Predicting the Compressive Strength of Concrete using Neural Network and Kernel Ridge Regression

Muhammad Amir Shafiq
School of Electrical and Computer Engineering,
Georgia Institute of Technology, Atlanta, GA, 30332
amirshafiq@gmail.com

***Abstract*—Kernel methods and neural networks (NN) are two of the most powerful tools of machine learning to solve the engineering and science problems. In this paper, we propose kernel ridge regression (KRR) and NN to estimate the compressive strength (CS) of concrete with recycled aggregate based on the values of cement, natural aggregate, recycled aggregate, sand, and water. We collected a dataset of 182 samples and carried out the data analysis for each material used in making concrete. The dataset is used for training, validation, and testing of concrete's CS using KRR and NN. We use both linear and nonlinear kernels in ridge regression, and up to four layers NN with three hidden layers and one output layer for CS prediction. KRR and NN are designed in MATLAB, and we use different NN types and training functions with objective evaluation criterion to obtain the optimum models for CS prediction. We choose gradient descent with momentum weight bias learning and mean square error (MSE) performance functions for model training. Simulation results show that both KRR and NN predict the CS of concrete accurately and efficiently with admissible MSE and correlation coefficient close to one. KRR with nonlinear 4th order polynomial kernel outperforms the NN for CS prediction.**

***Keywords*—Neural Networks; Kernel ridge regression; Compressive strength of concrete**

I. Introduction

Aggregate obtained by natural stone crushing is known as natural aggregate, whereas old concrete crushing produces an aggregate called recycled aggregate. The usage and availability of natural aggregate is getting severe due to highly developed architectures, rails construction, and roads development. Recycled aggregate (RA) provides a good alternative to natural aggregate (NA) in order to reduce the usage of raw materials. The debris obtained from crushing bridges, buildings, roads, and rails, and sometimes from calamities like floods, wars, and earthquakes can be used for producing RA [1-2]. RA has various stumpy value applications, which include gap filling, side-walks, capping purposes, and sub-base construction etc [3].

In the last few decades, the CS prediction of concrete has attracted the interest of many researchers due to the uncertainty and complexity of materials mixture. Conventional modeling and prediction schemes fail to estimate the concrete behavior due to mixtures nonlinearity and time consuming testing of material samples. Adaptive systems such as neural networks (NN) and kernel ridge regression (KRR) are very intelligent and computationally effective tools for predicting nonlinear behavior [4-7]. Various researchers have studied the effects of mixing different proportions of elements that contribute to CS of concrete and their estimation using NN [8-17]. The prediction of CS has become more convoluted problem due to materials characteristics and numerous parameters that contributes to problem nonlinearity and make it more challenging. In this paper, we propose to predict the CS of concrete on 28-day, usually measured by failure load test, using KRR and NN from the 182 samples dataset. The rest of the paper is structured as follows: the dataset analysis is given in section II; brief introductions of NN and KRR are presented in section III and IV, respectively; simulations results are illustrated in section V followed by conclusions in section VI.

