ADVANCES IN ATMOSPHERIC SCIENCES, VOL. 36, OCTOBER 2019, 1156–1170

• Original Paper •

A Model Output Machine Learning Method for Grid Temperature Forecasts in the Beijing Area

Haochen LI[1,4] , Chen YU[1] , Jiangjiang XIA[2] , Yingchun WANG[3] , Jiang ZHU[2] , and Pingwen ZHANG[*1]

[1]*School of Mathematical Sciences*, *Peking University*, *Beijing* 100871, *China*

[2]*Institute of Atmospheric Physics Chinese Academy of Sciences*, *Beijing* 100029, *China*

[3]*Beijing Meteorological Service*, *Beijing* 100089, *China*

[4]*School of Science*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*

(Received 2 February 2019; revised 15 April 2019; accepted 13 May 2019)

ABSTRACT

In this paper, the model output machine learning (MOML) method is proposed for simulating weather consultation, which can improve the forecast results of numerical weather prediction (NWP). During weather consultation, the forecasters obtain the final results by combining the observations with the NWP results and giving opinions based on their experience. It is obvious that using a suitable post-processing algorithm for simulating weather consultation is an interesting and important topic. MOML is a post-processing method based on machine learning, which matches NWP forecasts against observations through a regression function. By adopting different feature engineering of datasets and training periods, the observational and model data can be processed into the corresponding training set and test set. The MOML regression function uses an existing machine learning algorithm with the processed dataset to revise the output of NWP models combined with the observations, so as to improve the results of weather forecasts. To test the new approach for grid temperature forecasts, the 2-m surface air temperature in the Beijing area from the ECMWF model is used. MOML with different feature engineering is compared against the ECMWF model and modified model output statistics (MOS) method. MOML shows a better numerical performance than the ECMWF model and MOS, especially for winter. The results of MOML with a linear algorithm, running training period, and dataset using spatial interpolation ideas, are better than others when the forecast time is within a few days. The results of MOML with the Random Forest algorithm, year-round training period, and dataset containing surrounding gridpoint information, are better when the forecast time is longer.

Key words: temperature forecasts, MOS, machine learning, multiple linear regression, Random Forest, weather consultation, feature engineering, data structures

Citation: Li, H. C., C. Yu, J. J. Xia, Y. C. Wang, J. Zhu, and P. W. Zhang, 2019: A model output machine learning method for grid temperature forecasts in the Beijing area. *Adv. Atmos. Sci.*, **36**(10), 1156−1170, https://doi.org/10.1007/s00376-019-9023-z.

Article Highlights:

- The MOML method is proposed for improving NWP forecasts.
- MOML is a machine learning–based post-processing method that matches NWP forecasts against observations through a regression function.
- MOML can make full use of the spatial and temporal structure of a point on the grid. It is used here to forecast the 2-m surface air temperature in the Beijing area.
- With proper feature engineering, forecasts based on MOML perform better than the traditional MOS method, especially for winter.

1. Introduction

Weather forecasting plays an important role in many fields, such as agriculture, transportation, industry, commerce, atmospheric science research, and so on. In the past, people have forecast the weather by using meteorological knowledge, statistics and the observational data collected at weather stations. Numerical weather prediction (NWP) has made great progress in the last 50 years with the development of computer technology, modeling techniques, and observations (Molteni et al., 1996; Toth and Kalnay, 1997). Nevertheless, NWP model forecasts contain systematic biases due to imperfect model physics,

* Corresponding author: Pingwen ZHANG
Email: pzhang@pku.edu.cn

initial conditions, and boundary conditions (Paegle et al., 1997; Mass et al., 2002; Hart et al., 2003; Cheng and Steenburgh, 2007; Rudack and Ghirardelli, 2010).

Because the output of NWP and observations have different systematic errors, the forecasting performance for various regions, seasons and weather processes is different. Before the release of a weather forecast, in order to further improve its accuracy, a weather consultation is indispensable. The forecasters obtain the final results during this weather consultation by combining observations with the NWP results and giving opinions based on their experience. The process of weather consultation is actually a manual process of post-processing the NWP results, and thus the professional knowledge and practical experience of individuals have a crucial impact on the forecast results. Owing to the current increase in data size and the improvement of weather forecasting requirements, the current weather consultation model cannot meet the needs of the development of weather forecasting, and so suitable post-processing algorithms are needed to help the manual process of weather consultation (Hart et al., 2003; Cheng and Steenburgh, 2007; Wilks and Hamill, 2007; Veenhuis, 2013).

In order to remove systematic errors and improve the output from NWP models, a variety of post-processing methods have been developed for simulating weather consultation (Wilks and Hamill, 2007; Veenhuis, 2013)—for example, model output statistics (MOS) (Glahn and Lowry, 1972; Cheng and Steenburgh, 2007; Wu et al., 2007; Glahn et al., 2009; Jacks et al., 2009; Zhang et al., 2011; Glahn, 2014; Wu et al., 2016), the analog ensemble (Monache et al., 2013; Alessandrini et al., 2015; Junk et al., 2015; Plenković et al., 2016; Sperati et al., 2017), the Kalman filter (Delle Monache et al., 2011; Cassola and Burlando, 2012; Bogoslovskiy et al., 2016; Buehner et al., 2017; Pelosi et al., 2017), anomaly numerical-correction with observations (Peng et al., 2013, 2014), among which MOS is one of the most commonly used to produce unbiased forecasts (Glahn et al., 2009). MOS uses multiple linear regression to produce an improved forecast at specific locations by using model forecast variables and prior observations as predictors (Marzban et al., 2006; Cheng and Steenburgh, 2007). MOS remains a useful tool and, during the 2002 Winter Olympic Games, MM5-based MOS outperformed the native forecasts produced by MM5 and was equally or more skillful than human-generated forecasts by the Olympic Forecast Team (Hart et al., 2003). Glahn (2014) used MOS with a decay factor to predict temperature and dewpoint, and showed how different values of the decay factor affect MOS temperature and dewpoint forecasts (Glahn, 2014).

Although machine learning and statistics both draw conclusions from the data, they belong to two different modeling cultures. Statistics assumes that the data are generated by a given stochastic data model. Statistical methods have few parameters, and the values of the parameters are estimated from the data. Machine learning uses algorithmic models and treats the data mechanism as unknown. The approach of machine learning is to find an algorithm that can be fitted to the data, and it has lots of parameters (Breiman, 2001b). Machine learning has developed rapidly in fields outside statistics. It can be used both on large complex datasets and as a more accurate and informative alternative to data modeling on smaller datasets (Mirkin, 2011). Machine learning is becoming increasingly more important to the development of science and technology (Mjolsness and Decoste, 2001). To apply machine learning to practical problems, one of the most important things is to apply feature engineering and data structures (Domingos, 2012). The quality of feature engineering directly affects the final result. For some practical problems with special data structures, targeted feature engineering is required.

Since weather forecasts depend highly on data information and technology, how to make better use of machine learning and big-data technology to improve weather forecasts has become a research hotspot. Machine learning has been used in meteorology for decades (Haupt et al., 2009; Lakshmanan et al., 2015; Haupt and Kosovic, 2016; Cabos et al., 2017). For instance, the neural network technique was applied to the inversion of a multiple scattering model to estimate snow parameters from passive microwave measurements (Tsang et al., 1992). Schiller and Doerffer (1999) used a neural network technique for inverting a radiative transfer forward model to estimate the concentration of phytoplankton pigment from Medium Resolution Imaging Spectrometer data (Schiller and Doerffer, 1999). Chattopadhyay et al. (2013) put forward a nonlinear clustering technique to identify the structures of the Madden–Julian Oscillation (Chattopadhyay et al., 2013). Woo and Wong (2017) applied optical flow techniques to radar-based rainfall forecasting (Woo and Wong, 2017). Weather consultation data are unique, and mainly include NWP model data and observational data. They have different data structures and features, which makes feature engineering a complicated task. On the one hand, the observational data are real, but they only comprise the historical data of weather stations. If relying solely on observational data for prediction, only short-term weather can be predicted. On the other hand, model data reflect the average of a region, and can therefore help to make long-term predictions. Furthermore, model data have time series and a spatial structure that contain abundant intrinsic information on the problem to be solved.

Therefore, the key to using machine learning algorithms to solve the problems of weather forecasts is to apply feature engineering, such that the structure of observational and model data can be fully taken into account. This is a difficult but meaningful topic.

The ever-increasing demand for weather forecasts has made the accuracy of grid weather forecasts more and more important. However, most post-processing methods, such as MOS, can only consider the correction of one spatial point, without considering the spatial and temporal structure of the grid. Currently, one challenging and important area of research is finding a solution to apply post-processing methods to large volumes of gridded data across large dimensions in space and time. The feature engineering of constructing a data structure is a useful technology to achieve this goal.

In this paper, the model output machine learning (MOML) method is proposed for simulating weather consultation. MOML matches NWP forecasts against a long record of verify-

ing observations through a regression function that uses a machine learning algorithm. MOML constructs datasets by feature engineering based on spatial and temporal data, and it can make full use of the spatial and temporal structure of a point on the grid. In order to test the results of the application in practical problems, the MOML method is used to forecast 2-m grid temperature in the Beijing area. A variety of post-processing methods are used to calculate the 2-m temperature with different datasets in a 12-month period, including several machine learning algorithms, such as multiple linear regression and Random Forest, two training periods, and three datasets.

The paper is organized as follows: In section 2, the data and the problem concerned in this study are described. The MOML method is proposed in section 3. Section 4 compares the NWP forecasts with the numerical results from multiple linear regression, Random Forest, and MOS. Conclusions are drawn in section 5.

2. Data and problem

2.1. *Model data*

The six-hourly forecast data of the ECMWF model initialized at 0000 UTC up to a lead time of 360 hours from January 2012 to November 2016 are used in this paper. The model data are obtained for a grid of 5 × 6 points covering the Beijing area (39°–41°N, 115°–117.5°E) with a horizontal resolution of 0.5°, as well as the grid points on the edge of this area, and thus the model data on this 7 × 8 grid are used. Several predictors (e.g., land–sea mask) have the same value in the Beijing area and do not change with time. In addition to these unnecessary variables, 21 predictors are chosen, broadly based on meteorological intuition. Table 1 shows these predictors and their abbreviations.

