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

# Forecasting Time Series - A Layered Ensemble Architecture

Md. Mustafizur Rahman, Shubhra Kanti Karmaker Santu, Md. Monirul Islam and Kazuyuki Murase

***Abstract*— Time series forecasting (TSF) have been widely used in many application areas such as science, engineering and finance. The characteristics of phenomenon generating a series are usually unknown and information available for forecasting is only limited to the past values of the series. It is, therefore, necessary to use an appropriate number of past values, termed *lag*, for forecasting. This paper presents a layered ensemble architecture (LEA) for TSF problems. Our architecture is consisted of two layers, each of which uses an ensemble of neural networks. Unlike most previous studies on TSF, LEA puts emphasis on both accuracy and diversity among individual networks in an ensemble. While the ensemble of the first layer tries to find an appropriate lag of a given time series, it of the second layer makes forecasting using the obtained lag. The use of the appropriate lag signifies LEA's effort in producing accurate networks for constructing the ensemble. In order to maintain diversity among networks in the ensemble, LEA trains each network in the ensemble using a different training set. The proposed architecture uses a clustering based selection method that considers both accuracy and diversity in selecting networks to construct the ensemble. Accuracy is maintained here by selecting the best networks from each cluster. On the other hand, diversity is ensured by using the variance information in constructing clusters. LEA has been tested extensively on the time series data sets of NN3 competition. In terms of prediction accuracy, our experimental results have showed clearly that LEA is better than other ensemble and non-ensemble algorithms.**

## I. Introduction

TIME series forecasting (TSF) has been widely used in many application areas such as science, engineering and finance. It is the use of a model or technique to predict future values based on previously observed values. Generally, the characteristics of phenomenon generating time series are unknown and information available for forecasting is only limited to the past values of the series. It is thus important to use an appropriate number of past values, termed *lag*, for forecasting [1], [2].

The immense interests for understanding and predicting the future gives us many forecasting methods; most of them are relying on linear and non-liner statistical models [3], [4]. The limitations of the statistical models make the multi-layer perceptron (MLP) network, a kind of neural network, as a promising alternative to the forecasting society [5], [6], [7]. Although it has been shown that the MLP network is an universal approximator, no general guideline exists in choosing the appropriate network structure for a given problem. An ensemble of networks that brings together several networks to provide a single output alleviates the difficulty associated with the conventional design strategy for designing a single network with appropriate parameters.

The main issue in ensemble approaches is the consideration of accuracy and diversity of individual networks (base predictors) that constitute an ensemble [8]. Existing ensemble approaches for TSF problems employ different techniques for maintaining diversity without considering accuracy of the base predictors. For example, some work uses bagging [9], boosting [10] or a combination of bagging and random subspace [1] to create a different training set for each base predictor. A different lag is also used for creating a different training set [11]. The authors in [12] use different training parameters to generate different base predictors. In [13], different types of base predictors are used for constructing ensembles. A careful scrutiny of the existing methods reveals that they emphasize diversity either by using different training sets, different parameters of the base predictors or different types of base predictors.

This paper proposes a layered ensemble architecture (LEA) for TSF. Our LEA is consisted of two layers; each of which uses an ensemble of neural networks. The essence of LEA is that it considers both accuracy and diversity not only in generating individual networks, but also in combining the networks to construct ensembles. The proposed architecture has been tested extensively on the time series data of NN3 competition [14].

The rest of this paper is organized as follows. Section II describes our LEA algorithm in details. Section III presents results of our experimental study. Finally, section IV concludes the paper with a few remarks for future directions.

## II. Layered Ensemble Architecture

In order to reduce the detrimental effect of using a predefined lag and to devise an efficient forecasting scheme, a layered ensemble approach, LEA, is adopted in this work. Ensembles' requirement for maintaining diversity and accuracy among the base predictors match well with our emphasis on using a layered architecture. In its current implementation, LEA uses MLP networks as base predictors.

The major steps of LEA can be explained as follows.

1) Preprocess data of a given time series for handling seasonality, noise and missing attribute values.
2) Hold out $m$ data points (observations) for testing LEA and use the remaining observations for constructing a forecasting model.
3) **Ensemble Layer 1**
   a) Generate an ensemble consisting of $N$ MLP networks. Here $N$ is a user-defined parameter and

Md. Mustafizur Rahman, Shubhra Kanti Karmaker Santu and Md. Monirul Islam are with the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh. (email: {mustafiz_rahman, kantishubhra, mdmonirulislam}@cse.buet.ac.bd).

Kazuyuki Murase is with the Department of Human and Artificial Intelligence Systems, University of Fukui, Bunkyo, Fukui, Japan. (email: murase@u-fukui.ac.jp).

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

greater than $l_{max}$, the maximum lag of the series. For example, $l_{max}$ can be 12 for a monthly time series.

b) Assign a random lag, $l_i$, to the network $i$ of the ensemble. This can be done by generating a number uniformly at random between 1 and $l_{max}$.

c) Define the architecture of each network in the ensemble. The network has an input layer, a hidden layer and an output layer. The number of nodes in the input and hidden layers equals to the lag assigned to the network, while the number of nodes in the output layer is one.

d) Create $N$ training sets, one for each network, using the lags assigned to all $N$ networks in the ensemble.

e) Train each network in the ensemble on the training set generated for it by using the Levenberg-Marquardt (LM) algorithm [15].

f) Evaluate each network in the ensemble on a validation set containing $n$ data points. The symmetric mean absolute percent error (*sMAPE*) is used for evaluation. According to [14], *sMAPE* can be expressed as

$$sMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{(Y_i + \hat{Y}_i)/2} \times 100 \qquad (1)$$

where $Y_i$ and $\hat{Y}_i$ are the true and predicted values for the $i$-th data point, respectively. The above equation provides a value between $0\%$ and $200\%$. The smaller the *sMAPE* is, the better the prediction accuracy is. We use *sMAPE* because it is used in many previous studies (e.g. [16], [17]) and forecasting competitions (e.g. NN3 [14], NN5 [18]). However, any other performance measure can be used for evaluating the networks.

g) Obtain the ensemble output by combining the outputs of the individual networks. We use a model selection and combination method (Algorithm 1) to obtain the ensemble output. Since the aim of the first layer is to find appropriate lag, the ensemble output for this layer will be a lag.

