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

Neural Networks and AdaBoost Algorithm based Ensemble Models for Enhanced Forecasting of Nonlinear Time Series

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

Jonathan M. Garibaldi
(IMA) Research Group
School of Computer Science
University of Nottingham
Nottingham, UK
E-mail:jmg@cs.nott.ac.uk

***Abstract*—In this paper an optimized AdaBoost Regression and Threshold (AdaBoost.RT) algorithm based on feed-forward neural networks is evaluated. The AdaBoost.RT algorithm is used to combine an ensemble of feed-forward neural networks trained by using backpropagation algorithm (FFN-BP). The ensemble model is validated by using two typical time-series data, namely Chua's circuit and CATS benchmark data. The performance of the ensemble models is shown to outperform several existing approaches.**

***Index Terms*—AdaBoost.RT algorithm, ensemble learning, feed-forward neural networks, time-series prediction.**

I. Introduction

Over the last several decades, nonlinear time-series forecasting has been a popular and challenging research subject arising in a wide range of application areas. The artificial neural networks (ANNs) have been extensively employed, independently or as an auxiliary tool, to predict nonlinear time series [1]. They possess such characteristics as self-organization, data-based learning, and associative memory. Among all types of ANN models, the feed forward network model with backpropagation training procedure (FFN-BP) is one of the most commonly used networks [2]. However, this network is also notorious in being trapped to local optima [3], hence many variants have been proposed to overcome some of its drawbacks.

The ensembling is a typical tool for improving the performance of the individual regressive models since it was found that an ensemble of individual predictors usually outperforms a single predictor [4]. The Adaptive Boost (AdaBoost) scheme is one of the most popular techniques for generating ensemble models due to its adaptability as well as its simplicity. The main idea of AdaBoost is to construct a set of base learners by using different training datasets that are derived by resampling the original data. Through a weighted voting mechanism, these learners are combined to predict a new test instance. Normally, the performance of a base learner is slightly better than random guess [5]. In this paper, a more recent boosting scheme, that is AdaBoost Regression and Threshold (AdaBoost.RT), first outlined in (Solomatine and Shrestha, 2004) [6], is used to enhance the performance of "*individual*" learner, here, "*individual*" learner is FFN-BP. This method utilizes the embedding theorem to "*unfold*" the nonlinear time series and reconstruct the points in phase space.

The simulations results are presented by considering two typical nonlinear time-series data, say Chua's Circuit (1998) and the CATS Benchmark (2004). In the latter simulation example, we constructed and compared two different ensemble models trained by using different choice of training dataset. The prediction problem of the two time-series has been widely explored in literature [4], [7]—[14], thus allowing us to concretely compare the performance of our method and other methods in a quantitative way.

The paper is organized in the following structure. In the next section we will present the basics of FFN-BP. Section Ⅲ derives the Adaboost.RT algorithm and the ensemble model. Section Ⅳ presents the simulation results. Finally, Section Ⅴ draws conclusions and outlines some of future work along this line of research.

II. Feed Forward Network Based Predictive Model

The well-known three layer feed forward network is the modeling technique chosen, due to its success in single time series forecasting, simplicity of operation, ability to perform universal function approximation and robustness when compared to other techniques.

The feed forward network architecture is data dependent, but fixed for each time series. The neuron number of the input layer is selected to be equal to the dimensions of input data so that we can achieve better performance of prediction [15]. For the simplification of calculation, the neuron number of the hidden layer is selected to be $2d+1$ (d represents the dimension of input data) [16]. The output layer has only one neuron. Of course, the numbers of the neurons in the input layer or the hidden layer can be set to different numbers according to actual requirement. Meanwhile, the hyperbolic tangent function was used for the hidden layer and the linear transfer function-for the output layer. For all dataset, default value of the 0.1 was taken as learning rate. The number of epochs was 20. From our experience, the above setting is enough for the simulations in Section Ⅳ.

III. AdaBoost.RT-FNN-BP Based Predictive Method

The AdaBoost algorithm is an adaptive ensemble method which has been proved to be an effective method in classification and regression problems [17], [18]. In this paper, more recent boosting algorithm for regression problems called AdaBoost.RT is used (here R stands for regression and T for threshold) [19].

A. AdaBoost.RT Algorithm

AdaBoost.RT Algorithm

1) Input:
- Sequence of m examples (x_1, y_1), ⋯, (x_m, y_m), where output $y \in \mathbb{R}$
- 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 $\mathrm{t} = 1$
- Distribution $\mathrm{D_t(i)} = \frac{1}{\mathrm{m}}$ for all i
- Error rate $\varepsilon_{\mathrm{t}} = 0$

3) Iterate: While $\mathrm{t} \leq \mathrm{T}$
- Call weak learner, providing it with distribution $\mathrm{D_t}$
- Build the regression model: $\mathrm{f_t(x)} \rightarrow \mathrm{y}$
- Calculate absolute relative error for each training example as:
$$ARE_t(i) = \left|\frac{f_t(x_i) - y_i}{y_i}\right|. \quad (1)$$
- Calculate the error rate of $\mathrm{f_t(x_i)}$:
$$\varepsilon_t = \sum_{i:ARE_t(i)>\emptyset} D_t(i). \quad (2)$$
- Set $\beta_{\mathrm{t}} = \varepsilon_{\mathrm{t}}^{\mathrm{n}}$, where n = power coefficient (e.g. linear, square or cubic)
- Update distribution $\mathrm{D_t}$ as
$$D_{t+1}(i) = \begin{cases} \dfrac{D_t(i)}{Z_t} \times \beta_t, & ARE_t(i) \leq \emptyset, \\ \dfrac{D_t(i)}{Z_t} \times 1, & \text{otherwise.} \end{cases} \quad (3)$$
In Eq. (3), $\mathrm{Z_t}$ is a normalization factor chosen such that D_{t+1} will be a distribution
- Set $\mathrm{t} = \mathrm{t} + 1$

4) Output the final hypothesis:
$$f_{fin}(x) = \frac{\sum_t \left\{(lg\frac{1}{\beta})f_t(x)\right\}}{\sum_t (lg\frac{1}{\beta})}. \quad (4)$$

B. AdaBoost.RT-FFN-BP Based Predictive Model

It is of great significance to improve the performance of prediction in time series research areas. Although optimize single predictor such as FFN-BP can in some degree improve the accuracy of prediction, such simple neural networks cannot well forecast when it is applied to the complex nonlinear time-series prediction.

