden-layer ANNs, each layer applies different process function on the output of the previous layer [44]. Thus, the output of the first hidden layer is the input of the second hidden layer. Usually, a NNN with two hidden layers can approach the optimum performance [3]. However, in the current work, the optimum design has been reached with three hidden layers. In order to determine the neurons number in each hidden layer, the performance of the network has been monitored over different layers' sizes. The following procedure has been followed:

1. More than 30 different designs for the ANN have been set up. In each design, the dimensions of the three hidden layers have been randomly selected from a specified range of [2, 50].

2. For each design, the weights and biases values have been reinitiated to ensure the generality of the learning process.
3. Step no. 2 has been repeated 10 times for each design and the minimum mean square error (*mse*) has been evaluated.
4. The mean value of the *mse* and its standard uncertainty $u(mse)$ have been monitored over the different designs of the ANN. The results are presented in Fig. 5.

It is obvious from the figure that the optimum performance of the network can approach $mse < \sim 1$ with a total number of 80 neurons distributed over the three hidden layers. It was reported that the number of neurons should not be unnecessarily large or unnecessarily small to avoid the overfitting or the lack fitting,

Table 1 Input functions and parameters used in the designed ANN shown in Fig. 4

Object	Sub-object	Specifications
Input data	Size	2
	Processing function	Minimum–maximum mapping
	Dividing function	Random
1st hidden layer	Dimension	30
	Initialization function:	Nguyen–Widrow algorithm
	Net input function	Summation
	Transfer function	<i>tansig</i>
2nd hidden layer	Dimension	30
	Initialization function:	Nguyen–Widrow algorithm
	Net input function	Summation
	Transfer function	Log-sigmoid
3rd hidden layer	Dimension	20
	Initialization function:	Nguyen–Widrow algorithm
	Net input function	Summation
	Transfer function	Tan-sigmoid
Output layer	Dimension	1
	Initialization function:	Weight/bias function
	Net input function	Summation
	Transfer function	linear
Learning process	Learning mode	Batch mode
	Training function	Levenberg–Marquardt optimization
Optimization processing	Maximum epochs	1000
	Maximum training time	Inf
	Performance goal	0
	Minimum gradient	10^{-7}
	Maximum validation check	0
	μ -Factor	0.005
	μ -Decrease ratio	0.1
	μ -Increase ratio	10
	μ -Maximum	10^{10}

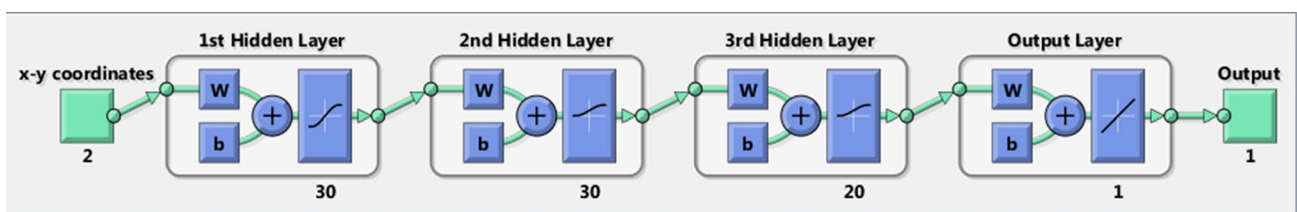
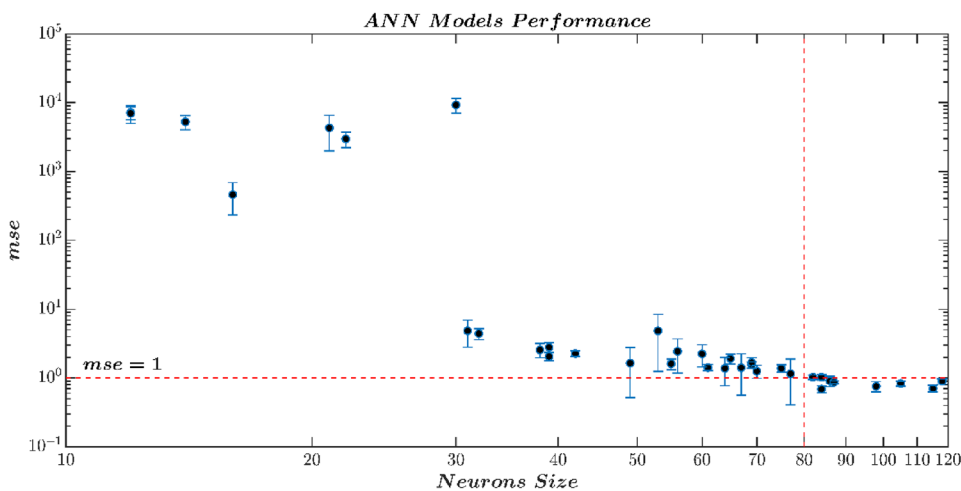


Fig. 4 Architecture representation for the final design of the artificial neural network used in the current work

respectively [7, 61]. It is obvious from the figure that increasing the neurons size more than 80 does not significantly improve the performance of the network. On the other hand, it significantly increases the computational time required for convergence. It is worthwhile to mention that it is not necessarily the neurons should be equally distributed over the three hidden layers. However, the total number of the neurons in the three hidden layers should not be < 80 .

