


Article

# Predicting the Compressive Strength of Concrete Using an RBF-ANN Model

Nan-Jing Wu 

Department of Civil and Water Resources Engineering, National Chiayi University, Chiayi 699355, Taiwan; njwu@mail.ncyu.edu.tw

**Abstract:** In this study, a radial basis function (RBF) artificial neural network (ANN) model for predicting the 28-day compressive strength of concrete is established. The database used in this study is the expansion by adding data from other works to the one used in the author's previous work. The stochastic gradient approach presented in the textbook is employed for determining the centers of RBFs and their shape parameters. With an extremely large number of training iterations and just a few RBFs in the ANN, all the RBF-ANNs have converged to the solutions of global minimum error. So, the only consideration of whether the ANN can work in practical uses is just the issue of over-fitting. The ANN with only three RBFs is finally chosen. The results of verification imply that the present RBF-ANN model outperforms the BP-ANN model in the author's previous work. The centers of the RBFs, their shape parameters, their weights, and the threshold are all listed in this article. With these numbers and using the formulae expressed in this article, anyone can predict the 28-day compressive strength of concrete according to the concrete mix proportioning on his/her own.

**Keywords:** radial basis functions; artificial neural networks; prediction model; compressive strength of concrete; mix proportioning of concrete



**Citation:** Wu, N.-J. Predicting the Compressive Strength of Concrete Using an RBF-ANN Model. *Appl. Sci.* **2021**, *11*, 6382. <https://doi.org/10.3390/app11146382>

Academic Editor: Stefano Invernizzi

Received: 15 June 2021  
Accepted: 8 July 2021  
Published: 10 July 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Concrete has characteristics of durability, impermeability, fire resistance, abrasion resistance, and high compressive strength. It can be cast into any shape and size. So, it is the most used construction material in the modern world. Its basic ingredients are water, cement, gravel which indicates the coarse aggregate, and sand which indicates the fine aggregate. For some environmental and sustainable considerations, various types of by-product materials are widely used as admixtures in concrete. The most commonly seen are the fly ash and the blast furnace slag. Some may also add other materials such as waste plastic, waste glass, rice husk, etc. [1–6]. Chemical and mineral admixtures might be added to enhance its workability or to change the time interval of the hardened state. The compressive strength is related to the proportioning of its ingredients. For economic reasons such as reducing the construction cost or saving the trial time for the strength requirement, the prediction of the compressive strength according to the proportioning design is a very important research topic. Studies have been implemented with mathematical models [7–12].

In recent decades, a lot of attention has been paid to the application of artificial neural networks (ANNs) in determining the compressive strength of concrete [13–27]. An ANN is like a black box. No theoretical relations between the compressive strength of concrete and the proportioning of its ingredients are needed for establishing an ANN model. The only requirement is the sufficiency of data for training and testing processes. With a well-trained ANN, one can input several numerical values which represent the proportioning of the concrete ingredients such as water, cement, sand, and other admixtures, and the ANN rapidly prompts out the predicted compressive strength of the concrete.

Typically, the database for the application of an ANN has to be divided into two sets, one for training and the other for testing. Usually, the testing set should contain more than

10 % of the database. Otherwise, the ANN has a tendency to over-fit the training set and will perform poorly in practical uses. In [27], an ANN model for predicting the 28-day compressive strength of concrete was established. They used the database of [26] in which an ANN model for the same application was also established. It was presumed that in [26] too little data were used for testing so that the ANN model was barely acceptable in the final validation. In [27], the data were rearranged. Fewer data were for training while more were used for testing. Therefore, the ANN model in [27] outperforms the one in [26], and it only has half the number of neurons in the hidden layer.

Both the ANN models in [26,27] are the back-propagation (BP) networks. Besides over-fitting, how to avoid over-training is another important issue when employing the BP-ANN. The ANN usually fits the training set better and better as more and more iterations of training are processed. However, it does not guarantee the over-trained ANN will also perform as well in practical uses. Because the training procedure of BP-ANN is extremely time-consuming, it is difficult to determine how many iterations of training are too excessive. In addition, it sometimes converges to a state of local minimum error.

Apart from the BP-ANN, there are many other types of ANNs. One of them is the Radial Basis Function (RBF) network. Unlike the BP-ANN, the solution of RBF-ANN is obtained in a rather straightforward way. The RBF centers and their shape parameters are usually determined in advance, then the weights are determined by the least-squares approach for a linear algebraic system in which the solution is unique. The only concern is how to choose the RBF centers and their shape parameters properly. Some may use the k-means approach to set the cluster centers as the RBF centers. Recently, other advanced algorithms have been employed in determining the RBF centers and their shape parameters in the RBF-ANNs which are widely used in various research areas [28–34].

In this paper, an RBF-ANN model for predicting the 28-day compressive strength of concrete is established. The database in [26] which was also used by [27] for establishing their model is expanded by adding more data listed in [35–39]. The Stochastic Gradient Approach presented in [40] is employed for determining the centers of RBFs and their shape parameters. Further validations are implemented by using the data of [25,26]. The results are compared with actual values as well as with the results of [27].

