Appl. Math. Mech. -Engl. Ed., **43**(2), 155–166 (2022)

APPLIED MATHEMATICS AND MECHANICS (ENGLISH EDITION)

https://doi.org/10.1007/s10483-022-2822-9

A station-data-based model residual machine learning method for fine-grained meteorological grid prediction*

Chuansai ZHOU[1], Haochen LI[2], Chen YU[1], Jiangjiang XIA[3], Pingwen ZHANG[1,†]

1. School of Mathematical Sciences, Peking University, Beijing 100871, China;
2. School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China;
3. Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China

(Received Oct. 20, 2021 / Revised Dec. 16, 2021)

Abstract Fine-grained weather forecasting data, i.e., the grid data with high-resolution, have attracted increasing attention in recent years, especially for some specific applications such as the Winter Olympic Games. Although European Centre for Medium-Range Weather Forecasts (ECMWF) provides grid prediction up to 240 hours, the coarse data are unable to meet high requirements of these major events. In this paper, we propose a method, called model residual machine learning (MRML), to generate grid prediction with high-resolution based on high-precision stations forecasting. MRML applies model output machine learning (MOML) for stations forecasting. Subsequently, MRML utilizes these forecasts to improve the quality of the grid data by fitting a machine learning (ML) model to the residuals. We demonstrate that MRML achieves high capability at diverse meteorological elements, specifically, temperature, relative humidity, and wind speed. In addition, MRML could be easily extended to other post-processing methods by invoking different techniques. In our experiments, MRML outperforms the traditional downscaling methods such as piecewise linear interpolation (PLI) on the testing data.

Key words machine learning (ML), post-processing, fine-grained weather forecasting, model residual machine learning (MRML)

Chinese Library Classification O29
2010 Mathematics Subject Classification 68T01, 68T05

1 Introduction

Weather forecasting closely relates to our daily life. As an interdisciplinary study, weather forecasting is a considerably complex problem due to the difficulty of observing large-scale

$*$ Citation: ZHOU, C. S., LI, H. C., YU, C., XIA, J. J., and ZHANG, P. W. A station-data-based model residual machine learning method for fine-grained meteorological grid prediction. *Applied Mathematics and Mechanics* (*English Edition*), **43**(2), 155–166 (2022) https://doi.org/10.1007/s10483-022-2822-9

† Corresponding author, E-mail: pzhang@pku.edu.cn

Project supported by the National Natural Science Foundation of China (Nos. 12101072 and 11421101), the National Key Research and Development Program of China (No. 2018YFF0300104), the Beijing Municipal Science and Technology Project (No. Z201100005820002), and the Open Research Fund of Shenzhen Research Institute of Big Data (No. 2019ORF01001)

atmosphere systems. In the past years, the primary methods for measurement including satellites, radars, and automatic meteorological stations (AMSs) have been proposed[1–4]. These traditional methods provide measurements of diverse meteorological elements such as rainfall, temperature, and infrared band. In addition, there is another form of data, called model data, provided by the numerical weather prediction (NWP) system. Model data are numerical predictions of multiple meteorological elements by solving the physical equations for the atmosphere with techniques such as data assimilation. The model data are given on the grid points. Every grid point represents a junction of longitude and latitude.

There are many kinds of model data provided by different organizations in the world. For instance, EC data given by European Centre for Medium-Range Weather Forecasts (ECMWF) are one of the commonly used global model data for reference by many countries. However, there are two limitations of the EC data. One is that the data are irrational under particular physico-geographic conditions and unexpected climatic changes. The other one is that the EC data with 0.125° resolution could not satisfy some specific requirements. For the former problem, the existing results on weather forecasting demonstrate that the post-processing methods are effective, but how to choose a suitable post-processing method is not straightforward. In this paper, we introduce some post-processing methods and apply one of them in our algorithm. For the latter problem, downscaling is one way of increasing the spatial resolution. Downscaling is a major problem in weather forecasting, especially in some specific applications[5–7]. There is a growing body of literature that applies downscaling methods such as piecewise linear interpolation (PLI) to generate high-resolution grid data. In addition, inverse distance weighted (IDW) interpolation is a commonly used method in downscaling[8]. We will introduce this interpolation method later. Recently, deep learning has been applied for downscaling. For example, the artificial neural network (ANN)[9–10] could learn fine representation from coarse data for downstream tasks. Although many downscaling methods have been proposed, more investigations are required to meet the high standard of specific applications, which is a main motivation of the current work.

Nowadays, machine learning (ML) reveals powerful capabilities and achieves notable successes in various fields[11–14]. Several attempts have been made to combine ML or deep learning with weather forecasting. Shi et al.[15] proposed encoding-forecasting structure embedded convolutional long short-term memory (ConvLSTM) for precipitation nowcasting, which means very short range prediction up to 8 hours. Rozas et al.[16] applied visual geometry group-16 layers (VGG-16) to derive total precipitation underlying NWP. This encoder-decoder network mainly learns the relationship among physical variables, especially geopotential height and total precipitation. Weyn et al.[17] used convolutional neural networks (CNNs) with up-sampling and down-sampling to predict 500 hPa geopotential height. They applied a common network to improve the prediction. However, the network only focused on a single meteorological element and was unable to improve the resolution of the input data. Casper et al.[18] designed the spatial-temporal (ST) MetNet, which could receive radar and satellite images instead of reanalysis data for precipitation prediction. MetNet could produce high-resolution grid prediction up to 8 hours into the future at the resolution of $1\,\mathrm{km}^2$. Our goal is to provide a medium range forecast up to 240 hours into the future.

