Co-evolutionary Genetic Multilayer Perceptron for Feature Selection and Model Design

Francisco Souza, Tiago Matias, and Rui Araújo
Institute of Systems and Robotics (ISR-UC), and
Department of Electrical and Computer Engineering (DEEC-UC),
University of Coimbra, Pólo II, PT-3030-290 Coimbra
fasouza@isr.uc.pt, tmatias@isr.uc.pt, rui@isr.uc.pt

Abstract

This paper proposes a method for Soft Sensors design using a Multilayer Perceptron model based on co-evolutionary genetic algorithms, called CEV-MLP. This method jointly and automatically selects the best input variables and the best configuration of the network for the prediction setting. The CEV-MLP is constituted by three levels, the first level selects the best input variables and respective delays set, the second level is composed by the parameters of hidden layers to be optimized (number of neurons in the hidden layers and transfer function), and the third level is the combination of first and second level. The method was successfully applied, and compared with two state-of-the-art methods, in three real datasets. In all the experiments, the proposed method shows the best approximation accuracy, while all the design of the prediction setting is performed automatically.

1 Introduction

Data-driven soft sensors (DDSS) are inferential models that use on-line available sensor measures, possibly complemented with measures obtained with laboratory analysis, for on-line estimation of variables which cannot be automatically measured at all, or can only be measured at high cost, sporadically, or with high delays (e.g. laboratory analysis) [4, 7, 16].

The selection of input variables and respective delays is essential to obtain an accurate and reliable reproduction of the target variable. If the network is trained with all available input variables in the dataset, it is being assumed that all features are good variables for prediction. However, this assumption is not valid when the data has irrelevant and/or redundant features. Moreover, the problem of selecting the most adequate delays for input variables remains. An MLP trained with irrelevant variables could be more flexible that without them and have better approximations in training set, but also will have a poor generalization performance [13]. Feature selection also decreases the training time of models and prevents network over-fitting [18]. In this work both steps of (1) selecting input variables and delays, and (2) selecting MLP network structure are solved using a multilayer perceptron model (MLP) by means of a co-evolutionary scheme. This approach of variable selection based on, and using, the learning model is called a wrapper approach [8].

The most common methods used to perform variable selection for MLP use a sequential backward search (SBS) or a sequential forward search (SFS) procedure and use the sum of squares error (SSE) as cost function. In [13] the variable selection algorithm using the traditional SBS was discussed, and the authors propose the retraining of the network when a feature is removed for evaluation. However, this method in some datasets becomes computationally expensive, because it is necessary retrain the network $\frac{n(n-1)}{2}$ times, where n is the input dimensionality.

In [17] a pruning algorithm for MLP networks is presented. In a pruning algorithm, the network is oversized and then the least significant hidden neurons and weights are pruned to find the smallest feasible size. In [17] it is proposed a sensitivity measure to verify the output sensitivity due to input perturbation, and a relevance measure to verify the relevance of neurons. Variable selection can be performed using both measures. In [5] it is proposed an evolutionary scheme for feature selection, called SAGA. This algorithm first uses simulated annealing to guide the global search in a solution space, and then uses a genetic algorithm to perform optimization. The main disadvantage of this method is that it does not take into account the optimization of the model, e.g. the number of the neurons in the hidden layer, when used in combination with a MLP model. In [14], a new cost function for simultaneous input variable and hidden node selection for an MLP model is proposed. This method penalizes the weights during fitting so that useless input variables can be excluded. The performance of the method depends on tuning the amount of penalization and the shape of the penalization function, i.e. the method is not fully automatic and depends on the dataset. In [10], a new fast model-based neural input selection method is pre-

sented. It is assumed that nonlinear models like polynomial NARX models, Volterra series and neural networks (NN) can achieve equivalent performance given that certain conditions are met. It is used a Volterra series model that is "linear-in-the-parameters", making it possible the use of existing model selection methods, e.g. orthogonal least-squares method. This method achieves significant reduction in the computational complexity but does not take care of the automatic design of the model used.

