Computer Assisted Methods in Engineering and Science, **21**: 59–66, 2014.

Soft methods in the prediction and identification analysis of axially compressed R/C columns

Dominika Ziaja, Zenon Waszczyszyn
Department of Structural Mechanics, Rzeszow University of Technology
Powstańców Warszawy 12, 35-959 Rzeszów, Poland
e-mail: dziaja@prz.edu.pl, zewasz@prz.edu.pl

Two problems are presented in the paper concerning axial loading of R/C columns: I) prediction of critical loads, II) identification of concrete strength. The problems were analyzed by two methods: A) Gaussian Processes Method, B) Advanced Back-Propagation Neural Network. The results of the numerical analysis are discussed with respect to numerical efficiency of the applied methods.

Keywords: Gauss Processes Method (GPM), Advanced Back-Propagation Neural Network (ABPNN), Reinforced Concrete (R/C), axial loading, Success Ratio (SR).

1. Introduction

R/C columns are building elements that carry loads from the upper to lower floors down to foundations. Many laboratory tests and theoretical investigations have been devoted to the evaluation of the load bearing capacity of R/C columns, see [1]. Artificial neural networks (ANN) turn out to be an efficient tool in both the direct analysis (prediction of critical loads) and reverse analysis (identification of material and its defects). ANN can also be applied in the analysis of these problems due to a great number of external and internal parameters which have to be taken into account.

The main goal of the paper is to compare the numerical efficiency of two methods: i) Gaussian Processes Method (GPM), and ii) Advanced Back-Propagation Neural Network (ABPNN). The GPM is a Bayesian Method slightly similar to the Radial Basis Function NN. In GPM the kernel basis functions are applied and the covariance matrix of input patterns is the main attribute of the method, see [2]. In the case of ABPNN, the extended error measure and barrier term, weighted by an hyperparameter, are attributes of this neural network.

Two data banks were taken from [1] corresponding to: 1) PEER data bank [3] and 2) K. Chudyba data bank [4]. From these data banks only axially loaded columns with rectangular cross-sections were adopted.

In order to have a high quality of both computational tools, the Levenberg-Marquardt learning method was used for the training of ABPNN and other advanced algorithms from the MATLAB Neural Network Toolbox [5] were applied. The algorithms of NETLAB, published in [6], were adopted for the learning of the GPM models.

2. Applied computational methods

2.1. Gauss Processes Method

GPM is a Bayessian method related to the application of kernel functions, cf. [7, 8]:

$$k(\mathbf{x}^m, \mathbf{x}^n) = k\|\mathbf{x}^m - \mathbf{x}^n\|, \tag{1}$$

where $\mathbf{x}^m$, $\mathbf{x}^n$ – points m, n in the input space. Function (1) is used for constructing the covariant matrix $\mathbf{C}_N$ with components:

$$c_{mn} = k(\mathbf{x}^m, \mathbf{x}^n) + \sigma_\nu^2 \cdot \delta_{mn}, \tag{2}$$

where σ_ν^2 regularization parameter corresponding to the variance of the target data distribution.

On the base of data set $\mathfrak{D}_N$ composed of N patterns we wish to predict a new pattern point of the target value t^{N+1} for the known input x^{N+1}. The conditional distribution $p\left(t^{N+1}|\mathbf{t}\right)$, where $\mathbf{t} = \{t^n\}_{n=1}^N$, can be found as Gaussian distribution:

$$p\left(t^{N+1}|x^{N+1}, \mathfrak{D}_N\right) = N\left(t^{N+1}|m_{N+1}, \sigma_{N+1}^2\right), \tag{3}$$

where the mean and variance are computed by the following formulas:

$$\mathbf{m}_{N+1}\left(\mathbf{x}^{N+1}\right) = \mathbf{k}^\mathrm{T}\mathbf{C}_N^{-1}\mathbf{t} \quad \sigma_{N+1}^2 = c - \mathbf{k}^\mathrm{T}\mathbf{C}_N^{-1}\mathbf{t}, \tag{4}$$

where vector $\mathbf{k}^\mathrm{T} = \{k^n\}_{n=1}^N$ and scalar $c = k\left(\mathbf{x}^{N+1}, \mathbf{x}^{N+1}\right) + 1/\sigma_\nu^2$ are used, cf. [7].

