Memetic Comp. (2012) 4:249–261
DOI 10.1007/s12293-012-0100-4

REGULAR RESEARCH PAPER

Meta-heuristic improvements applied for steel sheet incremental cold shaping

José R. Villar · Silvia González · Javier Sedano · Emilio Corchado · Laura Puigpinós · Joaquim de Ciurana

Received: 7 March 2012 / Accepted: 25 October 2012 / Published online: 6 November 2012

Abstract In previous studies, a wrapper feature selection method for decision support in steel sheet incremental cold shaping process (SSICS) was proposed. The problem included both regression and classification, while the learned models were neural networks and support vector machines, respectively. SSICS is the type of problem for which the number of features is similar to the number of instances in the data set, this represents many of real world decision support problems found in the industry. This study focuses on several questions and improvements that were left open, suggesting proposals for each of them. More specifically, this study evaluates the relevance of the different cross validation methods in the learned models, but also proposes several improvements such as allowing the number of chosen features as well as some of the parameters of the neural networks to evolve, accordingly. Well-known data sets have been use in this experimentation and an in-depth analysis of the experiment results is included. 5×2 CV has been found the more interesting cross validation method for this kind of problems. In addition, the adaptation of the number of features and, consequently, the model parameters really improves the performance of the approach. The different enhancements have been applied to the real world problem, an several conclusions have been drawn from the results obtained.

Keywords Genetic feature selection · Cross validation methods · Neural networks · Support vector machines · Real world applications

J. R. Villar (✉)
University of Oviedo, Campus de Viesques s/n,
33204 Gijon, Spain
e-mail: villarjose@uniovi.es

S. González · J. Sedano
Instituto Tecnológico de Castilla y León, Burgos, Spain
e-mail: silvia.gonzalez@itcl.es

J. Sedano
e-mail: javier.sedano@itcl.es

E. Corchado
University of Salamanca, Salamanca, Spain
e-mail: escorchado@usal.es

L. Puigpinós
Fundación Privada ASCAMM, Barcelona, Spain
e-mail: lpuigpinos@ascamm.com

J. de Ciurana
University of Girona, Girona, Spain
e-mail: qim.ciurana@udg.edu

1 Introduction

It is known that the complexity inherited in most of the new real world problems, among them the steel cold shaping industrial process, grows as the computer capacity does. Higher performance requirements with a lower amount of data examples are needed due to the costs of generating new instances, especially in those processes where new technologies are used

In this sense, the steel cold shaping, which represents an effervescent area, is a relatively new technology in the production of lots with a small number of pieces. NNs have been used to find relationships between the mechanical properties of the cold-rolled sheets of interstitial free and the chemical composition of the steel, and the rolling and the batch annealing parameters [13]. NNs have been also applied for identification of the parameters for operating conditions [26,27]. To the best of our knowledge, no specific study has been published in steel sheet incremental cold shaping (hence-after, SSICS).

Over recent years, there has been a significant increase in the use of artificial intelligence and soft computing (SOCO) methods to solve real world problems. Many different SOCO applications have been reported: the use of exploratory projection pursuit (EPS) and ARMAX for modelling the manufacture of steel components [3]; EPS and neural networks (NN) for determining the operating conditions in face milling operations [15] and in pneumatic drilling processes [17]; genetic algorithms and programming for trading rule extraction [4] and low quality data in lighting control systems [21]; feature selection and association rule discovery in high dimensional spaces [20] or NNs and principal component analysis and EPS in building energy efficiency [18,19].

In a previous study, a method for estimating the operating conditions in SSICS was developed [14]. In that study, feature selection NN and SVM were used for choosing the most promising features. Besides, NN and a SVM models were applied for estimating some operating condition and to determine whether a set of operating conditions would produce faulty pieces or not.

The aim of the present study focuses on some of the issues that were left open in the previous work, in order to obtain more robust and reliable models. Such open issues includes analysing the relevance of the different cross validation methods, the reduction of the parameter setting for the method and the study of the relevance of including auto-tuning methods, all of them applied in problems where the number of features is similar to the number of examples in the data set. This paper analyses the different options, introduce simple solutions to some of them, evaluates through a complete experimentation and proposes conclusions are drawn from the obtained results.

The organization of this paper is as follows. The problem description is included in the Sect. 1.1. Next Section is concerned with the description of the GA FS proposed from the original study [14]. Section 3 deals with a discussion on several issues for improving this algorithm. The proposal for the topics analysed in previous section are tested with standard data sets and with the real problem case in Sect. 4. Finally, conclusions are drawn and future work goals are set.

1.1 Steel sheet incremental cold shaping

The SSICS process is based on the concept of incremental deformation. This technology allows the manufacturing of pieces of metal sheet through the iteration of small sequential deformation stages until the desired shape is achieved and avoiding the axis-symmetric restrictions due to incremental rotatory deformation.

Comparing the incremental cold shaping with traditional deformation technologies it can be said that the former reduces the cost of specific machine tools and the manufacturing costs dramatically.

This type of technology has evolved from the well-known rapid manufacturing, allowing to generate pieces with complex geometries in a wide spread of materials without the need of frameworks or specific tools.

The main part of cold shaping has been controlled using numerical controlled tools in order to ensure as most as possible the fast, reliable, and low-cost manufacturing of lots with an small amount of metal pieces and prototypes. The scheme of metal sheet incremental cold shaping process is shown in Fig. 1.

The process of cold shaping starts with the design of a geometric shape in a 3D CAD file. This file should include as many layers as desired, each layer represents the bounds to be reached in each deforming step and are piled vertically. Consequently, the piece should be generated using the sequential and incremental layers, each one at a different depth and constraint within the defined bounds.

Plenty of parameters have to be fixed for the manufacture a metal piece, among them the force developed by the deforming head in each of the three dimensions, the speed change, the trajectory of the head, the surface roughness, the sheet pressure stress, the incremental step between layers, the number of steps or stages, the attack angle, the angle variation, the depth variation, etc.

From the computational point of view there are two problems to solve, a two-class problem and a regression problem. In both cases, feature selection since the optimum machinery parameter combinations are still unknown, therefore, experts are dealing with a completely new process and the operation

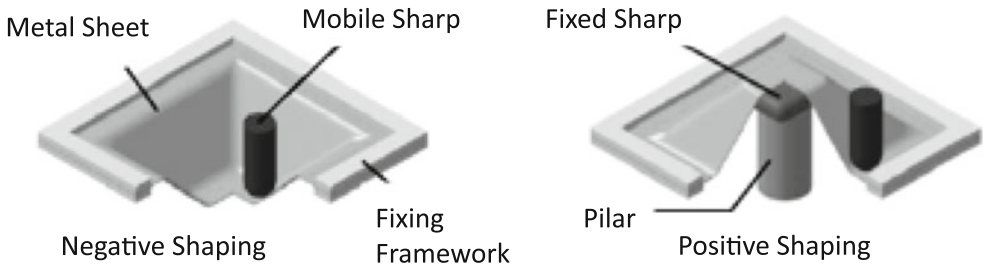

Fig. 1 The incremental cold shaping process of a steel sheet. A sharpening tool is iteratively applied onto the metal sheet at a different depth. In the negative shaping only the sharp tool is moved, while in the positive shaping both the metal sheet and the sharp tool are moved

conditions are not clearly understood. The two-class problem aims to model the operating conditions so the suitability of the experiment could be established. In other words, the aim is to analyse whether the operating conditions would generate a faulty piece or not while selecting the most relevant features involved. The repression problem aims to model the maximum suitable depth that can be reached with the given operating conditions.