The optimal weights and biases are obtained by training the ANN so that the error between the ANN output and the target is minimized. Therefore, unrealistic small or high weight values influence the performance of the network. Therefore, substantial improvements have been established to develop a method for selecting the appropriate initial values for the weights and biases that prevent the network for being stuck into a local minima. In the current work, the Nguyen–Widrow algorithm [43] has

Fig. 5 Performance of different ANN designs over different neurons sizes distributed randomly over three hidden layers



been applied. It was found that this method is convenient [62] and preferable than the random generation method [14, 47]. In this method, the initial values of the weights and biases are selected so that the active regions of the transfer functions are used.

3.2.4 Activation functions

For nonlinear input–output experimental data, the neurons within the layers are processed through a nonlinear activation function f to obtain the output a . Thus, for inputs vector X^T and weights W and bias vector b^T , the output is given as [44],

$$X^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \quad W = \begin{bmatrix} W_{11} & W_{21} & W_{31} & \dots & W_{r1} \\ W_{21} & W_{22} & W_{32} & \dots & W_{r2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{r1} & W_{r2} & W_{r3} & \dots & W_{rn} \end{bmatrix},$$

$$b^T = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_r \end{bmatrix} \rightarrow a = f(X^T W + b^T) \tag{1}$$

where n and r are the numbers of the inputs and neutrons, respectively and f is an appropriate transfer function. It has been reported that the multilayer networks typically use sigmoid transfer functions in the hidden layers [12]. The (*logsig*) and (*tansig*) sigmoid transfer functions are defined by Eqs. (2) and (3), respectively.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

Sigmoid Transfer Functions

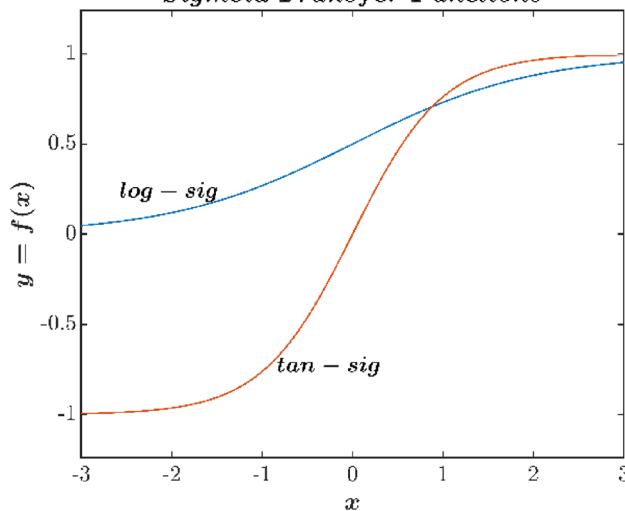


Fig. 6 A representation for the log-sigmoid and tan-sigmoid transfer functions used for the hidden layers

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

In fact, the tan-sigmoid function is a shifted and scaled version from the log-sigmoid function. The two functions are represented in Fig. 6.

The sigmoid functions transfer the input variable x to a scaled-range variable y . The output variables of the log-sigmoid and tan-sigmoid transfer functions are scaled to $[0, 1]$ and $[-1, 1]$, respectively. Furthermore, these two sigmoid transfer function are differentiable and thereby they fulfill the requirements of the back-propagation neural network [58]. However, there is no trustworthy evidence that supports applying these functions to any non-linear problem type [15]. Therefore, several trials have been carried out using different transfer

functions. However, the optimum performance has been addressed using a log-sigmoid transfer functions for the first and second hidden layer, and tan-sigmoid transfer function with the third hidden layer. Of course, a linear transfer function was used for the output layer so that the output of the neural network will not be restricted to a finite range.

3.2.5 Training function and learning process

The backpropagation algorithm propagates the error back to the network in order to adjust the values of the weight and biases so that the error is minimized. Occasionally, the weights and bias are adjusted using the gradient-descent technique [16]. In this technique, the weight adjustment is proportional to the negative gradient of the error with respect to the weight according to [10],

$$W_{k+1} = W_k - \eta \mathcal{G}_k \quad (4)$$

where W_k is the weights and biases vector at epoch k , $\mathcal{G}_k = \partial E_k / \partial W_k$ is the gradient of the error function E with respect to the weight W_k function, and η is the learning rate. Hence, the ANN approaches its optimum performance as the first derivative of the error function with respect to the weight approaches the zero, i.e. $|\mathcal{G}| \rightarrow 0$. However, updating the weights and biases based on this algorithm may lead to a local minimum. Therefore, Newton's methods [5, 6] were developed based on the second-derivative Hessian matrix of the error function with respect to weights/biases as

$$W_{k+1} = W_k - A_k^{-1} \mathcal{G}_k \quad (5)$$

where A_k is the Hessian second derivatives of the error with respect to the weight W_k . However, deducing the second-derivative of the error with respect to the weight would cause extensive computational time. Therefore, Levenberg–Marquardt suggested an approximated version of this method in which the Hessian matrix can be represented as [11, 36, 40]:

$$\mathcal{H} = J^T J \quad (6)$$

where J is the Jacobian matrix consisting of the first derivative of the error function with respect to the weight. In this case, the gradient can be computed as:

$$\mathcal{G} = J^T e \quad (7)$$

Thus, the weights and biases are being updated using [11]:

$$W_{k+1} = W_k - [J^T J + \mu I]^{-1} J^T e \quad (8)$$

where μ is the damping factor and I is the identity matrix. When μ is zero, Eq. (8) turns into Newton's method using

approximated Hessian. At large μ values, the Newton's method turns into the ordinary gradient-descent method with small step size. Because Newton's method is more efficient, μ starts with relatively large initial value, and decrease for each successful step in which the error decreases.