Genetic Algorithms (GA) have proved to be a useful tool to solve optimization problems. In [3], it is studied how to determine the optimum pipe size for networks used in natural gas applications. In [11] a Genetic Algorithm is used to maximize mutual information between the input and output variables, to prediction of oil flow. A multi-objective Genetic Algorithm (MOGA) based on the wrapper approach for feature selection is proposed in [9]. The MAGO is used to optimize a multi-objective problem, simultaneously minimizing the error rate and the model complexity. Compared with proposed algorithm, the main disadvantage is that this method does not optimize the hidden layer design, i.e. the number of hidden nodes and the activation functions. An approach using methods for nonlinear variable selection in conjunction with T-S fuzzy models was proposed by [2], for soft sensors applications. It uses T-S fuzzy models from available input/output data by means of a co-evolutionary genetic algorithm and a neuro-based technique. The soft sensor design is carried out in two steps. First, the input variables of the fuzzy model are pre-selected from the secondary variables of a dynamical process by means of correlation coefficients, Kohonen maps and Lipschitz quotients. Such selection procedure considers nonlinear relations among the input and output variables. Second, a hierarchical genetic algorithms is used to identify the fuzzy model itself. The input variable selection proposed by [2] has some shortcomings. First, the selection of the number of neurons in Kohonen maps is not automatically performed. Second, delays are not jointly selected with input variables, which can bring lower-accuracy results because a variable with the correct delay can contain more information about the output than a variable with the incorrect delay [16].

In this paper, a new co-evolutionary multilayer perceptron (CEV-MLP) method is proposed. Differently from previous approaches the method takes into consideration, and jointly selects, both the input variables and the configuration of an MLP prediction neural network. The CEV-MLP is constituted by three levels. The first level selects the best input variables and respective delays set. The second level is composed by, and selects, the parameters of hidden layers to be optimized (number of neurons in the hidden layers and transfer function). The third level is the combination of the first and second levels. The method was successfully applied, and compared with two state-of-the-art methods, in three real datasets, two publicly available datasets (Box Jenkins gas furnace, and Gas Mileage), and a dataset of a problem of flour concentration estimation in a real-world urban wastewater treatment plant (WWTP). The proposed CEV-MLP method exhibits the best prediction performance. This is achieved while the method automatically designs the prediction setting by jointly selecting input variables, the respective delays, and the MLP prediction model.

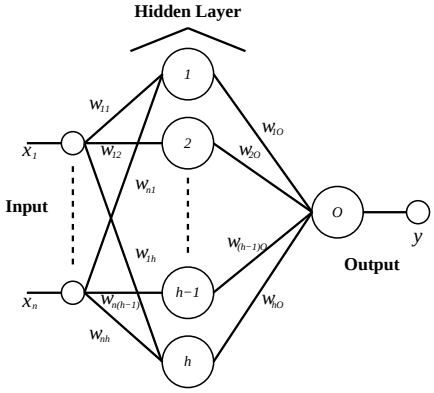

Figure 1: Topology of MLP-TL neural network; O is the output node, $\mathbf{W}_I = \mathbf{W} = [w_{ij}]$ is the $n \times h$ matrix of the weights connecting the inputs to the h hidden layer nodes, and $\mathbf{w}_O = [w_{1O}, \ldots, w_{hO}]^T$ is the output weight vector. The hidden layer biases $\mathbf{b}_I$ and the output bias b_O are omitted to simplify the diagram.

The paper is organized as follows. The MLP architecture is overviewed in Section 2. The new MLP variable selection algorithm proposed in this paper is presented in Section 3. Section 4 presents experimental results. Finally, Section 5 gives concluding remarks.

2 Multilayer Perceptron Architecture

An MLP neural network (NN) with two layers, that is used as the basis for the CEV-MLP, is represented in Figure 1. In [12, 6] it was shown that an MLP with one only hidden layer and a sufficient number of neurons can uniformly approximate any continuous function to any accuracy.

The MLP NN can be mathematically represented by:

$$y \;=\; g\left(f\left(\mathbf{x}^T\mathbf{W}_I + \mathbf{b}_I\right)\mathbf{w}_O + b_O\right), \qquad (1)$$

where $\mathbf{x} = [x_1, \ldots, x_n]^T$ is the input vector, y is the predicted output, $\mathbf{W}_I = \mathbf{W} = [w_{ij}]$ is the $n \times h$ matrix of the weights connecting the inputs to the h hidden layer nodes, $\mathbf{b}_I = \mathbf{b} = [b_1, \ldots, b_h]$ is the vector of biases of the hidden layer nodes. The output weights that connect the hidden neurons with the output neuron and the output bias are represented by $\mathbf{w}_O = [w_{1O}, \ldots, w_{hO}]^T$ and b_O, respectively. $f(\cdot)$ and $g(\cdot)$ represent the activation functions of the nodes of the hidden layer, and output layer, respectively. The network is trained by minimizing the mean square error (MSE) of all network output samples:

$$E_{mse}(y, y_d) \;=\; \frac{1}{L}\sum_{k=1}^{L}\left[y(k) - y_d(k)\right]^2,$$

where, $y(k)$ and $y_d(k)$ are the predicted and desired output of k-th input data sample, and L is the number of exemplars. In this work $f(\cdot)$ can be a tangent sigmoid or a linear function and $g(\cdot)$ is a linear function. To perform the MLP NN design there are several parameters to be considered: the types of the activation function, the number of neurons in the hidden layer and the best subset of input variables and respective delays.

3 CEV-MLP

The objective of the CEV-MLP is to optimize the final prediction model by jointly selecting the appropriate MLP architecture and the best subset of input variables and respective time delays. The optimization is performed by means of genetic algorithms. The CEV-MLP is constituted by three hierarchical levels (Fig. 2). The first level is constituted by the possible sets of input variables and respective delays, the second level is constituted by the possible set of hidden layer configurations and the third level represents the combination of the first and second level, i.e. the final model.

3.1 Hierarchical Architecture

Fig. 2 shows the detailed scheme of the CEV-MLP, it is constituted by 3 levels. The detailed description of each level is given below:

First Level is constituted by the possible sets of input variables and respective time delays that will be used to design the DDSS. The chromosome is represented by a binary encoding, where each allele (element of the chromosome that is located at a specific position) corresponds to each input variable and respective delay (see Fig. 2). One zero in one allele indicates that the input associated to this allele is not considered.

Second Level is constituted by the possible sets of hidden layer configurations. Each allele of the chromosome can get values from zero to two. The zero value indicates that the neuron of the hidden layer associated to this allele will be pruned, i.e. the number of the neurons is given by the number of alleles different from zero. If the allele gets the value one or two, this means that the activation function of the neuron of the hidden layer associated to this allele will be tangent sigmoid or linear, respectively.

Third Level is constituted by the possible sets of MLP configurations. Each chromosome is constituted by two alleles that can get a non-negative integer, being that the first allele represents the individual of level 1 and the second allele represents the set of level 2.

A predictor at the third level is denoted by $S_3^{(m,l)} = C(S_2^m, S_1^l)$, a combination of Levels 1 and 2, where S_2^m, S_1^l are the m-th and l-th, chromosomes of Levels 2 and 1, respectively, and C is an operator that generates a MLP model using the S_2^m and S_1^l chromosomes. The predicted output generated by $S_3^{(m,l)}$ is given by $y^{(m,l)}$.

input: Maximum number of chromosomes for each level, $l_{max}, m_{max}, k_{max}$
output: A optimal MLP model with best variables and configuration

$i \leftarrow 1$;
$int \leftarrow 1$;
while $int < N_{max}$ **do**
  forall $k = 1, \ldots, k_{max}$ **do**
    Evaluate $J_3^k(i)$ using Equ. (2a);
  end
  forall $m = 1, \ldots, m_{max}$ **do**
    Evaluate $J_2^m(i)$ using Equ. (2b);
  end
  forall $l = 1, \ldots, l_{max}$ **do**
    Evaluate $J_1^l(i)$ using Equ. (2c);
  end
  Select the two best chromosomes on Level 1*, Level* 2 *and Level* 3*, according with* J_1^l*,* J_2^m *and* J_3^k *to be parents. For a fast convergence all the other inputs are removed and reproduction is performed to obtain new ones. Perform mutation in all new children for all levels;*
  if $J_3^k(i) == J_3^k(i-1)$ **then**
    $int \leftarrow int + 1$;
  else
    $int \leftarrow 1$;
  end
  $i \leftarrow i + 1$;
end

Algorithm 1: Steps of CEV-MLP algorithm

The cost function for the k-th chromosome of the third level $J_3^k = J_3^{(m,l)}$, which is given by the mean square error between the predicted and real outputs in the training dataset, and cost functions for Levels 1 and 2 are, respectively, given by:

$$J_3^k = J_3^{(m,l)} = E_{mse}\left(y^{(m,l)}, y_d\right), \forall k, \quad (2a)$$

$$J_2^m = \min\left(J_3^{(m,1)}, J_3^{(m,2)}, \ldots, J_3^{(m,l_{max})}\right), \forall k, \quad (2b)$$

$$J_1^l = \min\left(J_3^{(1,l)}, J_3^{(2,l)}, \ldots, J_3^{(m_{max},l)}\right), \forall k. \quad (2c)$$

k_{max}, l_{max}, m_{max}, are the maximum number of chromosomes at Levels 3, 2, and 1, respectively. It is important note that $k_{max} \leq m_{max} \cdot l_{max}$, because the number of chromosomes of Level 3, is always lower than $m_{max} \cdot l_{max}$.

