Short-term Rainfall Forecasting Using Multi-layer Perceptron

Pengcheng Zhang, Yangyang Jia, Jerry Gao, Wei Song, Hareton Leung

Abstract—Rainfall forecasting is crucial in the field of meteorology and hydrology. However, existing solutions always achieve low prediction accuracy for short-term rainfall forecasting. Numerical forecasting models perform worse in many conditions. Machine learning approaches neglect the influences of physical factors in upstream or downstream regions, which make forecasting accuracy fluctuate in different areas. To improve the overall forecasting accuracy for short-term rainfall, this paper proposes a novel solution called Dynamic Regional Combined short-term rainfall Forecasting approach (DRCF) using Multi-layer Perceptron (MLP). First, Principal Component Analysis (PCA) is used to reduce the dimension of thirteen physical factors, which serves as the input of MLP. Second, a greedy algorithm is applied to determine the structure of MLP. The surrounding sites are perceived based on the forecasting site. Finally, to solve the clutter interference which is caused by the extension of the perception range, DRCF is enhanced with several dynamic strategies. Experiments are conducted on data from 56 real-world meteorology sites in China, and we compare DRCF with atmospheric models and other machine learning approaches. The experimental results show that DRCF outperforms existing approaches in both threat score (TS) and root mean square error (RMSE).

Index Terms—Rainfall forecast, deep neural network, multi-layer perceptron, short-term, atmospheric models

1 Introduction

As an important part of water resource ecosystem, rainfall plays an important role in the field of hydrology and meteorology. General speaking, rainfall is the results of multi-scale air system interaction, and it is affected by many environmental factors, such as *thermal power*, *flow field*, and *terrain* [1]. These complex physical mechanisms make forecasting rainfall very difficult. Furthermore, rainfall forecasting is closely related to resident life. Especially, short-time heavy rainfall is the main weather element that causes urban water-logging, and can also predict road water accumulation ahead of time [2]. Consequently, due to the complex dynamic changes inside the atmosphere and the real-time requirement of short-term rainfall forecasting, a large-scale and high-precision forecasting model is urgently needed, which poses a big challenge to the field of meteorology and hydrology.

Currently, there are mainly two ways to forecast rainfall: atmospheric models [3] and machine learning approaches [4]. Atmospheric models simulate atmospheric operation. The atmospheric equation is a closed system describing atmospheric motion. Atmospheric equations can be used to predict atmospheric physical quantities and weather elements, including rainfall. According to their purposes, atmospheric models can be further divided into atmospheric circulation models, climate models and numerical models [5]. Atmospheric circulation models can forecast atmospheric physical quantities. Climate models can forecast various climatic factors, which usually forecast long-term rainfall. At present, numerical models are the main mode of forecasting medium and short-term rainfall in China. Since atmospheric equations are made up of partial differential equations, the overall accuracy of numerical computation depends on reasonable approximation. Furthermore, atmospheric models usually require a lot of computations and resources. These strict requirements for hardware make atmospheric forecasting difficult to implement.

Machine learning approaches are based on the theory of various machine learning algorithms [6], [7], [8]. Generally speaking, one kind of machine learning approaches are based on statistics, which only learn the time series variation characteristics from historical rainfall series. Other machine learning methods usually analyze the factors that may affect rainfall and the internal relation between these factors and rainfall. Because machine learning can learn the internal relation of data, an appropriate machine learning algorithm is the key to accurately learn the relation between different factors and rainfall. Artificial neural network (ANN) [9] is a kind of machine learning algorithm which simulates the human brain, and it has strong learning ability. However, traditional neural networks usually contain only one hidden layer with the simpler structure and shallow layers. They tend to fall into the local minimum and over-fitting.

Deep Neural Network (DNN) [10], [11], a multi-layer neural network, becomes the most prominent learning network in ANN. Due to the outstanding learning ability, DNN-based approaches have become the most effective machine learning approaches. In recent years, DNN has been applied to the field of image recognition, speech recognition

- *P. Zhang and Y. Jia are with College of Computer and Information, Hohai University, Nanjing, P.R.China E-mail: pchzhang@hhu.edu.cn; 1817582819@qq.com*
- *J. Gao is with Department of Computer Engineering, San Jose State University, San Jose, CA, USA & Taiyuan University of Technology, Taiyuan, China. Email: jerry.gao@sjsu.edu*
- *W. Song is with School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China E-mail: wsong@njust.edu.cn*
- *H. Leung is with the Department of Computing, Hong Kong Polytechnic University, HongKong, China. E-mail: hareton.leung@polyu.edu.hk*

Manuscript received XXXX, 2018.

and Natural Language Processing [12], [13], [14]. Compared with traditional ANN, DNN has stronger learning ability. It can learn deep links among data, which is proved to be an effective approach for classification and prediction. Usually, DNN includes Multilayer Perceptron (MLP) [15], Deep Belief Network (DBN) [16], Long Short-Term Memory (LSTM) [17], Conventional Neural Network (CNN) [18] and so on. At present, DNN-based approaches usually take a certain area as the research object, and study the relation between factors and rainfall [19], [20], [21], [22]. How to select the appropriate factors and the learning algorithm is the key step for the successful application of this kind of approaches. While the state-of-the-art approaches make some progresses on rainfall forecasting, DNN-based methods still suffer from the following two limitations:

1) Rainfall in a region may be influenced by the factors in upstream or downstream regions. However, current approaches usually regard each region as independent entity, and the rainfall in this area is forecasted only according to the factors in this area.
2) These methods usually choose one or several regions as the experimental objects, which makes their generalization ability unconvincing.