2 Genetic algorithms and feature selection

In order to obtain a suitable feature subset some requirements are needed. As there are integer, nominal and real valued features, the algorithm should deal with any kind of data. Therefore, the same approach should be valid for the both subproblems, the two-class problem and the maximum depth. Besides, not only the best feature subset but also the best model are desired for each problem, a classifier in the former case and a regression model in he latter.

It is known that for this kind of problems the wrapper approach for feature selection performs better than filter solutions [5,23]. These studies proposed wrapper feature selection methods using genetic algorithms (GA) for dealing with the feature subset selection, that is, each individual is a feature subset. To evaluate individuals a modelling technique has been applied: the former proposed a lazy learning model as the K-Nearest Neighbour, the latter made use of a NN model that iteratively fix the number of hidden neurons.

Different approaches as to how the NN is learnt have been studied. In [1] a GA approach to fingerprint feature selection is proposed and selected features are supplied as input to NN for fingerprint recognition, while in [22] a similar approach has been applied to automatic digital modulation recognition. Moreover, this type of approach has been reported to perform better than using statistical models [25]. Besides, Support Vector Machines (SVM) have been also used in conjunction with evolutionary feature selection to reduce the input space dimensionality [9,11].

In this study, two different approaches are analysed. The first one is a specific NN+GA wrapper feature selection method for estimating the maximum depth problem, while the second approach makes use of SVM instead of NN for determining whether a set of operation conditions would produce a faulty piece or not. Preliminary studies [14] show that this combination leads to a valid solution for the SSICS problem.

2.1 GA+SVM+NN feature selection

In this study we adopt two different solutions depending on whether we are dealing with the two-class or the maximum depth problem. A hybridized method of GA evolving feature subsets and a SVM classifier is chosen in the former case, while in the latter a hybridized method of GA evolving feature subsets and a NN for modelling the desired output is used. In both modelling and feature selection problems the GA is a steady state approach with the percentage of elite individuals to be defined as a parameter. The algorithm has been implemented in Matlab [12], using both the NN and the SVM toolboxes.

The algorithm is outlined in Algorithms [1,2]. Algorithm [1] evaluates an individual (which is to say, a feature subset), while the latter shows the GA that evolves the feature subset and that calls Algorithm 1 whenever required. Actually, Algorithm 2 contains the main algorithm.

For the sake of simplicity we have neither reproduced the algorithm for the SVM nor for the NN cases. Instead, we

Algorithm 1 IND_EVALUATION: Evaluates an individual

Require: I the input variables data set
Require: O the output variable data set
Require: ind the individual to evaluate, with its feature subset
  $model$ {the best model learned for ind}
  $mse = 0$ {the associated mean of Mean Square Error for ind}
  $indMSE = 0$ {best MSE found in the cross validation}
  for $k = 1$ to 10 **do**
    {run the k fold in the cross validation scheme}
    generate the train and test reduced feature data set
    initialize the model $indModel$
    train $indModel$ with the train data set
    $indKMSE \leftarrow$ calculate the MSE for $indModel$ with the test data set
    $mse+ = indKMSE$
    if $k == 1$ **or** $indMSE > indKMSE$ **then**
      $indMSE = indKMSE$
      $model = indModel$
    end if
  end for
  $mse = mse/10$
  return $[model, mse]$

Algorithm 2 GA$^+$ Feature Selection

Require: I the input variables data set
Require: O the output variable data set
Require: N the feature subset size
  $FS \leftarrow \{\ \emptyset\}$ {the best feature subset}
  $model$ {the model learned for FS}
  $mse = 0$ {the associated mean of Mean Square Error for FS}
  Generate the initial population, Pop
  for all individual ind in Pop **do**
    $[ind.model, ind.mse] = IND_EVALUATION(I, O, ind)$
  end for
  $g \leftarrow 0$
  while $g < G$ **do**
    while $size(Pop') < (popSize - |E|)$ **do**
      Generate new individuals through selection, crossover and mutation
      add valid individuals to Pop'
    end while
    extract the elite subpopulation $E \in Pop$
    for all individual ind in Pop' **do**
      $[ind.model, ind.mse] = IND_EVALUATION(I, O, ind)$
    end for
    $Pop = \{E \cup Pop'\}$
    sort Pop
    g++
  end while
  $FS \leftarrow Pop[0]$
  $[model, mse] \leftarrow$ corresponding model and MSE
  return $[FS, model, mse]$

present the general case in the algorithms, and when it is said that a model is trained, the reader should consider which problem (the two-class or the regression problem) is related to the use of NN or SVM. The decision on how to fix the parameters for the model was based on preliminary studies and the corresponding experimentation was carried out [14].

The typical steady state GA parameters, like the crossover and mutation probabilities, the number of generations, the population size and the elite population size, are all given for each experiment. The individual representation is the string of indexes of the chosen feature subset. The tournament selection is implemented and one point crossover is used. After each genetic operation the validity of the offspring is analysed: repeated feature indexes are erased and random indexes are introduced to fill the individual feature subset.

Third order polynomials are used as kernel functions for the SVM. The number of hidden nodes in the NN is set as a parameter. The NN models are generated randomly and trained using the corresponding Matlab library functions. In the original approach, 10-fold cross validation is used, and the mean value of the mean squared error in each fold is the fitness of an individual.

3 Issues for enhancing the FS for SSICS

There are several issues for enhancing the above detailed FS method. Firstly, this FS method makes use of cross validation (hereinafter, CV). An interested reader might want to know the performance of the proposal using different CV schemes, especially for the regression problem. On the other hand, it is worth comparing different criteria for choosing the best feature subset for the classification problem. Finally, a predefined dimension is set for all the feature subsets. It is interesting to evaluate the algorithm when the feature subset size is given an upper bound and the dimensionality of the different feature subsets is allowed to evolve freely. Though the auto-tuning of the different models parameters is also quite interesting for the development of this kind of approaches, it is left as future work.

As seen in Algorithm 1, k-fold cross validation is used to evaluate the models and to select the one with the best mean error. Nevertheless, it is of interest to evaluate the performance of the feature selection using different types of CV, mainly in the industrial real world problems. This type of problem does not lack in high dimensionality and the number of features in a data set are usually relatively small. Moreover, the number of samples in the data sets is also small due to the cost of generating and gathering such data. In the case of SSICS, the cost of gathering the data includes designing and testing several pieces, so time and material for this task are used.

Consequently, it is worth comparing the results obtained with different CV methods, particularly, Leave-One-Out (hereinafter, LOO), 5×2 CV and 10-fold CV. LOO is a CV method devoted to regression problems, for which a collection of n pairs of train and test subsets is generated from the original data set, with n being the number of samples.

Each pair is generated choosing one sample from the original data set as the test subset, while keeping the rest of the data set as the train subset. LOO would provide an almost unbiased estimator of generalization performance and is known for its possible high-level of variability [7], but it is considered a useful measure to estimate the generalization of model [8], especially in model selection when the data set size is considered to be small [24].

On the other hand, 5×2 and 10-fold CV methods resamples the available observations into 2 disjoint subsets for training and testing purposes. These CV methods can be used for the validation either classification or regression problems. In the case of 5×2 CV, the original data set is divided in two parts of about the same size, then one part is used for training and the second for testing and vice-versa. Repeating this process 5 times we obtain ten independent runs. Besides, 10-fold CV generates 10 disjoint pairs of train and test subsets; again, we obtain ten independent runs [6]. Interestingly enough, when the data set number of samples is small, the three CV schemes also generate data subsets of similar sizes.