## 2. The Concrete Mix Proportioning

According to [26], the 28-day compressive strength of concrete is mainly related to seven factors. They are the mass of water, cement, fine aggregate (sand), coarse aggregate (gravel), blast furnace slag, fly ash, and superplasticizer mixed in 1 m<sup>3</sup> concrete. Therefore, the 28-day compressive strength of concrete can be expressed as a mathematical function with seven arguments:

$$y = f(x) \quad (1)$$

in which

$$x = [x_1 \quad x_2 \quad \cdots \quad x_7]^T \quad (2)$$

where  $x_1$  to  $x_7$  are the abovementioned proportioning factors while  $y$  is the 28-day compressive strength of concrete. For the input and output of the ANN, all the data have to be normalized into the same range. In this paper, the raw data are transformed into the range of 0 to 1 linearly.

$$\zeta_i = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}} \quad (3)$$

$$\eta = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \quad (4)$$

The database in [26] which was also used by [27] for establishing their model is expanded by adding 89 pieces of data listed in [35–39]. Totally, there are 571 data in the database. The testing set is the same as in [27], so 72 pieces of data are for testing while 499 pieces of data are for training. The size of the testing set is about 12.6% of the database. Printing out all the data might make this paper redundant. Those who need the raw data

may find them in the references [26,27,35–39]. The ranges from  $x_1$  to  $x_7$  and  $y$  are listed in Table 1. It is worthy to note that some of the values listed in Table 1 are slightly different from those in [27]. This is because the database has been expanded by adding data from other literature.

**Table 1.** The ranges of the inputs and the output of the raw data [26,27,35–39].

	Factors	Symbol	Maximum	Minimum	Unit
Inputs	water	$x_1$	295	116.5	kg/m <sup>3</sup>
	cement	$x_2$	643	74	kg/m <sup>3</sup>
	fine aggregate (sand)	$x_3$	1293	30	kg/m <sup>3</sup>
	coarse aggregate (gravel)	$x_4$	1226	436	kg/m <sup>3</sup>
	blast furnace slag	$x_5$	440	0	kg/m <sup>3</sup>
	fly ash	$x_6$	330	0	kg/m <sup>3</sup>
	superplasticizer	$x_7$	27.17	0	kg/m <sup>3</sup>
Output	28-day compressive strength	$y$	95.3	5	MPa

### 3. The RBF Artificial Neural Network

Now we have the function  $\phi$  with 7 arguments:

$$\eta = \phi(\xi) \tag{5}$$

in which

$$\xi = [ \zeta_1 \quad \zeta_2 \quad \dots \quad \zeta_7 ]^T \tag{6}$$

where all the arguments are in the range of 0 to 1. The linear combination of  $n$  RBFs is applied to approximate the input-output relation:

$$\eta \approx \hat{\eta} = w_0 + \sum_{j=1}^n w_j g_j(\xi) \tag{7}$$

in which  $g_j$  is the  $j$ -th RBF and  $w_j$  is its weight while  $w_0$  represents the threshold. The larger value of  $n$  indicates the better the RBF-ANN fits the input-output relation in the training data, but it also has a higher risk of over-fitting. With  $N$  pieces of training data, i.e.,  $N$  sets of  $\zeta_1, \zeta_2, \dots, \zeta_7$  and  $\eta$ , a linear-algebraic system can be formed

$$\mathbf{A}\mathbf{W} = \boldsymbol{\eta} \tag{8}$$

in which

$$\mathbf{A} = \begin{bmatrix} 1 & g_1(\xi_1) & \dots & \dots & \dots & g_n(\xi_1) \\ 1 & \vdots & \ddots & & & \vdots \\ \vdots & \vdots & & g_j(\xi_i) & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ 1 & g_1(\xi_N) & \dots & \dots & \dots & g_n(\xi_N) \end{bmatrix} \tag{9}$$

$$\mathbf{W} = [ w_0 \quad w_1 \quad \dots \quad w_j \quad \dots \quad w_n ]^T \tag{10}$$

$$\boldsymbol{\eta} = [ \eta_1 \quad \eta_2 \quad \dots \quad \eta_i \quad \dots \quad \eta_N ]^T \tag{11}$$

where  $\xi_i$  and  $\eta_i$  represent the input and output of the  $i$ -th piece of data. Equation (8) is a combination of  $N$  linear equations subjected with  $n + 1$  unknowns which are the threshold  $w_0$  and the weights  $w_1, w_2, \dots, w_n$ . The least-square-error solution can be obtained promptly by using the following matrix operation.

$$\mathbf{W} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \boldsymbol{\eta} \tag{12}$$