Rainfall forecasting is affected by many factors, such as *high-altitude physical factors* and *surface factors*. *High-altitude physical factors* play important roles in the movement of rainfall systems. *Surface factors* on the Earth also cause different rainfall. To address the limitations mentioned above, in this paper, a Dynamic Regional Combined short-term rainfall Forecasting model (DRCF) using Multi-Layer Perceptron (MLP) is proposed. The input of the model includes five *high-altitude factors* and eight *surface factors* from the target and around areas. The output of the model is the rainfall of the target area in the next three hours. In summary, the central contributions are summarized as follows:

1) Principle Component Analysis (PCA) is used to reduce dimension and determine the exact input of MLP. A special greedy algorithm and Stochastic gradient descent (SGD) with momentum are adopted to determine the suitable network structure and parameters of MLP, respectively.
2) The relation between the forecasting area and the surrounding area is established using a MLP model. Based on the relation, a dynamic regional combined rainfall forecasting model is proposed. A strategy of dynamically adjusting areas is also enhanced to effectively improve forecasting accuracy.
3) The generality of the model is finally validated by a large number of meteorological site data, including 56 sites, and these sites are distributed over all areas of China.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the main concepts used in the paper. Section 4 presents details of the DRCF model. The experimental validation of DRCF is performed in Section 5. Finally, Section 6 offers some conclusions and suggestions for future work.

2 Related work

2.1 Atmospheric Models

Since the numerical models are the main official atmospheric models adopted by short-term rainfall forecasting, we give a review of numerical forecasting models. Numerical forecasting models used in China mainly include European Centre for Medium-Range Weather Forecasts (ECMWF) [23], Japan Meteorological Agency numerical weather prediction model (JAPAN) [24], and Regional Assimilation and Prediction System (GRAPES) [25]. Due to the uncertainties caused by the initial and boundary conditions, there is always a deviation between numerical methods and the real situation. By summarizing the forecasting characteristics of numerical methods, some experts can correct the forecasting results based on their experiences. These methods have achieved certain acceptable results, but it often needs long-term experience and subjective judgment [26], [27]. Another disadvantage of numerical forecasting is that the hardware and software requirements are too high.

2.2 Machine Learning Approaches

The first kind of machine learning approaches are based on the theory of statistics. Time series analysis is a typical multivariate statistical analysis method. It is used to forecast rainfall based on the time series characteristics of rainfall. Time series analysis mainly includes Wavelet, Auto-regressive Integrated Moving Average Model (ARIMA), Markov model and so on [28]. These statistical approaches only start from the rainfall sequence and learn the various characteristics of the rainfall series. However, they do not take account of the influence of climate or other environmental factors on rainfall, and only grasp the general direction of rainfall change from the trend.

Other machine learning approaches can learn the relationship between meteorological/physical factors and rainfall. These kinds of approaches used in rainfall forecasting mainly include Support Vector Machine (SVM), Support Vector Regression (SVR), Grey Forecasting (GF), Artificial Neural Network (ANN) and so on [29]. Compared with statistical approaches, they can achieve better forecasting results [30]. Different machine learning algorithms have different effects on rainfall forecasting. Machine learning algorithm, factor selection, and prediction model are all important factors for successful forecasting, and it is particularly important to choose the appropriate machine learning algorithm. The traditional ANN models for rainfall forecasting usually include Feed-forward Neural Network (FNN), Back Propagation Neural Network (BPNN), Radial Basis Function Neural Network (RBFNN) and so on. Abbot [31] et al. use ANN to forecasts rainfall for sites in three geographically distinct regions in Queensland. The network has two hidden layers. Experimental results show that the approach is superior to General Circulation Model currently used. Abbot [32] et al. also use ANN to forecast monthly and seasonal rainfall in Queensland, Australia. This network takes meteorological or climatic circulation factors as input, such as temperature and El Nino index. They also study the input selection and optimisation of ANN. Farajzadeh [29] et al. forecast rainfall in the Urmia basin of northern Iran by using a feed-forward neural network which contains a

hidden layer. Wu [33] et al. take 500hPa height field and Sea Surface Temperature (SST) field as the inputs of principal component analysis and RBFNN. This model is used to predict the average rainfall in central Guangxi, China. Traditional neural networks usually has shallow layers and tend to fall into the local minimum and over-fitting problems.

