2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

An Improved Boosting Scheme based Ensemble of Fuzzy Neural Networks for Nonlinear Time Series Prediction

Yilin Dong
Department of Automation
East China University of Science and Technology
Shanghai, P. R. China
E-mail: yishuidefengling@163.com

Jianhua Zhang
Corresponding Author, Department of Automation
East China University of Science and Technology
Shanghai, P. R. China
E-mail: zhangjh@ecust.edu.cn

***Abstract*—This paper proposed a Modified AdaBoost.RT (AdaBoost Regression and Threshold) algorithm based on Fuzzy Neural Networks (FNNs) and its application to the accurate prediction of complex nonlinear time-series. The algorithm is validated by using four typical time-series data, namely Lorenz, Mackey-Glass, Sunspot and Dow Jones Indices data. The performance comparison of the proposed method and several existing approaches is also performed to show its advantages for nonlinear time series prediction problems.**

***Index Terms*—Modified AdaBoost.RT, ensemble learning, fuzzy neural networks, time series prediction**

I. Introduction

The predictor combination has received much attention in recent years and accordingly many ensembling methods have been developed [1], [2]. The ensembled predictor has been shown to have higher predictive accuracy than any individual predictors. Motivated by the idea of combined or hybrid predictors, ensembling technique has turned out to be an efficient strategy to achieve high performance of prediction, especially when the development of a powerful single predictor system requires considerable efforts and are difficult, if not impossible.

In this study, the fuzzy neural networks (FNNs) are used to construct an ensemble of time-series predictors. The main reasons of using FNNs are: 1) given their universal approximation properties, neural networks (NNs) are good candidate models for the purpose of global functional approximation [3], [4], [5], 2) a fuzzy inference system based on a set of fuzzy if-then rules can model the qualitative aspects of human knowledge and reasoning process as well as quantitative analyses [2], which is distinct difference between radial basis functions network (RBF) and FNN, 3) the FNN model obtained by combining NNs and fuzzy logic can overcome the drawbacks of either NNs or fuzzy systems when used alone, such as local optima, overfitting phenomenon and unability to adapt to new or variable cases/situations and 4) the learning process of FNNs is relatively fast.

In fact, the choice of an appropriate ensembling method is also quite important for ensemble modeling. Two popular ensembling algorithms are Bagging [6], [7] and Boosting [8]-[10]. Both algorithms combine the outputs of individual predictors to improve the accuracy of prediction. It has been demonstrated that Bagging and Boosting techniques are generally more accurate than individual models. In this paper, we base our method on the boosting algorithm.

The boosting algorithm was originally developed to cope with binary classification problems [9], and subsequently AdaBoost.M1 and AdaBoost.M2 was proposed by Freund and Schapire [8] to solve multi-class classification problems. So far, there have been many different variants of boosting algorithms for solving pattern classification problems [9], [10], [11].

In this work, we are mainly concerned with time series prediction problem, thus we use the boosting method for regression modeling of time series data. Freund and Schapire [8] extended AdaBoost.M2 to AdaBoost.R which projects the regression sample onto classification dataset. Drucker [12] proposed AdaBoost.R2 algorithm which is a modification of AdaBoost.R. Avnimelech and Intrator [13] extended the boosting algorithm for regression problems by introducing the notions of weak and strong learning and an approximate equivalence theorem between them. Shrestha and Solomatine [14], [15] proposed a new boosting algorithm AdaBoost Regression and Threshold (AdaBoost.RT) for regression problems. Recently, a Modified AdaBoost.RT algorithm, which will be used as an ensembling method in this paper, has been proposed by Tian [16] to alleviate the drawbacks of the original AdaBoost.RT.

The simulation results are presented by considering four typical nonlinear time series, say the Lorenz, Mackey-Glass, the Sunspot and the Dow Jones Indices, which were widely used in the literature such as [2], [3], [17]. The performance of our proposed method and that of other methods in literature is compared to show the effectiveness of our method.

The paper is structurally organized in the following way, In the section II we will present the basics of FNNs and AdaBoost.RT algorithm. Meanwhile, we derive the self-adaptive ensemble model in this section. Section Ⅲ presents the pertinent time-series data modeling and analysis results. Finally, Section Ⅳ draws some conclusions and outlines some of future work along this line of research.

II. FNNs and Ensemble Learning

This section first presents the basic concepts of FNNs, to give us an idea of how the operation of FNNs really is. In the second step we consider the ensemble learning, for a better understanding of how they are applied in the proposed method.

A. Basics of FNNs

For simplicity, consider a fuzzy inference system with rules of Takagi-Sugeno type [18]:

$$R^i: If\ \ x_1\ is\ A_1^i, x_2\ is\ A_2^i, \cdots, x_k\ is\ A_k^i,$$
$$then\ y_i = p_0^i + p_1^i x_1 + \cdots + p_k^i x_k$$

The operation of the notes in the same layer of FNN is of the same family of functions, as described below. (In subsequent, O_i^m denotes the ith of the layer m).

Layer 1: the number of nodes in input layer is equal to the dimension of input vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_k]$:

$$O_j^1 = \boldsymbol{x} \tag{1}$$

Layer 2: each node in this layer corresponds to a linguistic label and the output node is equal to the value of membership in this linguistic label. The parameters of a node can change the shape of the membership function used to characterize the linguistic label. For example, the function of the *ith* node is given by:

$$O_i^2 = \mu_{A_j^i} = \exp\left(-\left(x_j - c_j^i\right)^2 / b_j^i\right) \tag{2}$$

Where $j = 1,2,\cdots,k$, $i = 1,2,\cdots,n$, k is the input node; A_j^i is a linguistic label like, *small, large, etc.*. Associated with this node, and $\{b_j^i, c_j^i\}$ is the set of parameters. Parameters in this layer are called premise parameters.

