The Astrophysical Journal Supplement Series, 248:14 (11pp), 2020 May https://doi.org/10.3847/1538-4365/ab880e

Prediction of the Dst Index with Bagging Ensemble-learning Algorithm

S. B. Xu[1], S. Y. Huang[1], Z. G. Yuan[1], X. H. Deng[2], and K. Jiang[1]
[1] School of Electronic Information, Wuhan University, Wuhan, People's Republic of China; shiyonghuang@whu.edu.cn
[2] Institute of Space Science and Technology, Nanchang University, Nanchang, People's Republic of China
Received 2019 September 13; revised 2020 February 26; accepted 2020 April 7; published 2020 May 5

Abstract

The Dst index is a commonly geomagnetic index used to measure the strength of geomagnetic activity. The accurate prediction of the Dst index is one of the main subjects of space weather studies. In this study, we use the Bagging ensemble-learning algorithm, which combines three algorithms—the artificial neural network, support vector regression, and long short-term memory network—to predict the Dst index 1–6 hr in advance. Taking solar wind parameters (including the interplanetary total magnetic field, magnetic field B_z component, total electric field, solar wind speed, plasma temperature, and proton pressure) as inputs, we establish the Dst index models and complete not only the point prediction but also the interval prediction in forecasting the Dst index. The results show that the root mean square error (rmse) of the point prediction is always lower than 8.0936 nT, the correlation coefficient (R) is always higher than 0.8572 and the accuracy of interval prediction is always higher than 90%, implying that our model can improve the accuracy of point prediction and significantly promote the accuracy of interval prediction. In addition, an new proposed metric shows that the Bagging algorithm brings better stability to the model. Our model was also used to predicate a magnetic storm event from 2016 October 12–17. The most accurate prediction of this storm event is the 1 hr ahead prediction, which holds a result with the rmse of 3.7327 nT, the correlation coefficient of 0.9928, and the interval prediction accuracy of 96.69%. Moreover, we also discuss the balance in the Bagging ensemble model in this paper.

Unified Astronomy Thesaurus concepts: Space weather (2037); Planetary magnetosphere (997); Geomagnetic fields (646); Solar wind (1534)

1. Introduction

At present, people are aware of the fact that geomagnetic activities have significant impacts on human life. For example, telephone systems, power grid transmission systems, and space satellites can be disturbed or damaged by strong geomagnetic activity (Boteler 2001; Baker et al. 2004). Therefore, the modeling and predicting of geomagnetic activity has become an important subject in the field of space weather (Khabarova 2007). As the main source of geomagnetic activity is solar activity, the solar wind can be considered as the medium through which the Sun exerts influence on the Earth. As a result, the solar wind parameters can be used as the main inputs to model the geomagnetic activity. Many literatures have studied the relationship between solar wind and the geomagnetic index and predicted the geomagnetic index with solar wind parameters as inputs (Li et al. 2007; Uwamahoro & Habarulema 2014).

The Dst index is a common geomagnetic index to quantify the geomagnetic activity of a time resolution of 1 hr (Sugiura 1964). It is defined as the difference between the horizontal component of the measured magnetic field and the horizontal component of the corresponding quiet geomagnetic field. To calculate the Dst index, five low-latitude geomagnetic observatories should be selected. These observatories would not be affected by the auroral electrojet and the equatorial electrojet, and their longitude should be roughly uniformly distributed in the whole longitude line. Then, the average of the calculation results of these five observatories can be taken as the Dst index. As a result, the Dst index can be used to describe the intensity of the equatorial ring current and can usually reflect the intensity of the magnetic storm. In this paper, the data of Dst index used in our study has been downloaded from NASA's National Space Science Data Center (https://nssdc.gsfc.nasa.gov/).

In the early research, Burton et al. (1975) have tried to model the Dst index by differential equations. Their model considered not only the number of particles injected from the plasma sheet into the inner magnetosphere but also the solar wind parameters and set them as the sole source of the differential equations.

Recently, as machine learning has developed rapidly and provides new methods for space weather modeling, a series of machine-learning and deep-learning methods are applied in the modeling and predicting of the Dst index (e.g., Camporeale 2019). The earliest Dst predicting model based on neural network was built by Lundstedt & Wintoft (1994). Then, Gleisner et al. (1996) used a time-delay neural network with the solar wind parameters as the inputs, which improved the prediction performance. Later, Wu & Lundstedt (1997) applied a special neural network called the Elman recurrent network (Elman 1990) to forecast the Dst index 1–6 hr in advance and found that the recurrent network structure could provide a good forecast of the prediction of geomagnetic activity. Based on an artificial neural network (ANN), Lazzús et al. (2017) used a particle swarm optimization algorithm to train ANN connection weights and obtained higher accurate predictions of the Dst index. These experiments have shown that the combination of a neural network and heuristic algorithms can further optimize the training process of the network. Ahmed et al. (2018), also based on an ANN, used the Levenberg–Marquardt (Levenberg 1944; Marquardt 1963)

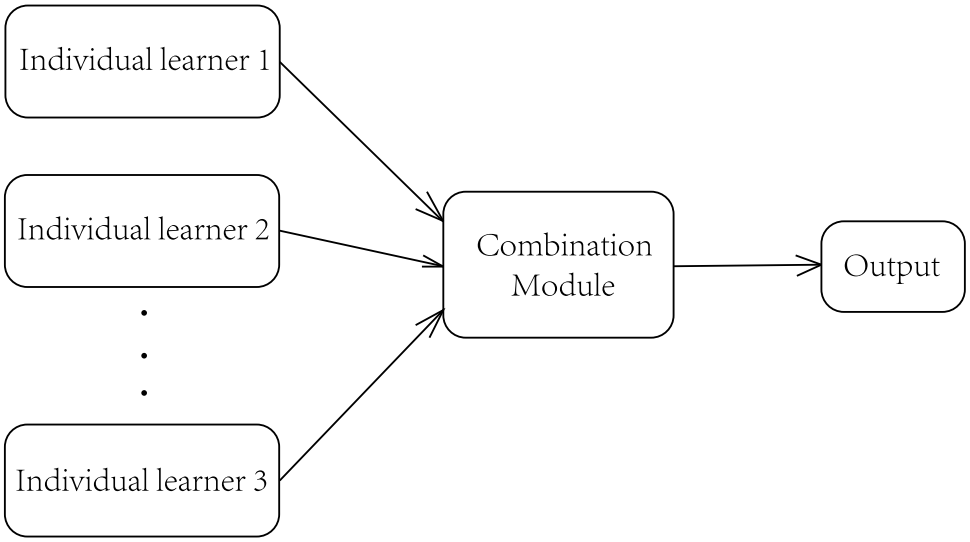

Figure 1. Structure of the ensemble algorithm.

training function to forecast the Dst index. Thus, these previous studies have demonstrated the success of neural networks in predicting the Dst index.