To distinguish from the traditional ANN architecture, DNN is used to define a kind of network with more layers, which is more complex than the traditional network. Currently, there are many attempts to forecast rainfall using different kinds of DNN [34]. These approaches usually take meteorological or physical factors of a certain area as the input, and rainfall as the output. They learn the internal relationship between factors and rainfall, which is the basis for rainfall forecasting [11]. Lin [35] et al. use MLP to predict typhoon rainfall. They classify some possible factors by self-organized mapping algorithm. These factors contain typhoon position, wind speed around typhoon, moving direction of the typhoon, moving speed of typhoon, pressure and so on. The approach can only be used to predict the typhoon rainfall effectively. Zhang [21], [36] et al. establish a rainfall forecasting model based on DBN. Validation of the model is proved by experiments on two areas: Zunyi and GuiYan, which are two cites of China. Mekanik [37] et al. analyze the relation between climate factors and spring rainfall in Vitoria area, Australia. They forecast spring rainfall based on multiple linear regression and MLP, respectively. It is found that the generalization performance and accuracy of MLP are better than other models. However, the approach still focuses on one area. Shi et al. [22] propose a novel model based on LSTM in HongKong area. This model is used to predict the structure and direction of clouds in the radar picture. It can predict short-term rainfall better, and the prediction effect is better than other approaches. As we already presented, most of these approaches still suffer from some technical challenges, such as do not consider the factors in other regions and the generalization is not validated, which are addressed in our approach.

3 PRELIMINARIES

3.1 Principal Component Analysis

Principal component analysis (PCA) is a data preprocessing and dimension reduction theory commonly used in multivariate statistics [38]. When the number of variables is too large, the complexity of processing increases. The main ability of PCA is that it can represent more information with fewer variables. When the correlation coefficient between the initial variables is not 0, it could be explained that there is a certain overlap between these variables. For all initial variables, PCA can delete repeated relations of variables and create new variables as few as possible [39]. These new variables are not related to each other, but the original information could be kept as much as possible. By transforming the vector space composed of input samples, PCA can make the direction of the largest error as the base vector of the new linear space. In fact, PCA can represent more information with relatively fewer factors. It usually simplifies the complexity of the problem and reduces the consumption of hardware resources. The calculation formulae of PCA is described as:

$$Z_i = \mu_{i1}X_1 + \mu_{i2}X_2 + \cdots + \mu_{ip}X_p, 1 \leq i \leq p \qquad (1)$$

where p dimensional random vector $X = (X_1, X_2, ..., X_p)_t$ represents the original variable. t represents the t moment. A linear transformation is carried out for X, and the original variable X is changed to new variables $Z = (Z_1, Z_2, ..., Z_p)_t$. By choosing the appropriate coefficient μ, the factors of Z cannot be correlated. The main information is concentrated in the first few components of Z. The first few components can be used to represent the whole information. The calculation of coefficients is usually done by a correlation coefficient matrix, which is numerically equivalent to the eigenvector of the correlation coefficient matrix.

3.2 Multi-Layer Perceptron

ANN is an operational model which consists of a large number of interconnected nodes (neurons). Each node contains a specific output function which is called an activation function. The connection between every two nodes represents a weighted value that passes through the connection signal, which is called weight. Weight is equivalent to the memory of ANN. DNN is a kind of ANN with deeper level and more complex structure, and MLP [40] is a representative structure of DNN. MLP has many layers, the first layer is the input layer, the last layer is the output layer, the middle layers are called hidden layers, and each layer includes several neurons. The neurons in MLP have the characteristics of inter-layer full connection and intra-layer connectionless [41]. Fig. 1 is a schematic diagram of MLP which has two hidden layers. The input layer has a total of N_1 nodes, which represent the dimension of the input data. The hidden layer 1 and the hidden layer 2 have N_2 and N_3 neurons, respectively. The output layer has N_4 neurons, which represent the dimension of output data. Apart from the output layer, each layer has a bias node.

3.3 Adjustment and Constructive Algorithm of MLP

MLP is a kind of feed-forward network. It is calculated from the input layer to the output layer successively. Each node at the same level is calculated at the same time, and it does not interfere with each other [42]. The value of every node is equal to the weighted summation of all nodes in the previous layer. This calculation process is called the feed-forward process of MLP. If there is a MLP which contains m hidden layers, its input and output dimensions are respectively equal to n_1 and n_{m+2}. The number of nodes in each hidden layer is n_2, n_3, ..., n_{m+1}, respectively. In the feed-forward process of this MLP, each node value is calculated using the following formula:

$$x_{ij} = f\left(W_i X_{i-1} + b_{i-1}\right) \qquad (2)$$

where X_{ij} represents the value of the j neuron in the i layer. W_i represents the weight vector of the j neurons in layer $i-1$ to layer i. X_{i-1} represents the value vector of all neurons in layer $i-1$. b_{i-1} represents the bias of the $i-1$ layer, and f is the activation function.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2018.2871151, IEEE Transactions on Big Data

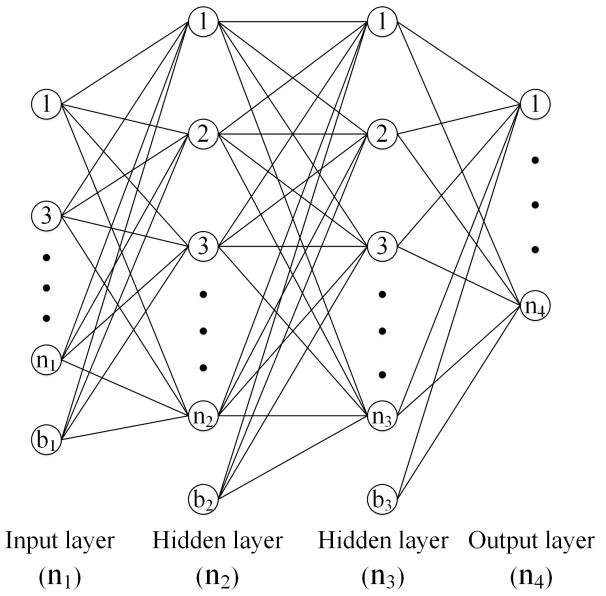

Fig. 1. Structure of MLP

The activation functions which are frequently-used mainly include nonlinear functions and linear functions. If the activation function is a linear function, MLP is essentially a single layer perceptron. Consequently, the nonlinear function is always used in the middle layer, and the output layer uses the linear function. Sigmoid function is a commonly used nonlinear activation function. The expression of Sigmoid function is described as follows:

$$f(x) = sigmod(x) = \frac{1}{1+e^{x}} \tag{3}$$

MLP is a supervised learning algorithm. There is an ideal output corresponding to any input. The loss function of the ideal output and the actual value is defined as:

$$J(W,b;x,y) = \frac{1}{2}\left\|h_{W,b(x)} - y\right\|^{2} \tag{4}$$

where h represents the output value, y represents the actual value, and $||.||$ represents any distance norm.

To minimize the loss function, the parameters of MLP are usually adjusted by error Back-Propagation algorithm (BP). The Gradient descent (GD) algorithm is the most commonly used and the simplest parameter adjustment algorithm in the BP algorithm. However, the GD algorithm has the disadvantages of slow convergence and easy to fall into the local minimum. To overcome the shortcomings of the GD algorithm, some improvements are proposed. Consequently, we use Stochastic gradient descent (SGD) with momentum to adjust the parameters of MLP. The calculation method of gradient and SGD with momentum are described as follows:

$$\bigtriangledown W = -\frac{\partial J(W,b;x,y)}{\partial W} \tag{5}$$

$$\delta W_t = \alpha \bigtriangledown W_t + \beta \delta W_{t-1} \tag{6}$$

where Formula 5 represents the calculation method of the gradient. In Formula 6, the value of α and β are taken from 0 to 1, respectively. In this paper, from our experiences, both α and β are set to 0.5.

For MLP, there are usually two ways to find the optimal or sub optimal network structure [43]. The first method is to reduce network size on a large network by pruning technology. The second method is to start from a small network, and by constantly increasing the size of the network. Because the second method is fast and low cost, it is adopted in this paper. The greedy algorithm is a commonly used constructive algorithm. The structure of MLP achieves the best state by adding hidden layer neurons layer by layer. However, the common greedy algorithm is usually determined from the first level to the last level. The innovation of this paper is to compare two orders, i.e., from the first layer to the last layer and from the last layer to the first layer. The order is chosen according to the real network performance.

4 Dynamic Regional Combined Forecast Model

In this section, we first give an overview of our approach in Section 4.1, which presents the main steps for building a DRCF model. Then, all the steps are detailed in the following subsections.

4.1 Overview

When DNN is applied to practical problems, the main difference lies in the network structure and parameters. DRCF is a novel application framework using MLP for rainfall forecasting. In general, using a single MLP may give wrong results. Then, it could make progress when different MLPs are used at the same time. Consequently, we use several MLPs to forecast rainfall in a region at the same time to improve prediction results. These MLPs have different inputs, but they have the same target output, i.e., rainfall. At the same time, the parameters of these MLPs are dynamically changed. Fig. 2 shows the main steps of DRCF.

1) *Data collection and pre-processing*. The data needed is collected and necessary pre-processed. In this approach, the main work of pre-processing includes *deviation standardization* and *PCA*.
2) *Structure selection and parameter training*. The structure of MLP is determined using a greedy algorithm, and the optimal parameters of MLP, including weight and bias, are determined through an adjustment algorithm.
3) *Range parameter Determination and model optimization*. During this step, we take the prediction area as the center and select the appropriate perception range. Different perception ranges mean different numbers of MLPs are chosen. The model tries to dynamically change the range to optimize itself, which means that the number of MLPs is dynamically changing during the forecasting process. Finally, suitable MLPs are used to forecast rainfall.

4.2 Data Collection and Pre-processing

The main work of pre-processing includes *min-max normalization* and *PCA*.

The initial data of this model include five high-altitude factors and eight earth surface factors. In meteorology,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2018.2871151, IEEE Transactions on Big Data