In order to enhance the performance of FFN-BP predictor, an ensemble model: AdaBoost.RT-FFN-BP is used.

The methodology developed in this study is as follow:

START
  Input: $(x_1, y_1), \cdots (x_m, y_m), x, y \in \mathbb{R}$;
  Initialize the distribution for all i: $\mathrm{D_t}(i) = \frac{1}{m}$;
FOR $\mathrm{t} = 1, \cdots, \mathrm{T}$
  Select the training data depend on distribution $D_t(i)$ and train FFN-BP;
  Obtain the FFN-BP: $f_t(x)$;
  Calculate $ARE_t(i) = \left|\frac{f_t(x_i) - y_i}{y_i}\right|$;
  Calculate the error rate of $f_t(x)$
$$\varepsilon_t = \sum_{i:ARE_t(i)>\emptyset} D_t(i)$$
  Calculate $\beta_t = \varepsilon_t^n$;
  Update $D_{t+1}(\mathrm{i})$;
End loop;
  Output: $f_{fin}(x)$;
END

C. The Bias-Variance Decomposition

If we average the output of several different models $f_t(x)$, we call it an ensemble model

$$\hat{f}(x) = \sum_{t=1}^{T} w_t f_t(x). \quad (5)$$

We further assume that the model weights w_t sum to one $\sum_{t=1}^{T} w_t = 1$. The central feature of this method its generalization ability. It was shown by several authors, that the generalization error of an ensemble model could be improved if the single models on which averaging is done disagree and if their output is uncorrelated [20], [21]. This becomes obvious, if we investigate the expected generalization error:

$$\mathrm{Err}(\mathbf{x}) = \left\|\hat{f}(x) - y\right\|^2. \quad (6)$$

and its bias/variance decomposition given by German in [3].

$$\mathrm{Err}(\mathbf{x}) = \sigma^2 + (Bias(\hat{f}(\boldsymbol{x})))^2 + Var\left(\hat{f}(\boldsymbol{x})\right). \quad (7)$$

In Eq. (7), σ^2 is the variance of y given x. The variance term could be decomposed in the following way:

$$\mathrm{Var}(\hat{f}) = \sum_{t=1}^{T} w_t^2\left(E[f_t^2] - E^2(f_t)\right) + 2\sum_{t<j} w_t w_j\left(E[f_t f_j] - E[f_t][f_j]\right). \quad (8)$$

where the expectation is taken with respect to the data set under investigation. The first sum in Eq. (8) marks the lower bound of the ensemble variance and is the weighted mean of the variances of the ensemble members. The second sum contains the cross-terms of the ensemble members and vanishes if the models are completely uncorrelated. This shows that the reduction in the variance of the ensemble is related to the degree of the independence of the single models. There are several ways to introduce model diversity to the ensemble in order to decorrelate the output of the individual ensemble members. A typical approach is to use AdaBoost algorithm as a training method to increase the diversity which have proved our idea in theory.

IV. SIMULATION RESULTS AND DISCUSSIONS

A. Phase Space Reconstruction

If we consider an equidistant sampled time series: $\{x_1, x_2, \cdots, x_i\}$, we can construct a d-dimensional state space vector $\boldsymbol{x}_i$ in the form

$$\boldsymbol{x}_i = \left(x_{(i-\lambda(d-1))}, x_{(i-\lambda(d-2))}, \cdots, x_i\right). \quad (9)$$

In Eq. (9), λ denotes the time lag. A "one-step ahead prediction" model $f(\boldsymbol{x})$ for iterated time series prediction has the form:

$$f: R^d \to R. \quad (10)$$

$$f(\boldsymbol{x}_i) = x_{i+1}. \quad (11)$$

we perform the iterated prediction in such a way, that we use the predicted value x_{n+1} to construct the next state space vector $\boldsymbol{x}_{n+1}$ which is used to predict the next time series sample x_{n+2} and so on. In the field of nonlinear time series analysis, this method was suggested by Farmer and Sidorowich [22] in order to make short term predictions of chaotic systems.

B. Predictive Performance Indices Used

The formulas of the indices use the notation $x(t)$ for the series actual value, $\hat{x}(t)$ for its predicted value, and I for the number of observations ($I = 20$).

The three metrics: mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), are calculated according to Eq. (12) — (14) respectively.

$$\mathrm{MAE} = |x(t) - \hat{x}(t)|. \quad (12)$$

$$\mathrm{MSE} = \frac{1}{I}\sum_{t=1}^{I}(x(t) - \hat{x}(t))^2. \quad (13)$$

$$RMSE = \sqrt{\frac{1}{I}\sum_{t=1}^{I}\left(x(t) - \hat{x}(t)\right)^2}. \quad (14)$$

In order to evaluate the prediction performance and compare it with the results reported in the literature, two error criterions are used for ranking different methods in the CATS Benchmark [23]. One is the mean square error E_1 (Eq. (15)) which will be computed on the five data segments using:

$$E_1 = \frac{\sum_{t=981}^{1000}(x_t - \hat{x}_t)^2}{100} + \frac{\sum_{t=1981}^{2000}(x_t - \hat{x}_t)^2}{100} + \frac{\sum_{t=2981}^{3000}(x_t - \hat{x}_t)^2}{100} + \frac{\sum_{t=3981}^{4000}(x_t - \hat{x}_t)^2}{100} + \frac{\sum_{t=4981}^{5000}(x_t - \hat{x}_t)^2}{100}. \quad (15)$$

Another is the mean square error E_2 (Eq. (16)) which will be computed on the four data segments using:

$$E_2 = \frac{\sum_{t=981}^{1000}(x_t - \hat{x}_t)^2}{80} + \frac{\sum_{t=1981}^{2000}(x_t - \hat{x}_t)^2}{80} + \frac{\sum_{t=2981}^{3000}(x_t - \hat{x}_t)^2}{80} + \frac{\sum_{t=3981}^{4000}(x_t - \hat{x}_t)^2}{80}. \quad (16)$$

The main difference between the two metrics defined by eqn. (15) and (16) is that the 5th data segment of 20 data samples is missing in the latter formula due to the unavailability of the real data for some participants in the competition [23].

