Nonlinear Combination of Forecasts Using Artificial Neural Network, Fuzzy Logic and Neuro-Fuzzy Approaches

Ajoy Kumar Palit* and D. Popovic

* DAAD fellow, On leave from CEDTI, under DoE, P.O.-REC, Calicut-673601, India
University of Bremen, NW1 / FB1, D-28359 Bremen, Germany
E-mails: palit@uni-bremen.de, and popovic@iat.uni-bremen.de

Abstract- In the actual practice, it becomes interesting from the efficiency point of view to combine various forecasts of a specific time series into a single forecast and to interrogate the resulting forecasting accuracy. The combination is usually nonlinear. Various intelligent combination techniques have been suggested for this purpose, based on different neural network architectures, including the feedforward neural network and evolutionary neural network [1, 2]. In this paper, the nonlinear combination of time series forecasts is proposed, based on isolated use of neural networks and of fuzzy logic and neuro-fuzzy systems. On some practical examples it is demonstrated, that the non-linear combination of a group of forecasts based on intelligent approach is capable of producing a single better forecast than any individual forecasts involved in the combination.

Index Terms:- Time Series, Forecasting, Combination of Forecasts, Neural networks, Fuzzy logic system, Fuzzification, Fuzzy Rules Generation, Defuzzification, Neuro-Fuzzy system, Levenberg – Marquardt algorithm.

I. INTRODUCTION

There are often several optimal methods available for forecasting of a particular class of time series. Rather than choosing the best one among them, which might even be only marginally the best, it is often worthwhile to seek for a combination of the most perspective methods, particularly if the methods support different aspects of the underlying forecasting problem to be solved. In this case, such a combination may produce the better forecasts than either forecast separately. Although a combination of methods may be itself optimal when based on a appropriate new model that combines the relevant features underlying the forecasting methods considered, it will, however, be easier simply to combine the forecast themselves in a linear or nonlinear way.

For example, to analyze and forecast a non-stationary, non-seasonal time series one can use the Autoregressive method, Holt-Winter’s Exponential smoothing technique, Box-Jenkins ARMA / ARIMA method, Extrapolation of trend curve, Kalman filter, etc., or any combination of them. More recently, various intelligent forecasting techniques have been launched, based on artificial neural networks, modular neural networks, fuzzy logic systems, neuro-fuzzy networks, genetic fuzzy predictor, etc., and demonstrated their high efficiency. Different methods, however, be they conventional or intelligent, provide different forecasting results for a given time series, i.e., it will very rarely be identical. Hence, it may be very confusing to some one who wants to take some decision on the basis of various forecasts suggested by various analysts. For instance, the governor of a Central Bank may have several forecasts of exchange rate movements available that need to be considered in setting the prime interest rate. These forecasts may come from forecasters of different ability and reputation and from forecasters with different information in hand [1]. Also, it is totally uncertain to put always preference to a particular method and take some decision accordingly because in some occasion method A may give better forecast and in other occasion method B can outperform. It is rather much more preferable to combine intelligently various forecasts suggested by different analysts into a single one and then decide accordingly.

Therefore, it has been stated in various literature that the combined forecast is generally more accurate than any individual forecast as the combined forecast gets more information and more expert opinion into consideration [6]. Moreover, the accuracy of the combined forecast improves as more methods are included in the combination [7]. This combination is essentially a nonlinear combination rather than direct linear combination or a combination of weighted average.

Various nonlinear approaches to combine a group of multiple forecasts have been suggested to generate the single better forecast, like the more recent literatures [1], [2] respectively suggesting the nonlinear combination of forecasts by feedforward artificial neural network (back-propagation neural network) and by evolving neural networks. In the present paper, besides the application of feedforward neural networks for nonlinear combination of forecasts, two additional methods have been proposed for performing such nonlinear combination based on fuzzy logic approach, and neuro-fuzzy approach.

The organization of the paper is as follows. After first defining the problem of forecasts combination and generating the individual forecast from a given time series in section II, and III respectively, the neural network, fuzzy logic, and the neuro-fuzzy approach to combination of forecasts are described in various sub-sections of IV, along with the
illustrations of the simulated results. Finally, in section V the corresponding concluding remarks are presented.