At the same time, some other algorithms also have been applied to the modeling and predicting of the Dst index, which have their own advantages and characteristics. Chandorkar et al. (2017) used a Gaussian process method to predict the Dst index. The highlight of this method is that it can predict the probability distribution of the Dst index and provide an interval prediction. Meanwhile, in order to improve the poor point prediction performance of Gaussian process method, Gruet et al. (2018) combined the Gaussian process method with the long short-term memory (LSTM) network to build the Dst index prediction model.

The aforementioned single machine-learning algorithms only can give point prediction, and it is almost impossible to further improve the prediction accuracy. Though the method combining the Gaussian process and the LSTM neuron network (NN) can provide interval predictions, the predicated intervals are too larger. In this study, we try to improve the accuracy of point prediction and also greatly optimize the prediction in two aspects: to decrease the prediction interval length and to increase the point prediction accuracy. Hence we use the Bagging ensemble algorithm to achieve the better probabilistic forecast of the Dst index. This algorithm gathers several machine-learning algorithms as component learners and adopts a strategy to combine the prediction results of component learners. In addition, we propose a special combination strategy, which could obtain the probability distribution forecast of the Dst index from a series of component learners.

This paper is divided as follows. Section 2 is a brief introduction of the computational method, including the Bagging ensemble algorithm and three component learning algorithms. Section 3 presents the models for forecasting and the data used in this study. Section 4 displays the results of the forecast 1–6 hr in advance. Section 5 discusses two kinds of balances between prediction accuracy and computational cost in our model. Section 6 presents the discussion and summary.

2. Computational Method

2.1. Bagging Algorithm

With the development of machine learning and deep learning, various algorithms have been proposed, which hold their own advantages, disadvantages, and applicable fields. In order to improve the performance, the community begins to combine various algorithms to participate in the construction of the model (Breiman 1996). This is the idea of the ensemble-learning algorithm. The ensemble algorithm accomplishes learning tasks by building and combining multiple learners to absorb the advantages of different learners. The structure, as shown in Figure 1, is usually built as follows: at first, a group of individual learners are generated, and then one uses strategy to combine them together.

The Bagging algorithm is a typical representative of the parallel algorithms in the ensemble-learning method (Breiman 1996). In accordance with the Bagging algorithm, individual learners do not interfere with the training process of other learner before the combination. In order to improve the combination effect, individual learners should be as independent and different with each other as possible, which is quite crucial in an ensemble-learning algorithm.

As for increasing the diversity of individual learners, the Bagging algorithm has its own sampling method: the bootstrap sampling method. We assume that the data set contains M samples. A random sample is selected into the sampling set and then put back into the initial data set, so that the next sampling may still select the selected samples. We repeat this action M times and will get a sample set with M samples. Due to the randomness of this sampling method, different learners could use different orders of training samples and even use different numbers of training samples. Hence the differences of the individual learners come from the different algorithms, different algorithm parameters, and different training samples. Thus, the bootstrap sampling method could greatly enhance the differences among individual learners and further improve the performance of the Bagging model.

We propose a combination strategy named probability distribution combination, which can combine the point predictions made by individual learners to obtain the prediction of probability distribution. First, we obtain the

prediction results of several individual learners, with the average value of the ith individual learners as $h_i(x)$ and the variance as the root mean square error (rmse) on the test set: rmse_i. Based on the central limit theorem and our experiments, we use the normal distribution of $h_i \sim \sigma(h_i(x), \text{RMSE}_i)$ to fit the prediction results of multiple individual learners. Finally, the normal distributions generated by the individual learners are averaged to obtain the integrated prediction distribution: H: $H = \frac{1}{T}\sum_{i=1}^{T} h_i$. In this probability distribution, each predicted value corresponds to a probability value, which has an integral value of 1 across the x-axis.

We also propose a method to get the point prediction from probability distribution. The center point that divides the whole distribution equally serves as the point prediction. It can be described by $\int_{-\infty}^{x_0} H(x)dx = \frac{1}{2}$. Finally, interval prediction can be obtained from probability distribution. Then the upper and lower alpha quantile of the probability distribution can be employed as the upper and lower boundary of the interval prediction, which can be described by $\int_{-\infty}^{x_1} H(x)dx = \alpha$, $\int_{-\infty}^{x_2} H(x)dx = 1 - \alpha$ (where α is a variable factor and its value can be set between 0.01 and 0.1).

2.2. ANN Method

Inspired by biological nerve systems, a neural network is a widely parallel interconnected network composed of adaptive simple units. In the field of machine learning, a neural network generally refers to the use of a neural network structure to train and learn data (Haykin 1994).

With an ANN used as the first algorithm of individual learners, the feedforward neural network structure is adopted in our study. In order to simplify and quantify the system, this structure is described in three layers: the input layer, the hidden layer, and the output layer. Either the input layer or the output layer is single layer, while the hidden layer has multiple sublayers. At the same time, it is stipulated that the neurons in the same layer cannot be connected, while only the neurons in the two adjacent layers can be connected. In order to make the network nonlinear, the sigmoid activation function (Kosko 1992; Poulton et al. 1992) is used in this study.

It is always a difficult task in the application of neural networks to select the number of hidden layers and the number of neurons in each layer. Since the ensemble algorithm used in this study allows multiple models to participate together, we train multiple network models with different numbers of the neuron, in which the number of hidden layer neurons ranges from 5 to 200.

There are some details we should explain in the building and training of ANN models. As we know, the training of the network is actually a process of modifying the connection weights and thresholds in the network under the gradient descent algorithm. Therefore, the initial weights and thresholds could affect the training result. In this study, we generate a cluster of random numbers that follows a standard normal distribution as the initial parameters. Meanwhile, the regularization term is added into the cost function (i.e., the mean square error between the calculated and observed values) to overcome overfitting and to improve generalization. We use the L2 regularization ($R = k\Sigma {w_i}^2$) and try a series of regularization coefficients (k), including 0.0001, 0.0003, 0.001, 0.003, 0.01, etc. in our training.

As for the iterations of gradient descent, the learning rate and batch size are two important parameters that influence the training process. A high learning rate would make gradient descent overshoot and miss the minimum, while a low learning rate results in slow learning. The batch size is defined in mini-batches learning as the training set is divided into several batches. In one iteration, the samples in the same batch are put together to calculate the gradient. Therefore, a large batch size means that some individual characteristics held by some unique samples would be ignored, while a small batch size would result in a low efficiency. After several tests and experiments, we choose 0.003 as the learning rate and 250 as the batch size in ANN models.

In addition, two types of gradient descent are employed for ANN individual learners, including the root mean square propagation algorithm (rmsprop) and adaptive moment estimation algorithm (Adam). These equations show the iteration principle in the tst iteration in rmsprop:

$$S_{dw} = \beta S_{dw} + (1 - \beta)dW^2$$
$$S_{db} = \beta S_{db} + (1 - \beta)db^2$$
$$W = W - \alpha\frac{dW}{\sqrt{S_{dw}} + \varepsilon} \quad b = b - \alpha\frac{db}{\sqrt{S_{db}} + \varepsilon}.$$

As S_{dw} and S_{db} are the gradient momentum accumulated by the cost function in the t-first iteration, β is a hyperparameter determined by the gradient accumulation, α is the learning rate, W and b are the weights and the thresholds in the networks, and ε is an extremely small value. The rmsprop algorithm uses the differential squared weighted average of the gradient dW^2 and db^2. This square term could eliminate the big swing amplitude in the secondary direction, which is not helpful to get close to the minimum value of the cost function. Therefore, the rmsprop could suppress the oscillating behavior of the cost function and make the function converge faster.