II. Dataset Analysis

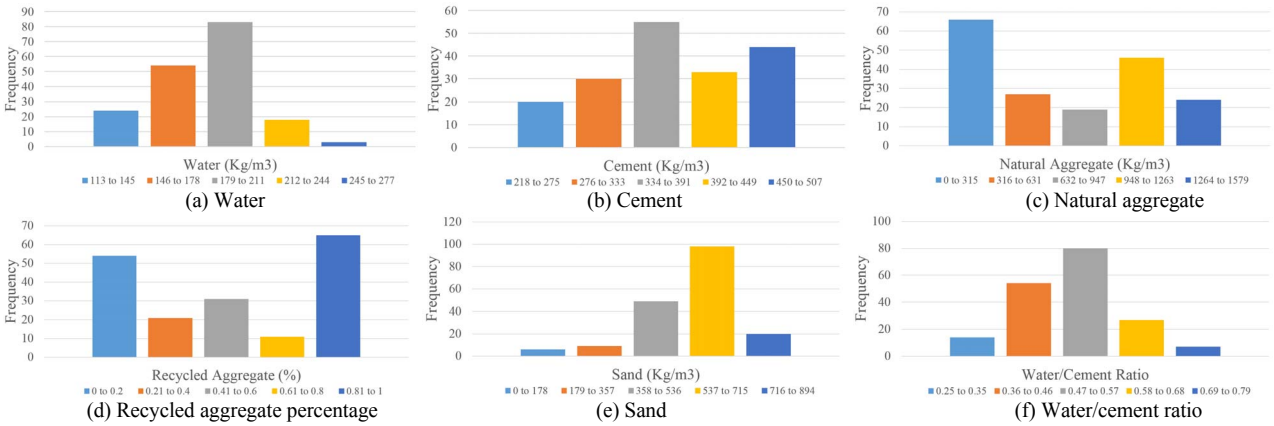
The strength of RA is influenced by the proportions of water, cement, fine aggregates, coarse, and different admixtures. Water/cement ratio (inversely proportional to compressive strength) plays a crucial role in determining the CS of concrete with RA. In this paper, the parameters collected to estimate the CS of concrete with RA includes cement (Kg/m3), natural aggregate (Kg/m3), recycled aggregate (Kg/m3), sand (Kg/m3), and water (Kg/m3). The collected dataset contains 182 samples, each showing the type and percentage of recycled aggregate, and the CS of concrete in MPa. The data range of each material used in CS prediction is shown in Table I. The range-frequency analysis of dataset is shown in Fig. 1.

To validate the dataset, we compared the relationship between different elements and CS, as shown in Fig. 2, with the dataset trends already published in the literature [1,8-19]. It can be observed that the CS of concrete is inversely proportional to the RA and water/cement ratio. Although, the values of CS are different in published datasets, but they yield same trends which validate our dataset. Furthermore, the plots of sand, cement, and natural aggregate follow a direct relationship with CS as shown in Fig. 3a-c, respectively.

TABLE I. Range of Collected Data

No.	Material	Unit	Range
1	Water	kg/m3	113-271
2	Cement	kg/m3	218-501
3	Sand	kg/m3	0-888
4	Recycled Aggregate	kg/m3	0-1484
5	Natural Aggregate	kg/m3	0-1576

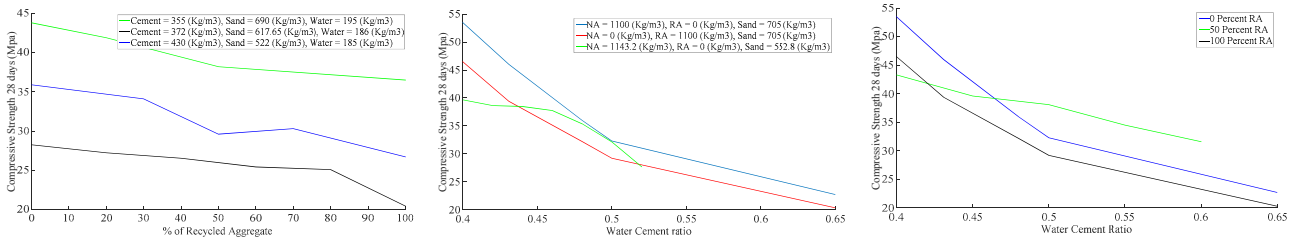
978-1-5090-4171-8/16/$31.00 ©2016 IEEE



(a) Water (b) Cement (c) Natural aggregate

(d) Recycled aggregate percentage (e) Sand (f) Water/cement ratio

Fig. 1. Range-frequency analysis of CS dataset



(a) CS vs RA percentage (b) CS vs water/cement ratio (c) CS vs water/cement ratio

Fig. 2. Relationship between CS, RA, and water/cement ratio

III. NEURAL NETWORK

Neurons, made up of nonlinear adaptive filters, combine to form a large network with complex interconnections, called neural network, which provide high computational capability for learning, adapting, controlling, and solving science and engineering problems. Large number of neurons arranged in multiple layers with nonlinear transfer functions enable the network to learn complex, linear and nonlinear relationships between inputs and outputs.