Computations in our paper are based on the estimation of either the Squared Exponential covariance function(SE) or the Rational Quadratic function RQ, see [8]:

$$\mathrm{SE}: c\left(\mathbf{x}^m, \mathbf{x}^n\right) = \nu_0 \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{D} a_i\left(x_i^m - x_i^n\right)^2\right) + b, \tag{4SE}$$

$$\mathrm{RQ}: c\left(\mathbf{x}^m, \mathbf{x}^n\right) = \nu_0 \cdot \left(1 + \sum_{i=1}^{D} a_i\left(x_i^m - x_i^n\right)^2\right)^{-\nu} + b, \tag{4RQ}$$

where the model parameters $\left\{\nu_0,\ b,\ a_1,\ ...,\ a_D,\ \sigma_N^2,\ \nu\right\}$ are to be positive. They can be evaluated by procedures given in [8].

In our paper scalar target t^n corresponds to either predicted critical loads P_{cr} or to identified concrete strength f_c.

2.2. Advanced Back Propagation Neural Network

Two ABPNNs of architecture $I - H - 1$ were adopted, where I – number of inputs, H – number of neurons in a single hidden layer with the bipolar sigmoid activation function. Single outputs with identity activation function correspond to either the predicted critical load P_{cr} or to identified concrete strength f_c.

Advanced neural networks ABPNNs are related to the standard BP network MLP (Multilayer Perceptron), see [6], but with an extended network error:

$$E(\mathbf{w}) = \frac{\gamma}{2}\frac{1}{P}\sum_{p=1}^{P}\left(y^p - t^p\right)^2 + \frac{1-\gamma}{2}\sum_{i=1}^{W} w_i^2, \tag{5}$$

where y^p – computed output for the p-th pattern, γ – regularization parameter, automatically optimized by procedures listed in [5], w_i – weights of synaptic connections and values of neural biases.

The Levenberg-Marquardt learning method was adopted, as a numerically efficient method, cf. [5].

3. Data adopted for the numerical analysis

Pattern pairs $(\mathbf{x}^p, \mathbf{t}^p)$ were selected from two data banks: 1) PEER [7] $P_1 = 65$ patterns was completed, 2) Chudyba's data bank containing $P_2 = 27$ patterns. From among these patterns $P_3 = 3$ patterns were eliminated since they gave responses non-consisted with the other patterns. Thus, the primary set has $PP = P_1 + P_2 - P_3 = 89$ patterns.

Following [1] the input vector $\mathbf{x}^p$ and scalar output y correspond to the following problems:

1. Prediction of critical load P_{cr}:

$$\mathbf{x}_{(6\times1)} = \{\mathbf{Ca}, f_c\}, \qquad y = P_{cr}; \tag{6}$$

2. Identification of concrete strength:

$$\mathbf{x}_{(6\times1)} = \{\mathbf{Ca}, P_{cr}\}, \qquad y = f_c. \tag{7}$$

The common parameter vector is:

$$\mathbf{Ca} = \{B, H, L, \rho, f_y\}, \tag{8}$$

where B, H – width and height of rectangular cross-section and L – length of column, ρ – reinforcement percentage and f_y – steel yielding stress.

4. Numerical analysis

4.1. Application of ABPNN

In the paper a relative network error was introduced:

$$\mathrm{Re} = \frac{1}{S} \cdot \sum_{p=1}^{S} \left| \left(\frac{y^p}{t^p} - 1 \right) 100\% \right|, \tag{9}$$

where S – number of patterns for subsets completed of selected patterns p.

A random split of primary pattern PP into the learning and training sets composed of $L = 0.7PP \cong 62$ patterns and $T = 0.3PP \cong 27$ was made. After the cross validation the number of hidden neurons was found separately for problems (6) and (7). In the case of prediction of force the number of hidden neurons was $H_{\mathrm{I}} = 6$ and for identification of concrete strength $H_{\mathrm{II}} = 10$. Corresponding to six inputs and one output, the neural networks had $W^{\mathrm{I}} = 49$ and $W^{\mathrm{II}} = 87$ parameters. This number of hidden layers give the total number of generalized weights (connection weights plus biases) equal $W^{\mathrm{I}} = 49$ and $W^{\mathrm{II}} = 87$. Such a number of the network parameters turned out to be too height in comparison with the number of learning patterns $L = 62$. That is why artificial noisy patterns were added to the learning and testing patterns, following the approach developed at the Rzeszow University of Technology, see e.g. [7].