The Adam algorithm adds the momentum term in comparison to the rmsprop. The equations are shown as follows:

$$V_{dw} = \beta_1 V_{dw} + (1 - \beta_1)dW$$
$$V_{db} = \beta_1 V_{db} + (1 - \beta_1)db$$
$$S_{dw} = \beta_2 S_{dw} + (1 - \beta_2)dW^2$$
$$S_{db} = \beta_2 S_{db} + (1 - \beta_2)db^2$$
$$W = W - \alpha\frac{V_{dw}}{\sqrt{S_{dw}} + \varepsilon} \quad b = b - \alpha\frac{V_{db}}{\sqrt{S_{db}} + \varepsilon}.$$

The momentum term (v_{dw} and v_{db}) considers the inertia of gradient descent, which means the final gradient descent is determined by not only the current gradient but also by the previous gradient. With the combination of the momentum and rmsprop, the Adam algorithm increases the stability and accelerates the convergence of the cost function. Moreover, the Adam algorithm has the ability to get rid of the local optimum due to the momentum term.

2.3. SVR Method

Support vector regression (SVR) is a nonlinear method developed from traditional statistics (Cortes & Vapnik 1995), adopting the concept of a soft interval and tolerating the error within a certain range when calculating the regression cost. It

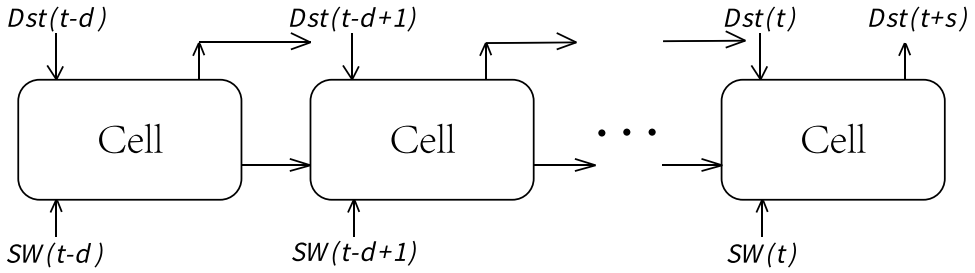

Figure 2. Prediction model of the LSTM NN.

could be represented by the following formulas,

$$\min_{\omega,b} C\sum_{i=1}^{m} l_\varepsilon(f(x_i) - y_i) + \frac{1}{2}\|\omega\|^2$$

$$l_\varepsilon(z) = \begin{cases} 0, & \text{if } |z| \leqslant \varepsilon \\ |z| - \varepsilon & \text{otherwise} \end{cases},$$

where x_i is the input vector, y_i is the target, $f(x_i)$ is the output of the model, and the penalty term C is a hyperparameter whose value is set before the learning process begins. The insensitive loss function l reflects the idea of the soft interval, and ε indicates the error that the SVR model can tolerate. ω is weight parameter vector, and the regularization term uses L2 norm of the weight parameter vector $\|\omega\|^2$. In our study, different values of C ranging from 1 to 100 are used to build models.

The SVR method uses a kernel function to project the features of samples in the sample space of higher dimensions for regression, which makes this method better adapted to nonlinear problems. In our study, we employ different kernel functions and different kernel function parameters to construct multiple models, including a polynomial kernel, Gaussian kernel, and sigmoid kernel:

Polynomial kernel: $\kappa(x_i, x_j) = (x_i{}^T x_j)^d$.

Gaussian kernel: $\kappa(x_i, x_j) = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right)$.

Sigmoid kernel: $\kappa(x_i, x_j) = \tanh(\beta x_i{}^T x_j + \theta)$.

2.4. LSTM Method

The LSTM NN is a new network structure proposed by Hochreiter & Schmidhuber (1997), which belongs to the structure of the recurrent neural network (RNN). The RNN would remember the information before the current time and apply it to the calculation of the current output. Under this circumstance, the neurons between the same hidden layers in the network are connected, and the input of the hidden layer includes not only the input of the current moment but also the output of the hidden layer at the previous moment. Hence this structural feature mentioned previously enables the RNN to deal well with the classification and regression problems in which inputs are time series.

In this study, the LSTM NN is further optimized on the basis of the RNN. Compared with the RNN, the LSTM network has stronger memory ability for information with long time distances. The LSTM NN structure introduces the cell state to record information passing over time. The recurrent structure of LSTM relies on some "gate" structures to allow information to selectively affect the state of each moment in the neural network, so as to preserve long-term memory. Among these "gates," the function of the forget gate is to make the network selectively forget the previous useless information. While the input gate supplements the new memory from the current input, the final output gate calculates the output according to the cell state.

The LSTM NN method is adopted as the third individual learner algorithm of the Bagging algorithm. Similar to the ANN method, we train multiple LSTM network models with different numbers (from 5 to 100) of the neurons based on the rmsprop algorithm. In these models, the β in the rmsprop is set to 0.9, with the batch size set to 100 and the learning rate set to 0.005, but the regularization term is not used in the LSTM NN models.

Table 1
Bagging Performance in Comparison to Previous Models

Root Mean Square Error (rmse)				
	Bagging Model	Gruet et al. (2018)	Ahmed et al. (2018)	Lazzús et al. (2017)
t+1h	2.85	5.25	8.60	4.24
t+2h	4.80	6.55	8.02	7.05
t+3h	6.11	7.59	8.05	8.87
t+4h	7.03	8.53	8.16	10.44
t+5h	7.65	9.18	8.19	11.65
t+6h	8.09	9.86	8.23	13.09
Correlation Coefficient (R)				
	Bagging Model	Gruet et al. (2018)	Ahmed et al. (2018)	Lazzús et al. (2017)
t+1h	0.983	0.966	0.845	0.982
t+2h	0.952	0.946	0.874	0.949
t+3h	0.921	0.928	0.872	0.918
t+4h	0.895	0.910	0.869	0.887
t+5h	0.874	0.892	0.865	0.858
t+6h	0.857	0.873	0.864	0.826

3. Data and Forecasting Model

3.1. Data

The Dst index data and solar wind parameter data used in this study, from 2007 January to 2017 May, have been obtained from NASA's National Space Science Data Center (https://nssdc.gsfc.nasa.gov/). The time resolution of the Dst index and solar wind parameters is set to one hour. Solar wind parameters include the following used in the previous studies (Ahmed et al. 2018): the interplanetary magnetic field (B), the Z component of the interplanetary magnetic field (B_z) in the GSE coordinate system, the plasma temperature (T), the plasma density (D), the plasma velocity (V), the plasma pressure (P), and the electric