In this paper, we aim to develop a new model, called model residual machine learning (MRML), to generate high-resolution medium range forecasts underlying high-precision stations prediction data. On account of lacking means to measure meteorological elements, observation from automatic weather stations always serves as the most reliable data. Our main idea is post-processing NWP to the maximum with the aid of the given accurate stations forecasting. Unlike the existing deep learning methods, we combine the station data to alleviate the limitations of NWP owing to particular physico-geographic conditions and unexpected climatic changes. Moreover, we use distance decay interpolation to generate high-resolution predictions in order to meet the requirements of the sponsor. Compared with the existing results, three main contributions are given in this paper. (i) MRML fits the residual between the model data and the observed data to generate high-resolution and high-precision grid prediction. (ii) MRML can be easily extended to other versions. For example, random forests and IDW in MRML could be replaced by other ML methods and interpolation methods, respectively. We only provide a way of combining the grid data and the observed data. (iii) Our experiments illustrate that MRML outperforms the traditional downscaling method at multiple meteorological elements, i.e., temperature, relative humidity, and wind speed.

The rest of the paper is organized as follows. In Section 2, mathematical notations of the data are given. We also introduce some relevant studies about ensemble learning and IDW. In Section 3, we describe the details of MRML for combining the model data and the observed data to generate high-resolution and high-precision grid prediction. In Section 4, we implement and test the algorithm on the EC data and the observed data at 226 automatic weather stations. The conclusions are then presented in Section 5.

2 Preliminaries

We describe the data used in our experiments and provide basic notations in Subsection 2.1. In Subsection 2.2, we briefly introduce ensemble learning that is an elementary topic in ML. IDW and model output machine learning (MOML) are discussed in Subsections 2.3 and 2.4, respectively. These methods are relevant to our scheme.

2.1 Data description

In this section, we describe our basic notations and briefly discuss two data sets that are used in our experiments. One is the EC data provided by the ECMWF, denoted by G. G is defined as a four-dimensional tensor with the dimensions of the number of time steps, latitude, longitude, and the number of meteorological elements. We use T to denote the number of time steps, m to denote the latitude, n to denote the longitude, and E to denote the number of meteorological elements, and hence, $G \in \mathbb{R}^{T\times m\times n\times E}$. G is a medium-range forecast up to 240 h into the future. The time dimension of G comprises $T = 53$ slices sampled over a 240 h interval. We are mainly interested in the weather in Beijing. The geographical area of Beijing corresponds to an $81 \times 81\,(m = n = 81)$ grid with a spatial resolution of $0.125°$. There are $E = 60$ meteorological elements (e.g., convective available potential energy, 10 meter U wind component, 2 meter temperature, and so forth) in our experiments (see Table A1 in Appendix A). Therefore, G represents an element of $\mathbb{R}^{53\times81\times81\times60}$. We pick the period from 15th January, 2015 to 30th April, 2017 of the EC data as the training set and the period from 12th January, 2018 to 31st December, 2018 of the EC data as the test set. Thus, 837 days and 354 days are used in the training set and the test set, respectively. We use a total of 1 191 days, i.e., 1 191 tensors, in our experiments. Each individual tensor is an element of $\mathbb{R}^{53\times81\times81\times60}$. Although we use the EC data in our experiments, MRML could be applied on any grid data. Throughout this paper, we use the EC data and the grid data interchangeably.

The other data set is the observed data of 3 meteorological elements including temperature, relative humidity, and wind speed, covering 226 automatic weather stations in Beijing. We let $p = 226$ denote the number of stations. Automatic weather stations are built to monitor the diurnal variation of meteorology. Each station hourly measures certain kinds of meteorological elements, called the observed data. An important part of MRML is forecasting at these stations, as discussed in Subsection 3.1. We focus on forecasting 3 meteorological elements, i.e., temperature, relative humidity, and wind speed. We use $S_{t,i} \in \mathbb{R}^3$ to denote the forecasts at the ith station, where t is the time step, $1 \leqslant t \leqslant T$, and $1 \leqslant i \leqslant p$. $\mathbb{R}^3$ represents the feature space of temperature, relative humidity, and wind speed. For combining the observed data and the EC data, we pick the same time period of the training set and the test set. Our goal is to generate high-resolution grid data. We use tensor $Y \in \mathbb{R}^{53\times(81+m')\times(81+n')\times3}$ to denote them, where m' and n' are integers. That is to say, the spatial resolution of our results is higher than that of the

EC data. Notice that the fourth dimension of tensor Y is 3. Since the high-resolution grid data Y are based on stations forecasts, we only provide Y of 3 meteorological elements including temperature, relative humidity, and wind speed. In Subsection 4.1, we discuss methods for evaluating results by using the observed data. Since we need to use a part of observed data for evaluation, we only use 172 stations for prediction in MRML, and take the others to evaluate our results. Figure 1 illustrates the structure of the grid data and the observed data.