II. PROBLEM DESCRIPTION

Given the k number of readily available forecasts \( f_1, f_2, f_3, \ldots, f_k \), of the random variable \( z \), represented by a discrete time series \( \{z,t,z_{t+1},z_{t+2},z_{t+3},z_{t+4},z_{t+5},z_{t+6}\} \), the problem is to combine them into a single forecast \( f_c \). For instance, if the linear combination

\[
f_c(z) = \sum_{i=1}^{k} W_i f_i(z)
\]

is used, where \( W_i \) is the assigned weight to the \( i^{th} \) forecast \( f_i \), the main problem is how to select the individual weights optimally. Here, the simplest one, i.e., the equal weighted (EW) combination based on arithmetic average of the individual forecasts, has proven relatively robust and highly accurate. This is evident from the case where two unbiased individual forecasts, \( f_1 \) and \( f_2 \) of a given time series are linearly combined to result in a new, also unbiased forecast \( f_c \)

\[
f_c = k \cdot f_1 + (1-k) \cdot f_2
\]

that will have a minimum mean square error for suitably chosen \( k \). The corresponding forecast errors \( e_c \), is defined using the individual errors \( e_1 \) and \( e_2 \) as

\[
e_c = k \cdot e_1 + (1-k) \cdot e_2
\]

If the two forecast errors are independent of each other, the \( E(e_c^2) \) will have its minimum for the following \( k \):

\[
k = E(e_1^2)/(E(e_1^2) + E(e_2^2)) = \frac{\bar{e}^2}{\bar{e}_1^2 + \bar{e}_2^2}
\]

where \( \bar{e}^2 \) is a local estimate of the expected error squared.

Among the other approaches to minimize the forecast error by assigning weights the minimum variance, or Bayesian approach is frequently used, although not being always robust. Anyhow, the linear combination of forecasts is not likely to be the appropriate combination in the forecasting practice. This can be demonstrated on the following example.

Suppose that \( k \) different forecast models are given, the \( i^{th} \) individual forecast having an information set \( \{I_i : I_c, I_{si}\} \), where \( I_c \) is the common part of information used by all \( k \) models and \( I_{si} \) is the special information for the \( i^{th} \) forecast only. Denoting the \( i^{th} \) forecast by \( f_i = F(I_i) \), we can express the linear combination of forecasts as:

\[
F_c = \sum w_i \cdot F_i(I_i)
\]

where \( w_i \) is the weight of the \( i^{th} \) forecast. On the other hand, every individual forecasting model given can also be regarded as a sub-system for information processing, while the combination model

\[
f_c = F_c(I_1, I_2, \ldots, I_k)
\]

is regarded as such a system. It follows, that the integration of forecasts is more than their sum, i.e., the performance of the integrated system is more than the sum of the performances of its sub-systems. So, the trustworthy of the linear forecast combination is quite questionable. More trust should be paid to a non-linear interrelation between the individual forecasts [2], such as

\[
f_c = \psi[F_c(I_1), F_c(I_2), F_c(I_3), \ldots, F_c(I_k)]
\]

where \( \psi \) is a nonlinear function. While the given information is processed by individual forecasting models, it is likely that the parts of the entire information can be lost. For instance, it could happen that the information set \( I \) is not used efficiently, or different forecast may have different parts of information lost. This is why preferably as many different forecasts as possible should be present in the combination, even when the individual forecast depends on the same set of information.

In the practice, it is generally difficult to determine the form of the nonlinear relationship \( \psi \). However, for a data driven forecasting procedure such as follows, this nonlinear mapping can be realized through isolated implementation of neural networks, fuzzy logic, and neuro-fuzzy approaches etc. The reason is that each of the technologies selected, when used separately or in combination, can implement any nonlinear mapping. More specifically the multi-layer neural networks and the fuzzy logic systems, both can approximate any nonlinear function to any degree of accuracy over the universe of discourse as confirmed by their respective universal approximation theorems [3, 4, 12].

III. GENERATING FORECASTS OF A GIVEN TIME SERIES

As a practical example, the temperature time series representing the every minute temperature readings of an uncontrolled chemical plant has been selected. It is a non-stationary non-seasonal time series [5] that has first been analyzed and forecasted using Box-Jenkins ARMA/ARIMA method and the Holt-Winter’s exponential smoothing technique respectively as the individual forecasting models. In the experiment carried out, the total of 226 time series data was taken, approximately fitted by the ARMA model

\[
Z_{t+i} = -0.8 * Z_{t+i-2} + 1.8 * Z_{t+i-1} + d_{t+i}
\]

\[
Or, \hat{Z}_t(l) = -0.8 * Z_{t+l-1} + 1.8 * Z_t
\]

where the time \( t \) is the origin at which the forecast \( \hat{Z}_t(l) \) is made and \( l \) is the lead time of forecast, representing the
number of time steps ahead the forecast should be made with respect to origin and \( \phi_{t+i} \) is the random shock. Based on the above model the forecast has been made with lead time \( l = 1 \) at different origin for \( t = 2, 3, 4, ..., 225 \). Consequently, a total of \( m = 224 \) data have been generated as a Box-Jenkins forecasted series. Similarly, the Holt-Winter’s exponential smoothing technique as described in (9) have been applied to generate the second forecasts of the same temperature series.