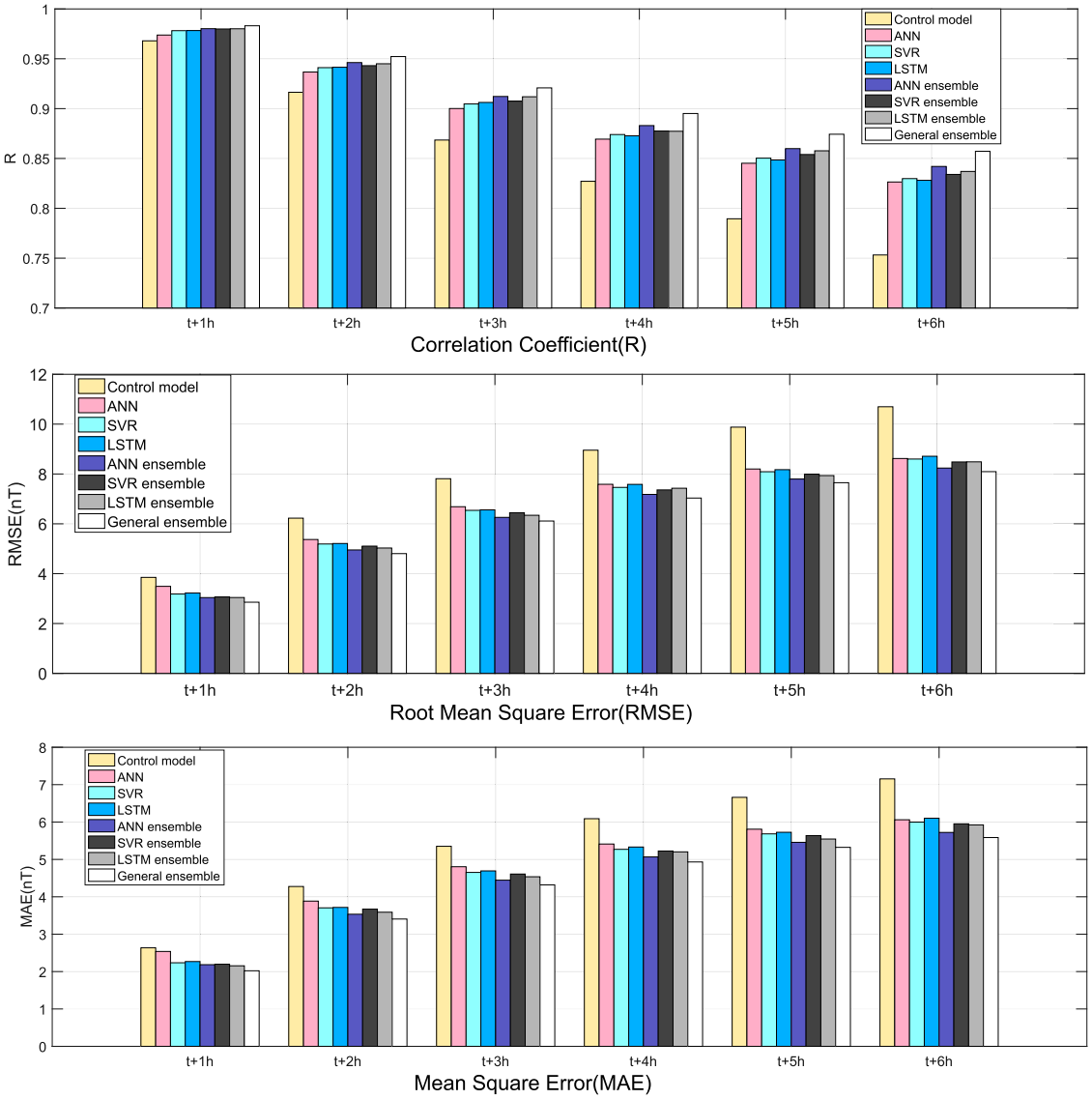

Figure 3. Prediction results of several models.

field intensity (E). The data set contains a total of 91,296 hr of data.

The data set is divided into two parts. The first part is the training set part, which contains data of 78,888 hr from 00:30 on 2007 January 1 to 23:30 on 2015 December 31. We use the bootstrap sampling method to sample data from the training set as the training input, and the samples that are in the training set but not be sampled as the training input are used as the test set. The second part is the prediction set part, which contains 12,408 hr of data from 00:30 on 2016 January 1 to 23:30 on 2017 May 31.

3.2. Prediction Model

Before the building of the prediction model, there are several variables that need to be set in advance. The prediction time variable is represented by s, meaning we would predict the Dst index ahead of s hours. This study predicts the Dst index ahead of 1–6 hr: $s \in [1, 6]$. The input time span is represented by d, meaning the time span of the input data of the model is d hours.

In this study, the initial ANN model was established as follows: the solar wind data and the Dst data of *0-d* hours before the current time are taken as the input of the network, and the Dst data after s hours are taken as the network output:

$$\mathrm{Dst}(t+s) = \mathrm{ANN}(\mathrm{SW}(t), \mathrm{SW}(t-1),\ldots\mathrm{SW}(t-d), \mathrm{Dst}(t), \mathrm{Dst}(t-1),\ldots,\mathrm{Dst}(t-d)).$$

Because the number of hidden layer neurons (denoted by N) in the ANN model is an important hyperparameter, we add it into the model. Meanwhile, time-dependent t can be removed from the equation. Therefore, the model is simplified as follows:

$$\mathrm{Dst}(N, s) = \mathrm{ANN}(\mathrm{SW}(d), \mathrm{Dst}(d), N).$$

Similarly, the establishment of the SVR model is shown as follows:

$$\mathrm{Dst}(t+s) = \mathrm{SVR}(\mathrm{SW}(t), \mathrm{Dst}(t), \mathrm{Dst}(t-1),\ldots,\mathrm{Dst}(t-d)).$$

