Adaptive Neuro - Fuzzy Inference Systems – An Alternative Forecasting Tool for Prosumers

Otilia DRAGOMIR[1], Florin DRAGOMIR[1], Veronica ȘTEFAN[2], Eugenia MINCĂ[1]

[1] Automation, Computer Science and Electrical Engineering Department,
Valahia University of Târgoviste,
2, Carol I Blvd., Târgoviște, 130024, Romania,
drg_otilia@yahoo.com, drg_florin@yahoo.com, minca.eugenia@gmail.com

[2] Faculty of Economic Sciences, Valahia University of Targoviste,
2 Carol I Bd., Târgoviște, 130024, Romania,
veronica.stefan@ats.com.ro

Abstract: The goal of this paper is to propose a forecasting tool to producers/ consumers (prosumers) of renewable energy sources, based on artificial intelligence techniques, trying to obtain optimal predictions. The exploration and the assessment of the criteria used for choosing the adequate forecasting tool are made in the artificial intelligence context. In this respect, firstly, the criteria used for choosing the best forecasting technology, in relation to each step of the modelling process are presented. Secondly, the identified criteria are tested on two Adaptive Neuro- Fuzzy Inference System (ANFIS) models, in order to underline the effects of these users' decisions over the forecasting performances.

Keywords: Forecasting, Neural Network, Adaptive Neuro - Fuzzy Inference Systems, Prosumers

1. Introduction

The successful entry on energy market, recently liberalized in Romania, depends on the capacity of each producer to predict through different methods the future evolution of the quantity of renewable energy sources. Thus, the integration of the predictive ability is a very important function in the command and control of a power system, especially at the micro-grid level, heavily dependent on the consumer and with reduced availability for storage.

European context. The current energy policy of the European Union (EU) considers the security of supply, competitiveness and sustainability as central targets. In order to achieve these targets, a series of constraints ("20-20-20 objective") is imposed through European strategies [1]: 20% reduction in emissions of greenhouse gases compared to 1990, providing 20% of entire EU energy consumption by Renewable Energy Sources (RES) and a 20% reduction in energy use in comparison with a similar scenario in which no action regarding sustainability has been taken. In order to achieve these objectives and generate a "sustainable growth", a policy of encouraging distributed generation from RES, such as solar power must be followed. Intense concerns at European level regarding the Distributed Power from RES (DP- RES) were materialized by setting up a giant cluster of projects called Integration of Renewable Energy Sources and Distributed Generation into the European Electricity Grid, IRED cluster [2]. The studies that were conducted by this consortium highlighted the need for an energy management system at micro to macro level, the existing control strategies not being always successfully applied.

National context. The percentage of RES in the electricity production in Romania is currently of 17.8%. EU has set a 24% target for Romania for energy generation from RES by 2020, but needs for investments and large operating costs as main barriers for the successful implementation of an increased generating capacity have also been identified. The compatibility with the EU objectives in the field of clean energy and national levels is achieved through a regional policy [3]. The European policies have had a national resonance since 2003, when the draft for the Strategy project for the use of renewable energy [4] was proposed.

Two new important trends on the national energy market are to be noticed: firstly, the consumers' involvement in the complex process of the energy efficient management from RES, and secondly, the increased attention paid to both the technical plan and the organizational and economical plan for the energy production from RES. The integration of RES in the national energy system structure has created a favorable context for our researches.

Unfortunately, Romania follows a centralized approach of the regional policy and although the country is covered adequately with electricity networks and the potential development for RES is high, its aging infrastructure (30% of it was built in the 1960s) causes significant losses along the energy supply chain.

Moreover, for a large number of energy resources, the current energy systems are hardly scalable. The European Commission believes that the current energy infrastructure is inadequate to connect and serve the entire Europe and recognizes the challenges [1].

In these contexts, two challenges have to be met: monitoring, diagnosis and forecasting functioning of grids, integrating renewable energy sources in on- grid or off- grid mode and ensuring the optimization of these ones functioning with the integrated, proactive management system based on multi-objective decision-making scheme.

To support these interests, we consider that the integration of the predictive ability is a very important function in the command and control of a power system, especially at the micro- grid level, heavily dependent on the consumer and with reduced availability for storage.

The challenge and the value added of our paper consist in proposing a forecasting tool to producers/ consumers (prosumers) of renewable energy resources, based on artificial intelligence techniques, trying to obtain optimal forecasts.

The paper is organized in two sections:
- first of all, there are recessed the most used criteria for choosing an optimal forecasting architecture, in relation with each step of the forecasting tool modeling: data preprocessing, forecasting tool architecture identification, parametrization and implementation;
- secondly, the identified criteria are tested on two Adaptive Neuro- Fuzzy Inference System (ANFIS) models, in order to underline the effects of these users' decisions over the forecasting performances.

