Prediction of Cement Compressive Strength by Combining Dynamic Models of Neural Networks

D. Tsamatsoulis[*]
Devnya Cement AD, HeidelbergCement
Group, Devnya 9160, Industrial Zone,
Bulgaria

doi: https://doi.org/10.15255/CABEQ.2021.1952
Original scientific paper
Received: April 1, 2021
Accepted: July 15, 2021

This study aimed at developing models predicting cement strength based on shallow neural networks (ANN) using exclusively industrial data. The models used physical, chemical, and early strength results to forecast those for 28- and 7-day. Neural networks were trained dynamically for a movable period and then used for a future period of at least one day. The study includes nine types of activation functions. The algorithms use the root mean square errors of testing sets ($RMSE_{Future}$) and their robustness as optimization criteria. The $RMSE_{Future}$ of the best models with optimum ANNs was in the range of 1.36 MPa to 1.63 MPa, which is near or within the area of long-term repeatability of a very competent laboratory. Continuous application of the models in actual conditions of a cement plant in the long-term showed a performance at least equivalent to that calculated during the design step.

Keywords:
cement, compressive strength, modeling, neural network, optimization

Introduction

Machine learning techniques and especially artificial neural networks (ANN) are widely used in predicting cement and concrete properties for quality control purposes. The product's compressive strength is the mainly utilized property, cumulatively expressing the quality. The most significant among the strength measured at different ages is considered to be the 28-day strength, at least for normative reasons. Compressive strength depends on a variety of the product's physical and chemical properties, which are due either to the raw materials used or acquired during the production processes. Successful implementation of ANNs and other machine learning techniques in forecasting concrete strength is referred to in numerous research papers, for multiple technologies of production and categories of this product[1–9]. In the cement industry, due to the high daily production rates and the particular impact of a quality accident, the 28-day strength forecast based on earlier analysis results has gained great interest from the past. Extensive reviews of published studies based on multilinear or polynomial models can be found[10,11].

In recent years, significant research has been elaborated on machine learning methodologies in predicting cement strength based on laboratory or industrial quality data sets. Many researchers have applied several techniques of this type, such as genetic algorithms (GA), ANNs, fuzzy logic (FL), support vector machines (SVM), support vector regression (SVR), adaptive-neuro-fuzzy inference algorithms (ANFIS), and other more specialized methods. Akkurt *et al.*[12] structure contained a GA – ANN correlating cement compressive strength with six months-worth of industrial quality data. Their results indicated that the increase in tricalcium silicate (C_3S), sulfates (SO_3), and specific surface led to increased strength. In further publication, Akkurt *et al.*[13] implemented an FL model to the same data set and compared the results with them of the ANN[12]. Thamma *et al.*[14] employed gene expression programming to predict the 28-day cement mortar strength, compared the results and those of FL and GA-ANN models, initially developed by Accurt *et al.*[12,13], and concluded that gene expression programming performs better than FL. Verma *et al.*[15] employed three different kernel-based models – SVR, relevance vector machine, and Gaussian process regression – in predicting cement strength. Afterwards, they compared the models with ANN and FL models provided by Akkourt *et al.*[12,13] using the same data set. Similarly, Motamedi *et al.*[16] used SVR and ANFIS algorithms in forecasting the compressive strength of cockle shell-cement-sand mixtures. Their findings showed that ANFIS improved the generalization capability compared to SVR. Chen *et al.*[17] provided an approach based on SVM for predicting the compressive strength of cement mortars exposed to sulfate attack. They compared

[*] Corresponding author: E-mail: d.tsamatsoulis@halyps.gr

this model with an ANN containing one hidden node, and concluded that SVM performance was higher than that of ANN. Neural networks were also used in two other cases[18,19]. Escadari-Nadaff *et al.*[18] developed a structure in predicting cement mortars strength, using three different cement strength classes to their mixtures. Pani *et al.*[19] used robust multivariate techniques and soft sensor models based on ANNs and FL to detect the outliers in the quality of clinker industrially produced. Most of these models, estimating the parameters from a determined training data set, and predicting the future strength, are characterized as static.

Tsamatsoulis[20,21] introduced a dynamic approach to predicting the strength of cement using movable time horizons based on long-term process results and several types of ANNs. The models incorporate the uncertainty due to the time variability of non-involved factors during the modeling –especially the clinker reactivity- and are dynamic. In this manner, the author provides a robust solution to the challenging issue of predicting strength during the daily quality control of cement production in a cement plant. Apart from chemical and physical measurements, these studies utilize results of early strength in predicting the typical 28-day strength. This technique uses data of a predetermined period for training, namely, for computing optimal parameters. Subsequently, the data belonging to a time interval that follows the training period are used for validating – or in other words, testing the model. The training and validation time intervals constitute the past and future periods, respectively. Tsamatsoulis[21] proved that the optimal future period is the 1-day interval, whereas the training time interval needs optimization to achieve the best prediction. This previous study employed three activation functions types: Sigmoid, hyperbolic tangent, and radial basis functions. It also used linear functions for comparison reasons. The aim of this study was threefold. Firstly, to deepen in and explore the ANN architectures implemented to forecast output, using the capability of a model to predict the future cement strength as a criterion, which can be characterized as the generalization ability[22] of modeling. Subirats *et al.*[23] clearly stated that one of the strategies for avoiding overfitting is the search for compact architectures. In this research, shallow ANNs were built with a varying number of nodes within a single hidden layer. A simple and robust method of pruning connections between nodes of input variables and the input layer was also examined and exploited, involving the correlations among input and output variables. Additionally, we used weight decay[24–26] using L2-regularization[27] for all the architectures concerned. The second purpose of this study was to explore the behavior of various activation functions (AF) in the application under examination, as recent work[28] demonstrated that AFs are of high relevance, and that research in this field was constantly evolving[29]. The third objective, though not least, was to build robust models predicting cement 28- and 7-day strength, able to be applied in the daily quality control of a cement plant. Specialized algorithms appropriately add up the models' outputs, considering the time delays, aiming at augmenting generalization ability. Throughout the present text, the errors corresponding to past periods are characterized as training errors, while the ones computed from future period data, for validation purposes, are named test errors.

Materials and testing methods

The application field of modeling comprised four cement types, produced according to EN 197-1:2011 (Cement, Part 1: Composition, specifications and conformity criteria for common cements): CEM I 52.5 R, CEM I 42.5 R, CEM II A-LL 42.5 R, CEM II B-LL 42.5 R. Cements of CEM I type contain clinker and gypsum as main constituents. Moreover, limestone is one of the main components in the denoted CEM II types. Table 1 shows the raw materials' typical chemical characteristics. The oxides analysis was performed with XRF. Table 1 also presents the mineral composition of the clinker, according to Bogue equations (1) – (4). These formu-

Table 1 – *Materials chemical analyses and clinker mineral composition*

	LOI	SiO_2	Al_2O_3	Fe_2O_3	CaO	MgO	SO_3	K_2O	Na_2O
	%	%	%	%	%	%	%	%	%
Clinker	0.50	20.89	5.20	3.63	65.72	1.18	1.00	0.93	0.43
Gypsum	21.84	0.90	0.32	0.19	31.76	0.12	44.75	0.05	0.02
Limestone	43.15	0.99	0.26	0.10	54.94	0.35	0.10	0.06	0.01

	C_3S	C_2S	C_3A	C_4AF
	%	%	%	%
Clinker	61.6	13.5	7.7	11.0

lae use the following notation: C=CaO, S=SiO_2, A=Al_2O_3, F=Fe_2O_3, and CaO_f denote the free lime of the clinker.

$$C\ S = 4.07 \cdot \left(C - CaO\ \ \right) - 7.6 \cdot S - 6.72 \cdot A - 1.43 \cdot F \quad (1)$$

$$C_2S = 2.87 \cdot S - 0.754 \cdot C_3S \quad (2)$$

$$C_3A = 2.65 \cdot A - 1.69 \cdot F \quad (3)$$

$$C_4AF = 3.04 \cdot F \quad (4)$$

Modeling utilized the results of composite daily average samples of cement produced in mills of the Devnya plant. Each of these daily samples corresponded to a cement type produced by a specific mill. Training and verification of the models included more than 1600 datasets of daily cement results produced in about 27 months. The cement's physical, chemical, and mechanical features, and the analysis methods are shown in Table 2, resulting in a population of 11 input and output variables. Equation (5) of section "Mathematical models" normalizes the values of these variables for the entire population. Fig. 1 demonstrates the frequency distributions of normalized values. Table 3 indicates the mean, minimum, and maximum values of these variables for each cement. These data clarify that the dispersion of values is quite large, providing an initial generalization capability to the models under consideration.

Mathematical models

The physical and chemical characteristics, namely, the residue at 45 μm sieve, loss on ignition, and SO_3, SiO_2, Al_2O_3, Fe_2O_3, CaO oxides, constituted the set of shared input variables in all the models elaborated. The chemical properties characterize both cement composition and clinker reactivity. The initial design included three basic models in predicting strength:

(i) a model predicting 28-day strength named Str_28_2, where the 1- and 2-day strength – *Str_1*, *Str_2,* respectively, were added as input variables to the basic set;

(ii) a second model which also predicted 28-day strength named Str_28_7, where the 7-day strength –*Str_7*- was also included except *Str_1*, *Str_2*;

(iii) a third model forecasting 7-day strength named Str_7_2 where physical, chemical features, and *Str_1*, *Str_2* made up the input variables.

Table 2 – *Physical, chemical, mechanical characteristics*

Characteristic	Norm	Description
R45, residue at 45 μm sieve (%w/w)	EN 196-6	Methods of testing cement – Part 6: Determination of fineness –air jet sieving
LOI, Loss on ignition (%w/w)	EN 196-2	Methods of testing cement – Part 2: Chemical analysis of cement
SO_3, SiO_2, Al_2O_3, Fe_2O_3, CaO (%w/w)	EN 196-2	Methods of testing cement – Part 2: Chemical analysis of cement – measurement by XRF
1-, 2-, 7-, and 28-day compressive strength (MPa)	EN 196-1	Methods of testing cement – Part 1: Determination of strength

Table 3 – *Average values of cements characteristics*

Characteristic	CEM I 52.5 R	CEM I 42.5 R	CEM II A-LL 42.5 R	CEM II B-LL 42.5 R	Minimum value	Maximum value
R45 (%w/w)	0.6	7.3	4.6	5.8	0.3	9.4
LOI (%w/w)	1.89	2.68	7.37	9.79	0.82	12.19
SO_3 (%w/w)	3.71	3.41	3.31	2.93	2.25	4.13
SiO_2 (%w/w)	19.54	19.27	17.10	16.06	15.13	20.24
Al_2O_3 (%w/w)	4.82	4.75	4.22	3.97	3.38	5.16
Fe_2O_3 (%w/w)	3.44	3.39	3.04	2.86	2.60	4.25
CaO (%w/w)	63.67	63.63	62.35	61.92	60.73	67.21
1-day strength (MPa)	28.4	18.4	19.5	17.4	12.0	36.5
2-day strength (MPa)	41.0	30.3	31.9	28.9	24.8	47.8
7-day strength (MPa)	52.0	43.4	42.8	39.9	35.0	57.8
28-day strength (MPa)	61.0	53.5	50.8	47.4	43.0	67.2

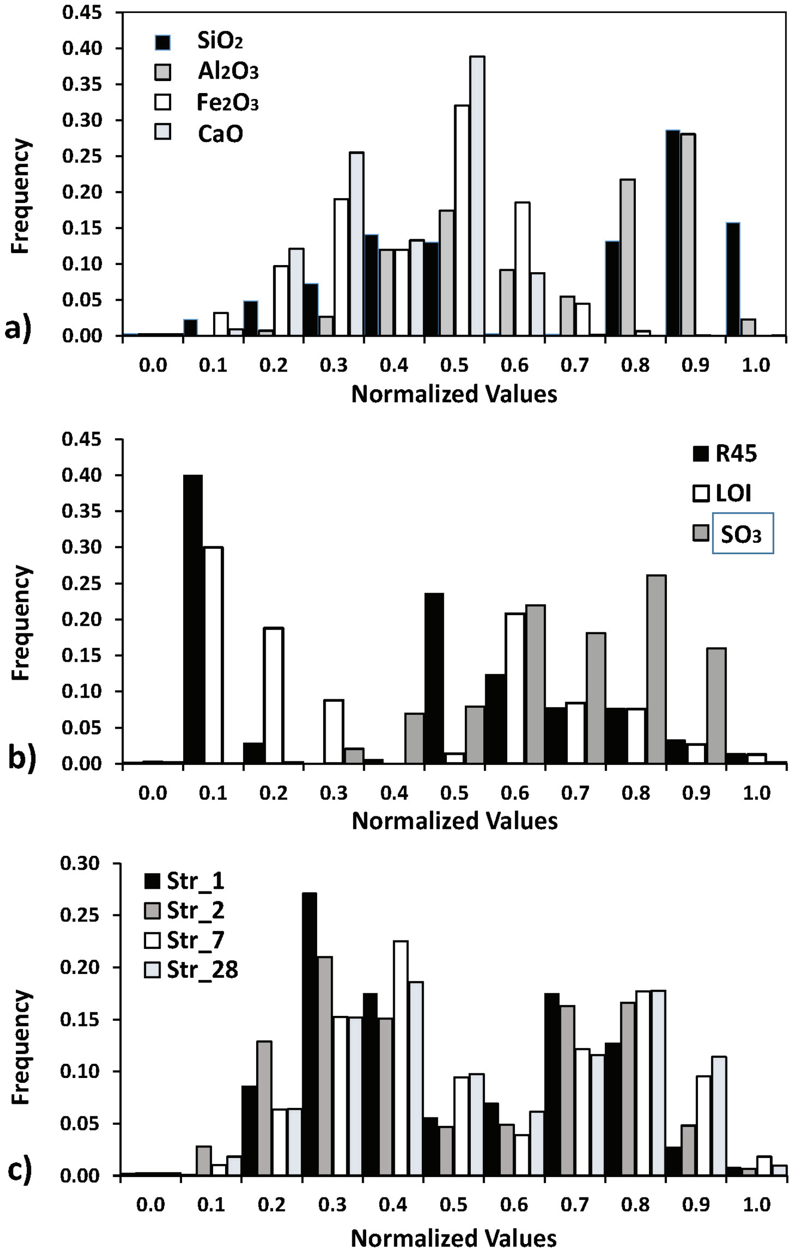

Fig. 1 – *Frequency distribution of normalized values of physical, chemical, and mechanical characteristics*

Therefore, the maximum number of input variables for models (i) and (iii) was nine, while for model (ii) it was ten. Usually, the plant laboratory prepares the mortar for the strength measurements the day after producing a cement batch. Thus, in terms of the derived predictions, the following time delays appeared for each model. The delay for models (i) and (iii) was three days, and the prediction delay for (ii) was eight days. Apart from the three basic models, the design included two additional ones, named C1_28_2 and C2_28_2. The C1_28_2 model adds to the strength predicted from Str_28_2, a correction term derived from Str_28_7. The C2_28_2 model exploits the result of Str_7_2 to add a similar quantity to the original forecasting of Str_28_2. As described by Tsamatsoulis[21], there is a stronger or weaker correlation among several of the model's physical, chemical, and mechanical inputs. For example, an increase in the contained limestone causes an increase in LOI and a decrease in SiO_2, Al_2O_3, Fe_2O_3, CaO. The content of these oxides, as well as their ratio, is also related to clinker reactivity. Early strength not only depends on R45 and chemical analysis, but also on "hidden" variables such as milling performance and clinker reactivity. Therefore, the input variables contain an independent part permitting to describe the process better. The models use the early strengths to include such hidden variables as clinker activity and grinding conditions in a cumulative manner. Despite that the

C_3S computed from equation (1) is an activity indication, its introduction to the models as an independent variable has weaknesses if the goal is to use the models in daily quality control:

(a) For a plant laboratory it is difficult to measure the average clinker consumed in each cement batch and each cement mill in a daily program. If the mills have a shared clinker silo, then the discrimination between each mill is very hard. Conversely, if there is one silo per mill, then a very intensive sampling and analysis program is required, hard to accomplish. Usually, the cement plants concentrate their efforts on measuring and stabilizing the clinker in the kiln outlet.