This study compares the three CV methods for the regression problem. In all the cases, the mean square error (MSE) is used to evaluate the models. Each feature subset is then assigned the mean value of the best model found in each independent run. The best model found among the different independent runs will be the one chosen. Actually, we obtain more than just the mean value: we obtain the deviation and more statistics, which can be used to compare the results. Nevertheless, analysing such extra statistic information introduces more computational requirements as we make use of imprecise objective instead of crisp, different strategies should be considered [16].

Apart from the regression problem, this study will compare the results of the main classification error of the feature subsets against the ROC criteria for the classification problem [2]. As detailed above, the current approach calculates the classification error for each fold and then evaluates each feature subset with the mean classification error. It could be interesting to calculate the true positive rate (TPR) and the false positive rate (FPR) in each fold and to compare the feature subsets using the the mean ratio TPR/FPR. The area under the ROC curve will not be considered as it has been found that for small sample data sets there is evidence of noise in the conclusions extracted from this criterion [10].

Hence, Algorithm 1 should be adapted to accept a new parameter with the type of CV to be carried out and then it should generate the corresponding collection of train/test data sets before starting the fitness measurement. In case of regression, the MSE will be used as a fitness function; while the mean ratio TPR/FPR will be used as the fitness function for the classification problems.

In addition, an adaptation of the GA to allow different feature subset sizes will be compared to the original method. This adaptation will generate either NN or SVM models using the same set of parameters. A simple improvement is to introduce a variation in the parameters according to the number of features that each individual chooses. Quite a simple approach will be used for evaluating the improvements this extension introduces: the number of hidden neurons will be a function of the size of the feature subset.

To implement these options, a genetic algorithm for feature selection with a variable feature subset size was developed. The individual representation is based on a Boolean array setting whether the feature is chosen or not. The number of features chosen can be fixed -to cover the original approach- or variable with MAXFS being the maximum number of features to be chosen.

The genetic operators have been adapted to the individual representation. The crossover is a one-point crossover: whenever the feature subset is of a fixed size, the crossover produces offsprings of the same feature subset size; otherwise, the offspring feature subset size varies from 1 to the MAXFS value. The mutation operator randomly changes the status of each feature from selected to unselected and vice versa. After adapting the genetic operators, the validity of the individuals with respect to the size of the feature subset is tested and the invalid individuals are eliminated.

4 Experimentation and results

The different adaptations proposed in previous section are to be analysed. Firstly, the different types of CV are to be compared for the regression problem. Then, the relevance of allowing the parameters of the method to adapt to the problem will be evaluated: the number of selected features will evolve and the NN number of hidden neurons will be fixed or proportional to the size of the feature subset. Finally, the discussion of the obtained results will conclude in some improvements to the original method detailed in Sect. 2. This enhanced new approach will be compared with the original method, with both methods facing the SSICS data set. The results concerning the CV method and the variable size of feature subset will be extended in future to SVM and classification problems.

For the two former experiments several public data sets have been collected. As the proposed method is to develop with problems of relatively small number of features and reduced number of samples or instances, all the chosen data sets are of relatively small dimension. Interested readers may wonder what relatively small number of features means: in the context of this study, we refer to problems with fewer than 100 features, where the number of instances is similar to the number of features. Moreover, the instances containing missing values, if any, have been deleted. In addition, the data sets have been resampled to reduce the number of

instances to resemble the type of problem this study aims at. For each case, the description of the filtering method will be explained, and the number of instances in the reduced data set will be shown. Interested readers will notice that no preemptive action has been considered in the filtering process.

4.1 Regression and the cross-validation methods

Several well-known regression public data sets have been selected for comparing the results of the NN based FS algorithm for the different CV methods, which are all reflected in Table 1. Nevertheless, they have been manually processed to reduce them to the focused problem type. Afterwards, the number of instances in the validation files are 1 for the LOO, about 5 for the 10-fold CV and about 22 for the 5×2 CV.

In the case of the Wisconsin breast cancer (WBC), the time to recur is the output value. Consequently, only the recurrent instances are considered. The second feature from the original data set (Recur/Does not Recur) is, therefore, not included. Besides, only the instances from the area with coordinates (3, 4) were chosen for the forest fires (FF) data set, reducing its dimension to 10 features including the predicted variable. The communities and crime (CC) data set has been filtered as follows. Features with almost all the instances with missing values have been deleted (features 102 to 118 and 122 to 127). The instances that remain with missing values were also deleted. Next, the instances for state number 36 were chosen, thus this feature has been deleted. Finally, feature 4 with the names of the cities was deleted, as well.

As NN is to be used in the modelling part of the feature selection GA method, the data set is normalized with means 0 and deviations 1.

The GA parameters have been fixed as follows: 50 individuals in the population, 100 of generations, the probability of crossover equals 0.75, while the mutation probability is 0.25. A steady state GA evolutionary scheme is used, with a number of 5 elite individuals that will be kept in the next generation.

The size of the feature subset has been fixed to three for the forest fires data set, otherwise it is set to five. The feed forward back-propagation NNs includes 6 neurons in the hidden layer, with $\mu = 0.001$, $\mu_{dec} = 10^{-10}$, and $\mu_{inc} = 10^{-6}$. The parameters of the NN are kept constant during the feature selection and the model learning. Each experiment has been run 10 times for statistical evaluation purposes.

Table 1 Data sets used for evaluating the CV methods relevance

Data set	# Attr.	# Instances	Reduced # Attr.	Reduced # Instances
Wisconsin breast cancer	34	198	33	46
Forest fires	13	517	11	43
Communities and crime	128	1994	104	44

Table 2 Results of the algorithm with the different CV methods and for the considered test/validation data sets

CV method	Statistic	FF	WBC	CC
10-fold	Mean MSE	0.6160	0.6990	0.5855
	Median MSE	0.6119	0.6948	0.5938
	MSE deviation	0.0190	0.0370	0.0258
5×2	Mean MSE	0.5157	0.4884	0.4282
	Median MSE	0.5121	0.5007	0.4361
	MSE deviation	0.0298	0.0305	0.0274
LOO	Mean MSE	0.5162	0.6326	0.5092
	Median MSE	0.5234	0.6343	0.5092
	MSE deviation	0.0252	0.0233	0.0208

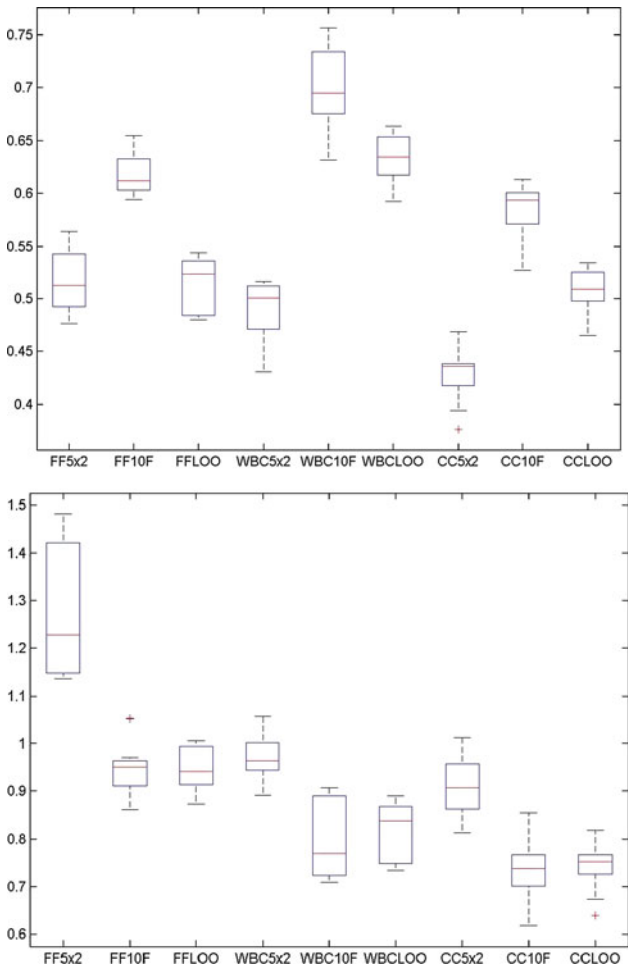

Fig. 2 Boxplot for the MSE values of the best individuals found for each data set. On *top*, the box plot with the test subsets; on *bottom*, the box plots with the whole data set (both train and test sets)

