IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Ensembles of Gradient Boosting Recurrent Neural Network for Time Series Data Prediction

Shiqing Sang [1], Fangfang Qu [2, 3], and Pengcheng Nie.[2,3,4]

[1] Jiaxing Vocational and Technical College, Jiaxing 314036, China
[2] College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310058, China
[3] Key Laboratory of Sensors Sensing, Ministry of Agriculture and Rural Affairs, Zhejiang University, Hangzhou 310058, China
[4] State Key Laboratory of Modern Optical Instrumentation, Zhejiang University, Hangzhou 310058, China

Corresponding author: Pengcheng Nie (e-mail: pcn@zju.edu.cn).

This work was supported in part by the Situation analysis and demonstration application of saffron cultivation and soil nutrients based on big data mining technology 201900014, Provincial Project in Data Fusion Analysis and Intelligent Regulation of Saffron Growth Environment Monitoring LGN19C130002

ABSTRACT Ensemble deep learning can combine strengths of neural network and ensemble learning and gradually becomes a new emerging research direction. However, the existing methods either lack theoretical support or demand large integrated models. To solve these problems, in this paper, Ensembles of Gradient Boosting Recurrent Neural Network (EGB-RNN) is proposed, which combines the gradient boosting ensemble framework with three types of recurrent neural network models, namely Minimal Gated Unit (MGU), Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). RNN model is used as base learner to integrate an ensemble learner, through the way of gradient boosting. Meanwhile, for ensuring the ensemble model fit data better, Step Iteration Algorithm is designed to find an appropriate learning rate before models being integrated. Contrast trials are carried out on four time series data sets. Experimental results demonstrate that with the number of integration increasing, the performance of three types of EGB-RNN models tend to converge and the best EGB-RNN model and the best degree of ensemble vary with data sets. It is also shown in statistical results that the designed EGB-RNN models perform better than six baselines.

INDEX TERMS Gradient boosting; LSTM; GRU; MGU; Ensemble learning; Time series data prediction

I. INTRODUCTION

Time series data can be used to describe objects that change over time, and the forecast of time series data has essential guiding effect on actual production whether in business, agriculture or industry [1-5]. For those research objects that changes over time, the internal mechanisms are usually complicated and hard to be described with complete theories. Therefore, designing accurate first principle models of such objects is a tough task [5]. Nevertheless, a large amount of collected time series data will help to build not only the first principle models but also data-driven models. With the rapid development of data science, data-driven based methods have gradually become effective for system identification and data analysis. Hence, data-driven modeling is a popular method to be used for predicting time series data [3, 5].

At present, there are a lot of data-driven based researches for analyzing time series data. Classical models based on parameter estimation include Auto-Regressive Moving Average Model [6], Auto-Regressive Integrated Moving Average Models [7] and Seasonal Vector Auto-Regressive Model [8] etc. Many of these models are linear or simple nonlinear, which makes them can be employed to fit the data that change smoothly with time. But it is hard for them to fit the data produced from complex stochastic processes [5]. Machine learning method is a great choice to improve the prediction accuracy of complicated systems. The prediction of time series data can be realized by some regression prediction methods in traditional machine learning models. Support Vector Machine (SVM) is frequently used in the prediction of financial and weather-related time series data [9, 10]. And some other variants of SVM are proposed to be applied in wind speed prediction [11, 12]. But, it is the support vectors that determine the performance of SVM, and the temporality is not taken into account. That means SVM does not consider the temporal nature among data and is not a specialized technique for processing time series data. Artificial Neural Network (ANN) is also a fashionable method in machine learning. Multi-Layer Perceptron neural network [13, 31], and Radical Basis Function neural network [14] are widely used in time series data prediction. But they have the same limitation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082519, IEEE Access

as SVM that they are not quiet suitable for forecasting sequential data.

Memory accumulated in the past can have impact on future learning. This is the core idea of the Recurrent Neural Network (RNN) model in data analysis. RNN is a sort of deep learning methods for processing sequential data in common use. Many RNN-based models were proposed for prediction in series data analysis. Long Short-Term Memory (LSTM) neural network [15] is a RNN model based on gated mechanism. Gated Recurrent Unit (GRU) neural network [16] is brought up to alleviate training difficulties in LSTM. Xie *et al*. [5] compared LSTM and GRU in melt spinning time series data prediction and confirmed that GRU could reduce training time while ensuring prediction accuracy. Zhou *et al*. [17] presented a further simplified model named Minimal Gated Unit (MGU). And through experiments on four data sets, it is verified that MGU can further reduce the training time while guaranteeing precision. Dong *et al.* [18] used deep MGU-based neural network to predict the round trip time data in networks.

With the growing of ensemble learning [19], for further improving the accuracy of time series data prediction, there are some works embedding machine learning or deep learning model into ensemble framework, including integrated deep forest method [20], ensemble neural network models [3] *etc*. All of these triggers the blossom of ensemble deep learning [21], which treats deep learning models as weak learners and aims to integrate them to a strong learner. However, some of these existing works demand large number of basic models [22], while some others just design complicated integrated frames that lack the support in theory. Therefore, how to design a better strategy for integrating deep learning model and meanwhile avoid these problems are the main focus of our research in this paper.

The main contributions of this paper are summarized as follows:

(1) For the first time, gradient boosting framework and RNN models are combined together to establish ensemble deep learning models. Gradient boosting can fit and remedy negative gradient or prediction error through integrating, which makes it more pertinent in integration process than other proposed ensemble strategies.

(2) In the process of integrating, the learning rate is a key point to choose. Traditional methods based on line search are used to find an optimum solution, but these ways can easily lead to over fitting. Hence, Step Integration algorithm is designed to find an appropriate learning rate. This algorithm can not only make the ensemble effect better but also avoid over fitting train data to some extent.

(3) The performance of the three models, MGU, GRU and LSTM, vary with different types of data. Most existing methods of ensemble RNN only focus on LSTM, but there are experiments demonstrating that ensemble LSTM performs poorer than the other two sometimes. Thus, this research extends the types of basic learners to break the limitation of integrating LSTM only.