4) **Ensemble Layer 2**

a) Generate an ensemble consisting of $N$ MLP networks.

b) Assign the same lag, which is obtained by the first layer, to each network in the ensemble.

c) Define the architectures of networks in the same way as described in the step 3c. Note that the architectures of all $N$ networks in this layer will be same, because the lag assigned to all the networks is same.

d) Create a training set, $D_{tr}$, using the lag.

e) Train each network, $j$, in the ensemble on a training subset, $D^j_{tr}$, using the LM algorithm [15]. The training subset is to be created from $D_{tr}$ by using a re-sampling technique (e.g. bagging [19] or boosting [20]).

f) Use the model selection and combination scheme (Algorithm 2) to obtain the output of the ensemble, which will indicate the final forecast.

The above layered ensemble architecture appears to be straight forward, but its essence is the techniques incorporated for maintaining accuracy and diversity among the base predictors of an ensembles. Our LEA also exhibits some additional features. Details of them are given in the following sections.

### A. Data Preprocessing

*1) Noise Removal:* In many TSF problems (e.g. time series of NN3 [14]), data points in the time series are heavily influenced by noise. A data point is considered as noise, if its value is very much different from other values of the series. Failure to take appropriate measures against noise may lead to a bad forecasting performance.

There are several ways by which one can remove noise from the time series data. For example, a large second order difference value is used as an indication of noise in [21]. In this paper, we use a noise detection mechanism proposed in [11], where a data point is identified as a noise whose absolute value is four times greater than the absolute medians of the three consecutive points before and after that point. That is, $Y_i$ is a noise if it satisfies the condition: $Y_i \geq 4 \times max\{|m_a|, |m_b|\}$, where $m_a = median(Y_{i-3}, Y_{i-2}, Y_{i-1})$ and $m_b = median(Y_{i+3}, Y_{i+2}, Y_{i+1})$. When a data point is identified as a noise, its value is simply replaced by the average value of the two points that are immediately before and after it.

*2) Deseasonalization:* Treatment of seasonality is one of the major issues in TSF literature, because many time series data [14], [18] contain some seasonality. There are basically two methods, namely, direct and deseasonalized, for handling seasonality [22]. In the direct method, the base predictors are trained directly on the raw data, whereas in the latter method, seasonal adjustments are made on the raw data before the predictors are trained on.

For the seasonal series, we adopt a simple deseasonalization procedure suggested in [23] which simply subtracts the seasonal average from the series to obtain a deseasonalized series. To make final forecast, we restore back the seasonality to provide the output of the forecasting model.

### B. Training Set Generation

The aim of our ensemble layer 1 is to find the appropriate lag. Lacking of knowledge about such a lag enforces LEA to vary the lag from 1 to $l_{max}$ and LEA generates a different training set using each of different lags. Let the lag equals to 5 and the data points $d_1, d_2, \ldots, d_k$ are used for generating the training set. The generation process takes the lag as a window and shifts it in generating the training set. That is, $X_1 = d_1 \ldots d_5$ and $Y_1 = d_6$, $X_2 = d_2 \ldots d_6$ and $Y_2 = d_7$ and this process continues until the $Y_i$ reaches at the end of

the series i.e., $d_k$. Here $X_i$ represents the training set and $Y_i$ is the target output.

In the ensemble layer 2, the training sets for base predictors are generated in two steps. At the first step, using the lag obtained from the ensemble layer 1 and the data points $d_1, d_2, \ldots, d_k$, LEA generates the training set, $D_{tr}$. In the second step, bootstrapped sampling (e.g. bagging [19], boosting [24]) is applied on $D_{tr}$ for generating $N$ training subsets $D_{tr}^j$, where $j = 1, 2, \ldots, N$.

*Algorithm 1:* Model selection and combination algorithm in ensemble layer 1

**Require:** Ensemble size $N$, number of clusters $c$.
**for** $i = 1$ to $N$ **do**
    calculate the variance of each MLPs $N_i$.
**end for**
Cluster the $N$ MLPs into $c$ classes ($c < N$) according to their variance value.
Select one MLP from each cluster which has lowest *sMAPE* over validation set in that cluster.
For each selected MLP record the corresponding lag which is assigned to that MLP.
Provide the final lag using the average of the lag of the selected MLP and then apply flooring on the average value to obtain an integer value.

**end**

### C. Model Selection for Ensemble

Since LEA has two different layers of ensembles, it is necessary to use a proper selection and combination method so that the objective of each layer can be achieved. To provide a better forecast, we improvise a clustering based model selection and combination method. The essence of our method is that it considers both accuracy and diversity of the MLP networks so that they could constitute a good ensemble.

We use variance, i.e., the performance variation of an MLP network due to the variation of a data set, for ensuring diversity. The computation of variance is quite straight forward. For each MLP, we first vary its training set by adding random noise and calculate how much error it makes. Then calculate variance based on the errors. We form clusters using the variance information of all the generated networks. To construct an ensemble, we select the best MLP network from each cluster. It indicates our objective of considering accuracy in forming ensembles. For ensemble layer 1, we select the best network based on *sMAPE* from each cluster and also record the associated lag of that MLP. Since the target of ensemble layer 1 is the optimal lag for a time series we simply take the average of the lag of MLPs. To obtain an integer value of that average, we simply apply flooring on the resultant value. For the ensemble layer 2, we select the best network based on *sMAPE* from each cluster and combine all the selected networks by weighted average. The weight of a network is inversely proportional to its corresponding *sMAPE* over the validation set. The pseudo code of the whole process is given in Algorithm 1 and 2.