These model data constitute a part of the original dataset D_0, and this part is denoted by $\boldsymbol{X}_0$. A record of meteorological elements on a certain day at a spatial point is called a sample S, and thus there are 1796 samples, i.e., S = 1, 2, …, 1796. Each sample has 61 six-hour time steps T_{Tem} with a forecast range of 0–360 hours (T_{Tem}= 0, 6, …, 360) and 21 predictors C, as listed in Table 1 ($C \in \{10\mathrm{U}, 10\mathrm{V}, \cdots, \mathrm{TP}\}$). The horizontal grid division is 7 × 8, and each spatial point of this region is denoted by (m, n), where m = 1, 2…, 7 and n = 1, 2, …, 8. Therefore, $\boldsymbol{X}_0$ consists of a 5D array, the size of which is 1796 × 61 × 21 × 7 × 8, and it can be written as

$$\boldsymbol{X}_0 = \left\{x_{m,n,S,T_{\mathrm{Tem}},C}\right\}_{m=1,2,\cdots,7,\ n=1,2,\cdots,8,S=1,2,\cdots,1796,\ T_{\mathrm{Tem}}=0,6,\cdots,360,\ C\in\{10\mathrm{U},10\mathrm{V},\cdots,\mathrm{TP}\}}. \quad (1)$$

2.2. *Observational data*

Data assimilation can determine the best possible atmospheric state using observations and short-range forecasts. The weather forecasts produced at the ECMWF use data assimilation and obtain the model analysis (zero-hour forecast) from meteorological observations. Therefore, for this study, the model analysis is used as the label, because not every grid point has an observation station. Furthermore, the model analysis data contain the observational information through data assimilation. The model analysis data used in this paper are the 2-m temperature of the ECMWF analysis in the Beijing area, with a horizontal resolution of 0.5° and recorded every 0000 UTC from 1 January 2012 to 15 December 2016.

Table 1. The predictors taken from the ECMWF model and their abbreviations.

Predictor	Abbreviation
10-m zonal wind component	10U
10-m meridional wind component	10V
2-m dewpoint temperature	2D
2-m temperature	2T
Convective available potential energy	CAPE
Maximum temperature at 2 m in the last 6 h	MX2T6
Mean sea level pressure	MSL
Minimum temperature at 2 m in the last 6 h	MN2T6
Skin temperature	SKT
Snow depth water equivalent	SD
Snowfall water equivalent	SF
Sunshine duration	SUND
Surface latent heat flux	SLHF
Surface net solar radiation	SSR
Surface net thermal radiation	STR
Surface pressure	SP
Surface sensible heat flux	SSHF
Top net thermal radiation	TTR
Total cloud cover	TCC
Total column water	TCW
Total precipitation	TP

These observational data constitute the other part of the original dataset D_0, and this part is denoted by $\boldsymbol{Y}_0$, $D_0 = (\boldsymbol{X}_0, \boldsymbol{Y}_0)$. Following the above notation, the samples S = 1, 2, …, 1796 are from January 2012 to November 2016, the predictor C is given the value 2T, 2T stands for 2-m temperature, the horizontal grid division is 5 × 6, and thus m = 2, 3, …, 6 and n = 2, 3, …, 7. Actually, the model analysis data include 1811 days, because, for a sample, the temperatures in the next 15 days are predicted by the model, and the corresponding true values need to be used. Let t be the forecast lead time, t = 24, 48, …, 360 hour, for a fixed (m, n) and S, the temperatures in the next t hours can be aggregated into vectors. $\boldsymbol{Y}_0$ can be written as

$$\boldsymbol{Y}_0 = \{y_{m,n,t,S}\}_{m=2,3,\cdots,6,\ n=2,3,\cdots,7,\ t=24,48,\cdots,360,S=1,2,\cdots,1796}, \quad (2)$$

where C = 2T was omitted. $\boldsymbol{Y}_0$ consists of a 4D array, of which the size is 1796 × 5 × 6 × 15.

2.3. *Problem*

For this study, the grid temperature forecast is actually a problem of using the predictions from the ECMWF model as

the input and obtaining the 2-m grid temperature forecasts as the output. Focusing on the samples from January to November 2016, for each sample, the 2-m grid temperature forecasts in the Beijing area at the forecast lead times of 1–15 days need to be forecast.

3. Methods

3.1. *Univariate linear running training period MOS*

Univariate linear MOS is one of the most important and widely used statistical post-processing methods (Glahn and Lowry, 1972; Marzban et al., 2006). The statistical method used by univariate linear MOS is unary linear regression; thus, only one predictor is used. The general unary linear regression equation of univariate linear MOS can be written as

$$y = w_0 + w_1 x_\mathrm{p} \,, \tag{3}$$

where y is the desired predicted value; x_p is the predictor, which is the NWP model output of this predicted value; w_0 is the intercept parameter, and w_1 is the slope parameter of the linear regression equation. Applying univariate linear running training period MOS to this problem, for a fixed (m, n, t) and S, the predicted value $y = y_{m,n,t,S}$, the predictor $x = x_{m,n,S,T_\mathrm{Tem}=t,C=2\mathrm{T}}$, and the parameters ($w_0$, w_1) have been estimated from a large amount of historical data. The running training period will be explained in section 3.3.1.

3.2. *Machine learning*

The meaning of machine learning in terms of weather forecasting can be understood in conjunction with the data mentioned above. In machine learning, a feature is an individual measurable property or characteristic of a phenomenon being observed, and a label is resulting information (Bishop, 2006). Machine learning obtains a model f by learning in the training set ($\boldsymbol{X}_\mathrm{train}$, $\boldsymbol{Y}_\mathrm{train}$). Then, for a test sample in the test set, its predicted value $y = f(\boldsymbol{x}), (\boldsymbol{x}, y) \in (\boldsymbol{X}_\mathrm{test}, \boldsymbol{Y}_\mathrm{test})$ can be obtained. For a sample S at fixed (m, n, t) in this problem, the features are $x_{m,n,S,T,C}$ or some combinations of them, and the label is $y_{m,n,t,S}$. The construction of the training and test set and the selection of features are the key to the problem, which will be explained in section 2.3.

Using the MOML method to solve the problem raised in section 2.3, the most important step is feature engineering. Feature engineering produces different datasets, and then a machine learning algorithm is used to process these datasets. As we all know, in order to obtain better results, the importance of feature engineering is far greater than that of the choice of machine learning algorithm. Therefore, this paper focuses on feature engineering in MOML (section 2.3), and two mature machine learning algorithms are used.

3.2.1. *Multiple linear regression*

Multiple linear regression attempts to model the relationship between two or more explanatory features and a response variable by fitting a linear equation to observed data. In this problem, a multiple linear regression model with d features $x_1, x_2, ..., x_d \in \boldsymbol{X}_\mathrm{train}$ and a label $y \in \boldsymbol{Y}_\mathrm{train}$, can be written as

$$f(x_1, x_2, ..., x_d) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b \,, \tag{4}$$

which can also be written in the vector form $f(\boldsymbol{x}) = \boldsymbol{w}^\mathrm{T}\boldsymbol{x} + b$. The aim of learning in the training set is to decide the coefficient $\boldsymbol{w} = (w_1, w_2, ..., w_d)^\mathrm{T}$ and b, so as to make $f(\boldsymbol{x})$ as close to y as possible (Alpaydin, 2014). The multiple linear model f can be used to predict the results $f(\boldsymbol{x})$ in the test set ($x \in \boldsymbol{X}_\mathrm{test}$).

Multiple linear model is a simple but powerful model to solve problems. The coefficient can intuitively express the importance of each independent normalized feature, which means the multiple linear model is an explanatory model.

3.2.2. *Random Forest*

A decision tree is a tree-like structure in which each internal node represents a test on an attribute, each branch represents the output of the test, and each leaf node represents a class label (Alpaydin, 2014). Decision trees can be classified into classification trees and regression trees. This problem is a regression problem, and thus a regression tree is used. A regression tree generation algorithm that can be applied to this problem is depicted in Fig. 1.

This regression tree generation algorithm chooses the optimal splitting features x_j and the optimal splitting point a_j, and solves

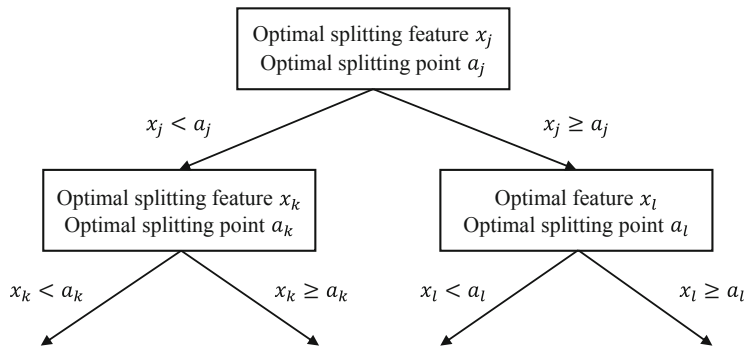

Fig. 1. Diagram of a regression tree generation algorithm, where x_j is the optimal splitting features and a_j is the optimal splitting point.

$$\min_{j,a_j}\left[\min_{c_1}\sum_{x\in R_1(j,a_j)}(y-c_1)^2+\min_{c_2}\sum_{x\in R_2(j,a_j)}(y-c_2)^2\right], \quad (5)$$

where $R_1(j,a_j)=\{\boldsymbol{x}|x_j<a_j\}, R_2(j,a_j)=\{\boldsymbol{x}|x_j\geqslant a_j\}$, and output the predicted values for each class $\hat{c}_m=\mathrm{E}[y|\boldsymbol{x}\in R_m]$, $m=1,2$. For R_1 and R_2, repeat the above steps, finally generating the regression tree

$$f(\boldsymbol{x})=\sum_m \hat{c}_m I(\boldsymbol{x}\in R_m)\,. \quad (6)$$

In this problem, a regression tree f is generated with the features $x_j\in \boldsymbol{X}_{\text{train}}$ and the label $y\in \boldsymbol{Y}_{\text{train}}$; thus, for $\boldsymbol{x}\in \boldsymbol{X}_{\text{test}}$, the predicted value is $f(\boldsymbol{x})$.