An example of the encoding and the hierarchical relations is given in Fig. 2. In this example, the first allele of the k-th set of Level 3 indicates that the set of input variables and delays for the network will be the 6th set represented at Level 1, while the second allele indicates that the configuration of the hidden layer of the MLP will be the

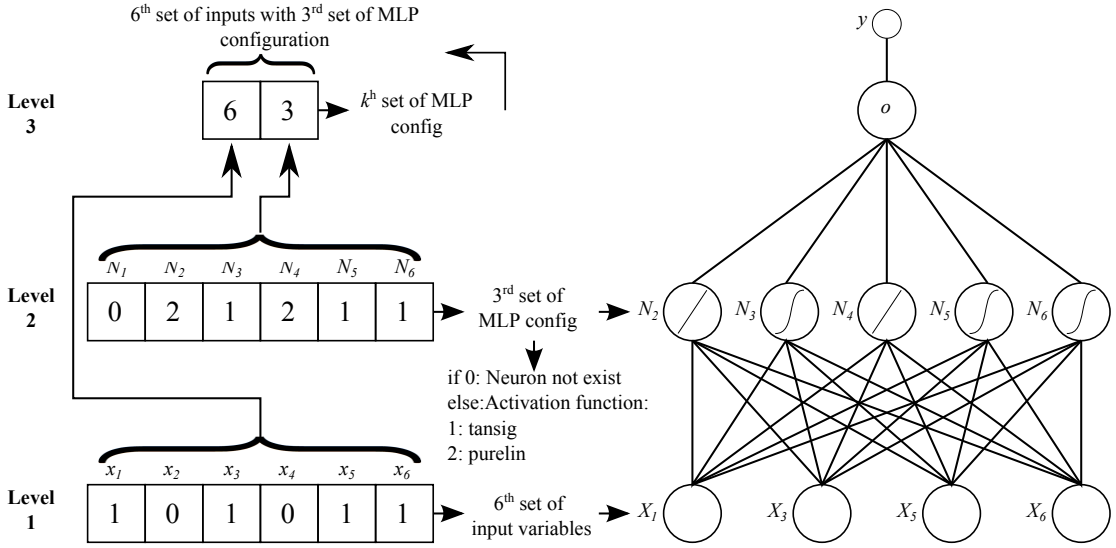

Figure 2: Graphical Representation Scheme of CEV-MLP algorithm.

one described in the 3rd chromosome of Level 2. The first allele of the chromosome of level 2 is zero. So the number of neurons in hidden layer will be five, the size of the chromosome (6) minus the number of zero valued alleles (1). The neurons associated to the 2nd and 4th allele will have a linear activation function and the remaining will have tangent sigmoid transfer function. Level 1, specifies that the inputs of the network will be the variables x_1, x_3, x_5 and x_6.

The CEV-MLP is specified in Algorithm 1. The first step of the algorithm is the random initialization of the populations of all levels. After this, while N_{max} is less than int, i.e. while the method does not reach the maximum number the of iterations with the same error: for all chromosomes of Level 3, the performance of the network is computed; for all levels the best two chromosomes are selected to be the parents and the other are removed; new ones are obtained by reproduction and mutation.

3.2 Genetic Algorithm Operators

The methods used for initialization, selection, reproduction and mutation in the genetic algorithm of the CEV-MLP approach are described in this section.

Initialization: The initial population is chosen randomly with uniform distribution. Is know that random initialization can affect convergence time, but good results have been obtained in CEV-MLP.

Selection: The selection method used was elitist selection [15]. This method selects the best chromosomes to be the parents, the remaining are removed and new chromosomes are generated, in this work the best chromosomes considered are the first two. A random number between 0 and 1, R_s, is generated, if $R_s < 0.5$ the first individual is used as a father and the second as a mother, and if the $R_s \geq 0.5$ the second individual is the father and the first is the mother.