Among many RBFs, the Gaussian RBF is chosen in this study:

$$g_j(\xi) = \exp\left(-\|\xi - \zeta_j\|^2 / \sigma_j^2\right) \quad (13)$$

in which  $\zeta_j$  represents the center of the  $j$ th RBF

$$\zeta_j = [\zeta_{j1} \quad \zeta_{j2} \quad \cdots \quad \zeta_{j7}]^T \quad (14)$$

while  $\sigma_j$  is the shape parameter and  $\|\xi - \zeta_j\|$  denotes the Euclidean distance from  $\xi$  to  $\zeta_j$ . Ususally, the centers of RBFs and their shape parameters are manually pre-determined. Some may use the k-means approach to set  $n$  cluster centers as the RBF centers. Recently, advanced algorithms like tabu and genetic are employed to determine them [30,32,34]. The structure of this RBF-ANN is similar to the one-hidden-layer BP-ANN which has just one output. An RBF can be deemed as a neuron in the hidden layer. The RBF centers are analogous to the synaptic weights from the input layer to the hidden layer. The shape parameters of the RBFs are similar to the biases (or thresholds) of the hidden neurons. The weights of the RBFs are like the connections from the hidden layer to the output while the threshold in Equation (7) is similar to the bias (or the threshold) of the output neuron in the BP-ANN.

#### 4. The Algorithm for Determining the Centers of RBFs and Their Shape Parameters

Though there are some advanced algorithms for determining the centers of RBFs and their shape parameters, the stochastic gradient approach presented in the textbook [40] is employed in this study. It was found in [27] that a single-hidden-layer ANN which composes just a few neurons in the hidden layer is sufficient for this application. Therefore, using the stochastic gradient approach is presumed to be enough. Besides, in the author's previous work [41] which was about solving the partial differential equations with the RBFs, the advantages of the stochastic gradient approach were clearly demonstrated.

Initially, all the values of  $\sigma_j$  and  $\zeta_{jk}$ , in which  $j = 1 \sim n$  and  $k = 1 \sim 7$ , are randomly chosen. For the reason that a negative  $\sigma$  is meaningless and too large  $\sigma$  makes the Gaussian RBF behave like a constant,  $\sigma_j$  is confined just in the range of 0 to 10. Similarly, a too far Euclidean distance from  $\xi$  to  $\zeta_j$  is also meaningless, so all the components of  $\zeta_j$  are confined in the range from  $-1$  to  $2$ , for the reason that all the components of  $\xi$  are just in the range of 0 to 1. After using Equation (12) to obtain the threshold  $w_0$  and the weights  $w_1, \dots, w_n$ , the error  $e_i$  corresponding to the  $i$ th piece of data can be calculated

$$e_i = \hat{\eta}_i - \eta_i \quad (15)$$

in which  $\eta_i$  is the exact output while  $\hat{\eta}_i$  is the predicted output by the RBF-ANN. The total square error is summed up as

$$E = \sum_{i=1}^N e_i^2 \quad (16)$$

The value of  $E$  is now dependant on values of  $\sigma_j$  and  $\zeta_{jk}$ . We can use the following formulae to update their values:

$$\sigma_j^{(\text{new})} = \sigma_j^{(\text{old})} - \mu \frac{\partial E}{\partial \sigma_j} \quad (17)$$

$$\zeta_{jk}^{(\text{new})} = \zeta_{jk}^{(\text{old})} - \mu \frac{\partial E}{\partial \zeta_{jk}} \quad (18)$$

in which  $\mu$  is called the stepping parameter. Smaller  $\mu$  results in a steady convergence but might make it trapped in a state of local minimum error. On the other hand, larger  $\mu$  results

in instability but may help to escape from the strap. In this study,  $\mu$  is set as a relaxation coefficient that varies according to the situation of convergence.

$$\mu = \begin{cases} 1000\mu^{(p)} & , \text{ if } \mu^{(p)} < 0.01\mu^{(i)} \\ 10\mu^{(p)} & , \text{ else if } N_{iter}/200 = \text{Int}(N_{iter}/200) \\ 1.03\mu^{(p)} & , \text{ else if } E < E^{(p)} \\ 0.5\mu^{(p)} & , \text{ otherwise} \end{cases} \quad (19)$$

in which the superscript ( $p$ ) denotes the word “previous” and the superscript ( $i$ ) denotes the word “initial”,  $N_{iter}$  represents the number of processed iterations, and the operator “Int( $\bullet$ )” means taking the integer part of a floating number. Multiplying  $\mu$  by 1000 or by 10 can help to escape from the state of local minimum error. The initial value of  $\mu$  is 0.01. The training algorithm is listed as follows.

Step 1: Randomly set initial values for  $\sigma_j$  and  $\zeta_{jk}$ . ( $j = 1 \sim n$  and  $k = 1 \sim 7$ )

Step 2: Use Equation (12) to obtain the threshold  $w_0$  and the weights  $w_j$ .

Step 3: Calculate the error  $e_i$  for each piece of the training data ( $i = 1 \sim N$ ), then sum up the total square error  $E$ .