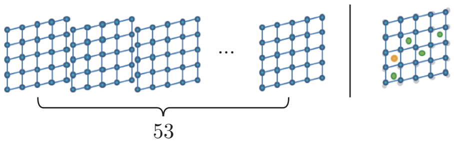


Fig. 1 Structure of the EC data and the observed data, where the left panel shows that the EC data provide 53 time steps. Each grid point in any timestamp represents a 60-dimensional vector, i.e., values of 60 meteorological elements. The right panel illustrates that the location of AMSs (green points and orange points) usually deviates from the grid. 172 green points in the MRML algorithm and the rest 54 orange points are used for evaluation (color online)

2.2 Ensemble learning

In this section, we briefly discuss ensemble learning[19–21], especially tree-based methods. There are a great variety of tree-based methods, whereas we only talk about random forests[20–21] and XGBoost[22]. Since the tedious technical details are beyond the scope of this paper, we only give a summary in this section. Ensemble learning plays an important role in ML. It combines a collection of simpler base models to construct more powerful prediction models. Most ensemble learning falls into one of the two categories: bagging or boosting. The difference between bagging and boosting seems like the difference between parallel operation and sequential operation. The random forests algorithm[20–21] is a substantial modification of bagging that builds a collection of trees. The idea in random forests is to improve the variance reduction of bagging by reducing the correlation between the trees. This is achieved in the tree-growing process through random selection of the input variables. The random forests algorithm has been demonstrated to give impressive improvements by combining together hundreds or even thousands of trees into a single procedure. We use the random forests algorithm to fit the observed data, as discussed in Subsection 3.2. Boosting works in a way of sequentially combining many base models produce a powerful committee. The main idea of gradient boosting is to use the gradient of the error to correct the mistake of the previous base model. XGBoost is a specific implementation of the gradient boosting method which gives impressive improvements by using the second-order derivative of the loss function, regularization, and parallel computing. MRML applies MOML[23] for stations forecasting with XGBoost as the main algorithm, as discussed in Subsections 2.4 and 3.1.

2.3 IDW interpolation

In this section, we then look at IDW interpolation, with the goal to downscale the MRML outputs to higher spatial resolution. MRML is a downscaling method that merges high-precision stations forecasts with grid data. As we described earlier in the introduction, plenty of downscaling methods have been proposed. We use IDW[8] to downscale the MRML outputs in our experiments. IDW[8] is appropriate for capturing the distance-decay relationship among data points. IDW estimates the value of an unknown point by using a weighted sum of the given values. The weights depend on the Euclidean distances of the given points from the unknown point. Suppose that the points $v_1, v_2, \cdots, v_n$ are given, where v_i represents an entity (e.g., a grid point). For each entity v_i, there is an associated value x_i (e.g., the temperature), $i = 1, 2, \cdots, n$. We aim for estimating the value x of another point v based on the given v_i. We use d_i to denote the Euclidean distance to v_i from v, and use w_i to denote the weight which represents levels of concern about the value x_i. To compute w_i with d_i, IDW uses the following formula:

$$w_i = \frac{1}{d_i^q + r}, \tag{1}$$

where r is a positive regularization parameter to avoid singularities at the locations of the data points and to control the smoothness of the interpolation, and q is a power parameter. Then, we can write x as follows:

$$x = \sum_{i=1}^{n} \frac{w_i}{\sum\limits_{j=1}^{n} w_j} x_i. \tag{2}$$

2.4 MOML

ML has achieved ever-increasing performance in the meteorological service field[9,23], especially weather forecasting. An important class of problems is forecasting at automatic weather stations, called stations forecasting. MOML[23] is a powerful post-processing method that applies XGBoost to stations forecasting. MOML uses grid data as features and provides an approach for performing feature selection. MRML relies on high-precision stations forecasting for improving the grid data quality. For this purpose, we use MOML to obtain $S_{t,1}, S_{t,2}, \cdots, S_{t,p}$ in our method, as discussed in Subsection 3.1.

3 MRML

In Section 3, we formalize our problem and introduce MRML. MRML is a post-processing method for generating high-resolution grid data on the basis of high-precision stations forecasting. There are three kinds of interpolations in MRML for different purposes. Figure 2 presents the architecture of MRML.

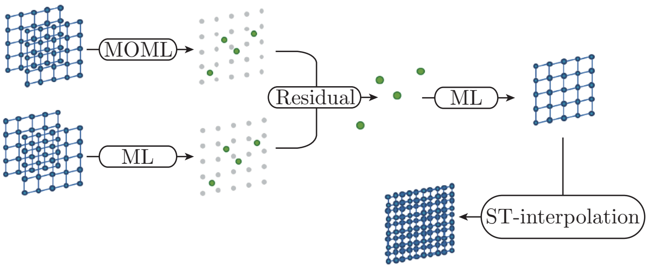


Fig. 2 Complete architecture of MRML, where green nodes represent station forecasts, MOML and ML are applied for stations forecasting, shadow grey grids bring out out-of-order distribution of AMSs, and ST-interpolation, i.e., IDW in our experiments, is used to generate high-resolution prediction (color online)