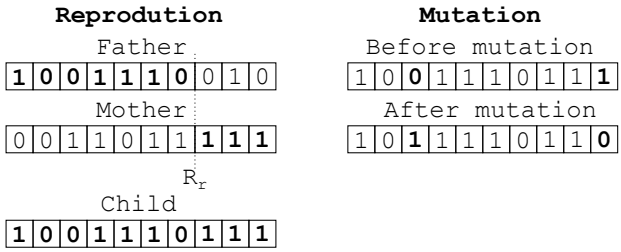

Figure 3: Genetic operators: Single-point Reproduction and Mutation of two alleles.

Reproduction: For reproduction the single point crossover technique was used. The process consists of taking two parents and produce a child [15]. As described above, all chromosomes except the parents are removed and new ones are obtained by the crossover of the two selected parents. For each child, the crossover process generates a random point of crossover, R_r, and the child will receive the alleles from 1 to R_r from the father and the rest of the alleles are received from the mother. This process is illustrated in Fig. 3.

Mutation: Mutation of two alleles was the third operation used. This is used to maintain the diversity of the population and to prevent the algorithm from being trapped in a local minimum. The mutation is an operator that alters the value of one or more randomly selected alleles in a chromosome. To perform crossover and mutation in the second and the third level, the chromosomes are converted in binary encoding. Mutation is also illustrated in Fig. 3.

4 Experiments and Results

This section presents experimental results in three distinct datasets, verifying the performance and demonstrat-

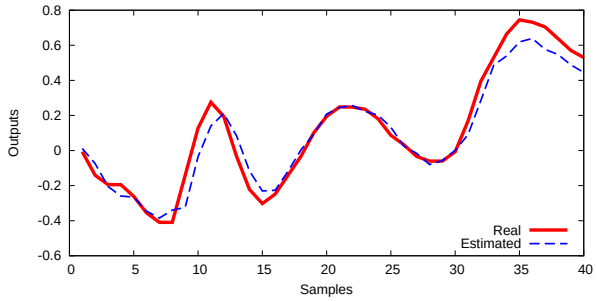

Figure 4: Predicted and target outputs using CEV-MLP for the Box and Jenkins furnace gas test dataset.

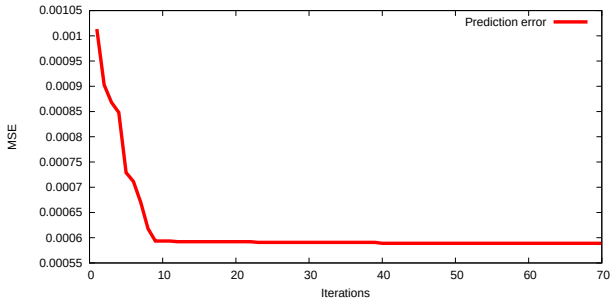

Figure 5: MSE of the network validation of the proposed algorithm for the Box and Jenkins furnace gas dataset.

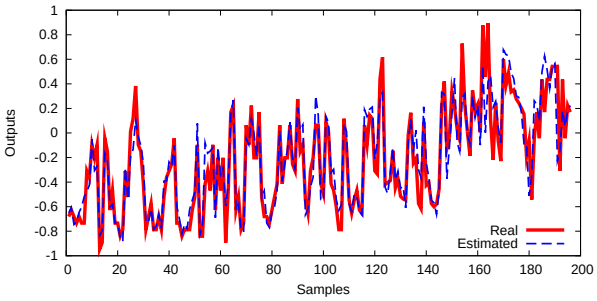

Figure 6: Predicted and target outputs using proposed algorithm for the Automobile MPG dataset.

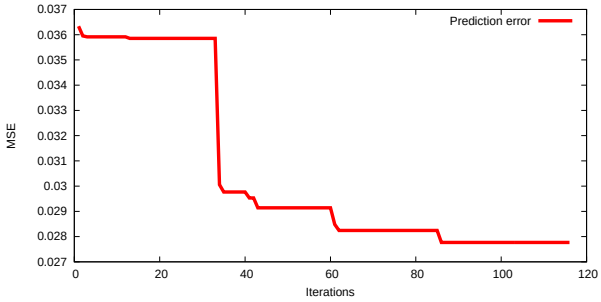

Figure 7: MSE of the network validation of the proposed algorithm for the Automobile MPG dataset.

ing the effectiveness of the proposed methods. The approximation performance of the soft sensors is evaluated using the mean square error (MSE) and the correlation coefficient between predicted and desired output, in the validation and test data. For all experiments the reproduction and mutation probabilities are 80% and 10%, respectively, and the number of chromosomes for each level are: $k_{max} = 20$, $m_{max} = 200$, $l_{max} = 200$ and $N_{max} = 30$. The datasets were divided in training and test data and, in turn the training data was randomly divided in 75% for training and 25% for validation. The proposed method is compared with (i) the traditional SBS method using MLP, and (ii) the method proposed in [10].