Step 4: If the total square error  $E$  is the smallest so far, save the values of  $w_0$ ,  $w_j$ ,  $\sigma_j$  and  $\zeta_{jk}$  in a file.

Step 5: If the smallest  $E$  has not been updated for 500,000 times of training iteration, reset  $\sigma_j$  and  $\zeta_{jk}$  with random values and go back to Step 2.

Step 6: Use Equation (19) to set the stepping parameter  $\mu$ .

Step 7: Update all the values of  $\sigma_j$  and  $\zeta_{jk}$  by using Equations (17) and (18).

Step 8: Repeat Steps 2 to 7 and stop when the required number of iterations is achieved.

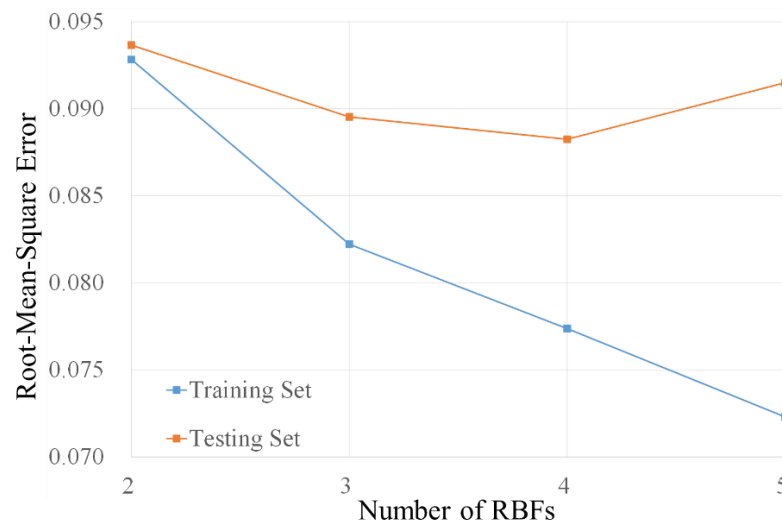
## 5. The Results of Training and Testing

The computer program is coded with Fortran 90 and compiled with the X64 compiler of Intel<sup>®</sup>-Fortran 2016. Each run uses only one processor of the computer of Intel<sup>®</sup> Core<sup>™</sup> i7-8700CPU @3.20GHz 3.19GHz. Artificial neural networks with 2–5 RBFs are trained and tested. Each RBF-ANN is trained with three runs in which  $2 \times 10^8$  iterations are processed. The computational time does not increase linearly with the number of RBFs. For the ANNs with 2 RBFs, the computational time is 2.98 h. As per those with 3, 4, and 5 RBFs, the computational times are 4.25, 5.56, and 6.81 h respectively. Though the centers of RBFs and their shape parameters are all set up with random numbers, after  $2 \times 10^8$  iterations of training, the ANNs with the same number of RBFs converge to results that are very close to each other. That means the convergence is valid for sure. It also confirms the presumption that using the stochastic gradient approach is enough. For the ANNs with just 2 or 3 RBFs, converging to the final result just needs several million times of iteration, which means the convergence is achieved within 1 hour.

The root-mean-square error is employed to examine the performance of the trained ANNs.

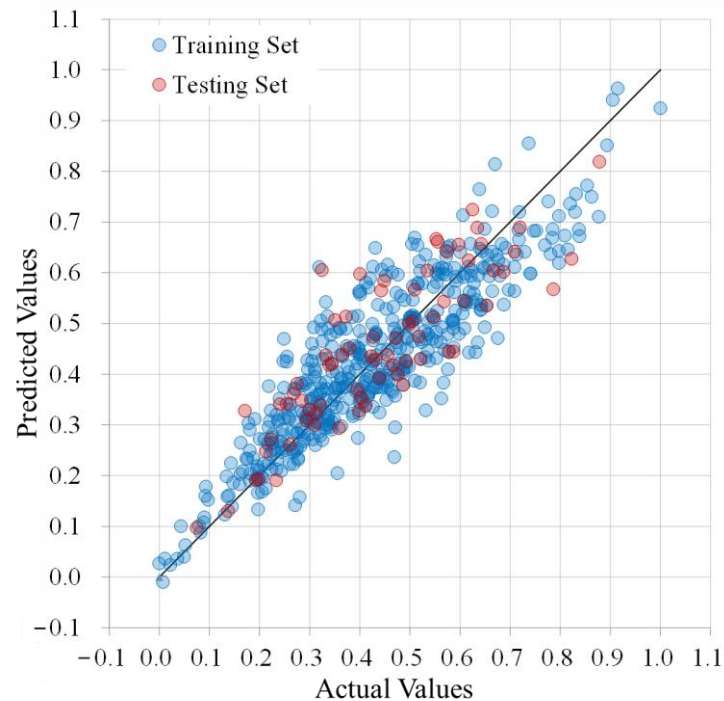
$$E_{r.m.s.} = \sqrt{\overline{(\eta - \hat{\eta})^2}} \quad (20)$$

where the overbar represents the average in the data set. Note that the numbers of data in the training set and testing set are 499 and 72, respectively. The results of training and testing are plotted in Figure 1.



