NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment

Ronay Ak, Yan-Fu Li, Valeria Vitelli, Enrico Zio

HAL Id: hal-00734441

https://centralesupelec.hal.science/hal-00734441

Submitted on 18 Dec 2012

NSGA-II-Trained Neural Network Approach to the Estimation of Prediction Intervals of Scale Deposition Rate in Oil & Gas Equipment

Ronay Ak[a], Yanfu Li [a], Valeria Vitelli[a], Enrico Zio[a,b,*], Enrique López Droguett[c], Carlos Magno Couto Jacinto[d]

[a]*Chair on Systems Science and the Energetic Challange, European Foundation for New Energy-Electricité de France*

École Centrale Paris, Grande Voie des Vignes, Châtenay-Malabry, 92290 France SUPELEC, Plateau du Moulon - 3 Rue Joliot-Curie, Gif-Sur-Yvette, 91192 France

[b]*Department of Energy, Politecnico di Milano, Via Ponzio 34/3*
Milan, 20133 Italy

[c]*Center for Risk Analysis and Environmental Modeling*
Federal University of Pernambuco, Recife, Brazil

[d]*Petrobras Research Center, CENPES, Petrobras,*
Rio de Janeiro, Brazil

Abstract

Scale deposition can damage equipment in the oil & gas production industry. Hence, the reliable and accurate prediction of the scale deposition rate is critical for production availability. In this study, we consider the problem of predicting the scale deposition rate, providing an indication of the associated prediction uncertainty. We tackle the problem using an empirical modeling approach, based on experimental data. Specifically, we implement a multi-objective genetic algorithm (namely, non-dominated sorting genetic algorithm–II (NSGA-II)) to train a neural network (NN) (i.e. to find its parameters, that is its weights and biases) to provide the prediction intervals (PIs) of the scale deposition rate. The PIs are optimized both in terms of accuracy (coverage probability) and dimension (width). We perform *k*-fold cross-validation to guide the choice of the NN structure (i.e. the number of hidden neurons). We use hypervolume indicator metric to evaluate the Pareto fronts in the validation step. A case study is considered, with regards to a set of experimental observations: the NSGA-II-trained neural network is shown capable of providing PIs with both high coverage and small width.

***Key words*:** Prediction intervals, Neural networks, Multi-objective genetic algorithms, Cross-validation, Hypervolume, Scale deposition rate.

1. Introduction

Degradation to failure of components may cause unplanned costs and production losses through downtime (Lins et. al., 2011). Then, prediction of component degradation to failure is important for production availability. In the context of oil & gas industry, scale deposition and corrosion continue to be serious and costly problems, because affecting the operation of the components (Moura et. al., 2011). Formation of scale on downhole equipment due to produced mineral salts is a common occurrence which is mitigated with chemical treatments or surface modifications. However, when chemical treatments are ineffective or undesirable, the scale buildup

---

[*] Corresponding author. Tel.: +33 (0)1 4113 1606; fax: +33 (0)1 4113 1272.
 E-mail address: enrico.zio@ecp.fr (E. Zio).

should be accounted for and predicted a priori for a given application. Accurate prediction of scale deposition and corrosion can give way to preventive maintenance. In this paper, we focus on the former process.

In oil & gas plant components, scale deposition is influenced by different factors, such as reservoir conditions (temperature, pressure), changes in pH, variation of flow rates, water cut, material structure, etc. (Nyborg, 2002). A predictive model is needed to explain the linear or nonlinear mapping between these input (explanatory) variables and the output (the scale deposition rate, hereafter called scale rate). In the literature some analytical models based on thermodynamics (Yuan, Todd, & Heriot-Waft 1991), kinetics (Larsen et. al., 2008) and hydrodynamics, or a combination of these (Stamatakis, Stubos, & Muller, 2011) have been proposed for the prediction of scale deposition in components of production wells. The output of these models is typically deterministic, with no consideration given to the variability/uncertainty caused in the output by the uncertain characteristics of the input variables. To account for this, statistical prediction methods based on learning algorithms (neural networks, NNs, support vector machines, SVMs, etc.) have been proposed (Lins et. al., 2011; Moura et. al., 2011; Cottis, Owen, & Turega, 2000).

Due to their capability of learning complex nonlinear relationships among variables from observed data, learning algorithms (e.g. NNs, SVMs, nonlinear regression models, etc.) have been successfully used in many fields of science and engineering. Lins et al. (2011) and Moura et al. (2011) proposed a SVM approach combined with particle swarm optimization (PSO) for reliability prediction in the context of oil production industry. The former work aimed at predicting scale deposition over time; the latter work aimed at predicting time between failures (TBFs) with simultaneous input variable selection and SVM parameters' tuning by PSO. Cottis, Owen, and Turega (2000) used a conventional multi-layer perceptron NN for the prediction of the corrosion rate of steel in seawater.

In practice, the predictions provided by a learning algorithm like NNs or SVMs are affected by uncertainties (Khosravi et al., 2011a; Khosravi et al., 2011b; Khosravi, Nahavandi, & Creighton, 2010). For this reason, it is important to provide prediction intervals (PIs) of the output. A prediction interval (PI) is an interval estimate for an (unknown) future value of the target. PIs are comprised of lower and upper bounds within which the actual target is expected to lie with a predetermined probability (Khosravi et al., 2011a; Khosravi et al., 2011b; Khosravi, Nahavandi, & Creighton, 2010). There are two conflicting criteria for assessing the quality of the estimated PIs: coverage probability (CP) and prediction interval width (PIW) (Moura et. al., 2011). The prediction interval coverage probability (PICP) represents the probability that the set of estimated PI values will contain a certain percentage of the true output values. The prediction interval width (PIW) simply measures the extension of the interval as the difference of the estimated upper bound and lower bound values. To obtain representative PIs, one should aim at maximizing the CP and minimizing the PIW, simultaneously.

In this paper, we propose the adoption of a multi-objective optimization approach to the construction of PIs for NN predictions of scale rate in oil & gas components. A multi-objective genetic algorithm (namely, non-dominated sorting genetic algorithm–II (NSGA-II)) (Sirinivas & Deb, 1994) is used to train a NN, i.e. optimize its parameters (weights and biases) with respect to accuracy (max) and width (min). A demonstration of the approach and a comparison with the Lower and Upper Bound Estimation (LUBE) method of (Khosravi et al., 2011b) on a synthetic case study of literature is given in (Ak, Li, & Zio, 2012) and testing of the method on a real case study of wind speed prediction is given in (Ak et al., 2012).

Genetic Algorithms (GAs) have been successfully applied in a number of applications of engineering and related fields (Coello, Lamont, & Van Veldhuizen, 2007; Chatterjee & Bandopadhyay, 2012). The major motivation for using the GA search paradigm is due to the following three recognized advantages (Gosselin, Tye-Gingras, & Mathieu-Potvin, 2009): (i) ease of use; (ii) robustness and (iii) capability of exploring large portions of the search space without falling into a local optimum. Further, GAs are capable of searching solutions from disjoint feasible domains and of operating on irregular functions (i.e. non-continuous and even non-differentiable); for proceeding in the search, GAs do not require the computation of gradients (Ozkol & Komurgoz, 2005).