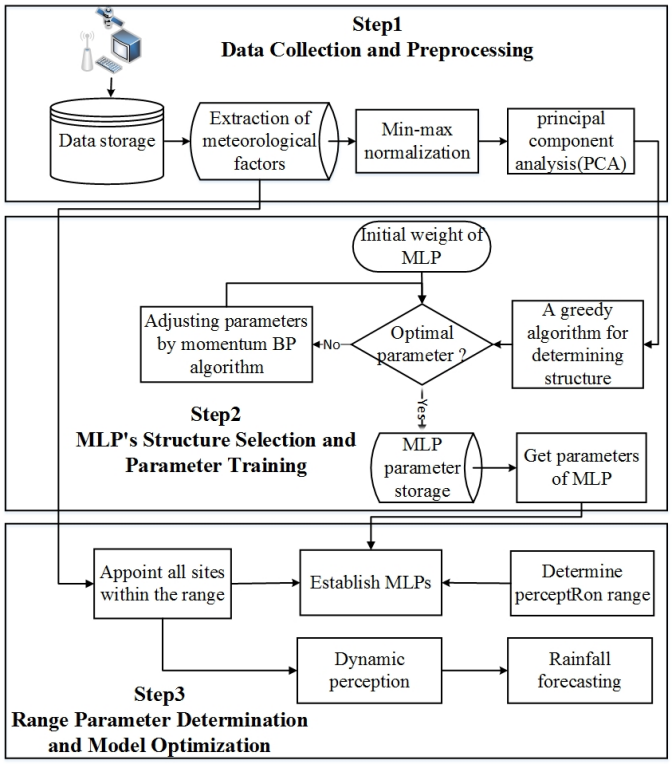

Fig. 2. Process of establishing DRCP

geopotential height is usually used instead of actual height, and an isobaric surface is used instead of horizontal height. Therefore, meteorological data is always in the format of the isobaric surface. For example, 500hPa is usually corresponding to the height of 5.5km. The rainfall system is often controlled by the weather system of 500hPa. According to the domain experiences, the five altitude factors selected by this model are *actual height* (X_1), *temperature* (X_2), *temperature dew point difference* (X_3), *wind direction* (X_4) and *wind speed* (X_5) at 500hPa height. *Wind direction* and *wind speed* affect the direction and speed of the movement of rainfall system. *Temperature dew point difference* is directly related to the humidity. *Temperature dew point difference*, *temperature* and *actual height value* affect the internal energy of rainfall system. *Ground surface factor* represents the local atmospheric condition in the area. The difference in surface factors in different regions leads to different rainfall. The eight surface factors used in this model include *total cloud amount* (X_6), *ground surface wind speed* (X_7), *ground surface wind direction* (X_8), *ground surface air pressure* (X_9), *ground surface 3 hour pressure change* (X_{10}), *ground surface temperature dew point difference* (X_{11}), *ground surface temperature* (X_{12}), and *rainfall over past three hours of surrounding areas* (X_{13}). For the same forecasting area, every surrounding area establishes a MLP with this forecasting area. Table 1 shows all the thirteen factors. These factors are the initial input of our model.

Min-max Normalization. One of the most commonly used data normalization methods is *Min-max normalization*. It can standardize data between 0 and 1. Because different factors have different magnitude, it is necessary to pre-process the data. For a sequence to be processed, the maximum value of the sequence corresponds to 1, the minimum value

TABLE 1
A list of factors used in the model

Factor	Input value
500hPa height(X_1)	Forecast area value
500hPa temperature(X_2)	Forecast area value
500hPa temperature dew point difference(X_3)	Forecast area value
500hPa wind direction(X_4)	Forecast area value
500hPa wind speed(X_5)	Forecast area value
Total cloud amount(X_6)	Forecast area value − neighbouring area value
Surface wind speed(X_7)	Forecast area value − neighbouring area value
Surface wind direction(X_8)	Forecast area value − neighbouring area value
Surface air pressure(X_9)	Forecast area value − neighbouring area value
Surface 3 hour pressure change(X_{10})	Forecast area value − neighbouring area value
Surface temperature dew point difference(X_{11})	Forecast area value − neighbouring area value
Surface temperature(X_{12})	Forecast area value − neighbouring area value
Rainfall over past 3 hours(X_{13})	Neighbouring area value

corresponds to 0, and the rest values are proportionally transformed between 0 and 1. The formula for *min-max normalization* is shown as follows:

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7}$$

where x represents a value in the sequence of primitive variables, x_{max} and x_{min} represent the maximum and minimum values in variables, respectively.

Principal Component Analysis. After normalization, PCA is used to reduce the dimension of the input. The criterion for determining the number of new factors is 99%, that is, the sum of the chosen factors eigenvalues accounts for more than 99% of the total eigenvalues. After calculation, the total information of new factors can represent more than 99% of the original data. This criterion defines the amount of information retained, but it does not specify the number of factors needed. For different forecasting areas, the number of factors may be different, but it will not exceed the initial input, i.e., thirteen. In most cases, the number of factors needed is between three and eight. After the PCA process, the computing resources required is greatly reduced.

Example 1. To explain the PCA process in detail, we illustrate the use of the PCA based on *Fuyang*, a city in *Anhui* province of China. It is located in the middle east part of China. Then we take *Hefei*, the capital of *Anhui* province, as the neighboring site. Table 2 is the PCA results of *Fuyang*. The first column represents the initial thirteen factors, and the first row gives the new factors from Z_1 to Z_4. The value of each column represents the coefficient from X to a new factor Z. The last row represents the eigenvalue of the new factor. It can be seen from the eigenvalues that four new factors occupy more than 99% of the total information. Consequently, four new factors are used to replace the original thirteen factors.