3.1 Stations forecasting

MOML is used to predict temperature, relative humidity, and wind speed at p stations. Forecast at each station is predicted separately. We describe the method at one station in detail, and the others are the same. Suppose that we want to predict $S_{t,i}$, the forecast on the ith station. MOML uses the grid data G as features and provides an approach for performing feature selection. Specifically, MOML first identifies the grid point in G that is the closest to the ith station, represented by $G^i_{t,j,k}$. Then, MOML identifies eight grid points that are the closest to $G^i_{t,j,k}$, i.e., $G^i_{t,j\pm\delta,k\pm\delta}$, where $\delta = 0, 1$. These nine grid points surrounding the ith station are included in spatial features. In addition, the grid data $G^i_{t-1,j\pm\delta,k\pm\delta}$, $G^i_{t-2,j\pm\delta,k\pm\delta}$, and $G^i_{t-3,j\pm\delta,k\pm\delta}$ ($\delta = 0, 1$), prior to the time step t, are included in temporal features. MOML uses the observed data at the ith station as labels, and applies XGBoost to learn. We use $\boldsymbol{x}_i$ to denote the ST features, $\boldsymbol{y}_i$ to denote labels, and θ_1 to denote parameters. We can formalize the problem by

$$\min_{\theta_1} \ \mathrm{Loss}(\boldsymbol{f}(\boldsymbol{x}_i;\theta_1),\boldsymbol{y}_1), \tag{3}$$

where $\boldsymbol{f}$ represents XGBoost. The specific formulation of the loss function could be seen in Ref. [22].

3.2 Station data fitting by using geographical features

MRML improves the low-resolution grid data quality by using the residual between two kinds of stations forecasting. One is produced by MOML, as discussed in Subsection 3.1. MOML provides the forecasts at stations by using local ST features. The random forests algorithm uses global geographical features. This process is similar to interpolation, while we add in geographical features. We assume that there is a relationship between meteorological elements and geographical features, i.e., latitude and longitude. We use $\mathscr{F}_1$, i.e., random forests, to denote the function of latitude and longitude. We use θ_2 to denote the parameters, h to denote the latitudes of all grid points, and l to denote the longitudes. We can formalize the problem by

$$\min_{\theta_2} \ \mathrm{Loss}(\mathscr{F}_1(l,h;\theta_2),G). \tag{4}$$

The specific formulation of the loss function could be seen in Ref. [21].

3.3 Residual extrapolation

Residual learning[24] is a superior technique in deep learning. This technique could make networks deeper and alleviate the problem of vanishing gradient. Unlike these purposes, we apply residual here to improve the low-resolution grid data quality. It is difficult to produce high-precision grid forecasting directly. Our main idea is fitting the residual between the original grid data and "true" grid data. In a real life situation in which the "true" grid data are unknown, we produce high-precision grid forecasting with the help of high-precision stations forecasting. As described in Subsections 3.1 and 3.2, the residual between two kinds of stations forecasting can be computed by using

$$\Delta_i = \mathscr{F}_1(l_i,h_i;\theta_2) - \boldsymbol{f}(\boldsymbol{x}_i;\theta_1), \quad 1 \leqslant i \leqslant p, \tag{5}$$

where Δ_i is the residual at the ith station, and l_i and h_i are the longitude and latitude of the ith station, respectively. Then, we use random forests to model the relationship between the residual and geographic features. Let $\mathscr{F}_2$ represent the function of longitude and latitude, and let θ_3 represent the parameters. We formalize this problem by

$$\min_{\theta_3} \ \mathrm{Loss}(\mathscr{F}_2(l_i,h_i;\theta_3),\Delta_i), \quad 1 \leqslant i \leqslant p. \tag{6}$$

Notice that Eq. (6) is similar to Eq. (4), but with different purposes. Equation (4) models the relationship between the geographical features and meteorological elements to estimate the values on stations. These values are used to compute the residual located in the stations. Equation (6) aims to generate the residual located in the grid points so that we could estimate the "true" grid data by

$$\widehat{Y} = \mathscr{F}_2(l,h;\theta_3) + G, \tag{7}$$

where $\widehat{Y}$ represents estimation of the "true" grid data, G is the grid data with the target meteorological elements, and l and h are the longitude and latitude of the grid data G, respectively. The resolution of $\widehat{Y}$ is the same as that of G at present. In Subsection 3.4, we will proceed to refine $\widehat{Y}$ to higher resolution.

3.4 Fine-grained grid forecasting

We use Eq. (7) to produce the high-precision grid forecast $\widehat{Y}$, which is an element of $\mathbb{R}^{T\times m\times n\times 3}$. Then, given $\widehat{Y}$ and $S_{t,1}, S_{t,2}, \cdots, S_{t,p}$, we directly apply IDW to produce the high-resolution $(T\times(m+m')\times(n+n')\times 3)$ grid forecasting Y. Let $V=\{\widehat{Y}\}\cup\{S_{t,i}, i=1,2,\cdots,p\}$, and use $\mathscr{F}_3$ to denote the IDW algorithm. Y could be computed by using

