Comparison of Dynamic Linear and Neural Network Models in Predicting Cement Compressive Strength

Dimitris C. Tsamatsoulis

***Abstract*—** A dynamic approach of two widely applied techniques has been used in predicting the typical 28-day compressive strength of Portland cement: Multiple linear regression (MLR) and artificial neural networks (ANN). Eight totally ANNs were built involving three layers with one or two nodes in the hidden layer and a cascade ANN as well, using sigmoid, hyperbolic tangent and radial basis functions. The comparison is based on the mean square residual error (MSRE) of testing sets. The ANNs of higher performance are the simplest ones, with only one node in the hidden layer and sigmoid or hyperbolic tangent function. An optimization of the future testing period was performed and the optimal value was equal to one day. Consequently the implementation of the dynamical model in the daily strength prediction needs an updating of the equations, every day that new results appear.

***Keywords*—**Cement, modeling, neural network, prediction, strength

I. Introduction

NEURAL NETWORKS consist an attractive tool to model non-linear processes and phenomena. In the cement industry artificial neural networks (ANN) are mostly used to describe and control the main production operations: Burning [1]-[4] and grinding [5]-[7]. Predicting cement typical compressive strength from earlier analyses results on the same sample remains a challenging issue. Mainly linear and polynomial models have been developed or techniques that can be reduced to such models. Extensive reviews of these techniques exist in [8]-[9]. Neural networks have successfully used for the prediction of the concrete strength and various ANN types have been developed. Examples of such implementation of ANN in concrete are referred in [10]-[19]. But there are relatively few studies based on ANN methodology or generally evolutionary and genetic algorithms (GA), correlating cement typical 28-day strength with other cement properties.

Akkurt et al. [20] developed a GA – ANN model of cement compressive strength by collecting and processing 6 months industrial data of chemical, physical and mechanical properties of the cement. Their results indicated that an increase of C_3S, SO_3 and specific surface leads to increased strength. Baykasoglu et al. [21] and Thamma et al. [22] utilized gene expression programming (GEP) and neural networks for the prediction of strength of Portland composite cement. Zhang et al. [23] introduced an algorithm named Double-layer Multi-expression Programming (DMEP). They applied the DMEP model to the prediction of 28-day Portland cement strength and they compared this model with other four computing models namely Multi-Expression Programming model (MEP), Gene Expression Programming model (GEP), Neural Networks model (NN) and Fuzzy logic model (FL). Madsen et al. [24] applied FL and GA techniques to predict the strength of CEM I cement of all the possible strength classes. Ren et al. [25] applied generalized regression neural network (GRNN) techniques to predict the heat of hydration and compressive strength of cement. A model combining principal components analysis (PCA) and ANN algorithms was developed by Yongzheng et al. [26]. The predictions are accurate inside their field of application. If the value of a parameter not contained in the set of the independent variables during the cement production process changes noticeably, the predictive model could fail. Consequently most of the models could be called “static” as the parameters are estimated by a given data set and the future strength is predicted.

Tsamatsoulis [9] performed a comparison of the static polynomial equations referred in [27] and movable time horizon models based on linear regression methods. The latter models incorporate the uncertainty due to the time variability of non involved factors during the modeling, thus they are dynamic. In [28] the particularities of these two classes of models have been investigated in detail and the superiority of the dynamic models was proved. This study aims at comparing two types of dynamic models: The multiple linear regression (MLR) technique with a second wide range of models based on several types of ANN. Criterion of the comparison is the capability of a model to predict the future cement strength. Considering the linear model as the basic one, the generalization ability [29] of the selected ANN is to be checked. Some simple steps have been also implemented to enhance the mentioned ability of the ANNs. An effort was also made to study the problem of over-fitting, frequently encountered during the ANN development and implementation and referred also by Haykin in an excellent and comprehensive foundation of neural networks [30].

D. C. Tsamatsoulis is with Halyps Building Materials S.A., Aspropyrgos, Attica, Greece (phone: 0030 6946 508587; e-mail: d.tsamatsoulis@halyps.gr).

Two Portland cement types produced according to EN 197-1:2011 were studied: CEM II A-L 42.5 N and CEM II B-M (P-L) 32.5 N. The modeling is based on the results of the daily average samples of cement produced in two cement mills (CM) of Halyps plant from 2006 to 2014. The following analyses data were utilized: (i) Residue at 40 μm sieve, R40 (%), measured with air jet sieving; (ii) Specific surface, Sb (cm^2/gr), measured according to EN 196-6; (iii) Loss on ignition, LOI (%), and insoluble residue, Ins_Res (%) of the cement measured according to EN 196-2; (iv) SO_3 (%) measured with X-ray fluorescence; (v) Compressive strength at 1 and 28 days (MPa). The preparation, curing and measurement of the specimens were made according to EN 196-1. The modeling predicting the 28-day strength was based on more than 3400 data sets.