Layer 3: each node in this layer calculates the firing power of each rule:

$$O_i^3 = \omega^i = \mu_{A_1^i}(x_1) * \mu_{A_2^i}(x_2) * \cdots * \mu_{A_k^i}(x_k) \tag{3}$$

Layer 4: the output layer calculate the overall output as:

$$O_i^4 = y_i = \sum_{i=1}^{n} \omega^i (p_0^i + p_1^i x_1 + \cdots + p_k^i x_k) \Bigg/ \sum_{i=1}^{n} \omega^i \tag{4}$$

In this way we have built an adaptive network, which combine with fuzzy inference system. The basic learning rule of FNNs is the gradient descent backpropagation, which calculates the error rates recursively from the output to the input nodes [2].

B. AdaBoost.RT Algorithm

In this paper, more recent boosting algorithm for regression problems called Modified AdaBoost Regression and Threshold (Modified-AdaBoost.RT) [16] is used to improve the performance of single FNN as an ensemble method.

The Modified-AdaBoost.RT algorithm can be summarized as follows:

1) Input:
- Sequence of m examples $(x_1, y_1), \cdots, (x_m, y_m)$, where output $y \in R$.
- Weak learning algorithm (Weak Learner).
- Integer T specifying number of iterations (machines).
- Threshold $\emptyset (0 < \emptyset < 1)$ for demarcating correct and incorrect predictions.

2) Initialize:
- Machine number or iteration $t = 1$.
- Distribution $D_t(i) = 1/m$ for all i.
- Error rate $\varepsilon_t = 0$.

3) Iterate while $t \leq T$:
- Call Weak Learner, providing it with distribution D_t.
- Build the regression model: $f_t(x) \rightarrow y$.
- Calculate absolute relative error for each training example as

$$ARE_t(i) = \left|\frac{f_t(x_i) - y_i}{y_i}\right|. \tag{5}$$

- Calculate the error rate of $f_t(x)$:

$$\varepsilon_t = \sum_{i: ARE_t(i) > \emptyset} D_t(i). \tag{6}$$

- Set $\beta_t = \epsilon_t^n$, where $n = 1,2$ or 3 (linear, square, or cubic).
- Update distribution D_t:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t, & if\ ARE_t(i) \leq \emptyset \\ 1, & otherwise. \end{cases} \tag{7}$$

  Where Z_t is a normalization factor chosen such that D_{t+1} will be a distribution.
- Set $t = t + 1$.

4) Output the final hypothesis:

$$f_{fin}(x) = \frac{\sum_t \left\{\left(lg\frac{1}{\beta}\right) f_t(x)\right\}}{\sum_t (lg\frac{1}{\beta})}. \tag{8}$$

C. Ensemble FNN Model

In this section, we describe the architecture that we have proposed for the ensemble of FNNs which is illustrated in Fig. 1.

The experiments with AdaBoost.RT algorithm have shown that the performance of ensemble machine is sensitive to parameter: $\emptyset$ [15]. The value of $\emptyset$ would affect the performance of the ensemble machine seriously and may make it unstable. Consequently, a new self-adaptive modifying the value of $\emptyset$ ensemble method is developed instead of the invariable $\emptyset$ [16]. This ensemble method can overcome the limitation of original AdaBoost.RT algorithm which attributes to estimating the value of threshold： $\emptyset$. In Solomatine's research [15], the ensemble model is stable while the value of $\emptyset$ is between 0 and 0.4 and the AdaBoost.RT becomes unstable at the value of around 0.4. Therefore, in our paper we choose 0.001 as the default initial value of $\emptyset$. The self-adaptive method is described as follows:

- Calculate the root mean square error (RMSE) of output e in every iteration:

$$\mathrm{e} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_t(x_i) - y_i)^2}. \tag{9}$$

- The value of $\emptyset$ will decrease, while $e_t < e_{t-1}$, reversely the value of $\emptyset$ will increase, while $e_t > e_{t-1}$. The detail of change is shown as the following
- equations:

$$\begin{cases} \emptyset_{t+1} = \emptyset_t \cdot (1 - \lambda), while\ e_t < e_{t-1} \\ \emptyset_{t+1} = \emptyset_t \cdot (1 + \lambda), while\ e_t > e_{t-1} \end{cases} \tag{10}$$

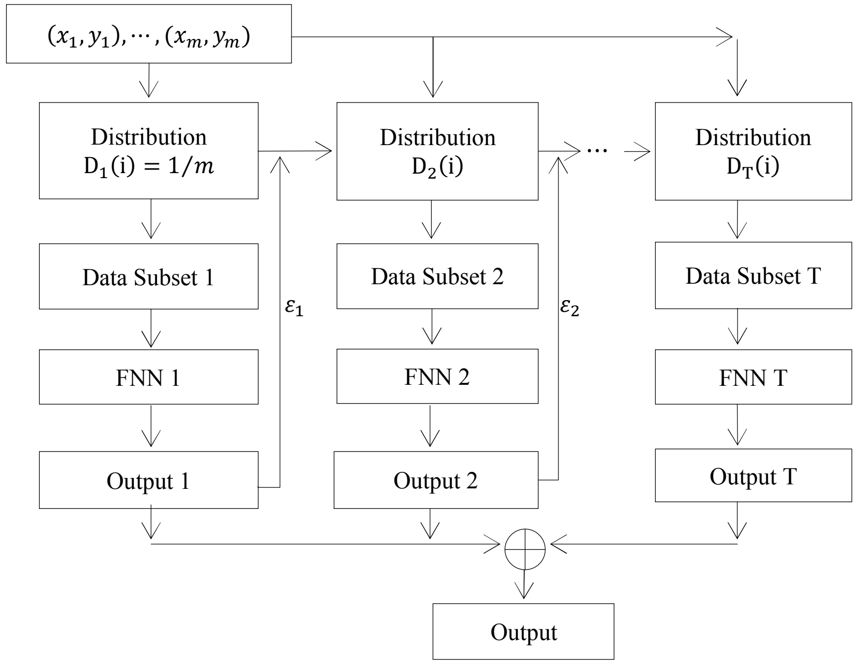

Fig. 1. The ensemble FNN model using the modified AdaBoost.RT Algorithm

- Where λ is relative to the change rate of RMSE:

$$\lambda = r \cdot \left|\frac{e_t - e_{t-1}}{e_t}\right|. \quad (11)$$

The default value of r is 0.5, it can also be determined by users for different problems.