II. RELATED WORK

Boosting is an ensemble method based on bootstrapping idea. At each iteration of training, the train data set is adjusted according to the results of previous integrations, so that the latter basic learners pay more attention to the learning of more error-prone train samples. By such a way, learning efficiency of the ensemble model can be improved. Further, gradient boosting [23] is obtained by introducing the thought of gradient descent into boosting algorithm. Gradient boosting algorithm fits negative gradient between the true value and predicted value through subsequent basic learners. Therefore, each time with a model being merged, the error of previous prediction can be corrected so as to achieve the boosting effect. In the light of different regression and classification tasks, there are many alternative types of loss functions. For example, squared loss, absolute loss and exponential loss are common selections and the negative gradient changes as the variety of loss functions. Moreover, because the gradient boosting algorithm can deal with different loss functions, it has been adopted to different learning tasks.

It is an effective strategy to improve the deficiency of general model based on boosting ensemble tactic. Freund *et al*. [24] proposed Adaptive Boosting (AdaBoost) method and an AdaBoost-based ensemble learning model which can be gained by embedding the decision tree into Adaboost freamwork. Friedman *et al*. [23] presented gradient boosting frame and employed Classification and Regression Trees (CART) [25] as the base learner in this framework. Therefore, a classical ensemble learning model called Gradient Boosting Decision Tree (GBDT) [23] can be gained. Based on GBDT, Extreme Gradient Boosting (XGBoost) [26] is proposed which is a widely used ensemble machine learning model currently. Inspired by ensemble learning, there is a new research direction to establish ensemble deep learning method by combining integration strategy and general deep learning model. Chen *et al*. [3] designed an ensemble model called EnsemLSTM for wind speed data prediction. Cao *et al*. [27] put forward a combination of Empirical Mode Decomposition and LSTM, and the integrated whole acts as an ensemble model to forecast financial time series data. Pak *et al.* [28] combined convolutional neural networks with LSTM to prognosticate ozone concentration. However, these integrated methods are implemented by combining multiple different types of models and these models are more like hybrid rather than ensemble. Generally known ensemble learning methods, e.g., boosting, have complete theoretical support. AdaBoost has rigorous deduction processes for calculating the weights of training data in learning and that of base learners in integrating. In gradient boosting, subsequent basic learners are arranged to fit the error between prediction and true value, so that the predicted result keeps approaching the truth. Such as the ways like them can be called ensemble learning methods in the usual sense. Hence, methods of assembling multiple sorts of models lack rigorous theoretical proof and cannot be deemed the real ensemble learning methods. At present, in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082519, IEEE Access

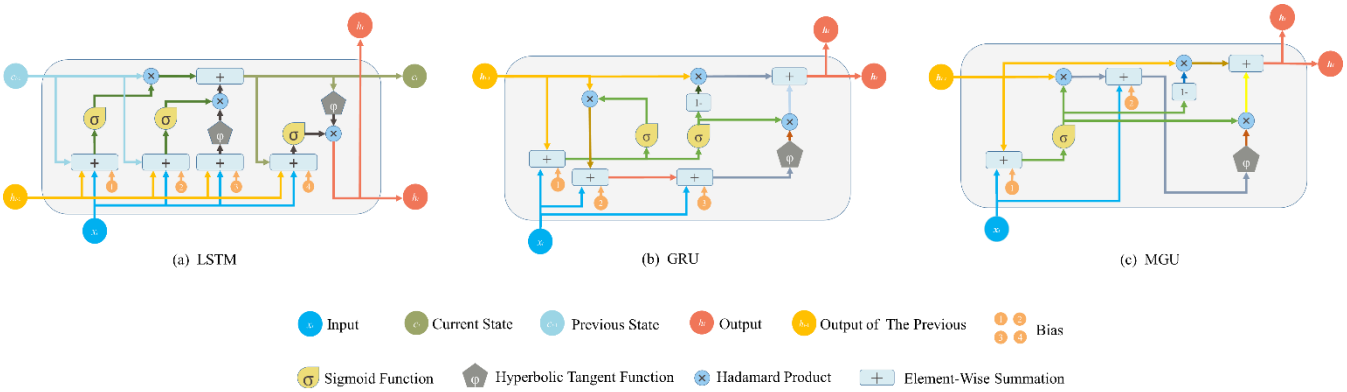


Fig. 1. Structures of LSTM, GRU and MGU. (a) is LSTM; (b) is GRU; (c) is MGU.

prediction of time series data, methods based on ensemble deep learning mostly adopt the combination of AdaBoost and LSTM. Wu *et al.* [22] proposed AdaBoost-based LSTM ensemble learning approach for forecasting finical time series data, in which LSTM is embedded as the basic learner in AdaBoost framework. Hao *et al.* [29] integrated Ensemble Empirical Mode Decomposition algorithm, LSTM and Adaboost simultaneously to construct an ensemble model for forecasting the change of Shanghai gold price. However, LSTM is not the only choice of RNN-based deep learning model in time series data prediction and there are still two more models to choose from [5, 17]. Therefore, using LSTM simply for integrating is far from enough. Moreover, the AdaBoost framework requires a large number of base LSTM, and the best result can be gained until 30 to 40 LSTM models [22]. Compared with AdaBoost, basic learners in gradient boosting aim to compensate errors, and this technique makes gradient boosting learning process more targeted. However, in the task of time series data prediction, the ensemble deep learning method based on gradient boosting framework has not been proposed yet.

III. THE PROPOSED EGB-RNN METHOD

Three RNN models based on gated mechanism are all effective methods for forecasting time series data. However, the prediction accuracy still needs to be further improved. Influenced by ensemble learning, various ensemble RNN models have been proposed and these models effectively improve the accuracy of data prediction to a certain extent [3, 4]. But there are two main problems in existing ensemble RNN models. The first one is that some of these ensemble methods do not use a mature ensemble framework and therefore lack theoretical bases. Accordingly, the number of ensemble models depends on random adjustment that leads to an inefficiently modeling process. Secondly, the widely used AdaBoost framework requires a large number of basic learners, which can cause more computational cost undoubtedly. Experiments showed that around 40 models are needed to achieve better prediction results [22]. Nevertheless, gradient boosting, as shown in Algorithm 1, is one of the effective and frequently adopted ensemble techniques. In the algorithm, $\mathcal{D}$ represents train data set and $l(\cdot)$ is loss function. In process, $F_0(\mathrm{x})$ is the initialized ensemble model, which is also a basic learner. k represents the degree of integration from 1 to K. $f_k(\mathbf{x})$ is base learner and ω_k represents the parameter of it. ν_k is the searched step size. $F_k(\mathbf{x})$ is the kth ensemble model and $F_K(\mathbf{x})$ is the final ensemble model. Each integration is a remedy for previous error so that the ensemble effect has a stronger pertinence than some proposed integration strategy. Hence, this paper considers combining the gradient boosting ensemble strategy with three types of RNN to construct Ensembles of Gradient Boosting Recurrent Neural Network (EGB-RNN) models. This method can not only improve the prediction accuracy but also reduce the computational cost mainly caused by the increase of basic learners, because of fewer weak learners being demanded.