The bagging decision tree algorithm is an ensemble of decision trees trained in parallel, and the Random Forest algorithm is an extended version of the bagging decision tree algorithm, which introduces random attribute selection in the training process of the decision tree (Breiman, 2001a).

Actually, the dataset is divided into a training set and test set, and the training set is also randomly divided into a training subset and validation subset. The training subset and validation subset used in the Random Forest algorithm are generated by the random selection of bagging from the training set (Breiman, 2001a). The test set is used to test the results of the algorithm.

Random Forest has low computational cost and shows strong performance in many practical problems. The diversity of the base learners in Random Forest is not only from the sample bagging, but also from the feature bagging, which enables the generalization performance of the final ensemble algorithm to be further improved by the increase in the difference among the base learners.

3.3. *MOML method*

MOML is a machine learning-based post-processing method, which matches NWP forecasts against observations through a regression function, and improves the output of the ensemble forecast. The MOML regression function uses an existing machine learning algorithm. Setting the MOML regression function as f, the MOML regression equation is written as

$$y=f(\boldsymbol{x};\boldsymbol{w})=f(x_1,x_2,\cdots;w_0,w_1,w_2,\cdots)\,, \quad (7)$$

where $y\in \boldsymbol{Y}_{\text{train}}$ are the labels, $\boldsymbol{x}=(x_1,x_2,\cdots)\in \boldsymbol{X}_{\text{train}}$ are the features, and the parameters $\boldsymbol{w}=(w_0,w_1,w_2,\cdots)$ can be learned by the machine learning algorithm.

This MOML method involves performing two steps: feature engineering and machine learning.

First, by utilizing feature engineering (sections 3.3.1 and 3.3.2), the original dataset $D_0=(\boldsymbol{X}_0,\boldsymbol{Y}_0)$ needs to be processed into the training set ($\boldsymbol{X}_{\text{train}}$, $\boldsymbol{Y}_{\text{train}}$) and the test set ($\boldsymbol{X}_{\text{test}}$, $\boldsymbol{Y}_{\text{test}}$) to fit machine learning. The test set can be obtained by dividing the samples; when S = 1442, 1443…, 1796, the data are in the test set. The feature engineering focuses on two aspects: the training period and the dataset.

On the one hand, for training set selection, the samples of the original training set are S = 1, 2…, 1441, and for each sample S in the test set there are different ways to select the training period to construct the training set. For this study, the original training set can be improved to some training periods that are more suitable for this problem (example in section 3.3.1).

On the other hand, for a sample S at fixed (m, n, t) in this problem, the label $y=y_{m,n,t,S}$ and the features $\boldsymbol{x}\in \boldsymbol{X}_0$ can be divided into different forms according to the various ways of adding the time series and the spatial structure. In order to select the features $\boldsymbol{x}$ that are suitable for solving this problem, it is necessary to construct a suitable dataset that contains the spatial structure and then add an effective historical forecast dataset that contains the time series (example in section 3.3.2).

Second, by using the machine learning regression function f and the training data, the parameter $\hat{\theta}$ that minimizes the loss function Loss_f on the training dataset can be learned,

$$\hat{\theta}=\underset{\theta}{\operatorname{argmin}}\,\text{Loss}_f\left[f(\boldsymbol{x}_{\text{train}};\theta),y_{\text{train}}\right],\boldsymbol{x}_{\text{train}}\in \boldsymbol{X}_{\text{train}}\,,\quad y_{\text{train}}\in \boldsymbol{Y}_{\text{train}}\,, \quad (8)$$

where argmin stands for arguments of the minimum, and are the points of domain of some function at which the function values are minimized, and the machine learning regression function f and the parameter $\hat{\theta}$ are applied to calculate the predicted value y_{predict} on the test dataset and evaluate the method according to T_{RMSE},

$$y_{\text{predict}}=f(\boldsymbol{x}_{\text{test}};\hat{\theta}),\boldsymbol{x}_{\text{test}}\in \boldsymbol{X}_{\text{test}}\,, \quad (9)$$

$$\mathrm{T}_{\text{RMSE}}(f;\hat{\theta})=\left\|y_{\text{predict}}-y_{\text{test}}\right\|_2, y_{\text{test}}\in \boldsymbol{Y}_{\text{test}}\,. \quad (10)$$

The flow diagram of the MOML method is shown in Fig. 2. The MOML diagram uses multi-layer structures to represent historical time series. The extracted square of the diagram refers to selecting the required data from all the features to form predictors on the one hand, and on the other hand it represents the spatial structure around a single grid point. MOML can calibrate not only the forecasts of a single point, but also those of all points on the grid by constructing a suitable dataset. Feature engineering is the first and most important step of MOML, which includes the processing of both the training period and datasets. In terms of the training period, a year-round training period and running training period are considered. For datasets, temporal and spatial grid data are considered and combined into three datasets.

3.3.1. *Feature engineering: training period*

The data of 366 days (from 1 December 2015 to 30 November 2016) from those of 1827 days (from 1 January 2012 to 30 November 2016) from the ECMWF model are taken as the test set, i.e.,

$$(\boldsymbol{X}_{\text{test}},\boldsymbol{Y}_{\text{test}})=(\boldsymbol{X}_i,\boldsymbol{Y}_i)|_{S=1442,1443,\cdots,1796.},\ i=1,2,3\,. \quad (11)$$

The training period is a set of times throughout the original training set, and the training set in this problem concerns the following two types of training period.

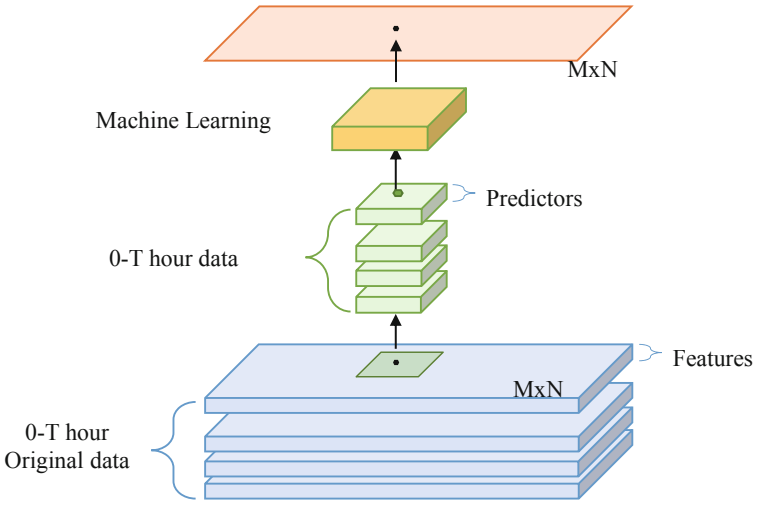

Fig. 2. Flow diagram of the MOML method. The blue cuboids are the original data in the Beijing area, and the green cuboids are the dataset with proper feature engineering. The yellow cuboid represents the process of machine learning, and the orange rectangle represents the output.

3.3.1.1. Year-round training period

One of the most natural ideas is for any month on the test set, all the previous data are taken as the training set. For example, when the temperatures from 1 February to 29 February 2016 need to be forecast, the training period is 1 January 2012 to 31 January 2016. This training period is named as the year-round training period. This training period is simple and suitable for most machine learning algorithms, but the training period is fixed for some samples.

3.3.1.2. Running training period

Some optimal training periods have been proposed in recent years. Wu et al. (2016) used the running training period scheme of MOS to forecast temperature (Wu et al., 2016). The idea of the running training period is that, for any day on the test set, the data for 35 days before the forecast time and 35 days before and after the forecast time for previous years are taken as a training set. For example, when the temperature of 1 May 2016 needs to be forecast, the training period is from 27 March to 30 April 2016 and from 27 March to 5 June 2012–15. This training period can be adjusted as the date changes.

3.3.2. *Feature engineering: dataset*

The original dataset $D_0 = (\boldsymbol{X}_0, \boldsymbol{Y}_0)$ is reconstituted into datasets 1–3 separately, $D_i = (\boldsymbol{X}_i, \boldsymbol{Y}_i)$, $i = 1, 2, 3$. The labels of datasets 1 and 2 are the same as that of the original data set, i.e.,

$$\begin{aligned}\boldsymbol{Y}_1 &= \boldsymbol{Y}_2 = \boldsymbol{Y}_0 \\ &= \{y_{m,n,t,S}\}_{m=2,3,\cdots,6,\ n=2,3,\cdots,7,\ t=24,48,\cdots,360,S=1,2,\cdots,1796}\,.\end{aligned} \tag{12}$$

The features are different according to diverse ways of dealing with the spatial structure.

3.3.2.1. Dataset 1

For the label $y_{m,n,t,S}$ of the MOML regression equation, Eq. (4), at a fixed spatial point (m, n) and a fixed forecast time tth hour $(m = 2,3,\cdots,6,\ n = 2,3,\cdots,7, t = 24,48,\cdots,360)$, the features $\boldsymbol{x}_{m,n,t,S}$ of dataset 1 take the predicted data of the last 66 hours from t, every 6 hours, which is denoted as

$$\boldsymbol{x}_{m,n,t,S} = \left\{x_{m,n,t,S,T_{\rm Tem},C}\right\}_{T_{\rm Tem}=t-66,t-60,\cdots,t,\ C\in\{10{\rm U},10{\rm V},\cdots,{\rm TP}\}}, \tag{13}$$

if $t \leqslant 60$, $T_{\rm Tem} = 0, 6\ldots, t$. Then, reshape the array $x_{m,n,t,S,T_{\rm Tem},C}$ into $x_{m,n,t,S,C_{T_{\rm Tem}}}$, and thus the number of features of $\boldsymbol{x}_{m,n,t,S}$ is the number of elements in $C_{T_{\rm Tem}}$, which is $21 \times 12 = 252$,

$$\boldsymbol{x}_{m,n,t,S} = \left\{x_{m,n,t,S,C_{T_{\rm Tem}}}\right\}_{C\in\{10{\rm U},10{\rm V},\cdots,{\rm TP}\},\ T_{\rm Tem}=t-66,t-60,\cdots,t}, \tag{14}$$

$$\boldsymbol{X}_1 = \{\boldsymbol{x}_{m,n,t,S}\}_{m=2,3,\cdots,6,\ n=2,3,\cdots,7,\ t=24,48,\cdots,360,S=1,2,\cdots,1796}, \tag{15}$$