The main idea of ensemble learning in FNNs, is that each FNN has been trained by datasets with different distributions and thus be simulated which makes something like an expert system. In general, it is widely accepted that weighted averaging with unequal weights is appropriate for combining learners to achieve a better performance.

III. Data Analysis Results

Mathematically, predicting the future of time series involves finding some nonlinear mapping f with several parameters such as

$$\hat{x}(t+p) = f\{x(t), x(t-\Delta), \cdots, x(t-(n-1)\Delta)\} \quad (12)$$

Where Δ is a lag time and n is an embedding dimension. The equation implies that an estimate $\hat{x}$ at the time (t) ahead of p can be obtained from the unknown mapping f with a proper combination of n points of the time series spaced Δ apart.

In this subsection we attempt a short-term prediction by means of the presented methods in the above subsection with regard to two simulated chaotic systems, Lorenz and Mackey-Glass equations and two real life time series: Sunspot time series and the Dow Jones Indices time series. Later, the performance of the prediction method is compared with the results reported in the literature for both Lorenz, Mackey-Glass equations and Sunspot time series.

A. Lorenz Time Series

Lorenz equations are written as:

$$\begin{cases} \dfrac{dx(t)}{dt} = \sigma[y(t) - x(t)] \\ \dfrac{dy(t)}{dt} = x(t)[\gamma - z(t)] - y(t) \\ \dfrac{dz(t)}{dt} = x(t)y(t) - bz(t) \end{cases} \quad (13)$$

Where σ, γ and b are dimensional parameters and the typical values for these parameters are $\sigma = 10$, $\gamma = 28$ and $b = 8/3$ [19], [20].

The x-coordinate of the Lorenz time series is considered for prediction and a time series with a length of 3000 is generated.

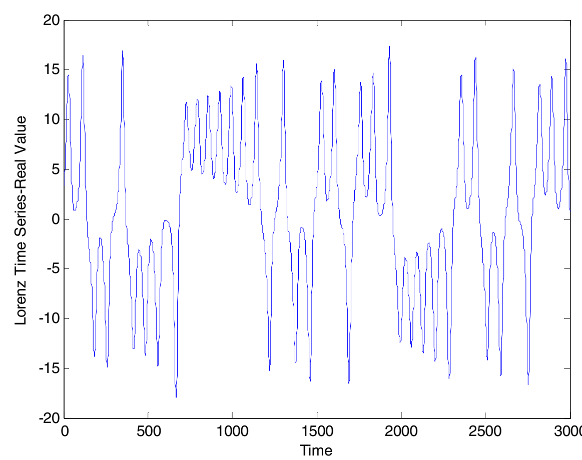

Fig. 2. Lorenz time series data

The first 2000 samples were used as training data, while the remaining 1000 were used to test the proposed model. The data in time domain is shown in Fig. 2. And the embedding

TABLE I

Comparison of the prediction accuracy of existing approaches and the proposed ensemble method (1000 test data samples of Lorenz time-series)

Prediction method	Prediction error			Ref.
	MSE	NMSE	RMSE	
Real time recurrent learning(RTRL-RNN)	-	1.72E-03	-	Mirikitani et al.[22]
Pseudo Gaussian-radial basis function	-	-	0.094	Rojas et al.[21]
ARMA-neural network	-	-	0.0876	Rojas et al.[23]
Support vector regression	-	1.46E-02	-	Martinez-Rego et al. [24]
TDL-MLP	-	1.56E-04	-	Martinez-Rego et al. [24]
DLE-VQIT	-	2.58E-04	-	Martinez-Rego et al. [24]
ROLSA	4.63E-02	-	-	Tao and Xiao [25]
Boosted recurrent neural networks(BRNN)	-	3.77E-03	-	Assaad et al.[26]
MDE-RBF	-	-	1.70E-01	Dhahri et al,[27]
Wavelet-networks B	1.64E-02	-	-	Garcia et al.[28]
FNN	3.28E-02	5.2466E-04	0.1810	
Ensemble 1 (with 100 iterations)	2.70E-02	4.3228E-04	0.1643	
Ensemble 2 (with 100 iterations)	5.9E-03	9.4749E-05	0.0769	

dimensions $n = 3$ and $\Delta = p = 3$ are estimated to allow a comparison with previous results.

In order to evaluate the prediction performance and compare it with the results reported in the literature, mean squared error (MSE), root mean squared error (RMSE) and normalized mean squared error (NMSE) are calculated according to Eqs. (14), (15) and (16) respectively.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (14)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (15)$$

$$NMSE = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad (16)$$

where y_i, $\hat{y}_i$ and $\bar{y}$ are observed data, predicted data and the average of observed data respectively, and N is the length of observed data.

Fig. 3(a) presents the comparison between the real time series and that predicted by the ensemble algorithm: Modified AdaBoost.RT-FNN, using 3 input variables, in order to predict the value of the time series. As they are practically identical, the difference can only be seen on a finer scale (Fig. 3(b)).