The main statistics of the results from experimentation are shown in Table 2. The 5×2 CV seems to outperform the rest of the CV methods as the median and MSE are smaller, though the MSE deviation is higher than that of the LOO. From Fig. 2, and when considering only the test data in order to select the best CV method (top part of the figure), the 5×2

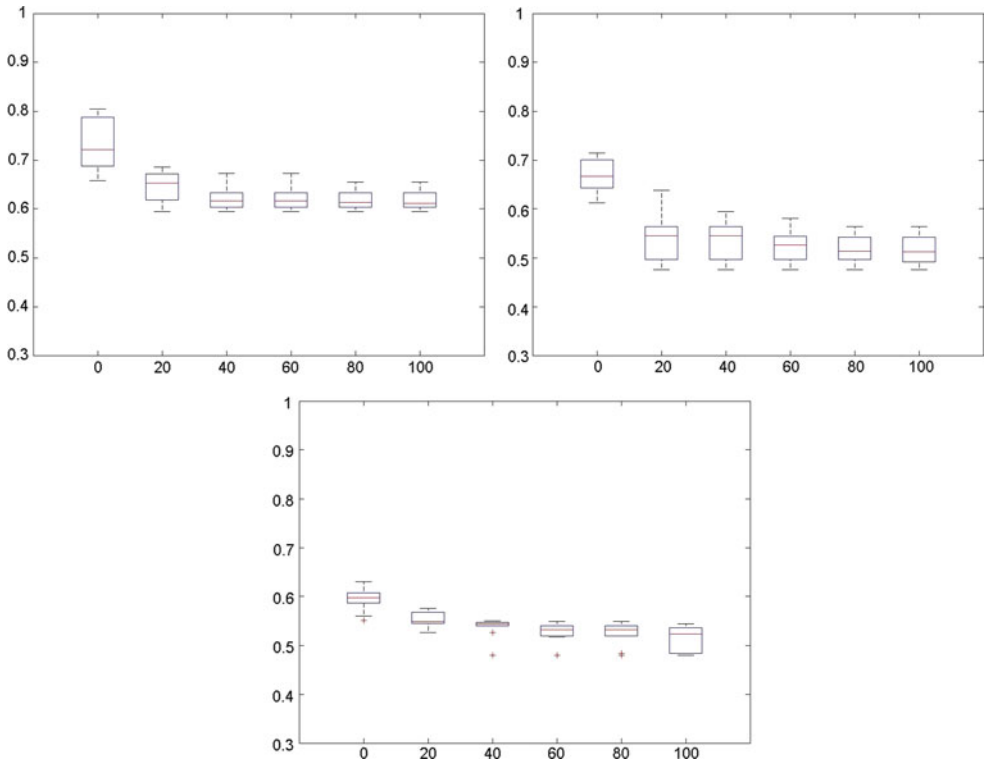

Fig. 3 Evolution of the MSE with the generations for the *FF data set* and the different CV methods. The *top-left*, *top-right* and *bottom* boxplots correspond with the 10-fold, 5×2 and LOO CV methods results

CV has a better performance due to its lower MSE value but also due to its extremely small deviation. Similarly, the method LOO behaves better than the 10-fold CV. Nevertheless, quite a different scenario deploys when the best model found in each run is evaluated with the whole data set (bottom part from the same figure), where the 10-fold and the LOO seem to clearly outperform the 5×2 CV.

Finally, the historical evolution of the best individual within the populations for each generation is presented from Figs. 3, 4, 5 for the FF, WBC and CC data sets correspondingly; the MSE is calculated for the validation test set. All the methods behave with neither premature convergence nor high deviation values for the whole collection of test data sets, with a rather similar evolution and with the above mentioned MSE characteristics for each method.

After the analysis of the results, different conclusions are inferred based on the validation data set considered: on the one hand, the test data; on the other hand, the whole data set. In general, 5×2 CV and 10-fold CV are faster than LOO due to the number of training sessions the latter induces. For the first case -using the test set for validating-, the 5×2 CV has the best mean and median statistics for the MSE, while its MSE deviation is similar to that obtained for the LOO. If we consider the whole data set for validation purposes, the LOO or the 10-fold CV clearly outperform the 5×2 CV. The good performance of the LOO is due to the fact that the models have been trained with all but one example, so the the error with the whole data set is clearly similar to that of validation: with the LOO we cannot infer what would happen with an unknown new example. Considering the problem of choosing a CV method, it could be said that the 5×2 CV is the best candidate when facing problems of relatively high dimension and quite few samples.

4.2 Evaluating different parameters configuration for the regression case

The data sets to be used in this experimentation are the same as the ones used in Sect. 4.1 and reflected in Table 1. The same manual filtering and pre-processing steps previously mentioned are considered in this experimentation. The objective of this experimentation is to evaluate whether it is better to allow the parameters of the method to evolve or to adapt. This is to be tested for the regression problem; if letting the parameters evolve improves the performance, then the same study should be conducted for the classification problem. Consequently, the original algorithm is compared to the variable number of chosen features released and the variable number of chosen features with adaptation of the number of hidden neurons.

Hence, this experiment will evaluate: the GA FS method with a fixed number of selected features and a fixed number

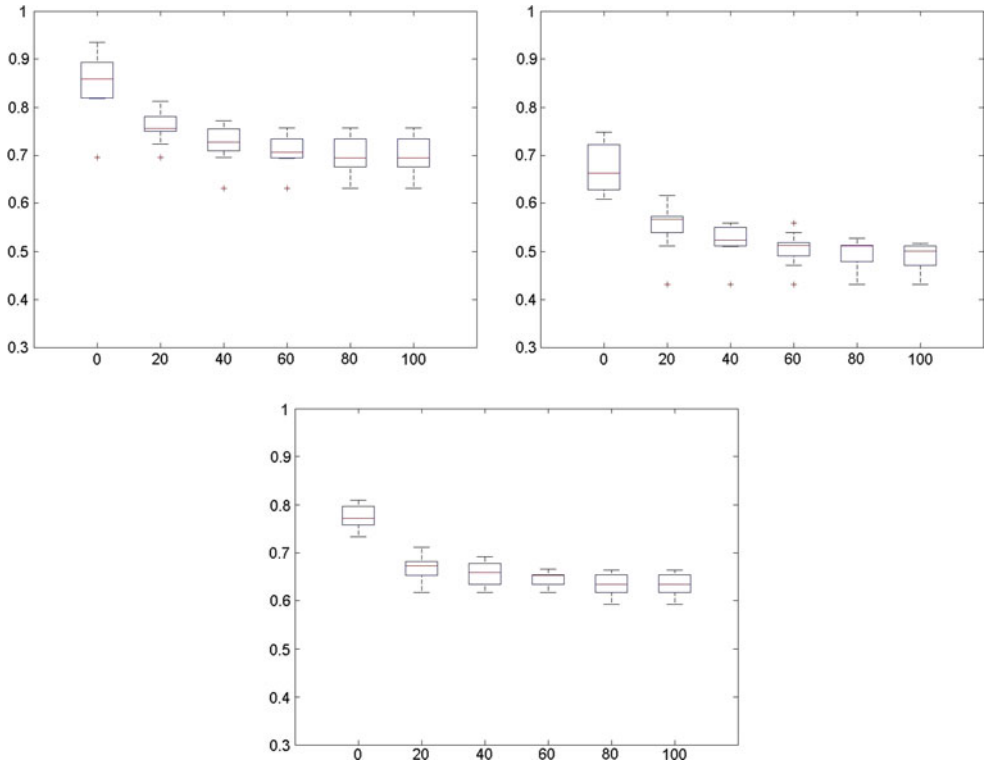

Fig. 4 Evolution of the MSE with the generations for the *WBC data set* and the different CV methods. The *top-left*, *top-right* and *bottom* boxplots correspond with the 10-fold, 5×2 and LOO CV methods results