In the ANNs with multiple non-linear hidden layers, the overfitting and lack of generalization are very common problems [24]. This includes stopping the training as soon as performance on validation set starts to get worse. In this case, the network will memorize the training data, but still cannot interpret the new data. The best way to overcome this problem is to regularize a fixed-sized model that considers the average of the predictions resulted from the various settings of the network parameters [52]. Henceforth, the necessity of a regularization method which adapts the selected parameters and ensures the generalization of the ANN is inevitable [29]. Typically, the regularization function F_{reg} should be considered in evaluation of the network performance instead of the usual mean square error mse function. Introducing the regularization γ , the regularization function is defined as [11],

$$F_{reg} = \gamma(mse) + (1 - \gamma)(msw), \quad msw = \frac{1}{r} \sum_{i=1}^r w_i^2 \quad (9)$$

It is clear that the determination of the regularization parameter γ value is the obstacle to using this method. Large γ values would cause in over fitting, and low values would cause in poor fitting. Bayesian regularization [35] was developed to automatically determine the optimal regularization parameters. In this technique, the weights and biases are considered random variable with certain distributions and are being updated using the Levenberg–Marquardt. The efficiency of this technique has been proven in difficult optimization problems [46], and therefore, it was implemented in the ANN toolbox MATLAB [21]. On the other hand, the Bayesian regularization function takes a long time compared with the usual Levenberg–Marquardt algorithm before it convergences [46]. Consequently, several parameters have been set in the current study to ensure that the network approaches the global minimum as follows:

1. The maximum epoch number has been set to 10^3 and the maximum training time has been expanded to infinity. This ensures the network will not stop because of insufficient epoch number or training time.
2. The target of the performance function F_{reg} has been set to 0 to approach the best possible solution.
3. The minimum gradient limit has been set to 10^{-7} to approach the lowest change in the error function with respect to the weights and biases.

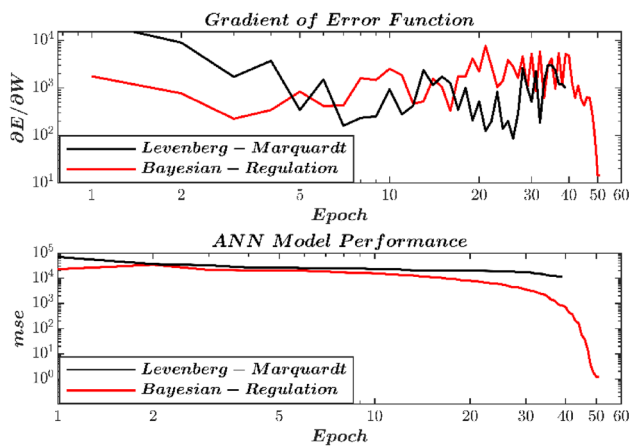


Fig. 7 Performance of the ANN model designed using the Levenberg–Marquardt and Bayesian-Regulation training functions

4. The damping factor μ has been initially set to 0.005 and the maximum limit is 10^{10} with decreasing and increasing rates of 0.1 and 1.0, respectively. This is to ensure a wide range for the damping factor.

A comparison between the performance of the network using the usual Levenberg–Marquardt and using the Bayesian-regularization is illustrated in Fig. 7.

It is obvious that the lower levels of the gradient $\partial E/\partial W$ and the mean square error mse could be reached using the Bayesian-Regulation training function than the case of using the usual Levenberg–Marquardt training function. It is worthwhile mentioning that the performance of the network has been also examined using other different training functions. However, the Bayesian-Regulation function still provides the optimum performance.

A way to ensure the generalization of the ANN is to monitor the performance of the network in both the training and testing stages. It is anticipated that the behavior of the ANN performance would be approximately identical in the two stages [38]. Therefore, in order to examine the generalization of the network, the data have been divided into two data sets. The first data set (70%) has been utilized in training the network, while the second data set (30%) has been used in testing the network. The results are reviewed in Sect. 4.1.