Algorithm 1: The Gradient Boosting algorithm

Input: Train data set $\mathcal{D}=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),...,(\mathbf{x}_n,y_n)\}$;
Loss function $l(\cdot)$.

Process:

1. Initialize: $F_0(\mathbf{x})=\arg\min_{\gamma}\sum_{i=1}^{n}l(y_i,\gamma)$
2. for $k=1,2,...,K$ do:
3. Calculate the negative gradient: $\epsilon_k=-\dfrac{\partial l(y,F_{k-1}(\mathbf{x}))}{\partial F_{k-1}(\mathbf{x})}$
4. Use base learner $f_k(\mathbf{x})$ to fit ϵ_k: $\omega_k=\arg\min[\epsilon_k-f_k(\mathbf{x};\omega)]^2$
5. Determine the step size ν_k with line search method:

$$\nu_k=\arg\min_{\nu}\sum_{i=1}^{n}l(y_i,F_{k-1}(\mathbf{x}_i)+\nu f_k(\mathbf{x}_i;\omega_k))$$

6. $F_k(\mathbf{x})=F_{k-1}(\mathbf{x})+\nu_k f_k(\mathbf{x};\omega_k)$
7. End for

Output: The ensemble model $F_K(\mathbf{x})$

Combined with RNN models, the gradient boosting algorithm needs to be modified. In the initialization process, one of three RNN models is used to fit the train data to initialize the ensemble model $F_0(\mathrm{x})$. This step is the same as that of ordinary RNN. Three gated mechanism based RNN models, LSTM, GRU and MGU, are chosen to do this work

separately. And mathematical expressions of LSTM are shown as follows.

$$\boldsymbol{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \boldsymbol{h}_{t-1} + \mathbf{b}_i), \tag{1}$$

$$\boldsymbol{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \boldsymbol{h}_{t-1} + \mathbf{b}_f), \tag{2}$$

$$\boldsymbol{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \boldsymbol{h}_{t-1} + \mathbf{b}_o), \tag{3}$$

$$\tilde{\boldsymbol{c}}_t = \phi(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \boldsymbol{h}_{t-1} + \mathbf{b}_c), \tag{4}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t, \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \phi(\boldsymbol{c}_t). \tag{6}$$

where t represents the current moment, $t-1$ represents the previous moment, $\mathbf{x}_t$ represents the current input, $\boldsymbol{h}_{t-1}$ represents the previous output, $\boldsymbol{i}_t$, $\boldsymbol{f}_t$ and $\boldsymbol{o}_t$ represent the input gate, forget gate and output gate, respectively, $\tilde{\boldsymbol{c}}_t$, $\boldsymbol{c}_t$ and $\boldsymbol{c}_{t-1}$ are the candidate, current state and previous state, respectively. $\boldsymbol{h}_t$ represents the current output, $\sigma(\cdot)$ is the activation function and the most frequently used is sigmoid function, ϕ represents hyperbolic tangent function, $\odot$ is Hadamard product, $\mathbf{W}_i$, $\mathbf{U}_i$, $\mathbf{W}_f$, $\mathbf{U}_f$, $\mathbf{W}_o$, $\mathbf{U}_o$, $\mathbf{W}_c$ and $\mathbf{U}_c$ represent corresponding weight matrices, $\mathbf{b}_i$, $\mathbf{b}_f$, $\mathbf{b}_o$, and $\mathbf{b}_c$ are relevant biases. The structure of LSTM is illustrated in Fig. 1(a). It can be observed that LSTM is a RNN based model with three gated structures. Equations of GRU are shown as follows.

$$\boldsymbol{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \boldsymbol{h}_{t-1} + \mathbf{b}_r), \tag{7}$$

$$\boldsymbol{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \boldsymbol{h}_{t-1} + \mathbf{b}_z), \tag{8}$$

$$\tilde{\boldsymbol{h}}_t = \phi(\mathbf{W}_h \mathbf{x}_t + [\mathbf{U}_h(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})] + \mathbf{b}_h), \tag{9}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t. \tag{10}$$

where $\boldsymbol{r}_t$ and $\boldsymbol{z}_t$ represent reset gate and update gate, respectively, $\tilde{\boldsymbol{h}}_t$ represents current hidden state, $\mathbf{W}_r$, $\mathbf{U}_r$, $\mathbf{W}_z$, $\mathbf{U}_z$, $\mathbf{W}_h$ and $\mathbf{U}_h$ represent corresponding weight matrices, $\mathbf{b}_r$, $\mathbf{b}_z$ and $\mathbf{b}_h$ are relevant biases. The remaining symbols have same meanings as them in LSTM. The graphical structure of GRU is shown as Fig. 1(b), from which it can be found that GRU also belongs to RNN model and contains two gated structures less than that in LSTM. Formulas of MGU are as follows.

$$\boldsymbol{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \boldsymbol{h}_{t-1} + \mathbf{b}_f), \tag{11}$$

$$\tilde{\boldsymbol{h}}_t = \phi(\mathbf{W}_h \mathbf{x}_t + [\mathbf{U}_h(\boldsymbol{f}_t \odot \boldsymbol{h}_{t-1})] + \mathbf{b}_h), \tag{12}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{f}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{f}_t \odot \tilde{\boldsymbol{h}}_t. \tag{13}$$

where all symbols appearing in LSTM and GRU have same meanings. The illustration of MGU is shown as Fig. 1(c). It can be seen that MGU has only one gated structure and the forget gate is retained that makes it the simplest of the three models.

Algorithm 2: The Step Integration algorithm

Input: Interval $[a,b]$;
Iteration times T ;
Step number s .