2. Identification of the Criteria Used for Choosing a Forecasting Tool

Our researches were oriented on identifying the most used criteria for choosing an optimal forecasting tool based on artificial intelligence techniques.

The approach was difficult because of the high numbers of variables and constraints that conditioned the parameterization of this kind of tool. The efficiency of using Neural Networks (NN) in the area of energy forecasting was demonstrated in our previous works [5 - 7]. Thus, we oriented our work to NN.

The parameters conditioning the work with these ones are: number of hidden layers, number of neurons from the NNs input, output and hidden layers, the testing/ validation data sets dimension, the learning algorithm to choose and the method used to measure the forecasting made by the tool.

2.1 Data preprocessing

Input selection is a method used often in the data preprocessing. It is an experimental method that supposes to find the most important input variables among a large number of them [8].

Some authors [9] assume that the input information can be classified into groups (hierarchical structure). To realize such a grouping, there is no general automatic approach but heuristics based on fusion of physical sensors.

Both input selection and hierarchical structure presume that inputs are independent and give no priority of the selected input variables.

When the analyzed process is complex, it is recommended to normalize the input and output real values into the interval between max and min of the transformation function, usually [0, 1] or [-1, 1] intervals. The most popular methods are the following:

$$SV = (0.9 - 0.1) / (MAX\_VAL - MIN\_VAL) * (OV - MIN\_VAL) \quad (1)$$

or:

$$\mathrm{SV} = TF_{min} + \frac{(TF_{max} - TF_{min})}{(MAX\_VAL - MIN\_VAL)} * (OV - MIN\_VAL) \quad (2)$$

where:

- SV: Scaled Value,
- MAX_VAL: maximum value of data,
- MIN_VAL: minimum value of data,
- TFmax: maximum of transformation
- TFmin: minimum of transformation
- OV: Original Value

2.2 Identifying the NN's architecture

There are many techniques for determining the forecasting tool architecture. In this section, we will cover some of the general "rules of thumb" that can be used. In nearly all cases, some additional experimentation will be required to determine the optimal structure.

Every input neuron from the input layer of the NN should represent some independent variable that has an influence over the output.

If a pattern is presented to the input layer of NN, then the output layer will generate another pattern. The output layer of the NN presents a pattern to the external environment [10]. Whatever pattern is represented by the output layer, this can be directly traced back to the input layer.

The number of output neurons is related to the type of work that the NN has to perform. If it is to be used to classify items into groups, then it is often preferable to have one output neuron for each group that the item is to be assigned into. If NN is to perform noise reduction on a signal then it is likely that the number of input neurons will match the number of output neurons.

One common mistake made by users is to add a large number of variables and neurons, without taking into account the number of parameters to be estimated [11].

The hidden layers are those that don't interact directly with the external environment. For many practical problems, there is no reason to use more than one hidden layer. The NN with one hidden layer can approximate well any continuous mapping, from one compact space to another. NNs without hidden layers are only capable of representing linear separable functions or decisions. Problems that require two hidden layers are rarely encountered. NN with two hidden layers can represent functions with any kind of shape. There is currently no theoretical reason to use neural networks with any more than two hidden layers. [10]

When the number of neurons in the hidden layers isn't correctly approximate, two problems may occur: under-fitting or over-fitting. Under fitting is due to underestimation of NN neurons of hidden layers. In this case the input signal isn't adequately detected. On the other side, using too many neurons generate over-fitting problems. The NN has too much information processing capacity and the limited amount of information contained in the training set is not enough to train all of the neurons in the hidden layers.

The scientists [11] propose as starting points to consider three rules: the number of hidden neurons should be in the range between the size of the input layer and the size of the output layer; the number of hidden neurons should be 2/3 of the input layer size, plus the size of the output layer; the number of hidden neurons should be less than twice the input layer size.

One additional method that can be used to reduce the number of hidden neurons is called pruning. It involves evaluating the weighted connections between the layers and if the network contains connections with weights equal with zero, they can be removed.

2.3 Parametrization of NN

The test samples must be appropriately selected. Since NNs are "data-driven" methods, they typically require large samples in testing. The input selection reduces the testing data dimension. The NN are tested with small subsets that include data from only a few past days, selected through statistical measures of similarity. This thing results in samples that are very homogeneous, but also very small [11].

Usually, the dimension of the test data set must be five times greater than the number of update parameters. The testing process is usually finished as soon as the testing error is reduced to a specified tolerance level (e.g., 10^{-5}). The adequate rate between the number of testing samples and the number of weights in the network has not been clearly defined yet [12].