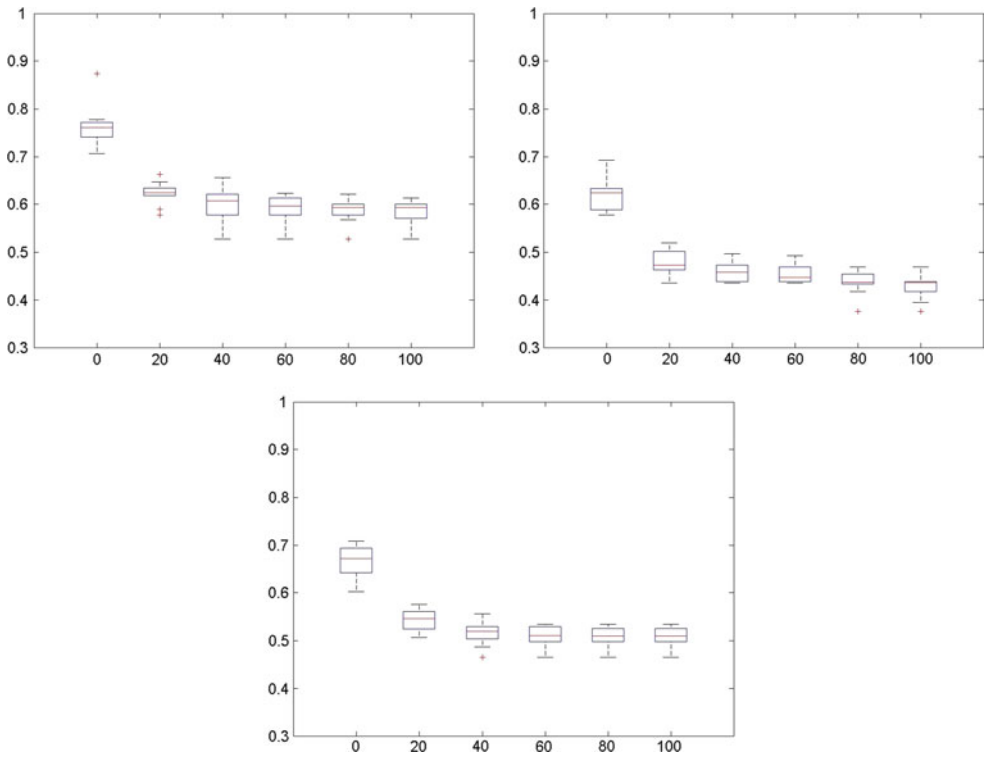

Fig. 5 Evolution of the MSE with the generations for the *CC data set* and the different CV methods. The *top-left*, *top-right* and *bottom* boxplots correspond with the 10-fold, 5×2 and LOO CV methods results

Table 3 Results from the 10-fold CV regression versus dynamic parameterization for the different data sets

CV method	Statistic	FF	WBC	CC
ORI	Mean MSE	0.6160	0.6990	05855
	Median MSE	0.6119	0.6948	0.5938
	MSE deviation	0.0190	0.0370	0.0258
	Feature subset size	3(10)	5(10)	5(10)
DFS	Mean MSE	0.6001	0.7084	0.5830
	Median MSE	0.6115	0.7098	0.5917
	MSE deviation	0.0331	0.0178	0.0200
	Feature subset size	2(4) 3(3)	2(4) 3(3)	3(3) 4(5)
		4(2) 5(1)	4(2) 5(1)	5(2)
DFDP	Mean MSE	0.5823	0.6911	0.5645
	Median MSE	0.5892	0.6782	0.5613
	MSE deviation	0.0279	0.0339	0.0283
	Feature subset size	1(1) 2(3)	2(6) 3(2)	2(1) 3(3)
		3(4) 4(2)	4(2)	4(4) 5(2)

For the feature subset size, the number in parenthesis is the runs in which the corresponding feature subset size was obtained

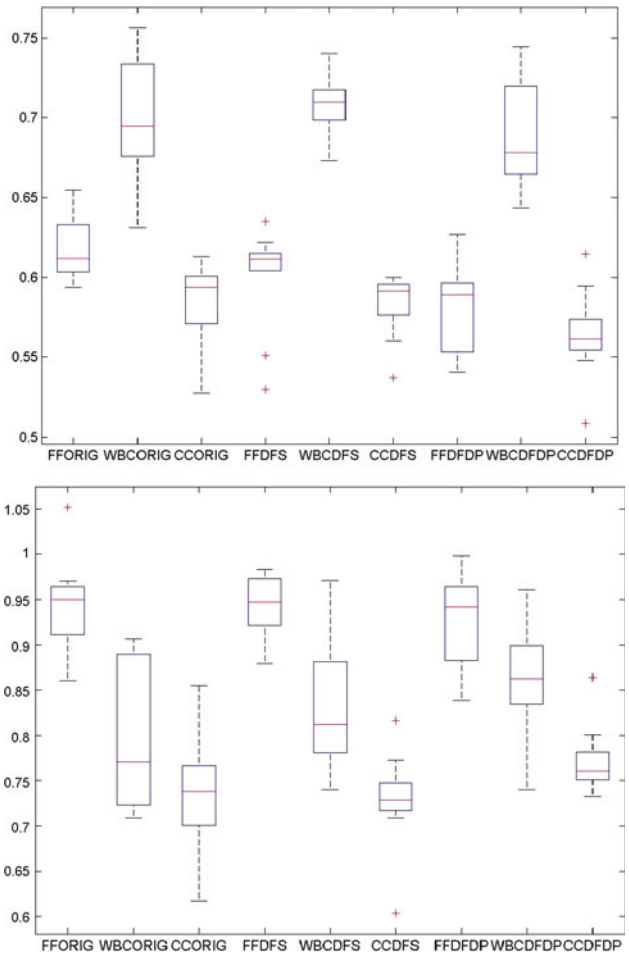

Fig. 6 Boxplot for the MSE values of the best individuals found for each data set and parameter combination -ORI, DFS and DF. On *top*, the box plot with the test subsets; on *bottom*, the box plots with the whole data set (both train and test sets)

of hidden neurons; the GA FS method with a maximum number of selected features and a fixed number of hidden neurons, too; and the GA FS method with a maximum number of selected features and the number of hidden neurons as a function of the number of features each individual chooses. In this latter case, the number of hidden neurons n_{hn} is determined linearly with the number of features each individual chooses (N) as stated in Eq. 1, with the n_{hn}, which has been bound to the range [4, 10].

$$n_{hn} = \min(10, \max(4, 4 + 2 \times (N - 4))) \quad (1)$$

The GA parameters have been fixed as follows: 50 individuals in the population, 100 of generations, the probability of crossover equals 0.75, while the mutation probability is 0.25. A steady state GA evolutionary scheme is used, with a number of 5 elite individuals that will be kept in the next generation.