4.1 Box and Jenkins Dataset

The Box-Jenkins gas furnace process data[1] was recorded from a combustion process of a methane-air mixture, and consists of 296 data points $[y(t), u(t)]$ [1]. The input $u(t)$ is the gas flow rate into the furnace and the output $y(t)$ is the carbon dioxide (CO_2) concentration in the outlet gas. The sampling interval is 9 [s]. To predict $y(t)$ the following set with all variables and possible considered delays is examined $X^{(t)} = \{y(t-1), y(t-2), y(t-3), y(t-4), u(t-1), u(t-2), u(t-3), u(t-4), u(t-5), u(t-6)\}$. The first 250 samples were used for training and the remaining for test/evaluation. The maximum number of neurons in the hidden layer was limited to three.

The results of the application of the three methods is presented in Table 1. Fig. 4 shows the predicted and the target outputs for the test dataset, and Fig. 5 shows the prediction error during the CEV-MLP operation, as can be seen the error stabilizes at iteration 40.

The proposed CEV-MLP chooses more pairs of input variables and delays than the traditional SBS and method of [10], but the mean square error and correlation coefficient has similar values. CEV-MLP attains these results while exhibiting the advantages discussed in Sec. 1, namely automatically selecting both the input variables and respective delays, as well as the MLP structure. The selected configuration for the MLP model is composed by two neurons in the hidden layer and both activation functions are tangent sigmoid.

4.2 Automobile MPG Dataset

The automobile gas mileage dataset corresponds to a problem of predicting the number of miles per gallon (MPG). It is a six input, single output regression problem. The gasoline consumption needs to be predicted based on some input variables. These variables are the number of cylinders, displacement, horsepower, weight, acceleration and model year. The original data is available in the UCI (Univ. of California at Irvine) Machine Learning Repository[2]. The input set considered is $X^{(t)} = \{u_1(t), u_2(t), u_3(t), u_4(t), u_5(t), u_6(t)\}$. Where u_1 is the number of cylinders, u_2 the displacement, u_3 the horsepower, u_4 the weight, u_5 the acceleration, and u_6 is the year. The train and test dataset are composed by 196 samples each. The maximum number of neurons to be se-

[1] Provided by IEEE Neural Networks Council Standards Committee Working Group on Data Modeling Benchmarks. Available: http://www.stat.wisc.edu/~reinsel/bjr-data/gas-furnace .

[2] Available: http://archive.ics.uci.edu/ml/datasets/Auto+MPG .

Table 1: Performance Results for the Box and Jenkins dataset

Method	Selected Inputs	MSE Test	Correlation Test
SBS	$y(t-1), y(t-2), u(t-3)$	$5,94e-3$	$\mathbf{0,981}$
Li & Peng [10]	$y(t-1), y(t-2), ut-3)$	$5,75e-3$	$\mathbf{0,981}$
CEV-MLP	$y(t-1), y(t-2), u(t-3), u(t-5), u(t-6)$	$\mathbf{5,69e-3}$	$0,980$

Table 2: Performance Results for the Gas Mileage dataset

Method	Selected Inputs	MSE Test	Correlation Test
SBS	$u_2(t), u_6(t)$	$3,30e-2$	$0,899$
Li & Peng [10]	$u_1(t), u_2(t), u_3(t), u_6(t)$	$2,70e-2$	$0,918$
CEV-MLP	$u_2(t), u_4(t), u_5(t), u_6(t)$	$\mathbf{2,43e-2}$	$\mathbf{0,927}$

lected by the CEV-MLP is three.

The results are presented in Table 2. Fig. 6 shows the predicted and the target outputs and Fig. 7 shows the prediction error of the network while the proposed method performs the optimization of the prediction setting (variables, delays, and network structure).

The CEV-MLP method chooses the most adequate subset of input variables and respective delays when compared with the SBS method and the method proposed by [10]: CEV-MLP has better results in terms of error and correlation coefficient. The MLP model selected by the CEV-MLP is composed by tree neurons in the hidden layer with tangent sigmoidal transfer function.