II. Mathematical Techniques

A. Multiple Linear Regression

The common independent variables in all models are: *LOI, SO_3, Ins_Res, Sb , R40* and one day strength, *Str_1*. The three chemical analyses characterize the cement composition. These six variables are named X_I with I=1 to N and N=6, where: $X_1=LOI$, $X_2=SO_3$, $X_3=Ins\_Res$, $X_4=Sb$, $X_5=R40$, $X_6=Str\_1$. The dependent variable is the 28 days strength, $Y=Str\_28$ and then algorithm proceeds as follows:

- For the experimental data set, the minimum and maximum values of X_I and Y, $X_{I,MIN}$, $X_{I,MAX}$, Y_{MIN}, Y_{MAX} respectively, are computed.
- The variables X_I, Y are normalized. The set of the new variables XN_I, YN is calculated by (1) and (2):

$$XN_I = \frac{X_I - X_{I,MIN}}{X_{I,MAX} - X_{I,MIN}} \qquad I = 1..N \tag{1}$$

$$YN = \frac{Y - Y_{MIN}}{Y_{MAX} - Y_{MIN}} \tag{2}$$

- The normalized 28 days strength, YN, is given by (3):

$$YN = A_0 + \sum_{I=1}^{N} A_I \cdot XN_I \tag{3}$$

- The calculated 28 days strength, Str_28_{Calc}, is provided by (4):

$$Str\_28_{Calc} = Y_{MIN} + Y_N \cdot (Y_{MAX} - Y_{MIN}) \tag{4}$$

The coefficients A_I, I=0 to N are determined by minimizing the residual error s_{Res} calculated by (5):

$$s_{Res}^2 = \frac{\sum_{J=1}^{M} (Str\_28_{Calc,J} - Str\_28_{Act,J})^2}{M - N - 1} \tag{5}$$

where Str_28_{act} = actual 28 days strength, M = number of data sets.

B. Dynamic Models

The common feature among the linear regression and neural network techniques is the dynamic modeling which is described by the subsequent algorithm:

(i) A date t is assumed where a 28-day strength result appears. The specimen was prepared 28 days ago. The production date is in distance t-29 days from date t.

(ii) A time interval of T_D days and the samples belonging to the period $[t-29-T_D, t-29]$ are presumed. The dynamic data set consists of this population of samples.

(iii) Using the selected model (MLR or ANN) an optimum set of parameters A optimizing the respecting s_{res} is computed.

(iv) At day t, the chemical and physical results of the cement produced in the previous day and the 1 day strength of the cement produced 2 days ago have been measured.

(v) Using the set A, the 28 days strength of cement produced at $t-2$ day is estimated, by applying the respecting model.

(vi) The steps (iv), (v) are repeated for all the next dates up to $t+T_F-1$, where T_F is a predetermined time interval without new parameters estimation. Consequently T_F is ≥ 1.

(vii) According to step (vi) for the dates belonging to the interval $[t, t+T_F-1]$, the future strength of the cement produced in the time interval $[t-2, t+T_F-3]$ is computed according to the equation of step (v). Otherwise if the date is greater than $t+T_F-1$, new parameters estimation is performed starting from step (i).

(viii) As the time span remains T_D, when the results of T_F days are completed, then, the time interval is moved on by T_F days. Thus the future 28-day strengths are calculated using models applied to data sets of movable time span T_D and in steps of length T_F.

(ix) For a pair (T_D, T_F), the classes of models, MLR and ANN, shall be optimized according to the following two criteria: (a) minimum mean square residual error $MSRE_{Past}$ during modeling and (b) minimum error $MSRE_{Futur}$ during the future application of the models. These two $MSRE$ are calculated by (6), (7):

$$MSRE_{Past} = \sqrt{\frac{\sum_{I=1}^{K_{TD}} s_{Res,TD}(I)^2}{K_{TD}}} \tag{6}$$

$$MSRE_{Future} = \sqrt{\frac{\sum_{I=1}^{K_{TD}} s_{Res,TF}(I)^2}{K_{TD}}} \tag{7}$$

For each model and for each past and future time interval, a set (A, $s_{Res,TD}$, $s_{Res,TF}$) is computed from the samples belonging to this interval. Depending on T_D and T_F values, the number of the consecutive sets (A, $s_{Res,TD}$, $s_{Res,TF}$) is K_{TD}, the number of data sets in each past interval I is $M_{TD}(I)$ and in each future interval I is $M_{TF}(I)$. The errors $s_{Res,TD}$, $s_{Res,TF}$ are computed from (5) where M is replaced by $M_{TD}(I)$ and $M_{TF}(I)$ respectively.

C. Neural Networks

Two main kinds of neural networks were implemented: The usual three layers feed-forward ANN and the more complicated cascade neural network. The back propagation method was applied in batch mode. In the hidden layer of the three layers ANN the modeling included one and two nodes. The non-linearity of the activation function was approached using, the sigmoid, hyperbolic and radial basis functions. The modeling also involves ANN with and without bias. The result of these combinations is an elevated number of models with nomenclature presented in Table I.