In order to choose the NN structure (number of hidden neurons) with good generalization performance, a k-fold cross-validation is performed. A hypervolume indicator metric is used to compare the Pareto fronts of each cross-validation fold.

The paper is organized as follows: Section 2 briefly introduces the basic concepts of NN and PIs, and the use of NSGA-II for training a NN to estimate PIs. The complete methodology set up for scale rate PIs estimation is illustrated in Section 3. Experimental results on the real case study of scale rate prediction are given in Section 4. Finally, Section 5 concludes the paper with a critical analysis of the results obtained and some ideas of future studies.

2. Modeling Framework

In this Section we describe NN-based PIs estimation in the theoretical framework of multi-objective optimization, and we give the details of our implementation of NSGA-II for tackling the problem at hand.

2.1. PIs

We consider the following mathematical problem of nonlinear regression (Zio, 2006; Yang et al., 2002):

$$y = f(x; w^*) + \varepsilon(x), \quad \varepsilon(x) \sim N(0, \sigma_\varepsilon^2(x)) \tag{1}$$

where x, y are the input and output vectors of the regression, respectively, and w^* represents the vector of values of the parameters of the model function f, in general nonlinear. The term $\varepsilon(x)$ is a random error with zero mean and variance $\sigma_\varepsilon^2(x) > 0$. For simplicity of illustration, in the following we assume y mono-dimensional. An estimate $\hat{w}$ of w^* is sought by minimizing the quadratic error function on a training set of input/output values $D = \{(x_n, y_n), n = 1, 2, \ldots, n_p\}$,

$$E(w) = \sum_{i=1}^{n_p} (\hat{y}_i - y_i)^2 \tag{2}$$

where $\hat{y}_i = f(x_i; \hat{w})$ represents the output provided by the model in correspondence of the input x_i, and n_p is the total number of training samples.

We want to quantify the uncertainty associated to the model output estimates, in terms of PIs. A PI is comprised of upper and lower bounds in which a future unknown value of the target is expected to lie with a predetermined confidence level $(1-\propto)$. The formal definition of a PI is thus (Geisser, 1993):

$$Pr(L(x) < y(x) < U(x)) = 1 - \alpha \tag{3}$$

where $L(x)$ and $U(x)$ are respectively the lower and upper bounds of the PI of the output $y(x)$ corresponding to input x; the confidence level $(1-\propto)$ refers to the expected probability that the true value of $y(x)$ lies within the PI, $(L(x), U(x))$.

The proposed approach is to train a NN to provide in output the two bounds of the PI corresponding to a given input x. The goodness of the PI estimate attained with the NN-based model is described by two measures of quality: the PI Coverage Probability (PICP) and the Normalized Mean PI Width (NMPIW) (Khosravi et al., 2011b). Their mathematical definitions are:

$$PICP = \frac{1}{n_p} \sum_{i=1}^{n_p} c_i \tag{4}$$

where $c_i = 1$, if $y_i \in [L(x_i), U(x_i)]$ and otherwise $c_i = 0$;

$$NMPIW = \frac{1}{n_p} \sum_{i=1}^{n_p} \frac{(U(x_i) - L(x_i))}{t_{max} - t_{min}} \tag{5}$$

where t_{min} and t_{max} represent the true minimum and maximum values of the outputs (i.e., the bounds of the interval in which the true values fall), respectively.

2.2. Multi-objective optimization

The development of the NN-based model for PIs estimation implies the optimization of PICP (maximization) and NMPIW (minimization). In other words, the NN structure (number of hidden neurons) and parameters (weights and biases) must be determined so as to have the desired PICP with minimum PIW.

In all generality, a multi-objective optimization problem considers a number of objectives, equality and inequality constraints, and bounds on the decision variables. Mathematically the problem can be expressed as follows (Sawaragi, Nakayama, & Tanino, 1985):

Minimise/Maximise $f_m(x)$, $\quad m = 1,2, \dots, M$; $\tag{6}$

subject to $\quad g_j(x) \geq 0, \quad j = 1,2, \dots, J$; $\tag{7}$

$h_k(x) = 0, \quad k = 1,2, \dots, K$; $\tag{8}$

$x_i^{(l)} \leq x_i \leq x_i^{(u)} \quad i = 1,2, \dots, I.$ $\tag{9}$

A solution, $x = \{x_1, x_2, \dots x_I\}$ is an I dimensional decision variable vector in the solution space R^I. The solution space is restricted by the constraints in (7) and (8), and bounds on the decision variables in (9).

The final goal is to identify a set of optimal decision variable vectors x_i^*, $i = 1,2, \dots, n$ such that each solution included in the set cannot be regarded as better than any other with respect to all the objective functions $f_m(\cdot), m = 1,2, \dots, M$. The concepts of Pareto optimality and dominance drive the comparison among solutions: in case of a minimization problem, solution x_a dominates solution x_b $(x_a \succ x_b)$ if both following conditions are satisfied (Sawaragi, Nakayama, & Tanino, 1985):

$\forall i \in \{1,2, \dots, M\}, f_i(x_a) \leq f_i(x_b)$ $\tag{10}$

$\exists j \in \{1,2, \dots, M\}, f_j(x_a) < f_j(x_b)$ $\tag{11}$

If any of the above two conditions is violated, the solution x_a does not dominate the solution x_b, and x_b is said to be non-dominated by x_a. A solution is said to be Pareto optimal if it is not dominated by any other solution in the solution space. The set of all feasible non-dominated solutions in R^l is referred to as the *Pareto optimal set*, and for a given *Pareto optimal set*, the corresponding values of the objective functions form the so called *Pareto optimal front* in the objective functions space.

2.3. NSGA-II optimization of a NN for PIs estimation

NSGA-II is one of the most efficient multi-objective evolutionary algorithms (Deb et al., 2002). It generates a Pareto optimal solution set, rather than a single solution, via comparison of the qualities of different solutions by using an elitist approach (i.e., a fast non-dominated sorting and crowding-distance estimation procedure (Konak, Coit, & Smith, 2006)). The practical implementation of NSGA-II on our specific problem involves two phases: initialization and evolution. These can be summarized as follows (Ak et al. 2012):

***Initialization phase*:**

Step 1: Split the input data set into training (D_{train}) and testing (D_{test}) subsets.

Step 2: Fix the maximum number of generations and the number of chromosomes (individuals) Nc in each population. Each chromosome codes a solution by G real-valued genes, where G is the total number of parameters (weights and biases) in the NN: thus, each chromosome represents a NN. Set the generation number $n = 1$. Initialize the first population P_n of size Nc, by randomly generating Nc chromosomes (corresponding to NNs).

Step 3: For each input vector x in the training set, compute the lower and upper bound outputs of the Nc NNs.

Step 4: Evaluate the two objectives PICP and NMPIW for the Nc NNs; then, one pair of values 1-PICP (for minimization) and NMPIW is associated to each of the Nc chromosomes in the population P_n.