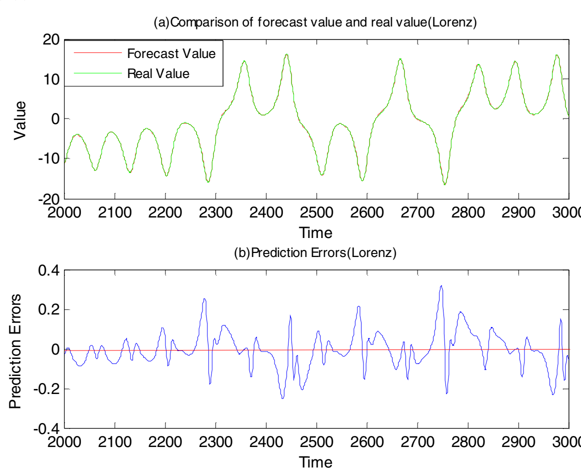

Fig. 3. Number of iterations $T = 100$, Prediction step $p = \Delta = 3$: (a) the original and predicted Lorenz time series, (b) predictive error.

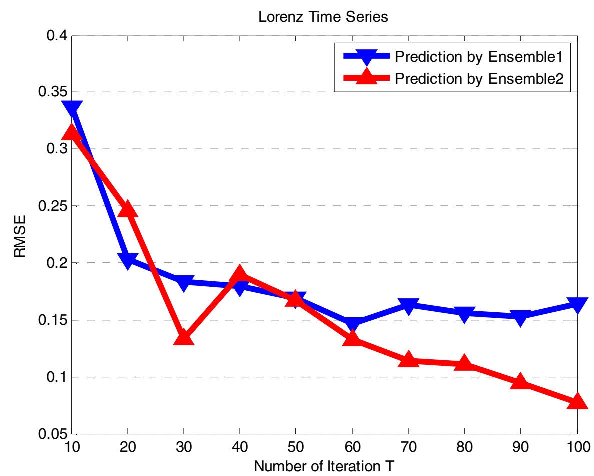

Fig. 4. Result of prediction of Lorenz time series (RMSE vs. iteration number) (Ensemble 1:AdaBoost.RT-FNN; Ensemble 2: Modified AdaBoost.RT-FNN)

When the number of iteration is increased, the error index decreases. Fig. 4 shows the results of predicting the time series of the Lorenz equations when the number of iteration changes (RMSE).

Table I compares the prediction accuracy of different computational paradigms presented in the bibliography for this benchmark problem (including our proposal) for various ensemble fuzzy systems, neural networks and other methods. The data are taken from [17], [21]. (Ensemble 1: AdaBoost.RT-FNN; Ensemble 2: Modified AdaBoost. RT - FNN)

The error indices, the mean square error (MSE), normalized mean squared error (NMSE) and the root mean square error (RMSE), for this simulation are 0.0059, 9.4749E-05 and 0.0769. It is important to note that other approaches appeared in the bibliography, for example, Mirikitani et al. [22] obtained an NMSE of 0.00172.

The simulation confirms that the proposed ensemble method: Modified AdaBoost.RT-FNN, when it is used to forecast the future values of Lorenz time series, generates better results compared to other prediction methods reports in the literature.

B. *Mackey-Glass Chaotic Time Series*

The Mackey-Glass equation originally has been proposed as a model of blood cell regulation. Mackey-Glass has been used in literature as benchmark model due to its chaotic characteristics. The differential equations leading to the time-series is demonstrated in:

$$\frac{\mathrm{dx}}{\mathrm{dt}} = \frac{ax(t-\tau)}{[1+x^{c}(t-\tau)]} - bx(t) \tag{17}$$

In the Mackey-Glass equation, the delay parameter, τ determines the characteristic of Eq. (17): i.e. $\tau < 4.43$ produces a fixed point attractor, $4.43 < \tau < 13.3$: stable limit cycle attractor, $13.3 < \tau < 16.8$: double limit cycle attractor and $\tau > 16.8$: chaos.

The parameters are selected according to the previous report by Ardalani-Farsa and Zolfaghari [17], where the constants are taken to be $\mathrm{a} = 0.2$, $\mathrm{b} = 0.1$ and $\mathrm{c} = 10$ and chaotic time series isgenerated by time delay $\tau = 17$ and initial value $\mathrm{x}(0) = 1.2$.

A chaotic time series samples set with length of 1000 is generated by the Eq. (17). The first 500 samples were used as training data, while the remaining 500 were used to validate the model identified. The data in time domain is shown in Fig. 5. To make the comparisons with earlier work fair, we choose the parameters of $n = 3$ and $\Delta = p = 6$.

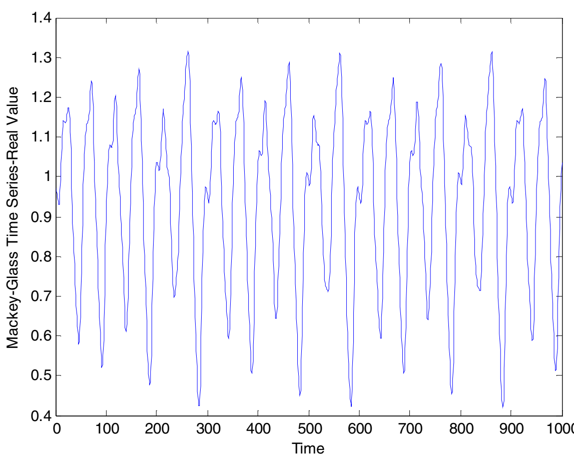

Fig. 5. Mackey-Glass time series data

Fig. 6(a) shows the predicted and desired values for the checking data. As they are practically identical, the difference can only be seen on a finer scale (Fig. 6(b)). The prediction performance of the ensemble model improves largely when the number of iteration T increases. Fig. 7 shows the results of predicting the time series of the Mackey-Glass Equations when the number of iteration changes (RMSE).