Table I. Types of ANN

	Three Layers ANN			
	Number of Nodes in Hidden Layer	***Number of Parameters***	***Activation Function***	***Bias***
S_1N	1	7	Sigmoid	NO
S_1N_B	1	8	Sigmoid	YES
HT_1N	1	7	Hyperbolic Tangent	NO
HT_1N_B	1	8	Hyperbolic Tangent	YES
RBF_1N	1	13	Radial Basis Function	NO
S_2N	2	14	Sigmoid	NO
RBF_2N	2	20	Radial Basis Function	NO
Cascade ANN				
	Number of Hidden Layers	***Number of Parameters***	***Activation Function***	***Bias***
CASC	4	12	Sigmoid	NO

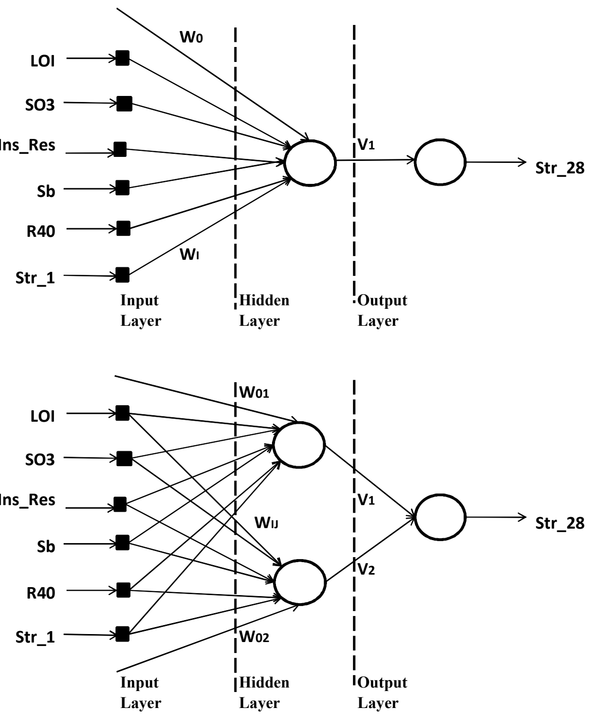

Fig. 1 Three layers ANN with one and two nodes in the hidden layer

The three layers ANNs with one and two nodes in the hidden layer are depicted in Fig. 1. The one node ANN accepts and processes all the data. In the case of two nodes, each one takes as inputs the data of each cement type (CEM B-M 32.5 and CEM A-L 42.5). A specific algorithm has been developed and incorporated in the software to recognize the cement type according to chemical analyses data. In each node the linear combination between the inputs and the synaptic weights is performed and introduced to the activation function. In the input layer the data are normalized. If the sigmoid or radial basis functions are used, equations (1) and (2) are applied for normalization, leading to normalized data belonging to the interval [0, 1]. In case of hyperbolic tangent activation function, data are normalized according to (8)-(9):

$$XN_I = \frac{2 \cdot X_I - \left(X_{I,MIN} + X_{I,MAX}\right)}{X_{I,MAX} - X_{I,MIN}} \qquad I = 1..N \tag{8}$$

$$YN = \frac{2 \cdot Y - \left(Y_{MIN} + Y_{MAX}\right)}{Y_{MAX} - Y_{MIN}} \tag{9}$$

In this case XN_I, YN belong to the interval [-1, 1]. Then the calculated *Str_28* is back calculated from its normalized value by applying (10):

$$Str\_28_{Calc} = \frac{Y_{MAX} + Y_{MIN}}{2} + Y_N \cdot \frac{\left(Y_{MAX} - Y_{MIN}\right)}{2} \tag{10}$$

The non-linear equations describing the activation functions are given by (11) – (13).

Sigmoid function

$$o(J) = \frac{1}{1 + \exp\left(-\sum_{I=0}^{N} W_{I,J} \cdot XN_I\right)} \tag{11}$$

Where *o(J)* is the output of the node *J*. In case of one node in the hidden layer *J=1*, while when two nodes exist, *J* is either 1 or 2. $W_{I,J}$ represent the synaptic weights.

Hyperbolic tangent function

$$o(J) = \frac{1 - \exp\left(-2 \cdot \sum_{I=0}^{N} W_{I,J} \cdot XN_I\right)}{1 + \exp\left(-2 \cdot \sum_{I=0}^{N} W_{I,J} \cdot XN_I\right)} \tag{12}$$

For the models without bias, $W_{I,0}=0$

Radial Basis function

$$o(J) = \exp\left(-\frac{\sum_{I=1}^{N} \left(XN_I - X0_I\right)^2}{\sigma_I^2}\right) \tag{13}$$

Where ($X0_1$, $X0_2$ … $X0_N$) is the vector of the center of the radial basis function, while (σ_0, σ_1 …σ_N) are variance parameters to be also trained by ANN.

Output layer activation function

A linear function has been used in this case, given by (14):

$$YN = \sum_{J=1}^{N_1} V(J) \cdot o(J) \quad (14)$$

Where N_1=1 or 2 depending on the number of nodes in the hidden layer. Then *YN* is transformed to calculated 28-day strength according to equations (4) or (10).

The more complicated structure of the cascade ANN is depicted in Fig. 2.