$$Y=\mathscr{F}_3(V). \tag{8}$$

Figure 2 illustrates the whole process of MRML. There are two pseudocodes, i.e., Algorithms 1 and 2 in the following. Algorithm 1 is the general version of MRML, and Algorithm 2 is the specific version of MRML used in our experiments.

```
Algorithm 1   MRML — general version
Require
    provided station predictions $S_{t,i}$, where $t=1,2,\cdots,T$ and $i=1,2,\cdots,p$,
    model data $G_{t,j,k}$, where $j=1,2,\cdots,m$ and $k=1,2,\cdots,n$,
    interpolations $\mathscr{F}_1, \mathscr{F}_2$, and $\mathscr{F}_3$.
Ensure
    for $t=1,2,\cdots,T$ do
        $S'_{t,i}=\mathscr{F}_1(G_{t,j,k})_i$, $\Delta_i=S'_{t,i}-S_{t,i}$,
        $Y_{\text{coarse}}=\mathscr{F}_2(\Delta_i)+G_{t,j,k}$, $Y_{\text{refined}}=\mathscr{F}_3(Y_{\text{coarse}}\cup S_{t,i})$,
    end for
    return fine-grained model data $Y_{\text{refined}}$.
Algorithm 2   MRML — applied in our experiment
Require
    station data $\boldsymbol{y}$,
    model data $G_{t,j,k}$, where $j=1,2,\cdots,m$ and $k=1,2,\cdots,n$,
    MOML $\boldsymbol{f}$, random forest $\mathscr{F}_1, \mathscr{F}_2$, and IDW $\mathscr{F}_3$.
Ensure
    for $t=1,2,\cdots,T$
        for $i=1,2,\cdots,p$ do
            train $\boldsymbol{f}(G,\boldsymbol{y};\theta)$,   $S_{t,i}=\boldsymbol{f}(G)$
        end for
        train $\mathscr{F}_1(l_{j,k},h_{j,k},G_{t,j,k};\theta)$,
        $S'_{t,i}=\mathscr{F}_1(l_i,h_i;\theta)$,   $\Delta_i=S'_{t,i}-S_{t,i}$,
        train $\mathscr{F}_2(l_i,h_i,\Delta_i;\theta)$,
        $Y_{\text{coarse}}=\mathscr{F}_2(\Delta_i)+G_{t,j,k}$,   $Y_{\text{refined}}=\mathscr{F}_3(Y_{\text{coarse}}\cup S_{t,i})$,
    end for
    return fine-grained model data $Y_{\text{refined}}$.
```

4 Numerical experiments

Before we experiment on our data, we have to define the criteria to judge the quality of our numerical results. In Subsection 4.1, we describe the methods for computing the accuracy and the root mean square error (RMSE) among temperature, relative humidity, and wind speed, which are commonly used in the field of meteorology. Then, we display our numerical results in Subsection 4.2.

4.1 Evaluation criteria

We introduce the quantitative evaluation criteria for the performance of MRML on the grid data in this section. It is difficult to measure the quality of the grid data. The main difficulty is due to the lack of the ground truth data. One of the most commonly used methods is utilizing the observed data as the ground truth data to evaluate the specific grid points.

As described in Subsection 2.1, we use the observed data at 54 stations as the ground truth data. Then, we find the grid points which are the closest to the 54 stations and evaluate the performance of these points. We use two evaluation metrics, i.e., the RMSE and accuracy, to measure the performance of our model. The approaches for quantifying the accuracy of diverse meteorological elements are different. We begin to introduce the formulas of accuracy for diverse meteorological elements in the following. These formulas are from the judgments of expert meteorologists in the cooperative institution.

The 3 meteorological elements, i.e., temperature, relative humidity, and wind speed, are quantitative variables which take on numerical values.

Case 1 Temperature

We split the prediction space into two regions, i.e., $\{X||X-X_0| \leqslant 2\}$ and $\{X||X-X_0| > 2\}$, where X_0 represents the observed data of temperature, and X represents the prediction of temperature. We think of the former case as a good prediction and the latter case as a bad prediction. We can treat the temperature as a binary variable by splitting the prediction space into these two regions. Let A_t denote the accuracy of the temperature. To compute A_t, we use the following formula:

$$A_t = \frac{A_1 + A_2}{A_1 + A_2 + A_3 + A_4} \times 100\%, \tag{9}$$

where A_1, A_2, A_3, and A_4 denote the number of true positive, true negative, false positive, and false negative values, respectively.

Case 2 Relative humidity

We split the prediction space into two regions, i.e., $\{X|\frac{|X-X_0|}{X_0} \leqslant 10\%\}$ and $\{X|\frac{|X-X_0|}{X_0} > 10\%\}$, where X_0 represents the observed data of relative humidity, and X represents the prediction of relative humidity. Likewise, we can obtain the four measures and compute the accuracy by Eq. (9).

Case 3 Wind speed

We assign the observation of wind speed[24] to different classifications by criteria (see Table 1).

Table 1 Wind speed levels

Wind speed/$(\mathrm{m \cdot s^{-1}})$	Level	Wind speed/$(\mathrm{m \cdot s^{-1}})$	Level
0.0–0.2	1	13.9–17.1	8
0.3–1.5	2	17.2–20.7	9
1.6–3.3	3	20.8–24.4	10
3.4–5.4	4	24.5–28.4	11
5.5–7.9	5	28.5–32.6	12
8.0–10.7	6	32.7–36.9	13
10.8–13.8	7	$\geqslant 37.0$	14

We use C_{w} to denote the score of a prediction and keep the score by

$$C_{\mathrm{w}} = \begin{cases} 1, & |L_{\mathrm{p}} - L_{\mathrm{t}}| = 0, \\ 0.6, & |L_{\mathrm{p}} - L_{\mathrm{t}}| = 1, \\ 0.4, & |L_{\mathrm{p}} - L_{\mathrm{t}}| = 2, \\ 0, & \text{others}, \end{cases} \tag{10}$$

where L_{p} and L_{t} are the prediction and ture values of the wind speed level, respectively. Let A_{w} denote the accuracy of wind speed, which is the average of score and

$$A_{\mathrm{w}} = \frac{\sum C_{\mathrm{w}}}{N_f} \times 100\%, \tag{11}$$

where $\sum C_{\mathrm{w}}$ represents the accumulation of score, and N_f is the total number of the samples.