C. Illustrative Example #1: Chua's Circuit time-series prediction

The data set was part of the time-series competition of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modelling in 1998 in Leuven, Belgium. It stems from a nonlinear transform of a 5-scroll generalized Chua's circuit (see [24] for a detailed description). The data set consists of 2000 points and the prediction of the following 200 data is shown in Fig. 1. To allow a comparison with previous results in this work, the value of dimension d = 50 and time lag λ = 1 are used to construct time delay vectors.

We construct an ensemble model AdaBoost.RT-FFN-BP based on 30 rounds training and the final outcome of the ensemble method is obtained. The output generated by AdaBoost.RT-FFN-BP and FFN-BP is shown in Fig. 2(a), and Fig. 2(b) shows the differences in prediction error between the ensemble model: AdaBoost.RT-FFN-BP and the base learner: FFN-BP.

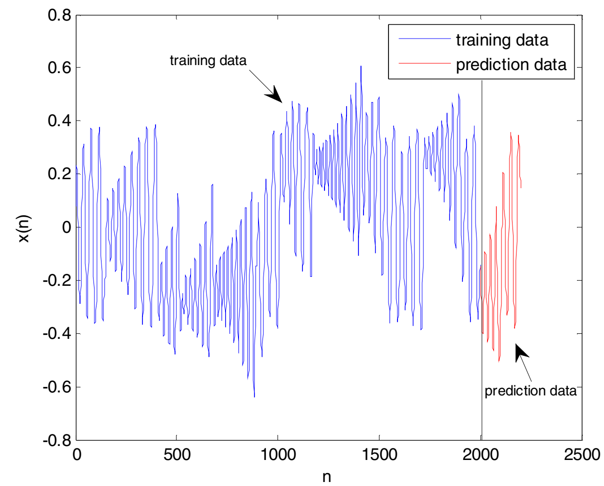

Fig.1. Chua's circuit time-series data

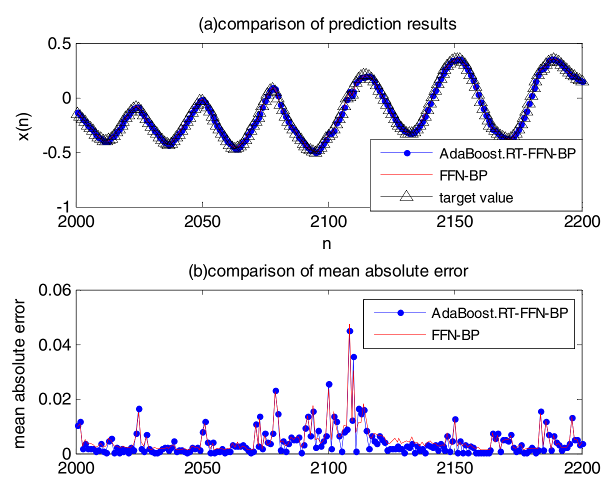

Fig.2. Comparison of the Chua's circuit time-series prediction results of AdaBoost.RT-FFN-BP and FFN-BP

The results of the simulation on Chua's circuit data are shown in Table Ⅰ. It is noteworthy that the MSE value of AdaBoost.RT- FFN-BP is 4.4499e-05 and the RMSE value is 0.0067; the MSE value of FFN-BP is 6.4729e-05 and the

RMSE value is 0.0080. It is clear that the ensemble method: AdaBoost.RT-FFN-BP performs better than the single FFN-BP.

TABLE Ⅰ
Comparison of time-series predictive performance of AdaBoost.RT-FFN-BP and FFN-BP

	FFN-BP	AdaBoost.RT-FFN-BP
MAE	0.0048	0.0040
MSE	6.4729e-05	4.4499e-05
RMSE	0.0080	0.0067

D. Illustrative Example #2: CATS Benchmark time-series prediction

The data set of the CATS Benchmark was provided by Lendasse [23] and consists of an artificial time-series with 5000 data points (see Fig. 3), wherein 100 values are missing. These missing values are divided in five blocks of 20 points that have to be predicted (circled portion). In order to cope with this complex time-series, we examined two different cases to construct the ensemble models from different choice of training datasets. Also, the ensemble size is selected to 30.

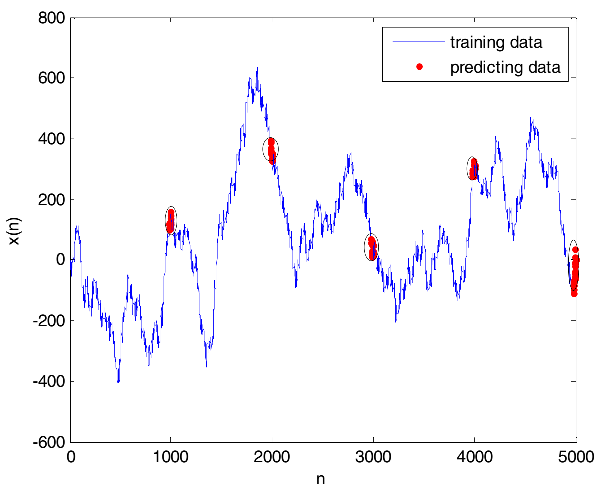

Fig.3. The CATS Benchmark

Case1:Multiple Local Models

To allow a comparison with previous results in this work, the value of dimension $d = 14$ and time lag $\lambda = 1$ are used to build time-delay vectors. The structure of FFN-BP is $14 \times 29 \times 1$ and the high-level diagram of multiple local model is shown in Fig. 4. Table Ⅱ illustrates the five segments which have been divided to predict.