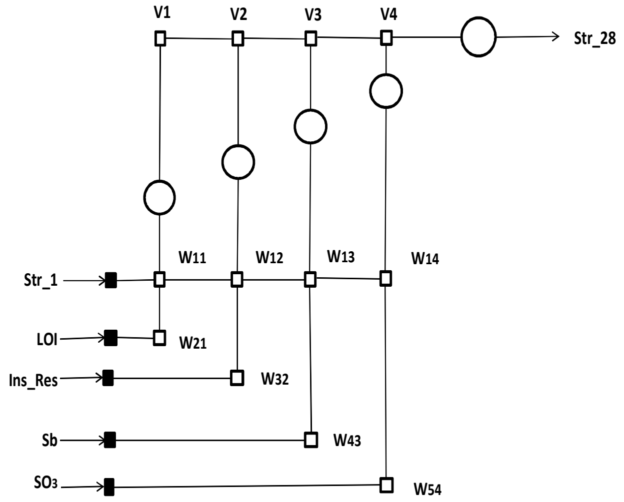

Fig. 2 Cascade ANN with four hidden layers

Cascade ANN has been applied for the prediction of concrete strength by Badde et al. [31]. To develop this kind of ANN the technique described by Shetinin [32] has been followed. Sigmoid functions have been utilized in all the hidden layers, while the output layer, combines linearly the outputs of each hidden layer with the weights V_I, I=1..4. In the first hidden layer *Str_1* and *LOI*, the two most significant inputs as regards their impact on the *Str_28*, are fed and combined with the synaptic weights W_{11}, W_{21}. The output *o(1)* is multiplied with the weight V_1 and fed to the output node. These three synaptic weights are trained in batch mode and then tested. The second step involves a new layer were *Str_1* and *Ins_Res* are fed. The weights W_{12}, W_{32}, V_1, V_2 are trained and tested as previously. The building of new hidden layers follows by adding each time *Str_1* and a one of the remaining variables. The algorithm stops when the addition of a new variable does not decrease further the training error, or results in an increase of testing error. The above means that the ANN is probably over-fitted. By applying this algorithmic logic, R40 was not added to the cascade ANN as it caused a worsening of the testing error.

III. Results and Analysis

A. *MSRE of training and test sets*

The dynamical models are applied for a movable training period T_D=180 days. Then the parameters of each model are applied to the results of T_F days which constitute the testing set. This period starts at least 29 days after the last date of the training period. The calculated and actual results of 28-day strength during T_F are compared. Thus, the errors $MSRE_{Past}$, $MSRE_{Futur}$ are determined, for T_D=180 and T_F=30, according to the formulae (6), (7) and shown in Table II.

Table II. Models *MSRE* for T_D=180, T_F=30

No	Model	$MSRE_{Past}$	$MSRE_{Future}$
1	MLR	1.664	2.010
2	S_1N	1.658	2.000
3	S_1N_B	1.641	1.990
4	HT_1N	1.726	2.041
5	HT_1N_B	1.646	2.008
6	RBF_1N	1.685	2.070
7	S_2N	1.610	2.077
8	RBF_2N	1.664	2.124
9	CASC	1.722	2.002

From Table 2 it is observed that the dynamic approach of the linear regression model is really efficient: Only four out of the eight ANN models, provide an equivalent or lower testing error. The rest more complicated ANNs with two nodes in the hidden layer or with the radial basis functions provide worse $MSRE_{Futur}$. The simple S_1N_B with one node and bias, provide the lower testing errors. Model S_2N, despite its lower training error, fails to predict the future strength better than MLR, fact that is normally due to over-fitting. The results of Table 2 do not lead to a clear comparison among the several models. Consequently a deeper analysis is necessary. Because $MSRE_{Future}$ is composed from the *MSRE* of each testing set, probably there are outlying values. To exclude such outliers the following techniques is implemented.

- In Table 2 the initial $MSRE_{Future}$ for each model are shown.
- Using the normal distribution and predefined probabilities of rejection, the corresponding z values are computed.
- Sets with $MSRE > z \cdot MSRE_{Future}$ are excluded.
- The percentage of excluded sets, P_r, is calculated.
- $MSRE_{Future}$ is recalculated from the remaining errors.
- For each model a curve of $MSRE_{Future}$ as function of P_r is designed and the models are compared.
- The comparison is shown in Fig. 3; in (a) and (b), MLR is compared with ANNs of higher $MSRE_{Future}$, while in (c) and (d) with the ANN models of improved error.

From these Figures the following remarks can be made for the set (T_D, T_F) = (180, 30):