4.2 Numerical results

Figures 3(a)–3(c) and 3(d)–3(f) show the accuracy and RMSE as a function of time steps (53 time steps), respectively. Blue solid curves represent the performance of MRML, while orange solid curves represent the performance of the original EC data. We evaluate the coarse-grained EC data to show that the post-processing methods outperform the original data. Green and red solid curves represent the performance of PLI[13] and IDW, respectively. PLI has been accepted in many meteorological centers for post-processing. Hence, we use PLI as a baseline. In addition, we display the performance of IDW to compare with MRML. We separate 54 stations from 226 stations for evaluation. We improve the spatial resolution of approximately 1 km (514 × 563 grid points).

For all cases, the accuracy decreases, while the RMSE increases as the time steps increase. For convenience, we only analyze the accuracy because the illustration of the RMSE comes to the same conclusion. In Fig. 3(a), the accuracy of temperature is higher than that of the baseline PLI at all time steps. PLI and IDW slightly improve the quality of the original EC data. In Figs. 3(b) and 3(e), the red solid curve nearly overlaps the orange and green solid curves, while the blue solid curve outperforms them. In Figs. 3(c) and 3(f), PLI still slightly improves the quality of the original EC data, and IDW slightly outperforms PLI. The accuracy of wind speed is over 80% at most time steps. Figure 3 illustrates that MRML outperforms the baseline in all three meteorological elements. It also demonstrates that stations forecasting could improve the quality of the grid data by comparing MRML with IDW.

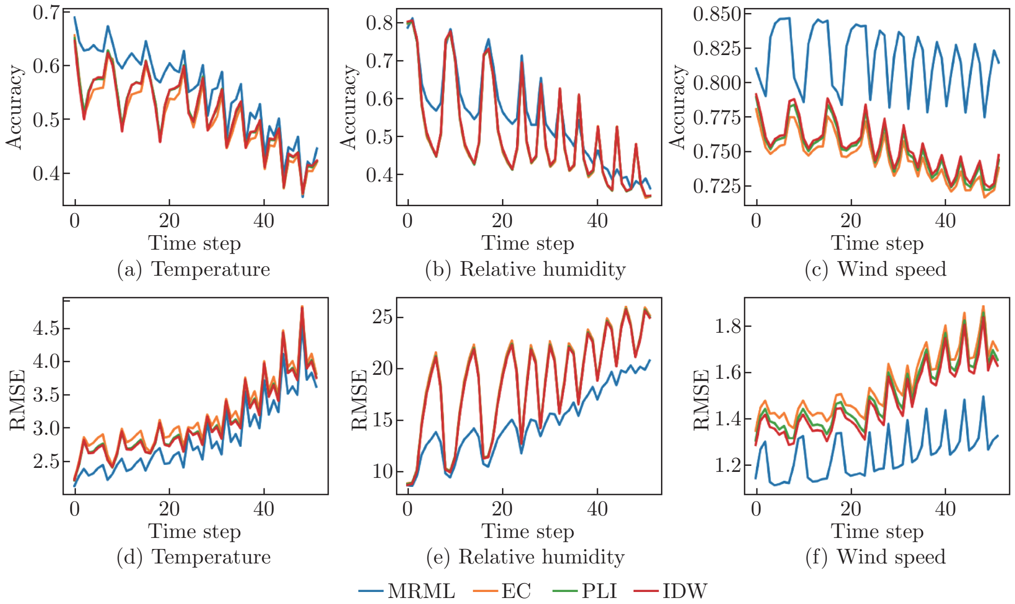

(a) Temperature
(b) Relative humidity
(c) Wind speed
(d) Temperature
(e) Relative humidity
(f) Wind speed

Fig. 3 Comparison of MRML prediction, PLI, IDW, and EC based on 54 stations. This figure displays the accuracy and RMSE versus time steps. Here, we get rid of the initial time step because it is meaningless in reality (color online)

The average RMSE of the first three days is given in Table 2, and the accuracy is given in Table 3. Tables 2 and 3 display that MRML substantially improves the quality of forecasts, while PLI and IDW provide marginal improvements. For example, the original EC data have an accuracy of wind speed of 75.88%. MRML has an accuracy of wind speed of 82.36%. PLI has an accuracy of wind speed of 76.49%, and IDW has an accuracy of wind speed of 76.74%. These four methods have the RMSE of temperature as 2.787 3, 2.437 9, 2.702 2, and

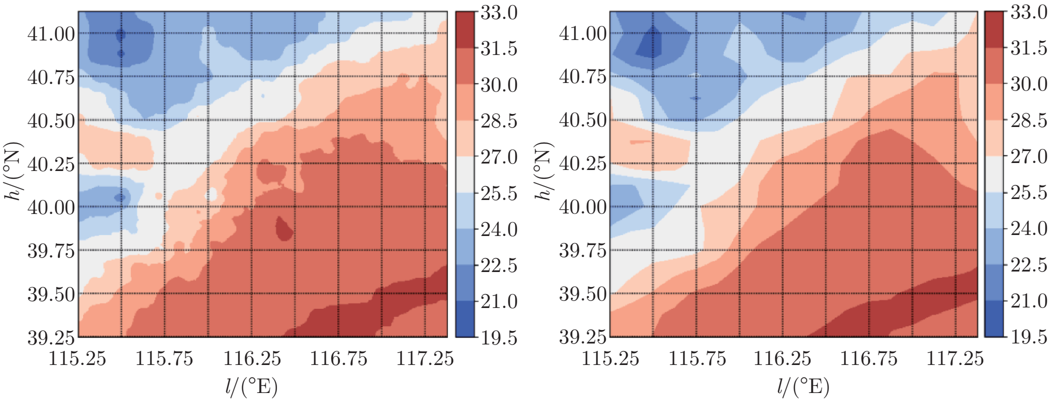
2.6970, respectively. These corroborate our previous conclusion. We visualize one prediction of temperature, which is shown in Fig. 4. Figures 4(a) and 4(b) represent the high-resolution results generated by MRML and the coarse-grained EC data, respectively.