$$\boldsymbol{x}_{m,n,t,S} = \left\{x_{i,j,t,S,T_{\rm Tem},C}\right\}_{i=m-1,m,m+1,\ j=n-1,n,n+1,T_{\rm Tem}=t-66,t-60,\cdots,t,C\in\{10{\rm U},10{\rm V},\cdots,{\rm TP}\}}, \tag{16}$$

if $t \leqslant 60$, $T_{\rm Tem} = 0, 6\ldots, t$. Then, reshape the array $x_{i,j,t,S,T_{\rm Tem},C}$ into $x_{m,n,t,S,C_{i,j,T_{\rm Tem}}}$, and thus the number of features of $\boldsymbol{x}_{m,n,t,S}$ is the number of elements in $C_{i,j,T_{\rm Tem}}$, which is $21 \times 12 \times 3 \times 3 = 2268$,

$$\boldsymbol{x}_{m,n,t,S} = \left\{x_{m,n,t,S,C_{i,j,T_{\rm Tem}}}\right\}_{C\in\{10{\rm U},10{\rm V},\cdots,{\rm TP}\},i=m-1,m,m+1,\ j=n-1,n,n+1,\ T_{\rm Tem}=t-66,t-60,\cdots,t}, \tag{17}$$

$$\boldsymbol{X}_2 = \{\boldsymbol{x}_{m,n,t,S}\}_{m=2,3,\cdots,6,\ n=2,3,\cdots,7,\ t=24,48,\cdots,360,S=1,2,\cdots,1796}, \tag{18}$$

and dataset 2 $D_2 = (\boldsymbol{X}_2, \boldsymbol{Y}_2)$. Figure 3b depicts dataset 2 diagrammatically.

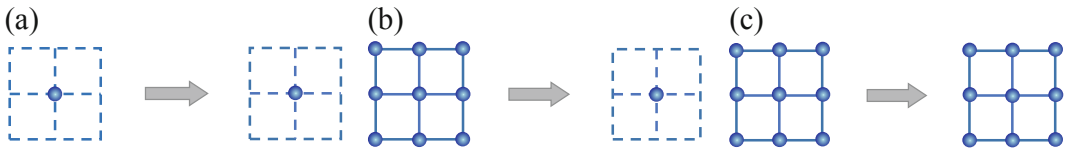

Fig. 3. Diagram of datasets 1–3. Dataset 1 focuses on the fixed spatial point, and dataset 2 adds the surrounding eight grid points. Dataset 3 takes all the 30 spatial points of the Beijing area into account in a unified way.

and dataset 1 $D_1 = (\boldsymbol{X}_1, \boldsymbol{Y}_1)$. Figure 3a depicts dataset 1 diagrammatically.

3.3.2.2. Dataset 2

For the label $y_{m,n,t,S}$ at a fixed spatial point (m, n) and a fixed forecast time tth hour ($m = 2,3,\cdots,6$, $n = 2,3,\cdots,7$, $t = 24,48,\cdots,360$), the features $\boldsymbol{x}_{m,n,t,S}$ of dataset 2, plus the surrounding eight grid points of the spatial point (m, n), are denoted as

3.3.2.3. Dataset 3

By using the idea of spatial interpolation, for a fixed spatial point (m, n), the longitude, latitude and altitude of the grid point are uniquely determined (m = 2, 3…, 6 and n = 2, 3…, 7). At a fixed forecast time tth hour ($t = 24,48,\cdots,360$) of a sample S, all the 30 spatial points of the Beijing area are taken into account separately, and thus there are 1796 × 30 samples S_L (S = 1, 2…, 1796 and L = 1, 2…, 30). The labels of dataset 3 are

$$\boldsymbol{Y}_3 = \left\{y_{t,S_L}\right\}_{t=24,48,\cdots,360, S=1,2,\cdots,1796,\ L=1,2,\cdots,30}. \tag{19}$$

Then, add the longitude, latitude and altitude of the 30 spatial points as the new predictors to the features $\boldsymbol{x}_{t,S_L}$ of dataset 3, which is denoted as

$$\boldsymbol{x}_{t,S_L} = \left\{x_{t,S_L,T,C}\right\}_{T_{\mathrm{Tem}}=t-66,t-60,\cdots,t,\ C\in\{10\mathrm{U},10\mathrm{V},L,\mathrm{TP},\mathrm{Lon},\mathrm{Lat},\mathrm{Alt}\}}, \tag{20}$$

if $t \leqslant 60$, T_{Tem}= 0, 6…, t. Then, reshape the array $\boldsymbol{x}_{t,S_L,T_{\mathrm{Tem}},C}$ into $\boldsymbol{x}_{t,S_L,C_{T_{\mathrm{Tem}}}}$, and thus the number of features of $\boldsymbol{x}_{t,S_L}$ is the number of elements in $C_{T_{\mathrm{Tem}}}$, which is (21 + 3) × 12 = 288,

$$\boldsymbol{x}_{t,S_L} = \left\{x_{t,S_L,T_{\mathrm{Tem}},C}\right\}_{C\in\{10\mathrm{U},10\mathrm{V},L,\mathrm{TP},\mathrm{Lon},\mathrm{Lat},\mathrm{Alt}\}\ T_{\mathrm{Tem}}=t-66,t-60,\cdots,t}, \tag{21}$$

if $t \leqslant 60$, T_{Tem} = 0, 6…, t. Also, we have

$$\boldsymbol{X}_3 = \left\{\boldsymbol{x}_{t,S_L}\right\}_{t=24,48,\cdots,360, S=1,2,\cdots,1796,\ L=1,2,\cdots,30}, \tag{22}$$

and dataset 3 $D_3 = (\boldsymbol{X}_3, \boldsymbol{Y}_3)$. Figure 3c depicts dataset 3 diagrammatically.

4. Results and discussion

In order to evaluate the forecast results of these methods, the root-mean-square error (RMSE) and temperature prediction accuracy are used to test the results of these algorithms.

The RMSE is one of the most common performance metrics for regression problems, and the RMSE of temperature is denoted by T_{RMSE},

$$T_{\mathrm{RMSE}}(f;D) = \left\{\frac{1}{K}\sum_{k=1}^{K}[f(x_k)-y_k]^2\right\}^{\frac{1}{2}}, \tag{23}$$

where f is the machine learning regression function, D is the dataset, K is the total number of samples of dataset D, x_k is the input, and y_k is the label.

The temperature forecast accuracy (denoted by F_a) in this study is defined as the percentage of absolute deviation of the temperature forecast not being greater than 2°C,

$$F_a = \frac{N_r}{N_f}\times 100\%, \tag{24}$$

where N_r is the number of samples in which the difference between the forecast temperature and the actual temperature does not exceed ±2°C and N_f is the total number of samples to be forecast.

In this section, the MOML method with the multiple linear regression algorithm ("lr") and Random Forest algorithm ("rf") is used to solve the problem of grid temperature forecasts, mentioned in section 2.3, and datasets 1–3 with two training periods, a year-round training period and running training period, are adopted. It is worth noting that the multiple linear regression algorithm is unsuitable for dataset 2 because it has too many features, and the running training period is unsuitable for Random Forest because of the heavy computation. The univariate linear MOS method is a linear regression method that uses only temperature data and does not require the datasets introduced in the last section. In fact, dataset 1 contains 21 features, dataset 2 contains 2268 features, and multiple linear regression is used on datasets 1 and 2 to obtain models lr_1 and lr_3. It can be considered that lr_1 and lr_3 are extensions of multi-feature MOS. The running training period of univariate linear MOS is an optimal training period scheme (Wu et al., 2016). Thus, univariate linear running training period MOS results in the running training period mos_r are used as a contrast. The methods used in the problem are listed in Table 2.

Table 2. List of methods used and their notation.

Method	Dataset	Training period	Notation
ECMWF	–	–	ECMWF
Univariate linear MOS	–	Running	mos_r
MOML (lr)	1	Year-round	lr_1_y
	3	Year-round	lr_3_y
	3	Running	lr_3_r
MOML (rf)	2	Year-round	rf_2_y
	3	Year-round	rf_3_y

4.1. *Whole-year comparison of the ECMWF model, univariate linear running training period MOS, and MOML*

In this subsection, a whole-year comparison of the ECMWF model, univariate linear running training period MOS, and MOML is presented. The results of MOML with the multiple linear regression algorithm, lr_3_r (Fig. 4), and Random Forest algorithm, rf_2_y, are reported.

Actually, the better the model data, the better the forecast results. The forecast accuracy is negatively correlated with the RMSE. Generally speaking, the lower the RMSE, the higher the forecast accuracy. The forecast ability of the model decreases linearly in a short time period, and nonlinearly in a long time period.