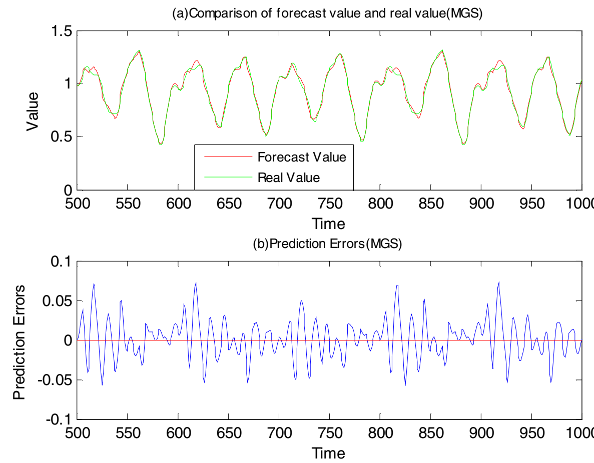

Fig. 6. Number of iterations $T = 100$, Prediction step $p = \Delta = 6$: (a) the original and predicted Mackey-Glass time series, (b) predictive error.

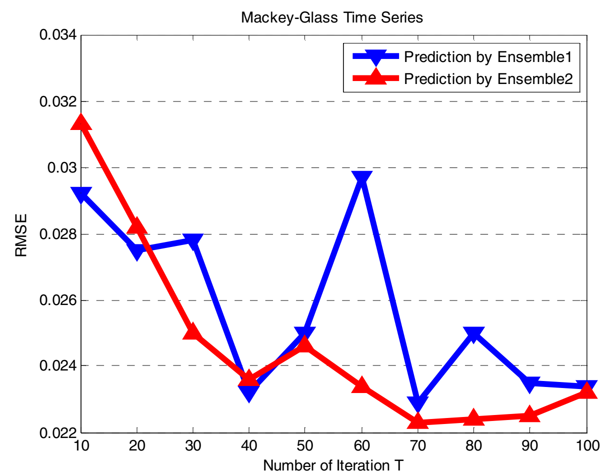

Fig. 7. Result of prediction of Mackey-Glass time series (RMSE vs. iteration number) (Ensemble 1:AdaBoost.RT-FNN; Ensemble 2: Modified AdaBoost.RT-FNN)

Table II compares the prediction accuracy of different computational paradigms presented in the bibliography for this benchmark problem (including our proposal), for various fuzzy system structures, neural systems and genetic algorithms. (Ensemble 1: AdaBoost.RT-FNN; Ensemble 2: Modified-AdaBoost.RT-FNN)

TABLE II
Comparison of the prediction accuracy of existing approaches and the proposed ensemble method (500 test data samples of MGS time-series)

Prediction method	Prediction error			Ref.
	MSE	NMSE	RMSE	
AutoRegressive model	-	-	0.19	Rojas et al.[21]
Genetic algorithm and fuzzy system -5 MFs	-	-	0.0492	Kim and Kim [29]
-7 MFs	-	-	0.0422	Kim and Kim [29]
-9 MFs	-	-	0.0378	Kim and Kim [29]
Hybrid neural network (HNN)	-	5.30E-02	-	Inoue et al. [30]
Product T-norm	-	-	0.0907	Wang et al. [31]
Min T-norm	-	-	0.0904	Wang et al. [31]
Hybrid NARX-Elman RNN with residual analysis	-	2.70E-08	3.72E-05	Ardalani-Farsa et al. [17]
FNN	7.1968E-04	0.0141	0.0268	
Ensemble 1 (with 100 iterations)	5.4953E-04	0.0107	0.0234	
Ensemble 2 (with 100 iterations)	5.3826E-04	0.0105	0.0232	

The error indices, the mean square error (MSE), normalized mean squared error (NMSE) and the root mean square error (RMSE), for this simulation are 5.3826E-04, 0.0105 and 0.0234. It is important to note that other approaches appeared in the bibliography, for example, Rojas et al. [21] obtained an RMSE of 0.19.

Meanwhile, as expected, the greater the number of iteration of the ensemble model is, the lower the error index for the real value (the remaining 500 data).

The simulation confirms that the proposed ensemble method: Modified AdaBoost.RT-FNN, when it is used to forecast the future values of Mackey-Glass time series, generates better results compared to other prediction methods reports in the literature.

C. *Sunspot time series*

Forecasting solar activity is a challenging area and important topic for various researchers and industries [32]. The Sunspot time series is a good indication of solar activity for solar cycles. The impact of solar activity has been observed on earth, climate, weather, satellites and space missions. Therefore, it is critical to forecast Sunspot time series. However, because of the complexity of the system and the lack of a mathematical model, forecasting solar cycle is extremely challenging [19].

The monthly smoothed Sunspot time series has been obtained from the SIDC (World Data Center for the Sunspot Index). To compare the results with some of the research works published in the literature, data are selected in the same conditions reported by Ardalani-Farsa et al. [17], Gholipour et al. [19] and Ma et al. [20]. Sunspot series from November 1834 to June 2001 (2000 points) are selected. The first 1000 samples of the time series are selected to train and the remaining 1000 samples are kept to test the prediction method. The data in time domain is shown in Fig. 8. The parameters $n = 5$ and $\Delta = p = 1$ are estimated and accordingly phase space points are reconstructed.

Fig. 9(a) shows the predicted and desired values for the checking data. As they are practically identical, the difference can only be seen on a finer scale (Fig. 9(b)). The prediction performance of the ensemble model improves largely when the number of iteration T increases. Fig. 10 shows the results of predicting the time series of the Sunspot time series when the number of iteration changes (RMSE).