Table 2 Average of RMSE in the first 3 days

Method	Relative humidity	Temperature	Wind speed
EC	16.1384	2.7873	1.4280
PLI	15.9366	2.7022	1.3848
IDW	15.8418	2.6970	1.3606
MRML	12.2118	2.4379	1.1900

Table 3 Average of accuracy in the first 3 days

Method	Relative humidity	Temperature	Wind speed
EC	57.71%	55.23%	75.88%
PLI	57.90%	56.16%	76.49%
IDW	58.18%	55.86%	76.74%
MRML	64.05%	61.75%	82.36%



(a) MRML

(b) EC

Fig. 4 Contour lines of fine-grained prediction (a) and EC data (b) in temperature, where MRML retains the overall arrangement of temperature provided by the EC data, and (a) depicts fine contour lines in boundary in different color intervals while (b) is short of the information in boundary (color online)

5 Conclusions

Since the existing grid data are unable to meet the high requirements of some specific applications such as the Winter Olympic Games, we focus on increasing the resolution of the grid data. Meanwhile, we intend to maintain the high quality of the grid data at some specific locations. In this paper, we propose a new method, called MRML, of high-resolution weather forecasting. Unlike most downscaling methods, coarse spatial resolution grid data merge with high-precision stations forecasting by applying ML with residual correction. Subsequently, MRML uses IDW to downscale the previous outputs. In summary, MRML generates high-resolution and high-precision grid predictions based on stations forecasting. The MRML algorithm could be easily modified to apply to other ML methods. However, the numerical results for these other types of MRML remain to be investigated. In future work, we will not only explore effective ML models, but also add reasonable information, e.g., time, to the geographical features for matching the characteristic of the weather.

The following results can be obtained. Numerical experiments demonstrate that the observed data could improve the quality of the grid data by applying the proposed algorithm.

MRML performs better ability than traditional downscaling methods such as PLI by merging with the observed data. In addition, MRML is a general method for diverse meteorological elements, which could be used in the same way. The numerical results show that MRML achieves good performance on temperature, relative humidity, and wind speed.

Acknowledgements The authors would like to express sincere gratitude to Lizhi WANG and Xiao LOU for unpublished data, Hanqiuzi WEN and Yingxin ZHANG for professional guidance, and Beijing Academy of Artificial Intelligence for finance support.

References

[1] NIES, H., BEHNER, F., REUTER, S., MECKEL, S., and LOFFELD, O. Radar imaging and tracking using geostationary communication satellite systems — a project description. *Proceedings of EUSAR* 2016: 11*th European Conference on Synthetic Aperture Radar*, IEEE, Piscataway, 1–4 (2016)

[2] LUOJUS, K. P., PULLIAINEN, J. T., METSAMAKI, S. J., and HALLIKAINEN, M. T. Snow-covered area estimation using satellite radar wide-swath images. *IEEE Transactions on Geoscience and Remote Sensing*, **45**(4), 978–989 (2007)

[3] SALTIKOFF, E., FRIEDRICH, K., SODERHOLM, J., LENGFELD, K., NELSON, B., BECKER, A., HOLLMANN, R., URBAN, B., HEISTERMANN, M., and TASSONE, C. An overview of using weather radar for climatological studies: successes, challenges, and potential. *Bulletin of the American Meteorological Society*, **100**(9), 1739–1752 (2021)

[4] JOSH, M. and TAHMEED, A. Evaluating meteorological data from weather stations, and from satellites and global models for a multi-site epidemiological study. *Environmental Research*, **165**, 91–109 (2018)

[5] BEONG, I. Y. A smoothening method for the piecewise linear interpolation. *Journal of Applied Mathematics*, **2015**, 376362 (2015)

[6] FOTHERINGHAM, S. and O'KELLY, M. *Spatial Interaction Models*: *Formulations and Applications*, 5th ed., Kluwer, Dordrecht, 62–84 (1989)

[7] LOPEZ, P., IMMERZEEL, W., RODRIGUEZ, S., STERK, G., and SCHELLEKENS, J. Spatial downscaling of satellite-based precipitation and its impact on discharge simulations in the magdalena river basin in Colombia. *Frontiers in Earth Science*, **6**, 68 (2018)

[8] PATRICK, M. B. and PETER, C. K. Multivariate interpolation to incorporate thematic surface data using inverse distance weighting (IDW). *Computers & Geosciences*, **22**(7), 795–799 (1996)

[9] NAM, D. H., KEIKO, U., and AKIRA, M. Downscaling global weather forecast outputs using ANN for flood prediction. *Journal of Applied Mathematics*, **2011**, 246286 (2011)