\[
\hat{z}_i(t) = C_0 Z_t + C_1 Z_{t-1} + C_2 Z_{t-2} + ... \\
\text{where, } C_i = (1 - \alpha)^i \text{ with, } i = 0, 1, 2, ...; \\
\text{and, } 0 < \alpha = \text{Const} \leq 1
\]  

Or, \( \hat{z}_i(t) = \alpha Z_t + (1 - \alpha) \hat{z}_{t-1}(t) \)

The two forecasted series are then arranged as column 1 and 2 respectively and the actual temperature series as the column 3 of a HBXIO matrix as follows:

\[
\text{HBXIO} = \begin{bmatrix} f_{B1} & f_{B2} & f_{Bn} \\ f_{H1} & f_{H2} & f_{Hn} \\ d_1 & d_2 & d_n \end{bmatrix}
\]

The sum squared error (SSE) of the generated forecast have been also computed as \( \text{SSE} = 0.5 * E^T * E \), where \( E \) is the column vector of errors \( e_i = (f_i - d_i) \), with \( f_i, d_i \) representing the forecast at \( i^{th} \) instant and actual value of the time series at \( i^{th} \) instant, and \( E^T \) is the transpose of the \( E \). Consequently, the SSE for Box-Jenkins’s forecast is 2.0080 and that of Holt-Winter’s forecast is 1.1688, computed for entire forecasted series. It is to be noted that smoothing constant \( \alpha = 1.6 \) have been selected in the above example of Holt-Winter’s smoothing technique, since this gave the minimum SSE for generated forecasts.

**IV. NONLINEAR COMBINATION OF FORECASTS USING ARTIFICIAL INTELLIGENT (AI) APPROACHES**

In the following, we will consider the first two columns of the HBXIO matrix representing the Box-Jenkins’s and Holt-Winter’s forecast of the given temperature series respectively, as the inputs to the AI system and the third column of the same matrix, representing the actual time series, will be considered as the output of the system. Furthermore, the first 150 rows from the HBXIO matrix will be used as the training samples for the neural and neuro-fuzzy network, whereas the same data will be used for automatic fuzzy rules generation in the fuzzy logic approach. Finally, the remaining data, i.e., the rows 151 to 224, will be used as the test samples to evaluate the efficiency of the described combination approach.

**A. Forecasts Combination Using Neural Network Approach**

In this method, a 2-6-6-1 feedforward neural network representing it’s architecture as 2 inputs, 2 hidden layers, with each layer containing 6 neurons and 1 output, is chosen.

The network is trained by the Levenberg-Marquardt (LMA) algorithm that guarantees a much faster learning speed than the standard back-propagation method. The training method also uses the steepest descent gradient algorithm based on Jacobian matrix, so that its update is

\[
\Delta x = -J^T(x) * J(x) + \mu I \\
x(k+1) = x(k) + \Delta x
\]

where \( J(x) \) is the Jacobian matrix with respect to the network adjustable parameters \( x \) (weights and biases) of the dimension \((q \times n), q \) being the number of training sets and \( n \) the number of adjustable parameters in the network. Finally, \( I \) is the identity matrix of dimension \((n \times n)\). In the procedure, the parameter \( \mu \) is multiplied by a factor \( \mu_{inc} \) whenever an iteration step increases the network performance index, i.e., the sum squared error, and is divided by \( \mu_{dec} \) whenever a step reduces the network performance index. Usually, the factor \( \mu_{inc} = \mu_{dec} \) and is selected to be 10.

The first 150 training samples were used from the HBXIO matrix to train the neural network. After the training the final values of network adjustable parameters are saved and the remaining input data sets from HBXIO matrix (151 to 224) were given to the network to test it’s performance. The result was that the trained network successfully matched the given time series data, as shown in plot-2 and table-1. It is clear that the network output very closely matches the actual time series, indicating that nonlinear combination of forecasts implemented using feedforward neural network is better than the individual forecast.