*Algorithm 2:* Model selection and combination algorithm in ensemble layer 2

**Require:** Ensemble size $N$, number of clusters $c$.
**for** $i = 1$ to $N$ **do**
    calculate the variance of each MLPs $N_i$.
**end for**
Cluster the $N$ MLPs into $c$ classes ($c < N$) according to their variance value.
Select one MLP from each cluster which has lowest *sMAPE* over validation set in that cluster.
For each selected MLP calculate combination weight using their *sMAPE*.
For each selected MLP perform out-of-sample prediction.
Provide the final out-of-sample forecast using weighted average.

**end**

### D. Accuracy and Diversity

Our LEA considers both accuracy and diversity in its two layers of ensembles. As mentioned earlier, LEA uses MLP networks as the base predictors in constructing ensembles. In the ensemble layer 1, it uses different lags for different networks for finding an appropriate lag of a given time series. To ensure accuracy among the networks of this layer, we use the same lag and architecture for several networks, since $l$ is randomly chosen in $[1, l_{max}]$. We thus choose N $>>$ $l_{max}$ with an hope that some of the networks will exhibit better performance compared to others. This will help in reaching towards the goal of the ensemble layer 1, i.e., finding an optimum or a near optimal lag. Since the lag parameter is vital for an accurate forecast, LEA uses the best lag obtained by the ensemble layer 1 and its associated network architecture for the networks of the ensemble layer 2. Note that the task of the ensemble layer 2 is forecasting. Furthermore, our selection and combination methods (algorithm 1 and 2) choose the best network from each cluster to construct ensembles for the layer 1 and 2. All these techniques indicate LEA's effort in encouraging accuracy of the networks.

LEA uses a different lag for each network in the ensemble layer 1. As the lag parameter determines the number of input neurons in the MLP networks and the construction of the training data, these two properties will ensure diversity among the networks of the first layer. The diversity among the networks of the second layer is ensured by training them using different training sets. In addition to that LEA uses a combination strategy that encourages diversity by using variance in selecting networks for constructing ensembles (Algorithm 1 and 2). In summary, all the techniques incorporated in LEA serve the three important points for TSF. Firstly, LEA preserves the autocorrelation information of the time series by using the appropriate lag. Secondly, it helps to construct an ensemble by combining accurate and diverse members. Thirdly, it increases the probability to get better

final forecast.

## III. Experimental Studies

In this section, we evaluate and compare the performance of LEA using the NN3 time series competition dataset [14]. Organizers of this competition categorize the time series into long and short based on the length of series. The short series contains less than 50 data points, while the long one contains more than 50 data points. According to the characteristics of data, the time series can be further categorized into seasonal and non seasonal. In a seasonal time series, there exists regularly spaced peaks and troughs that have a consistent direction and approximately have the same magnitude at every period. These are, however, not present in a non-seasonal one.

### A. Performance Measure

The performance of a forecasting model is usually evaluated by some accuracy measure. The *sMAPE* is used in NN3 competition. However, to make exhaustive evaluation, we use Median Root Absolute Error (*MdRAE*) and Mean Absolute Scaled Error (*MASE*) in addition to *sMAPE*. The *MdRAE* and *MASE* measures can be expressed as

$$MdRAE = median(|r_i|), \quad r_i = \frac{Y_i - \hat{Y}_i}{Y_i + \hat{Y}_i^*} \tag{2}$$

$$MASE = \frac{\sum_{i=1}^{n} |e_i|}{\frac{n}{n-1}\sum_{i=2}^{n} |Y_i - Y_{i-1}|}, \quad e_i = Y_i - \hat{Y}_i \tag{3}$$

where $\hat{Y}_t^*$ is the forecast made by a reference method, i.e., random walk [25], applied on the series data for a given forecast horizon $h$.

### B. Experimental Setup

A MLP network containing one hidden layer with the tanh sigmoid activation function and linear activation function for the output nodes is used as the base predictor for the ensemble layers 1 and 2. The ensemble in both the layer 1 and 2 are consisted of 50 MLP networks. According to [14], we withhold the last 18 data points of every time series for testing and use the remaining data points for building forecasting models. The layer based ensemble architecture for TSF proposed in this paper is implemented using MATLAB (R2012a, The Mathworks, Inc., Natick, MA, USA).

## IV. Results

We first compare our LEA with basic bagging to show the effect of layering in the ensemble design for TSF problem. We then compare LEA with several other ensemble, non-ensemble and benchmark statistical methods.

### A. Comparison with bagging

As mentioned in section II, LEA re-samples $b\%$ data points to create different training sets for different base predictors of the ensemble layer 2. To make a fair comparison with bagging, LEA uses here bagging as a method for re-sampling. We call this version of LEA as layered bagging. The value of $b$ used for LEA is set to 9%.

Tables I-IV and Figs. 1-3 show the results of basic bagging and layered bagging. Several observations can be made from the results summarized in the tables.