If time-dependent t is removed, then

$$\mathrm{Dst}(s) = \mathrm{SVR}(\mathrm{SW}, \mathrm{Dst}(d)).$$

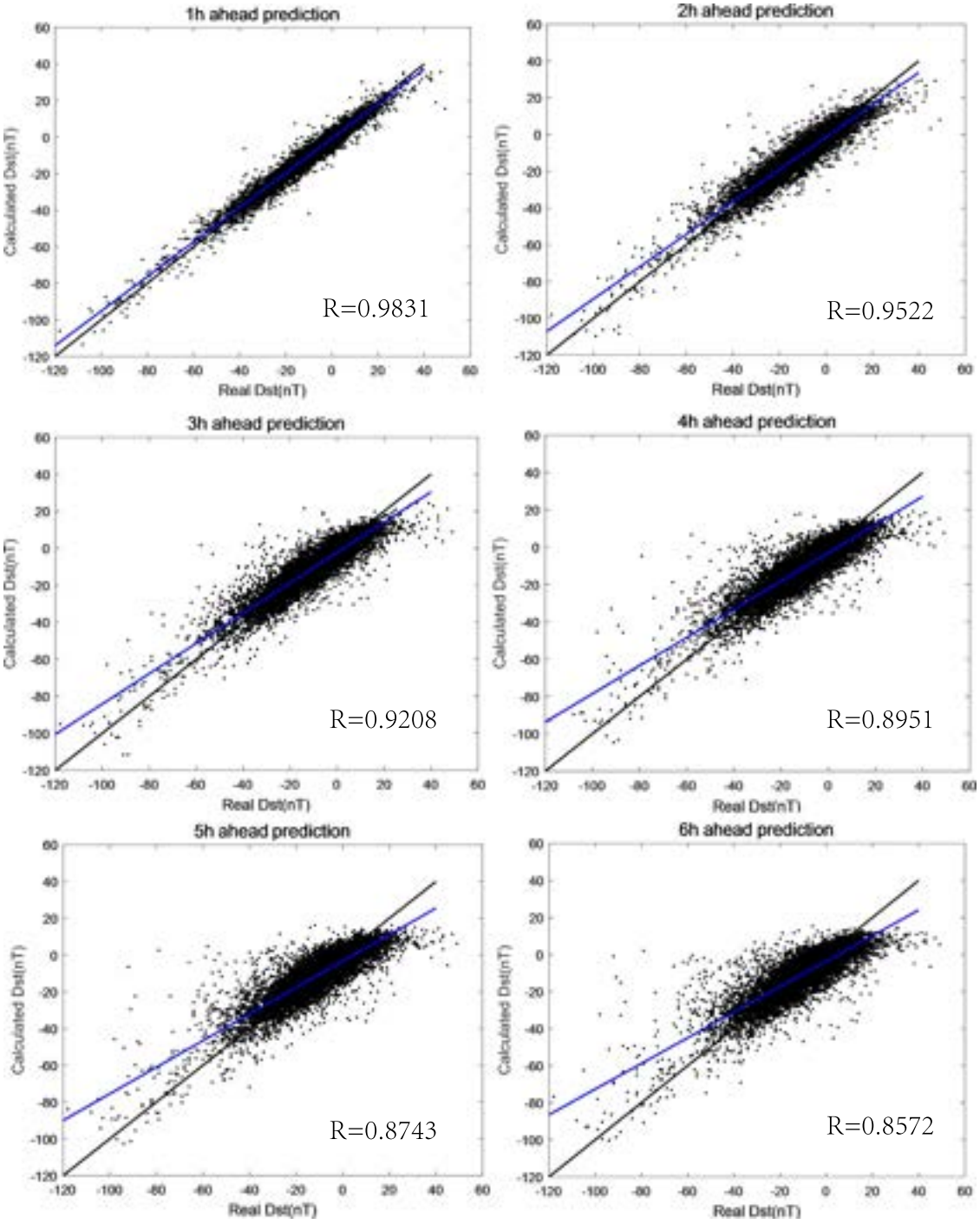

Figure 4. Comparison between the prediction Dst values based on the Bagging ensemble-learning algorithm and the real Dst values. The blue line presents the fitted result, and the black line is the linear curve (i.e., real Dst = prediction Dst).

The input data of LSTM NN has a relationship of time series, making it different from the above two models. It is not rigorous to use a function in which all input parameters do not have the timing relationship to express this model. Therefore, we present this model in Figure 2, in which the cell is the recurrent structure of LSTM and the input parameters (the Dst index and solar wind parameters) are input into the network in chronological order. Thus, the number of the neurons in the hidden layer of parameters (N) was added into the model, and time-dependent t was removed. As a result, the model was shown as follows:

$$\mathrm{Dst}(N, s) = \mathrm{LSTM}(\mathrm{SW}(d), \mathrm{Dst}(d), N).$$

During the establishing of the Bagging ensemble model, the integration process is divided into two steps. The first step is to integrate multiple learners of the same individual learner algorithm with different training parameters. Hence the corresponding model formulas of all the three methods (i.e., the ANN ensemble model, SVR ensemble model, and LSTM

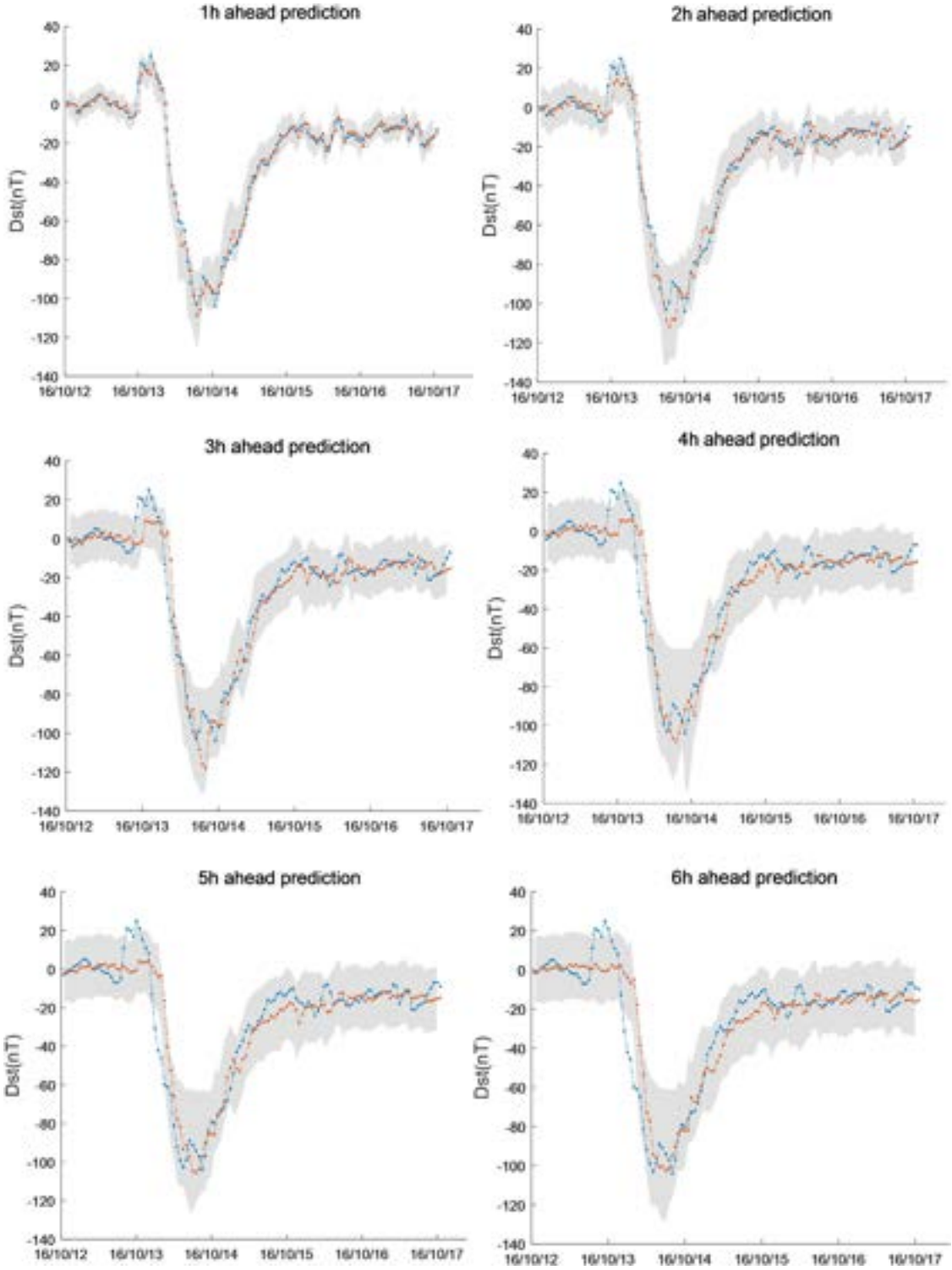

Figure 5. Prediction of results in a geomagnetic storm event. The blue dotted curves present the real Dst, and the dotted curves display the prediction Dst of our model. The gray shaded area displays the results of interval prediction.