The size of the feature subset (or the maximum number of features to select, correspondingly) has been fixed to five for all the cases. Whenever fixed, 6 hidden neurons of the feed forward back-propagation NNs have been used. The rest of NN parameters are $\mu = 0.001$, $\mu_{dec} = 10^{-10}$, and $\mu_{inc} = 10^{-6}$, which have been kept constant during the feature selection and the model learning. As in the previous Subsection, each experiment has been run 10 times for statistical evaluation purposes and 10-fold CV is used to compare results with the original approach. Consequently, three different sets of runs will be carried out: the original approach (hereinafter: ORIG), the Dynamic Feature Subset (hereinafter, DFS) with a variable size of the feature subset, and the Dynamic Feature subset with Dynamic neural network Parameters (hereinafter DFDP) with the hidden neurons calculated as a function of the size of the feature subset according to Eq. 1.

The main statistics of the results from experimentation are shown in Table 3. The box plots in Fig. 6 show the results for the best individual found in each run for the different methods. Clearly, the method using the DFDP approach improves the outcome of the algorithm. In fact, even the DFS method outperforms the ORIG design. Nevertheless, for the DFDP it is clear that either the number of hidden neurons or the number of generations must be increased to obtain better MSE values.

Finally, the historical evolution of the best individual within the populations for each generation is presented from Figs. 7, 8, 9 for the FF, WBC and CC data sets correspondingly; the MSE is calculated for the validation test sets. All the methods behave with neither premature convergence, nor high deviation values for the whole collection of test data sets,

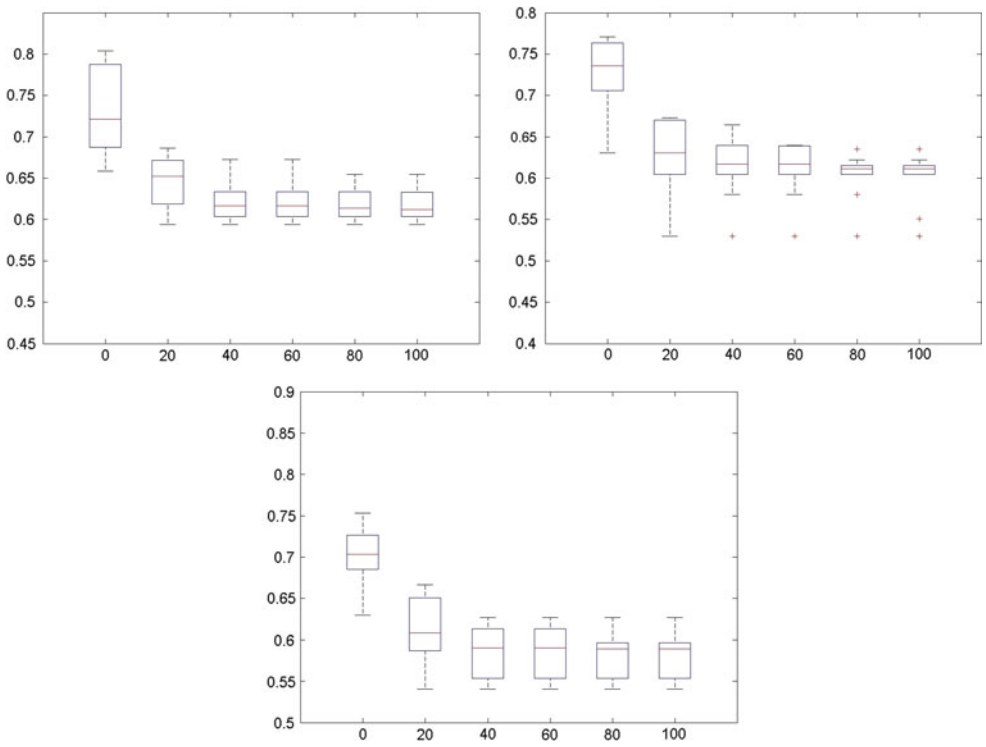

Fig. 7 Evolution of the MSE with the generations for the *FF data set* and the different options of parameter evolution. The *top-left*, *top-right* and *bottom* boxplots correspond with the ORI, the DFS and the DFDP options results

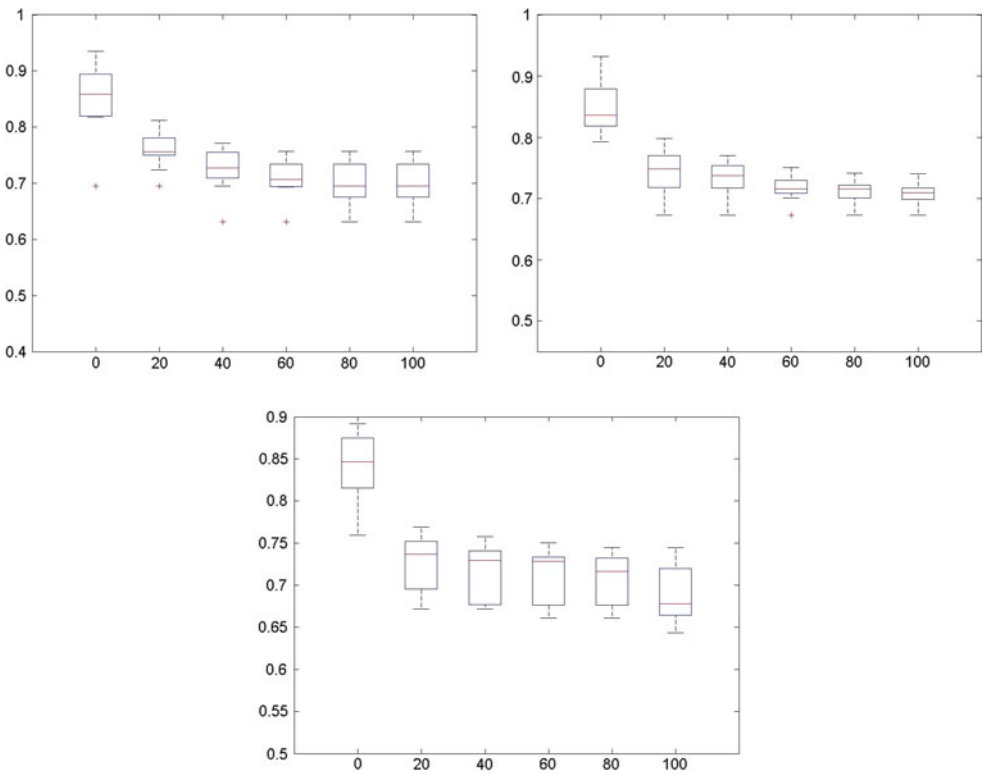

Fig. 8 Evolution of the MSE with the generations for the *WBC data set* and the different options of parameter evolution. The *top-left*, *top-right* and *bottom* boxplots correspond with the ORI, the DFS and the DFDP options results