In the presented paper it was assumed that geometrical input data B, H, and L were measured with the accuracy ±1.0 mm. In such a way the perturbations of noisy data were randomly selected from two ranges, $[-1.0, 0.0]$ and $[0.0, 1.0]$. Together with the primary patterns the noisy set was composed of $PN = (2 + 1) \times 3 \times 3 \times 89 = 2403$ patterns. In this way the ranges of the input components were extended, as shown in Table 1.

In Table 2 there are shown results for three different sets, after random selection of learning and testing patterns; i.e, i) $PP = 89$, $L = 69$, $T = 27$ patterns, ii) $PN1 = 2403$, $L \approx 0.7\ PN1 = 1682$, $T \approx 0.3\ PN1 = 721$ patterns, iii) $PN2 = 2403$, $L = T \approx 0.5\ PN1 = 1201$.

The patterns listed above were applied for the learning and testing of the networks, related to two analyzed problems. Because of the random selection of primary values and weights, the computations by means Levenberg-Marquardt method were repeated twenty times, separately for each analyzed problems. In Table 2 average results from twenty learning processes are given.

Table 1. Ranges of input values.

	Initial range of values	Range of values after elimination of three patterns
Width of cross-section B [mm]	80–350	80–305
Height of cross-section H [mm]	80–350	80–350
Height of column L [mm]	80–2134	80–1800
Percentage of reinforcement ρ [%]	1.01–4.69	1.27–4.69
Concrete compressive strength f_c [N/mm^2]	21.0–99.5	21.0–69.6
Yield strength of steel f_y [N/mm^2]	336.0–587.1	336.0–587.1
Critical force P_{cr} [kN]	95–2176	95–1090

Table 2. Relative errors Re [%] for input sets for prediction of critical loads P_{cr} and identification of concrete strength f_c.

Input sets	Number of patterns	Prediction of P_{cr}			Identification of f_c		
		Soft models	Re^{L} [%]	Re^{T} [%]	Soft models	Re^{L} [%]	Re^{T} [%]
PP	$PP = 89$ $L = 62$ (70%) $T = 27$ (30%)	ABPNN: 6-6-1	26.56	45.33	ABPNN: 6-10-1	6.61	7.60
		GPM-RQ	9.44	39.02	GPM-RG	6.98	10.18
PN1	$PP = 2403$ $L = 1682$ (70%) $T = 721$ (30%)	ABPNN: 6-10-1	13.06	13.82	ABPNN: 6-10-1	4.53	4.66
		GPM-RG	3.29	3.59	GPM-RG	1.42	1.93
PN2	$PP = 2403$ $L = 1201$ (50%) $T = 1202$ (50%)	ABPNN: 6-10-1	13.93	12.62	ABPNN: 6-10-1	4.01	4.97
		GP-SE	4.15	3.65	GP-SE	2.49	3.24

4.2. Application of GPM

The numerical analysis carried out for a variety of civil engineering problems, see e.g. [8], has proved a great numerical efficiency of GPM. In comparison with other Bayessian methods, the application of GPM is computationally cheap (a comparatively low number of mathematical operations). The other feature of GPM is that the evaluation of approximation errors is similar to that reached by more refined Bayessian approaches, cf. [8].

The application of GPM to the primary pattern set PP gave errors similar, or even smaller than in the case of ABPN application, see Table 2. The added noisy patterns improved the accuracy of GPM much more considerably than when ABPN was used.

4.3. Comparison of application of GPM and ABPNN on the pattern set PN2

In order to compare the numerical efficiency of GPM vs. ABPNN, attention is focused on the application of a noisy set PN2, in which the number of learning patterns $L = 0.5\ PN2$ was applied, although the errors of learning and testing are slightly worse than for the set PN1 completing $L = 0.7\ PN1$ patterns both for the learning and testing processes, see Table 2.