ensemble model) are shown as follows:

$$\begin{aligned}&\mathrm{Dst}(s) = \mathrm{Bag}(\mathrm{ANN}_1(\mathrm{SW}(d), \mathrm{Dst}(d), N_1),\ldots,\\&\mathrm{ANN}_m(\mathrm{SW}(d), \mathrm{Dst}(d), N_m)\\&\mathrm{Dst}(s) = \mathrm{Bag}(\mathrm{SVR}_1(\mathrm{SW}, \mathrm{Dst}(d)),\ldots,\\&\mathrm{SVR}_m(\mathrm{SW}, \mathrm{Dst}(d))\\&\mathrm{Dst}(s) = \mathrm{Bag}(\mathrm{LSTM}_1(\mathrm{SW}(d), \mathrm{Dst}(d), N_1),\ldots,\\&\mathrm{LSTM}_m(\mathrm{SW}(d), \mathrm{Dst}(d), N_m).\end{aligned}$$

The second step is the overall integration of learners with different kinds of algorithms, called the general ensemble model, is shown as follows:

$$\begin{aligned}\mathrm{Dst}(s) = \mathrm{Bag}(&\underset{i\in[1,m]}{\mathrm{ANN}_i}(\mathrm{SW}(d), \mathrm{Dst}(d), N_i),\\&\underset{i\in[1,m]}{\mathrm{SVR}_i}(\mathrm{SW}, \mathrm{Dst}(d)), \underset{i\in[1,m]}{\mathrm{LSTM}_i}(\mathrm{SW}(d), \mathrm{Dst}(d), N_i)).\end{aligned}$$

Table 2
Double-error Proportion of the Predictions Made by Several Models

	Bagging Model	ANN Model	SVR Model	LSTM Model
t+1h	3.59%	5.51%	4.49%	4.96%
t+2h	3.89%	5.12%	4.71%	4.98%
t+3h	3.87%	4.86%	4.88%	5.16%
t+4h	3.81%	4.90%	4.80%	4.93%
t+5h	3.93%	4.99%	5.14%	5.18%
t+6h	4.00%	5.12%	5.50%	5.06%

In the process of training, testing, and predicting, we need to calculate the errors to quantify the advantages and disadvantages of the algorithm. Three kinds of measure standards are calculated in this study, namely the rmse, mean absolute error (MAE), and correlation coefficient (R). The calculation formula is shown as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=0}^{n}(\mathrm{Dst}_i^{\mathrm{Cal}} - \mathrm{Dst}_i^{\mathrm{Real}})^2}{n}}$$

$$\mathrm{MAE} = \frac{\sum_{i=0}^{n}|\mathrm{Dst}_i^{\mathrm{Cal}} - \mathrm{Dst}_i^{\mathrm{Real}}|}{n}$$

$$R = \frac{\sum_{i=0}^{n}(\mathrm{Dst}_i^{\mathrm{Cal}} - \overline{\mathrm{Dst}_i^{\mathrm{Cal}}})(\mathrm{Dst}_i^{\mathrm{Real}} - \overline{\mathrm{Dst}_i^{\mathrm{Real}}})}{\sqrt{\sum_{i=0}^{n}(\mathrm{Dst}_i^{\mathrm{Cal}} - \overline{\mathrm{Dst}_i^{\mathrm{Cal}}})^2 \sum_{i=0}^{n}(\mathrm{Dst}_i^{\mathrm{Real}} - \overline{\mathrm{Dst}_i^{\mathrm{Real}}})^2}}.$$

3.3. The Implementation of the Models

We use TensorFlow to build and train the machine-learning models. TensorFlow, created and released by Google, is a Python library for fast numerical computing. With the computational graph defined by TensorFlow, the calculation and the training of machine-learning models could be simplified. The hardware we used is the 8 core CPU with a core frequency of 3.00 GHz.

In machine learning, the computational cost also becomes our concern, which is related to the stopping strategy used in the gradient descent process. In our study, we adopt a complex strategy to stop the gradient descent. First, we set an expected error and a maximum number of iterations. If the error of the model is smaller than the expected error or the number of iterations exceed the limit set, the gradient descent will be stopped. Usually we set the maximum number of iterations as 10^6. Moreover, another rule is adopted to decrease the computational cost—that is, the training would be stopped when the cost function cannot reach smaller during 20,000 steps. Under this stopping strategy, the computational cost of these models ranges from 30 minutes to 10 hr, and the average cost is roughly 3 hr per individual learner.

4. Results

Before presenting the results, we set a control model to serve as a baseline. This control model uses the previous values of the Dst index as the prediction for the next step: $\mathrm{Dst}(t+s) = \mathrm{Dst}(t)$ which represents the prediction result that spends zero computational cost and provides reference and comparison for the results of our model.

We present the prediction of the Dst index with the ANN model, SVR model, LSTM model, ANN ensemble model, SVR ensemble model, LSTM ensemble model, and ensemble model as described in Figure 3. The results include the correlation coefficient (R), rmse, and MAE between the predicted results and the actual real results. By comparing the prediction results of multiple models, we find that the prediction accuracy of the ensemble model for the Dst index, compared with the individual learner model, is significantly improved.

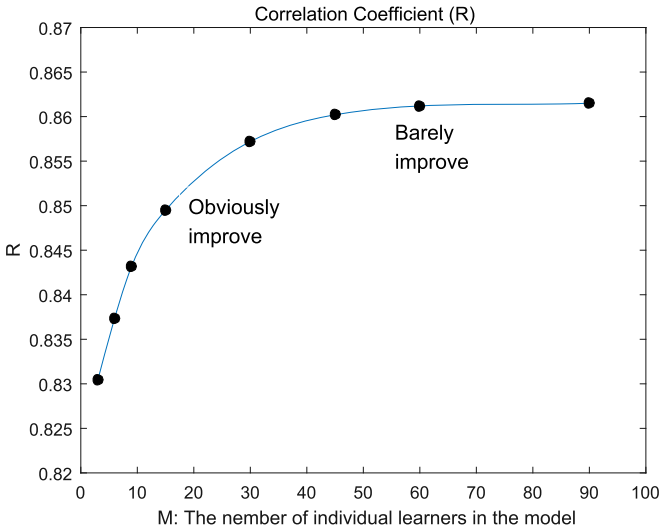

Figure 6. Correlation coefficient of the prediction results from our model composed by the different numbers of individual learners.