The testing algorithm has to be also established, the most common in use is the back propagation algorithm. The purpose of the backpropagation testing is to converge to a near-optimal solution based on the total squared error calculated.

K-nearest-neighbor learning algorithm is simpler than the backpropagation algorithm because there is no model to test on the data series. Instead, the data series is searched for situations similar to the current one, each time a forecast needs to be made [13] and [14]. In most of the papers reviewed, testing was stopped after a fixed number of iterations or after the error decreased below some specified tolerances.

2.4 Assessment of the performance

The next stage in modeling is the implementation of the NNs, which means the estimation of its parameters. The guidelines proposed in [15], for evaluating effectiveness of implementation, were based on the next question: was the NN properly tested so that its performance was the best it could possibly achieve? The choice of error measures to help comparing forecasting methods has been much discussed, as a consequence of the many competitions that were started in the 1980's [16]. J.G. De Gooijer and R.J. Hyndman [17] summarized in Table 1 the most used performance indicators.

Table 1. The most used performance indicators

Commonly used forecast accuracy measures	
MSE	Mean Squared Error: =mean $\left(e_t^2\right)$
RMSE	Root mean squared error = $\sqrt{MSE}$
MAE	Mean Absolute error =mean $\left(\lvert e_t\rvert\right)$
MdAE	Median absolute error =median $\left(\lvert e_t\rvert\right)$
MAPE	Mean absolute percentage error =mean $\left(\lvert p_t\rvert\right)$
MdAPE	Median absolute percentage error =median $\left(\lvert p_t\rvert\right)$
sMAPE	Symmetric mean absolute percentage error =mean $\left(2\lvert Y_T-F_t\rvert/\left(Y_t+F_t\right)\right)$
sMdAPE	Symmetric median absolute percentage error =median $\left(2\lvert Y_T-F_t\rvert/\left(Y_t+F_t\right)\right)$
MRAE	Mean relative absolute error =mean $\left(\lvert r_t\rvert\right)$
MdRAE	Median relative absolute error =median $\left(\lvert r_t\rvert\right)$
GMRAE	Geometric mean relative absolute error =gmean $\left(\lvert r_t\rvert\right)$
RelMAE	Relative mean absolute error =MAE/MAE$_b$
RelRMSE	Relative root mean squared error =RMSE/RMSE$_b$
LMR	Log mean squared error ratio =log(RelMSE)
PB	Percentage better =100mean $\left(I\{\lvert r_t\rvert<1\}\right)$
PB (MAE)	Percentage better (MAE) =100mean $\left(I\{MAE<MAE_b\}\right)$
PB(MSE)	Percentage better (MSE) =100mean $\left(I\{MSE<MSE_b\}\right)$
Here $I\{u\}=1$ if u is true and 0 other wise	

Most authors agree to use as performance indicators: MSE (Mean Squared Error), RMSE (Root mean squared error) and MAE (Mean Absolute Error) for training phase evaluation [17]. Only a few reported the MAPE (Mean Absolute Percent Errors) or the standard deviation of the errors [18] and [19] (Table 1).

However, recent studies and the experience of the system operators indicate that the loss function in the load forecasting problem is clearly nonlinear, and that large errors may have disastrous consequences for utility [20] and [21]. Because of this, measures based on squared error are sometimes suggested, as they penalize large errors (RMSE was suggested in [22], MAPE in [23]).

Most researchers test their models by examining their errors in other samples than the one used for parameter estimation because the goodness-of-fit statistics are not enough to predict the actual performance of a method [12].

Also, it is generally recognized that error measures should be easy to understand and closely related to the needs of the decision-makers. Some papers reported that the utilities would rather evaluate forecasting systems by the Absolute Errors Produced, and this suggests that MAE could be useful [11]. The shape of the distribution should be suggested. Some papers included graphs of the cumulative distribution of the errors [24]. Others suggested this distribution by reporting the percentage of errors above some critical values, percentiles [25], the maximum errors [26]. In any case, no single error measure could possibly be enough to summarize the efficiency of the forecast.

3. Simulation of Forecasting Tool

Considering the bibliographical research presented above, in the next sections we underline the consequences of the choices made during the tool modeling on the forecasting performances. The cases study is made using an ANFIS network. Neuro- fuzzy systems are a combination of NN and fuzzy sets and represent a powerful tool to model systems behavior. The NN is used to define the clustering in the solution space, which results in the creation of fuzzy sets [27] and [28]. A particular architecture of neuro-fuzzy systems is represented by the Adaptive Neuro-Fuzzy Inference System (ANFIS) [29].

ANFIS is a Sugeno-type fuzzy inference system in which the parameters associated with specific membership functions are computed using a backpropagation gradient descent algorithm alone or in combination with at least squares method. It has been widely applied to random data sequences with highly irregular dynamics [30] e.g. forecasting non-periodic short-term stock prices [31] and [32].