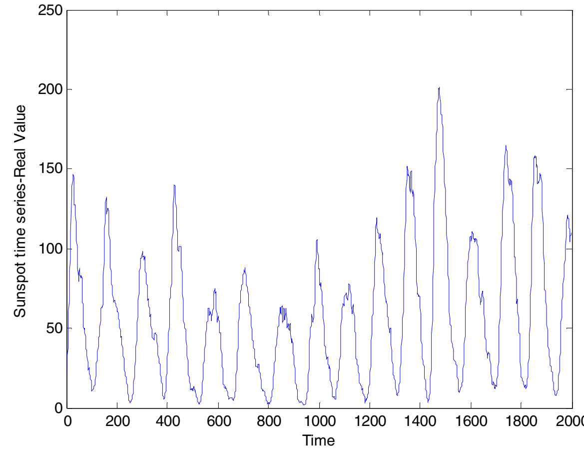

Fig. 8. Sunspot time series data

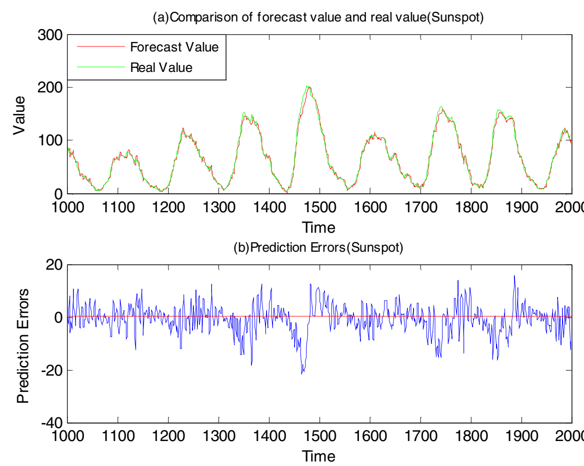

Fig. 9. Number of iterations $T = 100$, Prediction step $p = \Delta = 1$: (a) the original and predicted Sunspot time series, (b) predictive error.

Table III compares the prediction accuracy of different computational paradigms presented in the bibliography for this benchmark problem (including our proposal), for various fuzzy system structures, neural systems and genetic algorithms.

The error indices, the mean square error (MSE), normalized mean squared error (NMSE) and the root mean square error (RMSE), for this simulation are 32.6876, 0.0135 and 5.7173.

TABLE III

Comparison of the prediction accuracy of existing approaches and the proposed ensemble method (1000 test data samples of Sunspot time-series)

Prediction method	Prediction error			Ref.
	MSE	NMSE	RMSE	
WP-MLP(A)	-	1.25E-01	-	Teo et al.[33]
McNish-Lincoln	-	8.00E-02	-	McNish and Lincoln [34]
Sello-nonlinear method	-	3.40E-01	-	Sello [32]
Waldmeier	-	5.60E-01	-	Sello [32]
Denkmayr	-	1.85	-	Denkmayr and Cugnon [35]
RBF-OLS	-	4.60E-02	-	Gholipour et al. [19]
LLNF-LoLiMot	-	3.20E-02	-	Gholipour et al. [19]
ERNN	-	2.80E-03	1.29E-02	Ma et al.[20]
FNN	42.1264	0.0174	6.4905	
Ensemble 1 (with 100 iterations)	38.6937	0.0160	6.2204	
Ensemble 2 (with 100 iterations)	32.6876	0.0135	5.7173	

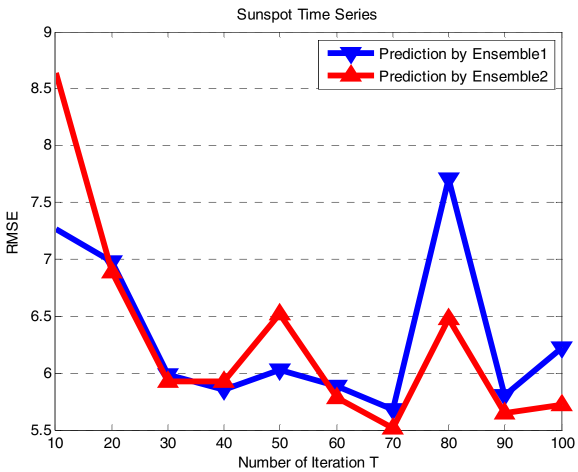

Fig. 10. Result of prediction of Sunspot time series (RMSE vs. iteration number) (Ensemble 1: AdaBoost.RT-FNN; Ensemble 2: Modified AdaBoost.RT-FNN)

It is important to note that other approaches appeared in the bibliography, for example, Ma et al. [20] obtained an NMSE of 2.80E-03.

D. *Dow Jones Indices*

For the problem of the Dow Jones time series (Dow Jones Indexes, 2010) we are using 800 pairs of data [2] that correspond from 11/03/05 to 01/08/09, and Fig. 11 shows a plot the Dow Jones time series. The first 400 samples of the time series are selected to train and the remaining 400 samples are kept to test the prediction method. The parameters $n = 3$ and $\Delta = p = 6$ are estimated and accordingly phase space points are reconstructed.

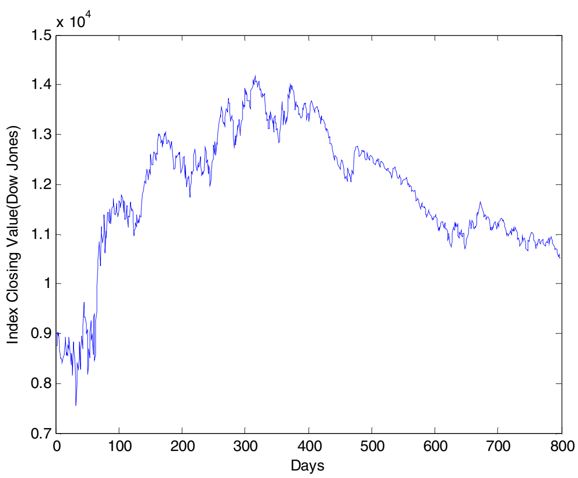

Fig. 11. Dow Jones time series data

Fig. 12(a) shows the predicted and desired values for the checking data. As they are practically identical, the difference can only be seen on a finer scale (Fig. 12(b)). The prediction performance of the ensemble model improves largely when the number of iteration T increases. Fig. 13 shows the results of predicting the time series of the Dow Jones time series when the number of iteration changes (RMSE).