Step 5: Rank the chromosomes (vectors of G values) in the population P_n by running the fast non-dominated sorting algorithm (Konak, Coit, & Smith, 2006) with respect to the pairs of objective values, and identify the ranked non-dominated fronts $F_1, F_2, ..., F_k$ where F_1 is the best front, F_2 is the second best front and F_k is the least good front.

Step 6: Apply to P_n a binary tournament selection based on the crowding distance (Konak, Coit, & Smith, 2006), for generating an intermediate population S_n of size Nc.

Step 7: Apply the crossover and mutation operators to S_n, to create the offspring population Q_n of size Nc.

Step 8: Apply Step 3 onto Q_n and obtain the lower and upper bound outputs.

Step 9: Evaluate the two objectives in correspondence of the solutions in Q_n, as in Step 4.

***Evolution phase*:**

Step 10: If the maximum number of generations is reached, stop and return P_n. Select the first Pareto front F_1 as the optimal solution set. Otherwise, go to Step 11.

Step 11: Combine P_n and Q_n to obtain a union population $R_n = P_n \cup Q_n$.

Step 12: Apply Steps 3-5 onto R_n and obtain a sorted union population.

Step 13: Select the Nc best solutions from the sorted union to create the next parent population P_{n+1}.

Step 14: Apply Steps 6-9 onto P_{n+1} to obtain Q_{n+1}. Set $n = n + 1$; and go to Step 10.

Finally, the best front in terms of ranking of non-dominance and diversity of the individual solutions is chosen. Once the best front is chosen, testing of the trained NN with optimal weight values is performed using the data of the testing set.

3. Model Identification

In this study a systematic process is followed in order to identify the optimal NN structure (i.e., the number of hidden neurons) via cross-validation, taking into account both measures of PIs quality (i.e. coverage probability and width) and comparing the set of solutions obtained in each fold in terms of the hypervolume indicator introduced in (Bringmann & Friedrich, 2009). Fig. 1 shows a general scheme of this process.
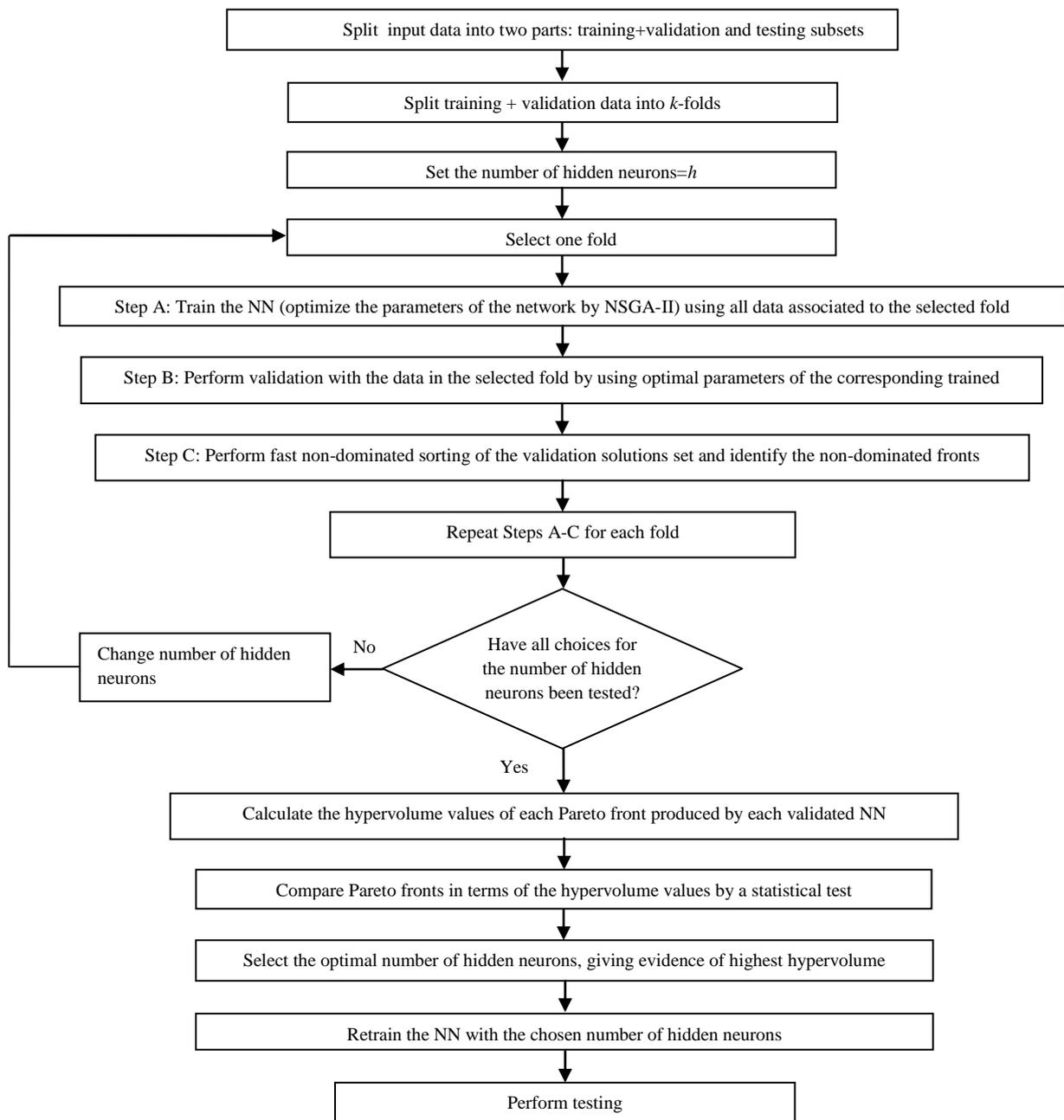

Fig. 1. Flowchart of the methodology.

3.1. k-fold cross-validation (CV)

Assessing the prediction accuracy, i.e. the generalization power, of a learning algorithm is essential for reliable prediction. In the case of NN, the structure of the model influences the learning capability. In practice, the choice of the number of network layers and neurons per layer often comes down to a compromise between the generalization error and the learning time (Khosravi, Nahavandi, & Creighton, 2010; Ileană, Rotar, & Incze, 2004). Cross-validation (CV) is an approach to evaluate the generalization performance of the NN, and it can be used for determining the optimal network architecture (i.e., the number of hidden neurons) (Setiono, 2001). CV is a statistical re-sampling method which uses multiple training and test subsamples (Zhang, et al. 1999). Different CV techniques such as *k*-fold CV, leave-one-out CV, bootstrap CV, etc, have been proposed in the statistical literature (Hastie, Tibshirani, & Friedman, 2008). In the basic *k*-fold cross-validation technique, the input data set is split into a partition of *k* equally (or nearly equally) sized segments or folds. At each round of cross-validation, one among the different folds is excluded from the dataset, and only the remaining *k*-1 folds are used for training; the excluded subset is then used for validation. The procedure is repeated until all the k folds have been used once for validation and *k*-1 times for training. Hence, the advantage of this technique is that, at least in successive rounds, all samples in the input data set are used for validation, while the dimension of the training set is kept high (Setiono, 2001). Fig. 2 demonstrates an example with *k* = 3. The entire data set is divided into 3 folds and in each CV iteration, for training we use a combination of two folds out of three that can be drawn from the whole data set: {2, 3}, {1, 3} and {1, 2}. Then, subsets {1}, {2}, and {3} are used for validation, respectively.