3.1 Adaptive neuro-fuzzy inference system

Figure 1 illustrates the ANFIS architecture for two inputs parameters, where nodes of the same layer have similar functions, as described next. (Here we denote the output of the itch node in layer j as $O_{i,j}(i)$:

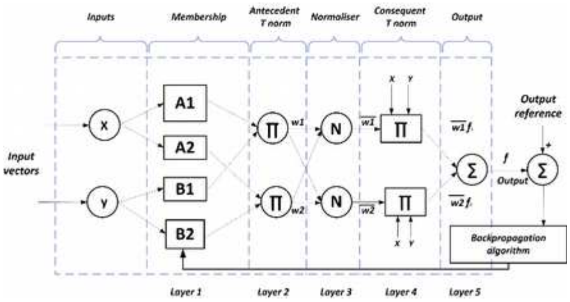

Figure 1. ANFIS architecture

Layer 1. Every *i* node in this layer is an adaptive node with a node function:

$$O_{1,i}=\mu_{A_i}(x),\ \ for\ i=1{,}2\, or$$
$$O_{1,i}=\mu_{B_{i-2}}(y),\ \ for\ i=3{,}4 \tag{3}$$

where x (or y) is the input to i node and Ai (or Bi-2) is a linguistic label (such as "small" or "large") associated with this node. In other words, O1, i is the membership grade of a fuzzy set A (=A1, A2, B1 or B2) and it specifies the degree to which the given input x (or y) meets the quantifier A.

The membership function for A can be any appropriate parameterized membership function introduced in here, such as the generalized bell function:

$$\mu_A(x)=\frac{1}{1+\left|\dfrac{x-c_i}{a_i}\right|^{2b}} \tag{4}$$

where $\{a_i, b_i, c_i\}$ is the parameter set. As the values of these parameters change, the bell-shaped function varies accordingly, thus various forms of membership function for fuzzy set A.

The parameters in this layer are referred to as premise parameters.

Layer 2. Every node in this layer is a fixed node labelled *Oi,j*, whose output is the product of all the incoming signals:

$$O_{2,i}=w_i=\mu_{A_i}(x)\cdot\mu_{B_i}(y),\ \ i=1{,}2 \tag{5}$$

Each node output represents the firing strength of a rule. In general, any other T-norm operators that performs fuzzy and can be used as the node function in this layer.

Layer 3. Every node in this layer is a fixed node labeled N. The i^{th} node calculates the ratio of the i^{th} firing strength and the sum of all firing strengths:

$$O_{3,i}=\overline{w_i}=\frac{w_i}{w_1+w_2},\ \ i=1{,}2$$

For convenience, the outputs of this layer are called *normalized firing strengths*.

Layer 4. Every *i* node in this layer is an adaptive node with a node function:

$$O_{4,1}=\overline{w_i}\cdot f_i=\overline{w_i}(p_i x+q_i y+r_i) \tag{6}$$

where w_i is a normalized firing strength from layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set of this node. The parameters in this layer are referred to as *consequent parameters*.

Layer 5. The single node in this layer is a fixed node labeled $O_{5,1}$ (overall output), which computes the overall output as the summation of all incoming signals:

$$O_{5,1}=\sum_i \overline{w_i} f_i=\frac{\sum_i w_i f_i}{\sum_i w_i} \tag{7}$$

Thus, the adaptive network is functionally equivalent to a Sugeno fuzzy model.

The ANFIS learning algorithm. When the premised parameters are fixed, the overall output is a linear combination of the consequent parameters. In symbols, the f output can be written as:

$$\begin{aligned} f=&(\overline{w_1}\cdot x)\cdot c_{11}+(\overline{w_1}\cdot y)\cdot c_{12}+\overline{w_1}\cdot c_{10}+\\ &+(\overline{w_2}\cdot x)\cdot c_{21}+(\overline{w_2}\cdot y)\cdot c_{22}+\overline{w_2}\cdot c_{20}\end{aligned} \tag{8}$$

which is linear in the consequent parameters $c_{i,j}$ *(i = 1, 2; j = 0, 1, 2)*. A hybrid algorithm adjusts the consequent parameters $c_{i,j}$ in a forward pass and the premise parameters $\{a_i, b_i, c_i\}$ in a backward pass [27].

In the layer 4 the consequent parameters are identified by the least-squares method. In the backward pass, the error signals propagate backwards and the premise parameters are updated by gradient descent.

Because the update rules for the premise and consequent parameters are decoupled in the hybrid learning rule, a computational speedup may be possible by using variants of the gradient method or other optimization techniques on the premise parameters.