The incremental and batch modes are two algorithms used to update the values of the weights and biases in each layer. The incremental mode updates the weights and biases simultaneously after applying each input to the network. While the batch mode updates the weights and biases after applying all the input to the network [50]. In general, the incremental mode superiors the batch mode [32, 51] and it is recommended when the training data

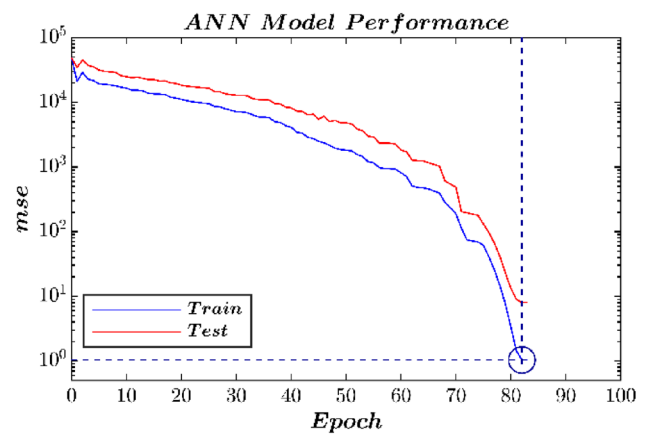


Fig. 8 Performance of the artificial neural network described in Table 1 for the train and test data set

set is large [13]. However, this is not the case in the current study. Furthermore, two modes become identical at sufficiently small learning rate [18]. Hence, in the current study, batch mode was implemented as the initial learning method in the design of the neural network.

3.2.6 Software used

In the current study, the MATLAB (R2019a) has been used to build up the artificial neural network via the deep learning Toolbox. This toolbox provides a framework for designing and implementing deep learning networks with algorithms and retrained models. The processing functions and parameters required to design the ANN model are divided mainly into four input categories. The first and second input categories describes the dimensions of the ANN model including the size of the input and the output, the number layers, and the number of the neurons in each layer. They also specify the mathematical operations to be applied on the input and the output before the network starts the learning process. The third and fourth input categories concern the learning and the optimization processes. In these categories, all the connections between neural network components should be described via the synaptic weights and bias options. Moreover, the functions and the parameters of the learning and optimization processes should be determined.

4 Results and discussion

4.1 The performance of the artificial neural network

In the current study, the global optimization of the ANN model design has been validated throughout three evidences;

1. The performance of the network using the training data is similar to the performance of the network using the testing data (Fig. 8).
2. The linear correlation coefficients between ANN output and target are $R > 0.99$ for both the training and testing stages (Fig. 9).
3. The error of the ANN model design has a normal distribution (Fig. 10).

The performances of the network in case of using the training data and testing data have been monitored over the epochs (Fig. 8). The most important conclusion in Fig. 8 is that there is no significant difference between the performance of the ANN model in case of the testing process and in case of the training process. This indicates the generality of the ANN learning process and the designed model can be used for predicting new data.

The ANN output and the target values have been also subjected to linear regression analysis (Fig. 9) and the correlation coefficient has been evaluated for the training data, test data, and all the data. It has been found that all the correlation coefficient values were $R > 0.99$ indicating a good linearity between the ANN output and the target values. This also implies that the performance

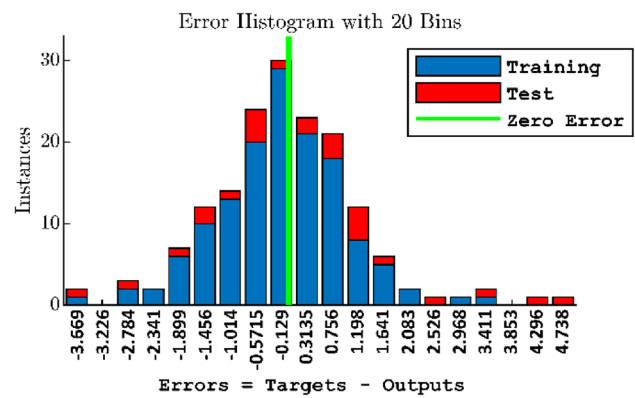


Fig. 10 Distribution of the ANN output error k for the training- and test-data sets

of the ANN model design for the new data is similar to the performance of model for the training data.

Usually, the error in the experimental measurements follows the Gaussian distribution with mean value zero. The error in the ANN output can be calculated as the difference between the target and the ANN output. The distribution of the ANN output error is presented in Fig. 10. It appears that the error of the ANN output has a Gaussian distribution with mean value around zero. This conclusion can be reported for both the training and testing processes. This also supports the evidence that the performance of the ANN model design has approached a global minimum. It also ensures the generality of the learning process of the ANN model design.