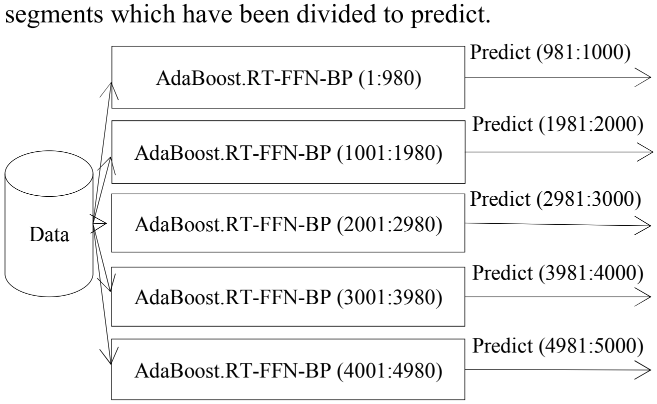

Fig.4. The structure of multiple local model

TABLE Ⅱ
Division of Training and Test (for prediction) Dataset

Segment	Training data	Predicting data
1	1:980	981:1000
2	1001:1980	1981:2000
3	2001:2980	2981:3000
4	3001:3980	3981:4000
5	4001:4980	4981:5000

The Fig. 5(a) illustrates the AdaBoost.RT-FFN-BP and FFN-BP forecast for the time series (segment 1), along with their actual data. The comparison of the forecasts against the actual data shows the differences in performances. Meanwhile, the differences in mean absolute error between the ensemble model and single model are presented in Fig. 5(b).

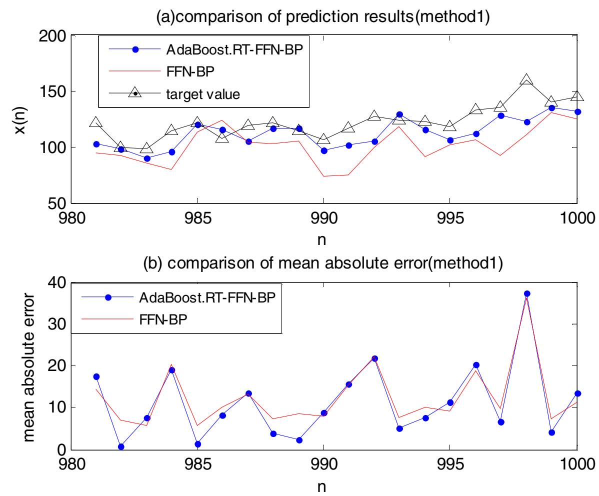

Fig.5. Comparison of time-series (segment 1) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

The Fig. 6(a) and Fig. 7(a) illustrate the AdaBoost.RT-FFN-BP and FFN-BP forecast for the time series (segment 2 and segment 3 respectively), along with their actual data. The comparison of the forecasts against the actual data shows the differences in performances. Meanwhile, the difference in mean absolute error between the ensemble model and single model are presented in Fig. 6(b) and Fig. 7(b).

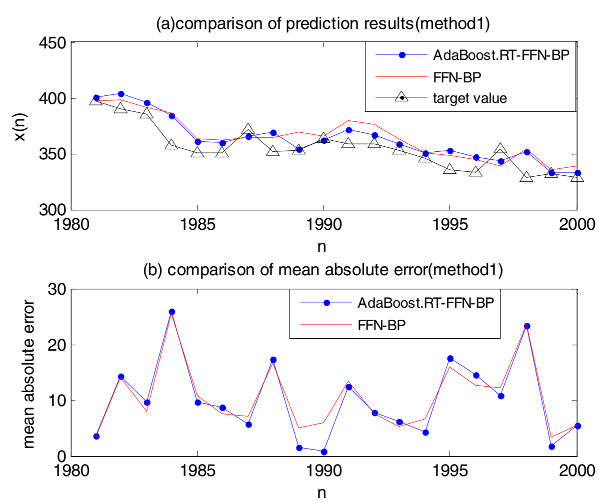

Fig.6. Comparison of time-series (segment 2) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

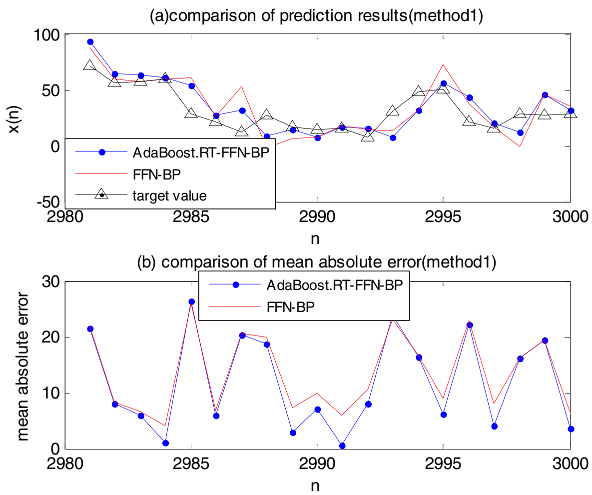

Fig.7. Comparison of time-series (segment 3) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

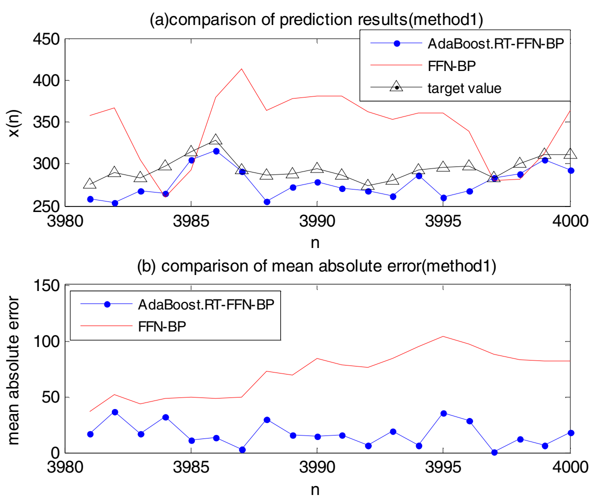

Fig.8. Comparison of time-series (segment 4) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

The Fig. 8(a) illustrates the AdaBoost.RT-FFN-BP and FFN-BP forecast for the time series (segment 4), along with their actual data. The comparison of the forecasts against the actual data shows the differences in performances. Meanwhile, the difference in mean absolute error between the ensemble model and single model is presented in Fig. 8(b).