(b) The C_3S calculated by the Bogue formula accumulates the uncertainties of each measurement on a spot basis.

The residue in some sieve or the specific surface is a partial indicator of grinding effectiveness and cement particle size. The measurement of particle size of each cement component after grinding is hard or impossible to accomplish in a daily program. The early strengths increase as the clinker activity increases or the clinker fraction in the cement becomes finer and reversely. Therefore, this is the rationale to introduce those strengths in the models.

The input and output variables for model Str_28_2 are named as follows: X_1=SiO$_2$, X_2=Al$_2$O$_3$, X_3=Fe$_2$O$_3$, X_4=CaO, X_5=SO$_3$, X_6=LOI, X_7=R45, X_8=*Str_1*, X_9=*Str_2*, *Y*=*Str_28*. For Str_28_7, X_{10}=*Str_7* was also added in the inputs, whereas model Str_7_2 for the same inputs as Str_28_2 derives output *Y*=*Str_7*. These variables were normalized as follows:

(1) For a given training and test set of data, the minimum and maximum values of the input variables X_I and the output *Y*, $X_{I,MIN}$, $X_{I,MAX}$, Y_{MIN}, Y_{MAX}, respectively, were computed.

(2) Equations (5) and (6) provide the normalized variables, depending on the activation function utilized in the ANN.

$$XN_I = \frac{X_I - X_{I,MIN}}{X_{I,MAX} - X_{I,MIN}} \quad with \quad I = 1 \text{ to } N_I;\; YN = \frac{Y - Y_{MIN}}{Y_{MAX} - Y_{MIN}} \tag{5}$$

$$XN_I = \frac{2 \cdot X_I - \left(X_{I,MIN} + X_{I,MAX}\right)}{X_{I,MAX} - X_{I,MIN}} \text{ with } I = 1 \text{ to } N_I; YN = \frac{2 \cdot Y - \left(Y_{MIN} + Y_{MAX}\right)}{Y_{MAX} - Y_{MIN}} \tag{6}$$

(3) Correspondingly, the predicted strength Str_{Calc} is back-calculated by equations (7) or (8).

$$Str_{Calc} = Y_{MIN} + YN \cdot \left(Y_{MAX} - Y_{MIN}\right) \tag{7}$$

$$Str_{Calc} = \frac{Y_{MAX} + Y_{MIN}}{2} + Y_N \cdot \frac{Y_{MAX} - Y_{MIN}}{2} \tag{8}$$

In the case of models (i) or (ii) application, $Str\_28_{Calc} = Str_{Calc}$. Otherwise, if 7-day strength is to be predicted, $Str\_7_{Calc} = Str_{Calc}$.

Shallow neural networks

Several architectures of feed-forward ANNs' with three layers, one of which is hidden, have been developed. The number of nodes in the hidden layer varied, aiming at succeeding the optimum generalization performance. Therefore, the examined ANNs were shallow and variable in width. All the software had been developed in C#. Before feeding the data of a training dataset to the input layer, a preprocessing of them preceded, as follows:

(i) For the given training set of size N_{Tr}, the correlation factors $Correl_I$, between each input variable X_I (for I=1 to N_I) and the 28-day strength, were computed.

(ii) A positive threshold of $Correl_{Min}$ was assumed.

(iii) XN_I is fed to the input layer only if the absolute value $|Correl_I| \geq Correl_{Min}$.

The described comparator acts as a filter of the input variables. If X_I passes the filter, then the filter's output is XF_{IP}=XN_I, else I=I+1. Thus, *IP* lies in the range from 1 to N_{IP}, where the total number of variables passed is $N_{IP} \leq N_I$. Afterwards, for each dataset belonging to the training set, the activation function of each node *J* of the hidden layer accepts the linear combination of XF$_I$s computed from formula (9).

$$Z_J = \sum_{I=0}^{N_{IP}} W_{IJ} \cdot XF_I \tag{9}$$

where XF_0=1, to take into account the bias and *J*=1 to N_N.

Activation functions

Nine activation functions (AF) were examinednd implemented in the software developed. The set of AFs comprised equations traditionally applied in ANNs and others developed over the last years. A short description of them follows:

(1) Linear function (Linear):

$$a(Z) = Z \tag{10}$$

The identity function, having its input as output, can be characterized as a special AF, where the ANN can have only one node in the hidden layer.

(2) Sigmoid function (Sigmoid):

$$\sigma(Z)=\frac{1}{1+\exp(-Z)} \qquad (11)$$

The sigmoid function is frequently referred to as the logistic function, and its results belong to the interval (0, 1).

(3) Hyperbolic tangent function (Tanh)

$$a(Z)=\frac{1-\exp(-Z)}{1+\exp(-Z)} \qquad (12)$$

The Tanh function lies within the range of –1 to 1 and is zero centered.

(4) Hard hyperbolic tangent function (HardTanh)

$$a(Z)=\max(-1,\min(Z,1)) \qquad (13)$$

The HardTanh function represents a computationally "cheaper" version of Tanh.

(5) Radial basis function (RBF)

$$a(Z)=\exp(-Z);\; Z=\sum_{I=1}^{N_{IP}}\left(\frac{XF_I - X0_{IJ}}{\sigma_{IJ}}\right)^2 \qquad (14)$$

where ($X0_{1J}$, $X0_{2J}$... $X0_{NIP\,J}$) is the vector of the RBF center, while (σ_{0J}, σ_{1J}... σ_{NIPJ}) are variance parameters for the node J, where J=1 to N_N.

(6) Rectified linear unit function (ReLU)

$$a(Z) = \max(0,Z) \qquad (15)$$

The ReLU function, proposed and used by Nair *et al.*[30], is the most widely used activation function for deep learning applications[29].

(7) Sigmoid-weighted linear unit function (SiLU)

$$a(Z) = Z \cdot \sigma(Z) \qquad (16)$$

(8) Derivative of sigmoid-weighted linear unit function (dSiLU)

$$a(Z)=\sigma(Z)\cdot\left(1+Z\cdot\left(1-\sigma(Z)\right)\right) \qquad (17)$$

The SiLU and dSiLU functions were introduced and applied by Elfwing *et al.*[31] The authors had been motivated by the high performance of similar equations in both classification and reinforcement learning.

(9) Softsign (SoftSign)

$$a(Z)=\frac{Z}{1+|Z|} \qquad (18)$$

Softsign function, introduced by Turian *et al.*[32], has been used in regression problems[33]. However, its performance is to be evaluated in this study by applying it in cement's strength prediction.

For each hidden layer's node J, the input of AF is Z_J, computed by equation (9). As for normalized inputs XF_I, the algorithm applied formulae (5) for the AFs with equations (10) to (11) and (14) to (17). For the three AFs, whose output lay in the interval (–1, 1), i.e., those described from relations (12), (13), and (18), formulae (6) were used. The outputs of all nodes enter the output layer, which calculates the ANN normalized output as follows:

$$YN=\sum_{J=1}^{N_N} V_J \cdot YN_j \qquad (19)$$

The back-calculated Str_{Calc}s are computed via equation (7) if the AFs are provided from (10) to (11) and (14) to (17). Otherwise, equation (8) is applied.

Dynamic models

First of all, a short description is needed as to how the dynamic models process the experimental datasets to extract training and test period: (1) A certain number of days is assumed as a training period, T_{Tr}. (2) For a given date t, all the production samples belonging to the interval [t, t +T_{Tr}–1] constitute the training data set. (3) The algorithm trains models Str_28_2, Str_28_7, Str_7_2 by computing the corresponding ANNs' weights via non-linear regression methodologies. (4) The future test set of each mentioned model is found considering the date t +T_{Tr}–1 and the delay between production date and early strength measurement. The test period is the minimum possible time interval of one day if the production dates allow it. According to the quality plan of the plant quality system, physical, chemical tests, and mortar preparation are performed on daily average samples of each mill the day after the production date. A date variable, T_C, is considered to be the current date. During this date, the following strength results appear if production samples exist:

(a) 1-day strength result of cement produced 2 days ago, i.e., in T_C–2.

(b) 2-day strength result of cement produced 3 days ago, i.e., in T_C–3.

(c) 7-day strength result of cement produced 8 days ago, i.e., in T_C–8.

(d) 28-day strength result of cement produced 29 days ago, i.e., in T_C–29.

Appendix A presents the algorithms implementing the five models and a basic set of equations. Fig. 2 demonstrates the block diagram of future predictions for all the dynamic models. All variables have been transformed according to the Laplace transformation, and bold characters are used in the vectors representation. Depending on the model chosen, T_{D1} and T_{D2} parameters take the subsequent values: (1) For the Str_28_2 model, T_{D1}=3 d, T_{D2}=29 d. (2) For the Str_28_7 model, T_{D1}=8 d, T_{D2}=29 d. (3) For the Str_7_2 model, T_{D1}=3 d, T_{D2}=8 d. The synaptic weights of each of the three models are provided by the vectors $W_{Mod,J}$, for J=0 to N_N.

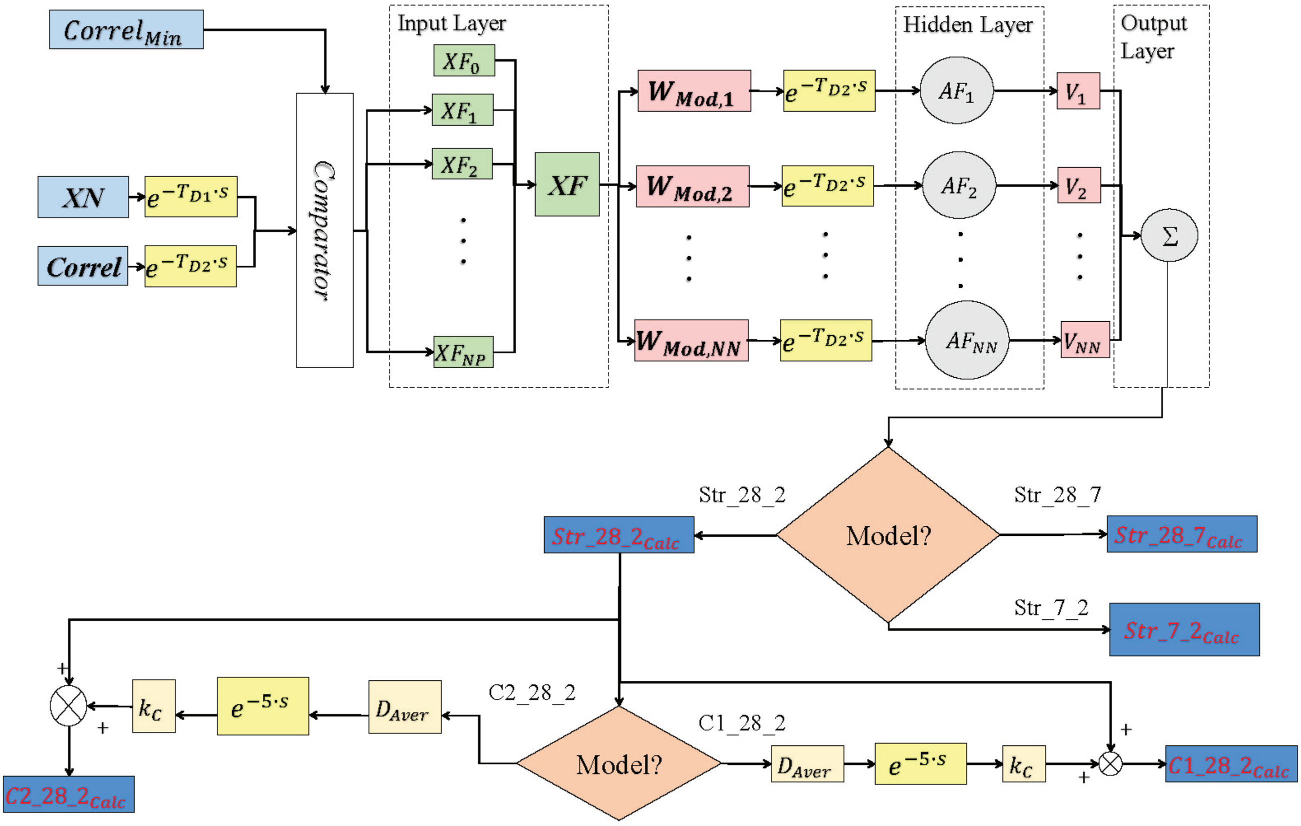

Fig. 2 – *Block diagram of dynamic models*

Results and discussion

The principal objective of the optimization attempted was the best generalization ability of the models, i.e., minimization of $RMSE_{Future}$. In succeeding this, the model with the optimized parameters would be utilized in the daily quality control of the studied cement types. The synaptic weights of each ANN were determined by minimizing the objective function given by equation (20), which does not imply that the residual error of the training set is minimal. As the purpose was to minimize the test error, the optimization of the λ parameter became highly significant. Another major factor requiring the optimum selection was the activation function type. This is the reason behind a large number of AFs being examined. Several other parameters also required optimization:

(i) Length of the training period.