- It can be seen that the lag obtained by the ensemble layer 1 of LEA is different for different time series (Table I). These results indicate that it is very important to determine the lag of a given time series automatically. Note that the basic bagging algorithm uses the same lag for all time series. According to the suggestion of NN3 [14], the lag is set to 12 for basic bagging.
- In terms of average *sMAPE*, *MASE* and *MdRAE*, layered bagging is found better than basic bagging irrespective of the nature of time series i.e., seasonal, non-seasonal, long or short. The performance of layered bagging is found more consistent compared to bagging across different time series. This can be observed by looking the standard deviation (std) of these two methods (Tables II-III).
- To get an idea about the performance on different time series, we count the number of times layered bagging is better (or worse) compared to bagging. Using any error measurement, if layered bagging exhibits better performance than bagging for a specific time series, we call it as a win for layered bagging; otherwise, it is a loss. Once again, layered bagging is proved superior than basic bagging with respect to three different performance metrics (Table IV).
- We present the series wise performance of the layered bagging and the basic bagging in Figs. 1-3, where the numbers 1, 2,...,103 used for identifying 103 different time series. It can be observed from these figures that the errors of the basic bagging surround those of the layered bagging in all most all case irrespective of the performance metric used. That is, the basic bagging produces more error compared to the layered bagging for most of the time series we consider in this study.

We use the Wilcoxon-signed-rank test to assess whether the performance difference between the layered bagging and the basic bagging is statistically significant. Table V shows the summary of the Wilcoxon-signed-rank test based on the *sMAPE*, *MASE* and *MdRAE* for the four different types time series. Here, $R^+$ corresponds to the sum of ranks for layered bagging and $R^-$ for the basic bagging algorithm. The results show that the null hypothesis has been rejected in favor of layered bagging with a significance level 0.05 for seasonal, non-seasonal, short and long time series.

TABLE I

LAG OBTAINED FROM THE ENSEMBLE LAYER 1 OF LEA FOR DIFFERENT TIME SERIES DATA OF NN3 [14] COMPETITION

| | | seasonal | non seasonal | short | long |
|---|---|---|---|---|---|
| LEA | mean | 7.3529 | 6.7674 | 7.0600 | 7.1803 |
| | minimum | 1 | 1 | 1 | 1 |
| | maximum | 12 | 12 | 12 | 12 |
| | std | 3.2816 | 3.3441 | 3.3649 | 3.2788 |

TABLE II

COMPARISON BETWEEN BASIC BAGGING AND LAYERED BAGGING IN TERMS OF *sMAPE*, *MASE* AND *MdRAE* FOR SEASONAL AND NON-SEASONAL TIME SERIES DATA OF NN3 [14] COMPETITION. HERE THE BEST RESULT IS HIGHLIGHTED USING BOLDFACE TEXT.

| | | Basic Bagging | | | Layered Bagging | | |
|---|---|---|---|---|---|---|---|
| | | sMAPE | MASE | MdRAE | sMAPE | MASE | MdRAE |
| seasonal | mean | 14.220 | 1.380 | 0.618 | **13.270** | **1.242** | **0.546** |
| | min | 0.875 | 0.484 | 0.206 | **0.806** | **0.442** | **0.185** |
| | max | **60.584** | 8.324 | 1.291 | 61.140 | **4.886** | **1.096** |
| | std | **11.784** | 1.174 | 0.246 | 12.023 | **0.907** | **0.236** |
| non seasonal | mean | 17.963 | 0.940 | 0.694 | **16.612** | **0.686** | **0.571** |
| | min | 5.819 | 0.499 | 0.289 | **4.198** | **0.296** | **0.204** |
| | max | **76.622** | 2.516 | 2.227 | 81.617 | **1.724** | **0.977** |
| | std | 13.703 | 0.501 | 0.340 | **13.633** | **0.273** | **0.209** |

TABLE III

COMPARISON BETWEEN BASIC BAGGING AND LAYERED BAGGING IN TERMS OF *sMAPE*, *MASE* AND *MdRAE* FOR SHORT AND LONG TIME SERIES DATA OF NN3 [14] COMPETITION. HERE THE BEST RESULT IS HIGHLIGHTED USING BOLDFACE TEXT.

| | | Basic Bagging | | | Layered Bagging | | |
|---|---|---|---|---|---|---|---|
| | | sMAPE | MASE | MdRAE | sMAPE | MASE | MdRAE |
| short | mean | 14.333 | 0.960 | 0.597 | **13.599** | **0.755** | **0.500** |
| | min | 5.819 | 0.499 | 0.206 | **4.198** | **0.296** | **0.185** |
| | max | **40.682** | 2.523 | 2.227 | 44.714 | **2.378** | **0.977** |
| | std | **8.226** | 0.535 | 0.348 | 8.423 | **0.356** | **0.213** |
| long | mean | 16.766 | 1.415 | 0.689 | **15.357** | **1.249** | **0.601** |
| | min | 0.875 | 0.484 | 0.273 | **0.806** | **0.442** | **0.204** |
| | max | **76.622** | 8.324 | 1.291 | 81.617 | **4.886** | **1.096** |
| | std | **15.320** | 1.213 | **0.220** | 15.397 | **0.944** | 0.227 |

TABLE IV

COMPARISON BETWEEN BASIC BAGGING AND LAYERED BAGGING IN TERMS OF WIN-LOSS COUNT FOR THE TIME SERIES DATA OF NN3 [14] COMPETITION. HERE THE BEST RESULT IS HIGHLIGHTED USING BOLDFACE TEXT.

| Performance Metrics | Number of Wins | |
|---|---|---|
| | Basic Bagging | Layered Bagging |
| sMAPE | 26 | **85** |
| MASE | 32 | **79** |
| MdRAE | 25 | **86** |

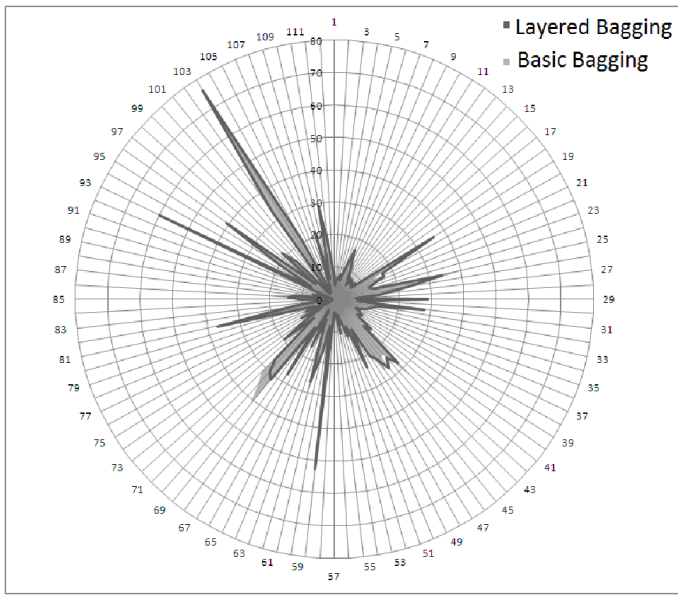


Fig. 1. Comparison between basic bagging and layered bagging in terms of *sMAPE* on time series data of NN3 [14] competition using star plot.