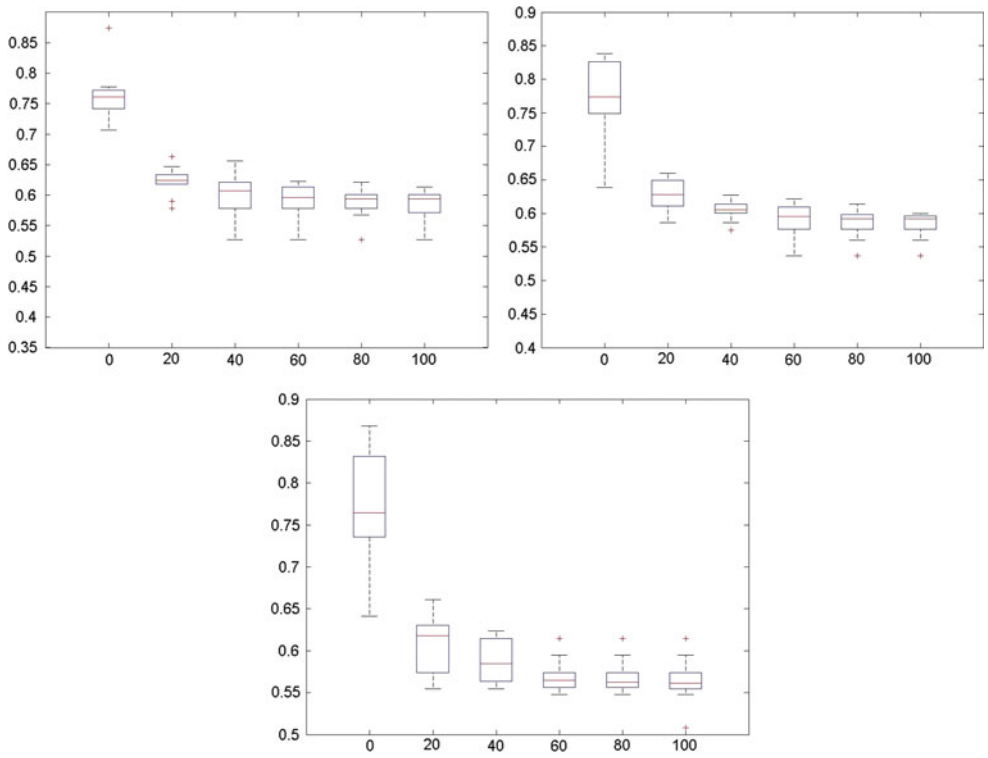

Fig. 9 Evolution of the MSE with the generations for the *CC data set* and the different options of parameter evolution. The *top-left*, *top-right* and *bottom* boxplots correspond with the ORI, the DFS and the DFDP options results

with a rather similar evolution and with the above mentioned MSE characteristics for each method.

Some conclusions are inferred after the analysis of the results. On the one hand, DFS obtained a smaller variability of the best individual MSE, which is highly desirable. Nevertheless, it cannot be said that DFS outperforms the ORIG approach. On the other hand, introducing simple heuristics as those used in DFDP enhances the results obtained: lower MSE values are achieved. Actually, the number of generations should be higher and the rules to calculate the parameter should be improved in order to reduce the variability among the different runs.

4.3 Comparison of different approaches for the real world process

Finally, the best set of options found in previous experimentation will be compared with the original approach results when facing the SSICS data set. The original approach makes use of 10-fold CV, with a fixed number of features to be chosen from and a fixed number of hidden neurons for the NN. The best set of options found so far comprises the use of 5×2 CV method and dynamic feature subset size including the adaptation of the neural network hidden neurons number.

The SSICS data set comprises 19 samples, each one with the whole set of parameter values. Once the piece is processed as the corresponding sample establishes, it is manually classified as {GOOD, BAD} according to the deformation or the quality faults that could appear in the piece. Besides, the maximum depth in each case is also measured. These two latter values are appended to each sample as the output variables. The data set is normalized with mean 0 and deviation 1.

The GA parameters have been fixed as follows: 50 individuals in the population, 100 of generations, the probability of crossover equals 0.75, while the mutation probability is 0.25. A steady state GA evolutionary scheme is used, with a number of 5 elite individuals that will be kept in the next generation.

For the ORIG, the 10-fold cross validation method with the size of the feature subset fixed to 3 has been used. Whenever fixed, 6 hidden neurons of the feed forward backpropagation NNs have been used. The rest of NN parameters are $\mu = 0.001$, $\mu_{dec} = 10^{-10}$, and $\mu_{inc} = 10^{-6}$, which have been kept constant during the feature selection and the model learning. This experiment has also been ran 10 times for statistical evaluation.

For the case of DFDP with 5×2 CV, the number of hidden neurons in the neural networks is fixed by means of Eq. 1. The individuals are allowed to evolve the size of the feature subset, though the maximum number of features to select is limited -as for the ORIG case, 3 features is the maximum feature subset size. The rest of NN parameters are $\mu = 0.001$,

Table 4 Comparison between the different methods for the real world problem: ORIG versus 5×2 DFDP

Method	Statistic	Maximum depth
10F+ORIG	Mean MSE	0.4514
	Median MSE	0.3971
	MSE deviation	0.1917
	Feature subset size	3 (10)
5×2+ORIG	Mean MSE	0.3513
	Median MSE	0.3418
	MSE deviation	0.0675
	Feature subset size	3 (10)
LOO+ORIG	Mean MSE	0.3067
	Median MSE	0.3060
	MSE deviation	0.0431
	Feature subset size	3 (10)
5×2 CV DFDP	Mean MSE	0.4208
	Median MSE	0.3376
	MSE deviation	0.2047
	Feature subset size	1 (10)

$\mu_{dec} = 10^{-10}$, and $\mu_{inc} = 10^{-6}$, which have been kept constant during the feature selection and the model learning. This experiment has also been ran 10 times for statistical evaluation.

The main statistics of the regression results from experimentation are shown in Table 4 and in the boxplots in Fig. 10. Clearly, LOO+ORIG outperforms the rest of the options for the real world problem. The 5×2 CV + DFDP is penalized for using a low value of hidden neurons -4 neurons against the 6 used in LOO+ORIG. Probably, this is the reason why the 5×2 CV + DFDP proposes individuals with only one feature in all the cases.

From the results, some conclusions are inferred. Firstly, the tuning of the parameters needs further study and no simple heuristics can be regarded as a general rule. Secondly, LOO could report even better results than 5×2 CV, although the time needed to obtain the models is somewhat increased. Finally, the type of problems for which the number of features is similar to the number of instances in the data set needs specific study in order to find the best model, and no generalization has been found yet.

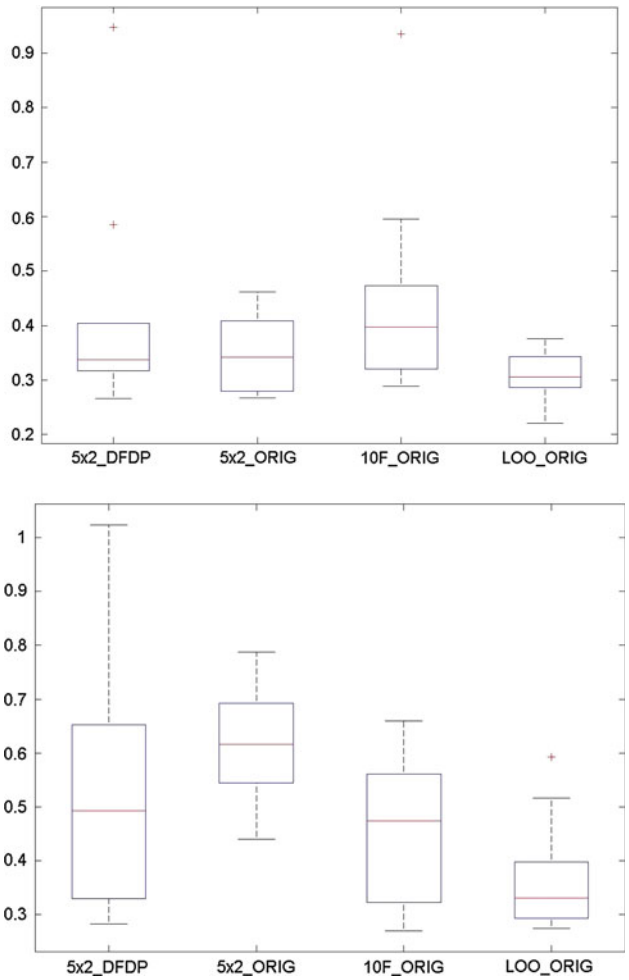

Fig. 10 Regression problem and NNs. Boxplot for the MSE values of the best individuals found for 5×2 DFDP, the 5×2, 10-fold and LOO CV for the original method. On *top*, the box plot with the test subsets; on *bottom*, the box plots with the whole data set (both train and test sets)