(ii) Number of nodes in the hidden layer for each chosen AF.

(iii) Number of days, N_{pr}, and the multiplier k_C, described in steps (xiv) – (xvi) of the Appendix.

The length of the training period selected was within the range of 20 and 300 days. For these training periods, the number of consecutive training sets, N_{Tot}, was between 400 and 640. Table 4 shows the median value and the 10 % and 90 % percentiles of the data sets' number within each training set as a function of T_{Tr}. Concerning testing sets, the median value of the population within each one was two, and the 10 % and 90 % percentiles, 1 and 6, respectively. The small amount of data in each testing set was the reason for choosing RMSE as the single metric. The calculation of $RMSE_{Future}$ is always feasible. Choosing the widely used correlation coefficient may lead to erroneous conclusions for small populations or may be inapplicable. If the set

Table 4 – *Median and 10 %, 90 % percentiles of samples population per training set*

T_{Tr}	Percentile (%)			T_{Tr}	Percentile (%)		
(d)	50	10	90	(d)	50	10	90
20	44	32	51	140	286	269	314
30	65	48	75	160	326	309	359
40	85	65	99	180	368	351	400
50	106	83	121	200	410	390	440
60	127	100	143	220	451	430	483
70	147	120	165	240	490	470	524
80	168	142	185	260	531	509	558
100	208	185	227	280	576	550	598
120	246	224	272	300	616	593	635

has only one element, then the calculation of the regression coefficient is not possible because there is no standard deviation of the experimental data.

Not only all the AFs but also the impact of nodes' number on the resulting errors were thoroughly. For presentation of the results, the following abbreviations were used: AFabbr_XN, where "AFabbr" is the abbreviation of each AF referred to in section "Activation Functions", and "X" is the number of nodes.

Correlations between input and output variables

The correlation among all the input variables and *Str_28* was initially examined through the computation of correlation coefficients. Fig. 3 shows the percentile distribution of these statistics for the whole population of training data sets for a training period of 50 days. Fig. 3 also depicts a threshold $Correl_{Min}$=0.8. The correlation coefficients between *Str_28* and SiO_2, LOI, *Str_1*, *Str_2*, *Str_7* were continuously higher than 0.8 in absolute value. The other input variables had correlation coefficients less than 0.8, up to a certain percentile. These results shall be examined from a physical standpoint. Higher LOI corresponds to higher limestone content in the cement composition, which has a negative impact on the 28-day strength. Thus, the correlation between LOI and *Str_28* was strongly negative.

Lower $CaCO_3$ in the cement leads to a higher SiO_2, Al_2O_3, Fe_2O_3, explaining the positive correlation between *Str_28* and each of the three oxides.

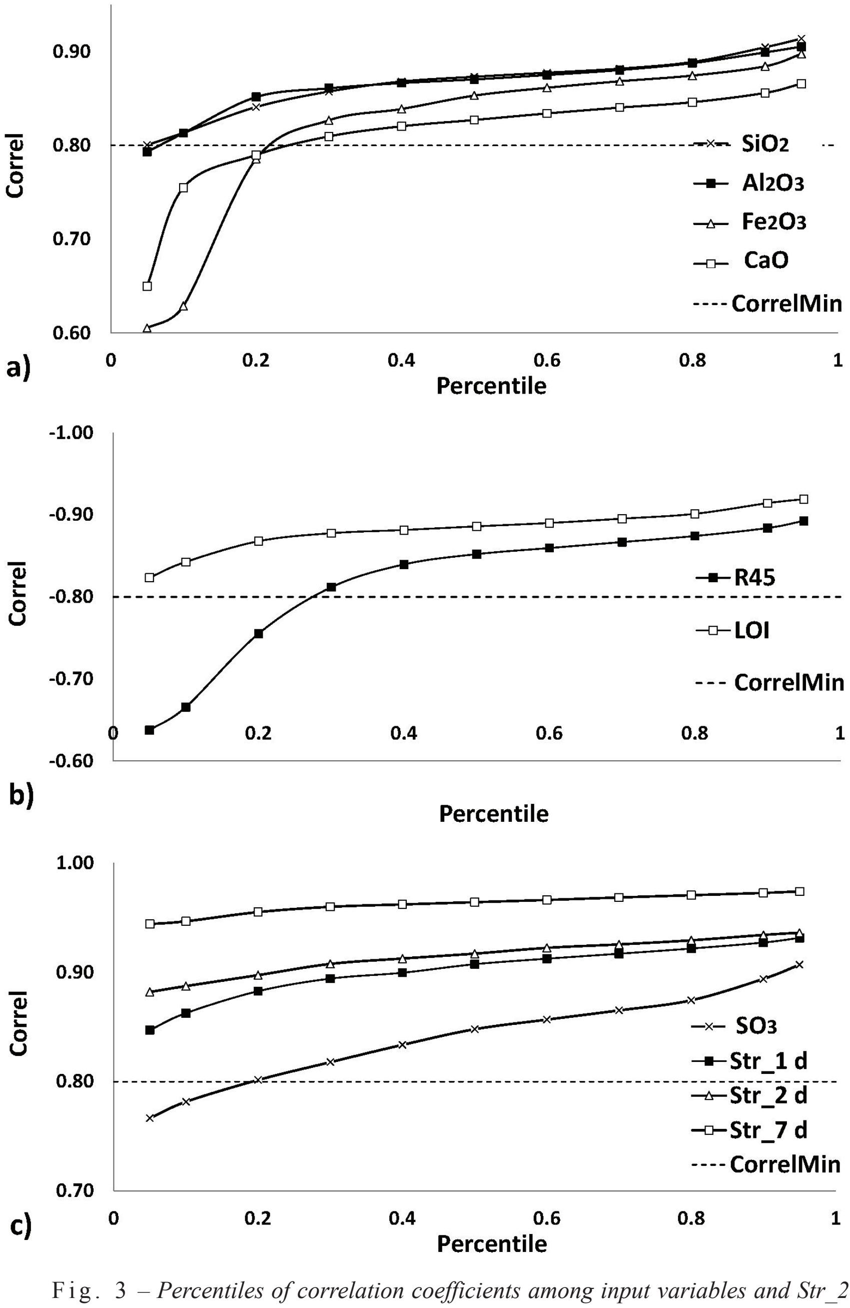

Fig. 3 – *Percentiles of correlation coefficients among input variables and Str_28*

On the other hand, CaO content in the cement was not only due to the clinker and gypsum, but also to the limestone. The result was a worsening of the correlation coefficient of CaO compared to those of the other three oxides. Table 3 indicates that types of cement with higher average LOI value, therefore lower clinker and strength, were designed with lower average SO_3 values, proving the positive correlation between SO_3 and *Str_28*. As expected, a high correlation existed between any early strength and the typical one of 28 days. The R45 shows a negative correlation with the *Str_28*, and a percentile of 0.3 presents a correlation less than 0.8 in absolute value. These low values were attributed to the impact of the grinding conditions of each cement mill, the wide range of compositions of cement types, and the variance of clinker reactivity.

A systematic search of the shape of the functions between *Str_28* and input variables follows. Concerning the early strength, the intervals of the percentiles from 0.1 to 1 with a step of 0.1 were computed. In each interval, the average values of early strength and *Str_28* were determined. The same procedure was also applied for R45. The results are plotted in Fig. 4. A significant non-linearity occurs in the function between *Str_1* and *Str_28* in Fig. 4a, as the shape of the curve is sigmoid. An increase in linearity exists as the early strength increases from 1-day to 7-day strength (Figs. 4b – 4c). Specifically, the function between 7-day and 28-day strength was strongly linear, meaning that the impact of the chemical and physical factors on both ages was similar for the cement types under consideration. The function between residue at 45 μm and *Str_28* in Fig. 4d verifies the results of Fig. 3b. The increase in strength for the last percentile of high R45 is explained from the data presented in Table 3: CEM I 42.5 R was the coarsest cement, but due to its high clinker content, it achieved a high typical 28-day strength.

Fig. 5 demonstrates the correlation between the chemical characteristics of each cement type and the 28-day strength. For each of these characteristics, the percentiles from 0.2 to 1 with a step of 0.2 were computed. The function between LOI and *Str_28* decreased for the entire population, but this trend was not verified for each cement type separately. For example, this function for CEM I 42.5 R increased non-linearly. The cause was attributed to other parameters, such as clinker activity, partly expressed by C_3S, and cement fineness partially explained by R45. The correlation between SO_3 and *Str_28* for each cement type generally passed from a weak optimum value or was flat, meaning that the selection of SO_3 target per cement type was in the optimal range. The functions between *Str_28* and the four oxides, SiO_2, Al_2O_3, Fe_2O_3, CaO, increased if taking all points as a whole. As concerns CEM I 42.5 R and CEM I 52.5 R, the types of the highest clinker content, the correlations for SiO_2, Al_2O_3, and Fe_2O_3 clearly decreased. The reason being that an increase of these oxides resulted in a decrease in clinker C_3S. For these cement types, an increasing function between *Str_28* and CaO was not observed as could be expected. This was attributed to the low limestone content permitted to be used as a minor component, which is also rich in calcium oxide.

Therefore, Figs. 4 and 5 contribute in detecting noticeable non-linearities, which hardly could be

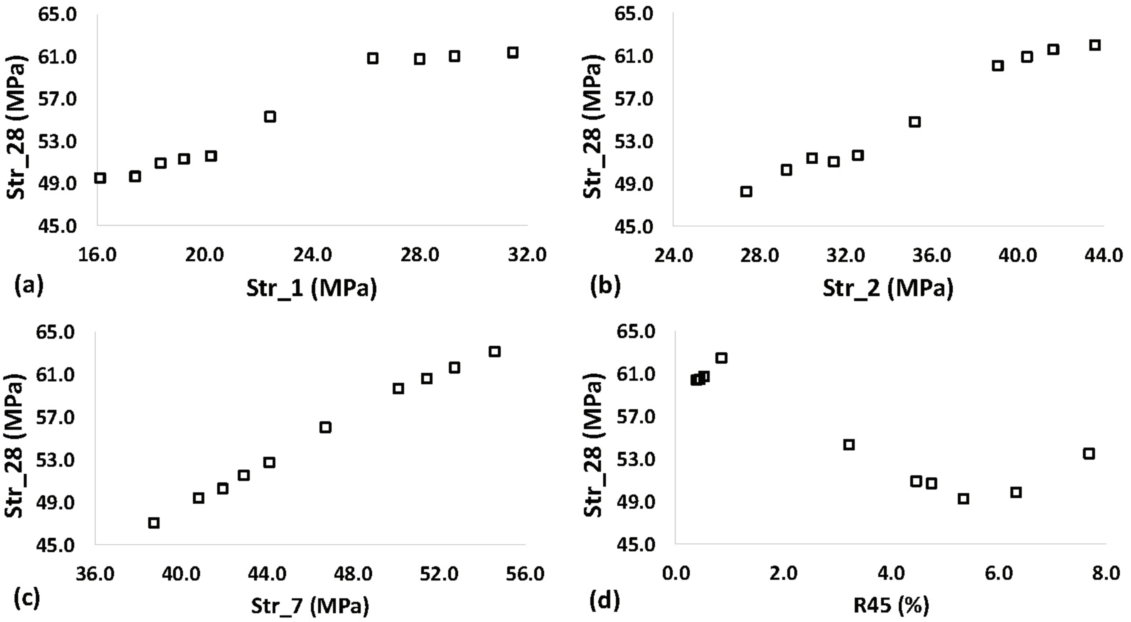

Fig. 4 – *Effect of early strength a) Str_1; b) Str_2; c) Str_7, and d) R45 residue on Str_28*