Process:

1. Initialize: $a_0 = 0, b_0 = 1$
2. for $t = 1,2,...,T$ do:
3. According to $[a_{t-1}, b_{t-1}]$ and s , select candidate values set:
   $\mathcal{O}_t = \{a_{t-1}, a_{t-1} + \frac{b_{t-1} - a_{t-1}}{s}, ..., b_{t-1}\}$
4. Find the optimal value among $\mathcal{O}_t$:
   $\varepsilon_t = \arg\min_{\varepsilon \in \mathcal{O}_t} l(y, F_{k-1}(\mathbf{x}) + \varepsilon f_k(\mathbf{x}))$
5. Update the left boundary: $a_t = \varepsilon_t - \frac{b_{t-1} - a_{t-1}}{s}$
6. if $a_t < 0$: $a_t = 0$
7. Update the right boundary: $b_t = \varepsilon_t + \frac{b_{t-1} - a_{t-1}}{s}$
8. if $b_t > 1$: $b_t = 1$
9. End for

Output: ε_T

Next, the loss function $l(\cdot)$ of this regression task is set to squared loss shown as Eq. (14). Hence, there is a definite equation for the negative gradient ϵ_k shown as Eq. (15).

$$l(y, F_{k-1}(\mathbf{x})) = \frac{1}{2}[y - F_{k-1}(\mathbf{x})]^2, \tag{14}$$

$$\epsilon_k = -\frac{\partial l(y, F_{k-1}(\mathbf{x}))}{\partial F_{k-1}(\mathbf{x})} = y - F_{k-1}(\mathbf{x}). \tag{15}$$

where $F_{k-1}(\mathbf{x})$ means (k-1)th ensemble model, y represents truth value.

It can be analyzed that ϵ_k is expressed as the error between the truth and predicted value. Then a same type RNN model is used to fit ϵ_k and the base learner $f_k(\mathbf{x})$ can be gained. When it comes to the solution of learning rate in the process of integration, line search based method, *e.g.*, newton and gradient descent, is used to try to find the optimal solution of learning rate. However, since the ensemble model is constructed by fitting train data, it is likely that there may be a case of over fitting if the optimal solution is obtained. Therefore, to avoid this situation, an optimization algorithm named Step Iteration is designed and described as Algorithm 2. The core idea of this method is to perform iterative search operation according to a certain step number within a given interval. In the process of Algorithm 2, since the learning rate is between 0 and 1, this method initializes the search interval to [0, 1]. a_0 and b_0 represent the initialized maximum and minimum boundaries, respectively. Then, in the light of step number S and interval, candidate values are selected as set $\mathcal{O}_t$ and the optimal value ε_t among them is found. Afterward, the interval is updated and the process is continued until the given

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082519, IEEE Access

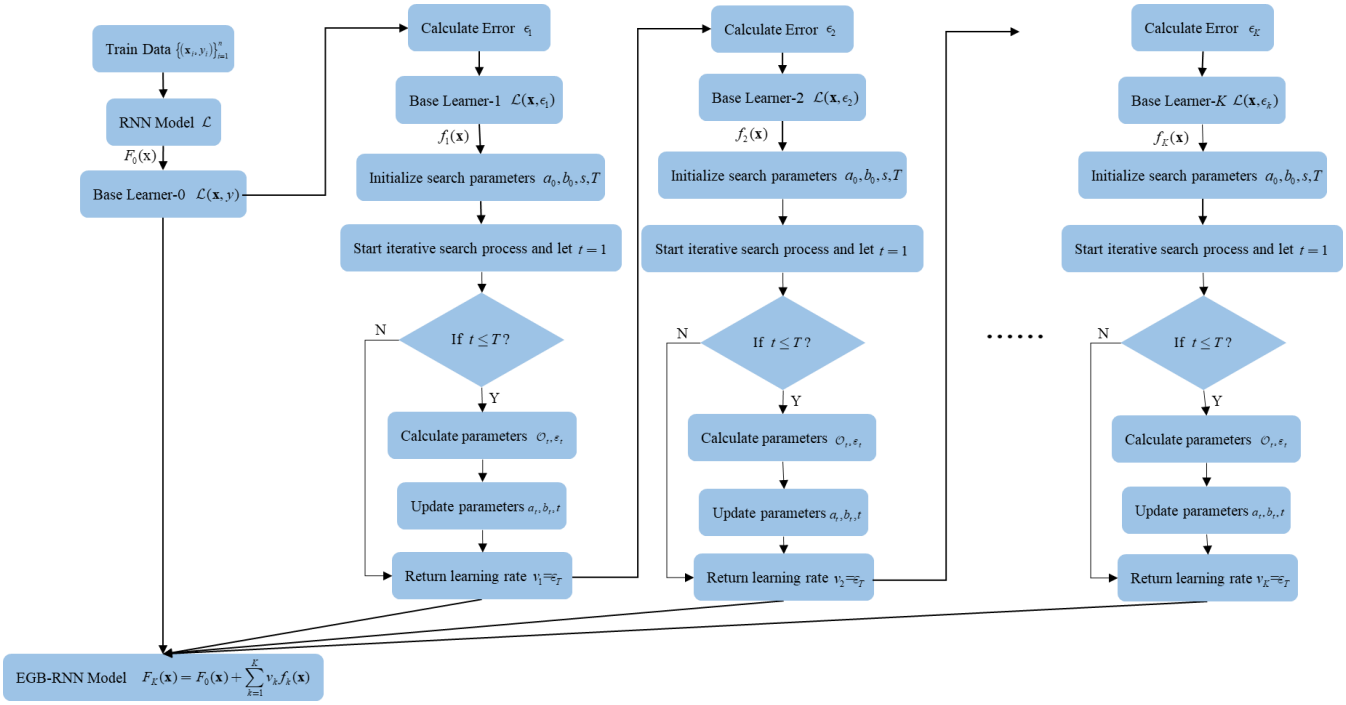


Fig. 2. The graphical representation of proposed EGB-RNN method

Algorithm 3: The EGB-RNN algorithm

Input: Train data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)\}$;
Base RNN learner $\mathcal{L}$;
The number of cascaded models K.

Process:

1. Initialize: $F_0(\mathrm{x}) = \mathcal{L}(\mathbf{x}, y)$
2. for $k = 1, 2, ..., K$ do:
3. Calculate error: $\epsilon_k = y - F_{k-1}(\mathbf{x})$
4. Train base learner: $f_k(\mathbf{x}) = \mathcal{L}(\mathbf{x}, \epsilon_k)$
5. Use Step Integration algorithm to search learning rate:

$$v_k = \arg\min_{v} \sum_{i=1}^{n} l(y_i, F_{k-1}(\mathbf{x}_i) + v f_k(\mathbf{x}))$$

6. $F_k(\mathbf{x}) = F_{k-1}(\mathbf{x}) + v_k f_k(\mathbf{x}; \omega_k)$
7. End for

Output: The ensemble model $F_K(\mathbf{x})$

iteration times T is completed and the final optimal value ε_T is gained. The purpose of processes 6 and 8 in Algorithm 2 is to prevent the updated interval from exceeding the specified range. Finally, the proposed EGB-RNN algorithm is obtained and shown as Algorithm 3. In this method, gradient boosting is taken as ensemble framework and RNN models are used to initialize the ensemble model and also fit negative gradient. At the same time, the Step Iteration algorithm is applied to find a suitable learning rate

It should be noted that the same type of RNN model should be used throughout the whole integration process. Because of three RNN models based on gated mechanism being used, three EGB-RNN models can be constructed, namely EGB-LSTM, EGB-GRU, and EGB-MGU separately. The graphical representation of EGB-RNN algorithm is illustrated in Fig. 2. In the figure, the first column on the left indicates the process of initialization and integration. The second to fourth columns represent the procedure of ensemble in turn, from *k*=1 to *K*. During the operation of each integration, Step Iteration algorithm is employed to search applicable learning rate. Macroscopically, each supplemented model could cause gradient boosting, thereby improving the performance of the integrated model as a whole.