The Fig. 9(a) illustrates the AdaBoost.RT-FFN-BP and FFN-BP forecast for the time series (segment 5), along with their actual data. The comparison of the forecasts against the actual data shows the differences in performances. Meanwhile, the difference in mean absolute error between the ensemble model and single model is presented in Fig. 9(b).

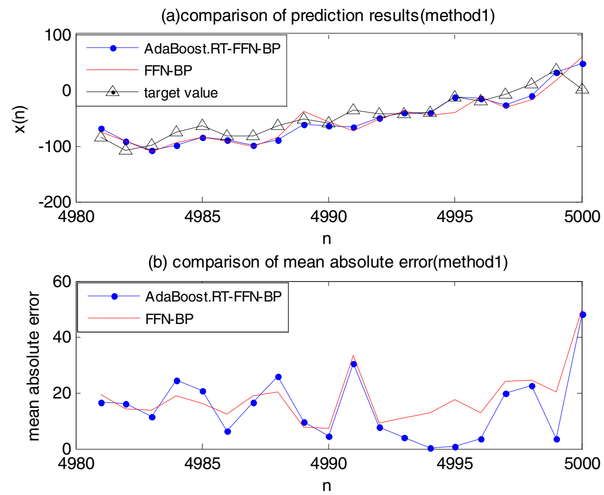

Fig.9. Comparison of time-series (segment 5) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

Table Ⅲ presents the results of the CATS benchmark time-series prediction. It is noteworthy that the best MAE value of AdaBoost.RT-FFN-BP is 10.0611, the MSE value is 148.8750 and the RMSE is 12.2014 in segment 2. In contrast, the best results, which are produced by FFN-BP, are the MAE value: 11.8036, the MSE value: 191.6047 and the RMSE value: 13.8421. From comparison, we can conclude that ensemble model performs better than single model.

TABLE Ⅲ
Comparison of the predictive results of AdaBoost.RT-FFN-BP (Ensemble 1) and FFN-BP based method 1. (case 1)

Segment	Model	MAE	MSE	RMSE
981:1000	FFN-BP	22.6435	670.8340	25.9005
	Ensemble 1	11.3187	202.8045	14.2409
1981:2000	FNN-BP	11.8036	191.6047	13.4043
	Ensemble 1	10.0611	148.8750	12.2014
2981:3000	FNN-BP	14.2491	337.1632	18.3620
	Ensemble 1	11.9159	210.6642	14.5143
3981:4000	FNN-BP	58.6956	4.99e+03	67.0270
	Ensemble 1	16.9304	394.7494	19.8683
4981:5000	FNN-BP	18.2709	497.4509	22.3036
	Ensemble 1	14.5792	350.3858	18.7186

It is unexpected, however, to observe that in segment 4 there is an obvious degradation between the performance of the AdaBoost.RT-FFN-BP and FFN-BP. In paper (Wichard et al., 2004) [4], the author pointed out the drawbacks of local model: this local strategy could only characterize local features and cannot characterize the global features of the whole time-series, so we need to optimize and change this local strategy.

Case2: A Single Global Model

The discussion in the previous section has shown that we can improve our method by using the strategy of global modeling instead of local model. In comparison to multiple local model, a single global model is proposed which cope with all the training data including the five segments. Finally, we use this single global model to predict the 100 missing data respectively which have been divided into five blocks.

To allow a comparison with previous results in this work, the value of dimension d = 14 and time lag $\lambda = 1$ are also used to build time-delay vectors. The structure of FFN-BP neural network is $14 \times 29 \times 1$ and the high-level diagram of a single global model is presented in Fig. 10.

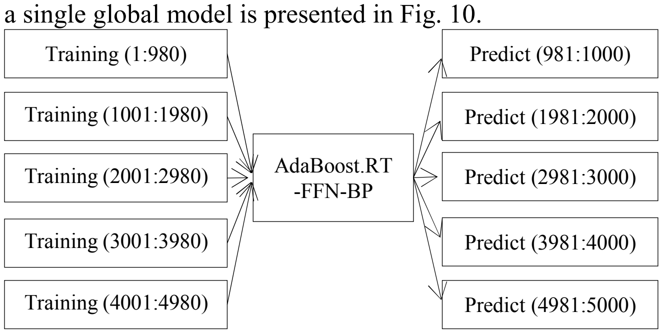

Fig.10. The structure of global model

The Fig. 11(a) illustrates the AdaBoost.RT-FFN-BP and FFN-BP forecast for the time series (segment 1), along with their actual data. The comparison of the forecasts against the actual data shows a difference in performances. Meanwhile, the differences in mean absolute error between the ensemble model and single model are presented in Fig. 11(b).