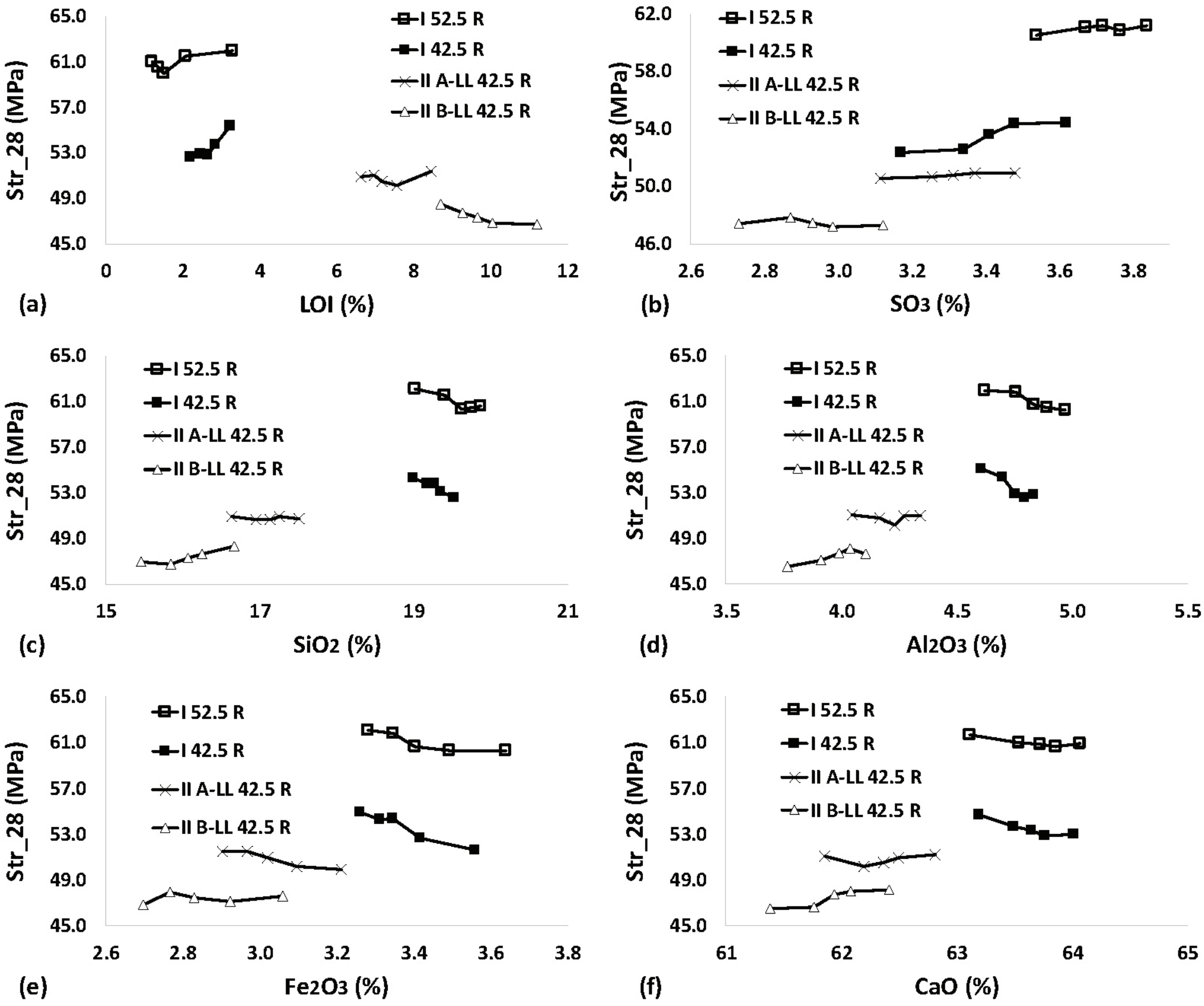

Fig. 5 – *Effect of chemical characteristics on Str_28*

described with multi-linear, polynomial, or simple exponential models. The above explains the selection of a deep analysis based on shallow ANNs.

Parameters' impact on the models' errors

Fig. 6 depicts the effect of the training period on $RMSE_{Future}$ for the models Str_28_2, Str_28_7, Str_7_2 by selecting several ANNs architectures and the majority of AFs. The optimal value of the λ multiplier was determined and used for each T_{Tr}. Each curve $RMSE_{Future}=f(T_{Tr})$ is convex, presenting a minimum for a definite training period. Initially, as T_{tr} increases, the testing error decreases. The cause of this behavior is the existence of values of variables not belonging to the values' range of these variables during the T_{Tr} period. Therefore, the models are obliged to extrapolate the computation leading to a worsening of prediction. Following this T_{Tr} value, an increase in the training period causes an increase in $RMSE_{Future}$, although the variables during the test period take values more likely to be within the values' range of the training period. As seen later, the function $RMSE_{Past} = f(T_{Tr})$ is increasing for larger values of T_{Tr}, which means that after a certain T_{Tr} level, an increase in training error leads to a worsening of generalization ability. One may notice that the minimum is not heavily acute. For an allowable range $[RMSE_{Future}, (1+\varepsilon)\cdot RMSE_{Future}]$, where ε is a small positive number, the corresponding T_{Tr}s are spread to a narrower or wider range. However, the correct training period selection is crucial: For the Str_28_2 model and Tanh_2N architecture, a choice T_{Tr}=260 days instead of the optimum of 50 days results in a 21 % higher error. Table 5 presents the ratio between the maximum and minimum $RMSE_{Future}$ for each ANN, and a wide range of training periods. The selection for the maximal training period is 200 days. The minimum T_{Tr} is 30 days when one or two nodes exist in the hidden layer, and 40 days in the case of three nodes. The results verify that the optimal selection of the training period is critical for all the models and ANNs.

Figs. 7a and 7b demonstrate the impact of T_{Tr} on $RMSE_{Past}$ for the model Str_28_2, implemented with the Tanh and ReLU as AFs, and using struc-

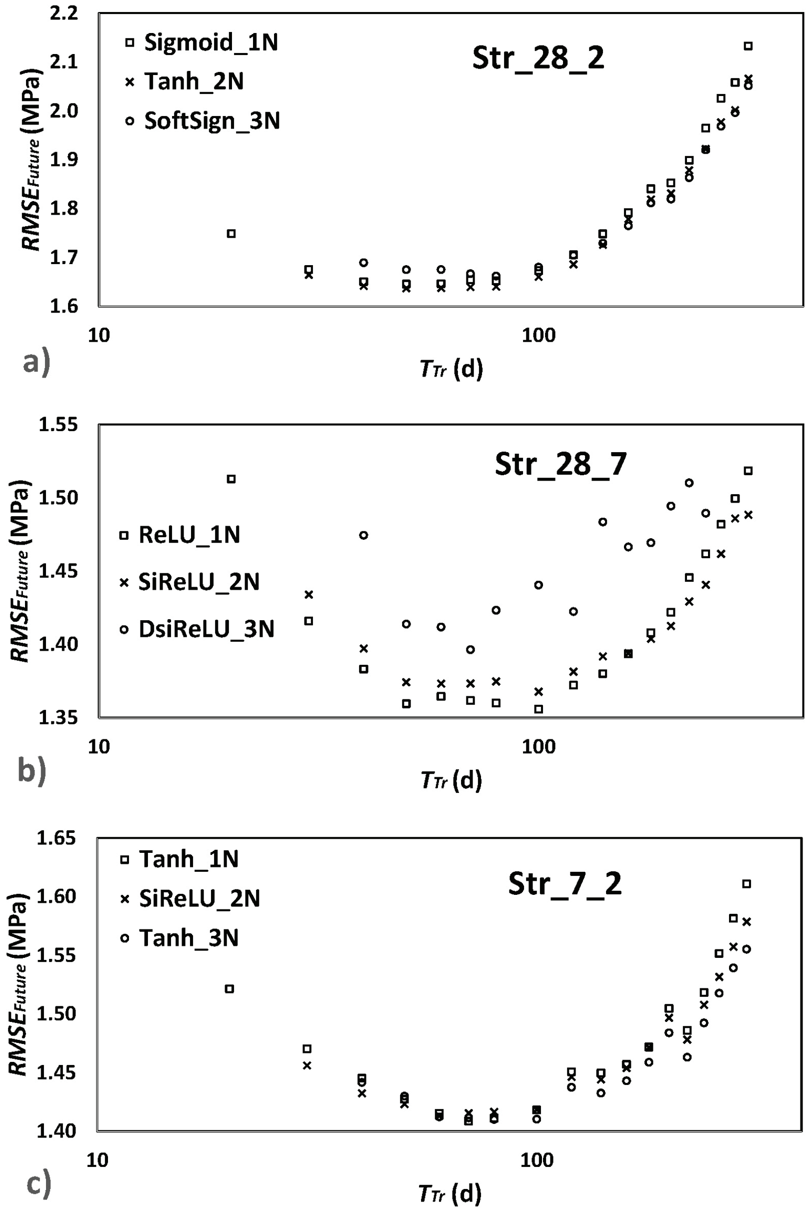

Fig. 6 – *Effect of the training period on $RMSE_{Future}$ for the models Str_28_2, Str_28_7, and Str_7_2*

Table 5 – *Ratio of Maximum to Minimum $RMSE_{Future}$*

ANN	Str_28_2	Str_28_7	Str_7_1
Linear_1N	1.22	1.21	1.20
Sigmoid_1N	1.13	1.07	1.06
Sigmoid_2N	1.10	1.07	1.06
Tanh_1N	1.12	1.06	1.07
Tanh_2N	1.12	1.06	1.06
Tanh_3N	1.11	1.04	1.05
HardTanh_1N	1.12	1.06	1.08
HardTanh_2N	1.11	1.08	1.10
HardTanh_3N	1.08	1.07	1.08
RBF_1N	1.06	1.13	1.10
ReLU_1N	1.14	1.05	1.07
ReLU_2N	1.13	1.08	1.08
ReLU_3N	1.08	1.14	1.15
SiLU_1N	1.13	1.05	1.06
SiLU_2N	1.12	1.05	1.07
dSiLU_1N	1.14	1.05	1.07
dSiLU_2N	1.15	1.06	1.11
dSiLU_3N	1.14	1.08	1.09
SoftSign_1N	1.08	1.07	1.09
SoftSign_2N	1.09	1.10	1.07
SoftSign_3N	1.09	1.06	1.05

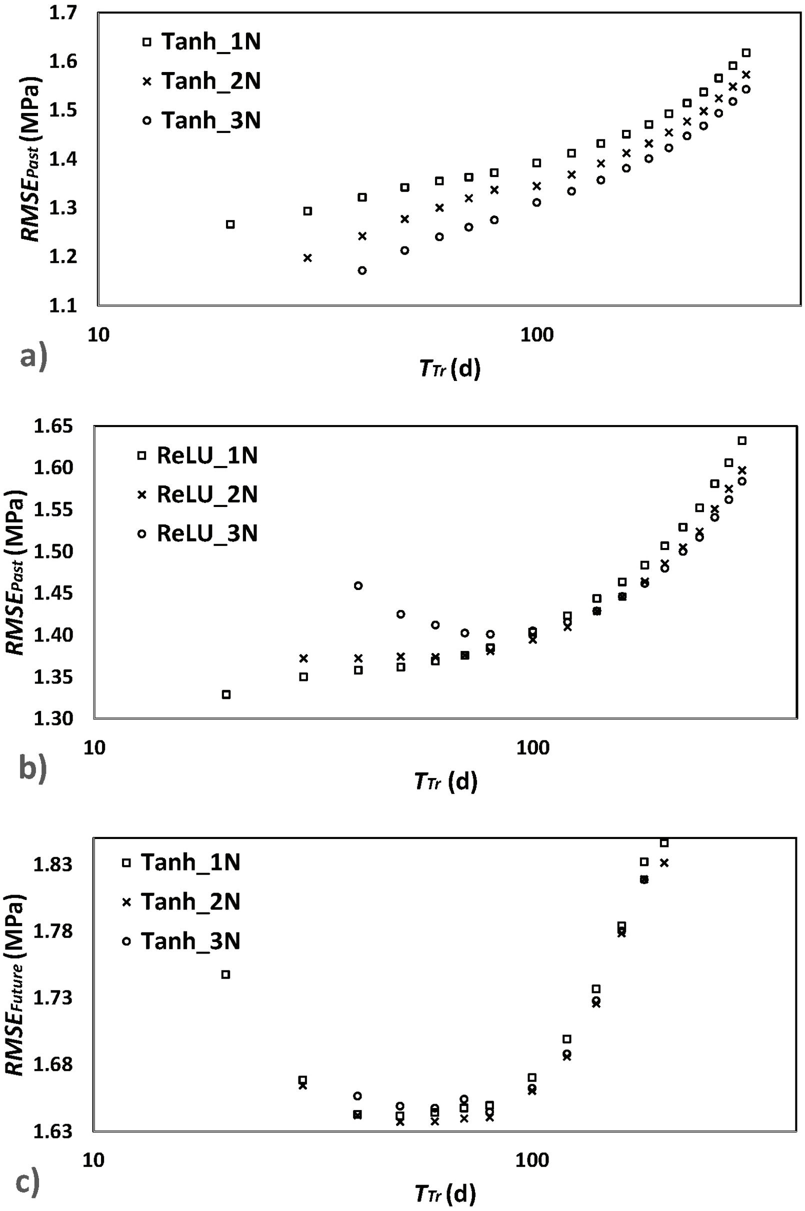

Fig. 7 – *Effect of the training period on a) – b) $RMSE_{Past}$; c) $RMSE_{Future}$ using Tanh, for the model Str_28_2*

tures of one, two, and three nodes within the hidden layer. The training errors were determined by zeroing the λ multiplier in equation (20) to obtain the minimal ones. The functions $RMSE_{past}=f(T_{Tr})$ are increasing or pass from a minimum value. Fig. 7c. shows the corresponding test errors for the model Str_28_2 and the Tanh activation function, where the λ's value optimizing the $RMSE_{Future}$ was determined for each training period and ANN's architecture. The comparison of Figs. 7a and 7c leads to the subsequent conclusions: As far as training errors are concerned, $RMSE_{Past}$ decreases as the number of nodes increases. Fig. 7c does not verify this trend, especially in the area of optimum test errors: These errors of the ANN with two nodes are smaller than are those of the architectures with one and three nodes.

Fig. 8 depicts training and testing errors as a function of T_{Tr} for the model Str_28_7, SoftSign, and SiReLU functions, using one to three nodes in the hidden layer. One can observe that, for the same AF and T_{Tr}, increasing the node number, both types of error decrease. With the two-node case, the two ANNs behave differently, especially in the region of training periods from 50 to 100 days, which provide the optimum testing errors: While the training error of the *SoftSign_2N* is less than that of the *SiReLU_2N*, as to the corresponding testing errors the inequality is reversed. This proves the effect of AF type on the resulting errors.