IV. EXPERIMENTS AND ANALYSIS

A. EXPERIMENTAL PREPARATIONS

Four data sets from https://fred.stlouisfed.org/ [1] are used in experiments. The first data set is 10-Year Treasury Constant Maturity Rate (DGS10), in which the time range is from January 2010 to April 2019 and the time span between adjacent data points is one day. It has 2,313 data points after removing missing days and the unit is percentage. The second data set is 20-Year Treasury Inflation-Indexed Security (DFII20) which owns 2,316 records in total after ignoring the missing part and also uses percentage as unit. The time changes from January 2010 to April 2019 in this data set and the time span is also one day. The third one is Average Weekly Hours of Production and Nonsupervisory Employees about Durable Goods (AWH) with month as frequency and hour as the unit. It contains 964 points from

January 1939 to April 2019. The last one is Gold Fixing Price (GFP) in London Bullion Market, based in U.S. Dollars. There are 2,777 data points in GFP from January 2010 to December in 2020, with the frequency of day. Experiments are carried out on these four datasets and the data allocation is that the former 80% are used as train data and the remaining 20% as test data to verify the performance of models. In regard to parameter settings in Algorithm 2, the iteration times T and step number s are set as three and ten, respectively. The Root Mean Square Error (RMSE) is adopted as evaluation criterion and the equation is:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2}. \tag{16}$$

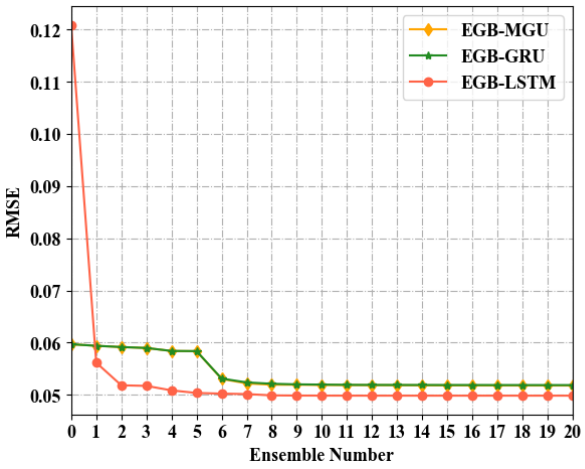
where n, f_i, y_i represents the amount of data, predicted value, and true value, respectively.

For comparing with the proposed EGB-RNN method, MGU, GRU, LSTM, GBDT and XGBoost which is commonly used in sequential data analysis nowadays are chosen as baselines. Moreover, a new proposed ensemble learning method, called Boosted Random Forest (BRF) [30], is also chosen as comparison. Simultaneously, although integrating weak learners can obtain a strong learner, this does not imply that more models can be used to improve better, *i.e.*, over fitting is always an inevitable problem in ensemble learning method. So, besides contrasting with baselines, it is equally necessary to explore the number of integration several times to find the best ensemble degree.

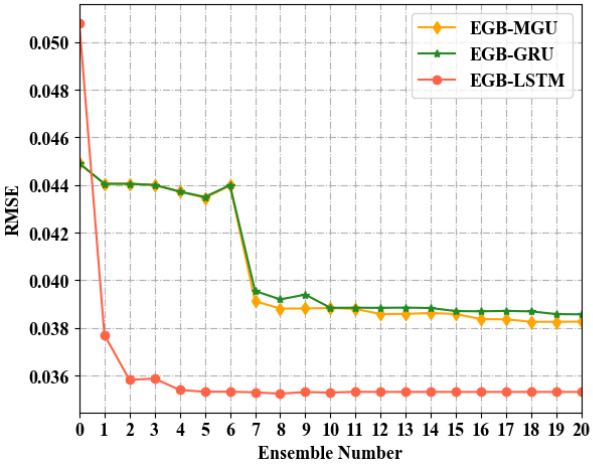
B. COMPARATIVE EXPERIMENTS AND RESULTS

Three RNN models, MGU, GRU and LSTM, need to be introduced into EGB-RNN to construct three types of ensemble models, EGB-MGU, EGB-GRU and EGB-LSTM, to carry out experiments, respectively. Meanwhile, EGB-LSTM-0 represents no integration and this model can be obtained just by a LSTM after fitting train data. EGB-LSTM-1 means the error caused by pervious model, *i.e.*, EGB-LSTM-0, is remedied and the whole model is integrated once. The rest can be comprehended in the same manner.

Firstly, experiments are conducted on DGS10 dataset. According to Algorithm 3, a RNN model is chosen to initialize the ensemble model. Then, the same type of RNN model is used to fit the train error and the whole model is integrated. The RMSE results in training process are showed in Fig. 3(a), in which it can be found that with the number of integrated models increasing, three kinds of ensemble models all ultimately reach a stabilized state. After a point around six, the capability of EGB-RNN models attain superior limits and this phenomenon means integrated models cannot learn train data endlessly. On the contrary, the ensemble capacity has an upper limit. RMSE results in test data are also diagrammed in Fig. 3(b) where, as integration degree deepening, the prediction error of EGB-RNN models stably near a fixed value that is similar to training process. Because of the learning ability in train data reaching saturation, the performance of EGB-RNN models in test data appears convergence. Besides, although for the single MGU and GRU models have comparable prediction accuracy and are superior to LSTM, the EGB-LSTM model performs better than the other two EGB-RNN models with the same level of ensemble. At the same time, the prediction



(a) RMSE results in train data



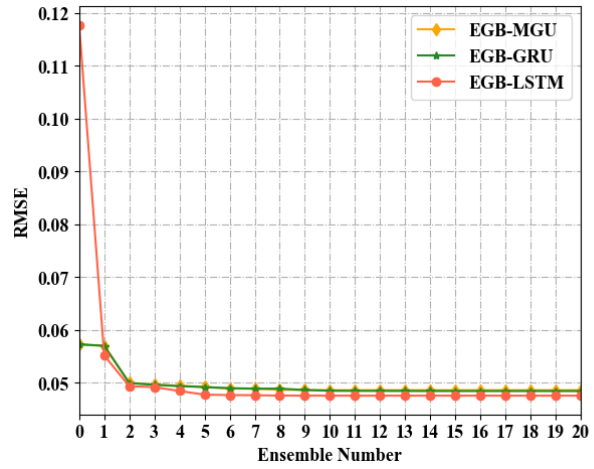
(b) RMSE results in test data

Fig. 3. RMSE results of three EGB-RNN models with ensemble numbers increasing on DGS10 dataset.