**Figure 1.** The performance of the trained radial basis functions-artificial neural networks (RBF-ANNs).

It is found that setting more RBFs in the ANN results in smaller  $E_{r.m.s.}$  in the training set. However, it does not go the same way in the testing set. This implies too many RBFs in the ANN have a risk of over-fitting. Though the ANN with 4 RBFs fits testing data the best, using just 3 RBFs is supposed to be safer from over-fitting. The comparisons of exact values and the predicted values by the ANN with 3 RBFs are shown in Figure 2. Note that the plotted values in this figure are the normalized values. The root-mean-square errors of the training set and the testing set are 0.0822 and 0.0895, respectively. It is worthy to note that the testing set is the same as in [27] while the RBF-ANN in this study fits the testing set better. The root-mean-square error of the testing set in [27] was 0.0914.



**Figure 2.** The comparison of the outputs of ANN with their target values in which the ANN only has 3 RBFs.

Finally, the centers of the 3 RBFs are listed here.

$$\zeta_1 = [ -1.00000 \quad -1.00000 \quad 0.44034 \quad 0.60326 \quad -0.00486 \quad 0.52630 \quad 1.24615 ]^T \quad (21)$$

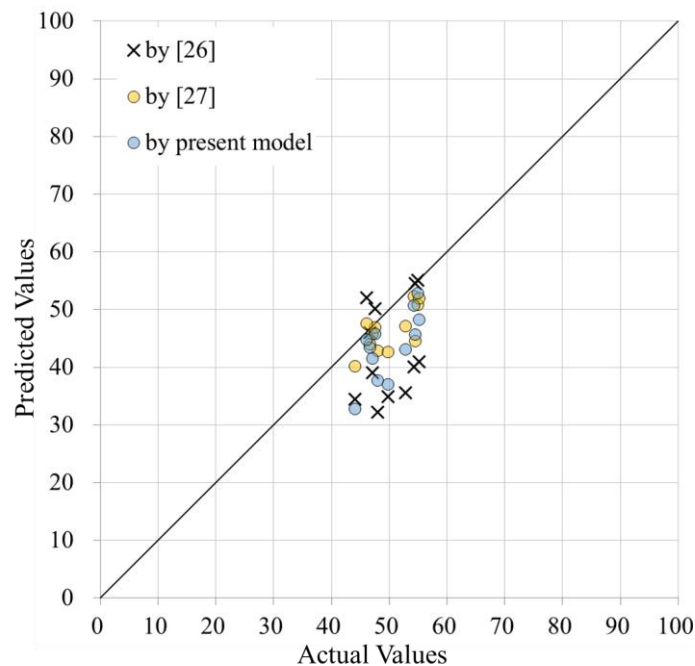
$$\zeta_2 = [ -0.84767 \quad 0.61160 \quad 0.46294 \quad 0.56556 \quad 0.67973 \quad 0.61628 \quad 0.65405 ]^T \quad (22)$$

$$\zeta_3 = [ -0.55917 \quad 1.05027 \quad 1.08523 \quad 0.79403 \quad 0.86121 \quad 0.77691 \quad 0.46370 ]^T \quad (23)$$

The shape parameters of these 3 RBFs are 2.06819, 0.90677, and 0.45945, respectively. Their weights are  $-4.17952$ ,  $11.2345$ , and  $-5275.82$  while the threshold is  $1.05900$ . With these numbers, one can use Equations (7) and (13) to calculate the corresponding output of the ANN by any input. The 28-day compressive strength of the concrete can be retrieved by using the backward transform of Equation (4).

## 6. The Verification

Additional 12 mix designs listed in [26] are used for the verification of the present RBF-ANN model. They are excluded from the database for training and testing. The predicted values are also compared with those predicted by the BP-ANN models in [26,27]. The results are plotted in Figure 3. It should be noted that the values plotted in this figure are the values of the 28-day compressive strength of concrete whose range in the database is from 5 MPa to 95.3 MPa. The  $E_{r.m.s.}$  by using the present RBF-ANN model is 7.53 MPa. It performs a little worse than the BP-ANN model in [27] whose  $E_{r.m.s.}$  is just 4.73 MPa. As per the comparison with the BP-ANN model in [26] whose  $E_{r.m.s.}$  is 10.70 MPa, the present RBF-ANN is still better.