Fig. 9 demonstrates the correlation between the λ-multiplier and the testing error for several combinations of ANNs' architectures and AFs, for the Str_28_2 model. For each combination, the training

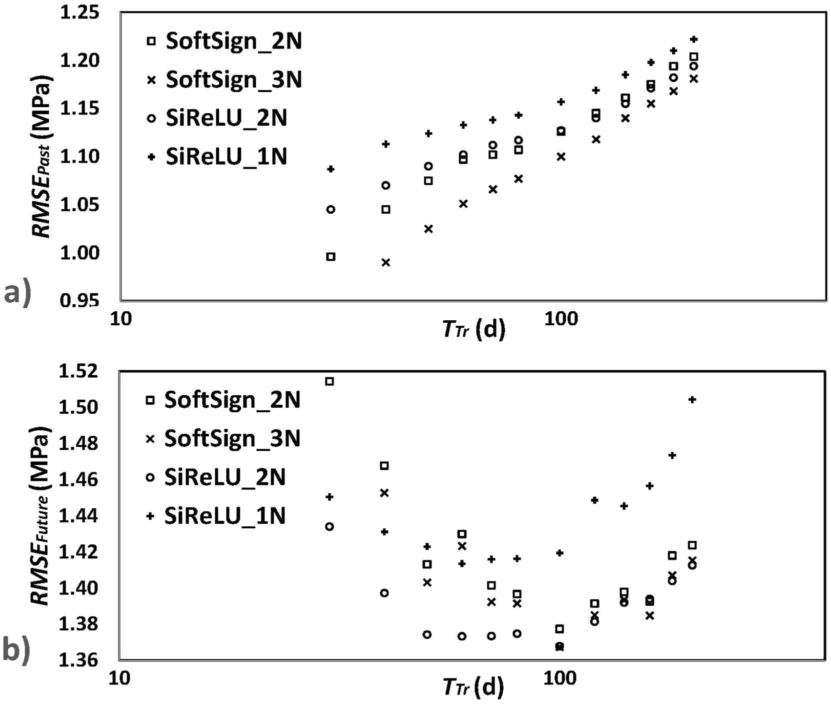

Fig. 8 – *Effect of the training period on a)* $RMSE_{Past}$*; b)* $RMSE_{Future}$*, for the model Str_28_7*

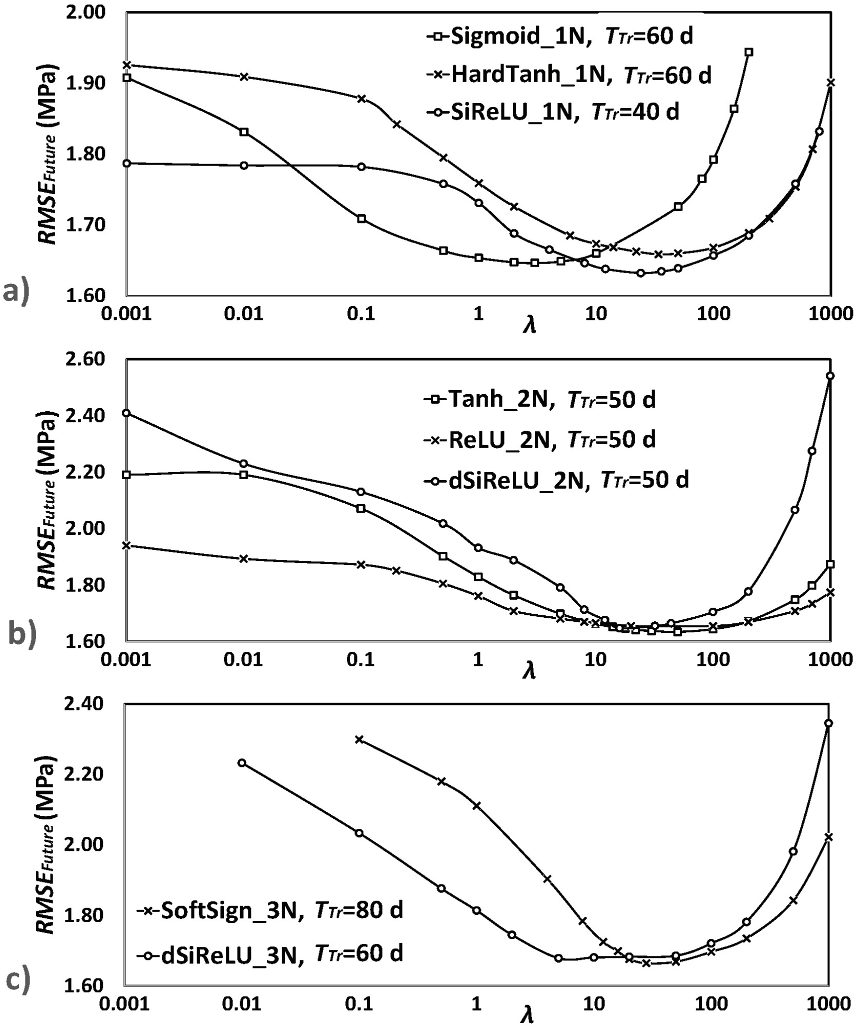

Fig. 9 – *Impact of λ-multiplier on* $RMSE_{Future}$ *for various AFs and ANNs with a) one node, b) two nodes, and c) three nodes in the hidden layer, for Str_28_2*

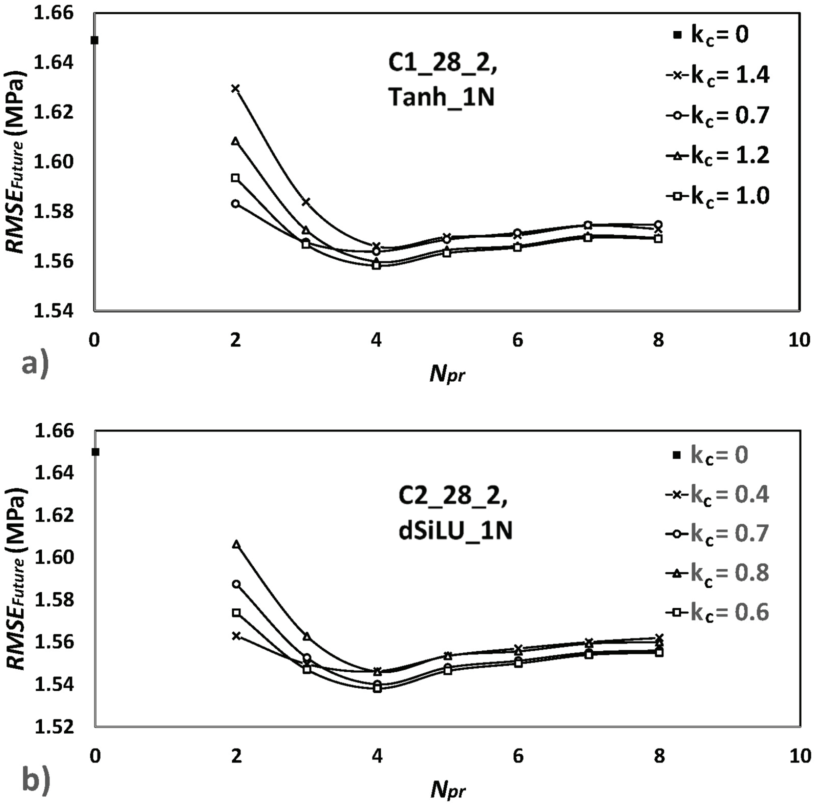

Fig. 10 – *$RMSE_{Future}$ as a function of N_{Pr} and k_C for a) C1_28_2, and b) C2_28_2 models*

period corresponds to that providing the minimum *$RMSE_{Future}$* for the optimum λ. A wider or narrower but distinct range of minimum errors occurs in each curve. The shapes of the curves differ noticeably, and the minimal errors are located in a large region of λ for the eight cases presented. One could see that, in the majority of the cases, the λ of optimum error is between 10 and 100. However, the optimum value of this multiplier is 3 for Sigmoid_1N, whereas it is 50 for Tanh_2N. Therefore, no general rule exists for the multiplier's selection, and a case-by-case search is needed. The minimal error of each curve is significantly less than the corresponding error for $\lambda = 0$, concluding that L2-regularization is an effective technique for improving generalization ability.

An optimal selection of the N_{pr} and k_c parameters appearing in equations (29) – (30) results in a significant diminishment of the test errors of C1_28_2, C2_28_2 models compared with those of Str_28_2. Fig. 10 shows the *$RMSE_{Future}$* as a function of the mentioned two parameters for both models, using two different AFs. The test errors of Str_28_2 also appear in this Figure. It is worth noting that the optimal value of λ is not necessarily the same for the Str_28_2 model and the two models under consideration.

Optimal neural networks

The combination of activation functions with the number of nodes results in 21 different ANNs. All the cases studied present an optimum *$Correl_{Min}$* threshold between 0.5 and 0.7 for the minimal *$RMSE_{Future}$*. Figs. 11a, 11b, 11c show the minimal mean test errors of each investigated ANN corresponding to optimal *$Correl_{Min}$*, *T_{Tr}*, and λ values for the models Str_28_2, Str_28_7, and Str_7_2. To enlarge the height of each column, in each Figure, the maximum value of the Y-axis is 10 % greater than the minimum error. The black columns represent testing errors up to 1 % higher than the minimum *$RMSE_{Future}$*, whereas the errors shown with grey columns are at least 10 % higher than the minimum one. Therefore, the 1 % margin is considered as the optimality criterion. The optimal point of the ANNs Tanh_1N, Tanh_2N, Tanh_3N, ReLU_1N, SiLU _2N, dSiLU_1N is continuously found in the optimum region for all three models. Additionally, in two out of the three models, ANNs SiLU_1N and SofSign_3N behave optimally. For any model, the ANN with linear activation functions failed to meet the optimality criterion. Some tests performed with four-node ANNs gave *$RMSE_{Future}$* worse than those of three-nodes ANNs, which is a clear indication of overfitting, concluding that a systematic search of such ANNs is not needed.

The robustness of generalization ability for a particular ANN can be expressed by the range of training periods having a testing error in a suboptimal region. In each model, as such region, the interval [*$RMSE_{Future,Min}$*, $\beta \cdot$ *$RMSE_{Future,Min}$*] is assumed,

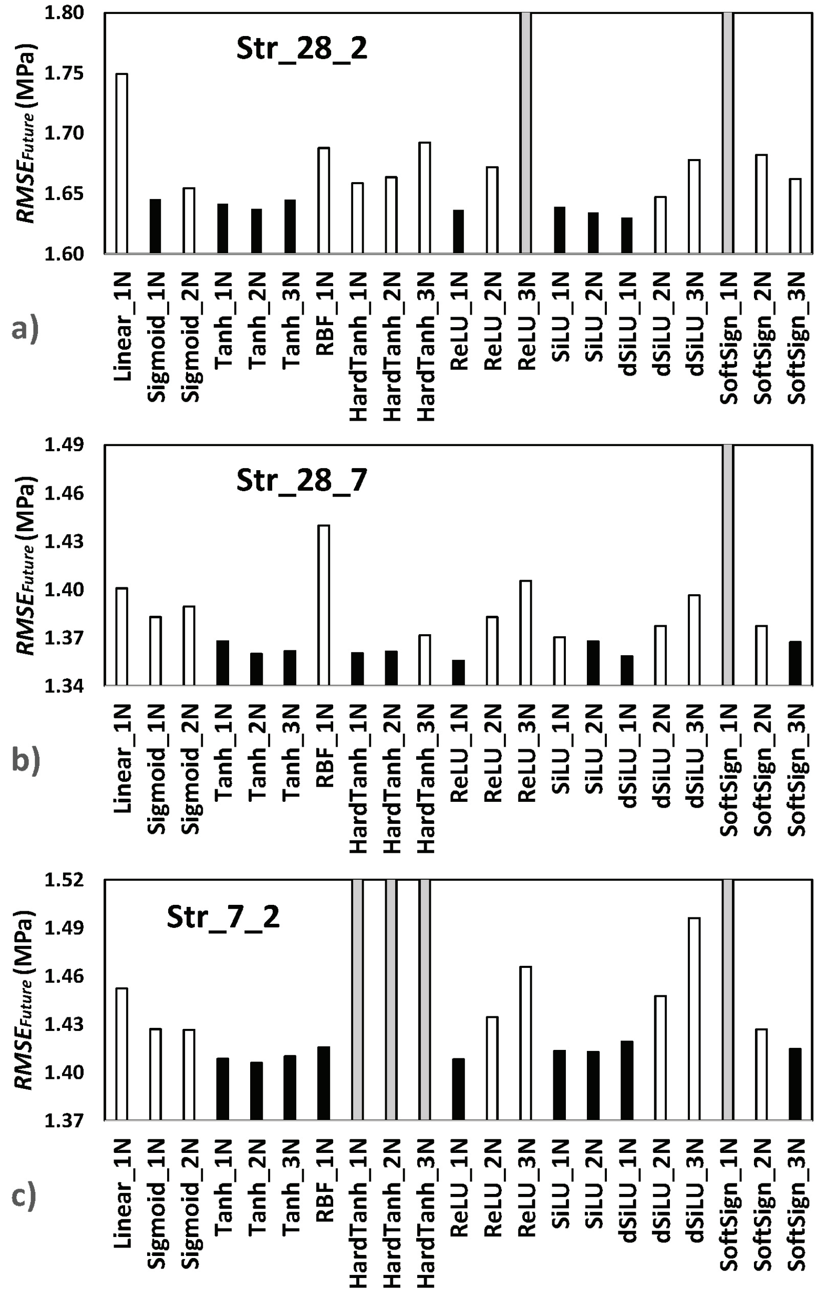

Fig. 11 – *Optimum errors for all the ANNs' architectures and AFs for a) Str_28_2, b) Str_28_7, and c) Str_7_2 models*

where $RMSE_{Future,Min}$ is the minimum mean testing error of all the ANNs. The value of 1.025 is selected for β. Then the following procedure is applied:

(i) In the coordinates system (T_{Tr}, $RMSE_{Future}$) the rectangle defined by the section of the lines $x = T_{Tr,IN}$, $x = T_{Tr,FIN}$, $y = RMSE_{Future,Min}$, $y = \beta \cdot RMSE_{Future,Min}$ is considered.

(ii) The surface of this rectangle S_{Ref} is computed and considered as a reference.

(iii) For each ANN, the surface between the function $RMSE_{Future} = f(T_{Tr})$ and the line y= $\beta \cdot RMSE_{Future,Min}$ is computed, named S_{Opt}. Fig. 12 shows a graphical example.

(iv) The ratio $F_{Opt} = S_{Opt}/S_{Ref}$ expresses the fraction of the reference surface covered with optimal or suboptimal mean test errors, and it is the measure of the ANNs' robustness.

(v) Among all the F_{Opt}, the maximum value $F_{Opt,Max}$ is found. Then the ratios R_{Opt}=$F_{Opt}/F_{Opt,Max}$ are compared.