[10] SCHOOF, J. T. and PRYOR, S. C. Downscaling temperature and precipitation: a comparison of regression-based methods and artificial neural networks. *International Journal of Climatology*, **21**(7), 773–790 (2001)

[11] JACOB, D., CHANG, M. W., KENTON, L., and KRISTINA, T. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv*:1810.04805 (2018) http://arxiv.org/abs/1810.04805

[12] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G., and DEAN, J. Distributed representations of words and phrases and their compositionality. *Proceedings of the* 26*th International Conference on Neural Information Processing Systems*, Curran Associates Inc., New York, 3111–3119 (2013)

[13] MIKOLOV, T., CHEN, K., CORRADO, G., and DEAN, J. Efficient estimation of word representations in vector space. *International Conference on Learning Representations*, Scottsdale, Arizona, 2–4 (2013)

[14] YU, C., LI, H. C., XIA, J. J., WEN, H. Q. Z., and ZHANG, P. W. A data-driven random subfeature ensemble learning algorithm for weather forecasting. *Communications in Computational Physics*, **28**(4), 1305–1320 (2020)

[15] SHI, X., CHEN, Z., WANG, H., YEUNG, D. Y., WONG, W. K., and WOO, W. C. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. *Neural Information Processing Systems*, **9**, 802–810 (2015)

[16] ROZAS, L. P., RENZULLO, L. J., INZA, I., and LOZANO, J. A. A data-driven approach to precipitation parameterizations using convolutional encoder-decoder neural networks. *arXiv*: 1903.10274 (2019) https://arxiv.org/abs/1903.10274

[17] WEYN, J. A., DURRAN, D. R., and CARUANA, R. Can machines learn to predict weather? using deep learning to predict gridded 500 hPa geopotential height from historical weather data. *Journal of Advances in Modeling Earth Systems*, **11**, 2680–2693 (2019)

[18] CASPER, K. S., LASSE, E., JONATHAN, H., MOSTAFA, D., AVITAL, O., TIM, S., SHREYA, A., JASON, H., and NAL, K. A neural weather model for precipitation forecasting. *arXiv*: 2003.12140 (2020) https://arxiv.org/abs/2003.12140

[19] BREIMAN, L. Random forests. *Machine Learning*, **45**, 5–32 (2001)

[20] JAMES, G. *An Introduction to Statistical Learning*, 2nd ed., Springer, New York, 316–332 (2017)

[21] HASTIE, T. *The Elements of Statistical Learning*, 2nd ed., Springer, New York, 587–622 (2009)

[22] CHEN, T. Q. and CARLOS, G. XGBoost: a scalable tree boosting system. *Association for Computing Machinery*, **10**, 785–794 (2016)

[23] LI, H. C., YU, C., and XIA, J. J. A model output machine learning method for grid temperature forecasts in the Beijing area. *Advances in Atmospheric Sciences*, **36**, 1156–1170 (2019)

[24] HE, K. M., ZHANG, X., REN, S., and SUN, J. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Piscataway, 770–778 (2016)

Appendix A

Table A1 Illustration of meteorological elements in our data

Predictor name	Unit	Predictor name	Unit
100 meter U wind	m/s	Total column water vapour	kg/m^2
100 meter V wind	m/s	Total precipitation	m
10 meter wind gust in 3 hours	m/s	Visibility	m
10 meter wind gust in 6 hours	m/s	Divergence at 850 Pa	1/s
10 meter U wind component	m/s	Divergence at 900 Pa	1/s
10 meter V wind component	m/s	Divergence at 1 000 Pa	1/s
2 meter dewpoint temperature	K	Geopotential height at 850 Pa	K
2 meter temperature	K	Geopotential height at 900 Pa	K
Convective available potential energy	J/kg	Geopotential height at 1000 Pa	K
Convective available potential energy shear	m^2/s^2	Potential vorticity at 850 Pa	K · m^2/(kg · s)
Convective precipitation	m	Potential vorticity at 900 Pa	K · m^2/(kg · s)
0 °C isothermal level (atm)	m	Potential vorticity at 1 000 Pa	K · m^2/(kg · s)
Forecast albedo	–	Specific humidity at 850 Pa	kg
Accumulated freezing rain	m	Specific humidity at 900 Pa	kg
Low cloud cover	–	Specific humidity at 1 000 Pa	kg
Large-scale precipitation	m	Relative humidity at 850 Pa	–
Minimum temperature in 3 hours	K	Relative humidity at 900 Pa	–
Minimum temperature in 6 hours	K	Relative humidity at 1 000 Pa	–
Mean sea level pressure	Pa	Temperature at 850 Pa	K
Maximum temperature in 3 hours	K	Temperature at 900 Pa	K
Maximum temperature in 6 hours	K	Temperature at 1 000 Pa	K
Precipitation type	–	East-West U wind at 850 Pa	m/s
Snow density	kg/m^3	East-West U wind at 900 Pa	m/s
Snow depth	m	East-West U wind at 1 000 Pa	m/s
Snowfall	m	North-South V wind at 850 Pa	m/s
Skin temperature	K	North-South V wind at 900 Pa	m/s
Surface pressure	Pa	North-South V wind at 1 000 Pa	m/s
Sea surface temperature	K	Vertical velocity at 850 Pa	Pa/s
Total cloud cover	–	Vertical velocity at 900 Pa	Pa/s
Total column water	kg/m^2	Vertical velocity at 1 000 Pa	Pa/s