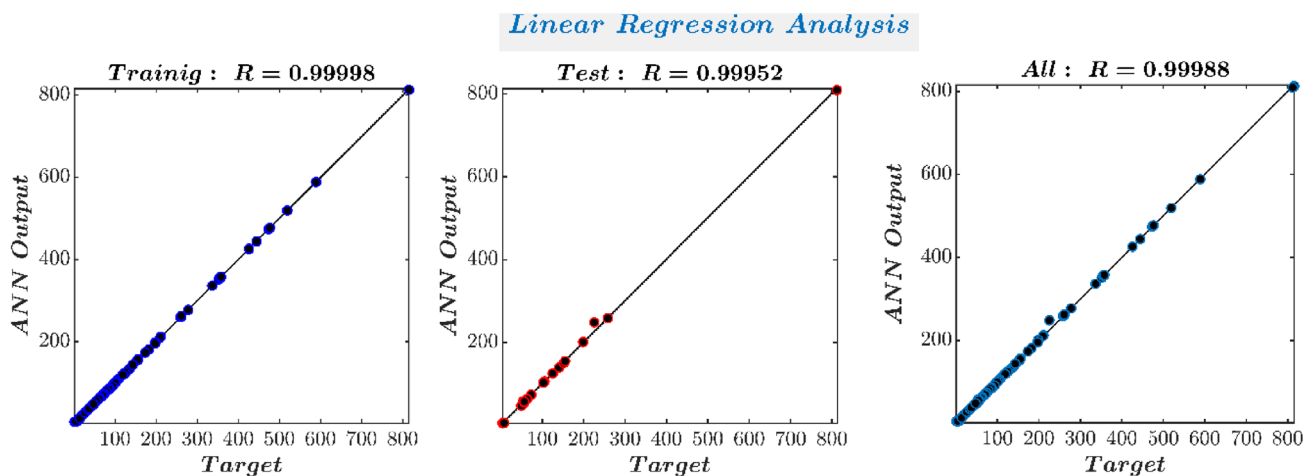


Fig. 9 Linear regression analysis for ANN output and the target values in cases of training, testing and all data. The correlation coefficients have been denoted in each case

4.2 The standard uncertainty of the designed ANN model

One of the features of the ANN algorithm is that the solution is never exact. Indeed, each time the weights and biases are initiated, the ANN model would have a different solution. However, it would be useful to estimate the standard uncertainty of the ANN output for certainty replicated number. This standard uncertainty provides the range in which the ANN output should possible lie. However, in order to have a reliable standard uncertainty, the replicate number should not be < 30 [28]. Hence, the standard uncertainty of the ANN model design has been evaluated for 30 replicated number. The standard uncertainty has been estimated as [28]:

$$u(x) = \frac{s(x)}{\sqrt{n}} \tag{10}$$

where $s(x)$ is the sample standard deviation, and n is the replicated number. This process is the same as conducting experimental measurements under the repeatability conditions. Figure 11 presents the percentage standard uncertainty of the evaluated dose rate predicted using the designed ANN mode at different dose-rate levels.

It is obvious that the dose rate can be calculated with uncertainty < 2% in the dose-rate level [10–100] mGy/h and with uncertainty less than 0.5% in the dose-rate level [10²–10³] mGy/h. For simplicity, an uncertainty of 2% can be used then for any value estimated from the ANN model design. However, it is always recommended to specify the uncertainty for each level or point for the measurand [28].

4.3 The bias in the designed ANN model

The bias in the ANN model designed has been estimated over different dose-rate points. These new points have not

Fig. 11 Uncertainty of the ANN output as a function in the dose-rate level

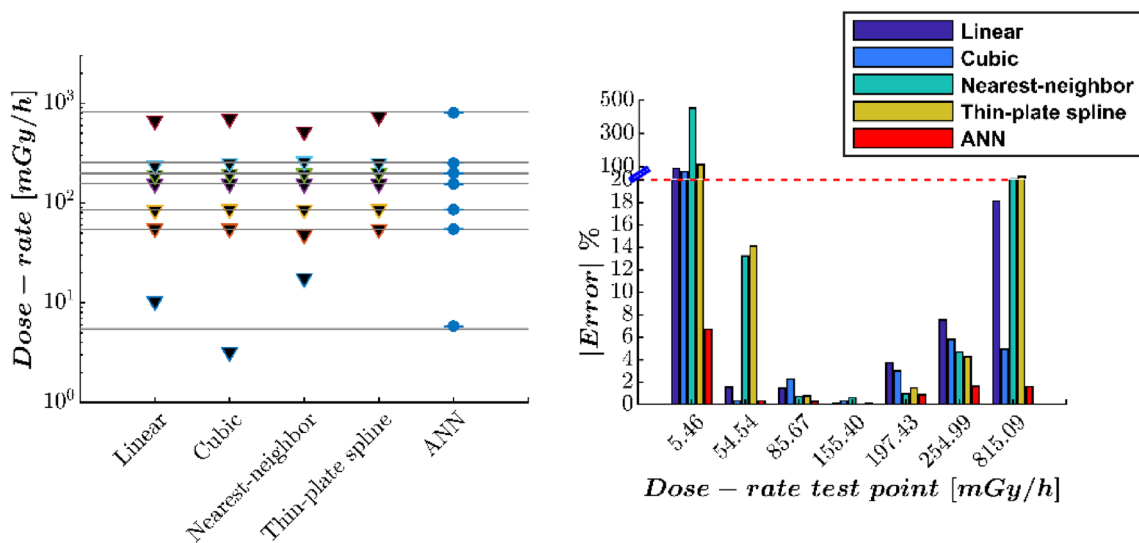
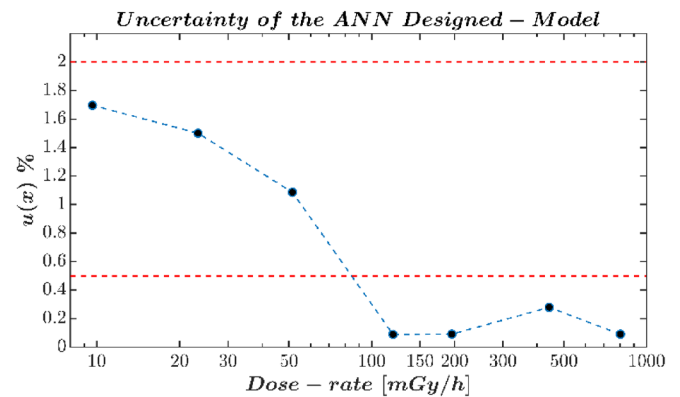
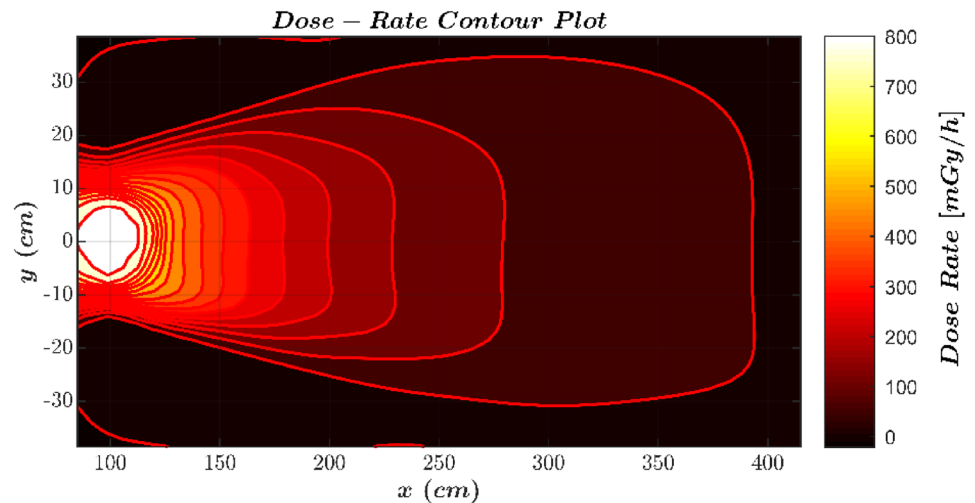