The success of ANFIS is given by aspects such as: the designated distributive inferences stored in the rule base, the effective learning algorithm for adapting the system's parameters or by the own learning ability to fit irregular or non-periodic time series. On the other hand, used in application alone to non-periodic short-term forecasting, ANFIS predictions make large residual errors due to high residual variance, consequently degrading prediction accuracy [33]. It is very difficult to interpret for a non-expert the fuzzy rules generated by ANFIS because of the form of consequents (linear combination of inputs).

3.2 Simulation conditions and strategy

The ANFIS is applied on a data base obtained from an experimental photovoltaic system, where the difference between the electricity produced and consumed from renewable energy sources (DPcg) is considered as output to be forecast by ANFIS - *y (t)* and the exterior temperature as input - *u (t).*

The used data base has 296 data points $\{(y(t),u(t)) \mid t=1,\ldots, 296\}$. The estimation procedure is carried out as two ANFIS Models:

- Model 1 represents an ANFIS structure with 2 inputs: *{(y(t-a), u(t-b))}*
- Model 2 represents an ANFIS structure with 3 inputs: *{(y(t-a), u(t-b), u(t-c))}, a=1,..4; b; c=1,..6.)* (Figure 2).

The investigation uses MAPE, RMSE, and MAE, changing each time the number of inputs (2 or 3) and the model's membership functions (Gaussian or Gauss Bell type). For ANFIS the input selection method is based on the assumption that the ANFIS model with the smallest RMSE [28] after one epoch of training has a greater potential of achieving a lower RMSE when given more epochs of training. This assumption is not absolutely true, but it is heuristically reasonable. ANFIS can usually generate satisfactory results right after the first epoch of training, that is, only after the first application of the least-squares method.

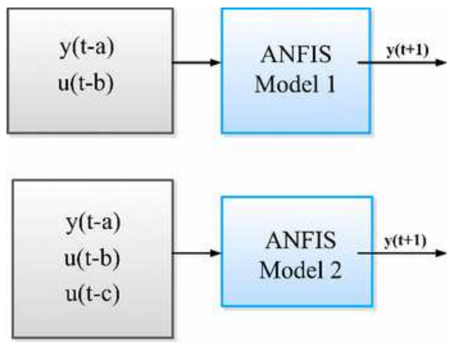

Figure 2. ANFIS models used for simulation

4. Evaluation of Tool's Performance

Considering the metrics described above, we have observed through simulations that "classical" ANFIS with two selected inputs (Model 1), has satisfactory results for a short term prediction (Figure 3) [2].

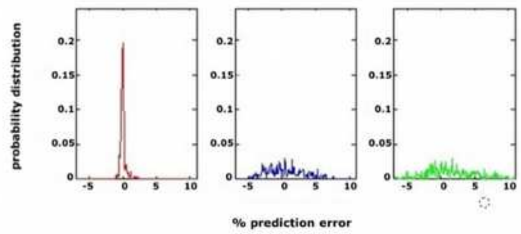

Figure 3. Error prediction distributions for ten, twenty and forty steps ahead forecasting

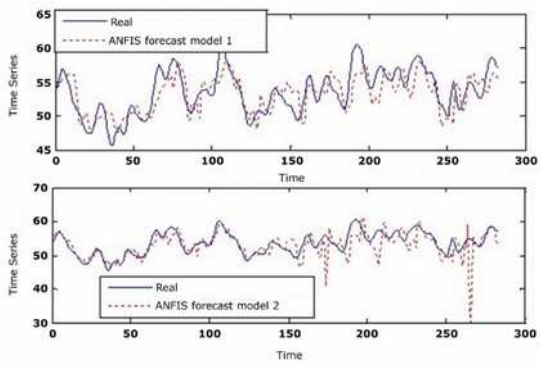

Figure 4. Prediction error and real measurements for Model 1 and Model 2

For medium and long term, the obtained errors have become bigger and bigger and this affects the forecasting in terms of accuracy and confidence.

The performance metrics related to Model 1 and Model 2 show that these models don't achieve a satisfactory compromise between short-term accuracy and long-term stability (Table 2, Figure 5a and Figure 5b).