In multi-layered feed-forward neural network (MFNN), no feedback exists between the layers of neurons. MFNN input layer receive weighted neural inputs and provides output to the first hidden layer of MFNN. There may exist multiple hidden layers in MFNN. The output of last hidden layer becomes input to the final layer (output layer) of MFNN. A typical structure of MFNN with five inputs and one output is shown in Fig. 4 [4].

Back propagation, a gradient descent algorithm, is one of the most commonly used learning algorithms for NN, in which the NN weights follow negative gradient to satisfy performance function [23-24]. In this paper, we implement MFNN with back propagation, and use scaled conjugate gradient (SCG) and Levenberg–Marquardt (LM) functions for training neurons. First layer of hidden neurons processes the output of input layer and extracts useful features to reconstruct mapping from the input space. The neighboring hidden layers are fully linked to each other via complex weights interconnections to learn complex nonlinear input mapping. Finally, the neurons of output

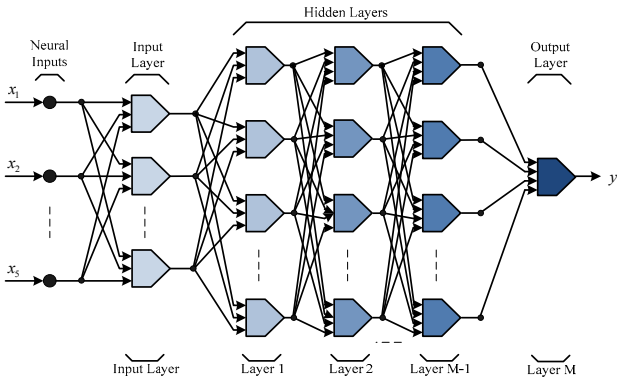


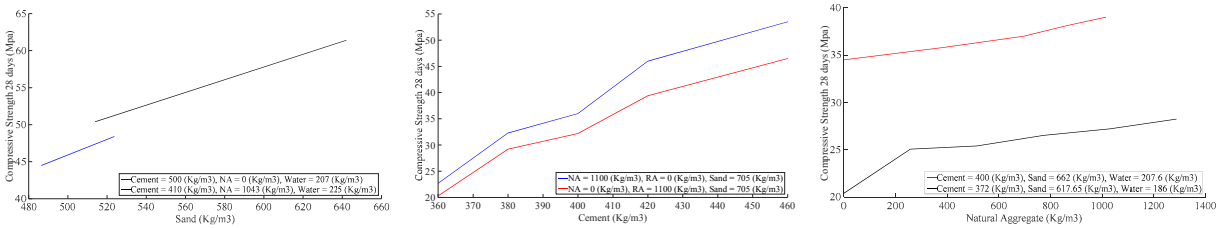
Fig. 3. A typical MFNN structure with five inputs and one output

layer yield estimate of concrete's CS in response to the different input samples combination. The number of hidden layers and the number of neurons in each layer are problem dependent and influence the quality of network training as well.

IV. KERNEL RIDGE REGRESSION

KRR combines the l2-norm regularization of linear least-squares with the kernel method [20]. Therefore, KRR learns a linear function in the space induced by the respective kernel and the data. For nonlinear kernels, this corresponds to a nonlinear function in the original space.

978-1-5090-4171-8/16/$31.00 ©2016 IEEE



(a) CS vs sand (b) CS vs cement (c) CS vs natural aggregate

Fig. 4. Relationship of CS with sand, cement, and NA

The classical way [21-22] to model the dependencies between covariates x_i and response variables y_i is to minimize the quadratic cost as

$$C(w) = \frac{1}{2}\sum_i (y_i - W^T x_i)^2, \quad (1)$$

Where W represents the weight vector. However, equation (1) presents a risk of over fitting if we work in a feature space, which maps the $x_i \rightarrow \phi(x_i)$, where $\phi(\cdot)$ is a mapping function. In order to regularize the cost function, we penalize the norm of w, which is simple yet an effective way of regularization, also known as weight-decay. The learning rate λ can be determined using cross validation or estimation using leave-one-out method. The updated cost function, which needs to be minimized is then given as

$$C(w) = \frac{1}{2}\sum_i (y_i - W^T x_i)^2 + \frac{1}{2}\lambda ||w||^2. \quad (2)$$