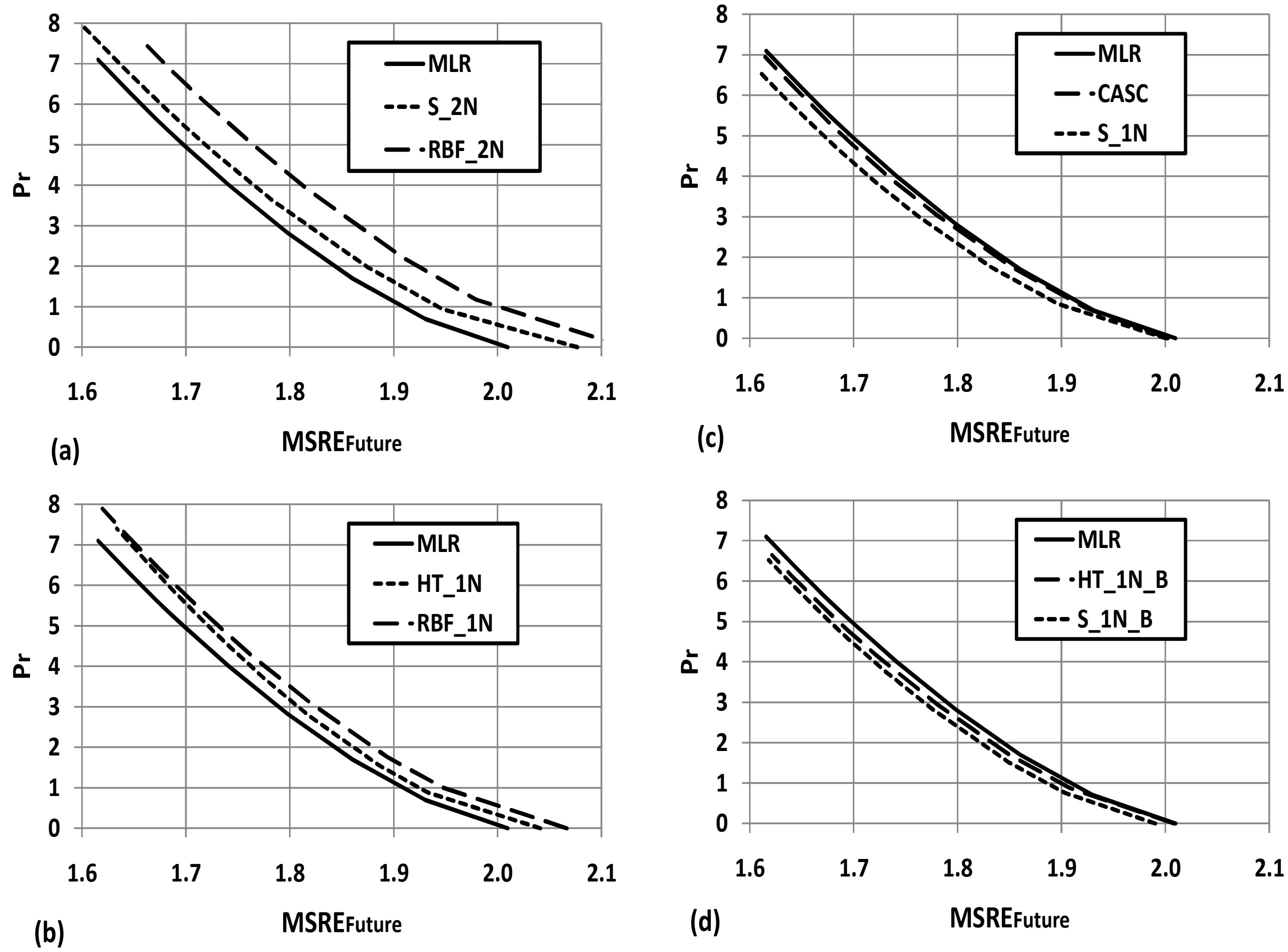

Fig. 3 Comparison of MLR and ANN

The selected ANN configurations based on RBF are continuously worse than MLR. Especially when two nodes are used, despite the lower training error, that is equal to that of MLR, the testing error is higher. This is an indication of over-fitting. In case of ANN with sigmoid activation function, when two nodes are used causing an $MSRE_{Past}$ lower than that of MLR, the test error is again higher. Thus S_2N model seems that over-fits the data. The HT_1N model is also worse than MLR, which probably means that some significant parameter is missing.

Cascade ANN is the one that is closer to MLR as concerns $MSRE_{Future}$, for all the range of P_r, but is the most complicated ANN and needs the higher computational time. In any case it is the first ANN model from the chosen ones, with improved performance compared to the highly efficient and simple MLR model. Concerning HT_1S model, probably the addition of an independent variable could improve its performance. Such parameter is the bias; HT_1S_B model behaves much better; its $MSRE_{Future}$ is always lower than that of MLR as shown in Fig. 4 (b). Both models S_1N and S_1N_B provide training and test errors lower than the respecting of MLR for the pair (T_D, T_F) = (180, 30). The same is also verified for the test errors for the full range of P_r. Consequently these two models are the optimal among the eight ANN configurations chosen. Probably a specific handling of the synaptic weights could improve the performance of some of the ANNs, but this is out of the scope of this paper.

B. Performance analysis of MLR and optimal ANN

The testing residual errors of MLR were compared with the ones of S_1N over all the range of testing sets, e. g. for all the K_{TD} sets and for P_r=5%. $MSRE_{Future}$ as function of testing set I, for I=1 to K_{TD}, is shown in Fig. 4 (a) for both models. The curves are generally of the same shape and the lower total MSRE of S_1N cannot be easily observed from this figure. The sorted differences of errors between MLR and S_1N (MLR-S_1N) are depicted in Fig. 4 (b) from where becomes obvious that the positive differences are more and higher than the negative ones resulting in a better performance of the S_1N ANN. For comparison reason the sorted differences of errors between S_2N and S_1N are shown in Fig. 4 (c). The over-fitting problem of the configuration selected in S_2N ANN, due to probably to the large number of the synaptic weights, is apparent.