The prediction error obtained by using a CV strategy is sensitive to the specific way in which data have been split (Kwok, 1995). For small *k* values, the bias of *k*-fold cross-validation may become a problem in real-data analysis. If *k* = *N*, the so called leave-one-out CV, the cross-validation estimator is approximately unbiased for the true prediction error, but it has high variance and it is very computationally intensive for use in NN (Hastie, Tibshirani, & Friedman, 2008). For this reason, as we shall see, in our case study we use 20-fold cross-validation in order to minimize the bias-variance trade-off while also attaining the required accuracy in feasible computation times (Refaeilzadeh, Tang, & Liu, 2008; Wada & Kawato, 1992; Fushiki, 2011).

3.2. Comparison of Pareto fronts by the hypervolume indicator

The hypervolume indicator has been widely used as a measure to compare Pareto solution sets (fronts) returned by multi-objective optimizers (Bringmann & Friedrich, 2009). Given a Pareto front, it measures the volume of the portion of the objective space dominated by the front. Therefore, different multi-objective algorithms' performances can be compared in terms of the quality of the outcomes, by detecting the dominance between their different Pareto solution sets (Bringmann & Friedrich, 2009; While, Bradstreet, & Barone, 2012; While et al., 2006).

In our study, we calculate the hypervolume indicator by Monte Carlo simulation (Everson, Fieldsend, & Singh, 2002). A reference point, *R*, is selected as the "worst possible" point in the objective functions space. An estimate of the hypervolume (in percentage) is then obtained by sampling *N* uniformly distributed random points within the hyper-cuboids bounded by the reference point R in R^M. Then, the hypervolume indicator estimate is obtained as the percentage of the points dominated by the approximated Pareto front set P (composed by n

points in R^M), i.e. in a rejection sampling fashion (Cao, 2008). If a solution set A has a greater hypervolume than a solution set B, then A is taken to be a better set of solutions than B (While et al., 2006).

For the minimization problem of our two (positive) objectives, 1-PICP and NMPIW ($M = 2$), we split the hypervolume computation by partitioning the objective functions space into three regions with three different reference points of same NMPIW value but different CP. A schematic representation of the objective functions space splitting into three regions and of the position of the three reference points is given in Fig. 3. We fix NMPIW and not CP, because the latter is more important than PIW for our scopes. The overall hypervolume measure is obtained as the weighted sum of the partial hypervolumes in the three regions. By the splitting into three regions, we have given the flexibility to weigh differently the hypervolume measure obtained on different ranges of CP, coherently with the relevance of the corresponding region of the objective functions space.
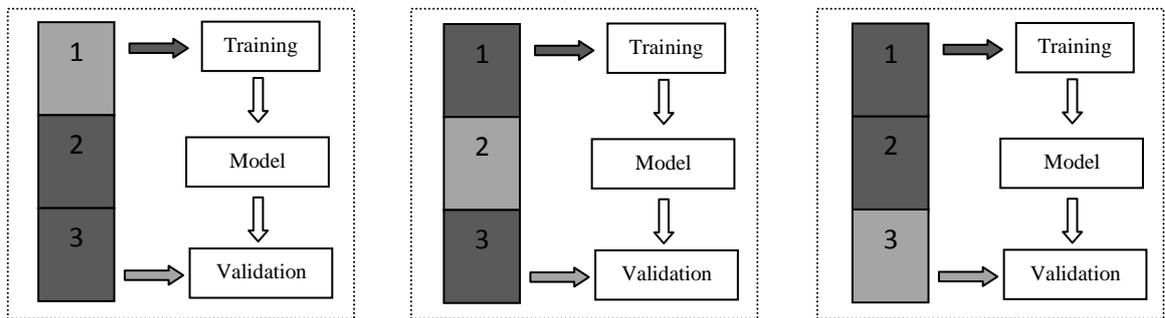

Fig. 2. Scheme of the CV procedure (Refaeilzadeh, Tang, & Liu, 2008).
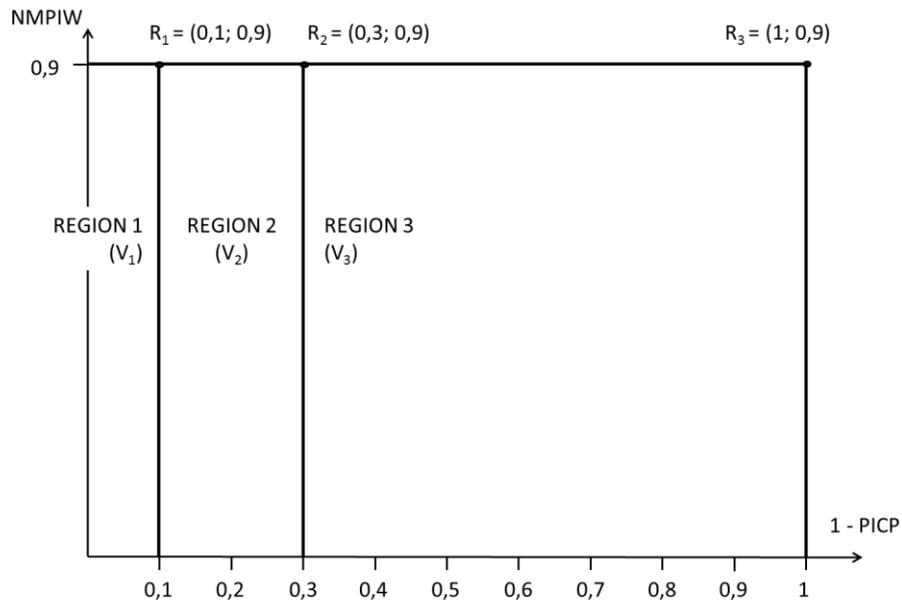
Fig. 3. A schematic representation of the splitting of the objective functions space into three regions for hypervolume computation. The overall hypervolume measure is obtained as the weighted sum of the partial hypervolumes (V_1, V_2 and V_3) computed in the three regions identified by the red lines. The three different reference points (R_1, R_2 and R_3) used in each region for partial hypervolumes computations are also indicated in the figure.

4. Case study

The case study concerns the scale (deposition) rate on the metal surfaces of equipment used in offshore oil wells. The output variable y is the scale rate; the influencing input variables are: temperature (T) and pressure (P), water composition (W) and fluid velocity (V) near the metal surfaces.