TableⅣcompares the prediction accuracy of three methods: FNN, AdaBoost.RT-FNN and Modified AdaBoost.RT-FNN which have been used in our paper. It is obviously that the self-adaptive ensemble model achieves the best result.

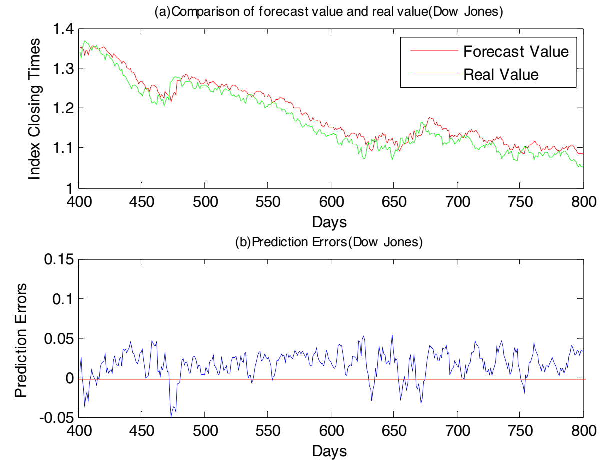

Fig. 12. Number of iterations $T = 100$, Prediction step $p = \Delta = 6$: (a) the original and predicted Dow Jones time series, (b) predictive error.

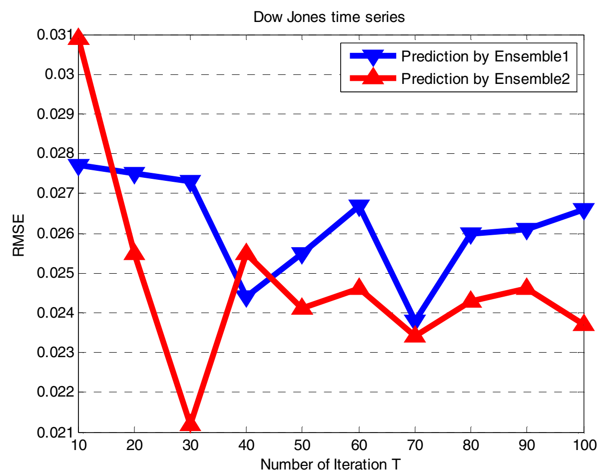

Fig. 13. Result of prediction of Dow Jones time series (RMSE vs. iteration number) (Ensemble 1: AdaBoost.RT-FNN; Ensemble 2: Modified AdaBoost.RT-FNN)

TABLE Ⅳ

Comparison of the prediction accuracy of existing approaches and the proposed ensemble method (400 test data samples of Dow Jones time-series)

T	Model	Prediction error		
		MSE	NMSE	RMSE
-	FNN	7.1737E-04	0.1278	0.0268
100	Ensemble 1	7.0609E-04	0.1258	0.0266
100	Ensemble 2	5.6023E-04	0.0820	0.0237

IV. CONCLUSIONS

In this work, a new time-series prediction method is developed by combining FNNs and a modified AdaBoost.RT algorithm. Based on the embedding theorem, the original time series can be unfolded with embedding dimensionality and time delay and hence reconstructed in the phase space. A FNN with rules of Takagi-Sugeno type is used to predict future values of the time series data points in embedded phase space. Furthermore, a new ensemble learning method based on a modified AdaBoost.RT algorithm is used to enhance the predictive accuracy of single FNN. The resulting ensemble model is shown to be able to achieve better time-series forecasting performance.

The developed method has been applied to multi-step-ahead (long-range) prediction of nonlinear time series and compared with other time-series forecasting methods reported in the literature in terms of several different predictive error indices. The issue of long-range time-series forecasting and the analysis of the computational burden required by the proposed ensemble modeling method should be considered in future work as an extension to the present study.

The proposed model exhibits a superior performance in prediction of chaotic time series and features simplicity for implementation. Based the simulation results presented above, we can conclude that the modified AdaBoost.RT-FNN method is more suitable for nonlinear time series prediction problem than previous methods, for instance FNN and AdaBoost.RT-FNN.

In future work, we would consider:

1) to optimize the structure of ensemble model;

2) to define a proper individual model diversity metrics to further improve the performance of resultant ensemble model;

3) to use the developed ensemble modeling technique to other real-world nonlinear time series prediction problems to validate its practicability in a more comprehensive fashion.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant No. 61075070 and Key Grant No. 11232005.

REFERENCES

[1] P. Melin, A. Mancilla, M. Lopez, and O. Mendoza, "A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting," *Appl.Soft Comput.*, vol. 7, pp. 1217-1226, 2007.

[2] P. Melin, J. Soto, O. Castillo, and J. Soria, "A new approach for time series prediction using ensembles of ANFIS models," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3494-3506, 2012.

[3] M. Assaad, R. Bone, and H. Cardot, "A new boosting algorithm for improved time-series forecasting with recurrent neural networks," *Information Fusion*, vol. 9, pp. 41-55, 2008.

[4] G. B. Huang, L. Chen, and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879-892, 2006.

[5] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," *In Proceeding of International Conference on Neural Networks (IJCNN 2004)*, pp. 985-990.