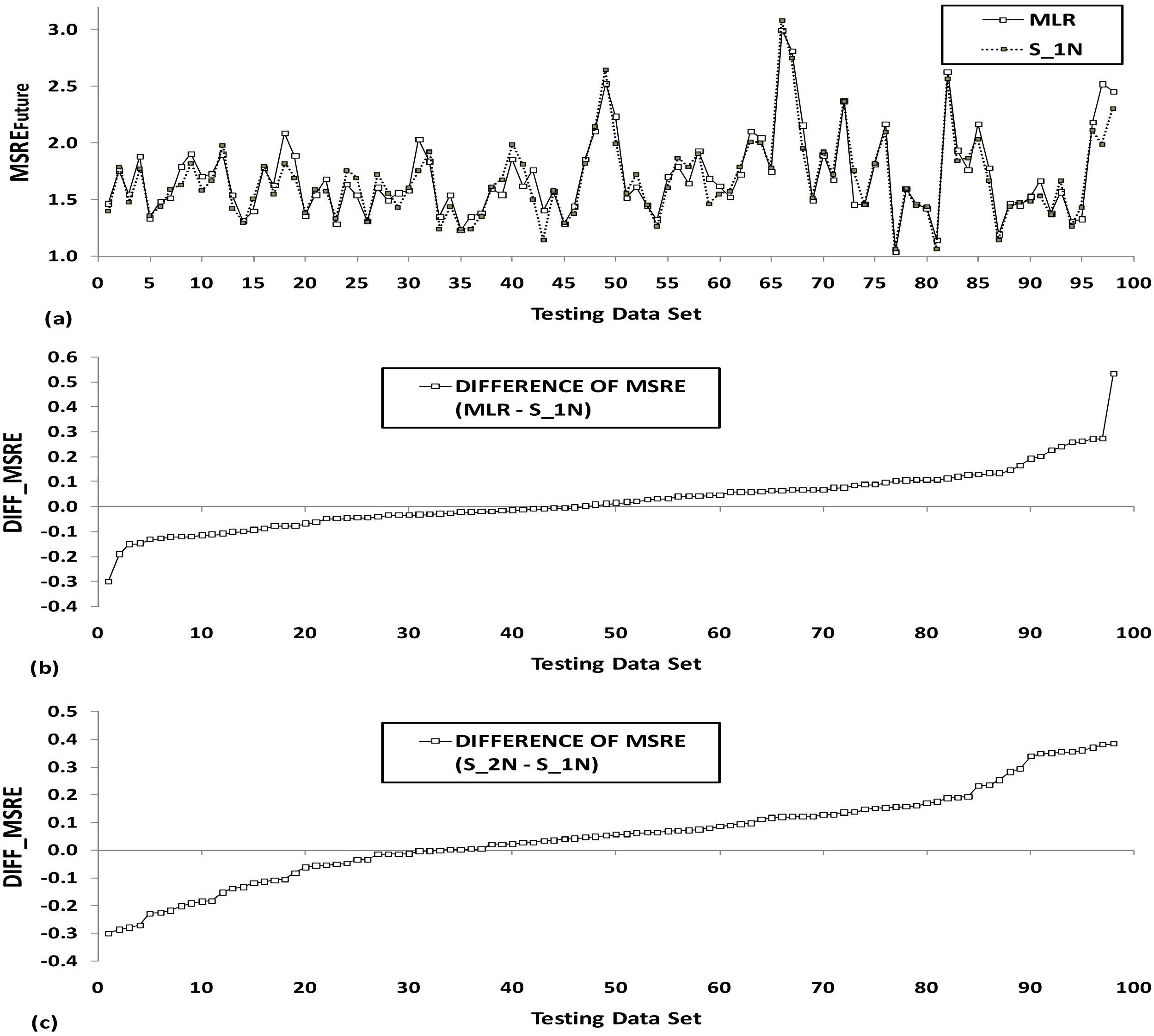

Fig. 4 (a) Comparison of MLR and S_1N errors (b) sorted differences of MSRE(MLR) – MSRE(S_1N)

Table III. Synaptic weights average and standard deviations

Weight	Average	Std. Dev.
W_0	0.87	1.47
W_1	-1.57	0.99
W_2	-0.05	0.38
W_3	-1.18	0.84
W_4	-0.59	0.52
W_5	-0.36	0.35
W_6	2.08	0.83
V_1	1.00	0.12

The addition of bias in the simple ANNs of one node with sigmoid or hyperbolic activation functions enhances their performance. The dynamical modeling has the ability to adjust the synaptic weights during time. To have a clearer picture for the variance of each W_I and V_I the average values and standard deviations are computed and shown in Table III for the model S_1N_B. The relatively high variance of the synaptic weights can be characterized as a measure of the ability of the dynamic model to adjust these weights, except if some weights are not statistically significant and could be neglected. This latter analysis is out of the scope of this study.

C. Minimization of MSRE of testing sets

For the basic time horizons (T_D, T_F) = (180, 30) several types of ANN have been already studied, four out of them providing an *MSRE* in testing sets lower than that of MLR model. This group of ANNs needs further investigation, in order to find the time periods of minimal values of $MSRE_{Future}$. In the current study and for the models MLR, S_1N, S_1N_B, HT_1N_B and CASC a search was performed to optimize T_F period : This parameter was permitted to vary from 1 day to 60 days and the respecting *MSRE* of training and testing sets were calculated. $MSRE_{Past}$ is relatively invariant as function of T_F. On the contrary the results indicate that there is a strong function between testing errors and T_F. This function for the several models is shown in Fig. 5 and the following can be observed:

- For all T_F except T_F=1, *MSRE* of cascade ANN is continuously lower than that of MLR. This type of ANN presents a minimum error for T_F=5.
- The *MSRE* of the remaining neural networks is kept always lower than that of MLR and except for T_F=5, it shows a monotonic decrease. Consequently the minimum error of testing sets is obtained for T_F=1 day.