Data were obtained from experiments aimed at observing the process of deposition of the scale layer (Lins et al., 2011): if the scale layer achieves a predefined thickness, the equipment fails to properly perform its function. The total data set includes 131 observations; among these, the first 90% of the data (118 observations) are used for training and cross-validation purposes, and the rest is used for testing. All data have been normalized within the range [0.1, 0.9]. In order to perform k-fold cross-validation, the 118 training data are randomly partitioned into 20 subsamples, two of which include 5 samples while the others 6 samples.

The architecture of the NN consists of one input, one hidden and one output layers. The number of input neurons is 4, one for each input variable; the number of hidden neurons is chosen via the cross-validation process described in Section 3.1; the number of output neurons is 2, one for the lower and one for the upper bound values of the PIs. As activation functions, the hyperbolic tangent function is used in the hidden layer and the logarithmic sigmoid function is used at the output layer (these choices have been found to give the best results by trial and error, although the results have not shown a strong sensitivity to them).

In order to obtain an optimal NN architecture, 11 different choices for the number of hidden neurons, (5, 7-11, 13, 15, 17, 18, 20) have been explored. Hence, 220 NNs have been trained individually to obtain the results shown in Table 1.

After performing 20-fold cross-validation, we obtain 220 Pareto fronts, one for each fold and choice of the number of hidden neurons. The fronts are obtained after NSGA-II training of a NN with the training data associated to the relevant fold. In order to evaluate different neural network structures and select the optimal one, the Pareto fronts are compared in terms of their hypervolume indicators, V_1, V_2, V_3, on the partitioned objective functions space with reference points (1-PICP, NMPIW): $R_1 = (0.1, 0.9)$, $R_2 = (0.3, 0.9)$, $R_3 = (1, 0.9)$ (see Fig. 3). The hypervolume value V_1 indicates the dominated space between (0, 0.9) and (0.1, 0.9), which represents the region of interest in terms of coverage probability; the hypervolume value V_2 indicates the dominated space between (0.1, 0.9) and (0.3, 0.9); the hypervolume value V_3 indicates the dominated space between (0.3, 0.9) and (1, 0.9). The reference value for NMPIW has been fixed to 0.9, because there is no NMPIW value greater than 0.9.

To compare the Pareto fronts, a total hypervolume score, Vscore, for each Pareto front is computed as weighted sum of the V_1, V_2 and V_3 values. To give higher importance to the regions of higher PICP values, in our application we have arbitrarily chosen the weight vector $[w_1 \ w_2 \ w_3]$ to be [4/7 2/7 1/7]. Table 1 reports the total hypervolume scores computed on the validation data set, for each choice of the number of hidden neurons (different rows of the Table), and for each fold (different columns of the Table). For each fold and number of hidden neurons, the NN has been trained on the training data corresponding to the fold, and then used for prediction on the validation data included in the fold. Since the so obtained set of solutions does not necessarily form a Pareto front in the objective functions space, one step of non-dominated sorting of the solutions has been performed, before computing the corresponding total hypervolume score. The computation of V_{score} has also been done using the Pareto front resulting from the training of each NN: for the sake of brevity, we do not report

the Table of total hypervolume scores obtained from training data, but the results are synthetized in the boxplots in Fig. 4.

To fix the number of hidden neurons, it is natural to choose that number for which the trained NN leads to the highest total hypervolume score. However, for a given number of hidden neurons there are 20 different NNs trained on the 20 folds, and hence 20 total hypervolume scores. Due to the variability of the data included in the folds, there is no number of hidden neurons leading to total hypervolume scores consistently superior across all folds. Figures 4 and 5 show the boxplots of the total hypervolume scores for the different numbers of hidden neurons considered, with reference to the training and validation dataset, respectively. From Fig. 4 it is evident that the choice of 10 hidden neurons is optimal, with reference to the training dataset. For confirmation, a pairwise comparison of the median of the 20 total hypevolume scores obtained by the NNs with 10 hidden neurons, with the medians obtained by the NNs with other number of hidden neurons has been performed. The pairwise comparison is conducted by a statistical test, whose aim is rejecting the null hypothesis (H_0) of equality of the medians being compared; the test is based on asymptotic normality of the median and roughly equal sample sizes for the two medians being compared, and it is rather insensitive to the underlying distributions of the samples (Chambers et al., 1983; McGill, Tukey, & Larsen, 1978). By fixing the level of each test, i.e. the probability of rejecting a true H_0, to 10%, nearly all comparisons allow concluding for the superiority of the total hypervolume score obtained with a choice of 10 hidden neurons.
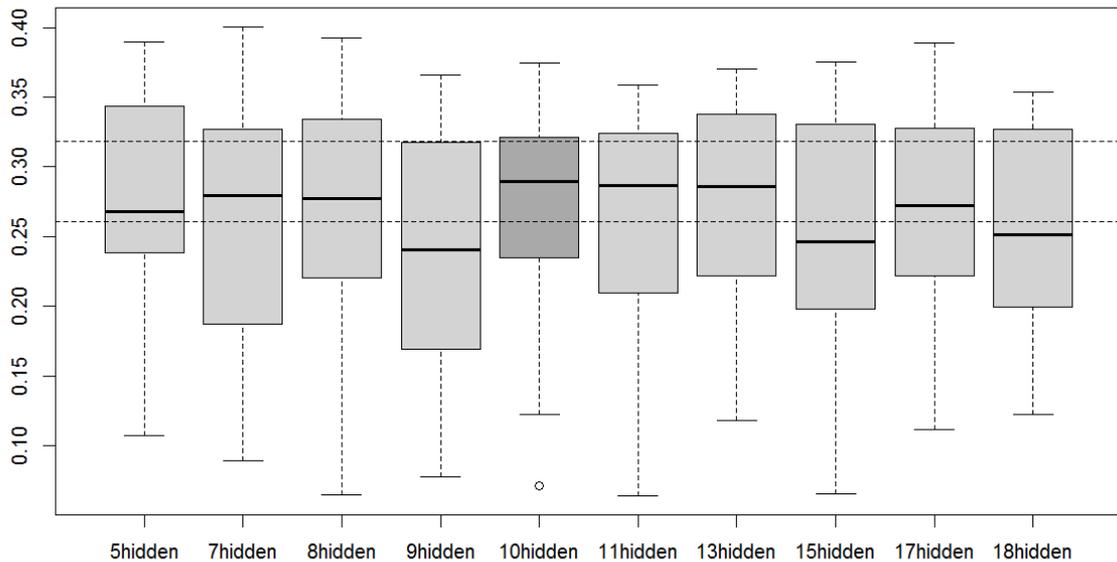
In Fig. 5, the boxplots of the total hypervolume scores with respect to the validation dataset are shown for all numbers of hidden neurons. The choice of 10 hidden neurons confirms to be one among the best, in terms of higher values of the median and of the lower whisker. Given also the superior performance on the training dataset, the choice of 10 hidden neurons is retained.