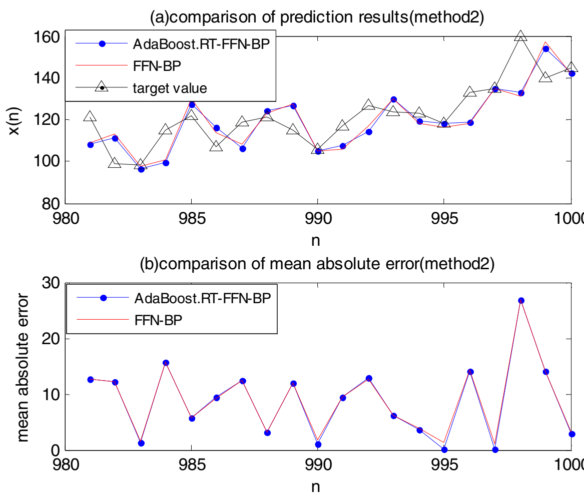

Fig.11. Comparison of time-series (segment 1) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

The Fig. 12(a) and Fig. 13(a) illustrate the AdaBoost.RT-FFN-BP and FFN-BP forecast for the time series (segment 2 and segment 3 respectively), along with their actual data. The comparison of the forecasts against the actual data shows the differences in performances.

Meanwhile, the differences in mean absolute error between the ensemble model and single model are presented in Fig. 12(b) and Fig. 13(b).

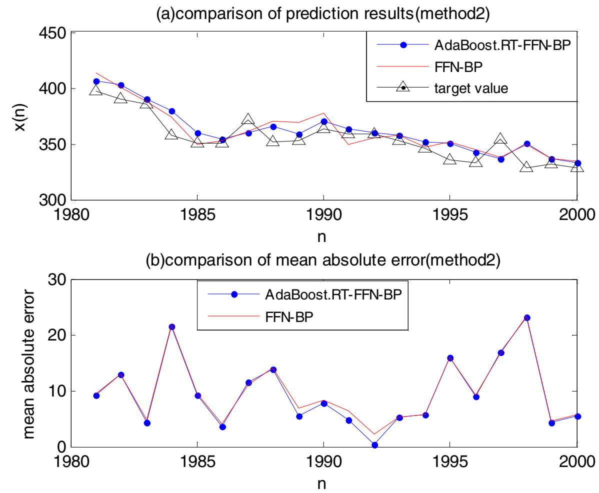

Fig.12. Comparison of time-series (segment 2) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

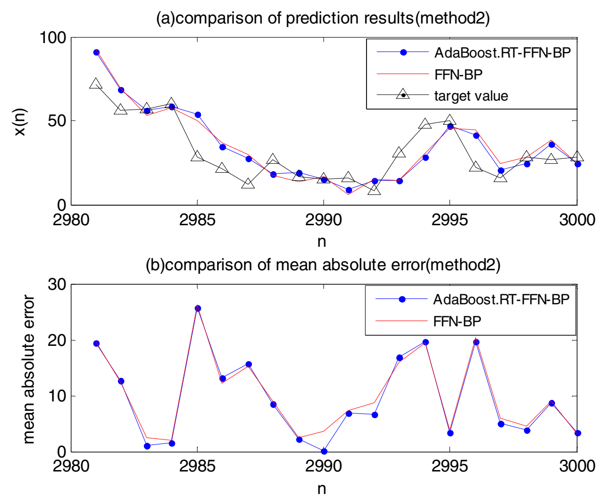

Fig.13. Comparison of time-series (segment 3) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

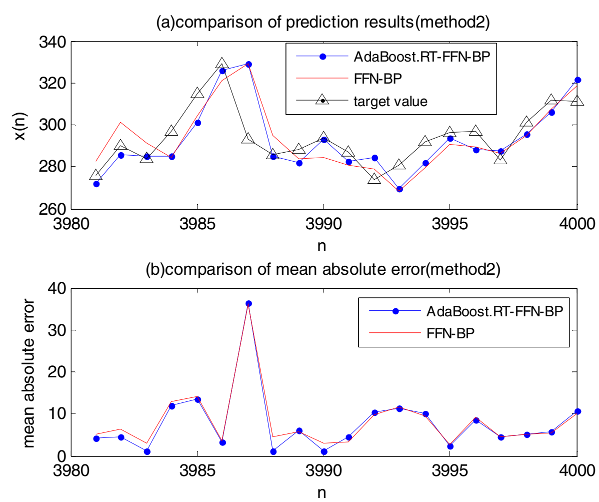

Fig.14. Comparison of time-series (segment 4) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

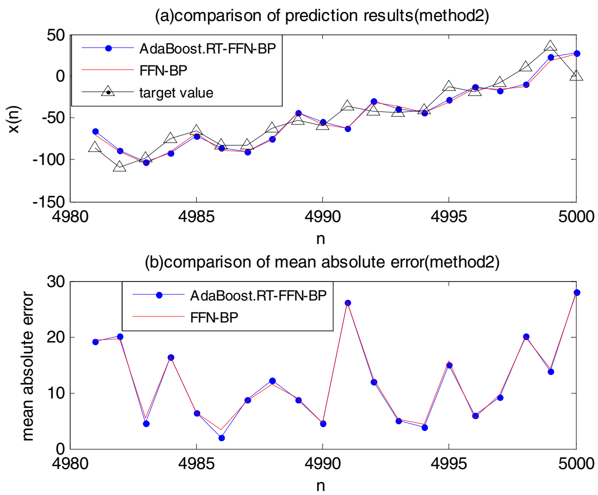

Fig.15. Comparison of time-series (segment 5) prediction results of AdaBoost.RT-FFN-BP and FFN-BP

The Fig. 14(a) and Fig. 15(a) illustrate the AdaBoost.RT-FFN-BP and FFN-BP forecast for the time series (segment 4 and segment 5 respectively), along with their actual data. The comparison of the forecasts against the actual data shows the differences in performances. Meanwhile, the differences in mean absolute error between the ensemble model and single model are presented in Fig. 14(b) and Fig. 15(b).

The results of the CATS benchmark time-series are shown in Table Ⅳ. It is noteworthy that the best MAE value of AdaBoost.RT-FFN-BP is 7.8029, the MSE value is 118.3892 and the RMSE value is 10.8807 in segment 4. In contrast, the best results, which are produced by FFN-BP, are the MAE value: 9.0218, the MSE value: 129.6210and the RMSE value: 11.3851.