The comparison is based on distribution of patterns within the error bounds, see Figs. 1–3. The definition of the error bounds ReB corresponds to the relation to relative errors for the percentage of total numbers of points $(t^p,\ y^p)$ placed in-between straight lines for which the inequalities are $|Rep| \leq ReB\%$ satisfied, where:

$$Rep = \left(\frac{y^p}{t^p} - 1\right) 100\%. \tag{10}$$

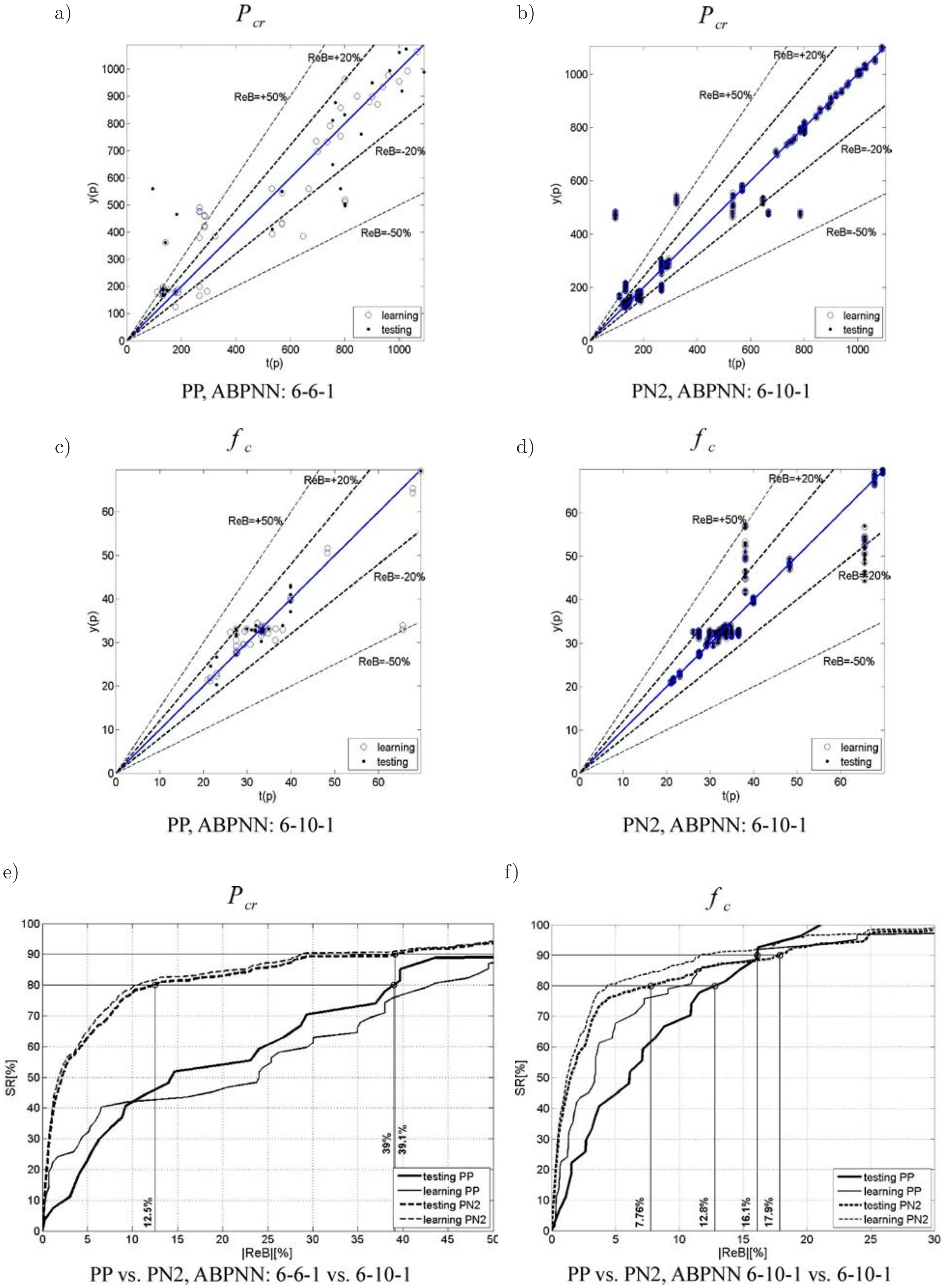

Fig. 1. a, b, c, d) Bound errors for application of ABPNN for prediction P_{cr} and identifying f_c input pattern sets computed for primary and noisy patterns PP and PN2; e, f) Comparison of Success Ratio Curves for data sets PP vs. PN2.