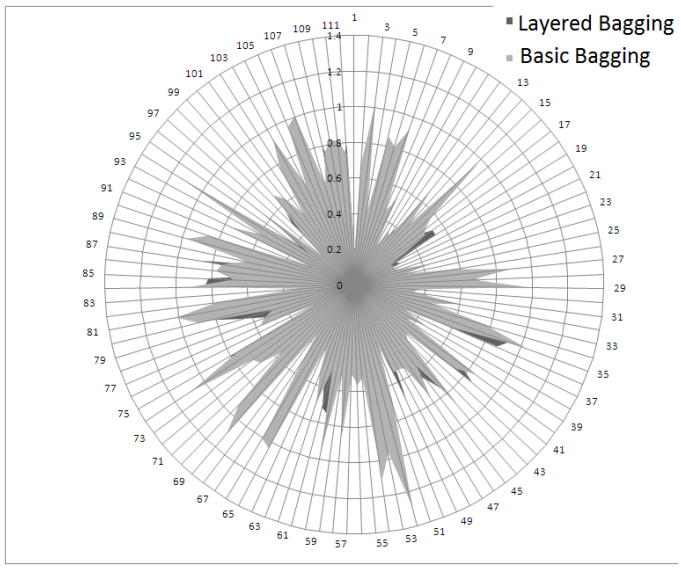


Fig. 2. Comparison between basic bagging and layered bagging in terms of *MdRAE* on time series data of NN3 [14] competition using star plot.

## B. Analysis

In order to understand the reasons behind the better performance of LEA, we analyze the ensembles produced by layered bagging and basic bagging. We employ bias-variance-covariance decomposition, double fault and disagreement for analysis.

*1) Bias-Variance-Covariance Estimation:* Mean-squared-error ($E_{mse}$) of an ensemble can be decomposed into bias ($E_{bias}$), variance ($E_{var}$) and co-variance ($E_{cov}$). For regression problems, this decomposition has been widely used (e.g. [26]) for analyzing the performance of ensembles and can be expressed as

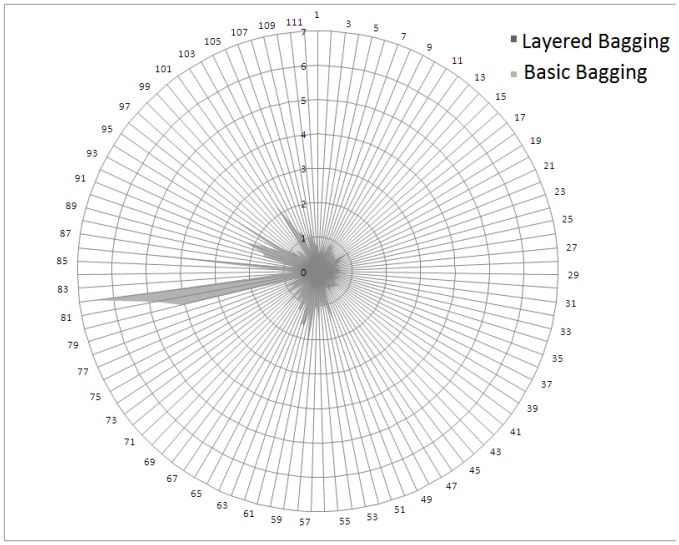


Fig. 3. Comparison between basic bagging and layered bagging in terms of *MASE* on time series data of NN3 [14] competition using star plot.

TABLE V
WILCOXON SIGNED RANK TEST SUMMARY BETWEEN LAYERED BAGGING AND BASIC BAGGING FOR THE TIME SERIES DATA OF NN3 [14] COMPETITION.

| | Performance Metrics | $R^+$ | $R^-$ | p-value | Null hypothesis Significance level=0.05 |
|---|---|---|---|---|---|
| Seasonal | sMAPE | 1899 | 447 | 9.16E-06 | Rejected for layered bagging |
| | MASE | 1802 | 544 | 0.00012 | Rejected for layered bagging |
| | MdRAE | 1867 | 479 | 2.2E-05 | Rejected for layered bagging |
| Non-seasonal | sMAPE | 819 | 127 | 2.94E-05 | Rejected for layered bagging |
| | MASE | 749 | 197 | 0.0086 | Rejected for layered bagging |
| | MdRAE | 787 | 159 | 0.00016 | Rejected for layered bagging |
| Short | sMAPE | 1068 | 207 | 3.24E-05 | Rejected for layered bagging |
| | MASE | 930 | 345 | 0.00048 | Rejected for layered bagging |
| | MdRAE | 1085 | 190 | 1.56E-05 | Rejected for layered bagging |
| Long | sMAPE | 1564 | 327 | 8.89E-06 | Rejected for layered bagging |
| | MASE | 1560 | 331 | 1.02E-05 | Rejected for layered bagging |
| | MdRAE | 1512 | 379 | 4.72E-05 | Rejected for layered bagging |

$$E_{mse} = E_{bias} + E_{var} + E_{cov} \tag{4}$$

The above equation indicates that to achieve good performance, the bias, variance and covariance of the ensemble should be small.