For comparison, as in Table Ⅲ and Table Ⅳ, we can see that the best results in Global Model are MAE: 7.8029, MSE: 118.3892, RMSE: 10.8807 while the best result obtained from local model are MAE: 10.0611, MSE: 148.8750, RMSE: 12.2014. Accordingly, the single global model performs better than multiple local model.

TABLE Ⅳ
Comparison of predictive results of AdaBoost.RT-FFN-BP (Ensemble 2) and FFN-BP based method 2. (case 2)

Segment	Model	MAE	MSE	RMSE
981:1000	FFN-BP	9.0218	129.6210	11.3851
	Ensemble 2	8.8165	121.0322	11.0015
1981:2000	FFN-BP	10.1799	143.6330	11.9847
	Ensemble 2	9.5166	126.7921	11.2602
2981:3000	FFN-BP	10.5531	161.0653	12.6912
	Ensemble 2	9.7116	149.8367	12.2408
3981:4000	FFN-BP	9.2884	132.6701	11.5183
	Ensemble 2	7.8029	118.3892	10.8807
4981:5000	FFN-BP	12.2897	205.5321	14.3364
	Ensemble 2	12.0716	200.2973	14.1526

The error indices E_1 for the missing 100 points and the mean square error E_2 for the missing 80 points in the CATS Benchmark time series are compared with the results reported in the literature and presented in Table Ⅴ and Table Ⅵ. The experiment performed for the CATS benchmark confirms that the performance of the proposed method is better in prediction when compared to other prediction methods reported in the literature.

TABLE Ⅴ
Comparison of the predictive performance of the former methods and our method

Model	E_1	Reference
Kalman Smoother	408	[10]
Recurrent Neural Networks	441	[9]
Competitive Associative Net	502	[25]
Weighted Bidirectional Multi-stream Extended Kalman Filter	530	[26]
SVCA Model	577	[14]
MultiGrid-Based Fuzzy System	644	[27]
Double Quantization Forecasting Method	653	[11]
Time-reversal Symmetry Method	660	[28]
BYY Harmony Learning Based Mixture of Experts Model	676	[29]
Ensemble Models	725	[5]
Chaotic Neural Networks	928	[7]
Evolvable Block-based Neural Networks	954	[8]
Time-line Hidden Markov Experts	1037	[30]
Fuzzy Inductive Reasoning	1050	[31]
Business Forecasting Approach to Mulitlayer Perceptron Modelling	1156	[12]
A hierarchical Bayesian Learning Scheme for Autoregressive Neural Network	1247	[13]
Hybrid Predictor	1425	[32]
Multiple local models(case1)	261.5	
Single global model(case2)	143.3	

TABLE Ⅵ
Comparison of the predictive performance of the former methods and our method

Model	E_2	Reference
Ensemble Models	222	[5]
Fuzzy Inductive Reasoning	278	[31]
Kalman Smoother	346	[10]
Double Quantization Forecasting Method	351	[11]
Weighted Bidirectional Multi-stream Extended Kalman Filter	370	[26]
SVCA Model	395	[14]
Recurrent Neural Networks	402	[9]
Time-line Hidden Markov Experts	402	[30]
Competitive Associative Net	418	[25]
Time-reversal Symmetry Method	442	[28]
MultiGrid-Based Fuzzy System	542	[27]
BYY Harmony Learning Based Mixture of Experts Model	677	[29]
Chaotic Neural Networks	762	[7]
Hybrid Predictor	894	[32]
Evolvable Block-based Neural Networks	994	[8]
Business Forecasting Approach to Mulitlayer Perceptron Modelling	995	[12]
A hierarchical Bayesian Learning Scheme for Autoregressive Neural Network	1229	[13]
Multiple local models(case1)	239.3	
Single global model(case2)	129.0	

An additional aspect to be measured is the differences in performance between the two similar methods: ensemble models [5] and the methods used in this paper. The major difference between the two methods is complexity.

Considering the suggestions that the more complexity the models are, the more harder the ensemble model fit the test data and will not generalize well [15], that is why the method used in this paper performs better in the results on the test data.

V. CONCLUSIONS

In this work, two famous nonlinear time series are predicted by the ensemble method. The time-series prediction method uses a combination of AdaBoost.RT algorithm, feed forward networks and phase space reconstruction. The original time series data is unfolded with embedding dimension and time delay and reconstructed into the phase space. An ensembling method, that is AdaBoost.RT-FFN-BP, is used to predict future values of the embedded phase-space points.

Despite its simplicity in implementation, the ensemble model exhibited a superior performance in prediction of chaotic time series. Based the simulation results presented above, we can conclude that the AdaBoost.RT-FFN-BP ensemble method is suitable for time-series prediction in terms of smaller predictive error than that of the established methods.

In the future, we would consider to use different types of "*weak*" learners and to incorporate fuzzy logic so as to further improve the time-series predictive performance achieved in the present investigation.

ACKNOWLEDGMENT

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

REFERENCES

[1] D. Karunasinghea, S. Y. Liongb, "Chaotic time series prediction with a global model: Artificial neural network," *Journal of Hydrology*, vol. 323, no. 1-4, pp. 92-105, 2006.

[2] D. M. Wang, L. Wang and G. M. Zhang, "Short-term wind speed forecast model for wind farms based on genetic BP neural network," *Journal of Zhejiang University (Engineering Science)*, vol. 46, no. 5, pp. 837-842, 2012.

[3] S. Geman, E. Bienenstock and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, pp. 1-58, 1992.

[4] J. D. Wichard and M. Ogorzalek, "Time Series Prediction with Ensemble Models," *In Proceedings of International Joint Conference on Neural Networks*, vol. 2, pp. 1625-1630, 2004.

[5] J. D. Wichard and M. Ogorzalek, "Time Series Prediction with Ensemble Models Applied to the CATS Benchmark," *Neurocomputing*, vol. 70, pp. 2371 – 2378, 2007.