Fig. 13 demonstrates the ratios R_{Opt}, for all the three models Str_28_2, Str_28_7, and Str_7_2, and all ANNs investigated. Implementing this optimization criterion, the range of optimal ANNs narrows considerably. Neural networks having R_{Opt} around 0.90 or higher are only two or three per model. Fig. 14 shows the corresponding R_{Opt} ratios

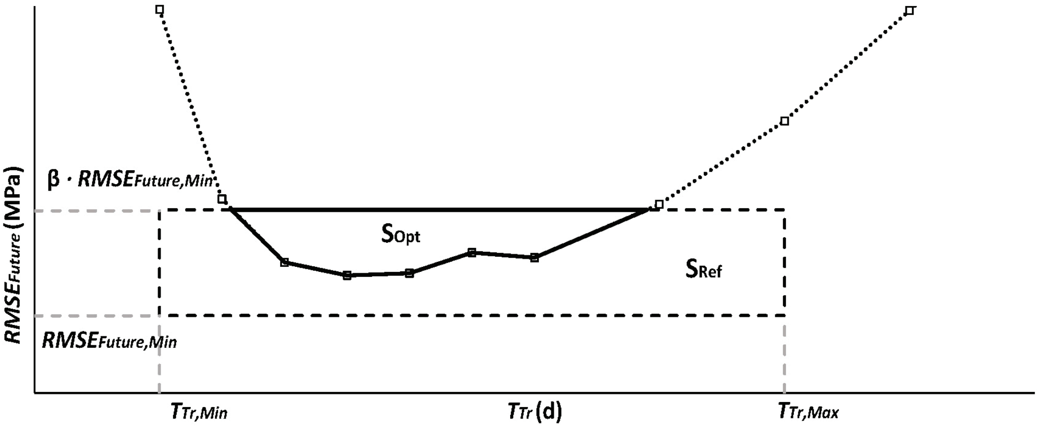

Fig. 12 – *Optimum and reference surfaces of* $RMSE_{Future}$

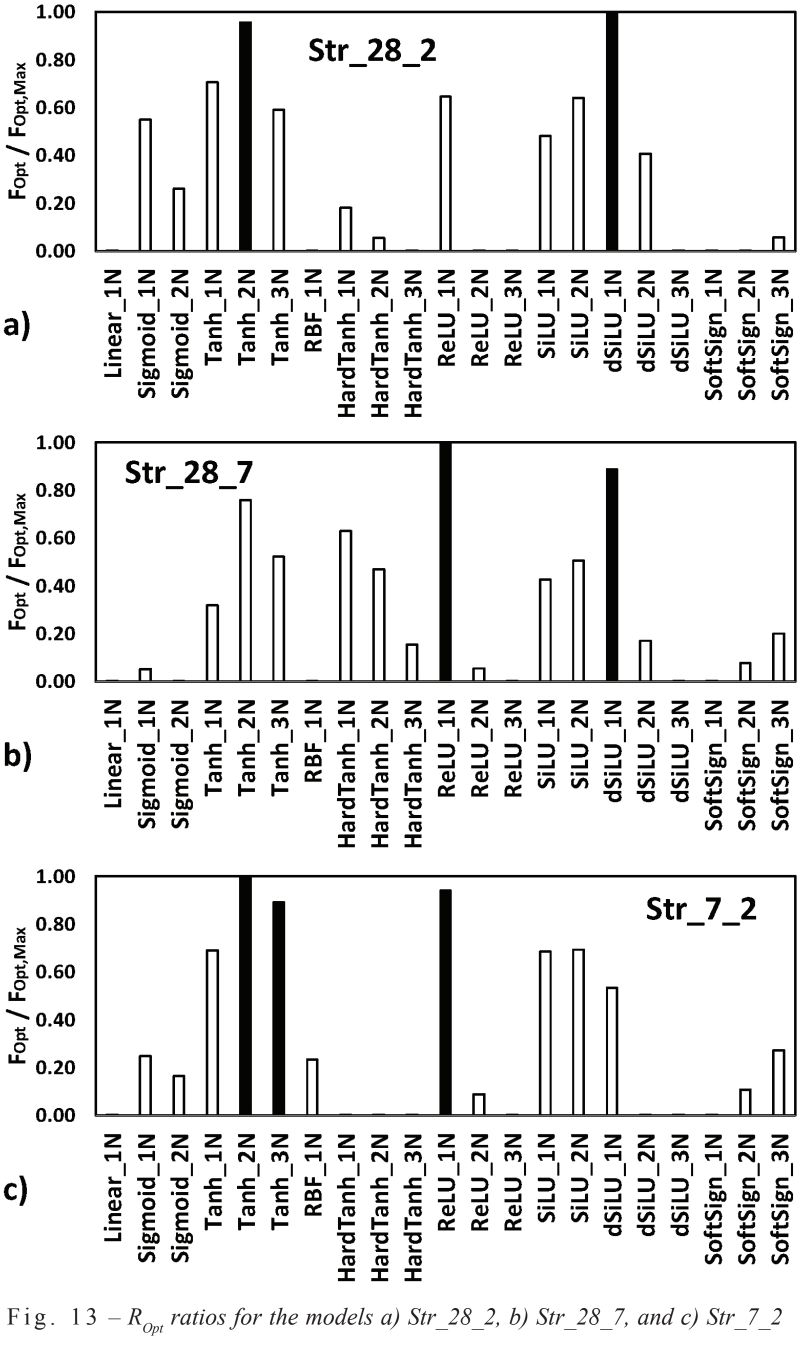

Fig. 13 – R_{Opt} *ratios for the models a) Str_28_2, b) Str_28_7, and c) Str_7_2*

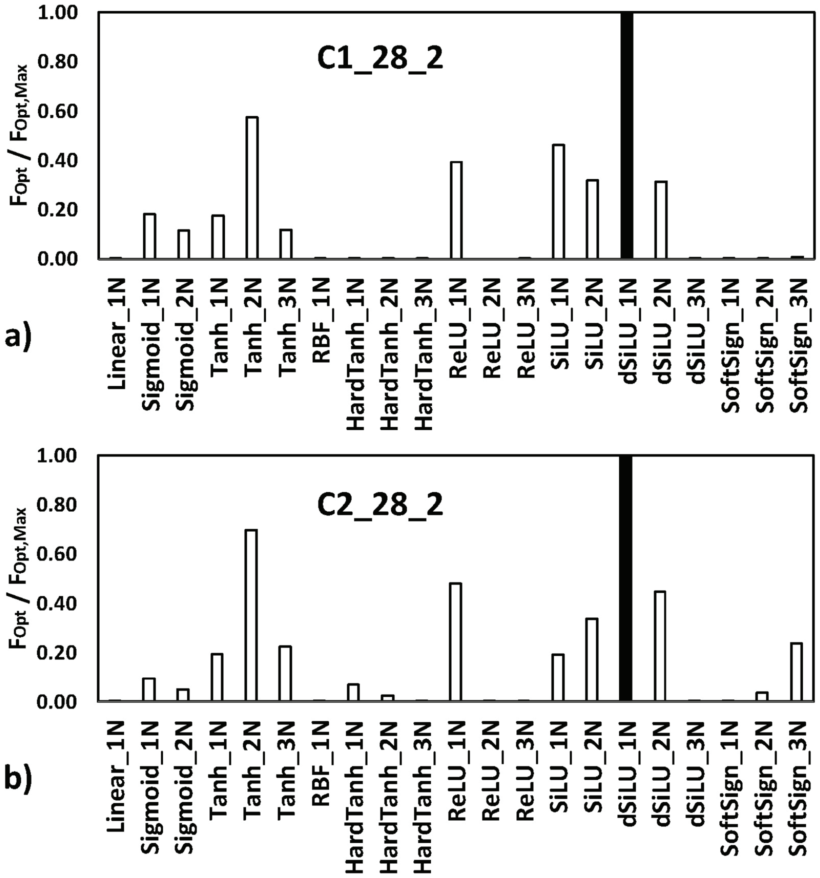

Fig. 14 – R_{Opt} *ratios for the models a) C1_28_2 and b) C2_28_2*

per ANN for the C1_28_2 and C2_28_2 models. The conclusion of comparing the results of this Figure is that only the dSiLU_1N meets both optimality criteria. Besides, these two models provide an improved minimum test error compared to that of the model Str_28_2. The minimal $RMSE_{Future}$ is 1.54 MPa for the former models, whereas it is 1.63 for the latter, resulting in a 5.5 % improvement.

Table 6 presents the ANNs that meet optimization criteria for each of the five models, the best test errors, R_{Opt}s, and model parameters. The following conclusions can be drawn from these results:

(i) The Str_28_7 model is the most accurate among all, reaching an $RMSE_{Future}$ of 1.36 MPa by implementing ReLU_1N and dSiLU_1N, but its usage presents a delay time of eight days, while the Str_28_2 model shall take into account a delay of only three days. The meaning of the above result is that the incorporation of 7-day strength better takes into account parameters non-involved in the models, such as clinker reactivity, but with a loss in terms of early forecasting.

(ii) The optimal training periods of the Str_28_7 model are approximately twice that of the Str_28_2. The comparison of the models Str_7_2, Str_28_2 results in the same trend.

Table 6 – *Optimal ANNs and parameters for each of the models*

ANN	$RMSE_{Future}$	R_{Opt}	$Correl_{Min}$	T_{Tr}	λ
	(MPa)			(d)	
Str_28_2					
Tanh_2N	1.637	0.96	0.6	50	30
dSiLU_1N	1.630	1.00	0.5	40	14
Str_28_7					
ReLU_1N	1.356	1.00	0.6	100	28
dSiLU_1N	1.359	0.89	0.5	100	12
Str_7_2					
Tanh_2N	1.406	1.00	0.5	70	26
Tanh_3N	1.410	0.89	0.6	100	18
ReLU_1N	1.408	0.94	0.6	60	36
C1_28_2 N_{Pr}=12, K_c=1.5					
dSiLU_1N	1.539	1.00	0.6	80	24
C2_28_2 N_{Pr}=4, K_c=0.6					
dSiLU_1N	1.538	1.00	0.6	80	40

(iii) Corrective models C1_28_2, C2_28_2 provide a significant improvement in the test error of Str_28_2, with a simultaneous increase in T_{Tr} and λ parameters, in terms of optimum neural network dSiReLU_1N.

The critical step in finding the optimal network is selecting the type of activation function and the number of nodes. Among all architectures applied, dSiLU_1N, ReLU_1N, and Tanh_2N provide the best results.

Table 7 – *Cumulative distributions of the residual test errors*

Model – ANN	Residual test errors (MPa)					
	Percentile (%)					
	50	60	70	80	90	95
Str_28_2 dSiLU_1N	1.04	1.34	1.65	2.01	2.70	3.30
C1_28_2 dSiLU_1N	1.24	1.42	1.64	1.88	2.34	2.80
C2_28_2 dSiLU_1N	1.21	1.38	1.60	1.87	2.37	2.80
Str_28_7 ReLU_1N	0.82	1.03	1.36	1.67	2.16	2.70
Str_7_2 Tanh_2N	0.94	1.17	1.44	1.74	2.30	2.72

Fig. 15 illustrates a more thorough search of the correlation of test errors between the corrective models and the Str_28_2, for a wide range of training periods, for two different ANNs. The test error for the first two models is consistently smaller than that of the latter and is significantly better for longer training periods. The above proves that these models are more robust than Str_28_2 in terms of generalization ability.

Except for the $RMSE_{Future}$ that constitutes a cumulative criterion of optimization, it is worth further deepening and looking at the distribution of the residual test errors, provided by equation (26), for the optimal ANN of each one of the five models. Table 7 shows cumulative distributions of these errors for percentiles from 50 % to 95 %, from which the following can be concluded:

(i) The median residual error for all models is between 0.82 MPa and 1.24 MPa. This area belongs to the limits of the long-term repeatability of a very competent laboratory. The norm EN 196-1 "Methods of testing cement – Part 1: Determination of strength" provides a value of 2.5 % expressed as coefficient of variation. The expected long-term repeatability for 50 MPa and 60 MPa strength is 1.25 MPa and 1.5 MPa, respectively. The optimal $RMSE_{Future}$– 1.36 MPa for Str_28_7 and 1.63 MPa for Str_28_2– are also near these limits.

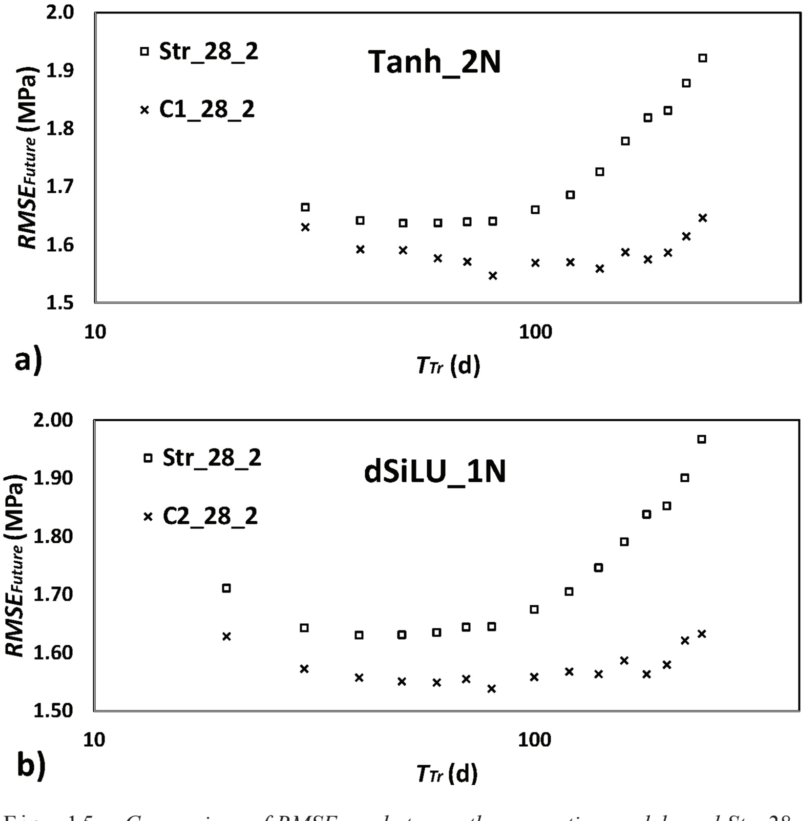

Fig. 15 – *Comparison of $RMSE_{Future}$ between the corrective models and Str_28_2*

(ii) The errors of the models C1_28_2, C2_28_2 are significantly smaller than those of Str_28_2, for percentiles 90 % and higher. The above proves that the contribution of corrective models is to reduce the worst errors of Str_28_2.