Table 2. RMSE and MAE values

t+prev	ANFIS	RMSE_test		MAE_test	
		epochs=20	epochs=100	epochs=20	epochs=100
1	model 1	0,5532	0,5514	0,0809	0,0816
1	model 2	0,4083	0,4432	0,0597	0,0708
4	model 1	1,2228	1,23	0,2384	0,2591
4	model 2	1,1955	1,1959	0,2042	0,2239
10	model 1	3,0388	3,2713	1,1357	1,226
10	model 2	4,5201	7,0661	0,5617	1,0066
20	model 1	4,1209	4,7385	3,1039	3,382
20	model 2	4,374	8,1093	2,5285	2,6426
40	model 1	4,5597	4,622	3,1039	3,0073
40	model 2	5,7804	5,7492	2,7198	2,2013

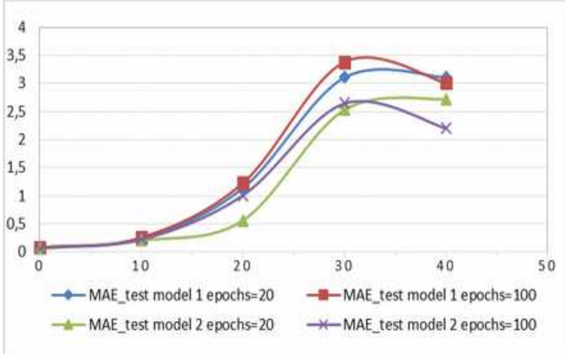

Figure 5a. MAE for model 1 and model 2 on different prediction on test data set

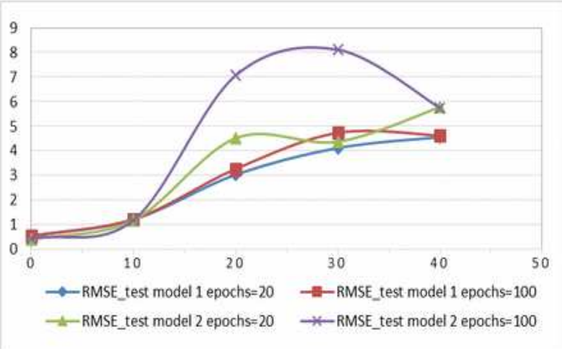

Figure 5b. RMSE for model 1 and model 2 on different prediction on test data set

The second simulation scenario identifies in terms of forecasting performance (RMSE and MAE) the effect of membership function type changing.

The models have been configured with 2 respectively 3 *Membership Functions* (MFs) and the type of MFs are Gaussian and Gaussian Bell. (Table 3 and Table 4).

The testing parameters in each case remain the same: *number of epochs= 20, ss= 0.01, ss_dec_rate= 0.5, ss_inc_rate=1.5, training/ test data set= 145/145.*

Changing the type of MFs, for each model case, does not improve the forecasting and the error in terms of MAE or RMSE who remain the same. This is also the consequence of the small number of MFs used in test. A bigger number of MFs will refine the partitioning but will increase the computational time with no relevant influence over the forecasting precision.

Table 3. MAE and RMSE values for MFs models with 2 inputs

t+prev	ANFIS		MAE_test		RMSE_test	
	type	no	model 1	model 2	model 1	model 2
1	gbellmf	2	0,0718	0,0579	0,4186	0,5325
4	gbellmf	2	0,2665	0,2863	1,1395	1,228
10	gbellmf	2	1,0757	-915	10,515	3,2758
20	gbellmf	2	3,3012	1,6894	5,3083	4,8206
40	gbellmf	2	3,0616	1,3997	5,5443	4,7185
1	gaussmf	2	0,0718	0,0579	0,4186	0,5325
4	gaussmf	2	0,2665	0,2863	1,1395	1,228
10	gaussmf	2	1,0757	-915	10,515	3,2758
20	gaussmf	2	3,3012	1,6894	5,3083	4,8206
40	gaussmf	2	3,0616	1,3997	5,5443	4,7185

Table 4. MAE and RMSE values for MFs models with 3 inputs

t+prev	ANFIS		MAE_test		RMSE_test	
	type	no	model 1	model 2	model 1	model 2
1	gbellmf	3	0,0718	0,0598	0,5739	0,5325
4	gbellmf	3	0,2665	0,4808	1,2949	1,228
10	gbellmf	3	1,0757	-0,915	10,515	3,2758
20	gbellmf	3	3,0232	1,6526	5,4733	4,4743
40	gbellmf	3	3,2591	1,3997	5,5739	5,1016
1	gaussmf	3	0,0718	0,0598	0,5739	0,5325
4	gaussmf	3	0,2665	0,4808	1,2949	1,228
10	gaussmf	3	1,0757	-0,915	10,515	3,2758
20	gaussmf	3	3,0232	1,6526	5,4733	4,4743
40	gaussmf	3	3,2591	1,3997	5,5739	5,1016

We have compared the effects of MFs number and type changing for the two models (Figure 6a and Figure 6b). The MAE and RMSE put in relation the target and network's output on testing data set and give an overview of generalization and memorization abilities of ANFIS.