To obtain bias, variance and co-variance of an ensemble architecture, we follow the experimental methodology suggested in [26]. According to [26], several (say, 25) simulations of each ensemble architecture has to be conducted. The only difference in different simulations is the training sets used for training the base predictors. Since the NN3 competition [14] contains a large number of time series, we select only one series from each of the four different types of time series, namely the series number 71, 73, 2 and 110 for seasonal, non-seasonal. short and long series, respectively. However, similar results can be obtained for other series.

Table VI summarizes the results of the bias-variance-covariance decomposition of layered bagging and basic bagging. It can be observed from the table VI that layered bagging provides less bias than basic bagging. Once again, the effectiveness of achieving the accurate lag from layer 1 of layered bagging is evident here. Apart from this, layered bagging also produces less variance and covariance in most of the cases than basic bagging. The positive effect of less bias, variance and co-variance is the less $E_{MSE}$, as shown in the last column of the table VI. Like *sMAPE*, *MdRAE* and *MASE*, layered bagging also defeats here basic bagging.

TABLE VI
COMPARISON BETWEEN LAYERED BAGGING AND BASIC BAGGING IN TERMS OF AVERAGE BIAS, VARIANCE AND CO-VARIANCE DECOMPOSITION FOR THE TIME SERIES DATA OF NN3 [14] COMPETITION. HERE THE BEST RESULT IS HIGHLIGHTED USING BOLDFACE TEXT.

| | | $E_{bias}$ | $E_{var}$ | $E_{cov}$ | $E_{mse}$ |
|---|---|---|---|---|---|
| Layered Bagging | Seasonal | **0.0705** | **4.21E-04** | **0.0101** | **0.0810** |
| | Non-seasonal | **0.0513** | **4.40E-04** | **0.0055** | **0.0572** |
| | Short | **0.0538** | **8.55E-04** | **0.0317** | **0.0863** |
| | Long | **0.0419** | 7.87E-04 | **0.0276** | **0.0702** |
| Basic Bagging | Seasonal | 0.1026 | 4.63E-04 | 0.0046 | 0.1076 |
| | Non-seasonal | 0.0874 | 3.00E-03 | 0.0376 | 0.1280 |
| | Short | 0.0735 | 1.20E-03 | 0.0476 | 0.1223 |
| | Long | 0.1157 | **6.78E-04** | 0.0112 | 0.1275 |

*2) Disagreement and double fault:* Now we like to analyze layered bagging and basic bagging based on their ability to generate diverse ensemble members using disagreement and double fault measurement. The disagreement ($D_{da\{m,n\}}$) and double fault ($D_{df\{m,n\}}$) between two predictors $m$ and $n$ can be expressed as

$$D_{da\{m,n\}} = \frac{N^{01} + N^{10}}{N^{11} + N^{01} + N^{10} + N^{00}} \tag{5}$$

$$D_{df\{m,n\}} = \frac{N^{00}}{N^{11} + N^{01} + N^{10} + N^{00}} \tag{6}$$

Let $N$ be the number of instances, 1 denotes correct classification and 0 denotes incorrect classification. In Eqs. (5) and (6), $N^{ij}$ denotes the number of examples that the first predictor $m$ puts label $i$ on a particular example, while the second predictor $n$ puts label $j$ on the same example. To use disagreement and double fault measures for TSF problems, we use the extension suggested in [27]. From Eqs. (5) and (6), it is evident that a larger disagreement value indicates better diversity. In contrast, a larger double-fault value indicates worse diversity.

Table VII summarizes the average result of disagreement and double fault for basic bagging and layered bagging for 111 time series. It can be observed that in terms of double fault, layered bagging is generating more diverse ensemble than basic bagging irrespective of the nature of the time series. Since layered bagging is trying to enforce accuracy among the members of an ensemble, it is obvious that the number of instances for which a pair of MLPs makes mistake will be less. This is the main reason for obtaining better double fault results by our approach for all four different

TABLE VII

COMPARISON BETWEEN BASIC BAGGING AND LAYERED BAGGING IN TERMS OF DISAGREEMENT AND DOUBLE FAULT FOR SEASONAL, NON-SEASONAL, SHORT AND LONG TIME SERIES DATA OF NN3 [14] COMPETITION. HERE THE BEST RESULT IS HIGHLIGHTED USING BOLDFACE TEXT.

| | Basic Bagging | | Layered Bagging | |
|---|---|---|---|---|
| | disagree | double fault | disagree | double fault |
| Seasonal | 0.255 | 0.393 | **0.271** | **0.365** |
| Non-seasonal | **0.352** | 0.342 | 0.321 | **0.319** |
| Short | **0.370** | 0.390 | 0.327 | **0.361** |
| Long | 0.228 | 0.359 | **0.260** | **0.335** |

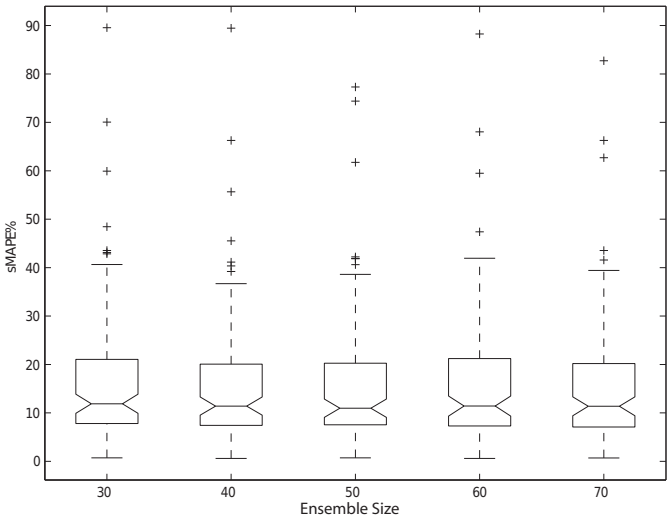


Fig. 4. Boxplot Comparison of sMAPEs of our proposed scheme using different ensemble size

types of time series. In terms of disagreement, both basic bagging and layered bagging are better for two cases.