Solving equation (2) by taking derivative and equating equal to zero yields

$$w = (\lambda I + \sum_i x_i x_i{}^T)^{-1}\left(\sum_j y_j x_j\right), \quad (3)$$

Where I represents the identity matrix. It can be observed that the regularization term stabilizes the inverse numerically by bounding the smallest eigenvalues away from zero. In this case, we can replace all data-cases with their feature vectors such that: $x_i \rightarrow \phi_i = \phi(x_i)$, in which the number of dimensions can be much higher, or even infinitely higher, than the number of data-cases. However, using the identity in equation (4), we can compute the inverse in equation (3) in the smallest space of the two possibilities, either the dimension of feature space or the number of data-cases.

$$(P^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = PB^T (BPB^T + R)^{-1}. \quad (4)$$

It is worth noting that if B is not square, then inverse is performed in the space of different dimensionality. Therefore, in order to adapt, we define $\phi = \phi_{ai}$ and $y = y_i$. The solution is then given as

$$(\lambda I_d + \phi\phi^T)^{-1}\phi y = \phi(\phi^T\phi + \lambda I_n)^{-1} y. \quad (5)$$

Equation (5) can be rewritten as

$$\begin{cases} w = \sum_i a_i \phi(x_i) \\ \alpha = (\phi^T\phi + \lambda \mathrm{I}_n)^{-1} \mathrm{y} \end{cases}. \quad (6)$$

The solution w must lie in the span of data-cases, even if the dimensionality of feature space is much larger than the number of data-cases. The predicted value for a new test point, x, can be computed by projecting it onto the solution w as

$$y = w^T\phi(x) = y(\phi^T\phi + \lambda I_n)^{-1}\phi^T\phi(x) = y(K + \lambda I_n)^{-1}\kappa(x), \quad (7)$$

Where $\mathrm{K}(\mathrm{bx}_i, \mathrm{bx}_j) = \phi(\mathrm{x}_i)^T\phi(\mathrm{x}_j)$ and $\kappa(\mathrm{x}) = \mathrm{K}(\mathrm{x}_i, \mathrm{x})$. The bias to the kernel K can be realized by adding one more constant feature to ϕ: $\phi_0 = 1$. The value of w_0 then represents the bias as

$$w^T\phi = \sum_a w_a \phi_{ai} + w_0. \quad (8)$$

V. Simulation Results

In this section, we present the simulation results for predicting the CS of concrete with RA using MFNN and KRR. We use MSE as a stopping criterion for training, which can be defined as average squared difference between the output of MFNN or KRR and the desired target. Lower values of MSE mean better performance and good model adaptation, whereas, regression value (R-value) measures the correlation between the output of MFNN or KRR and the targets. R-value of one and zero indicate a close and random relationships, respectively. These two criteria, MSE and R-value, are used as the basis for training MFNN and selecting the idealized network. In this paper, As shown in Fig. 5, 70% of the total data is used for MFNN training, 15% is used for validation, and remaining 15% is used for testing.

- Neural Network

In order to design an optimum model for CS prediction, MFNN with different number of hidden layers and different number of neurons in hidden layers are investigated in this paper. The MFNN simulation parameters are given in Table II.

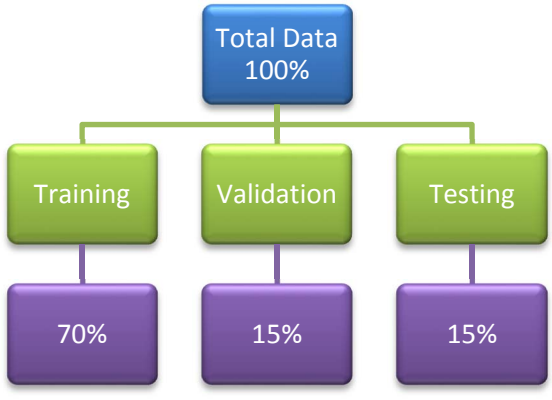


Fig. 5. Data management for MFNN Training, validation, and testing

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

TABLE II. SIMULATION PARAMETERS

Parameters	Value
Training function	TrainLM, TrainSCG
Performance function	MSE
Number of layers	Up to 4
Input layer neurons	5
Number of hidden layers	1,2,3
Output layer transfer function	Tansig,Purelin,Tanh
Performance goal	0
Minimum gradient	1e-06
Maximum validation fail	6
Maximum epochs	1000