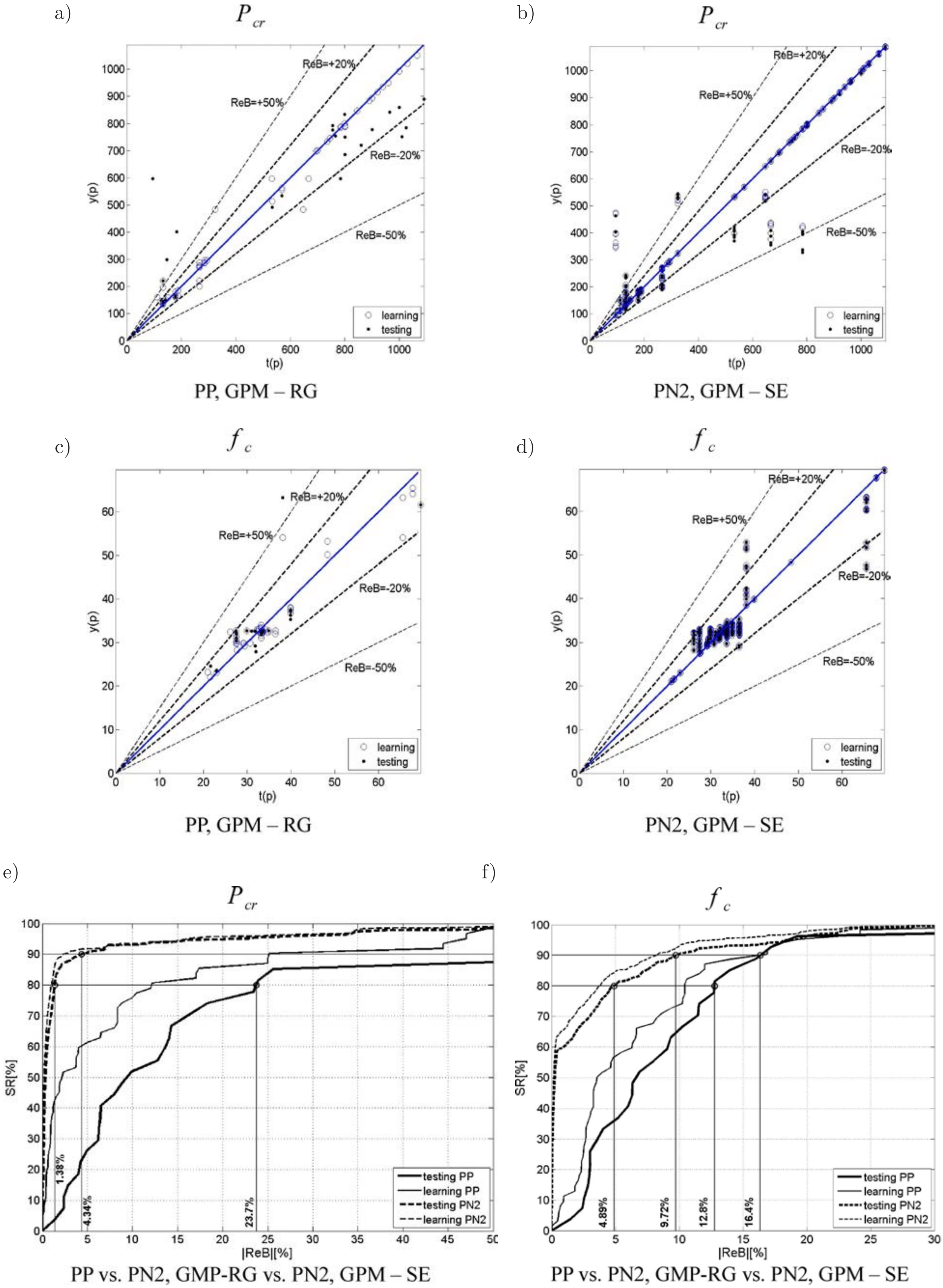

Fig. 2. a, b, c, d) Bound errors for application of GPM for prediction P_{cr} and identifying f_c input pattern sets computed for primary and noisy patterns PP and PN2; e, f) Comparison of Success Ratio Curves for data sets PP vs. PN2.