Fig. 12 Comparison between the accuracy of the output of the artificial neural network and the other interpolation algorithms

Fig. 13 A contour plot for the dose-rate computed by the artificial neural network model. The dose-rate at each point has been represented by the average of 100 runs. In each run, the ANN parameters have been reinitialized



been used either in the training or validating processes. Furthermore, they have been chosen to cover the range of the experimental data. The dose-rate at these additional test points have been also measured by the ionization chamber system. A comparison between the accuracy of the results obtained by the neural network and the other interpolant algorithms is illustrated in Fig. 12.

A draw-back in the accuracy of the interpolation fitting methods can be clearly observed at the extreme low and high dose-rate. At low dose-rate, the error in the output computed by the interpolation fitting algorithms approached the 100% and even 500% levels. At the high dose-rate, the error in the output computed by the interpolation methods approached the 20% and in some cases approached the 50% levels. This is due to the fact that these methods estimate the dose-rate using the lines connecting neighboring evaluated dose points which is difficult to be determined at the extreme dose-rate values. Thereby, the interpolation fitting techniques can be used essentially to estimate values within the range of the fitted data [27]. Furthermore, the accuracy of these interpolation fitting methods varies over the different dose-rate level. Hence, selecting the appropriate interpolation fitting method would be a challenging task.

On the other hand, the artificial neural network model has shown a superiority over the interpolation fitting methods. This could recover the draw-back in the accuracy of the interpolation fitting method and succeeded to estimate the dose-rate at the extreme low- and high dose-rate levels with error not exceeding the 7% and 2% levels, respectively. At the intermediate dose-rate levels, the accuracy of the estimated dose-rate values was likewise better than the accuracy of the interpolation fitting method.

Once the accuracy of the neural network has been validated, it can be used for different dosimetric applications.

A contour plot for the distribution of the dose-rate simulated using the artificial neural network model is shown in Fig. 13.

It worthwhile to mention that each dose-rate point in the contour plot has been evaluated under the repeatability conditions discussed in Sect. 4.2. The symmetry of the contour plot is observed at all the dose-rate levels. In other words, there was no unexpected behavior for the ANN model over the different dose-rate level. This, in fact, supports the reliability of using the designed model to estimate the dose-rate value at any point in the geometrical field.

The ANN technique can be used to simulate and evaluate the dose distribution in human phantom which is widely used in diagnostic and therapeutic medical applications. Usually, the Monte Carlo algorithm is used for this purpose. However, it has been discussed previously the complexity of this algorithm. Therefore, the ANN modeling technique is expected have a great benefit in diagnostic and therapeutic applications.

5 Conclusions

- The ANN modeling algorithm could successfully simulate the response of an ionization-chamber system to irradiator source.
- The optimum performance of the ANN model could be approached with three hidden layers and a total number of 80 neurons.
- The dose-rate could be evaluated by the designed ANN model with standard uncertainty $< 2\%$ for dose-rates less than 10^2 mGy/h and $< 0.5\%$ for dose-rates greater than 10^2 mGy/h.

- The errors in the evaluated dose-rate using interpolation methods have exceeded the 100% error level and approached the 20% error level for the dose rates less and greater than 10 mGy/h, respectively. While, the error in the evaluated dose-rate using the designed model did not exceed 7% and 2% for the same dose-rate levels, respectively.

Acknowledgements The author is grateful to Prof. Dr. M. Mansy, professor of radiation physics at the ionizing radiation metrology laboratory (IRML), National Institute for Standards (NIS), Egypt, for providing the experimental data used in the current work.