Table 1. The hypervolume scores of the Pareto fronts produced after validation of the NN with cross-validation procedure.

n_h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5	0,34	0,24	0,24	0,32	0,35	0,20	0,39	0,33	0,25	0,25	0,35	0,33	0,24	0,12	0,35	0,22	0,11	0,28	0,24	0,36
7	0,34	0,30	0,13	0,30	0,28	0,24	0,34	0,32	0,14	0,26	0,39	0,15	0,28	0,20	0,29	0,20	0,09	0,18	0,38	0,40
8	0,32	0,25	0,13	0,29	0,29	0,25	0,34	0,33	0,07	0,22	0,34	0,35	0,24	0,22	0,27	0,21	0,07	0,28	0,39	0,38
9	0,29	0,26	0,12	0,33	0,31	0,18	0,33	0,34	0,16	0,21	0,37	0,14	0,22	0,13	0,27	0,23	0,08	0,23	0,31	0,35
10	0,29	0,29	0,23	0,31	0,31	0,25	0,35	0,31	0,12	0,26	0,37	0,32	0,27	0,13	0,34	0,19	0,07	0,24	0,32	0,37
11	0,33	0,28	0,12	0,31	0,30	0,22	0,32	0,35	0,10	0,21	0,29	0,33	0,25	0,11	0,33	0,21	0,06	0,28	0,32	0,36
13	0,30	0,27	0,12	0,37	0,27	0,17	0,34	0,33	0,13	0,23	0,33	0,35	0,26	0,12	0,30	0,22	0,24	0,30	0,35	0,35
15	0,37	0,30	0,12	0,35	0,29	0,19	0,33	0,30	0,21	0,21	0,33	0,14	0,22	0,13	0,34	0,21	0,07	0,22	0,27	0,36
17	0,33	0,26	0,13	0,33	0,39	0,21	0,32	0,34	0,28	0,24	0,31	0,29	0,24	0,12	0,25	0,18	0,11	0,24	0,31	0,37
18	0,32	0,25	0,12	0,34	0,27	0,24	0,32	0,33	0,27	0,25	0,33	0,35	0,25	0,23	0,13	0,20	0,16	0,20	0,13	0,34
20	0,32	0,31	0,24	0,34	0,27	0,32	0,32	0,31	0,16	0,17	0,39	0,38	0,31	0,12	0,29	0,18	0,11	0,24	0,29	0,34
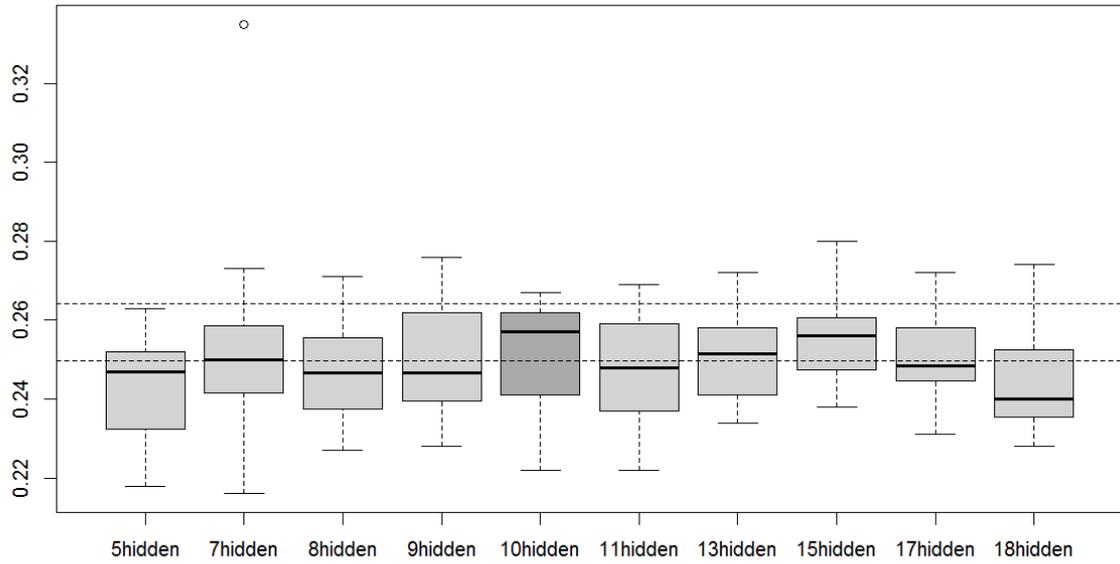
Fig. 4. Boxplots of the total hypervolume scores for different numbers of hidden neurons with respect to the training dataset. Each box extends from Q_1 to Q_3, where Q_1 and Q_3 are the first and third quartile of the dataset, respectively; the position of the median is evidenced in each box by a solid horizontal line. The upper (lower) whisker of each boxplot extends to the highest (lowest) value in the dataset smaller (greater) than $Q_3 + 1.5*IQR$ ($Q_1 - 1.5*IQR$), where $IQR = Q_3 - Q_1$. The boxplot corresponding to 10 hidden neurons is highlighted in dark grey. The horizontal dotted lines are the limits of the 90% confidence interval for the median total hypervolume score obtained with 10 hidden neurons: the medians falling outside these limits are statistically different from the one obtained with 10 hidden neurons.


Fig. 5. Boxplots of the total hypervolume scores for different numbers of hidden neurons with respect to the validation dataset. Each box extends from Q_1 to Q_3, where Q_1 and Q_3 are the first and third quartile of the dataset, respectively; the position of the median is evidenced in each box by a solid horizontal line. The upper (lower) whisker of each boxplot extends to the highest (lowest) value in the dataset smaller (greater) than $Q_3 + 1.5*IQR$ ($Q_1 - 1.5*IQR$), where $IQR = Q_3 - Q_1$. The boxplot corresponding to 10 hidden neurons is highlighted in dark grey. The horizontal dotted lines are the limits of the 90% confidence interval for the median total hypervolume score obtained with 10 hidden neurons: the medians falling outside these limits are statistically different from the one obtained with 10 hidden neurons.

After choosing the number of hidden neurons, the NN has been retrained using all data in the training and validation sets, for a total of 118 samples. The first (best) Pareto front found after training includes 50 non-dominated solutions and it is shown in Fig. 6.

To verify a posteriori the selection of 10 hidden neurons, retraining has also been performed for other numbers of hidden neurons: the resulting V_1, V_2, V_3 and V_{score} hypervolume values are reported in Table 2. Fig. 7 shows the trend of V_1 (top) and V_{score} (bottom) values with the number of hidden neurons. From inspection of Fig. 7, one can conclude that the choice of 10 hidden neurons corresponds to the highest hypervolume values V_1 (referred to the region of interest) and V_{score}.

Table 2. The hypervolume values of the Pareto fronts produced after retraining of the NN with different number of hidden neurons.