According to Fig. 4, all of the three methods (mos_r, lr_3_r and rf_2_y) can revise the 2-m temperature data of the ECMWF model quite well in the sense of the annual mean. The result of lr_3_r is better than that of mos_r when the forecast time is 1–9 days, which also explains why the multiple linear regression model is better than the univariate linear MOS model after extending the features with appropriate feature engineering. In particular, the forecast accuracy of the first day can reach more than 90%, which is 10% higher than that of the ECMWF model. The result of rf_2_y is better than that of mos_r in the whole forecast period, especially in the longer period. Because the temperature forecasting problem has strong linearity when the forecast period is short, multiple linear regression produces good results. However, the temperature forecasting problem has nonlinearity when the forecast period is longer, and thus some nonlinear algorithms, such as Random Forest, are more suitable for solving it. Accordingly, the numerical perform-

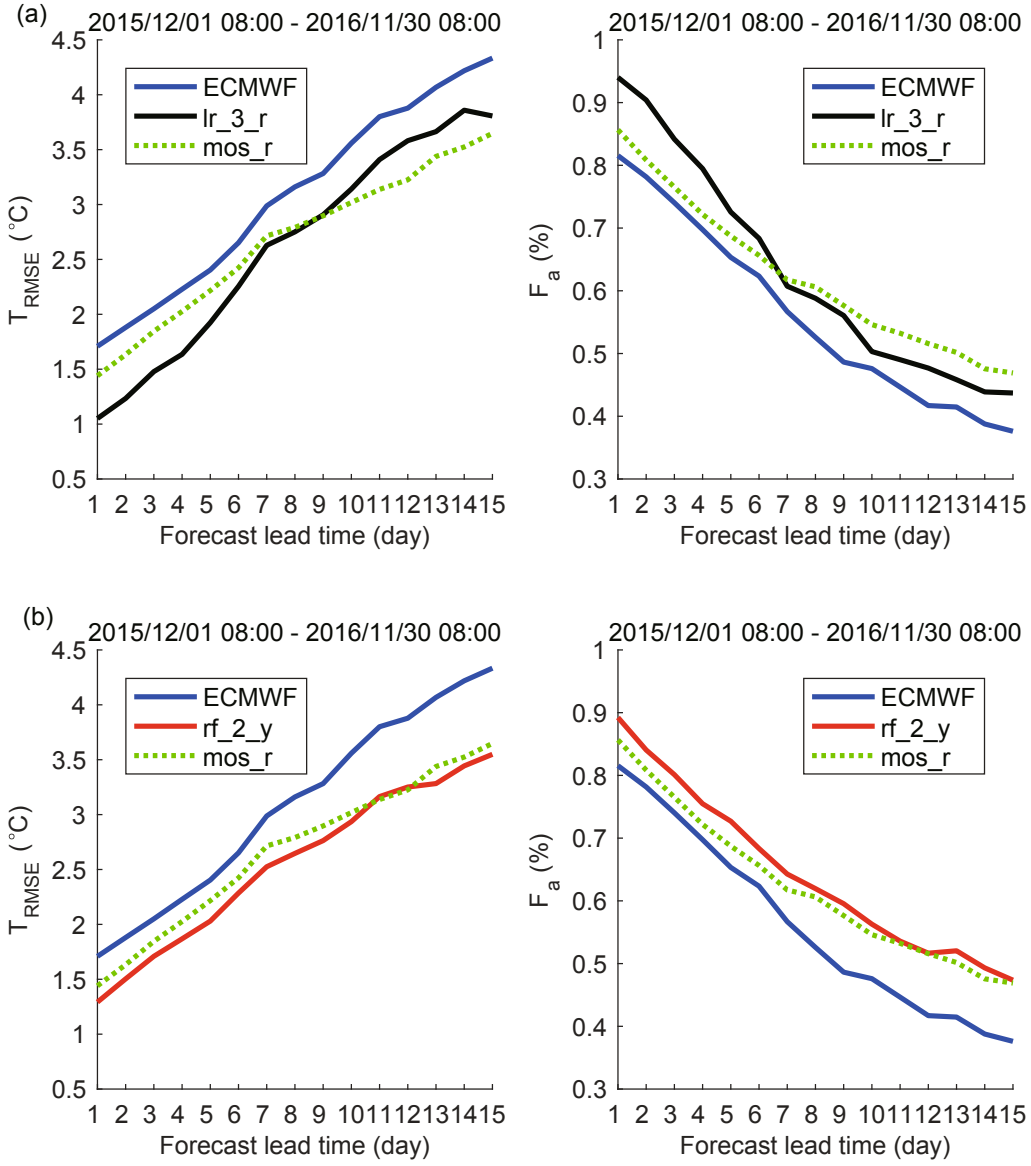

Fig. 4. Results of the lr_3_r, rf_2_y and mos_r models, using one-year temperature grid data in the Beijing area as the test set. Left: T_{RMSE} (RMSE; units: °C). Right: F_{a} (forecast accuracy; units: %). (a) shows lr_3_r has obvious advantages when the forecast time is 1–9 days, and (b) shows rf_2_y is superior to other models in the whole forecast period, especially in the longer period.

ance of each algorithm in the running and year-round training periods conform to this rule. Therefore, lr_3_r produces the best result, i.e., the highest accuracy and the smallest RMSE, when the forecast time is 1–6 days, while rf_2_y produces the best result when the forecast time is 7–15 days. Thus, a feasible solution (denoted by f_{MOML}) is presented for the grid temperature correction in the Beijing area, which involves using the lr_3_r method for days 1–6 of the forecast lead time and the rf_2_y method for days 7–15 (as shown in Fig. 5).

The average T_{RMSE} and F_{a} of the solution f_{MOML} and the ECMWF model (or univariate linear MOS) are calculated respectively, and these values are then used to evaluate the difference between the forecasting abilities of the two methods. In conclusion, the average T_{RMSE} and average F_{a} of the solution for the f_{MOML} method decreases by 0.605°C and increases by 9.61% compared with that for the ECMWF model, respectively, and by 0.189°C and 3.42% compared with that of the univariate linear running training period MOS, respectively.

4.2. *Month-by-month comparison of the three algorithms*

Considering that the change in temperature is seasonal within a year and fierce in some months in the Beijing area, a month-by-month comparison of the ECMWF model, univariate linear running training period MOS (mos_r) and MOML method (lr_3_r, rf_2_y, rf_3_y, lr_3_y, lr_1_y) is presented in this subsection.

4.2.1. *Winter months*

It is more important to improve the accuracy of temperat-

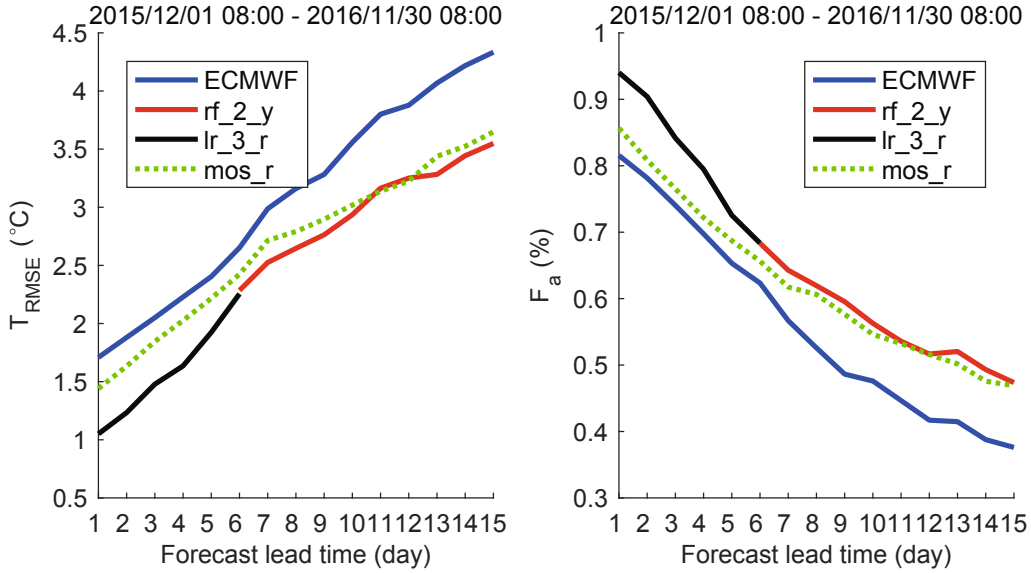

Fig. 5. A feasible solution f_{MOML} to the grid temperature correction in the Beijing area. f_{MOML} uses the lr_3_r method for days 1–6 of the forecast lead time and the rf_2_y method for days 7–15, and it has a lot of advantages in the whole forecast period.

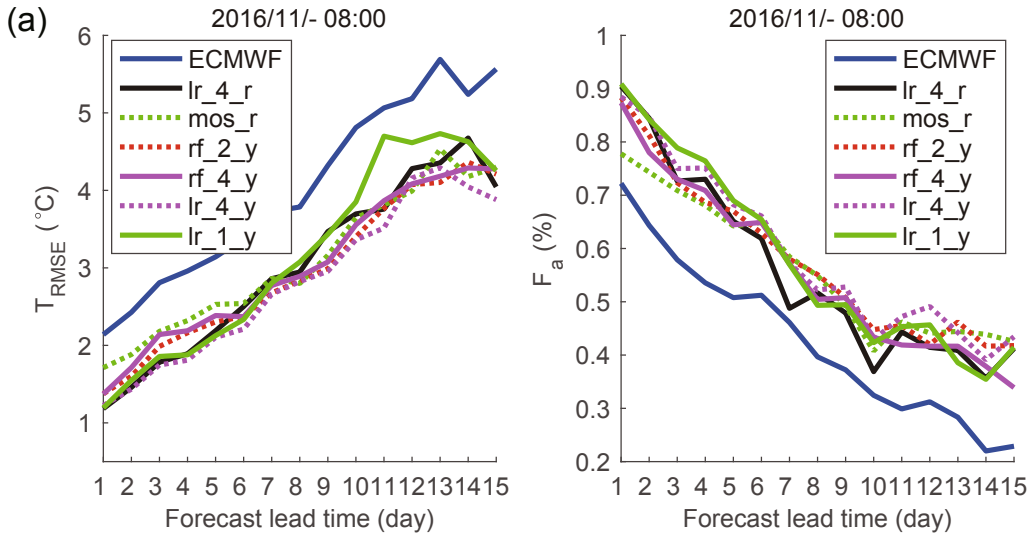

Fig. 6. Results of grid temperature forecasts in the Beijing area in November (a), December (b), January (c) and February (d). In these months, the forecast results of the ECMWF model do not work well, and the linear methods lr_3_r are better than other methods.