Table 3
Statistical Analysis of Prediction Results during a Geomagnetic Storm Event

	rmse (nT)	MAE (nT)	R	Length of Prediction Interval (nT)	Accuracy (%)
t+1h	3.7327	2.5607	0.9928	17.8236	96.69
t+2h	6.5676	4.4703	0.9783	25.8637	95.87
t+3h	8.3959	5.7308	0.9645	31.1267	95.04
t+4h	9.5575	6.6374	0.9516	36.1033	92.56
t+5h	10.7288	7.3721	0.9256	37.5064	93.78
t+6h	12.0663	7.9379	0.9203	39.6033	91.74

Table 1 compares the performance of the Bagging model with that of the previous model. The third array is the LSTM model created by Gruet et al. (2018), which combines the LSTM NN and Gaussian process and represents the best prediction of the Dst index. Our model also uses the LSTM NN as one kind of individual learner and provides an approximate performance in the correlation coefficient and a better performance in the rmse compared with the model by Gruet et al. (2018). Based on the ANN, Lazzús et al. (2017) used particle swarm optimization algorithm in their model, and Ahmed et al. (2018) used the Levenberg–Marquardt algorithm in their model. Both of the two models could represent the good level of Dst prediction only using the ANN, but their performances are worse than our Bagging model in not only rmse but also in the correlation coefficient. Moreover, it is a pity that the SVR method was not used to predict the Dst index in previous research. In our study, we built the SVR model and compared it with the Bagging model. Obviously, the result shows the superiority of the Bagging algorithm again.

Figure 4 presents the fitting between the real value of the Dst index and the values predicted by the general ensemble model.

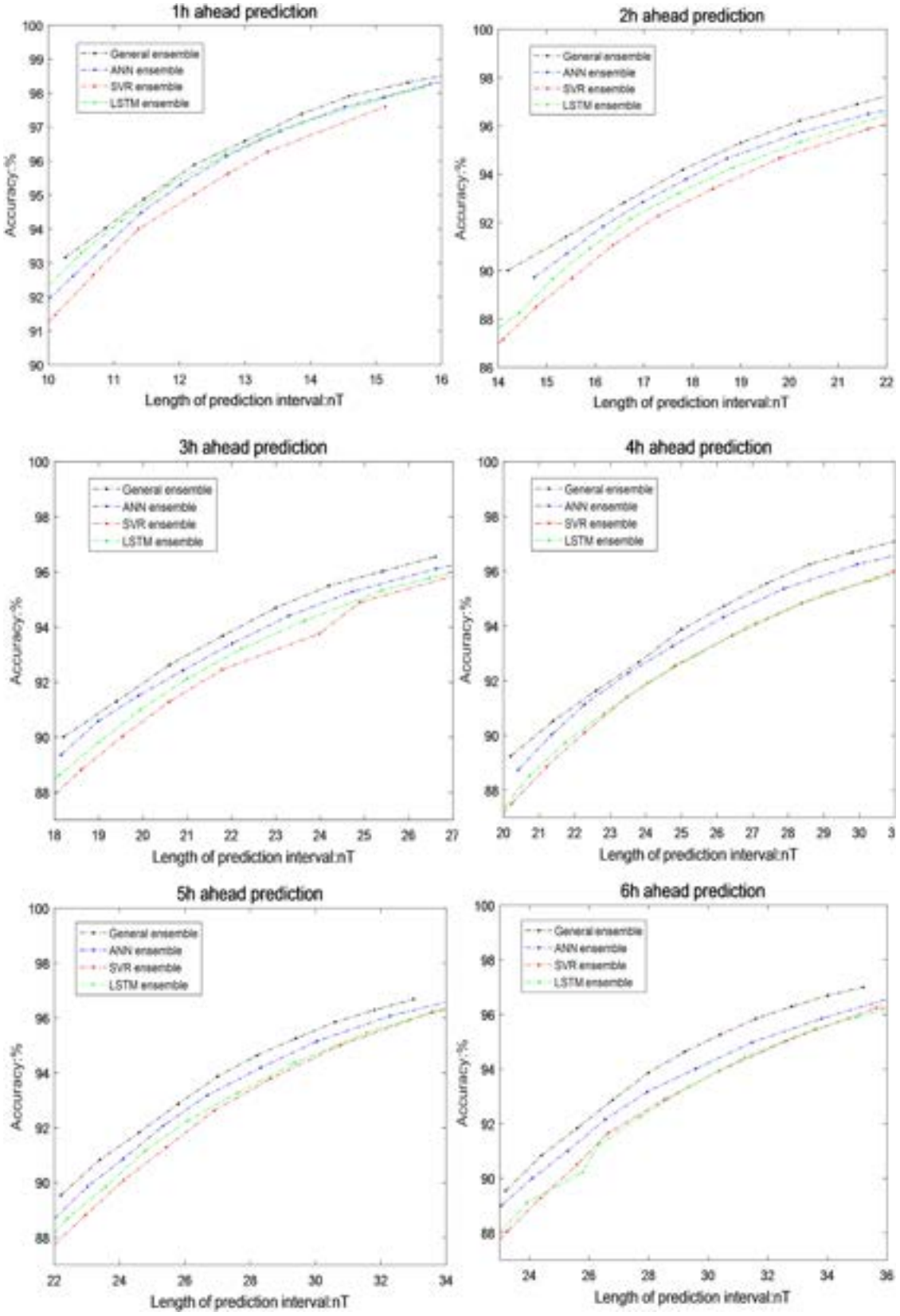

Figure 7. Relationship between the interval length and the accuracy of the prediction interval.

The black line draws the function $y = x$, which serves as a reference. When the scatter points are close to the line, it means that the predicted results are close to the real ones. Meanwhile, there is also a blue line in Figure 4, which is fitted by the scatter graph using the least-squares method. This line can be used to measure the prediction effect on the whole prediction set. When the fitted line is close to the reference line, the overall prediction has good performance. With the increase of the

prediction time s, the scatter points of the prediction gradually scatter away from the reference line, indicating that the prediction accuracy decreases gradually. At the same time, the slope of the fitting line is generally less than 1 and decreases gradually with the increase of the prediction time s, indicating that the model is conservative in predicting the extreme value of the Dst index; in other words, it is not easy to make a radical prediction.

As the errors in the entire test set would constitute a distribution, rmse reflects the mean of the error distribution, which determines the performance of the model. However, the limitation of rmse is that it cannot illustrate the variance of error distribution, which corresponds to the model's stability. Therefore, we propose a new parameter called double-error proportion (DEP). It is defined by the proportion that the predicting errors are greater than twice rmse:

$$\mathrm{DEP} = P(|\text{ prediction values} - \text{real values}| > 2\mathrm{rmse}).$$

A large DEP means that the error distribution holds a big variance, which corresponds to the less instability of the model. As Table 2 shows the calculated DEP of each model, it is easy to find that the Bagging model provides the most stable predictions in comparison to other models.