TABLE III. MSE AND R VALUES WITH ONE HIDDEN LAYER

Hidden Neurons	MSE	R
1	102.3	0.8
2	168.5	0.59
3	80.5	0.82
4	56.2	0.89
5	79.5	0.83

TABLE IV. SE AND R VALUES WITH TWO HIDDEN LAYERS

HN(1)	HN(2)	MSE	R
4	2	102.5	0.8
4	1	68.5	0.67
5	2	60.1	0.9
5	1	102.5	0.51
6	3	40.5	0.93
6	2	48.6	0.92
6	1	50.5	0.89

FNN simulation results with one hidden layer are shown in Table III. The best result is obtained with 4 hidden neurons i.e. MSE = 56.2 and Regression = 0.89. MFNN simulation results with two hidden layers are shown in Table IV, where HN(X) represents the number of hidden neurons in layer X and R represent the regression values. The best result is obtained using 6 neurons in 1st hidden layer and 3 neurons in 2nd hidden layer yielding MSE = 40.5 and Regression = 0.93. MFNN simulation results with three hidden layers are shown in Table V, which illustrates that the best result is obtained with 7 neurons in 1st hidden layer, 3 neurons in 2nd hidden layer, and 2 neurons in 3rd hidden layer yielding MSE = 29.6 and Regression = 0.93. The results of MFNN simulation with large number of neurons in hidden layers are summarized in Table VI.

TABLE V. MSE AND REGRESSION VALUES WITH THREE HIDDEN LAYERS

HN(1)	HN(2)	HN(3)	MSE	R
4	2	1	155.5	0.78
4	1	1	31.2	0.92
5	2	2	96.2	0.73
5	2	1	47.2	0.93
6	3	3	35.1	0.93
6	2	2	61.2	0.89
6	2	1	66.5	0.86
7	3	2	29.6	0.93
8	4	2	39.2	0.93
9	4	1	62.3	0.91
10	5	2	50.5	0.91

TABLE VI. MSE AND REGRESSION VALUES WITH DIFFERENT NUMBER OF HIDDEN LAYERS & NEURONS

HN(1)	HN(2)	HN(3)	MSE	R
8	-	-	65.1	0.88
10	-	-	39.2	0.94
15	-	-	27.2	0.95
30	-	-	63.2	0.91
5	3	-	86.9	0.87
10	6	-	55.9	0.92
20	10	-	80.7	0.91
50	15	-	52.5	0.92
50	20	-	98.6	0.9
100	25	-	65.3	0.93
70	35	-	76.1	0.92
10	5	3	29.1	0.93
20	10	8	39.3	0.93
30	10	8	63.5	0.91
40	20	15	32.5	0.94
50	20	15	69.4	0.91

Although, the large number of hidden neurons learns the complex nonlinear mapping, but they may adapt a different model, which results in higher MSE and large errors. As shown in experimental results, the best result for CS prediction in our dataset is obtained with only 15 neurons in 1st hidden layer yielding MSE = 27.2 and Regression = 0.95. After selecting the optimum number of hidden layers and hidden neurons for the MFNN, we simulated the trained network for CS prediction on 15% testing data. MSE and regression plots for training, validation, and testing phase are shown in Fig. 6 and Fig. 7, respectively, which illustrate that the linear regression between MFNN output and desired output is achieved successfully for all phases.