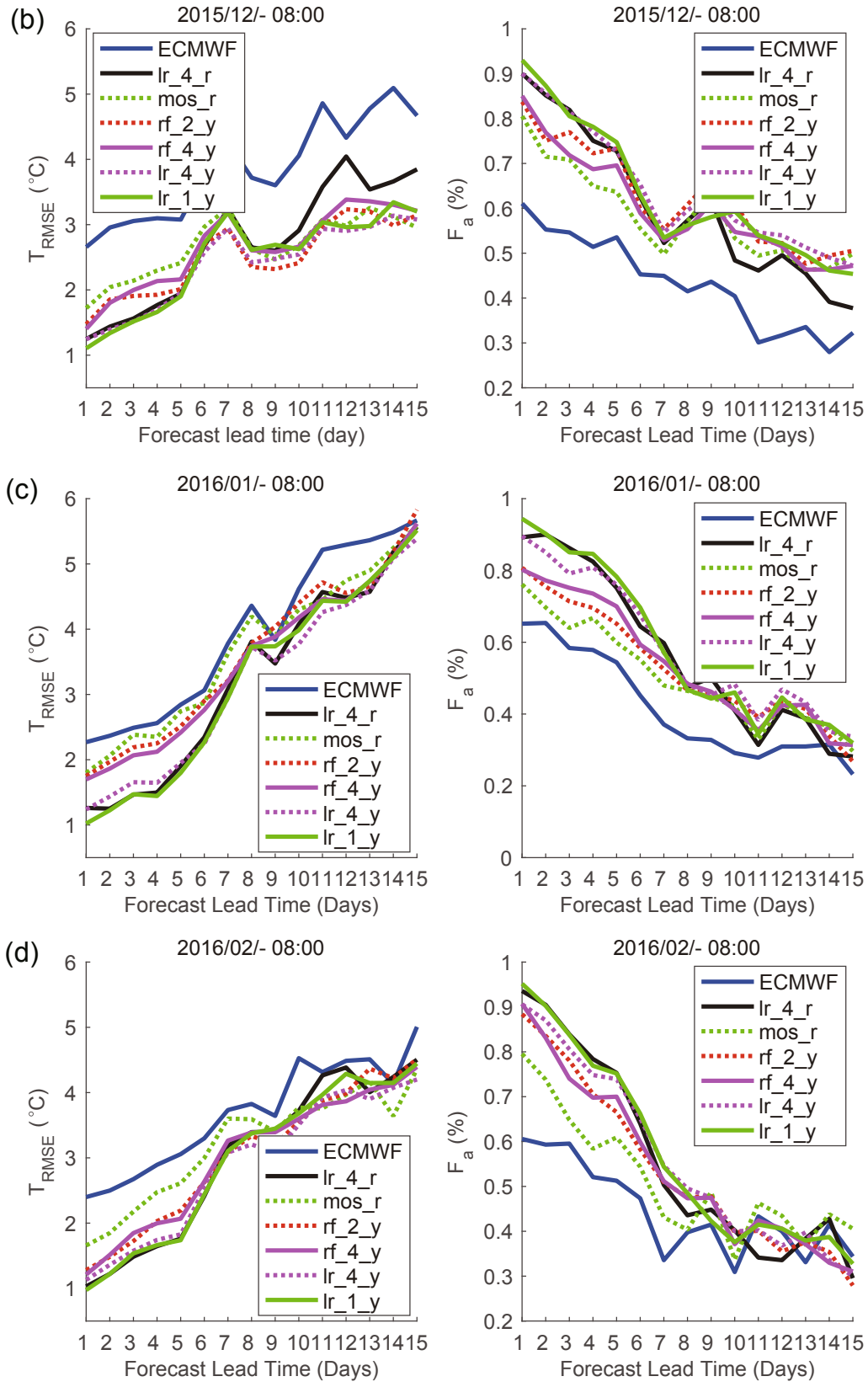

Fig. 6. (Continued)

ure forecasts in winter, because the forecast results of the ECMWF model do not work well in winter months. The forecast data in winter months are revised by the six methods listed in Table 2. Figure 6 shows the correction results of the grid temperature data in the Beijing area in November, December, January and February.

From December to February, the average T_{RMSE} and average F_a of the lr_3_r method decreases by 1.267°C and increases by 27.91%, respectively, compared with that of the ECMWF model, and by 0.652°C and 15.52% compared with that of the univariate linear running training period MOS, respectively. As shown in the figures, the forecast results of the ECMWF model do not work well in these four months, while the results of the MOML methods are all better than those of the ECMWF model. On the whole, in these months, the results of linear methods are better than other methods when the forecast lead time is relatively short, and the results of MOML with Random Forest are better when the forecast lead time is relatively long. The results of the running training period are better than those of the year-round training period when applying a linear method. On the whole, lr_3_r method is the best method in winter months.

4.2.2. *Other months*

Figure 7 shows the results of the grid temperature data in the Beijing area in March, June, July, August and October. In these five months, the forecast result of the ECMWF model are better than those in winter months, and the results of some MOML methods do not work better than the ECMWF model. On the whole, in these months, the results of MOML with the multiple linear regression algorithm are better than those of other methods in the first few days of the forecast period, and those with Random Forest are better than other methods when the forecast time is relatively long. Also, the results of the running training period are better than those of the year-round training period when applying a linear method.

Figure 8 shows the correction results of the grid temperature data in Beijing in April, May and September. The forecast results of the ECMWF model in these three months are better than those in the other months, and there is no need for revision in selected times of the forecast period. On the whole, in these three months, the results of MOML with the multiple linear regression algorithm are best in the first few days of the forecast period, and those with the Random Forest algorithm are bet-

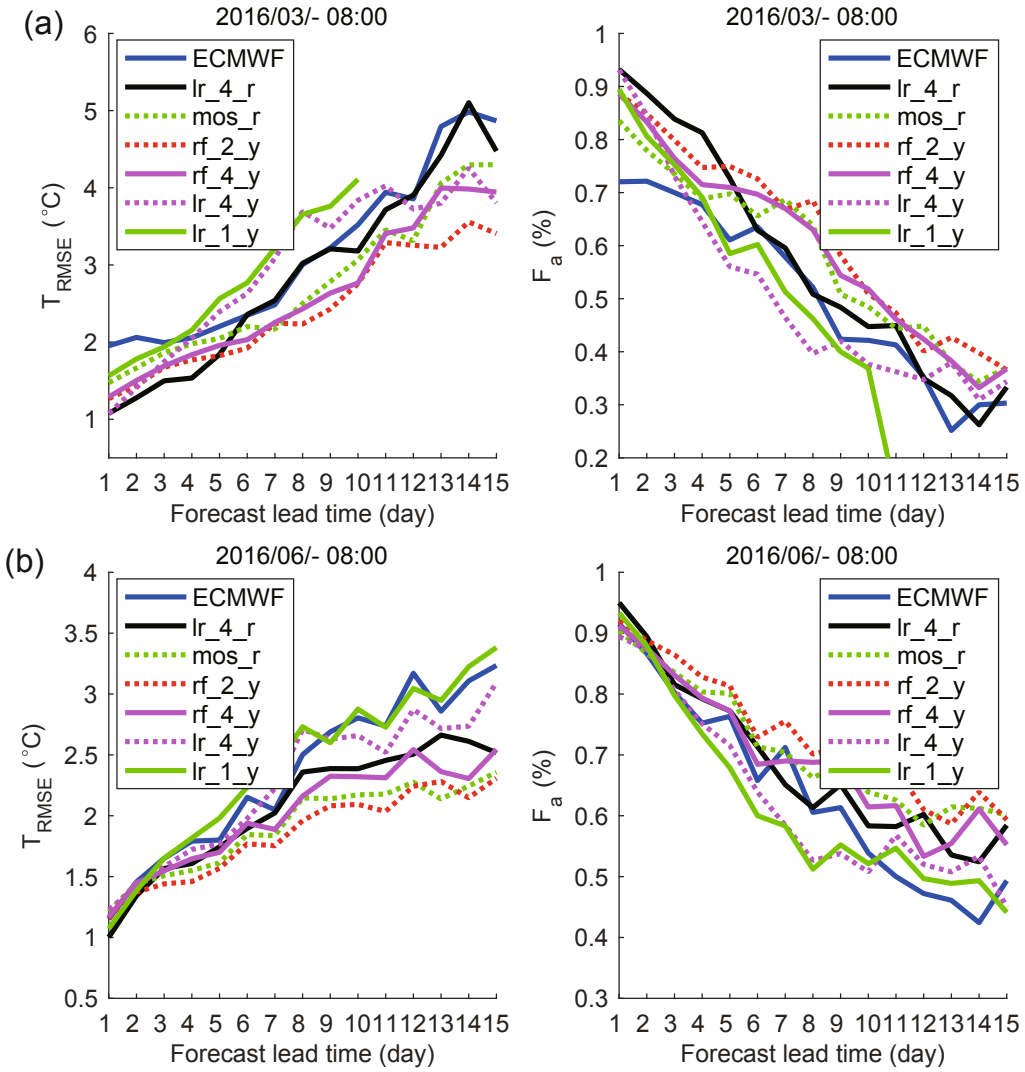

Fig. 7. Results of grid temperature forecasts in the Beijing area in March (a), June (b), July (c), August (d) and October (e). In these five months, the forecast results of the ECMWF model are better than those in winter months. The linear methods are better than other methods when the forecast lead time is short, and Random Forest algorithm are better when the forecast lead time is relatively long.