Compliance with ethical standards

Conflict of interest The author declares that he has no conflict of interest.

References

1. Ali SA, Ahmed A (2019) Forecasting MSW generation using artificial neural network time series model: a study from metropolitan city. *SN Appl Sci*. <https://doi.org/10.1007/s42452-019-1382-7>
2. Angelo PL (2015) Representing nuclear criticality excursion experiment data by an artificial neural network. *Nucl Technol* 189:219–240
3. Balaga H, Gupta N, Vishwakarma DN (2015) GA trained hidden layered ANN based differential protection of three phase power transformer. *Electr Power Energy Syst* 67:286–297
4. Baltas D, Sakelliou L, Zamboglou N (2007) The physics of modern brachytherapy for oncology. CRC Press, London
5. Barnard E (1992) Optimization for training neural nets. *IEEE Trans Neural Netw* 3:232–240
6. Battiti R (1992) First and second-order methods for learning: between steepest descent and Newton methods. *Neural Comput* 4:141–166
7. Behnood A, Golafshani EM (2018) Predicting the compressive strength of silica fume concrete using hybrid neural network with multi-objective grey wolves. *J Clean Prod* 202:54–64
8. Bielajew AF, Rogers D (1992) A standard timing benchmark for EGS4 Monte Carlo calculations. *Med Phys* 19:303–304
9. Burduk A (2015) The role of artificial neural network models in ensuring the stability of systems. In: 10th international conference on soft computing models in industrial and environmental applications. Springer, Cham
10. Chattopadhyay S, Chattopadhyay G (2018) Conjugate gradient descent learned ANN for Indian summer monsoon rainfall and efficiency assessment through Shannon–Fano coding. *J Atomo Sol Terres Phys* 179:202–205
11. Chen G et al (2019) Strengths prediction of particulate reinforced metal matrix composites (PRMMCS) using direct method and artificial neural network. *Compos Struct* 223:110951
12. Choong CE, Ibrahim S, El-Shafie A (2020) Artificial neural network (ANN) model development for predicting just suspension speed in solid–liquid mixing system. *Flow Meas Instrum*. <https://doi.org/10.1016/j.flowmeasinst.2019.101689>
13. Cichocki A, Unbehauen R (1992) Neural networks for optimization and signal processing. Wiley, New York
14. Correa DC, Levada A, Satio JH (2008) Improving the learning speed in 2-layered LSTM network by estimating the configuration of hidden units and optimizing weights initialization. In: *Proc ICANN Czech Republic*. Springer, pp 109–118
15. Dixit PM, Dixit US (2008) Modeling of metal forming and machining processes by finite element and soft computing methods. Springer, Berlin
16. Du K, Swamy M (2014) Neural networks and statistical learning. Springer, London
17. Esonye C, Onukuli OD, Ofoefule AU, Ogah EO (2019) Multi-input multi-output (MIMO) ANN and Nelder–Mead’s simplex based modeling of engine performance and combustion emission characteristics of biodiesel–diesel blend in CI diesel engine. *Appl Therm Eng* 151:100–114
18. Finnoff W (1994) Diffusion approximations for the constant learning rate backpropagation algorithm and resistance to local minima. *Neural Comput* 6:285–295
19. Garrido JM (2014) Introduction to computational modeling using C and open-source tools. CRC Press, New York
20. Geroge UA, Elvis MM (2019) Modelling of the mechanical properties of concrete with cement ratio partially replaced by aluminium waste and sawdust ash using artificial neural network. *SN Appl Sci*. <https://doi.org/10.1007/s42452-019-1504-2>
21. Giannetti N et al (2020) Prediction of two-phase flow distribution in microchannel heat exchangers using artificial neural network. *Int J Refrig* 111:53–62
22. Golafshani EM, Behnood A, Arashpour M (2020) Predicting the compressive strength of normal and high-performance concretes using ANN and ANFIS hybridized with Grey Wolf Optimizer. *Constr Build Mater* 232:117266
23. Gurney K (1997) An introduction to neural networks. UCL Press Limited, London
24. Hasson U, Nastase SA, Goldstein A (2020) Direct fit to nature: an evolutionary perspective on biological and artificial neural networks. *Neuron* 105:416–434
25. Holmes MA et al (1993) The application of correlated sampling to the computation of electron beam dose distribution in heterogenous phantoms using the Monte Carlo method. *Phys Med Biol* 38:675–688
26. Honarnezhad R, Fathinia M, Khataee A (2019) Mechanical production and sonocatalytic application of Cu₂S nanoparticles for degradation of isopropylxanthic acid: kinetic modeling via white and black box methods. *J Mol Liq* 287:110899
27. Igor VF (2016) Digital terrain analysis in soil science and geology, 2nd edn. Elsevier, Amsterdam
28. JCGM-100 (2008) Evaluation of measurement data—guide to the expression of uncertainty in measurement. JCGM
29. Khatami A et al (2020) A weight perturbation-based regularization technique for conventional neural networks and the application in medical imaging. *Expert Syst Appl* 149:113196
30. Kitisci M (2019) Comparison of artificial neural network and logistic regression model for factors affecting birth weight. *SN Appl Sci*. <https://doi.org/10.1007/s4245-019-0391-x>
31. Klosowski G, Gola A, Swic A (2016) Application of fuzzy logic in assigning works to production tasks. In: *Distributed computing and artificial intelligence*. Advances in intelligent systems and computing. Springer, Cham
32. Kurihara M, Kondo H (1999) Heuristics and experiments on BDD representation of Boolean functions for expert system in software verification dominas. In: 12th Australian Joint Conference. Springer, Australia
33. Larsen KH, Burch SE (1991) Generation of dose calculation data tables using cubic spline interpolation. *Med Dosim* 16:147–151