[6] L. Breiman, "Stacked regressor," *Machine Learning*, vol. 24, pp. 49-64, 1996.

[7] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123-140, 1996.

[8] Y. Freund and R. Schapire, "A decision-theroretic generalization of on-line learning and an application of boosting," *J.Comput. Syst. Sci.* vol. 55, pp. 119-139, 1997.

[9] R. Schapire, "The strength of weak learnability," *Mach. Learn.,* vol. 5, pp. 197-227, 1990.

[10] R. Schapire, Y. Freund, P. Barttlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Ann. Stat.*, vol. 29, pp. 1651-1686, 1998.

[11] T. Dietterrich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, Boosting and randomization," *Mach. Learn.* vol. 40, pp. 139-157, 2000.

[12] H. Drucker, "Improving regressor using boosting," *In Proc. 14th Int. Conf. Mach. Learn.,* 1997, pp. 107-115.

[13] R. Avnimelech and N. Intrator, "Boosting regression estimators," *Neural Comput.*, vol. 11, pp. 499-520, 1999.

[14] D. P. Solomatine and D. L. Shrestha, "AdaBoost.RT: A boosting algorithm for regression problems," *In Proceeding of International Joint Conference of Neural Networks*, 2004, pp. 1163-1168.

[15] D. L. Shrestha and D. P. Solomatine, "Experiments with AdaBoost.RT: An improved boosting scheme for regression," *Neural Comput.*, vol. 18, pp. 1678-1710, 2006.

[16] H. X. Tian and Z. Z. Mao, "An Ensemble ELM Based on Modified AdaBoost.RT Algorithm for Predicting the temperature of molten steel in ladle furnace," *IEEE Transactions on automation science and engineering*, vol. 7, no. 1, 2010.

[17] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman-NARX neural networks," *Neurocomputing*, vol. 73, no. 13, pp. 2540-2553, 2010.

[18] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, pp. 116-132, 1985.

[19] A. Gholipour, B. N. Araabi and C. Lucas, "Predicting chaotic time series using neural and neurofuzzy models: a comparative study," *Neural Processing Letters*, vol. 24, pp. 217-239, 2006.

[20] Q. Ma, Q. Zheng, H. Peng, T. Zhong and L. Xu, "Chaotic time series prediction based on evolving recurrent neural networks," *In Proceeding of the Sixth International Conference on Machine Learning and Cybernetics*, Hong Kong, 2007.

[21] I. Rojas, H. Pomares, J. L. Bernier, J. Ortega, B. Pino, F. J. Pelayo and A. Prieto, "Time series analysis using normalized PG-RBF network with regression weights," *Neurocomputing*, vol. 42, pp. 267-285, 2002.

[22] D. Mirikitani and N. Nikolaev, "Recursive Bayesian recurrent neural networks for time-series modeling," *IEEE Trans. Neural Networks*, vol. 21, no. 2, pp. 262-274, 2010.

[23] I. Rojas, O. Valenzuelab, F. Rojasa, A. Guillena, L. J. Herreraa, H. Pomaresa, L. Marquezb and M. Pasadasb, "Soft-computing techniques and ARMA model for time series prediction," *Neurocomputing*, vol. 71, pp. 519-537, 2008.

[24] D. Martinez-Rego, O. Fontenla-Romero and A. Alonso-Betanzos, "A method for time series prediction using a combination of linear models" *In Proceedings of the European Symposium on Artificial Neural Network-Advances in computational Intelligence and Learning*, Bruges, Belgium, 2008.

[25] D. Tao and H. Xiao, "Chaotic time series prediction based on radial basis function network," *In Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel Distributed Computing*, vol. 1, 2007, pp. 595-599.

[26] M. Assaad, R. Bone and H. Cardot, "Predicting chaotic time series by boosted recurrent neural networks," *In Proceedings of the International Conference on Information Processing*, vol. 4233, 2006, pp. 831-840.

[27] H. Dhahri and A. M. Alimi, "The modified differential evolution and the RBF(MDE-RBF) neural network for time series prediction," *In Proceedings of the International Joint Conference on Neural Networks*, Vancouver, Canada, 2006, pp. 2938-2943.

[28] E. S. Garcia-Trevino and V. Alarcon-Aquino, "Single- step prediction of chaotic time series using wavelet-networks," *In Proceedings of the Electronics, Robotics and Automotive Mechanics Conference*, vol. 1, 2006, pp. 243-248.

[29] D. Kim and C. Kim, "Forecasting time series with genetic-fuzzy predictor ensemble," *IEEE Transactions on Fuzzy Systems*, vol. 5, pp. 523-535, 1997.

[30] H. Inoue, Y. Fukunaga and H. Narihisa, "Efficient hybrid neural network for chaotic time series prediction," *In Proceedings of the International Conference on Artificial Neural Networks*, pp. 712-718, 2001.

[31] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Systems Man Cybernet*, vol. 22, no. 6, pp. 1414-1427, 1992.

[32] S. Sello, "Solar cycle forecasting: a nonlinear dynamics approach," *Astronomy and Astrophsics*, vol. 377, 2001, pp. 312-320.

[33] K. Teo, L. Wang and Z. Lin, "Wavelet packet multi-layer perceptron for chaotic time series prediction: effects of weight initialization," *Computational Science*, vol. 2074, pp. 310-317, 2001.

[34] A. G. McNish and J. V. Lincoln, "Prediction of sunspot numbers," *Transactions American Geophysical Union*, vol. 30, pp. 673, 1949.

[35] K. Denkmayr and P. Cugnon, "About sunspot number medium-term predictions," *In: G. Heckman et al.(Ed), Solar-Terrestrial Prediction Workshop V, Hiraiso Solar Terrestrial Research Center*, pp. 103, 1997.