4.3 WWTP Dataset

In the third experiment the objective is to estimate the flour concentration in the effluent of a real-world urban wastewater treatment plant (WWTP). The dataset of plant variables that is available for learning consists of 11 input variables, $U^{(t)} = \{u_1(t), \ldots, u_{11}(t)\}$, and one target output variable to be estimated, y. The variables correspond to physical values, such as pH, turbidity, color of the water and others, see Table 3 for further details. The possible input variables with respective delays used as input for the optimization problem are: $X^{(t)} = \{u_1(t), u_1(t-1), u_1(t-2), \ldots, u_{11}(t), u_{11}(t-1), u_{11}(t-2)\}$. The input variables are measured online by plant sensors, and the output variable in the dataset is measured by laboratory analysis. The sampling interval is 2 [hours]. The proposed algorithm jointly selects the best variables and delays, as well as the MLP structure, for the flour prediction setting. The train and test dataset are composed by 196 samples.

The results of the application of the three methods (SBS-MLP, [10], CEV-MLP) are presented in Table 4. Fig. 8 shows the predicted and the target outputs, and Fig. 9 shows the prediction error of the test set while the CEV-MLP performs the network optimization and selection of best input variables and delays. The proposed algorithm has chosen less input (variable, delay) pairs than SBS and the method of [10], and the MSE and correlation values

Table 3: Variables of the wastewater treatment plant dataset

Variables	Description
u_1	Amount of chlorine in the influent;
u_2	Amount of chlorine in the effluent;
u_3	Turbidity in the raw water;
u_4	Turbidity in the influent;
u_5	Turbidity in the effluent;
u_6	Ph in the raw water;
u_7	Ph in the influent;
u_8	Ph in the effluent;
u_9	Color in the raw water;
u_{10}	Color in the influent;
u_{11}	Color in the effluent;
y	Flour in the effluent.

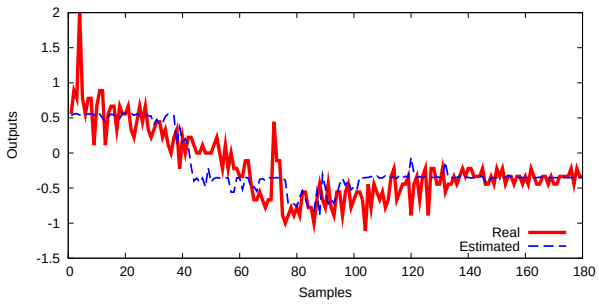

Figure 8: Predicted and target outputs using proposed algorithm for the WWTP dataset.

between the target and predicted outputs show that the proposed CEV-MLP method has better prediction performance results. The selected MLP is composed by eight neurons in the hidden layer, where three of them are linear and the remaining are tangent sigmoidal.

5 Conclusion

The paper proposed a new method for jointly selecting input variables and corresponding delays, as well as constructing the structure of an MLP prediction model. An evolutionary scheme using genetic algorithms selects the best set of input (variable, delay) pairs and the best MLP model, making it suitable for Soft Sensor applica-

Table 4: Performance Results for the WWTP dataset.

Method	Selected Inputs	MSE Test	Correlation Test
SBS	$u_9(t-4)$, $u_{10}(t-4)$, $u_1(t-2)$, $u_3(t-2)$, $u_4(t-2)$, $u_5(t-2)$, $u_6(t-2)$, $u_1(t)$, $u_3(t)$, $u_4(t)$, $u_6(t)$, $u_7(t)$, $u_9(t)$	$8,05e-2$	$0,815$
Li & Peng [10]	$u_1(t-4)$, $u_3(t-4)$, $u_5(t-4)$, $u_7(t-4)$, $u_9(t-4)$, $u_{10}(t-4)$, $u_{11}(t-4)$, $u_3(t-2)$, $u_7(t-2)$, $u_8(t-2)$, $u_1(t)$, $u_5(t)$, $u_7(t)$, $u_8(t)$, $u_9(t)$, $u_{10}(t)$, $u_{11}(t)$	$8,56e-2$	$0,804$
CEV-MLP	$u_4(t-4)$, $u_1(t-2)$, $u_4(t-2)$, $u_8(t-2)$, $u_4(t)$	$\mathbf{6,87e-2}$	$\mathbf{0,844}$