### C. Effect of Ensemble size

To understand the effect of ensemble size, we vary the ensemble size from 30 to 70 and present the average performance of LEA over 111 time series of the NN3 competition. Fig. 4 illustrates the result obtained using average *sMAPE* for different size of ensemble. It is evident from this figure that increasing the size of an ensemble increases its performance. For example, the average *sMAPE* reduces from 16.47% to 14.56% when we increase the size of the ensemble from 30 to 50. However, increasing the size beyond 50 is not enhancing the performance rather in some case giving inferior results. So, for the sake of computational cost, we prefer to choose 50 is the optimal size of ensemble for the dataset of NN3 competition [14].

### D. Effect of Data Re-sampling Rate

In ensemble layer 2 of LEA, we apply random re-sampling on the dataset $D_{tr}$ to obtain a set of dataset to train each member of ensemble. Unlike basic bagging which uses a data re-sampling rate of 36.2% percentage, we apply here an adaptive strategy to obtain a best data re-sampling rate.

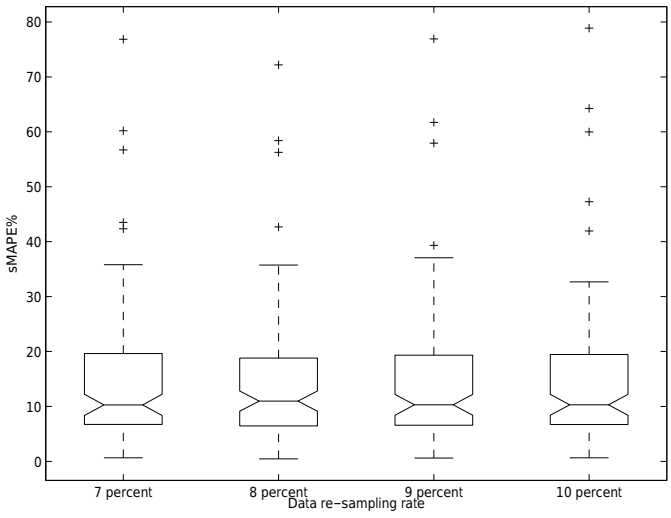


Fig. 5. Boxplot Comparison of *sMAPE* of our proposed scheme using different data re-sampling rate on the time series data of NN3 competition [14]

Since the most of the time series data of NN3 [14] contains data point between 69 to 150, a data re-sampling rate of 36.2% will cause a huge loss of correlation information among the time series data. So we start with a data re-sampling rate of 7% and increments it up to 10. Fig. 5 illustrates the average *sMAPE* obtains over 111 time series data of NN3 competition for the data re-sampling rate of 7%, 8%, 9% and 10% respectively. From the Fig. 5 it is evident that a data re-sampling rate of 9% gives the best result and thats why we uses a data re-sampling rate of 9% in our experiment.

### E. Comparison with other work

The NN3 competition attracts 59 submissions from computational intelligence (CI) based methods and statistical methods, making it the largest CI competition on time series. We choose the best five benchmarked statistical methods and the best five CI based methods for our comparison. Detailed description of these methods can be found in [28]. Furthermore, we choose recently proposed ensembles of RBF networks by Yan [11] for comparison.

Table VIII presents the average results over 111 time series of layered bagging. The result of one algorithm is collected from [11] and 10 other algorithms are compiled from [28]. The model IDs with letter C as prefix stand for CI based models and those with letter B stand for statistical benchmark models. It can be observed from this table that layered bagging beats not only CI based methods but also the benchmarked statistical methods in terms of average *sMAPE*, *MASE* and *MdRAE*. This comparison indicates that the layered ensemble approach with proper techniques for maintaining accuracy and diversity is useful for obtaining a good forecasting accuracy.

## V. CONCLUSION

Consideration of accuracy and diversity among the members of the ensemble is the most important fact for the success of ensemble based algorithms developed either for

TABLE VIII

COMPARISON AMONG LAYERED BAGGING, YAN [11] AND 10 OTHER METHODS [28] BASED ON AVERAGE *sMAPE*, *MASE* AND *MdRAE*. NOTE THAT THE RESULTS ARE AVERAGE OF 111 TIME SERIES DATA OF NN3 COMPETITION AND '-' REPRESENTS DATA ARE NOT AVAILABLE. HERE THE BEST RESULT IS HIGHLIGHTED USING BOLDFACE TEXT.

| ID | Method | sMAPE | MdRAE | MASE |
|---|---|---|---|---|
| - | **Layered bagging** | **14.56** | **0.55** | **1.02** |
| B09 | Wildi | 14.84 | 0.82 | 1.13 |
| B07 | Theta | 14.89 | 0.88 | 1.13 |
| C27 | Illies | 15.18 | 0.84 | 1.25 |
| B03 | ForecastPro | 15.44 | 0.89 | 1.17 |
| - | Yan | 15.80 | - | - |
| B16 | DES | 15.90 | 0.94 | 1.17 |
| B17 | Comb S-H-D | 15.93 | 0.90 | 1.21 |
| C03 | Flores | 16.13 | 0.93 | 1.20 |
| C46 | Chen | 16.55 | 0.94 | 1.34 |
| C13 | D'yakonov | 16.57 | 0.91 | 1.26 |
| C50 | Kamel | 16.92 | 0.90 | 1.28 |

TSF or classification problems. For solving TSF problems, existing ensemble algorithms (e.g. [1], [9], [10]) consider only accuracy or diversity but not both. In this paper, we propose LEA, a layered ensemble architecture, for efficiently forecasting time series data. Our layered architecture is consisted of two layers, each of which is an ensemble of neural networks. Accuracy of the base predictors used for forecasting is ensured by using the appropriate lag, while diversity is encouraged by training the predictors using a different training set. Furthermore, LEA does not combine all base predictors rather only a subset of predictors that exhibit high degree of accuracy and diversity.