**B. Forecasts Combination Using Fuzzy Logic Approach**

In the fuzzy logic approach, used here, the fuzzy logic system selected is having 2 inputs and 1 output, where to the first and the second input the elements of the column 1 and the column 2 of the HBXIO matrix are fed respectively and the desired output from the fuzzy logic system are the elements of the column 3 of the HBXIO matrix. The first 1 to 150 (input-output) data sets from HBXIO matrix are initially used to generate the fuzzy rules automatically using the method as described in [3] by the present authors. The automated fuzzy rules generation uses the \( n = 21 \) number of symmetric Gaussian membership functions, designated by \( G_1 \) through \( G_n \). Furthermore, the domain interval, i.e., \( Xlo \) and \( Xhi \) have been selected as 18, and 28 respectively for automated fuzzy rules generation. Consequently, the number of means and variances of the implemented Gaussian membership functions is also 21, so that \( C_1 = Xlo = 18 \), and \( Xhi = 28 \). Therefore, the segment length \( S = (Xhi - Xlo) / (n-1) \); and \( C_2 = C_1 + S, C_3 = C_2 + S = C_1 + (2-1)*S, C_4 = C_1 + (r-1)*S, \) and \( C_n = C_1 + (n-1)*S = Xhi \), where \( C_1, C_2, C_3, ....,\)
Cn are the means and the σ1, σ2, σ3, ..., σn being the variances of the implemented Gaussian membership functions. The variances σ1 = σ3 = σ5 = 0.4 and σ2 = σ4 = σ6 = 0.2 have been selected in the example.

A total of 150 fuzzy rules were generated, based on the above data sets but some of them proved to be redundant or conflicting, so that they had to be removed from the rule base as per the procedure stated in [3], that results in a consistent fuzzy rule base. The fuzzy rule base is then stored for the nonlinear combination purpose. Finally for the defuzzification the center of area strategy as follows

\[ y = \left[ \sum_{i=1}^{M} y_i \mu_1(j_1) \mu_2(j_2) \right] / \left[ \sum_{i=1}^{M} \mu_1(j_1) \mu_2(j_2) \right] \] (12)

was used to determine the corresponding output from the fuzzy logic system, with \( y_i \) as the center of fuzzy region and \( I_i \) representing the ith input, and \( y \) as the defuzzified output. Based on this, the nonlinear combination of forecast, using the above rule base and the remaining 151 to 224 input data sets from HBXIO matrix, was generated. Finally, the performance of the approach was measured computing the performance indices such as SSE, MSE, RMSE, and MAE etc., for the 151 to 224 input data sets as illustrated in table-1. The result confirms the high suitability of the fuzzy logic approach as a nonlinear forecasts combiner.

C. Forecasts Combination Using Neuro-Fuzzy Approach

In the neuro-fuzzy approach, proposed here for nonlinear forecasts combination, the fuzzy logic system (13)

\[ f(x) = \frac{\sum_{i=1}^{M} c_i^l \mu_i(c_i^l)}{\sum_{i=1}^{M} \mu_i(c_i^l)} \] (13)

\[ c_i^l = \prod_{j=1}^{n} \exp \left(-\frac{(x_j - c_i^l)}{\sigma_j} \right) \]

based on the Gaussian membership function, singleton fuzzifier, product inference rule, and center of area defuzzifier, was selected.

The fuzzy logic system (13) is equivalent to the multilayer feedforward network, as described in [4] as neuro-fuzzy network, with the means \( c_i \), the variances \( \sigma_i \), and the fuzzy region centers \( y_i \) as the adjustable parameters of the network. The neuro-fuzzy network selected has 2 inputs and 1 output, and therefore \( n = 2 \). In addition, M is selected to be 10 and the first 150 (input-output) data sets from HBXIO matrix, normalized by multiplying the matrix elements with \( (1/30) \) taken for network training. For neuro-fuzzy network training the modified Levenberg- Marquardt algorithm, based on the application of both the momentum term and modified error index extension of network performance function as described in [4], was used. The selected training algorithm trains the network much faster than that of the classical back-propagation and usual Levenberg-Marquardt training algorithm. The following parameters were set for network training: \( M = 10 \), the maximum number of epochs 20, µinc value 10, µdec value 1000, µ = 0.001, gama = 0.098 etc. The selected training algorithm takes only 15 epochs to bring the initial SSE of 5.0437 down to desired error goal of SSE equal to 0.0002. Once the training is completed, the remaining data sets (151 to 224) from HBXIO matrix were used for testing the network’s performance as forecasts combiner.

To measure the effectiveness of the neuro-fuzzy approach, the performance indices such as SSE, MSE, RMSE and MAE etc., were computed and listed in table-1. Furthermore, the plot-5 shows the training performance of the neuro-fuzzy network, whereas the plot-6 shows the nonlinear combination of forecasts using the trained neuro-fuzzy network. The results show that the nonlinear combination of forecasts, based on neuro-fuzzy approach, is highly superior to any individual forecast involved in the combination.