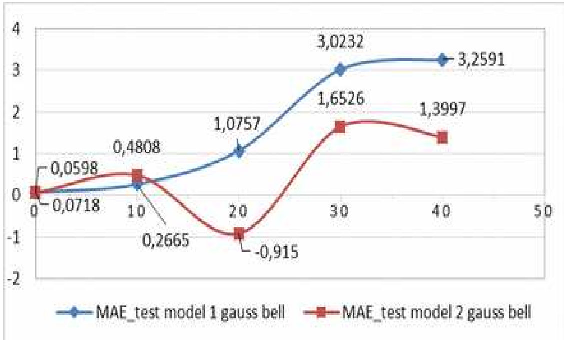

Figure 6a. MAE for Gaussian Bell of MFs typologies

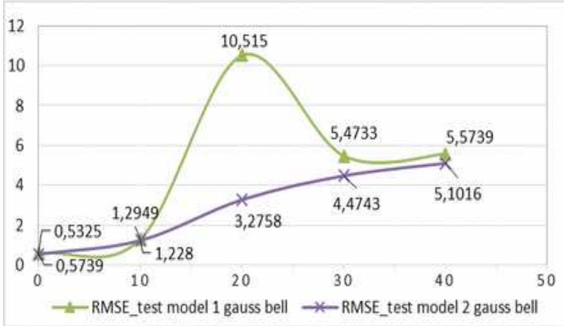

Figure 6b. RMSE for Gaussian Bell of MFs typologies

As expected, the error values in case of Model 1 are higher.

5. Conclusions

The added value of our paper consists in proposing a forecasting tool to producers/ consumers (prosumers) of renewable resources energy, based on artificial intelligence techniques, trying to obtain optimal forecasts. The exploration and the assessment of criteria used for choosing a forecasting tool were made in the neural networks context.

In this respect, firstly, the criteria used for choosing the best forecasting tool in relation with each step of the modeling process were presented. Considering the identified aspects from the bibliographical research, in the second section we have proposed two case studies based on ANFIS models, underlining the consequences of the choices made by users during the tool modeling over the forecasting performances.

The work is still in progress and the developments are at present extended in designing and implementing an intelligent informatics platform for forecasting and control of energy generation and load, in a distributed power system from renewable energy resources.

Acknowledgements

This work was supported by the Romanian National Authority for Scientific Research, project code PN-II-PT-PCCA-2011-3.2-1616, "*Intelligent Decision Support and Control System for Low Voltage Grids with Distributed Power Generation from Renewable Energy Sources*" (INDESEN), Contract 42/ 2012.

REFERENCES

1. Strategia Europa 2020, , http://ec.europa.eu/europe2020/europe-2020-in-a-nutshell/priorities/index_ro.htm
2. IRED cluster, http://www.ired-cluster.org/
3. Politica de coeziune 2014-2020, http://ec.europa.eu/regional_policy/activity/energy/index_ro.cfm
4. Autoritatea Nationala de Reglementare in domeniul Energiei, http://www.anre.ro/documente.php?id=393
5. DRAGOMIR, O., F. DRAGOMIR, E. MINCA, **An Application Oriented Guideline for Choosing a Prognostic Tool,** AIP Conference Proceedings: 2nd Mediterranean Conference on Intelligent Systems and Automation, vol. 1107, 2009, pp. 257-262.
6. DRAGOMIR, O., F. DRAGOMIR, R. GOURIVEAU, E. MINCA, **Medium Term Load Forecasting using ANFIS Predictor**, Proceedings of the 18th IEEE Mediterranean Conference on Control and Automation, Marrakech, Morocco, pp. 551-556, 2010.
7. DRAGOMIR, O., DRAGOMIR, F., MINCA, E., **Forecasting of renewable energy load with radial basis function (RBF) neural networks**, Proceedings of the 8th Int. Conference on Informatics in Control, Automation and Robotics, Noordwijkerhout, Olanda, 2011.
8. JANG, J.-S. R., C.-T. SUN, E. MIZUTAN, **Neuro-Fuzzy and Soft Computing,** Prentice Hall, 1997.
9. LACROSE, A., A. TILTI, **Fusion and Hierarchy Can Help Fuzzy Logic Controller Designer**, Proceedings of IEEE International Conference on Fuzzy Systems, Barcelona, 1997.
10. HEATON, J., **Introduction to Neural Networks with JAVA,** Heaton Research, Inc. www. heatonresearch.com
11. HIPPERT, H. S., **Neural Networks for Short-term Load Forecasting: A Review and Evaluation**, IEEE Transactions on Power Systems, 2001.

12. MAQSOOD, I., M. R. KHAN, A., ABRAHAM, **Intelligent Weather Monitoring Systems using Connectionist Neural Models,** Parallel & Scientific Computations, vol. 10, 2002, pp. 157-178.