To evaluate how well LEA performed, extensive experiments have been carried out in this paper on different TSF problems in comparison with other ensemble and non-ensemble algorithms. In almost all cases, LEA was found better compared to popular ensemble algorithm bagging. These results indicate that irrespective of the type of time series, LEA can help to improve the forecasting accuracy of basic ensemble algorithms. When we compare LEA with other state-of-art statistical and CI methods, our algorithm was also found better in this case. Current implementation of LEA uses MLP networks as base predictors; in future other types of networks such as RBF networks and recurrent neural networks can be used as base predictors. LEA uses a fixed ensemble size and a same $b$ value for data re-sampling irrespective of the type and complexity of time series data. One of the future improvements to LEA would be to make it adaptive in terms of its parameters.

**Acknowledgments.** The work has been done in the Computer Science & Engineering Department of Bangladesh University of Engineering and Technology (BUET). The authors would like to acknowledge BUET for its generous support.

## REFERENCES

[1] H. Chen and X. Yao, "Ensemble regression trees for time series predicitions," *methods*, vol. 11, p. 6, 2007.

[2] D. Ruta, B. Gabrys, and C. Lemke, "A generic multilevel architecture for time series prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 3, pp. 350–359, 2011.

[3] M. Olsson and L. Soder, "Modeling real-time balancing power market prices using combined sarima and markov processes," *Power Systems, IEEE Transactions on*, vol. 23, no. 2, pp. 443–450, 2008.

[4] F.-M. Tseng and G.-H. Tzeng, "A fuzzy seasonal arima model for forecasting," *Fuzzy Sets and Systems*, vol. 126, no. 3, pp. 367–376, 2002.

[5] A. Bouchachia, "Radial basis function nets for time series prediction," *International Journal of Computational Intelligence Systems*, vol. 2, no. 2, pp. 147–157, 2009.

[6] A. Chitra and S. Uma, "An ensemble model of multiple classifiers for time series prediction," *International Journal of Computer Theory and Engineering*, vol. 2, pp. 454–458, 2010.

[7] W. Goh, C. Lim, and K. Peh, "Predicting drug dissolution profiles with an ensemble of boosted neural networks: a time series approach," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 459–463, 2003.

[8] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Information Fusion*, vol. 6, no. 1, pp. 5–20, 2005.

[9] A. Inoue and L. Kilian, "Bagging time series models," 2004.

[10] Z. Zheng, "Boosting and bagging of neural networks with applications to financial time series," Working paper, Department of Statistics, University of Chicago, Tech. Rep., 2006.

[11] W. Yan, "Toward automatic time-series forecasting using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1028–1039, 2012.

[12] I. Maqsood, M. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Computing & Applications*, vol. 13, no. 2, pp. 112–122, 2004.

[13] J. Wichard and M. Ogorzalek, "Time series prediction with ensemble models," in *IEEE International Joint Conference on Neural Networks*, vol. 2. IEEE, 2004, pp. 1625–1630.

[14] "Time series forecasting competition for neural networks and computational intelligence." [Online]. Available: http://www.neural-forecasting-competition.com/NN3

[15] J. J. Moré, "The levenberg-marquardt algorithm: implementation and theory," in *Numerical analysis*. Springer, 1978, pp. 105–116.

[16] A. Gilchrist, "Long-range forecasting," *Quarterly Journal of the Royal Meteorological Society*, vol. 112, no. 473, pp. 567–592, 1986.

[17] R. R. Andrawis and A. F. Atiya, "A new bayesian formulation for holt's exponential smoothing," *Journal of Forecasting*, vol. 28, no. 3, pp. 218–234, 2009.

[18] "Time series forecasting competition for neural networks and computational intelligence." [Online]. Available: http://www.neural-forecasting-competition.com/NN5

[19] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[20] R. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.

[21] M. Adya, F. Collopy, J. Armstrong, and M. Kennedy, "Automatic identification of time series features for rule-based forecasting," *International Journal of Forecasting*, vol. 17, no. 2, pp. 143–157, 2001.

[22] R. R. Andrawis, A. F. Atiya, and H. El-Shishiny, "Forecast combinations of computational intelligence and linear models for the nn5 time series forecasting competition," *International Journal of Forecasting*, vol. 27, no. 3, pp. 672–688, 2011.

[23] M. Adya and F. Collopy, "How effective are neural networks at forecasting and prediction? a review and evaluation," *J. Forecasting*, vol. 17, pp. 481–495, 1998.

[24] H. Drucker, C. Cortes, L. Jackel, Y. LeCun, and V. Vapnik, "Boosting and other ensemble methods," *Neural Computation*, vol. 6, no. 6, pp. 1289–1301, 1994.

[25] J. S. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," *International Journal of Forecasting*, vol. 8, no. 1, pp. 69–80, 1992.

[26] R. A. Jacobs, "Bias/variance analyses of mixtures-of-experts architectures," *Neural computation*, vol. 9, no. 2, pp. 369–383, 1997.

[27] H. Dutta, "Measuring diversity in regression ensembles," in *IICAI*, vol. 9, 2009, p. 17p.

[28] S. F. Crone, M. Hibon, and K. Nikolopoulos, "Advances in forecasting with neural networks? empirical evidence from the nn3 competition on time series prediction," *International Journal of Forecasting*, vol. 27, no. 3, pp. 635–660, 2011.