V. CONCLUDING REMARKS

In the paper, various methods for nonlinear combination of a group of forecasts of a single time series were proposed, based on neural network, fuzzy logic, and neuro-fuzzy technology. It was shown that the intelligent technologies are highly capable of combining nonlinearly a group of forecasts of a given time series. It was also shown that the proposed combination generates a single better forecast than any individual forecast involved in the combination. From the illustration, however, it should not be concluded that the nonlinear combination of forecasts is best performed by neuro-fuzzy network, and followed by neural network approach, and fuzzy logic approach respectively. Because, it can also be demonstrated that for some other neural network configuration or for some other choice of means and variances of the implemented Gaussian membership functions, and also for some other number of membership functions in the fuzzy inference system, the fuzzy logic system may outperform other two methods.

REFERENCES

TABLE - 1
COMPARISON OF NONLINEAR COMBINATION OF FORECASTS ACCURACY OBTAINED THROUGH DIFFERENT AI APPROACHES

<table>
<thead>
<tr>
<th>Serial No.</th>
<th>Forecast- BJ / HW, or Combination Method</th>
<th>Data sets from HBXIO matrix</th>
<th>SSE</th>
<th>MSE</th>
<th>RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BJ-forecast 151 to 224 (column-1)</td>
<td>0.4516</td>
<td>0.0124</td>
<td>0.112</td>
<td>0.0871</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>HW-forecast 151 to 224 (column-2)</td>
<td>0.3174</td>
<td>0.0087</td>
<td>0.0933</td>
<td>0.0745</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ANN (2-6-6-1) 1 to 150 (training)</td>
<td>0.1306</td>
<td>0.0035</td>
<td>0.0594</td>
<td>0.0524</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ANN (2-6-6-1) 151 to 224</td>
<td>0.2425</td>
<td>0.0066</td>
<td>0.0810</td>
<td>0.0641</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ANN (2-2-6-1) 151 to 224</td>
<td>0.1680</td>
<td>0.0046</td>
<td>0.0678</td>
<td>0.0504</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Fuzzy Logic 151 to 224</td>
<td>0.0704</td>
<td>0.0019</td>
<td>0.0436</td>
<td>0.0319</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Neuro-fuzzy 151 to 224</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Plot-1 and Plot-2 illustrate the nonlinear combination of forecasts using feedforward ANN. In Plot-1 the ANN is of architecture 2-2-6-1, whereas in Plot-2 the ANN is of 2-6-6-1 architecture. Lower part of Plot-2 shows that forecasts combination errors with ANN are approximately within the range of +0.06 to – 0.13, whereas the forecasts combination errors as shown by the lower part of Plot-1 are approximately within the range of +0.12 to – 0.21. So, nonlinear combination of forecasts are better performed by feedforward ANN with 2-6-6-1 architecture. Same conclusion can be also drawn from the serial number 4 and 5 of Table-1. Since in serial number 4 SSE is 0.1306 (much lower), almost half of the SSE as in sr. no. 5.
Plot-3 shows Gaussian Membership function plot used for fuzzy combination of two forecasts of temperature series. Parameter for Gaussian Membership function plot: $\sigma_1 = \sigma_n = 0.4$ and $\sigma_2 = \sigma_3 = \ldots = \sigma_{n-3} = \sigma_{n-2} = \sigma_{n-1} = \sigma_b = 0.2$, $n = n_{MF} = 21$, $X_{lo} = 18$, $X_{hi} = 28$, $C_1 = X_{lo} = 18$, $S = (X_{hi} - X_{lo}) / (n-1)$; $C_2 = C_1 + S$, $C_3 = C_2 + S = C_1 + 2*S$, $C_r = C_1 + (r-1)*S$ and finally $C_n = C_1 + (n-1)*S = X_{hi}$, where $C_1$, $C_2$, $\ldots$, $C_n$ are the means and $\sigma_1$, $\sigma_2$, $\ldots$, $\sigma_n$ are the variances of symmetrical Gaussian function for the fuzzy combination of forecasts.

Plot-4 illustrates the nonlinear combination of 2 forecasts (BJ & HW) of temperature series using fuzzy logic approach & fuzzy combination error (bottom part of plot-4). Number of GMF used is 21, and number of Fuzzy Rules generated is 150.

Plot-5 and 6 illustrate the training performance and forecasts combination using Neuro-Fuzzy network. Plot-5 upper part shows that network attains the error goal only within 15 epochs, indicating the very high efficiency of the modified Levenberg-Marquardt training algorithm.