	5	7	8	9	10	11	13	15
V_1	0,28	0,26	0,27	0,33	0,33	0,32	0,32	0,28
V_2	0,17	0,15	0,14	0,12	0,14	0,13	0,12	0,14
V_3	0,28	0,31	0,30	0,28	0,27	0,28	0,28	0,29
V_{score}	0,25	0,24	0,24	0,26	0,27	0,26	0,26	0,24
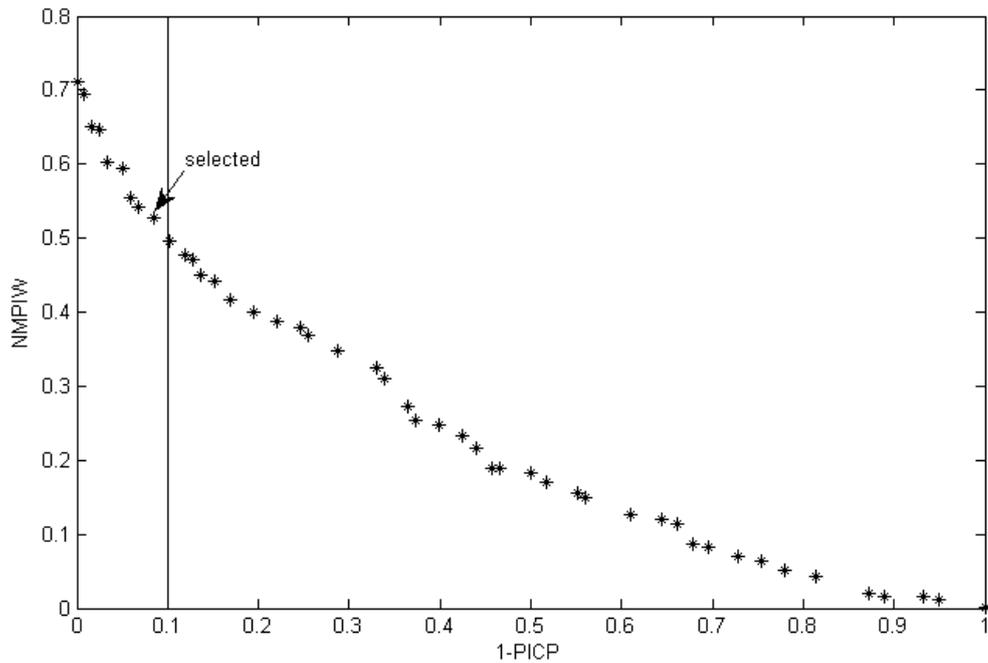
Fig. 6. The best Pareto front obtained by retraining of the NN with the optimal choice of 10 hidden neurons.
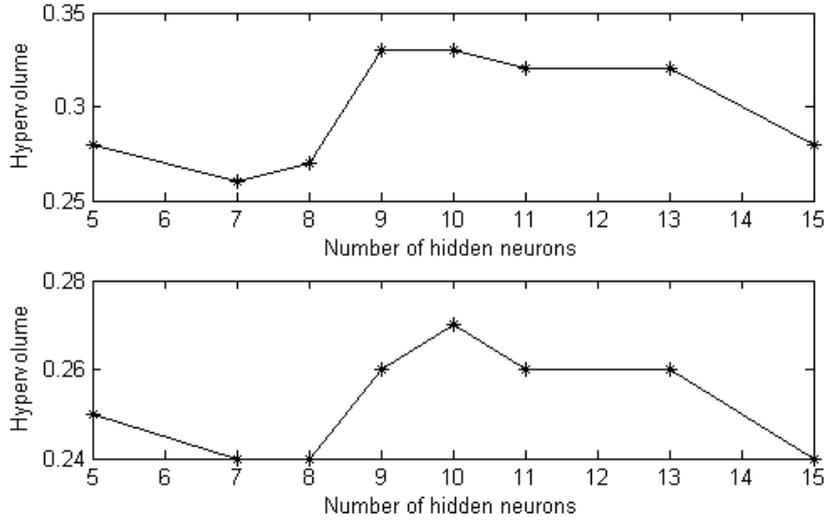
Fig. 7. Hypervolumes values V_1 and V_{score} of the fronts obtained after retraining versus the number of hidden neurons.

Finally, Fig. 8 shows the prediction intervals for the scale rate values in the test dataset obtained by the trained NN with 10 hidden neurons and corresponding to a Pareto solution chosen subjectively. The solution has been chosen as the one with smallest NMPIW among those with PICP ≥ 0.9 in Fig. 6. The results on the test dataset give a coverage probability of 100 % and an interval width of 0.494.
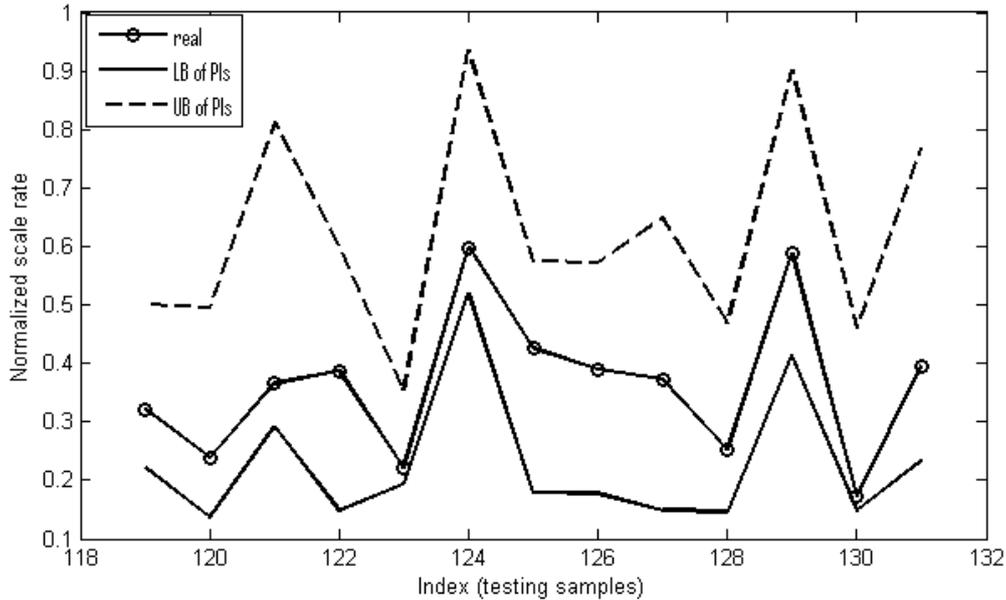

Fig. 8. The prediction intervals for scale rate on the test dataset.

5. Conclusion and Future Work

A method for the estimation of PIs by NN has been proposed for scale rate prediction. The originality of the approach is the multi-objective formulation of the problem, to achieve high coverage with intervals of small width. The multi-objective framework allows considering a set of optimal solutions to select from, according to preferences and to the application purposes. Moreover, a systematic process for selecting the optimal NN

structure (number of hidden neurons) for the problem at hand has been proposed, based on cross-validation analysis and on the comparison of hypervolume indicators. The approach is based on the quantitative evaluation of the superiority of performance attained with the chosen number of hidden neurons with respect to other possible choices, proved via statistical testing.

As future research, we aim at using ensemble methods to further increase the accuracy of the NN-based predictions. Moreover, we aim at exploring different measures for comparing Pareto solutions set.