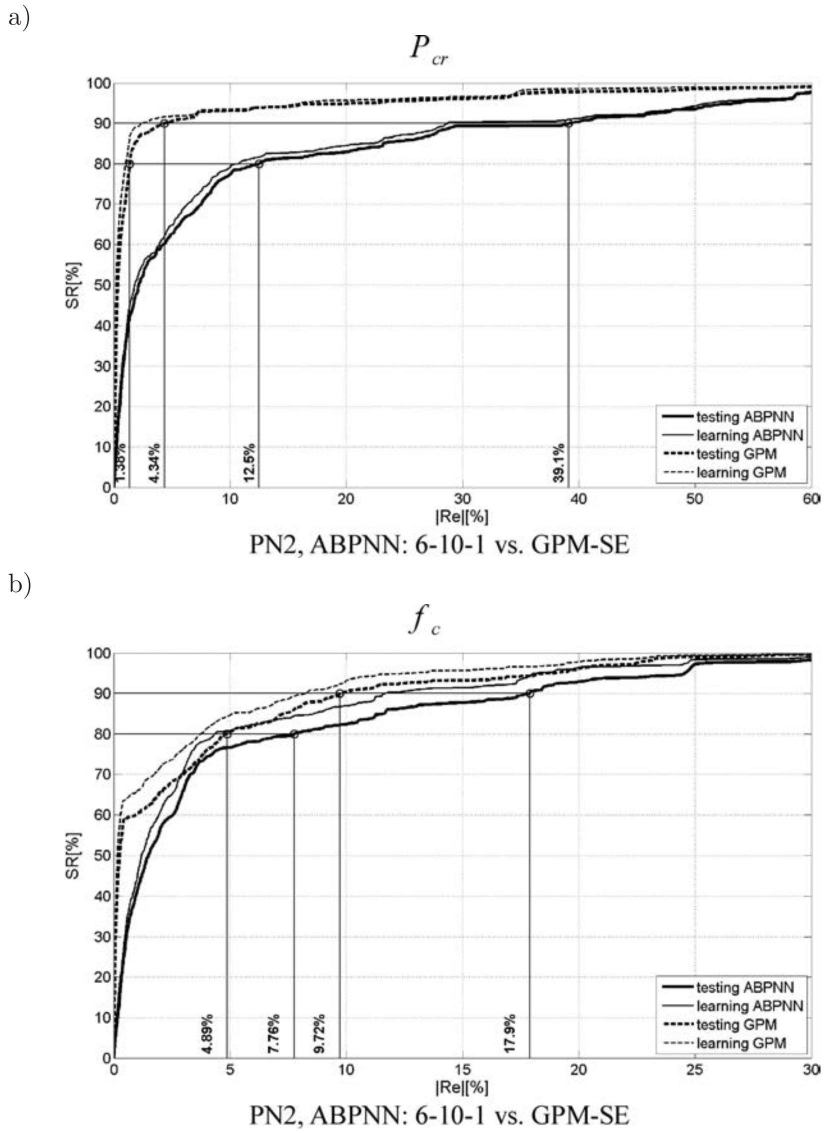

Fig. 3. Comparison of Success Ratios for data set PN2 applying ABPNN and GPM for prediction P_{cr} and identification f_c.

Having defined ReB, the cumulative error called as the Success Ratio $SC(ReB)$ is defined as:

$$SC = (SRe/S)\,100\%, \tag{11}$$

where SRe – number of points $(t^p,\ y^p)$ within the bounds $\pm ReB$, S – total number of points corresponding to the learning and testing data sets, completed of either L or T pattern points, respectively. In Fig. 1a,b, the distribution of points is shown separately for the problems of simulation of the axial load P_{cr}, and identification of concrete strength f_c. These figures also illustrate the bounds $ReB = \pm 20,\ \pm 50\%$.

Below, in Fig. 1 e,f the cumulative curves for both problems of prediction and identification by ABPNN are presented. Looking at the $SC(ReB)$ relations corresponding to the testing data, it is clearly visible that the buckling values of the force P_{cr} can be predicted with a higher accuracy, i.e. $RepB\%$ errors than identification of concrete strength f_c. For instance if the Success Ratio for the input set PP is assumed to be $SC = 80\%$ than the related testing errors equal $ReB_{Pcr}(\text{PP}) = 39\%$, $ReB_{fc}(\text{PP}) = 12.8\%$. For $SC = 90\%$, the following values of errors were computed: $ReB_{Pcr}(\text{PP}) > 50\%$, $ReB_{fc}(\text{PP}) = 16.1\%$.