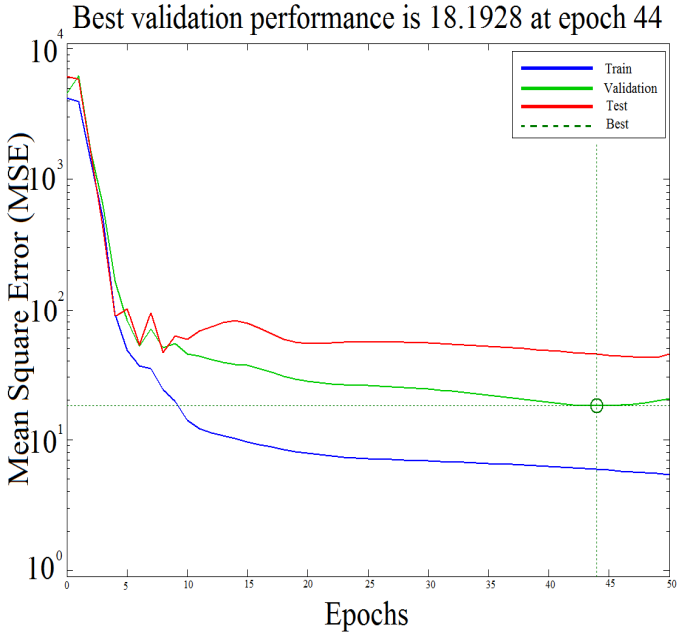


Fig. 6. MSE of training, validation, and testing phase

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

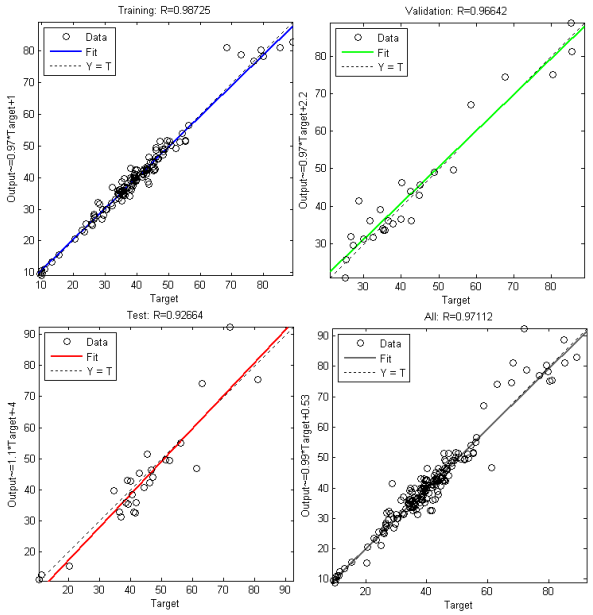


Fig. 7. Regression plots for MFNN training phase

Simulation results show that MFNN was trained adequately and predicted the test data with very good accuracy. Overall regression value for training, validation, and testing states is about 0.9711, which is very close to one, illustrating the effectiveness of MFNN.

- Kernel Ridge Regression

We simulated kernel ridge regression with both linear and nonlinear kernels and the best results are obtained using nonlinear kernel of 4th order polynomial. Added input materials plot of training data is presented in Fig. 8, which shows that the model is trained adequately using KRR. We have also shown the effects of different materials on concrete CS and are presented in Fig. 9. It can be seen that RA has greatest impact on compressive strength computation followed by natural aggregate (NA).

Test data and the output of trained KRR model are shown in Fig. 10, which show a strong correspondence and effective model fitting. The regression of testing data is 0.99197 as shown in Fig. 11, which is approximately equal to one. Furthermore, the MSE of KRR testing phase is 4.7682. Therefore, the experimental results show that the KRR performs very well for CS prediction and outperforms the MFNN in terms of MSE and regression.