5 Conclusions

This study presents a feature selection method for choosing the best feature subset in steel sheets cold shaping process divided in a two-class problem and a maximum depth estimation problem. Moreover, a genetic algorithm is hybridized, on the one hand, for the first case, with a support vector machine model to choose the best feature subset and on the other hand, for the second case, with a feed forward back-propagation neural network.

Different method options have been analysed and several of them implemented and the results compared. From the experimentation it has been found that the 5×2 CV is the one with the best performance when facing problems with similar number of features and instances. Also, using simple heuristics for adapting and tuning the method parameters clearly improves the results obtained. Consequently, more complex heuristics or meta-heuristics are supposed to generate much better results, although the time spent in calculations will be much higher.

Finally, applying this method to the original real world problem we have found that the type of problems for which the number of features is similar to the number of instances

in the data set needs specific study in order to find the best model, and no generalization has been found yet. It has been found that LOO could report even better results than 5×2 CV, although the time needed to obtain the models is slightly bigger.

Acknowledgments This research has been funded by the Spanish Ministry of Science and Innovation, under projects TIN2008-06681-C06-04 and TIN2011-24302, the Spanish Ministry of Science and Innovation [PID 560300-2009-11], the Junta de Castilla y León [CCTT/10/BU/0002] and by the ITCL project CONSOCO.

References

1. Altun AA, Allahverdi N (2007) Neural network based recognition by using genetic algorithm for feature selection of enhanced fingerprints. In: ICANNGA '07 Proceedings of the 8th international conference on adaptive and natural computing algorithms, vol II, Springer, Berlin
2. Ariola A, Pahikkala T, Waegeman W, Baets BD, Salakoski T (2011) An experimental comparison of cross-validation techniques for estimating the area under the roc curve. Comput Stat Data Anal 55(4):1828–1844
3. Bustillo A, Sedano J, Curiel L, Villar J, Corchado E (2008) A soft computing system for modelling the manufacture of steel components. Adv Soft Comput 57(3):601–609
4. de la Cal E, Fernández EM, Quiroga R, Villar JR, Sedano J (2010) Scalability of a methodology for generating technical trading rules with gaps based on risk-return adjustment and incremental training. Lecture Notes Comput Sci 6077:143–150
5. Casillas J, Cordón O, del Jesus MJ, Herrera F (2001) Genetic feature selection in a fuzzy rule-based classification system learning process. Inf Sci 136(1–4):135–157
6. Cawley GC, Talbot NLC (2004) Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. Neural Netw 17(10):1467–1475
7. Celisse A, Robin S (2008) Nonparametric density estimation by exact leave-p-out cross-validation. Comput Stat Data Anal 52(5):2350–2368
8. Dong M, Wang N (2011) Adaptive network-based fuzzy inference system with leave-one-out cross-validation approach for prediction of surface roughness. Appl Math Model 35:1024–1035
9. Fung GM, Mangasarian OL (2004) A feature selection newton method for support vector machine classification. Comput Optim Appl 28(2):185–202
10. Hanczar B, Hua J, Sima C, Weinstein J, Bittner M, Dougherty ER (2010) Small-sample precision of roc-related estimates. Bioinformatics 26(6):822–830
11. Huanga CL, Wang CJ (2006) A ga-based feature selection and parameters optimizationfor support vector machines. Experts Syst Appl 31(2):231–240
12. MathWorks T (2012) MATLAB-The Language Of Technical Computing. http://www.mathworks.com/products/matlab/
13. Mohanty I, Datta S, Bhattacharjeeb D (2009) Composition-processing-property correlation of cold-rolled if steel sheets using neural network. Mater Manuf Process 24(1):100–105
14. Puigpinós L, Villar JR, Sedano J, Corchado E, de Ciurana J (2011) Steel sheet incremental cold shaping improvements using hybridized genetic algorithms with support vector machines and neural networks. In: Pelta DA, Krasnogor N, Dumitrescu D, Chira C, Lung RI (eds) Nature inspired cooperative strategies for optimization, NICSO 2011. Studies in computational intelligence, vol 387. Springer, Berlin, pp 323–332
15. Redondo R, Santos P, Bustillo A, Sedano J, Villar JR, Correa M, Alique JR, Corchado E (2008) A soft computing system to perform face milling operations. In: 4th international workshop on soft computing models in industrial applications. LNCS, vol 5518, pp 1282–1291
16. Sánchez L, Couso I, Casillas J (2009) Genetic learning of fuzzy rules based on low quality data. Fuzzy Sets Syst 160:2524–2552
17. Sedano J, Corchado E, Curiel L, Villar JR, Bravo P (2008) The application of a two-step ai model to an automated pneumatic drilling process. Int J Comput Math 86(10–11):1768–1777
18. Sedano J, Curiel L, Corchado E, de la Cal E, Villar JR (2009a) A soft computing based method for detecting lifetime building thermal insulation failures. Integr Comput Aided Eng 17(2):103–115
19. Sedano J, Villar JR, Curiel L, de la Cal E, Corchado E (2009b) Improving energy efficiency in buildings using machine intelligence. In: 10th international conference on intelligent data engineering and automated learning (IDEAL 2009). LNCS, vol 5788, pp 773–782
20. Villar JR, Suárez MR, Sedano J, Mateos F (2009) Unsupervised feature selection in high dimensional spaces and uncertainty. In: Corchado E, Wu X, Oja E, Herrero Á, Baruque B (eds) 4th International workshop on hybrid artificial intelligence systems, pp 563–572
21. Villar JR, Berzosa A, de la Cal E, Sedano J, García-Tamargo M (2012) Multi-objective simulated annealing in genetic algorithm and programming learning with low quality data. Neural Comput 75(1):219–225
22. Wong MLD, Nandi AK (2004) Automatic digital modulation recognition using artificial neural network and genetic algorithm. Singal Process 84(2):351–365
23. Yang J, Honavar V (1998) Feature subset selection using a genetic algorithm. IEEE Intell Syst 13(2):44–49
24. Yuan J, Liu X, Liu C (2012) Leave-one-out manifold regularization. Expert Syst Appl 39(5):5317–5324
25. Zhang P, Kumar BVK (2005) Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection. Pattern Recognit Lett 28(7):909–919
26. Zhao J, Wang F (2005) Parameter identification by neural network for intelligent deep drawing of axisymmetric workpieces. J Mater Process Technol 166(3):387–391
27. Zhao J, Cao HQ, Ma LX, Wang FQ, Li SB (2005) Study on intelligent control technology for the deep drawing of an axi-symmetric shell part. J Mater Process Technol 151(1–3):98–104