[6] D. P. Solomatine and D. L. Shrestha, "AdaBoost.RT: a boosting algorithm for regression problems," *In Proceedings of International Joint Conference on Neural Network*, vol. 2, pp. 1163-1168, 2004.

[7] R. Kozma and I. Beliaev, "Time Series Prediction Using Chaotic Neural Networks: Case Study of IJCNN CATS Benchmark Test," *In Proceedings of the International Joint Conference on Neural Network*, vol. 2, pp. 1609-1613, 2004.

[8] S. G. Kong, "Time Series Prediction with Evolvable Block-based Neural Networks," *In Proceedings of the International Joint Conference on Neural Network*, vol. 2, pp. 1579-1583, 2004.

[9] X. D. Cai, "Time series prediction with recurrent neural networks trained by a hybrid PSO-EA algorithm," *Neurocomputing*, vol. 70, no. 13/15, pp. 2342-2353, 2007.

[10] S. Sarkka, A. Vehtari and J. Lampinen, "Time Series Prediction by Kalman Smoother with Cross Validated Noise Density," *In Proceedings of the International Joint Conference on Neural Network*, vol. 2, pp. 1653-1657, 2004.

[11] G. Simon, J. A. Lee, M. Verleysen and M. Cottrell, "Double Quantization Forecasting Method for Filling Missing Data in the CATS Time Series," *In Proceedings of the International Joint Conference on Neural Network,* vol. 2, pp. 1635-1640, 2004.

[12] S. F. Crone, H. Kausch and D. Pressmar, "Prediction of the CATS benchmark using a Business Forecasting Approach to Multilayer Perceptron Modelling," *In Proceedings of the International Joint Conference on Neural Network*, vol. 4, pp. 2783-2788, 2004.

[13] A. Eleuteri, F. Acemese, F. Barone, R. De. Rosa and L. Milano, "A hierarchical Bayesian leairning scheme for autoregressive neural networks:application to the CATS benchmark," *In Proceedings of the International Joint Conference on Neural Network*, vol. 2, pp. 1585-1590, 2004.

[14] F. Palacios-Gonzalez, "A SVCA Model for The Competition on Artificial Time Series," *In Proceedings of the International Joint Conference on Neural Network*, vol. 4, pp. 2777-2782, 2004.

[15] T. Hastie, R. Tibshirani and J. Friedman, "The Elements of Statistical Learning," *Springer Series in Statistics*, Springer, Berlin, 2001.

[16] H. Y. Wang and G. D. Shi, "Artificial Neural Network Technology and Its Applications," *China petrochemical Press*, pp. 34-36, 2002.

[17] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm" *In: Proceedings of the international conference on Machine Learning*, pp. 148-156, 1996.

[18] Y. Freund and R. E. Schapire, "A decision theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.

[19] D. L. Shrestha and D. P. Solomatine, "Experiments with AdaBoost.RT, an improved boosting scheme for regression," *Neural Computation*, vol. 18, no. 7, pp. 1678-1710, 2006.

[20] A. Krogh and P. Sollich, "Statistical mechanics of ensemble learning," *Phy. Rev. E*, vol. 55, no. 1, pp. 811-825, 1997.

[21] U. Naftaly, N. Intrator and D. Horn, "Optimal ensemble averaging of neural networks," *Network, Comput. Neural Syst,* vol. 8, pp. 283-296, 1997.

[22] J. Farmer and J. Sidorowich, "Predicting chaotic time series," *Phys. Rev. Lett*. vol. 59, no. 8, pp. 845-848, 1987.

[23] A. Lendasse, E. Oja, O. Simula and M. Verleysen, "Time series prediction competition: the CATS Benchmark," *In Proceedings of the International Joint Conference on Neural Networks*, vol. 2, pp. 1615-1620, 2004.

[24] J. McNames, J. Suykens, and J. Vandewalle, "Winning Entry of the K.U.Leuven Time Series Prediction Competition," *International Journal of Bifurcation and Chaos*, vol. 9, no. 8, pp. 1485-1500, 1999.

[25] S. Kurogi, M. Sawa and T. Yeno, "Time series prediction of the CATS benchmark using Fourier bandpass filters and competitive associative nets," *Neurocomputing*, vol. 70, pp. 2354 – 2362, 2007.

[26] X. Hu and D. Wunsch, "Time Series Prediction with a Weighted Bidirectional Multi-stream Extended Kalman Filter," *Neurocomputing*, vol. 70, pp. 2392 – 2399, 2007.

[27] L. J. Herrera, H. Pomares, I. Rojas, A. Guillen, J. Gonzalez and M. Awad, A. Herrera, "MultiGrid-Based Fuzzy Systems for Time Series prediction: CATS Competition," *Neurocomputing*, vol. 70, no. 13, pp. 2410-2425, 2004.

[28] P. F. Verdes, P. M. Granitto, M. I. Szeliga, A. Rebola and H. A. Ceccatt, "Prediction of the CATS benchmark exploiting time-reversal symmetry," *In Proceedings of the International Joint Conference on Neural Network*, vol. 2, pp. 1631-1634, 2004.

[29] H. W. Chan, W. C. Leung, K. C. Chiu and L. Xu, "BYY Harmony Learning Based Mixture of Experts Model for Non-stationary Time Series Prediction," *In Proceedings of the International Joint Conference on Neural Network*, 2004.

[30] X. Wang, "Time-line Hidden Markov Experts for the Prediction of CATS time series," *In Proceedings of the International Joint Conference on Neural Network*, 2004.

[31] Fr. E. Cellier and A. Nebot, "Multi-resolution Time-Series Prediction Using Fuzzy Inductive Reasoning," *In Proceedings of the International Joint Conference on Neural Network*, Budapest, vol. 2, pp. 1621-1624, 2004.

[32] C. Y. Ping, W. S. Nan, and S. H. Shing, "A Hybrid Predictor for Time Series Prediction," *In Proceedings of the International Joint Conference on Neural Network*, vol. 2, pp. 1597-1602, 2004.