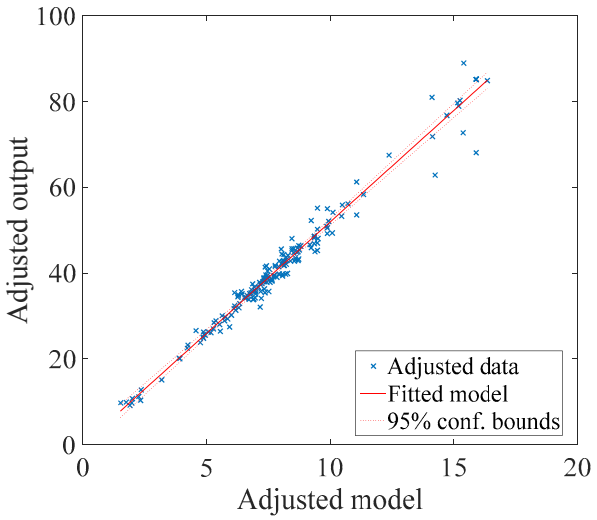


Fig. 8. Added input plot of KRR training phase

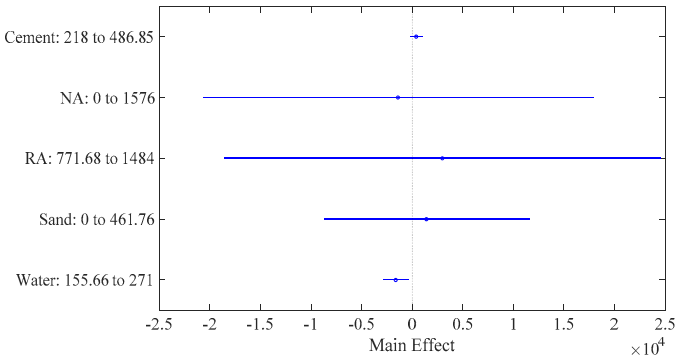


Fig. 9. Effect of different elements on CS Prediction

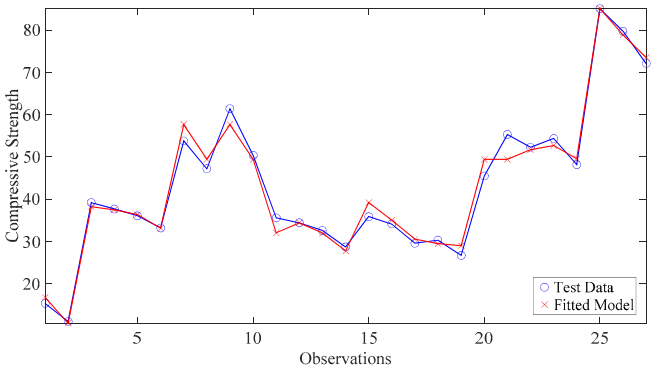


Fig. 10. Model fitting of KRR on test data

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

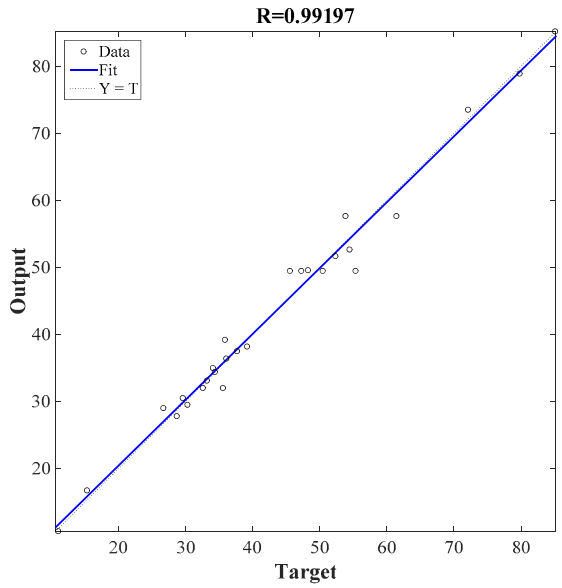


Fig. 11. Regression of KRR testing phase

VI. CONCLUSION

The MFNN and KRR models developed in this paper were used for predicting the CS of concrete. The performance of designed models on training and testing data were very good. Both MFNN and KRR successfully learnt the relationship between the different input and output parameters. The results of testing phase show that both KRR and MFNN were able to learn relationship between input and output variables with reasonably good accuracy. Validation and testing errors were very small and final MSE was admissible for both models. Over fitting didn't happen in both models and very good values of regression were obtained. KRR outperforms NN in terms of MSE and regression. MFNN and KRR have strong potential as a feasible tool for predicting the CS of concrete. Additionally, to obtain desired strength and suitable workability, technical personnel's can try several mixture proportions as an input to trained model for obtaining the best proportion of elements to yield the highest CS. The developed models are expected to not only save time but also decrease design cost.