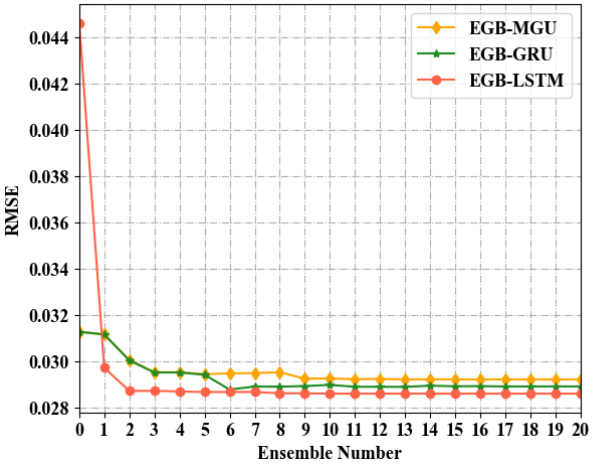
TABLE I. The summary of comparison RMSE results on DGS10 Dataset

Model Name	RMSE
XGBoost	0.038129
GBDT	0.042367
MGU	0.044909
GRU	0.044909
LSTM	0.050814
BRF	0.038481
EGB-MGU-19	0.038252
EGB-GRU-20	0.038566
EGB-LSTM-20	**0.035308**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082519, IEEE Access



(a) RMSE results in train data



(b) RMSE results in test data

Fig. 4. RMSE results of three EGB-RNN models with ensemble numbers increasing on DFII20 dataset.

TABLE II. The summary of comparison RMSE results on DFII20 dataset

Model Name	RMSE
XGBoost	0.030904
GBDT	0.035617
MGU	0.031272
GRU	0.031272
LSTM	0.044620
BRF	0.031068
EGB-MGU-20	0.029214
EGB-GRU-20	0.028911
EGB-LSTM-20	**0.028600**

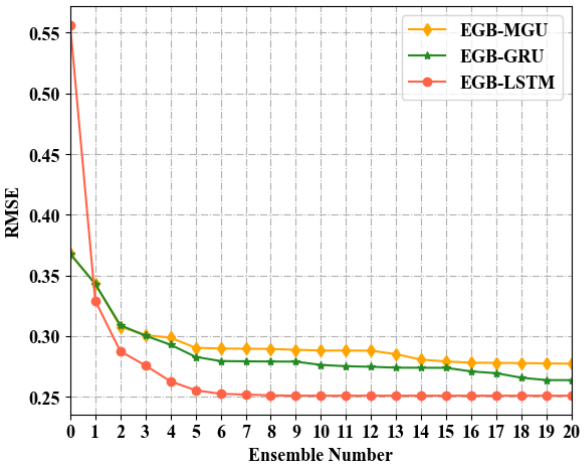
results of EGB-MGU models and EGB-GRU models are similar when the degree of ensemble is the same. From Fig. 3 it can be seen that weather in train or test part, rapid decline appears in EGB-LSTM-1 which uses one LSTM to fit errors calculated by EGB-LSTM-0, which manifests LSTM can fit prediction error faster than the other two in current data set. RMSE results of comparative experiments with other models are displayed in Table I. For ensemble MGU models, EGB-MGU-19 integrated nineteen times has the smallest prediction error. For ensemble GRU models, EGB-GRU-20 has the best prediction result. For ensemble LSTM models, EGB-LSTM-20 performs best and also outweighs all the other models. It can also be educed that the prediction results of ensemble RNN models are always better than that of single RNN models.

Next, comparative experiments are performed on DFII20 dataset. As the number of integrated models increases, the performance of EGB-RNN models tends to converge, regardless of whether they are in training set or test set, as shown in Fig. 4. It can also be obtained that LSTM can fit error more quickly than the other two, and they all have equally matched performance eventually, after being integrated 5 times in training process. In test part, their final performance is also close, while EGB-MGU and EGB-GRU converge a little bit slowly. Meanwhile, EGB-RNN models always outperform ordinary RNN models. The summarized contrastive results gained are displayed in Table II. By comparison, for general RNN models, the prediction results of MGU and GRU are the same and better than LSTM. EGB-MGU-20, EGB-GRU-20, EGB-LSTM-20 are the best ensemble model, respectively. For all models, EGB-LSTM-20 manifests the best performance. The prediction accuracy of EGB-LSTM is the best under the same ensemble level, and same as the result of the previous experiment, the final ensemble LSTM model has the best performance.

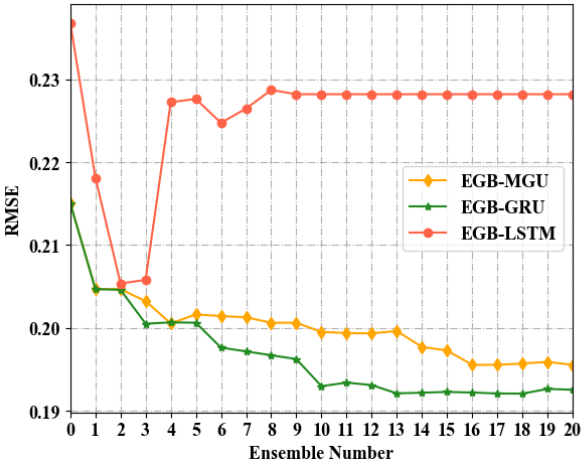
Then, AWH dataset, the third part, is used to carry out experiments. In Fig. 5(a), it can be found that similar to previous trials, the learning ability of three EGB-RNN models tends to converge with the increasing of ensemble degree. However, in Fig. 5(b), after the point around two or three, there is a dramatic increase happening in the line of EGB-LSTM, before it converging. While the other two show a slow convergence trend. Although EGB-LSTM models have optimistic performance on training set, EGB-LSTM performs poorer than the other two on test set. This phenomenon may be caused by over fitting, *i.e.*, subsequent integrated LSTM models over fit training data whereas produce adverse effect on generalization performance. In contrast, EGB-MGU and EGB-GRU behave well, even though their performance on the test set is not as good as that in previous experiments. Statistical results are exhibited in Table III, where EGB-MGU-20, EGB-GRU-18 and EGB-LSTM-2 represents each of three ensemble methods and meanwhile EGB-GRU-18 has the smallest prediction error among all models.