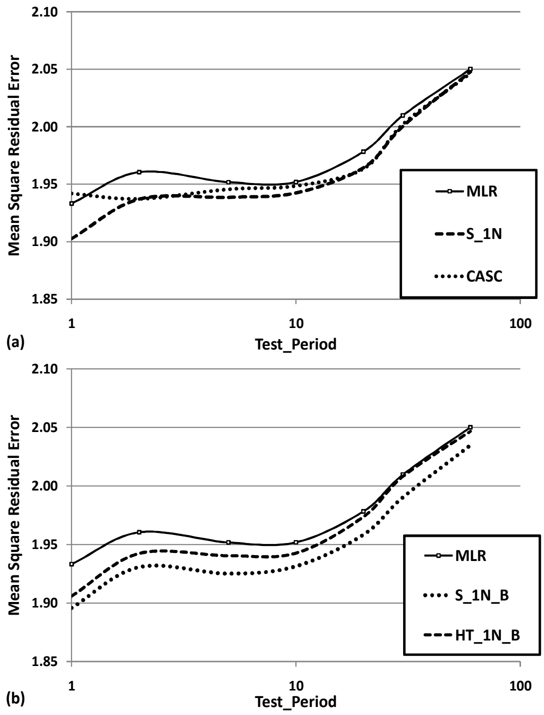

Fig. 5 MSRE of testing sets as function of T_F

- Among all the ANNs, the minimum *MSRE* is achieved for S_1N_B model and T_F=1, i.e. these are the optimum model and the future testing period. Therefore the implementation of this dynamic model in the daily strength prediction needs an updating of the equations, every day that new results appear.

IV. Conclusions

Two largely applied modeling techniques have been used to simulate and predict the typical 28-day compressive cement strength: Multiple linear regression (MLR) and artificial neural networks (ANN). The modeling is restricted to Portland cement produced according to EN 197-1:2011 and utilizes analyses of daily average samples of cement industrially produced. Physical and chemical results as well the 1-day compressive strength are used to predict the typical cement strength. A variety of ANNs has been developed involving three layers with one or two nodes in the hidden layer. In parallel a cascade ANN has been built. Three types of activation functions have been implemented; sigmoid, hyperbolic tangent and radial basis functions. The comparison is based on the MSRE of testing sets. The dynamic approach of MLR is an efficient tool for strength prediction; only four out of the eight ANN models developed, show an MSRE lower than this of the linear model. Cascade ANN is one of them, but needs much more computational time without to provide a highly better performance than MLR. The three ANNs of higher performance are the simplest ones, with only one node in the hidden layer and sigmoid or hyperbolic tangent function. The introduction of bias improves testing error. More complicated networks probably need a further processing because they are suffering from over-fitting.

An optimization of the future testing period, T_F, was performed and the optimal value was equal to 1day. The optimum model is the dynamical ANN, with three layers and one node in the hidden layer, which includes bias and sigmoid activation function.

References

[1] F. F. Steege and H. M. Grob, “Comparison of Long-Term Adaptivity for Neural Networks” in *Proc. 22th. Int. Conf. on Artificial Neural Networks (ICANN 2012)*, Lausanne, Switzerland, 2012, Part II, pp. 50-57.

[2] M. A. Oleiwi and Z. A. Aljasem, “Neural Network Control for PPDCV Clinker Cooler System*”, Int. J. Comput. Appl.*, vol 85, no 2, pp. 29-34, January 2014.

[3] E. B. Karikov, V. G. Rubanov and V. K. Klassen, “Construction of a Dynamic Neural Network Model as a Stage of Grate Cooler Automation”, *World Appl. Sci. J.*, vol. 25, no 2, pp. 227-232, 2014.

[4] O. Kadri, L. H. Mouss and M. D. Mouss, “Fault diagnosis of rotary kiln using SVM and binary ACO”, *J. Mech. Sci. Technol.*, vol. 26, no.2, pp. 601-608, 2012.

[5] V. Singh, P. K. Banerjee, S. K. Tripathy, V. K. Saxena and R. Venugopal, “Artificial Neural Network Modeling of Ball Mill Grinding Process”, *Powder Metallurgy & Mining*, vol. 2, no. 2, pp. 1-4, 2013.

[6] J. R. Gómez Sarduy, J. P. Monteagudo Yanes, M. E. Granado Rodríguez, J. L. Quiñones Ferreira and Y. Miranda Torres, “Determining cement ball mill dosage by artificial intelligence tools aimed at reducing energy consumption and environmental impact” *Ingenieria Investicacion*, vol. 33, no. 3, pp. 49-54, December 2013

[7] U. Yildiran and M. O. Kaynak, “Neural network based control of a cement mill by means of a VSS based training algorithm”, in *Proc. of the 2002 IEEE International Symposium on Industrial Electronics, 2002(ISIE 2002),* L'Aguila, Italy, 2002, vol. 1, pp.326-331.

[8] I. Odler, “Cement Strength”, *Mater. Struct.*, vol. 24, no 2, pp. 143-157, March 1991.

[9] D. Tsamatsoulis, “Prediction of cement strength: analysis and implementation in process quality control”, *Journal of Mechanical Behavior of Materials*, vol. 21, no 3-4, pp. 81-93, 2012.

[10] I. C. Yeh, “Modeling of Strength of High-Performance Concrete Using Artificial Neural Networks”, *Cem. Concr. Res.*, vol. 28, no. 12, pp. 1797–1808, 1998.

[11] J. Z. Wang, X. J. Wang and H. J. Ni, “An Algorithm of Neural Network and Application to Data Processing in Concrete Engineering”, *Informatica*, vol. 14, no. 1, pp. 95-110, 2003.