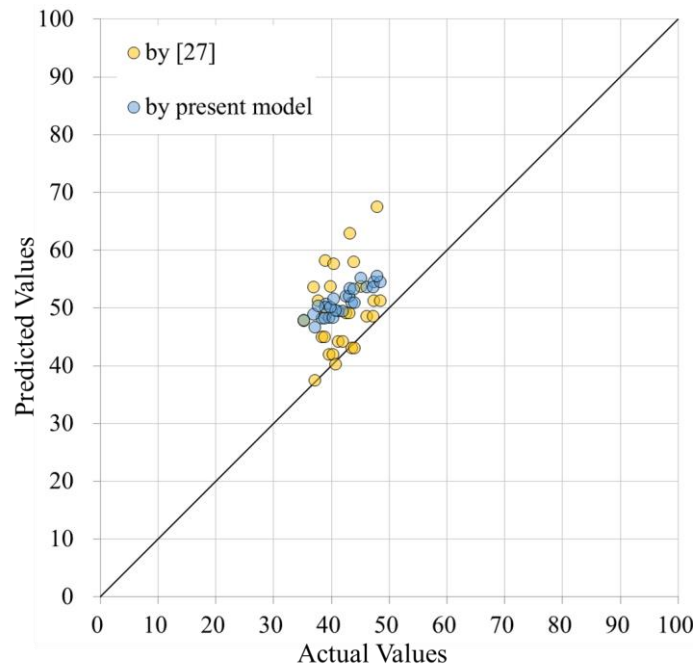


**Figure 3.** Comparison of the predicted values with the actual 28-day compressive strength of concrete [26] (Unit: MPa).

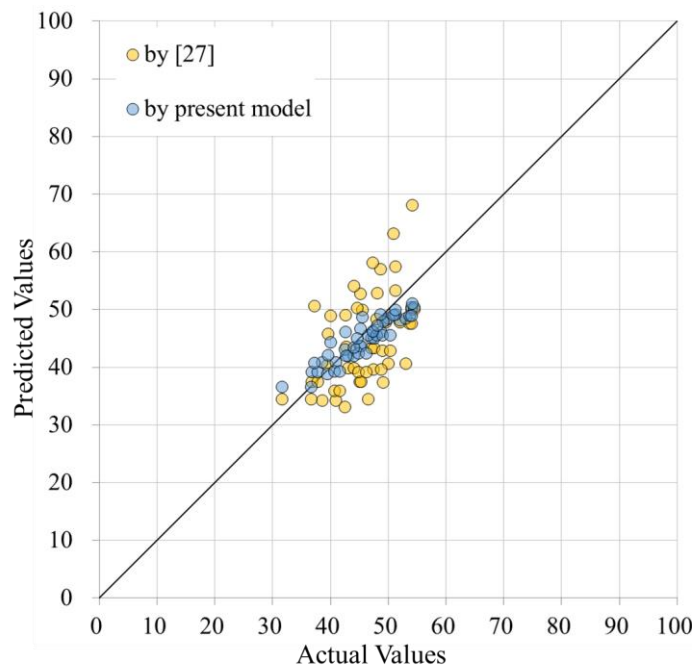
In [25], the genetic algorithm was employed to train the artificial neural network which was established to predict the 28-day, 56-day, and 91-day compressive strengths of concrete admixing with fly ash or without fly ash. Blast furnace slag and superplasticizer are not admixed in their samples. Though the predicted results of [25] were not listed in their paper, the actual compressive strengths were available. So, the data were used to validate the BP-ANN model of [27] further. Here, the data in [25] are also used to validate the present RBF-ANN model. The comparisons of the predicted and the actual



28-day compressive strengths of the concrete admixed with and without fly ash are plotted in Figures 4 and 5. For those admixed with fly ash, the present RBF-ANN model has a root-mean-square error of 9.42 MPa while the BP-ANN model of [27] performs with the root-mean-square error of 10.28 MPa. Both ANN models over-estimate the compressive strength, but the results of the present model are generally closer to the actual values. For those admixed without fly ash, the root-mean-square errors are 2.65 MPa and 7.19 MPa, respectively. The present model performs much better than the model in [27].



**Figure 4.** Comparison of the predicted values with the actual 28-day compressive strength of concrete in which fly ash was admixed [25] (Unit: MPa).



**Figure 5.** Comparison of the predicted values with the actual 28-day compressive strength of concrete in which no fly ash was admixed [25] (Unit: MPa).



The result is quite reasonable because the variation of particle sizes and the chemical composition in different fly ash may be the source of some uncertainties. Generally speaking, the performance of the present RBF-ANN is quite acceptable.

## 7. Conclusions

The focus of this study is on establishing an RBF-ANN model for predicting the 28-day concrete compressive strength according to the proportioning of the ingredients. The stochastic gradient approach present in [40] is employed for tuning the centers of RBFs and their shape parameters. Using a similar treatment in [41], the stepping parameter in the stochastic gradient approach is treated as a relaxation coefficient which can help to escape from the solution of local minimum error. An RBF in the ANN is analogous to a neuron in the BP-ANN which was employed for the same purpose in the author's previous work [27]. Artificial neural networks with 2–5 RBFs are trained and tested. Each RBF-ANN is trained with 3 runs in which  $2 \times 10^8$  iterations are processed. After the iteration processes, all the ANNs with the same number of RBFs converge to results that are very close to each other. This implies the convergence is valid. Therefore, the only issue concerning over-fitting is the number of RBFs. Finally, the ANN with just 3 RBFs is chosen for further verification. Data for verification are the same as those used in the author's previous work [27] so they can be compared. The comparisons show that the present model performs better than the author's previous model [27] which was established by using the BP-ANN and has 7 neurons in the hidden layer.