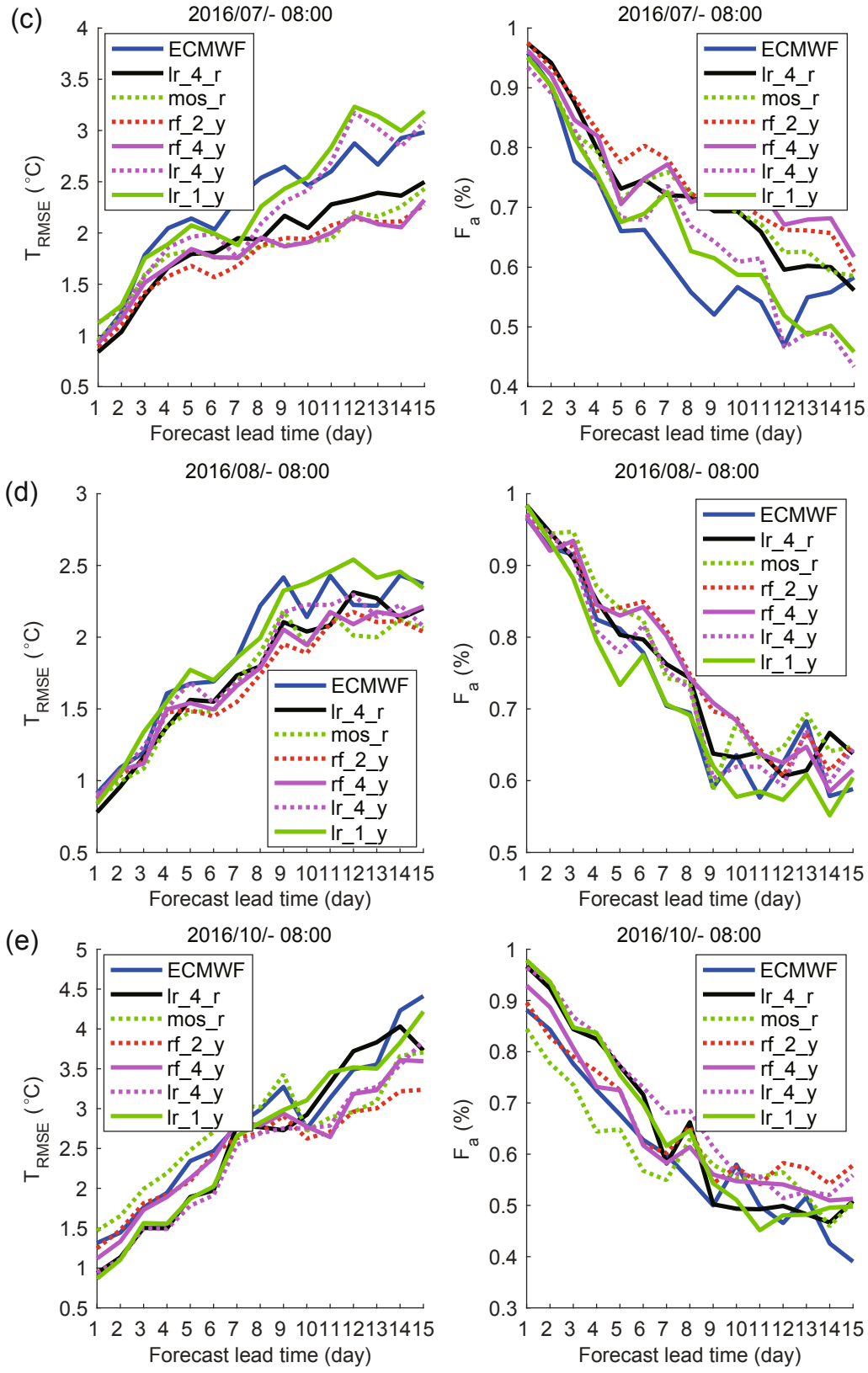

Fig. 7. (Continued)

ter than for other methods in the next few days. Also, the results of the running training period are close to those of the year-round training period when applying a linear method.

5. Conclusions

A model output machine learning method is proposed in

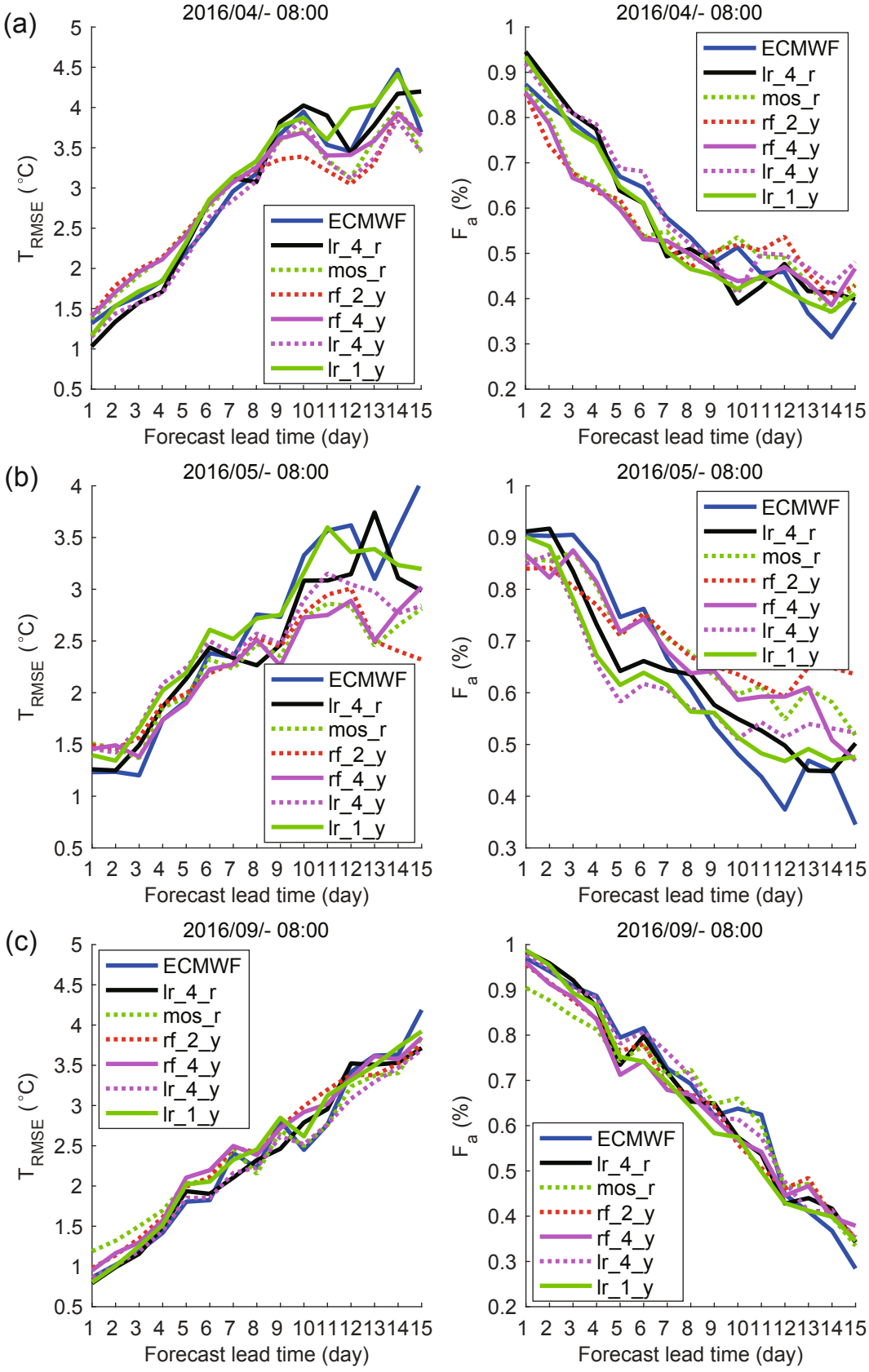

Fig. 8. Results of grid temperature forecasts in the Beijing area in April (a), May (b) and September (c). In these three months, the forecast results of the ECMWF model in these three months are better than those in the other months. The multiple linear regression algorithm is best in the first few days of the forecast period, and the Random Forest algorithm is better than for other methods in the next few days.

this paper and applied with multiple linear regression or the Random Forest algorithm to forecast grid temperature in the Beijing area in three datasets and two training periods. MOML takes advantage of machine learning algorithms, which has feature engineering for existing data structures, so that it can better fit the model and obtain more accurate results. The most important step of MOML is feature engineering. Feature engineering takes into account the temporal and spatial grids of the problem, as well as the training period, and processes the original dataset into different datasets. The selection of the machine learning algorithm should ensure that it can match the corresponding datasets. Selecting other machine learning algorithms may improve the results slightly, but this is not the focus of this paper. The main work of this paper is the feature engineering of MOML. In the sense of the annual mean, MOML with the multiple linear regression algorithm, dataset 3, and running training period produces the best result when the forecast time is 1–6 days, while MOML with the Random Forest, dataset 2, and a year-round training period produces the best result when the forecast time is 7–15 days. Generally, in each month, the results of MOML with the linear algorithm are better than others when the forecast time is relatively short, and the results of MOML with the nonlinear algorithm (Random Forest) are better when the forecast time is relatively long. Also, the results of the running training period are better than those of the year-round training period when applying the linear algorithm, and the results of datasets 2 and 3 are better than those of dataset 1. The numerical experiments show that the ECMWF model produces the worst temperature forecast results in winter. Thus, the temperature forecast in winter is urgently in need of revision. Fortunately, the correction effect of MOML for winter months is better than other months. Finally, a feasible solution of MOML is presented for grid temperature correction in the Beijing area.

In summary, the MOML method is better than the univariate linear running training period MOS method with a running training period, and has the ability to improve grid temperature forecast results in the Beijing area. In addition, as a post-processing method, MOML can be applied to the weather consultation process. This approach has good application prospects and can greatly reduce the manpower consumption during the consultation. In terms of its practical applicability, the machine learning algorithm in this paper adopts a mature, fast, multiple linear regression algorithm and the Random Forest algorithm, which do not need a lot of parameter adjustment and are easy to use. Constructing a more accurate and efficient machine learning algorithm to solve the problem will be our focus in future work.

Acknowledgments. The authors would like to express sincere gratitude to Lizhi WANG, Quande SUN and Xiaolei MEN for unpublished data; Zongyu FU and Yingxin ZHANG for professional guidance; and Zhongwei YAN and Fan FENG for critical suggestions. This work is supported by the National Key Research and Development Program of China (Grant Nos. 2018YFF0300104 and 2017YFC0209804) and the National Natural Science Foundation of China (Grant No. 11421101) and Beijing Academy of Artifical Intelligence (BAAI).

REFERENCES

Alessandrini, S., L. D. Monache, S. Sperati, and G. Cervone, 2015: An analog ensemble for short-term probabilistic solar power forecast. *Applied Energy*, **157**, 95–110, https://doi.org/10.1016/j.apenergy.2015.08.011.

Alpaydin, E., 2014: *Introduction to Machine Learning*. 3rd ed., The MIT Press, 640 pp.

Bishop, C. M., 2006: *Pattern Recognition and Machine Learning* (*Information Science and Statistics*). Springer-Verlag, 738 pp.

Bogoslovskiy, N. N., S. I. Erin, I. A. Borodina, L. I. Kizhner, and K. A. Alipova, 2016: Satellite data assimilation in global numerical weather prediction model using kalman filter. Proceedings of SPIE 10035, *22nd International Symposium Atmospheric and Ocean Optics*: *Atmospheric Physics*, Tomsk, Russian Federation, SPIE, 100356Z, https://doi.org/10.1117/12.2249275.

Breiman, L., 2001a: Random forests. *Machine Learning*, **45**(1), 5–32, https://doi.org/10.1023/A:1010933404324.

Breiman, L., 2001b: Statistical modeling: The two cultures. *Statistical Science*, **16**(3), 199–215.