[12] M. Özturan, B. Kutlu and T. Özturan, “Comparison of Concrete Strength Prediction Techniques with Artificial Neural Network Approach”, *Building Research Journal*, vol. 56, no. 1, pp. 23-36, 2008.

[13] A.A.R. Heshmati, H. Salehzade , A.H. Alavi, A.H. Gandomi and M. Mohammad Abadi, “A Multi Expression Programming Application to High Performance Concrete”, *World Appl. Sci. J.*, vol. 5, no 2, pp. 215-223, 2008.

[14] F. Özcan, C. D. Atis, O. Karahan, E. Uncuoglu, H. Tanyildizi, “Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete”, *Adv.Eng. Software*, vol. 40, no. 9, pp 856-863, September 2009.

[15] U. K. Nath , M. K. Goyal and T. P. Nath, “Prediction of Compressive Strength of Concrete using Neural Network”, *International Journal of emerging trends in engineering and development*, vol. 1, no. 1, pp. 32-43, August 2011.

[16] Y. Aggarwal1 and P. Aggarwal, "Prediction of Compressive Strength of SCC Containing Bottom Ash using Artificial Neural Networks", *World Acad. Sci. Eng. Technol.*, vol. 5, no. 5, pp. 735-740, 2011.

[17] S. Rao, S. Reddy, V, Ghorpade and C. Reddy, "Development of Genetic Algorithm Based Hybrid Neural Network Model for Predicting the Ultimate Flexural Strength of Ferrocement Elements", *International Journal of Engineering Science and Technology*, vol. 4, no. 3, pp. 867-873, 2012.

[18] S. Gupta, "Using Artificial Neural Network to Predict the Compressive Strength of Concrete containing Nano-silica", *Civil Engineering and Architecture*, vol. 1, no. 3, pp. 96-102, 2013.

[19] B. A. Omran, Q. Chen and R. Jin, "Prediction of Compressive Strength of "Green" Concrete Using Artificial Neural Networks", in *Proc. 50th ASC Annual International Conference Proceedings,* Blacksburg, Virginia, 2014.

[20] S. Akkurt, S. Ozdemir, G. Tayfur, G. and B. Akyol, "The Use of GA–Anns in the Modeling of Compressive Strength of Cement Mortar", *Cem. Concr. Res*., vol. 33, no 7, pp. 973-979, July 2003.

[21] A. Baykasoğlu, T. Dereli, and S. Tanış, " Prediction of Cement Strength Using Soft Computing Techniques", *Cem. Concr. Res.*, vol. 34, no. 11, pp. 2083-2090, November 2004.

[22] P. Thamma, and S. V. Barai, "Prediction of Compressive Strength of Cement Using Gene Expression Programming", *Advances in Intelligent and Soft Computing*, vol. 58, pp 203-212, 2009.

[23] Q. Zhang, B. Yang, L.Wang, and F. Zhu, "Predicting Cement Compressive Strength Using Double-Layer Multi-expression Programming", in *Proc Fourth International Conference on Computational and Information Sciences (ICCIS)*, 2012, pp. 94-97.

[24] M. Madsen, P. Thyregod, F. Popentiu, G. Albeanu, M. Coarna, L. Serbanescu, D. Mirza, and A. Tautan, "Considerations Concerning a Fuzzy - Genetic Algorithm with Application to CEM I Type Cements Quality Optimization", in *Proc. 9th World Multi-Conference on Systemic, Cybernetics and Informatics*, Orlando, vol. X, 2005, pp. 350-353.

[25] S. Ren, D. Yang, F. Ji, and X. Tian, "Application of generalized regression neural network in prediction of cement properties", in *Proc. International Conference on Computer Design and Applications (ICCDA)*, vol. 2, 2010, pp. 385-388.

[26] Z. Yongzheng, X. Biyun, and Z. Yong, "The prediction research about the strength of cement based on principal component neural network", in *Proc. IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, 2011, pp. 238-241.

[27] D. Tsamatsoulis, "Control Charts and Models Predicting Cement Strength: A Strong Tool Improving Quality Control of Cement Production", in *Proc. 16th WSEAS International Conference on SYSTEMS*, Kos, Greece, 2012, pp. 136-145.

[28] D. Tsamatsoulis, "Application of the static and dynamic models in predicting the future strength of pozzolanic cements", in *Proc. 18th International Conference on Circuits, Systems, Communications and Computers*, Santorini, Greece, 17-21 July 2014, Latest trends in Systems vol. I, pp. 138-146.

[29] S. Haykin, *Neural Networks. A comprehensive Foundation.* Delhi: Prentice Hall, 2005, pp. 24.

[30] S. Lawrence, C. Lee Giles and A. C. Tsoi, "Lessons in Neural Network Training: Overfitting May be Harder than Expected", in *Proc. Fourteenth National Conference on Artificial Intelligence*, Menlo Park, California, 1997, pp. 540–545.

[31] D. S. Badde, A. K. Gupta, and V. K. Patki, "Cascade and feed forward back propagation artificial neural network models for prediction of compressive strength of ready mix concrete," *IOSR Journal of Mechanical and Civil Engineering*, Second International Conference on Emerging Trends in Engineering, Civil Group, vol. 3, pp. 1–6, 2013.

[32] V. Shetinin,"A Learning Algorithm for Evolving Cascade Neural Networks", *Neural Processing Letter*, vol. 17, pp. 21-31, 2003.