TABLE 2
PCA result of 13 factors in *Fuyang*

Factors	Z_1	Z_2	Z_3	Z_4
X_1	-0.557	-0.009	-0.014	0.008
X_2	-0.001	-0.050	-0.490	0.816
X_3	-0.052	-0.018	-0.068	0.056
X_4	-0.072	0.006	-0.005	0.051
X_5	0.082	-0.009	-0.011	0.016
X_6	0.003	0.105	-0.060	-0.080
X_7	0.006	0.082	-0.018	-0.072
X_8	0.003	-0.365	0.535	0.229
X_9	0.005	-0.393	0.485	0.182
X_{10}	-0.005	0.818	0.459	0.305
X_{11}	-0.003	0.092	0.075	0.058
X_{12}	0.535	0.120	-0.117	-0.365
X_{13}	1.000	0.007	-0.002	0.002
Characteristic value	25.154	9.199	5.060	1.037
Cumulative ratio(%)	62.021	84.703	97.179	99.736

4.3 Structure Selection and Parameter Training

The main tasks of this step include determining the structure of MLP using a greedy algorithm and determining MLP parameters using SGD with momentum.

We first explain the main idea and the process of building MLP. To predict rainfall in a certain area, the model establishes links between the forecasting area and surrounding areas. This connection refers to the relationship between the thirteen factors and the rainfall in the forecasting area. Since MLP has a strong learning ability, DRCF uses MLP to learn this connection. The input of MLP is the data preprocessed by PCA, and the output is the forecasting area.

***Example* 2.** Considering *Fuyang*, the input of MLP is Z_1, Z_2, Z_3 and Z_4 in Table 1, and the output is rainfall that we need to forecast. The forecast interval for this model is short-term, i.e., three hours. If the neighboring area is now raining and MLP indicates that the forecasting area rains, it is predicted that the region will rain in the future and the rainfall is equal to the output of MLP. The forecast area is used as the center area to search for the neighboring areas that have already rained, and the rainfall in the next three hours is forecasted by MLP. This process is also called *dynamic perception*. Fig. 3 shows the process of *dynamic perception*. The larger circle in the middle indicates the forecasting area. The surrounding smaller circles represent neighboring areas. The solid line represents the relation between the adjacent area and the forecasting area. Different dashed rings represent the different range of perception. By learning historical data, we can adjust the parameters of the MLP network, and learn the relation between neighboring sites and the forecasting site. Then, the model stores these parameters of MLP and uses them later.

The structure of MLP includes the depth and width of the network. Larger depth and width may lead to overfitting problem. Finding the right depth and width can not only save computer resources but also enhance the generalization ability of the model. The main idea of structural selection is to compare the prediction accuracy of MLP with different structures. MLP with high accuracy and simple structure is selected. Meanwhile, Root Mean Square Error (RMSE) is used to calculate the accuracy of MLP. Smaller RMSE represents better prediction effect. The formulae of RMSE is described as follows:

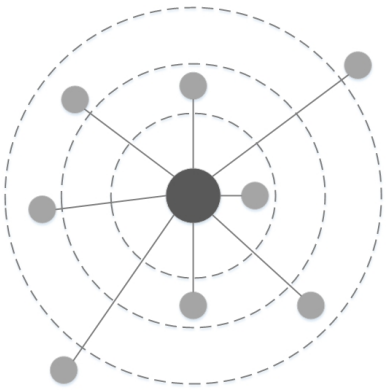

Fig. 3. MLP's establishment and dynamic perception process

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y - \tilde{y})} \tag{8}$$

where N represents the number of samples, y and $\tilde{y}$ represent the actual rainfall and the predicted rainfall of each sample, respectively.

Because Breadth and width of MLP are two-dimensional variables, it is impractical to compare all possible situations. Therefore, in this paper, a general greedy algorithm is used to determine the structure of MLP. First, we calculate the accuracy of the first hidden layer of MLP. The hidden layers of these MLPs contain the different number of nodes. The number of nodes corresponding to the minimum *RMSE* is selected as the number of nodes in the first hidden layer. In this way, the number of nodes in the first hidden layer is determined. Then, we calculate the second hidden layers according to *RMSE*. The number of nodes corresponding to the minimum *RMSE* is selected as the number of nodes in the second hidden layers. In this way, we can determine the number of nodes in the next hidden layer. Normally, there are two sequential ways to determine the number of hidden layer nodes, from the input layer to the output layer or from the output layer to the input layer. Algorithm 1 gives the detail process of using the greedy algorithm to determine the MLP structure. The algorithm consists of two parts, including the greedy method to determine the structure of MLP and the SGD algorithm with momentum to optimize the corresponding parameters. In the algorithm, we store the final number of factors for each layer through an array $array$, and $layer$ represents the number of hidden layers. Through the outermost *WHILE* cycle, we fix the number of layers, and calculate *RMSE* of MLP with the different number of factors. Finally, the structure of MLP is determined according to the minimum *RMSE*.

While the greedy algorithm is adopted in the structure selection of MLP, SGD with momentum is used to adjust weights and bias. When the *RMSE* of MLP reaches the minimum, the model stops training. This *RMS*E is called the *RMSE* of this MLP. Example 3 clearly illustrates the process of using the greedy algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2018.2871151, IEEE Transactions on Big Data

Algorithm 1 Using the greedy algorithm to determine the structure of MLP

Require: $array$ is an empty array to storage Structural elements, δ and β are learning speed.

Ensure: $layer$ is the hidden layer number of the best MLP, $array$ which stores the optimal number of neurons in each hidden layer.