Buehner, M., R. McTaggart-Cowan, and S. Heilliette, 2017: An ensemble Kalman filter for numerical weather prediction based on variational data assimilation: VarEnKF. *Mon. Wea. Rev.*, **145**(2),617–635,https://doi.org/10.1175/MWR-D-16-0106.1.

Cabos, R., P. Hecker, N. Kneuper, and J. Schiefele, 2017: Wind forecast uncertainty prediction using machine learning techniques on big weather data. *Proceedings of the 17th AIAA Aviation Technology, Integration, and Operations Conference*, Denver, Colorado, AIAA.

Cassola, F., and M. Burlando, 2012: Wind speed and wind energy forecast through Kalman filtering of numerical weather prediction model output. *Applied Energy*, **99**, 154–166, https://doi.org/10.1016/j.apenergy.2012.03.054.

Chattopadhyay, R., A. Vintzileos, and C. D. Zhang, 2013: A description of the Madden-Julian oscillation based on a self-organizing map. *J. Climate*, **26**(5), 1716–1732, https://doi.org/10.1175/JCLI-D-12-00123.1.

Cheng, W. Y. Y., and W. J. Steenburgh, 2007: Strengths and weaknesses of MOS, running-mean bias removal, and Kalman filter techniques for improving model forecasts over the western United States. *Wea. Forecasting*, **22**(6), 1304–1318, https://doi.org/10.1175/2007WAF2006084.1.

Delle Monache, L., T. Nipen, Y. B. Liu, G. Roux, and R. Stull, 2011: Kalman filter and analog schemes to postprocess numerical weather predictions. *Mon. Wea. Rev.*, **139**(11), 3554–3570, https://doi.org/10.1175/2011MWR3653.1.

Domingos, P., 2012: A few useful things to know about machine learning. *Communications of the ACM*, **55**, 78–87.

Glahn, B., 2014: Determining an optimal decay factor for bias-correcting MOS temperature and dewpoint forecasts. *Wea. Forecasting*, **29**(4), 1076–1090, https://doi.org/10.1175/WAF-D-13-00123.1.

Glahn, B., M. Peroutka, J. Wiedenfeld, J. Wagner, G. Zylstra, B. Schuknecht, and B. Jackson, 2009: MOS uncertainty estimates in an ensemble framework. *Mon. Wea. Rev.*, **137**(1), 246–268, https://doi.org/10.1175/2008MWR2569.1.

Glahn, H. R., and D. A. Lowry, 1972: The use of model output stat-

istics (MOS) in objective weather forecasting. *J. Appl. Meteor.*, **11**(8), 1203–1211, https://doi.org/10.1175/1520-0450(1972)011<1203:TUOMOS>2.0.CO;2.

Hart, K. A., W. J. Steenburgh, D. J. Onton, and A. J. Siffert, 2003: An evaluation of mesoscale-model-based model output statistics (MOS) during the 2002 Olympic and Paralympic winter games. *Wea. Forecasting*, **19**(2), 200–218, https://doi.org/10.1175/1520-0434(2004)019<0200:AEOMMO>2.0.CO;2.

Haupt, S. E., and B. Kosovic, 2016: Big data and machine learning for applied weather forecasts: Forecasting solar power for utility operations. *Proceedings of 2015 IEEE Symposium Series on Computational Intelligence*, Cape Town, South Africa, IEEE, 496–501, https://doi.org/10.1109/SSCI.2015.79.

Haupt, S. E., A. Pasini, and C. Marzban, 2009: *Artificial Intelligence Methods in the Environmental Sciences*. Springer, https://doi.org/10.1007/978-1-4020-9119-3.

Jacks, E., J. Brent Bower, V. J. Dagostaro, J. Paul Dallavalle, M. C. Erickson, and J. C. Su, 2009: New NGM-based MOS guidance for maximum/minimum temperature, probability of precipitation, cloud amount, and surface wind. *Wea. Forecasting*, **5**(5), 128–138, https://doi.org/10.1175/1520-0434(1990)005<0128:NNBMGF>2.0.CO;2.

Junk, C., L. Delle Monache, and S. Alessandrini, 2015: Analog-based ensemble model output statistics. *Mon. Wea. Rev.*, **143**(7), 2909–2917, https://doi.org/10.1175/MWR-D-15-0095.1.

Lakshmanan, V., E. Gilleland, A. McGovern, and M. Tingley, 2015: *Machine Learning and Data Mining Approaches to Climate Science*. Springer International Publishing, https://doi.org/10.1007/978-3-319-17220-0.

Marzban, C., S. Sandgathe, and E. Kalnay, 2006: MOS, perfect prog, and reanalysis. *Mon. Wea. Rev.*, **134**(2), 657–663, https://doi.org/10.1175/MWR3088.1.

Mass, C. F., D. Ovens, K. Westrick, and B. A. Colle, 2002: Does increasing horizontal resolution produce more skillful forecasts? *Bull. Amer. Meteor. Soc.*, **83**(3), 407–430, https://doi.org/10.1175/1520-0477(2002)083<0407:DIHRPM>2.3.CO;2.

Mirkin, B., 2011: Data analysis, mathematical statistics, machine learning, data mining: Similarities and differences. *Proceedings of 2011 International Conference on Advanced Computer Science and Information System*s, Jakarta, Indonesia, IEEE, 1–8.

Mjolsness, E., and D. Decoste, 2001: Machine learning for science: State of the art and future prospects. *Science*, **293**(5537), 2051–2055, https://doi.org/10.1126/science.293.5537.2051.

Molteni, F., R. Buizza, T. N. Palmer, and T. Petroliagis, 1996: The ECMWF ensemble prediction system: Methodology and validation. *Quart. J. Roy. Meteor. Soc.*, **122**(529), 73–119, https://doi.org/10.1002/qj.49712252905.

Monache, L. D., F. A. Eckel, D. L. Rife, B. Nagarajan, and K. Searight, 2013: Probabilistic weather prediction with an analog ensemble. *Mon. Wea. Rev.*, **141**(10), 3498–3516, https://doi.org/10.1175/MWR-D-12-00281.1.

Paegle, J., Q. Yang, and M. Wang, 1997: Predictability in limited area and global models. *Meteor. Atmos. Phys.*, **63**(1–2), 53–69, https://doi.org/10.1007/BF01025364.

Pelosi, A., H. Medina, J. Van Den Bergh, S. Vannitsem, and G. B. Chirico, 2017: Adaptive Kalman filtering for postprocessing ensemble numerical weather predictions. *Mon. Wea. Rev.*, **145**(12), 4837–4584, https://doi.org/10.1175/MWR-D-17-0084.1.

Peng, X. D., Y. Z. Che, and J. Chang, 2013: A novel approach to improve numerical weather prediction skills by using anomaly integration and historical data. *J. Geophys. Res.*, **118**(16), 8814–8826, https://doi.org/10.1002/jgrd.50682.

Peng, X. D., Y. Z. Che, and J. Chang, 2014: Observational calibration of numerical weather prediction with anomaly integration. *Proceedings of the EGU General Assembly Conference Abstracts*, Vienna, Austria, EGU.

Plenković, I. O., L. D. Monache, K. Horvath, M. Hrastinski, and A. Bajić, 2016: Probabilistic wind speed predictions with an analog ensemble. *Proceedings of the 6th EMS Annual Meeting & 11th European Conference on Applied Climatology*, Trst, Italija, ECAC.

Rudack, D. E., and J. E. Ghirardelli, 2010: A comparative verification of localized aviation model output statistics program (LAMP) and numerical weather prediction (NWP) model forecasts of ceiling height and visibility. *Wea. Forecasting*, **25**(4), 1161–1178, https://doi.org/10.1175/2010WAF2222383.1.

Schiller, H., and R. Doerffer, 1999: Neural network for emulation of an inverse model operational derivation of case II water properties from MERIS data. *Int. J. Remote Sens.*, **20**(9), 1735–1746, https://doi.org/10.1080/014311699212443.

Sperati, S., S. Alessandrini, and L. Delle Monache, 2017: Gridded probabilistic weather forecasts with an analog ensemble. *Quart. J. Roy. Meteor. Soc.*, **143**, 2874–2885, https://doi.org/10.1002/qj.3137.

Toth, Z., and E. Kalnay, 1997: Ensemble forecasting at NCEP and the breeding method. *Mon. Wea. Rev.*, **125**(12), 3297–3319, https://doi.org/10.1175/1520-0493(1997)125<3297:EFANAT>2.0.CO;2.

Tsang, L., Z. Chen, S. Oh, R. J. Marks, and A. T. C. Chang, 1992: Inversion of snow parameters from passive microwave remote sensing measurements by a neural network trained with a multiple scattering model. *IEEE Trans. Geosci. Remote Sens.*, **30**, 1015–1024, https://doi.org/10.1109/36.175336.

Veenhuis, B. A., 2013: Spread calibration of ensemble MOS forecasts. *Mon. Wea. Rev.*, **141**(7), 2467–2482, https://doi.org/10.1175/MWR-D-12-00191.1.

Wilks, D. S., and T. M. Hamill, 2007: Comparison of ensemble-MOS methods using gfs reforecasts. *Mon. Wea. Rev.*, **135**(6), 2379–2390, https://doi.org/10.1175/MWR3402.1.

Woo, W. C., and W. K. Wong, 2017: Operational application of optical flow techniques to radar-based rainfall nowcasting. *Atmosphere*, **8**(3), 48, https://doi.org/10.3390/atmos8030048.

Wu, J., H. Q. Pei, Y. Shi, J. Y. Zhang, and Q. H. Wang, 2007: The forecasting of surface air temperature using BP-MOS method based on the numerical forecasting results. *Scientia Meteorologica Sinica*, **27**(4), 430–435, https://doi.org/10.3969/j.issn.1009-0827.2007.04.012. (in Chinese)

Wu, Q., M. Han, H. Guo, and T. Su, 2016: The optimal training period scheme of MOS temperature forecast. *Journal of Applied Meteorological Science*, **27**(4), 426–434, https://doi.org/10.11898/1001-7313.20160405. (in Chinese)

Zhang, X. N., J. Cao, S. Y. Yang, and M. H. Qi, 2011: Multi-model compositive MOS method application of fine temperature forecast. *Journal of Yunnan University*, **33**(1), 67–71. (in Chinese)