(iii) The Str_28_7 model constantly provides the best errors for all percentiles. However, for the 95 % percentile, perhaps its error cannot be considered significantly different from that of corrective models.

(iv) The errors of the Str_7_2 are between those of Str_28_7 and the corrective models for all the percentiles.

The minimal values and the robustness of $RMSE_{Future}$ indicate the C1_28_2, C2_28_2, Str_28_7, and Str_7_2 models could adequately be used for controlling the 28-day or 7-day strength of cement produced in the mills. A manual or automatic control technique can use, as a control variable, the clinker percentage and process variable the prediction from the corresponding model. The need to construct a dynamic function of the effect of the clinker percentage on strength is evident. The process dynamics presented in Fig. 2 would be the core of the control loop in such a case.

Comparison of results and implementation in daily quality control

The main characteristic of this study is the exclusive usage of industrial quality data. A comparison of the results of this research with literature results in the same field is necessary to gain a more comprehensive evaluation, much more if the published studies include industrial data. Table 8 presents the results of the 28-day strength prediction reporting the method implemented and the RMSE of test sets. All RMSE belong to the interval [1.3, 1.9], where the low limit is related to the long-term repeatability of the test method, as explained in the section “Optimal neural networks”. The models developed in this study present RMSEs of testing sets lower than the mean value of the mentioned interval, proving their generalization ability.

Table 8 – *Comparison with literature results*

Reference	Method	RMSE in testing set (MPa)
Accurt *et al.*[12,13]	ANN – Sigmoid	1.70
Accurt *et al.*[13]	FL	1.84
Thamma *et al.*[14]	Gene expression programming	1.50
Verma *et al.*[15]	Symbiotic organism search – SVR	1.90
Verma *et al.*[15]	Particle swarm optimization – SVR	1.85
Verma *et al.*[15]	Relevance vector machine	1.80
Verma *et al.*[15]	Gaussian process regression	1.86
Tsamatsoulis[21]	Multiple linear regression	1.89
Tsamatsoulis[21]	ANN – Sigmoid, Tanh	1.86
This work: Str_28_2	ANN – dSiLU	1.63
This work: Str_28_7	ANN – ReLU	1.36

Finally, the application of the models in daily quality control was the actual metric of their performance: The two basic models, Str_28_2 and Str_28_7, were applied continuously for more than 15 months in the Devnya plant, predicting around 1000 daily 28-day strength measurements. Table 7 shows the models’ settings and the statistical results. Comparison of the actual results of the two models with those shown in Fig. 11 and Table 7 led to the following conclusions:

(1) All the results were in line with the results found during the ANNs’ design, and especially the RMSE of Str_28_7 was lower than predicted.

(2) The operation of the algorithms in actual conditions verified that the design procedure built robust models of sufficient generalization ability.

(3) The optimization of the ANNs parameters in combination with the movable training period guarantee long-term performance.

Conclusions

A series of five models predicting cement strength was developed based on shallow ANNs and exclusively industrial quality data. Three out of the five were independent of each other. The models used physical and chemical data as well as:

(i) 1- and 2-day strength to predict those of 28- and 7-day – models Str_28_2, Str_7_2, respectively;

(ii) additionally, the 7-day strength to predict that of the 28-day – model Str_28_7.

The two last models – C1_28_2, C2_28_2 – combined the results of the previous ones aiming at improving the predictions of the model Str_28_2, taking into account the time delays. Input variables were filtered before datasets insertion into the ANN’s input layer: Only variables whose correlation coefficient with the 28-day strength was higher than the threshold, $Correl_{Min}$, entered the input layer. ANNs contained from one to three nodes within the hidden layer, and were trained dynamically for a period of T_{Tr} days. Prediction of future strength followed for a test period of at least one day. Upon completion of this period, the process was repeated by moving forward the training period. For a given ANN, achieving the best test error needs optimization of the training period. Nine types of activation functions were studied: The traditional ones such as the linear, sigmoid, hyperbolic tangent, and some developed in recent years like the rectified linear unit, sigmoid-weighted linear unit, and its derivative. The algorithm applied L2-regularization to the objective function used to find the ANN's synaptic weights, and optimized the corresponding weight coefficient λ to obtain the minimal test errors. As optimality criteria, the algorithm used the minimization of $RMSE_{Future}$ and its robustness.

The type of activation function is a decisive factor in selecting the optimum ANN for each model. The ANN implementing the derivative of the sigmoid-weighted linear unit and having one node in the hidden layer – dSiLU_1N – showed the smallest and most robust $RMSE_{Future}$, for the models Str_28_2, C1_28_2, C2_28_2. The test errors for the three models were 1.63 MPa, 1.54 MPa, and 1.54 MPa, respectively. Therefore, the two corrective models were equivalent to each other, and improved the error of Str_28_2 by 5.5 %. The optimum ANN of model Str_28_7 contained one node in the hidden layer utilizing the rectified linear unit or the derivative of the sigmoid-weighted linear unit – ReLU_1N and dSiLU_1N, respectively – with an $RMSE_{Future}$ of 1.36 MPa. This error was excellent and 11.7 % lower than those of corrective models, but was achieved at the expense of delay time compared to previous ones. The median test residual errors for the optimal ANNs were between 0.82 MPa and 1.24 MPa, belonging to the range of the long-term repeatability of a very competent laboratory.

In conclusion, the main contribution of the present research is the combination of four factors impacting the generalization ability decisively:

(a) The usage of early strengths as independent variables to uncover hidden variables like clinker activity and milling conditions.

(b) The selection of the optimal activation function among many traditional and modern ones and the determination of the number of nodes in the hidden layer.

(c) The implementation of the L2-regularization technique to avoid overfitting.

(d) The length of time of the moving training period.

The fact that the models were dynamic and applied to industrial data spread over 27 months was a guarantee that they could be implemented effectively to the daily quality control of the same cement plant. Continuous application of the models in the Devnya plant in actual conditions for more than 15 months showed a performance at least equivalent to that calculated during the design step.

Furthermore, the modeling can be a part of a control algorithm, which will use, as a control variable, the clinker percentage and process variable the prediction from the corresponding model. The algorithms developed can be applied to the quality control of any cement plant after training with long-term plant data, as long as they characterize the main properties of cement. The accuracy and reliability of the raw data characterize the quality of the strength prediction. A further improvement of these techniques could include as inputs the chemical characteristics of the clinker, such as C_3S, C_3A, equivalent alkalis, free lime, in case the above are measured routinely by some automatic sampling and measurement system.

Appendix

A summary of the algorithm implementing the model Str_28_2 follows:

(i) All datasets are sorted according to the production date. For each date, more than one set usually exists.

(ii) An initial training date, $T_{In,Tr}$, is selected. The first day initializes the algorithm, and the training set index k is equal to 1.

(iii) The latest final training date, $T_{Fin,Tr}$, is found so that $T_{Fin,Tr} - T_{In,Tr} \leq T_{Tr}$.

(iv) The training set of index k comprises all datasets in the interval $[T_{In,Tr}, T_{Fin,Tr}]$, and its size is N_{Tr}.

(v) The initial date of the test set, $T_{In,Test}$ is found, which is the earliest date satisfying $T_{In,Test} - T_{Fin,Tr} \geq T_D$, where T_D=26 d in the case of Str_28_2 model. T_D derives from the subtraction of T_C–29 from T_C–3.

(vi) A search for the final test date is performed: The production date just after the end of training date, T_{Next}, is considered. The dates after $T_{In,Test}$ are compared with T_{Next}. If they differ less than T_D from T_{Next}, these dates belong to the testing set. Therefore, $T_{Fine,Test}$ is the latest date where $T_{Fin,Test} - T_{Next} < T_D$. In conclusion, the test period is usually a one-day interval, except if the production dates are sparsely distributed.

(vii) All data belonging to the interval $[T_{In,Test}, T_{Fin,Test}]$ constitute the testing set with a size of N_{Test} datasets.

(viii) As to the training set of index k, the ANN is trained by minimizing the next objective function:

$$of_k = s_{res,k}^2 + \lambda \cdot \sum_{I=1}^{M} b_I^2; \tag{20}$$

$$s_{res,k}^2 = \sum_{J=1}^{N_{Tr}} \frac{\left(Str\_28_{Calc,J} - Str\_28_{Act,J}\right)^2}{N_{Tr} - M} \tag{21}$$

Equation (20) incorporates the weight decay term by applying the L2-regularization. Parameter λ

needs optimization to receive the best generalization ability of the ANN under study. For a given λ value, the optimal parameters of formula (20) are obtained using the Levenberg – Marquardt technique. It is worth mentioning that this method has been efficiently used in ANNs' training[34,35]. The parameters b_I are a function of the number of nodes, N_N, in the hidden layer, provided by the following formulae:

$$b_I = W_{mn}; m = 0 \text{ to } N_{IP}; n = 1 \text{ to} N_N; I = 1 \text{ to } N_N \cdot \left(N_{IP}+1\right) \quad (22)$$

$$b_{N_N \cdot (N_{IP}+1)+I} = V_I; I = 1 \text{ to} N_N;\ M = N_N \cdot \left(N_{IP}+1\right) + N_N \quad (23)$$

$$b_{(J-1)\cdot N_{IP}+I} = X0_{IJ};\ b_{J\cdot N_{IP}+I} = \sigma 0_{IJ};\ I = 1 \text{ to} N_{IP};\ J = 1 \text{ to} N_N \quad (24)$$

$$b_{2\cdot N_N \cdot N_{IP}+I} = V_I;\ I = 1 \text{ to} N_N;\ M = 2\cdot N_N \cdot N_{IP} + N_N \quad (25)$$

where equations (22), (23) are used if AFs (10–13) or (15–18) are implemented, and equations (24), (25) are applied in the case of AF (14) application.

(ix) The b_I parameters are used to predict the 28-day strength, $Str\_28_{Calc,J}$, for all the data belonging to the test set. Then, the error of index k is computed as follows:

$$s^2_{Test,k} = \sum_{J=1}^{N_{Test}} \frac{\left(Str\_28_{Calc,J} - Str\_28_{Act,J}\right)^2}{N_{Test}} \quad (26)$$

(x) If the test set contains the last dataset of the entire population, the searching for new training set stops. Otherwise, the procedure is repeated starting from step (iii), setting $T_{Fin,Tr} = T_{Next}$, finding $T_{In,Tr}$ by the formula $T_{Fin,Tr} - T_{In,Tr} \leq T_{Tr}$, and increasing index k by one. The above means that, when the algorithm adds datasets for the new $T_{Fin,Tr}$, it subtracts the datasets between the old $T_{In,Tr}$, and before the new $T_{In,Tr}$, making the training period movable.

(xi) By executing steps (i) to (x), processing of N_{Tot} consecutive training and test sets occurs. Afterwards, the root mean square training and testing errors, $RMSE_{Past}$ and $RMSE_{Future}$, respectively, are computed as follows:

$$RMSE_{Past} = \sqrt{\frac{\sum_{k=1}^{N_{Tot}} s^2_{res,k}}{N_{Tot}}} \quad (27)$$

$$RMSE_{Future} = \sqrt{\frac{\sum_{k=1}^{N_{Tot}} s^2_{Test,k}}{N_{Tot}}} \quad (28)$$

(xii) The impact of multiplier λ, characterizing the weight decay, on the model generalization is assessed by executing the algorithm for a range of λ from 0 to λ_{Max}. The λ's value providing the minimum $RMSE_{Future}$ corresponds to the best generalization for the given training period.

The same algorithm implements the two other models, Str_28_7 and Str_7_2, with the only difference that T_D=21 d for the former and T_D=5 d for the latter. There is no peculiarity in ANN's architecture for each model, except for the fact that there is a comparator before the input layer: For each training set, the correlation factor between each input variable and Str_28 compares with a threshold value, and only if higher, the variable enters the input layer.

The implementation of the C1_28_2 model follows the next steps, in addition to those already described:

(xiii) After the execution of step (xii) for the trinity Str_28_2, Str_28_7, Str_7_1, the first date of the first testing set, T_C, is considered, as determined from the Str_28_2 model.

(xiv) The latest date with results of the Str_28_7 and Str_28_2 is at least five days before this date, and there is the 7-day strength measurement on this date. Starting from this date, T_{pr}, the algorithm considers N_{pr} consecutive past days.

(xv) For all the datasets contained in the time interval [T_{pr}–N_{pr}+1, T_{pr}], the algorithm calculates the differences between the Str_28_7 and Str_28_2 predictions.

(xvi) The mean value of the differences, D_{Aver}, is multiplied with a coefficient k_C, and the product is added to all the results of the Str_28_2 predictions, belonging to the test set of first date T_C. So, for output of Str_28_2 equal to $Str\_28\_2_{Calc}$, the result is the following:

$$C1\_28\_2_{Calc} = Str\_28\_2_{Calc} + k_C \cdot D_{Aver} \quad (29)$$

(xvii) The residual error is computed using equation (26) and the output of C1_28_2 instead of that of the Str_28_2.

(xviii) The test date, T_C, moves to the first one of the next testing set, and the process is repeated starting from step (xiv) until the T_C becomes the last date of the entire population.

(xix) Equation (28) is applied to calculate the test error, $RMSE_{Future}$.