$N \leftarrow num_{max}, M \leftarrow layer_{max}$
$num \leftarrow 1, layer \leftarrow 1$
$RMSE \leftarrow 0$
while $layer \leq M$ && $RMSE$ $increase$ **do**
  while $num \leq N$ **do**
    $array[layer] \leftarrow num, rmse \leftarrow 0$
    if from input layer to output layer **then**
      establish a MLP with the structure $\{input, array[1],..., array[layer], output\}$
    else if from output layer to input layer **then**
      establish a MLP with the structure $\{input, array[layer],..., array[1], output\}$
    end if
    $w_{ij} \leftarrow random()$
    $b_{ij} \leftarrow random()$
    $\delta W = 0$
    while $step \leq STEP$ && J $increase$ **do**
      $x_{ij} \leftarrow f(W_i X_i - 1 + b_{i-1})$
      $J \leftarrow \frac{1}{2}(h_{W,b} - y)^2$
      $\bigtriangledown W = -\frac{\partial J(W,b;x,y)}{\partial W}$
      $\delta W = \alpha \bigtriangledown W + \beta \delta W$
    end while
    $rmse[num] \leftarrow J$
    $num \leftarrow num + 1$
  end while
  $best_{num} \leftarrow \{num | rmse[num] = \min(rmse)\}$
  $RMSE \leftarrow rmse[best_{num}]$
  $array[layer] \leftarrow best_{num}$
  $layer \leftarrow layer + 1$
end while

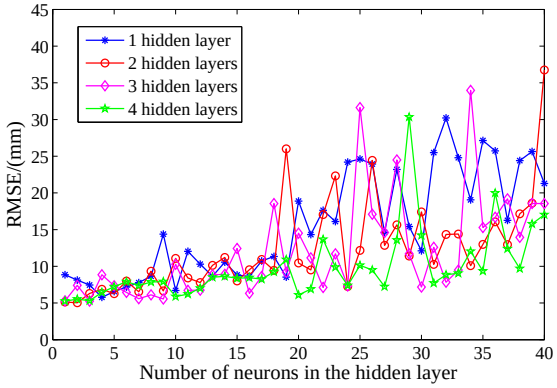

Fig. 4. Process of selecting the number of neurons in MLP's hidden layer from input layer to output layer

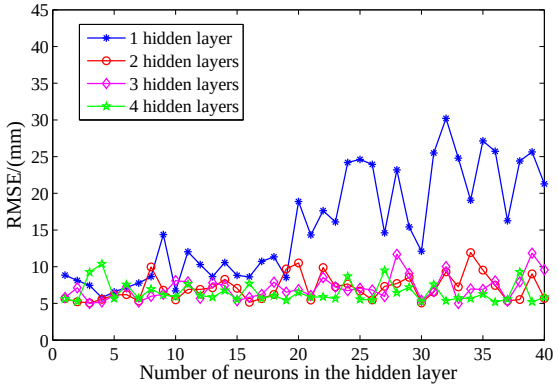

Fig. 5. Process of selecting the number of neurons in MLP's hidden layer from output layer to input layer

Example 3. We still take *Fuyang* as an example. When we forecast rainfall in *Fuyang*, we need to establish an MLP model between *Fuyang* and *Hefei*. The input of this MLP is the four new factors that have been pre-processed by PCA, and the output is *Fuyang*'s rainfall. Fig. 4 and Fig. 5 show the process of determining the structure of MLP using the greedy algorithm under two orders, respectively. The horizontal axis in the two graphs represents the number of hidden layer nodes, and the vertical axis represents the *RMSE* of MLP. The four fold lines represent the MLP of the 1-4 hidden layers. Fig. 4 is the process of selecting the number of neurons in MLP's hidden layer from the input layer to the output layer. Fig. 5 is the process of selecting the number of neurons in MLP's hidden layer from the output layer to the input layer. The four lines from the input layer to the output layer are intersecting each other. From the first hidden layers to the fourth hidden layers, *RMSE* does not decrease significantly. The fluctuation of *RMSE* is too large to distinguish the difference between multi-layer and single layer. This indicates that the multi-layer structure has no obvious effect compared with the single-layer structure. The difference between the four lines from the output layer to the input layer is quite different. *RMSE* difference between 2-4 hidden layers of MLP is relatively small, which cannot be significantly different. *RMSE* of the second hidden layers is significantly less than that of the first hidden layers, which indicates that the prediction ability of MLP is obviously improved. In contrast, we can see that from the output layer to the input layer, it is more effective to determine the MLP structure using a greedy algorithm. From the output layer to the input layer, the prediction ability of the MLP with 2-4 hidden layers is not significantly different. Considering the cost of computation, MLP with two hidden layers is selected. From the input layer to the output layer, the number of nodes in each hidden layer is determined to be 40 and 10. For the MLP established between any two sites, the structure of MLP is determined from the output layer to the input layer, and the number of nodes in each hidden layer is determined using a greedy algorithm.

4.4 Range Parameter Determination and Model Optimization

In this step, we use multiple MLPs to forecast rainfall, and these MLPs have been trained in the previous step. The main work of this step is to determine the appropriate number of MLPs to be used. A single MLP may predict with large errors. To improve prediction accuracy, several MLPs are used to make decisions together. These MLPs must forecast rainfall from different aspects so that they can make up for each other. The perception process mentioned in the previous section can solve this problem. Taking the forecast area as the center area, we can establish an MLP with all the surrounding areas. The number of surrounding areas directly determines how many MLPs can be established. These MLPs can be used to forecast rainfall in the area. When there is an MLP which forecasts rainfall, this model will forecast rainfall, and the rainfall forecasted by the model is equal to the rainfall forecasted by MLP. When the number of MLPs which forecasts rainfall is more than 1, the average value of these forecasts is used as the forecast value of the model.