The interpretation of the above figures can be expressed as follows. If we adopt the Success Ratio $SC = 80\%$ and apply the ABPNN and testing set composed of $0.3PP = 27$ patterns for predicting the critical load P_{cr} then 80% of testing patterns points are placed within bounds $|ReB| = 39\%$.

In the case of the noisy set of patterns PN2 the assumption of $SC = 90\%$ gives $ReB_{Pcr}(\text{PN2}) = 12.5\%$, and $ReB_{Pcr}(\text{PN2}) = 7.76\%$.

After the application of GPM, the errors listed above are lower than those errors obtained for the use of ABPNN. For instance, adopting the noisy set of data PN2 the application of GPM leads to the error bounds: $ReB_{Pcr}(\text{PN2}) = 4.34\%$, and $ReB_{fc}(\text{PN2}) = 9.72\%$, see Fig. 2. The corresponding results for ABPNN are $ReB_{Pcr}(\text{PN2}) = 39.1\%$ and $ReB_{fc}(\text{PN2}) = 17.9\%$.

In Fig. 3 the Success Ratio Curves $SC(Re)$ are shown for the data set PN2. They graphically illustrate a comparison of GPM versus ABPNN. In case of load P_{cr} prediction, the superiority of the application of GPM over the ABPNN is clearly visible.

For $SC = 90\%$ the boundary error $ReB_{Pcr}(\text{PN2}) = 4.34\%$ was obtained for the GPM versus $ReB_{Pcr}(\text{PN2}) = 39.1\%$ after the use of ABPNN. Quite similar results were obtained for the f_c identification. Assuming $SC = 90\%$ the bound errors were computed $ReB_{fc}(\text{PN2}) = 9.72\%$, versus $ReB_{fc}(\text{PN2}) = 17.9\%$, if either GPM or ABPNN were respectively applied.

5. Final remarks and conclusions

1. The main problem of a successful numerical analysis are statistically numerous and consistent data. The adopted data do not fully satisfy the mentioned requirements. This concerns especially Chudyba's data [4] since they were completed in a selected area of input data. That is why this set cannot be explored for good testing of results obtained only by means of PEER data bank.

2. Introduction of noisy data turned out to be a good remedy for the drawback caused by statistically non-representative primary data. A great effort, devoted to the elimination of bad influences of random selection of initial weight on output values, was partially successful by a repeated random selection of networks weights and adoption of average values of the obtained outputs. The selection of appropriate learning and testing pattern sets seems much more complicated. That it why in this field the noisy data gave very satisfactory results.

3. An initial trend of smaller errors identification of concrete strength f_c than those errors in the prediction of load P_{cr}, was partially improved also by introduction of noisy data.

4. The conclusion formulated above is supported by the estimation of average errors listed in Table 1. They clearly point out the superiority of GPM over ABPNN. This is also proved by the Success Ratio Curves drawn in Fig. 3.

References

[1] M. Jakubek. Neural network prediction of load capacity for eccentrically loaded reinforced concrete columns. *Computer Assisted Methods in Engineering and Sciences*, **19**(4): 339–349, 2012.

[2] Ch.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] PEER (Pacific Earthquake Engineering Research Center). University of California, 2004, database.

[4] K. Chudyba. *Influence of material and geometrical parameters on load carrying capacity of axially loaded R/C columns* [in Polish]. PhD, Cracow UT, 1999, database.

[5] H. Demuth, M. Beale. *Neural Network Toolbox for use with MATLAB*. User's Guide version 3. Math Works Inc. Natick MA, 2004.

[6] J.P. Nabney. *Netlab – Algorithms for Pattern Recognition*. Springer, 2004.

[7] B. Miller. *Neurocomputing in selected problems of structural identification and damage detection*. Oficyna Wydawnicza Politechniki Rzeszowskiej, 2013.

[8] Z. Waszczyszyn, M. Słoński. Selected problems of artificial neural networks development. In: Z. Waszczyszyn, Ed., *Advances of Soft Computing in Engineering*, CISM Courses and Lectures, Vol. 512, Springer 2010.