The C2_28_2 model is implemented using the same steps as C1_28_2, with the only distinction being the step's (xv) calculations. This model uses the differences between the 7-day strengths and those provided by the Str_7_2 model. Afterwards, the model's output is as described below:

$$C2\_28\_2_{Calc} = Str\_28\_2_{Calc} + k_C \cdot D_{Aver} \quad (30)$$

Nomenclature

b_I – ANN's weights to be optimized, I=1 to M

$C1\_28\_2_{Calc}$ – Calculated 28-day compressive strength from C1_28_2 model, MPa

$C2\_28\_2_{Calc}$ – Calculated 28-day compressive strength from C2_28_2 model, MPa

C_2S – Dicalcium aluminate, %

C_3A – Tricalcium aluminate, %

C_3S – Tricalcium silicate, %

C_4AF – Tetra-calcium aluminate, %

$Correl_I$ – Correlation factor between X_I and *Str_28*, I=1 to N_I

$Correl_{Min}$ – Minimum correlation factor

D_{Aver} – Mean value of difference

F_{Opt} – Fraction of S_{Ref} covered by S_{Opt}

k – Training set increasing index ranging from 1 to maximum number of sets

k_C – Multiplier used in models C1_28_2, C2_28_2

M – Number of ANN weights to be optimized

N_I – Number of input variables, N_I=9 or N_I=10

N_{IP} – Number of input variables, passing the filter of correlation factors

N_N – Number of nodes in the hidden layer

N_{Pr} – Consecutive past days used in models C1_28_2, C2_28_2, d

N_{Tr} – Size if the training data set

N_{Test} – Size of test data set

N_{Tot} – Total number of consecutive training sets

of_k – Objective function to be minimized for the training set of index *k,* MPa^2

$RMSE$ – Root mean square error, MPa

$RMSE_{Future}$ – Root mean square testing error, MPa

$RMSE_{Past}$ – Root mean square training error, MPa

R_{Opt} – Ratio of F_{Opt} to maximum F_{Opt}

$s^2_{Res,k}$ – Residual error of training set of index k, MPa^2

$s^2_{Test,k}$ – Residual error of testing set of index k, MPa^2

S_{Opt} – Optimum surface of errors, MPa d

S_{Ref} – Reference surface of errors, MPa d

Str_1 – 1-day compressive strength, MPa

Str_2 – 2-day compressive strength, MPa

Str_7 – 7-day compressive strength, MPa

$Str\_7\_2_{Calc}$ – Calculated 7-day compressive strength, MPa

Str_28 – 28-day compressive strength, MPa

Str_28_{Calc} – Calculated 28-day compressive strength, MPa

$Str\_28\_2_{Calc}$ – Calculated 28-day compressive strength from Str_28_2 model, MPa

$Str\_28\_7_{Calc}$ – Calculated 28-day compressive strength from Str_28_7 model, MPa

Str_{Calc} – Calculated compressive strength, MPa

S_{Ref} – Reference surface for ANNs' robustness computations, MPa d

S_{Opt} – Optimal surface for ANNs' robustness computations, MPa d

T_C – Current date or testing date, d

T_{D1}, T_{D2} – Time delay parameters, d

$T_{In,\ Tr}$ – Initial training date, d

$T_{In,\ Test}$ – Initial testing date, d

$T_{Fin,\ Tr}$ – Final training date, d

$T_{Fin,\ Test}$ – Final testing date, d

T_{Next} – Production date just after final training date, d

T_{Tr} – Training period, d

X_I – Model's input variables, I=1 to N_I

$X0_I$ – Centre's parameters of RBF function, I=1 to N_I

XF_I – Normalized variables inserted to input layer, I=1 to N_{IP}

XF – Vector of normalized variables inserted to input layer

XN_I – Normalized input variables, I=1 to N_I

Y – Model's output variable

YN – Normalized output variable

Z_J – Input to the J node of the ANN's hidden layer

V_J – Synaptic weights between hidden and output layers

W_{IJ} – Synaptic weights for equations (6–9), (11–14*)* between input and hidden layers

$W_{Mod,J}$ – Vectors of synaptic weights for each model, J=1 to N_N

Greek Symbols

$\alpha(Z)$ – Activation function of input Z

β – Coefficient of suboptimal region

λ – Multiplier of weight decay term

λ_{Max} – Maximum value of λ

$\sigma(Z)$ – Sigmoid activation function of input Z

σ_I – Variance's parameters of RBF function, I=1 to N_I

Subscripts

Aver	–	Mean value
*D*1, *D*2	–	Delay
I, J, IP, k	–	Indexes
Calc	–	Calculated value
Future	–	Future period
Max	–	Maximum value
Min	–	Minimum value
Mod	–	Model
N	–	Indexes
Next	–	Next date
Opt	–	Optimum value
Pr	–	Past days
Tr	–	Training data set
Test	–	Test data set
Tot	–	Total
Past	–	Past period
Ref	–	Reference
Res	–	Residual

Abbreviations

AF	–	Activation function
ANFIS	–	Adaptive –neuro-fuzzy inference
ANN	–	Artificial neural networks
CEM	–	Cement type
dSiLU	–	Derivative of sigmoid-weighted linear unit function
EN	–	European norm
FL	–	Fuzzy logic
GA	–	Genetic algorithms
HardTanh	–	Hard hyperbolic tangent function
RBF	–	Radial basis function
ReLU	–	Rectified linear unit function
SiLU	–	Sigmoid-weighted linear unit function
SVM	–	Support vector machine
SVR	–	Support vector regression
Tanh	–	Hyperbolic tangent function

ACKNOWLEDGEMENTS

The author is thankful to reviewers for their constructive comments, which have helped to improve the present paper.

References

1. *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. **25** (2011) 242. doi: https://doi.org/10.1061/(ASCE)CP.1943-5487.0000088
2. *Yuan, Z., Wang, L. N., Ji, X.*, Prediction of concrete compressive strength: Research on hybrid models genetic based algorithms and ANFIS, Adv. Eng. Software **67** (2014) 156. doi: https://doi.org/10.1016/j.advengsoft.2013.09.004
3. *Chou, J. S., Tsai, C. F., Pham, A. D., Lud, Y. H.*, Machine learning in concrete strength simulations: Multi-nation data analytics, Constr. Build. Mater. **73** (2014) 771. doi: https://doi.org/10.1016/j.conbuildmat.2014.09.054
4. *Chithra, S., SenthilKumar, S. R. R., Chinnaraju, K., Alfin Ashmita, F.*, A comparative study on the compressive strength prediction models for High Performance Concrete containing nano silica and copper slag using regression analysis and Artificial Neural Networks, Constr. Build. Mater. **114** (2016) 528. doi: https://doi.org/10.1016/j.conbuildmat.2016.03.214
5. *Naderpour, H., Rafiean, A. H., Fakharian, P.*, Compressive strength prediction of environmentally friendly concrete using artificial neural networks, J. Build. Eng. **16** (2018) 213. doi: https://doi.org/10.1016/j.jobe.2018.01.007
6. *Nguyen, T., Kashani, A., Ngo, T., Bordas, S.*, Deep neural network with high-order neuron for the prediction of foamed concrete strength, Comput. -Aided Civ. Infrastruct. Eng. **34** (2019) 316. doi: https://doi.org/10.1111/mice.12422
7. *Hammoudi, A., Moussaceb, K., Belebchouche, C., Dahmoune, F.*, Comparison of artificial neural network (ANN) and response surface methodology (RSM) prediction in compressive strength of recycled concrete aggregates, Constr. Build. Mater. **209** (2019) 425. doi: https://doi.org/10.1016/j.conbuildmat.2019.03.119
8. *Young, B. A., Hall, A., Pilon, L., Gupta, P., Sant, G.*, Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: New insights from statistical analysis and machine learning methods, Cem. Concr. Res. **115** (2019) 379. doi: https://doi.org/10.1016/j.cemconres.2018.09.006
9. *Feng, D. C., Liu, Z. T., Wang, X. D., Chen, Y., Chang, J. Q., Wei, D. F., Jiang, Z. M.*, Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach, Constr. Build. Mater. **230** (2020) 117000. doi: https://doi.org/10.1016/j.conbuildmat.2019.117000
10. *Odler, I.*, Cement strength, Mater. Struct. **24** (1991) 143. doi: https://doi.org/10.1007/BF02472476
11. *Tsamatsoulis, D.*, Prediction of cement strength: Analysis and implementation in process quality control, J. Mech. Behav. Mater. **21** (2012) 81. doi: https://doi.org/10.1515/jmbm-2012-0023
12. *Akkurt, S., Ozdemir, S., Tayfur, G., Akyol, B.*, The use of GA–ANNs in the modelling of compressive strength of cement mortar, Cem. Concr. Res. **33** (2003) 973. doi: https://doi.org/10.1016/S0008-8846(03)00006-1
13. *Akkurt, S., Tayfur, G., Can, S.,* Fuzzy logic model for the prediction of cement compressive strength, Cem. Concr. Res. **34** (2004) 1429. doi: https://doi.org/10.1016/j.cemconres.2004.01.020
14. *Thamma, P., Barai, S. V.,* Prediction of Compressive Strength of Cement Using Gene Expression Programming, in Mehnen, J., Koeppen, M., Saad, A. and Tiwari, A. (Eds.), Applications of Soft Computing. Advances in Intelligent and Soft Computing, Vol. 58, Springer, Berlin, Heidelberg, 2009, pp. 203-212. doi: https://doi.org/10.1007/978-3-540-89619-7_20
15. *Verma, M., Thirumalaiselvi, A., Rajasankar, J.*, Kernel-based models for prediction of cement compressive strength, Neural. Comput. Appl. **28** (2017) 1083. doi: https://doi.org/10.1007/s00521-016-2419-0

16. *Motamedi, S., Shamshirband, S., Hashim, R., Petković, D., Roy, C.*, Estimating unconfined compressive strength of cockle shell–cement–sand mixtures using soft computing methodologies, Eng. Struct. **98** (2015) 49.
doi: https://doi.org/10.1016/j.engstruct.2015.03.070
17. *Chen, H., Qian, C., Liang, C., Kang, W.*, An approach for predicting the compressive strength of cement-based materials exposed to sulfate attack, PLOS ONE **13**(1) e0911370
doi: https://doi.org/10.1371/journal.pone.0191370
18. *Eskandari-Naddaf, H., Kazemi, R.*, ANN prediction of cement mortar compressive strength, influence of cement strength class, Constr. Build. Mater. **138** (2017) 1.
doi: https://doi.org/10.1016/j.conbuildmat.2017.01.132
19. *Pani, A. K., Mohanta, H. K.*, Online monitoring of cement clinker quality using multivariate statistics and Takagi-Sugeno fuzzy-inference technique, Control Eng. Pract. **57** (2016) 1.
doi: https://doi.org/10.1016/j.conengprac.2016.08.011
20. *Tsamatsoulis, D.*, Comparison of dynamic linear and neural network models in predicting cement compressive strength.
URL: https://www.researchgate.net/publication/280125745_Comparison_of_Dynamic_Linear_and_Neural_Network_Models_in_Predicting_Cement_Compressive_Strength (04.03.2021.)
21. *Tsamatsoulis, D.*, Improving the prediction of cement compressive strength by coupling of dynamical models, Chem. Biochem. Eng. Q. **30** (2016) 161.
doi: https://doi.org/10.15255/CABEQ.2015.2298
22. *Haykin, S.,* Neural Networks. A comprehensive Foundation, Pearson Prentice Hall, Delhi, 2005, pp 24.
23. *Subirats, J. L., Franco, L., Molina, I., Jerez, J. M.*, Active learning using a constructive neural network algorithm, in Franco, L., Elizondo and D., Jerez, J. M., (Eds.), Constructive Neural Networks. Studies in Computational Intelligence, Vol. 258, Springer, Berlin, Heidelberg, 2009, pp. 193–206.
doi: https://doi.org/10.1007/978-3-642-04512-7_10
24. *Treadgold, N. K., Gedeon, T. D.*, Simulated annealing and weight decay in adaptive learning: The SARPROP algorithm, IEEE Trans. Neural Netw. **9** (1998) 662.
doi: https://ieeexplore.ieee.org/document/701179
25. *Smith, L. N.*, A disciplined approach to neural network hyper-parameters: Part 1 – Learning rate, batch size, momentum, and weight decay.
URL: https://arxiv.org/abs/1803.09820 (04.03.2021.)
26. *Nakamura, K., Hong, B. W.*, Adaptive weight decay for deep neural networks, IEEE Access **7** (2019) 118857.
doi: https://ieeexplore.ieee.org/document/8811458
27. *Nabian, M. A., Meidani, H.*, Physics-driven regularization of deep neural networks for enhanced engineering design and analysis, J. Comput. Inf. Sci. Eng. **20** (2020) 1.
doi: https://doi.org/10.1115/1.4044507
28. *Basirat, M., Roth, P. M.*, The quest for the golden activation function.
URL: https://arxiv.org/abs/1808.00783 (04.03.2021.)
29. *Nwankpa, C. E., Ijomah, W., Gachagan, A., Marshall, S.*, Activation functions: Comparison of trends in practice and research for deep learning.
URL: https://arxiv.org/abs/1811.03378 (04.03.2021.)
30. *Nair, V., Hinton, G. E.*, Rectified linear units improve restricted Boltzmann machines, in Fürnkranz, J. and Joachims, T. (Eds.), Proceedings of the 27th International Conference on International Conference on Machine Learning, Omnipress, Madison, 2010, pp. 807-814.
31. *Elfwing, S., Uchibe, E., Doya, K.*, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, Neural Netw. **107** (2018) 3.
doi: https://doi.org/10.1016/j.neunet.2017.12.012
32. *Turian, J., Bergstra, J., Bengio, Y.*, Quadratic Features and Deep Architectures for Chunking, in Ringger, E., Haertel, R., Tomanek, K., (eds), Proceedings of the NAACL HLT 2009, Association for Computational Linguistics, Boulder, 2009, pp. 245-248.
33. *Le, P., Zuidema, W.*, Compositional Distributional Semantics with Long Short Term Memory, in Palmer, M., Boleda, G., Rosso, P. (Eds.), Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics, Denver, 2015, pp.10-19.
34. *Badi, I., Boutalline, M., Safi, S.,* The neural networks: Application and optimization application of Levenberg – Marquardt algorithm for Tifinagh character recognition, International Journal of Science, Environment and Technology **2** (2013) 779.
35. *Asklany, S., Mansouri, W., Othmen, S.*, Levenberg-Marquardt deep learning algorithm for sulfur dioxide prediction, IJCSNS **19** (2019) 7.