13. ELSHOLBERG, A., S. P. SIMONOVIC, U. S. PANU, **Estimation of Missing Streamflow Data using Principles of Chaos Theory**, Journal of Hydrology, vol. 255, 2002.

14. SIFAOUI, A. ABDELKRIM, S. ALOUANE, M. BENREJEB, **On New RBF Neural Network Construction Algorithm for Classification**, Studies in Informatics and Control, ICI Publishing House, vol. 18, no. 2, pp. 6-17, 2009.

15. BABUSKA, R., JAGER, R., VERBRUGGEN, H. B., **Interpolation Issues in Sugeno-Takagi Reasoning**, 3rd IEEE International Conference on Fuzzy Systems, pp. 859-863, Orlando, FL, 1994.

16. FILDES, R., **The Evaluation of Extrapolative Forecasting Methods**, International Journal of Forecasting, vol. 8, 1992, pp. 81-98.

17. De GOOIJER, J. G., HYNDMAN, R.J., **25 Years of Time Series Forecasting**, International Journal of Forecasting, vol. 22, 2006, pp 443-473.

18. ALFUHAID, A. S., EL-SAYED, M. A., MAHMOUD, M. S., **Cascaded Artificial Neural Networks for Short-term Load Forecasting**, IEEE Transactions on Power Systems, 12: 524–1529, 1997.

19. CHOW, T. W. S., LEUNG, C. T., **Neural Network based Short-term Load Forecasting using Weather Compensation**, IEEE Transactions Power Systems, vol. 11, 1996, pp. 1736-1742.

20. HOBBS, B. F., JITPRAPAIKULSARN, S., KONDA, S., CHANKONG, V., LOPARO, K. A., MARATUKULAM, D. J., **Analysis of the Value for Unit Commitment of Improved Load Forecasting**, IEEE Transactions on Power Systems, vol. 14, 1999, pp. 1342-1348.

21. KARADY, G. G., FARNER, G. R., **Economic Impact Analysis of Load Forecasting**, IEEE Transactions on Power Systems, vol. 12, 1997, pp 1388-1392.

22. ARMSTRON, J. S., COLLOPY, F., **Error Measures for Generalizing about Forecasting Methods: Empirical Comparisons,** International Journal of Forecasting, vol. 8, 1992, pp. 69–80.

23. ARMSTRONG, J. S., FILDES, R., **Correspondence on the Selection of Error Measures for Comparisons Among Forecasting Methods**, Journal of Forecasting, vol. 14, 1995, pp. 67-71.

24. MOHAMMED, O., PARK, D., MERCHANT, R., DINH, T., TONG, C., AZEEM, A., FARAH, J., DRAKE, C., **Practical Experiences with an Adaptive Neural Network Short-term Load Forecasting System,** IEEE Transactional Power Systems, vol. 10, 1995, pp. 254-265.

25. PAPALEXOPOULOS, A. D., HAO, S., PENG, T. M., **An Implementation of a Neural Network based Load Forecasting Model for the EMS**, IEEE Transactional Power Systems, vol. 9, 1994, pp 1956-1962.

26. PENG, T. M., HUBELE, N. F., KARADY, G. G., **Advancement in the Application of Neural Networks for Short-term Load Forecasting**, IEEE Transactional Power Systems, vol. 7, 1992, pp. 250-257.

27. JANG, J. S. R., SUNI, C.T., MIZUTANI, E., **Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence**, Prentice Hall, New York, 1997.

28. NOROUZI, A., HAMEDI, M., ADINEH, V. R., **Strength Modeling and Optimizing Ultrasonic Welded Parts of ABS-PMMA using Artificial Intelligence Methods**., International Journal of Advanced Manufacturing Technologies, vol. 61, issue 1-4, 2012, pp. 135-147.

29. JANG, J. S. R., **ANFIS: Adaptive Network based Fuzzy Inference Systems**, IEEE Transactions on Systems, Manufacture, Cybernetics, vol. 23(3), 1993, pp. 665-685.

30. WANG, J. S., **An Efficient Recurrent Neuro-fuzzy System for Identification and Control of Dynamic Systems**, IEEE International Conference, Systems, Manufacture and Cybernetics, 2003.

31. CHIANG, L.H., E. RUSSEL, R. BRAATZ, **Fault Detection and Diagnosis in Industrial Systems**, Springer, UK, 2001.

32. TEODORESCU, H.-N., M.-D. ZBANCIOC, L. PISTOL, **Parallelizing Neuro-fuzzy Economic Models in a GRID Environment,** Studies in Informatics and Control, ICI Publishing House, vol. 17, no. 1, 2008, pp. 5-16.

33. GOURIEROU, C., **ARCH Models and Financial Applications**, Springer, New York, 1997.