REFERENCES

[1] N. S. Chai, "High-Strength Structural Concrete with Recycled Aggregates," Thesis, Faculty of Engineering and Surveying, University of Southern Queensland, 2004

[2] A. Katz, "Properties of concrete made with recycled aggregate from partially hydrated old concrete," Cement and Concrete Research, Vol. 33, pp. 703-711, 2003

[3] T. Manzar, "Compressive Strength of Concrete with Recycled Aggregates," Faculty of Materials and Metallurgy, GIKI, 2001

[4] M.M. Gupta, L. Jin, and N. Homma, Static and Dynamic Neural Networks, Wiley-IEEE, 2003

[5] M. Shafiq and M. A. Shafiq, "Direct Adaptive Inverse Control," IEICE Electron. Express, Vol. 6, No. 5, pp. 223-229, 2009

[6] A. Janczak, Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-oriented Approach, Springer, NY, 2005

[7] K. S. Narendra, "Neural Networks for Control theory and practice," Proceedings of the IEEE, Vol. 84, No. 10, pp. 1385-1406, Oct 1996

[8] Buck and Alan, "Recycled Concrete," U.S. Army Engineer Waterways Experiment Station, 1972

[9] A. R. Khaloo, "Crushed tile coarse aggregate concrete," Cement Concrete Aggregate, Vol. 17, No. 2, pp.119-25, 1995

[10] C. S. Poon and D. Chan, "Paving blocks made with recycled concrete aggregate and crushed clay brick," Journal of Construction and Building Materials, Vol. 20, pp. 569-577, 2006

[11] C. S. Poon, S. C. Kou and L. Lam, "Use of Recycled Aggregate in Molded Concrete Bricks and Blocks," Department of Civil and Structural Engineering, The Hong Kong Polytechnic University.

[12] F. M. Abed, "Using of Recycled Aggregate in Producing Concrete Elements," 2009

[13] E. Rasa, H. Ketabchi and M.H. Afshar, "Predicting Density and Compressive Strength of Concrete Cement Paste Containing Silica Fume Using Artificial Neural Networks," Transaction on Civil Engineering, Vol. 16, No. 1, pp. 33-42, 2009

[14] V. K. Alilou and M. Teshnehlab, "Prediction of 28-Day Compressive Strength of Concrete on the Third Day using Artificial Neural Networks," International Journal of Engineering (IJE), Vol. 3, No. 6, pp. 565-576, 2000

[15] B. K. R. Prasad, H. Eskandari, B.V. V. Reddy, "Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN," 2008

[16] W. P. S. Dias and S. P. Pooliyadda, "Neural networks for predicting properties of concretes with admixtures," Construction and Building Materials, Vol. 15, pp. 371-379, 2001

[17] N. H. Guang and W. J. Zong, "Prediction of compressive strength of concrete by neural networks," Cement and Concrete Research, Vol. 30, pp. 1245-1250, 2000

[18] A. Zuhud, "Performance of Recycled Aggregate Concrete", Master Thesis, Islamic University Gaza, Palestine, December 2008.

[19] V. Corinaldesi, '' Mechanical and elastic behavior of concretes made of recycled-concrete coarse aggregates'', Construction and Building Materials, Vol. 24, pp. 1616–1620, 2010.

http://scikit-learn.org/stable/modules/kernel_ridge.html

[20] Max Welling, "Kernel ridge Regression," Department of Computer Science University of Toronto 10 King's College Road Toronto, Canada.

[21] Yuri Kalnishkan, "An Introduction to Kernel Methods," Technical Report CLRC-TR-09, May 2009

[22] H. Naderpour, A. Kheyroddin and G. G. Amiri, "Prediction of FRP-confined compressive strength of concrete using artificial neural networks," Composite Structures, Vol. 92, 2010.

[23] Ahmet, M. Pala, Erdogan, C. Naci and M. Asghar Bhatti, "Predicting the compressive strength and slump of high strength concrete using neural network," Construction and Building Materials, Vol. 20, pp. 769–775, 2006.

978-1-5090-4171-8/16/$31.00 ©2016 IEEE