34. Lie X, Song S, Lei T (2011) Application of BP neural networks in prediction of the material dynamic properties. Springer, Berlin, pp 68–75
35. MacKay D (1992) Bayesian interpolation. *Neural Comput* 4:415–447
36. Ma C, Long G, Yang Y (2011) Evaluating model on effectiveness of network defense missile based on LMBP neural network. In: 8th international symposium on neural networks. Springer, Berlin
37. Manai K, Trabelsi A (2013) Neural network modelling of dose distribution and dose uniformity in the tunisian gamma irradiator. *Radiat Prot Dosim* 157:125–132
38. Matignon R (2005) Neural network modeling using SAS enterprise miner. AUTHORHOUSE, Bloomington
39. Meirovitch L (2001) Fundamental of vibrations. Waveland Press, USA
40. More JJ (1977) The Levenberg–Marquardt algorithm: implementation and theory. In: Waston, GA (ed) Numerical analysis, lectures notes in mathematics. Springer, Berlin
41. Nalbant N, Kesen D, Hatic B (2014) Pre-treatment dose verification of Imrt using Gafchromic Ebt3 film and 2d-array. *J Nucl Med Radiat Ther* 5:182
42. Naresh C, Bose P, Rao C (2020) Artificial neural networks and adaptive neuro-fuzzy models for predicting WEDM machining responses of Nitinol alloy: comparative study. *SN Appl Sci*. <https://doi.org/10.1007/s42452-020-2083-y>
43. Nguyen D, Widrow B (1990) Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In: 1990 IJCNN international joint conference on neural networks. IEEE, pp 21–26
44. Olcer AI, Farag Y (2020) The development of a ship performance model in varying operating conditions based on ANN and regression techniques. *Occan Eng* 198:106972
45. Ortiz-Rodriguez JM, Martinez-Blanco MR, Viramontes JM, Vega-Carrillo HR (2013) ROBust design of artificial neural networks methodology in neuron spectrometry. In: Artificial neural networks—architecture and applications. InTech, Croatia
46. Ozonoh M, Oboirien BO, Higginson A, Daramola MO (2020) Performance evaluation of gasification system efficient using artificial neural network. *Renew Energy* 145:2253–2270
47. Pavelka A, Prochazka A (2004) Algorithm for initialization of neural network weights. In: Proceedings of the 12th annual conference, MATLAB, pp 453–459
48. Pazhamannil, R. V. & Sooraj, P. G., 2020. Predicting of the tensile strength of polylactic acid fused deposition models using artificial neural network technique. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2020.01.199>.
49. Raju MM et al (2011) Development of artificial neural-network-based models for the simulation of spring discharge. *Adv Artif Intell* 2011:1–11
50. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representation by backpropagating error. *Nature* 323:533–536
51. Sannen D et al (2012) Learning in non-stationary environments: methods and applications. Springer, New York
52. Srivastava N et al (2014) Dropout: a simple way to prevent neural network from overfitting. *J Mach Learn Res* 15:1929–1958
53. Stoffel M, Gulakala R, Bamer F, Markert B (2020) Artificial neural network in structural dynamics: a new modular radial basis function approach vs. convolutional and feedforward. *Comput Methods Appl Mech Eng* 364:112989
54. Thekkuden DT, Mourad AI (2019) Investigation of feed-forward back propagation ANN using voltage signals for early prediction of the welding defect. *SN Appl Sci*. <https://doi.org/10.1007/s42452-019-1660-4>
55. Trabelsi A, Manai K (2013) Neural network modeling of dose distribution and dose uniformity in the Tunisian Gamma irradiator. *Radiat Prot Dosim* 157:125–132
56. Tumer AE (2019) Modeling of trivalent chromium sorption onto commercial resins by artificial neural network. *Appl Artif Intell* 33:349–360
57. Tumer AE, Akkus A (2018) Forecasting gross domestic product per capita using artificial neural networks with non-economical parameters. *Phys A Stat Mech Appl* 512:468–473
58. Ukil A (2007) Intelligent systems and signal processing in power engineering. Springer, Berlin
59. Vieira F, Cavalcante G, Campos E, Taveira-Pinto F (2020) A methodology for data gap filling in wave records using artificial neural networks. *Appl Ocean Res* 98:102109
60. Wu X, Zhu Y (2000) A neural network regression model for relative dose computation. *Phys Med Biol* 45:913–922
61. Xia S, He J, Hongyu C (2011) The study on models adjustment and generation capability of artificial neural network. In: 8th international symposium on neural networks. Springer, China, pp 587–594
62. Zhou S et al (2020) Prediction of the ground temperature with ANN, LS-SVM and fuzzy LS-SVM for GSHP application. *Geothermics* 84:101757

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