The centers of the RBFs, their shape parameters, their weights, and the threshold are all listed in this article. With these numbers and using Equations (3), (4), (7), and (13), anyone can predict the compressive strength of concrete according to the concrete mix proportioning on his/her own.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Institutional Review Board.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** All the data used in this paper can be traced back to the cited references.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Al-Manaseer, A.A.; Dalal, T.R. Concrete containing plastic aggregates. *Concr. Int.* **1997**, *19*, 47–52.
2. Jin, W.; Meyer, C.; Baxter, S. Glasscrete—Concrete with glass aggregate. *ACI Mater. J.* **2000**, *97*, 208–213.
3. Coutinho, J.S. The combined benefits of CPF and RHA in improving the durability of concrete structures. *Cem. Concr. Compos.* **2003**, *25*, 51–59. [[CrossRef](#)]
4. Batayneh, M.; Marie, I.; Asi, I. Use of selected waste materials in concrete mixes. *Waste Manag.* **2007**, *27*, 1870–1876. [[CrossRef](#)]
5. Siddique, R. *Waste Materials and By-Products in Concrete*; Springer: Berlin/Heidelberg, Germany, 2008.
6. Paul, S.C.; Panda, B.; Garg, A. A novel approach in modelling of concrete made with recycled aggregates. *Measurement* **2018**, *115*, 64–72. [[CrossRef](#)]
7. Tsivilis, S.; Parissakis, G. A mathematical-model for the prediction of cement strength. *Cem. Concr. Res.* **1995**, *25*, 9–14. [[CrossRef](#)]
8. Kheder, G.F.; Al-Gabban, A.M.; Abid, S.M. Mathematical model for the prediction of cement compressive strength at the ages of 7 and 28 days within 24 hours. *Mater. Struct.* **2003**, *36*, 693–701. [[CrossRef](#)]
9. Akkurt, S.; Tayfur, G.; Can, S. Fuzzy logic model for the prediction of cement compressive strength. *Cem. Concr. Res.* **2004**, *34*, 1429–1433. [[CrossRef](#)]
10. Hwang, K.; Noguchi, T.; Tomosawa, F. Prediction model compressive strength development of fly-ash concrete. *Cem. Concr. Res.* **2004**, *34*, 2269–2276. [[CrossRef](#)]
11. Zelić, J.; Rušić, D.; Krstulović, R. A mathematical model for prediction of compressive strength in cement-silica fume blends. *Cem. Concr. Res.* **2004**, *34*, 2319–2328. [[CrossRef](#)]
12. Zain, M.F.M.; Abd, S.M. Multiple regressions model for compressive strength prediction of high performance concrete. *J. Appl. Sci.* **2009**, *9*, 155–160. [[CrossRef](#)]
13. Ni, H.-G.; Wang, J.-Z. Prediction of compressive strength of concrete by neural networks. *Cem. Concr. Res.* **2000**, *30*, 1245–1250. [[CrossRef](#)]