Finally, comparative trials are conducted on GFP dataset, while there is a little difference from previous experiential results. In Fig. 6(a), EGB-GRU converges rapidly and after it only being integrated once, RMSE result almost remains unchanged. Analogously, EGB-MGU reaches stable state after only being integrated twice, while RMSE number of it is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3082519, IEEE Access

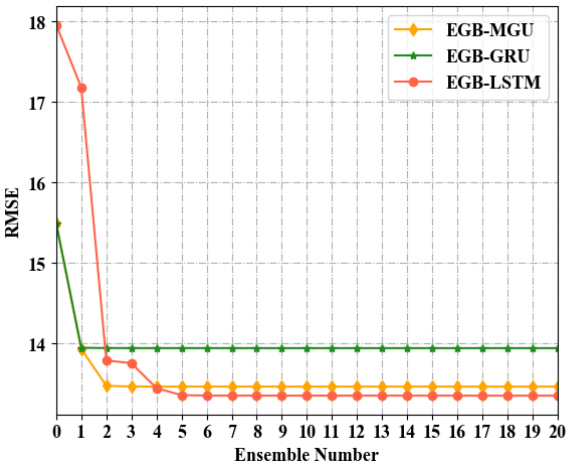


(a) RMSE results in train data

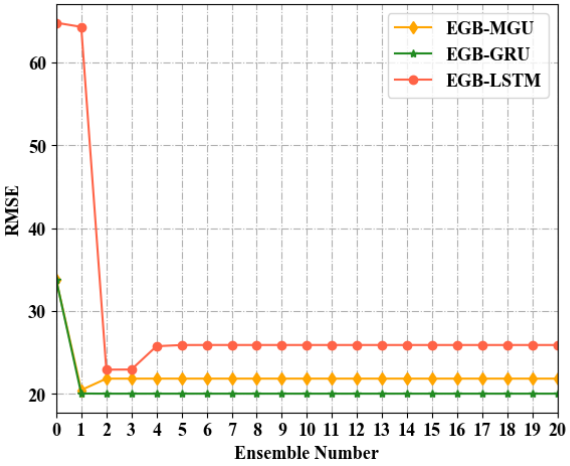


(b) RMSE results in test data

Fig. 5. RMSE results of three EGB-RNN models with ensemble numbers increasing on AWH dataset



(a) RMSE results in train data



(b) RMSE results in test data

Fig. 6. RMSE results of three EGB-RNN models with ensemble numbers increasing on GFP dataset

TABLE III. The summary of comparison RMSE results on AWH dataset

Model Name	RMSE
XGBoost	0.221419
GBDT	0.233971
MGU	0.215048
GRU	0.215048
LSTM	0.236828
BRF	0.211507
EGB-MGU-20	0.195531
EGB-GRU-18	**0.192058**
EGB-LSTM-2	0.205358

TABLE IV. The summary of comparison RMSE results on GFP dataset

Model Name	RMSE
XGBoost	35.227079
GBDT	28.469657
MGU	33.794473
GRU	33.796796
LSTM	64.796280
BRF	34.267309
EGB-MGU-20	21.823166
EGB-GRU-3	**20.009821**
EGB-LSTM-2	25.736367

smaller than that of EGB-GRU. However, EGB-LSTM performs dissimilarly. After dramatically declining, the error of prediction drops gently, before the model being integrated five times and reaching convergence state. The phenomenon of rapid decline also exists in previous experiments. Although EGB-LSTM has the minimum prediction error in training dataset, it performs the worst in test set that can be found in Fig. 6(b). EGB-GRU has the fastest convergence speed and also the best performance, while EGB-MGU is in the middle. That shows a performance overturn of these three ensemble

models between training set and test set on GFP. Table IV displays all comparative RMSE values, where EGB-MGU-20, EGB-GRU-3 and EGB-LSTM-4 represent the best ensemble model in 20 integration experiments respectively. EGB-GRU-3 has the best ensemble result.

C. EXPERIMENTS SUMMARIES

In this section, three RNN models are embedded into EGB-RNN method and three ensemble models, EGB-MGU, EGB-GRU and EGB-LSTM, are constructed. The comparative experiments are conducted on DGS10, DFII20, AWH and GFP datasets. As Fig. 3, 4, 5 and 6 illustrating, the capability of proposed EGB-RNN method shows convergence, *i.e.*, with the number of integrated models increasing, EGB-RNN models are less and less affected by the number of supplemented models and eventually reach a convergent state. This character can help boost generalization in test data, but it does not mean over fitting cannot occur. During the process of integration, the best degree of ensemble varies with the type of RNN model and the dataset. Sometimes the integration is around 19 or 20, such as EGB-MGU-19, EGB-GRU-20 and EGB-LSTM-20 on DGS10 dataset, sometimes it is only 2 or 3, such as EGB-GRU-3 and EGB-LSTM-2 on GFP dataset. Moreover, in different experiments, the type of ensemble model that performs best is different. For example, EGB-LSTM outstrips the other two on DGS10 and DFII20, while EGB-GRU surpasses others on AWH and GFP.

From Table I, II and III, it can be found that proposed EGB-RNN method perform best than all other baselines. Compared with MGU, GRU and LSTM, it is undoubtable that the ensemble pattern outdo the single mode, which demonstrates the superiority of integration. Besides, traditional ensemble learning method GBDT and XGBoost are also outdone. The main reason is the use of RNN model as basic learner, while weak learner in those two ensemble strategy is decision tree. This difference reflects the strong power of deep learning method, because we employ a stronger basic learner RNN, which helps us go further. Moreover, EGB-RNN also outdoes a new proposed model BRF. This model embeds random forest, a type of ensemble decision tree tactic, into the frame of Adaboost. That is like under the basis of ensemble strategy, another ensemble model is embedded, but the proposed ensemble deep learning method is superior to it. To sum up, the established EGB-RNN technique is a competitive and high-precision method.