We have investigated the strongest geomagnetic storm event in the test set. Figure 5 shows the event prediction of the general ensemble model. The storm event began on 2016 October 12 and ended on 2016 October 17, with the minimum Dst reaching −120 nT. In Figure 5, the blue curve presents the real values of Dst, the red curve represents the prediction values, and the gray area represents the prediction range. It can be found that the blue curve representing the real Dst value is basically covered by the gray area, and the predictions 1–6 hr ahead hold the accuracy higher than 90%, indicating a good overall prediction effect. Meanwhile, the average length of the prediction interval of the 1–6 hr ahead prediction ranges from 17.8236 to 39.6 nT (as shown in Table 3), which is much smaller compared with the general range of the Dst index from −120 to 20 nT. Thus, the interval prediction of this model has practical application value.

5. Balance in the Bagging Ensemble Model

In order to obtain higher accuracy and higher prediction accuracy, some costs are often necessary in machine learning. Therefore, it is an important issue to balance the relationship between cost accuracy and prediction accuracy. In our Bagging algorithm model, we employ multiple individual learners to jointly participate in the prediction of the Dst index and the number M of individual learners is a variable that can be adjusted. The larger M is, the longer the computational time and the higher the computational cost we pay to build the model. The time complexity of building the model is $O(M)$. At the same time, the more individual learning machines there are, the more accurate the model prediction will be. Taking 6 hr ahead prediction as an example, Figure 6 shows the prediction results (the correlation coefficient) of general ensemble models with different values of M. It can be seen that the prediction performance of the model improves rapidly with the increase of M when M is small. When M is larger than 40, the prediction performance improvement brought by the increase of M is limited.

There is another relationship that needs to be balanced in our model. When our model forecasts the Dst index, according to the formula given above, $\int_{-\infty}^{x_1} H(x)dx = \alpha$, $\int_{-\infty}^{x_2} H(x)dx = 1 - \alpha$, we can regulate the value of α to adjust the length of the prediction interval. Obviously, when the interval is larger, the accuracy of the interval prediction is higher. However, a large prediction range is meaningless in practice. Therefore, one need to strike a balance between the length of the prediction interval and the accuracy of the prediction interval. Figure 7 shows the relationship between the prediction interval length and prediction accuracy when four models (the ANN ensemble model, SVR ensemble model, LSTM ensemble model ,and Bagging general ensemble model) predict the Dst index 1–6 hr in advance. Within a certain range, the relationship between the length of the prediction interval and the accuracy of the prediction interval is approximately linear. In the forecasting process, we can select the accuracy and the interval length freely according to the need of forecasting.

6. Discussion and Conclusion

In this paper, we build one model to predict the Dst index 1–6 hr ahead based on the Bagging ensemble-learning algorithm. We use three popular machine-learning algorithms (the ANN method, SVR method, and LSTM NN method) to provide Dst predictions and each algorithm has its own advantages and disadvantages. We adopt Bagging ensemble algorithm to combine the three algorithms to participate in the prediction together and propose the probability distribution combination strategy to obtain the probability distribution prediction from several point predictions.

Based on three types of measure standards (rmse, MAE, and R), the results show the Bagging model provides better performance in comparison to other methods. Stability analyzation is also achieved with a new metric. We calculate the DEP of these models and find the smallest DEP for the Bagging model, implying that the Bagging algorithm can stabilize the prediction model to a certain extent.

We also compare the Bagging model to the models in previous studies. As for point prediction, our model could achieve the best prediction with minimum rmse and MAE. In recent studies, Gruet et al. (2018) combined the Gaussian process method with LSTM and got the 6 hr ahead Dst prediction with 9.86 rmse, while Ahmed et al. (2018) used an ANN and got the 6 hr ahead prediction with 8.2332 rmse and 6.21 MAE. Our model shares the 6 hr ahead prediction result with 8.0936 rmse and 5.5849 MAE, indicating that our point prediction error has been much more reduced. As for the interval prediction, most of the previous studies cannot provide interval forecast except only two studies, i.e., the studies by Chandorkar et al. (2017) and Gruet et al. (2018). However, their interval predictions held lower accuracy and large interval length. In our models, we could obtain the interval prediction with the accuracy of higher than 90% with the length of interval prediction also relatively short.

Previous machine-learning algorithms have reached a high level in the prediction of the Dst index. It is quite difficult to propose an innovative method that could achieve great improvement. But the Bagging algorithm sheds new lights on this issue and focuses on combining the advantages of multiple algorithms and could achieve better performance in comparison to all single algorithms. We hope that the Bagging algorithm could be applied to resolve other astrophysical problems.

This study is supported by the National Natural Science Foundation of China (41674161, 41874191, and 41925018), Young Elite Scientists Sponsorship Program by CAST (2017QNRC001), and the National Youth Talent Support Program. Dst and solar wind data are publicly available from NASA's National Space Science Data Center (https://nssdc.gsfc.nasa.gov/).

References

Ahmed, L., El-Eraki, M. A., Aalaa, S., et al. 2018, SpWea, 16, 1277
Baker, D. N., Daly, E., Daglis, I., Kappenman, J. G., & Panasyuk, M. 2004, SpWea, 2, S02004
Boteler, D. H. 2001, GMS, 125, 347
Breiman, L. 1996, Mach. Learn., 24, 123
Burton, R. K., Mcpherron, R. L., & Russell, C. T. 1975, JGR, 80, 4204
Camporeale, E. 2019, SpWea, 17, 1166
Chandorkar, M. H., Camporeale, E., & Wing, S. P. 2017, SpWea, 15, 1004
Cortes, C., & Vapnik, V. 1995, Mach. Learn., 20, 273
Elman, J. L. 1990, Cognit. Sci., 14, 179
Gleisner, H., Lundstedt, H., & Wintoft, P. 1996, AnGeo, 14, 679
Gruet, M. A., Chandorkar, M., Sicard, A., & Camporeale, E. 2018, SpWea, 16, 1882
Haykin, S. 1994, Neural Networks: A Comprehensive Foundation (Upper Saddle River, NJ: Prentice Hall)
Hochreiter, S., & Schmidhuber, J. 1997, Neural Comput., 9, 1735
Khabarova, O. 2007, SunGe, 2, 33
Kosko, B. 1992, Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence/Book and Disk (Upper Saddle River, NJ: Prentice Hall)
Lazzús, J. A., Vega, P., Rojas, P., & Salfate, I. 2017, SpWea, 15, 1068
Levenberg, K. 1944, QApMa, 2, 164
Li, X., Oh, K. S., & Temerin, M. 2007, JGR, 112, A06224
Lundstedt, H., & Wintoft, P. 1994, AnGeo, 12, 19
Marquardt, D. W. 1963, J. Soc. Industrial Applied Math., 11, 431
Poulton, M. M., Sternberg, B. K., & Glass, C. E. 1992, Geop, 57, 1534
Sugiura, M. 1964, Ann. Int. Geophys. Year, 35, 945
Uwamahoro, J. C., & Habarulema, J. B. 2014, EP&S, 66, 95
Wu, J. G., & Lundstedt, H. 1997, JGR, 102, 14255