14. Lee, S.-C. Prediction of concrete strength using artificial neural networks. *Eng. Struct.* **2003**, *25*, 849–857. [[CrossRef](#)]
15. Öztaş, A.; Pala, M.; Özbay, E.; Kanca, E.; Çağlar, N.; Bhatti, M.A. Predicting the compressive strength and slump of high strength concrete using neural network. *Constr. Build. Mater.* **2006**, *20*, 769–775. [[CrossRef](#)]
16. Topçu, I.B.; Sarıdemir, M. Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic. *Comput. Mater. Sci.* **2008**, *41*, 305–311. [[CrossRef](#)]
17. Alshihri, M.M.; Azmy, A.M.; El-Bisy, M.S. Neural networks for predicting compressive strength of structural light weight concrete. *Constr. Build. Mater.* **2009**, *23*, 2214–2219. [[CrossRef](#)]
18. Bilim, C.; Atiş, C.D.; Tanyildizi, H.; Karahan, O. Predicting the compressive strength of ground granulated blast furnace slag concrete using artificial neural network. *Adv. Eng. Softw.* **2009**, *40*, 334–340. [[CrossRef](#)]
19. Sobhani, J.; Najimi, M.; Pourkhorshidi, A.R.; Parhizkar, T. Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and ANFIS models. *Constr. Build. Mater.* **2010**, *24*, 709–718. [[CrossRef](#)]
20. Atici, U. Prediction of the strength of mineral admixture concrete using multivariable regression analysis and an artificial neural network. *Expert Syst. Appl.* **2011**, *38*, 9609–9618. [[CrossRef](#)]
21. Chou, J.-S.; Chiu, C.-K.; Farfoura, M.; Al-Taharwa, I. Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques. *J. Comput. Civ. Eng.* **2011**, *25*, 242–253. [[CrossRef](#)]
22. Duan, Z.H.; Kou, S.C.; Poon, C.S. Prediction of compressive strength of recycled aggregate concrete using artificial neural networks. *Constr. Build. Mater.* **2013**, *40*, 1200–1206. [[CrossRef](#)]
23. Chopra, P.; Sharma, R.K.; Kumar, M. Artificial Neural Networks for the Prediction of Compressive Strength of Concrete. *Int. J. Appl. Sci. Eng.* **2015**, *13*, 187–204.
24. Nikoo, M.; Moghadam, F.T.; Sadowski, L. Prediction of Concrete Compressive Strength by Evolutionary Artificial Neural Networks. *Adv. Mater. Sci. Eng.* **2015**, *2015*, 849126. [[CrossRef](#)]
25. Chopra, P.; Sharma, R.K.; Kumar, M. Prediction of Compressive Strength of Concrete Using Artificial Neural Network and Genetic Programming. *Adv. Mater. Sci. Eng.* **2016**, *2016*, 7648467. [[CrossRef](#)]
26. Hao, C.-Y.; Shen, C.-H.; Jan, J.-C.; Hung, S.-K. A Computer-Aided Approach to Pozzolanic Concrete Mix Design. *Adv. Civ. Eng.* **2018**, *2018*, 4398017.
27. Lin, C.-J.; Wu, N.-J. An ANN Model for Predicting the Compressive Strength of Concrete. *Appl. Sci.* **2021**, *11*, 3798. [[CrossRef](#)]
28. Segal, R.; Kothari, M.L.; Madnani, S. Radial basis function (RBF) network adaptive power system stabilizer. *IEEE Trans. Power Syst.* **2000**, *15*, 722–727. [[CrossRef](#)]
29. Mai-Duy, N.; Tran-Cong, T. Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Netw.* **2001**, *14*, 185–199. [[CrossRef](#)]
30. Ding, S.; Xu, L.; Su, C.; Jin, F. An optimizing method of RBF neural network based on genetic algorithm. *Neural Comput. Appl.* **2012**, *21*, 333–336. [[CrossRef](#)]
31. Li, Y.; Wang, X.; Sun, S.; Ma, X.; Lu, G. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transp. Res. Part C Emerg. Technol.* **2017**, *77*, 306–328. [[CrossRef](#)]
32. Aljarah, C.I.; Faris, H.; Mirjalili, S.; Al-Madi, N. Training radial basis function networks using biogeography-based optimizer. *Neural Comput. Appl.* **2018**, *29*, 529–553. [[CrossRef](#)]
33. Hong, H.; Zhang, Z.; Guo, A.; Shen, L.; Sun, H.; Liang, Y.; Wu, F.; Lin, H. Radial basis function artificial neural network (RBF ANN) as well as the hybrid method of RBF ANN and grey relational analysis able to well predict trihalomethanes levels in tap water. *J. Hydrol.* **2020**, *591*, 125574. [[CrossRef](#)]
34. Karamichailidou, D.; Kaloutsa, V.; Alexandridis, A. Wind turbine power curve modeling using radial basis function neural networks and tabu search. *Renew. Energy* **2021**, *163*, 2137–2152. [[CrossRef](#)]
35. Jiang, L.H.; Malhotra, V.M. Reduction in water demand of non-air-entrained concrete incorporating large volumes of fly ash. *Cem. Concr. Res.* **2000**, *30*, 1785–1789. [[CrossRef](#)]
36. Demirboğa, R.; Türkmen, İ.; Karakoç, M.B. Relationship between ultrasonic velocity and compressive strength for high-volume mineral-admixture concrete. *Cem. Concr. Res.* **2004**, *34*, 2329–2336. [[CrossRef](#)]
37. Yen, T.; Hsu, T.-H.; Liu, Y.-W.; Chen, S.-H. Influence of class F fly ash on the abrasion–erosion resistance of high-strength concrete. *Constr. Build. Mater.* **2007**, *21*, 458–468. [[CrossRef](#)]
38. Oner, A.; Akyuz, S. An experimental study on optimum usage of GGBS for the compressive strength of concrete. *Cem. Concr. Compos.* **2007**, *29*, 505–514. [[CrossRef](#)]
39. Durán-Herrera, A.; Juárez, C.A.; Valdez, P.; Bentz, D.P. Evaluation of sustainable high-volume fly ash concretes. *Cem. Concr. Compos.* **2011**, *33*, 39–45. [[CrossRef](#)]
40. Ham, F.M.; Kostanic, I. *Principles of Neurocomputing for Science & Engineering*; McGraw-Hill Higher Education: New York, NY, USA, 2001.
41. Wu, N.-J.; Chang, K.-A. Simulation of free-surface waves in liquid sloshing using a domain-type meshless method. *Int. J. Numer. Methods Fluids* **2011**, *67*, 269–288. [[CrossRef](#)]