V. CONCLUSION

In this paper, a novel ensemble deep learning method called EGB-RNN is proposed to improve the prediction accuracy of time series data. This method combines gradient boosting as ensemble framework and RNN model as embedded base learner. Simultaneously, in order to averting fitting the train data unduly, the Step Iteration algorithm is designed to find appropriate learning rate in the process of integration. By introducing three different RNN models into the proposed method, three different ensemble models named EGB-MGU, EGB-GRU and EGB-LSTM can be obtained. In addition, comparative experiments are carried out on four data sets to verify the proposed ensemble method. For different dataset, EGB-RNN models present different manifestations. The choice of RNN model type and setting of ensemble number are sensitive to data. Nevertheless, appropriate EGB-RNN models can be found in acceptable times of trials, and experimental results demonstrates EGB-RNN method transcends all baselines.

Although the proposed EGB-RNN method can further improve the prediction accuracy, there is lack of intensive research on why EGB-RNN is susceptible to data. So, the explanatory work around this problem is one of the future works to consider, which contains the influence of statistical characteristics of data on the performance of model and so on. Meanwhile, not only boosting is a usual ensemble learning framework, but also bagging and stacking are included. Assembling deep learning models under other ensemble strategies is also a valuable research direction in future.

REFERENCES

[1] Huang L, Jiang H, Wang H. A novel partial-linear single-index model for time series data[J]. Computational Statistics & Data Analysis, 2019, 134: 110-122.

[2] Ramendra Prasad, Ravinesh C. Deo, Yan Li, Tek Maraseni. Soil moisture forecasting by a hybrid machine learning technique: ELM integrated with ensemble empirical mode decomposition[J]. Geoderma, 2018, 330: 136-161.

[3] Chen J, Zeng G Q, Zhou W, et al. Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization[J]. Energy Conversion and Management, 2018, 165: 681-695.

[4] Wu Y, Gao J. AdaBoost-based long short-term memory ensemble learning approach for financial time series forecasting[J]. Current Science (00113891), 2018, 115(1).

[5] Ruimin Xie, Kuangrong Hao*, Biao Huang*, Lei Chen, Xin Cai. Data-driven Modeling Based on Two-stream λ Gated Recurrent Unit Network with Soft Sensor Application, IEEE Transaction on Industrial Electronics, DOI: 10.1109/TIE.2019.2927197.

[6] Krishnan N, Cao J. 537 Estimation of Optimal Blank Holder Force Trajectories in Segmented Binders using an ARMA Model[C]//The Proceedings of the JSME Materials and Processing Conference (M&P) 10.2. The Japan Society of Mechanical Engineers, 2002: 391-396.

[7] Wang W, Chau K, Xu D, et al. Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition[J]. Water Resources Management, 2015, 29(8): 2655-2675.

[8] Wang L, Ting M, Chapman D, et al. Prediction of northern summer low-frequency circulation using a high-order vector auto-regressive model[J]. Climate Dynamics, 2016, 46(3-4): 693-709.

[9] Kim K. Financial time series forecasting using support vector machines[J]. Neurocomputing, 2003, 55(1-2): 307-319.

[10] Mellit A, Pavan A M, Benghanem M. Least squares support vector machine for short-term prediction of meteorological time series[J]. Theoretical and applied climatology, 2013, 111(1-2): 297-307.

[11] Chen K, Yu J. Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach[J]. Applied Energy, 2014, 113: 690-705.

[12] Jiang P, Wang Y, Wang J. Short-term wind speed forecasting using a hybrid model[J]. Energy, 2017, 119: 561-577.

[13] Ghorbani M A, Deo R C, Karimi V, et al. Design and implementation of a hybrid MLP-GSA model with multi-layer perceptron-gravitational search algorithm for monthly lake water level

forecasting[J]. Stochastic Environmental Research and Risk Assessment, 2019, 33(1): 125-147.

[14] Zhang C, Wei H, Xie L, Shen Y, Zhang K. Direct interval forecasting of wind speed using radial basis function neural networks in a multi-objective optimization framework. Neurocomputing 2016;205:53–63.

[15] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[16] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[17] Zhou G B, Wu J, Zhang C L, et al. Minimal gated unit for recurrent neural networks[J]. International Journal of Automation and Computing, 2016, 13(3): 226-234.

[18] Dong A, Du Z, Yan Z. Round Trip Time Prediction using Recurrent Neural Networks with Minimal Gated Unit[J]. IEEE Communications Letters, 2019.

[19] Dong X, Yu Z, Cao W, et al. A survey on ensemble learning[J]. Frontiers of Computer Science, 2020, 14(2): 241-258.

[20] Wang Q W, Yang L, Li Y F. Learning from weak-label data: A deep forest expedition[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(04): 6251-6258.

[21] Ganaie, M. A., and Minghui Hu. "Ensemble deep learning: A review." arXiv preprint arXiv:2104.02395 (2021).

[22] Wu Y, Gao J. AdaBoost-based long short-term memory ensemble learning approach for financial time series forecasting[J]. Current Science (00113891), 2018, 115(1).

[23] Friedman J H. Greedy function approximation: a gradient boosting machine[J]. Annals of statistics, 2001: 1189-1232.

[24] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of computer and system sciences, 1997, 55(1): 119-139.

[25] Breiman L. Classification and regression trees[M]. Routledge, 2017.

[26] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016: 785-794.

[27] Cao J, Li Z, Li J. Financial time series forecasting model based on CEEMDAN and LSTM[J]. Physica A: Statistical Mechanics and its Applications, 2019, 519: 127-139.

[28] Pak U, Kim C, Ryu U, et al. A hybrid model based on convolutional neural networks and long short-term memory for ozone concentration prediction[J]. Air Quality, Atmosphere & Health, 2018, 11(8): 883-895.

[29] Hao D , Xuejun Z , Zili Z . Commodity Price Forecasting Based on EEMD-LSTM-Adaboost[J]. Statistics & Decision, 2018, 13: 72-76

[30] Iwendi, Celestine, et al. "COVID-19 patient health prediction using boosted random forest algorithm." Frontiers in public health 8 (2020): 357.

[31] Sujath R, Chatterjee J M, Hassanien A E. A machine learning forecasting model for COVID-19 pandemic in India[J]. Stochastic Environmental Research and Risk Assessment, 2020, 34: 959-972.