References

Ak, R., Li, Y., & Zio, E. (2012). Estimation of prediction intervals of neural network models by a multi-objective genetic algorithm. In *Proceedings of the 10th International FLINS Conference on Uncertainty Modeling in Knowledge Engineering and Decision Making* (*Flins 2012*), *Istanbul, Turkey*.

Ak, R., Li, Y., Vitelli, V., & Zio, E. (2012). Estimation of wind speed prediction intervals by multi-objective genetic algorithms and neural networks. In *Acts of the XLVI Scientific Meeting of the Italian Statistical Society*, *Rome, Italy*.

Bringmann, K.., & Friedrich, T. (2009). Don't be greedy when calculating hypervolume contributions. In *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms* (*Foga'09*), *Orlando, USA* (pp.103-112).

Cao, Y. (2008). Computation of hypervolume indicator. Online: http://www.mathworks.com/matlabcentral/fileexchange/19651-hypervolume indicator/content/hypervolume.m. Access date: June 2012.

Chambers, J. M., Cleveland, W. S., Kleiner, B., & Tukey, P. A. (1983). *Graphical methods for data analysis*. New York: Champman & Hall.

Chatterjee, S., & Bandopadhyay, S. (2012). Reliability estimation using a genetic algorithm-based artificial neural network: An application to a load-haul-dump machine. *Expert Systems with Applications*, *39*, 10943-10951.

Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. (2nd edition). USA: Springer.

Cottis, R. A., Owen, G., & Turega, M. (2000). Prediction of the corrosion rate of steel in seawater using neural network methods. In *NACE International Corrosion 2000, Orlando, USA*.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*, 182-197.

Everson, R. M., J. Fieldsend,J. E., & Singh,.S. (2002). Full elite sets for multi-objective optimization. In *Proceedings of the fifth international conference on adaptive computing in design and manufacture* (*ACDM 2002*), *Devon, UK* (pp. 87−100).

Fushiki, T. (2011). Estimation of prediction error by using k-fold cross-validation. *Statistics and Computing*, *21*, 137-146.

Geisser, S. (1993), *Predictive inference: An introduction*. Chapman & Hall.

Gosselin, L., Tye-Gingras, M., & Mathieu-Potvin, F. (2009). Review of utilization of genetic algorithms in heat transfer problems. *International Journal of Heat and Mass Transfer*, *52*, 2169–2188.

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *The elements of statistical learning: Data mining, inference and prediction*. (2nd ed.). Springer-Verlag.

Ileană, I., Rotar, C., & Incze, A. (2004). The optimization of feed forward neural networks structure using genetic algorithms. In *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics* (*ICTAMI 2004*), *Thessaloniki, Greece*.

Khosravi, A., Nahavandi, S., & Creighton, D. (2010). A prediction interval-based approach to determine optimal structures of neural network metamodels. *Expert Systems with Applications*, *37*, 2377-2387.

Khosravi, A., Nahavandi, S., Creighton, D., & Atiya, A. F. (2011a). Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on Neural Networks*, *22*, 1341-1356.

Khosravi, A., Nahavandi, S., Creighton, D., & Atiya, A. F. (2011b). Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Transactions on Neural Networks*, *22*, 337-346.

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, *91*, 992-1007.

Kwok, T-Y. (1995). Efficient cross-validation for feed-forward neural networks. In *Proceedings of IEEE International Conference on Neural Networks*, *Perth, Australia* (pp. 2789-2794).

Larsen, T., Randhol, P., Lioliou, M., Jøsang, L. O., & Østvold, T. (2008). Kinetics of CaCO3 scale formation during core flooding, In *SPE International Oilfield Scale Conference*, *Aberdeen*, *UK*.

Lins, I.D., Moura, M.C., Droguett, E.L., Zio, E., & Jacinto, C.M. (2011). Reliability prediction of oil wells by support vector machine with particle swarm optimization for variable selection and hyperparameter tuning, In *Advances in Safety, Reliability and Risk Management Proceedings of the European Safety and Reliability Conference* (*Esrel 2011*), *Troyes, France*.

McGill, R., Tukey, J. W., & Larsen, W. A. (1978) Variations of box plots. *The American Statistician*, *32*, 12−16.

Moura, M.C., Lins, I.D., Ferreira, R. J., Droguett, E.L., & Jacinto, C.M.C. (2011). Predictive Maintenance Policy for Oil Equipment in Case of Scaling through Support Vector Machines. In *Proceedings of Advances in Safety, Reliability and Risk Management* (*Esrel 2011*), *Troyes, France* (pp.503-507).

Nyborg, R. (2002). Overview of CO2 corrosion models for wells and pipelines, In *NACE International Corrosion 2002*, Denver, USA.

Ozkol, I., & Komurgoz, G. (2005). Determination of the optimum geometry of the heat exchanger body via a genetic algorithm. *Numerical Heat Transfer Part A –Applications*, *48*, 283–296.

Refaeilzadeh, P., Tang, L., & Liu, H. (2008). Cross-validation. Online: http://www.cse.iitb.ac.in/~tarung/smt/papers_ppt/ency-cross-validation.pdf. Access date: June 2012.

Sawaragi, Y., Nakayama, H., & Tanino, T. (1985). Theory of multi-objective optimization. *Mathematics in Science and Engineering*, *17*, 1-293.

Setiono, R. (2001). Feedforward neural network construction using cross validation. *MIT Press Journals Neural Computation*. *13*, 2865-2877.

Sirinivas, N., & Deb, K. (1994). Multi-objective optimization using non-dominated sorting in genetic algorithms. *Journal of Evolutionary Computation*, *2*, 221-248.

Stamatakis, E., Stubos, A., & Muller, J. (2011). Scale prediction in liquid flow through porous media: A geochemical model for the simulation of $CaCO_3$ deposition at the near-well region. *Journal of Geochemical Exploration*, *108*, 115-125.

Wada, Y., & Kawato, M. (1992). A new information criterion combined with cross-validation method to estimate generalization capability. *Systems and Computers in Japan*, *23*, 955-965.

While, L., Bradstreet, L., & Barone, L. (2012). A fast way of calculating Exact Hypervolumes. *IEEE Transactions on Evolutionary Computation*, *16*, 86-95.

While, L., Hingston, P., Barone, L., & Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, *10*, 29-38.

Yang, L., Kavli, T., Carlin, M., Clausen, S., & F. M. de Groot, P. (2002). An evaluation of confidence bound estimation methods for neural networks. *International Series in Intelligent Technologies*, *18*, 71-84.

Yuan, M. D., Todd, A. C., & Heriot-Waft, U. (1991). Prediction of sulfate scaling tendency in oilfield operations. *SPE Production Engineering*, *6*, 63-72.

Zhang, G., Hu, M. Y., Patuwo, B. E., & Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. *European Journal of Operational Research*, *116*, 16-32.

Zio, E. (2006). A study of bootstrap method for estimating the accuracy of artificial NNs in predicting nuclear transient processes. *IEEE Transactions on Nuclear Science*, *53*, 1460-1478.