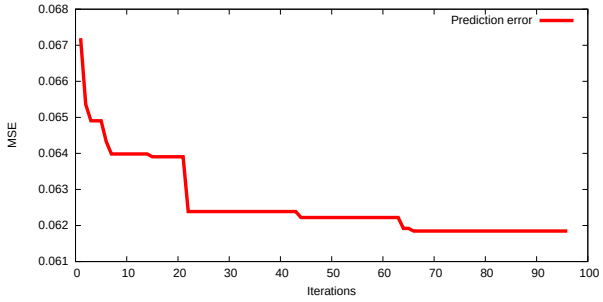

Figure 9: MSE of the network validation of the proposed algorithm for the WWTP dataset.

tions. The proposed method does not require any prior knowledge concerning of the model and about the best input variables; Only the empirical input-output data is required.

To validate and demonstrate the performance and effectiveness of the proposed methodology, the algorithm was applied on three prediction problems with real-world datasets, and compared with two state-of-art methods. The experimental results have shown the effectiveness of the proposed method. The proposed CEV-MLP method exhibits the best prediction performance, while automatically designing the prediction setting (jointly selecting inputs, delays, and prediction model).

Future work will implement different optimization methods, and incorporate more criteria to be optimized.

Acknowledgment

This work was supported by Mais Centro Operacional Program, financed by European Regional Development Fund (ERDF), and Agência de Inovação (AdI) under Project SInCACI/3120/2009.

Francisco Souza has been supported by Fundação para a Ciência e a Tecnologia (FCT) under grant SFRH/BD/63454/2009.

References

[1] G. E. P. Box and G. M. Jenkins. Time series analysis. *Cambridge Univ. Press*, 2003.

[2] M. R. Delgado, E. Y. Nagai, and L. V. R. de Arruda. A neuro-coevolutionary genetic fuzzy system to design soft sensors. *Soft Computing*, 13(5):481–495, 2008.

[3] O. F. M. El-Mahdy. M. E. H. Ahmed, and S. Metwalli. Computer aided optimization of natural gas pipe networks using genetic algorithm. *Applied Soft Computing*, pages 1141–1150, 2010.

[4] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia. *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer, 2007.

[5] I. A. Gheyas and L. S. Smith. Feature subset selection in large dimensionality domains. *Pattern Recognition*, 43(1):5–13, 2010.

[6] K. M. Hornik, M. Stinchcombea, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[7] P. Kadlec, B. Gabrys, and S. Strandt. Data-driven soft sensors in the process industry. *Computers & Chemical Engineering*, 2009. 33(4):795 - 814.

[8] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence Archive*, 97(1-2):273–324, December 1997.

[9] H. C. Lac and D. A. Stacey. Feature subset selection via multi-objective genetic algorithm. In *International Joint Conference on In Neural Networks*, volume 3, pages 1349–1354, 2005.

[10] K. Li and J.-X. Peng. Neural input selection-a fast model-based approach. *Neurocomputing*, 70(4-6):762–769, 2007.

[11] O. Ludwig, U. Nunes, R. Araújo, L. Schnitman, and H. A. Lepikson. Applications of information theory, genetic algorithms, and neural models to predict oil flow. *Communications in Nonlinear Science and Numerical Simulation*, 17(7):2870–2885, 2009.

[12] R. H. Nielsen. Theory of the back propagation neural network. In *International Joint Conference on In Neural Networks*, pages 593–605, June 1989.

[13] E. Romero and J. M. Sopena. Performing feature selection with multilayer perceptrons. *IEEE Transations on neural networks*, 19(3), March 2008.

[14] T. Similä and J. Tikka. Combined input variable selection and model complexity control for nonlinear regression. *Pattern Recognition Letters*, 30(3):231–236, 2009.

[15] S. N. Sivanandam and S. N. Deepa. *Introduction to Genetic Algorithms*. Springer, Berlin, Germany, 2008.

[16] F. Souza, P. Santos, and R. Araújo. Variable and delay selection using neural networks and mutual information for data-driven soft sensors. In *Proc. 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2010)*, pages 1–8, September 2010.

[17] X. Zeng and D. S. Yeung. Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure. *Neurocomputing*, 69(7-9):825–837, 2006.

[18] G. P. Zhang. Avoiding pitfalls in neural network research. *IEEE Transactions on systems man and cybernetic Part C applications and reviews*, 37(1):3–16, 2007.