There are two ways to determine the number of MLPs. The first way is to determine the number of MLPs directly. This way needs to count the optimal number of MLPs in many areas, which makes the prediction accuracy of the model highest. The second way is based on the distance between the two regions, called *perception radius*. Assuming that the forecasting area is the center of the circle, MLP is built according to all the surrounding areas within a certain distance. The distance is calculated using the following formula:

$$l = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{9}$$

where x_1 and y_1 represent the longitude and latitude of the forecasting sites, respectively. x_2 and y_2 represent the longitude and latitude of the surrounding sites, respectively. The distribution density of meteorological stations in different regions is different. The second way takes into account the influence of this factor on meteorological data. Consequently, the second way is more appropriate.

The greater the number of MLP, the higher the forecasting accuracy of forecast rainfall. Under such circumstances, very little rainfall will be missed, but there will be many redundant forecasts, called *clutter interference*. With the increase of the *perception radius*, the new perceptive sites sometimes make erroneous predictions on the forecast area and interfere with the rainfall forecast in the forecasting area. At this time, there is no rainfall in the forecasting area, but the model predicts rainfall. This indicates that rainfall may occur in other areas, and does not affect the forecasting area. The *perception radius* is too large. Therefore, the prediction ability depends on the suitable size of the *perception radius*. To reduce *clutter interference*, some improvement strategies have been put forward. If there is rainfall in the forecasting area, *perception radius* will increase. If there is no rainfall in the forecasting area, *perception radius* will decrease. The specific strategies are described as follows:

- If there was no rainfall during past 3 time units (9 hours), and there was no rainfall in the neighboring area, the *perception radius* is set as 1.5.
- If there was no rainfall during past 3 time units, and some neighboring areas began to rain, the *perception radius* is set as 2.5.
- If there was no rainfall during past 1 time units, the *perception radius* is set as 1.5.
- For other cases, the *perception radius* is set as 2.

5 Experimental Evaluations

In this section, we conduct a set of experiments to validate DRCF based on meteorological data issued by the China Meteorological Administration[1]. The experiments are designed to investigate the following three research questions:

- RQ1: Does the number of MLPs affect forecasting accuracy?
- RQ2: What is the exact forecasting accuracy of DRCF?
- RQ3: Does DRCF outperform state-of–the-art models or approaches?

All the codes and experimental results can be downloaded from https://github.com/QXL4515/DRCF.

5.1 Experimental Setup

The experimental environment is a Windows PC with Lenovo Intel Core i3-2330M CPU 8G RAM. The operating system of the computer is Windows 7. The implementation environment is MATLAB 2013a. Both images and data processing are implemented in MATLAB. We evaluate the forecasting results based on the following indexes:

RMSE. *RMSE* is a commonly used error calculation index in statistics. As *RMSE* is already defined, we will still use it in our experiments.

Threat Score. TS is a common used method for calculating skill-scores of the forecast results, which is mainly used in China Meteorological business [2]. In our experiment, TS is also used to evaluate the forecasting quality. The China Meteorological Administration stipulates that TS is the calculation method of short-term rainfall forecast accuracy and prediction capability. TS_1 is used to represent the proportions of samples which are forecasted correctly, including accurate forecasting of rainfall or no rainfall. TS_2 is used to represent the proportions of samples which can only accurately forecast rainfall. TS_3 is the average value of TS_1 and TS_2. The detail descriptions of calculating TS score are described in the following:

$$TS_1 = \frac{N_1 + N_2}{N_1 + N_2 + N_3 + N_4} \tag{10}$$

$$TS_2 = \frac{N_1}{N_1 + N_3 + N_4} \tag{11}$$

$$TS_3 = \frac{TS_1 + TS_2}{2} \tag{12}$$

where N_1 represents the number of samples to predict the correct precipitation. N_2 represents the correct number of samples for forecasting no precipitation. N_3 represents the

1. http://data.cma.cn/
2. http://www.nmc.cn/publish/nwp/modelvalidation/meteor-elements/ts/

number of samples that predict no precipitation but actually rain. N_4 represents the number of samples that predict precipitation but do not actually have precipitation.

Contour Line. CL [3] is a line describing the distribution characteristics, which is usually used in geography, meteorology, mathematics and other fields. Contour line is a curve of points with same value in the map. At the same time, interpolation method is used to find the points on the contour line, and these points do not have detailed data. A contour map is a map with several contour lines. Different contour lines are drawn on a map to show the distribution and variation characteristics of variables. A total of 56 meteorological stations are selected in this experiment, which are distributed in all parts of China. It is very messy or redundant to differ the forecasting accuracy rate of 56 areas with each other using tables or curves. Although 56 areas are distributed in various provinces of China, it is difficult to represent all areas of China. The contour map can be used to solve the distribution discontinuity problem of areas by interpolation. The biggest advantage of contour map is that we can intuitively get the distribution characteristics of accuracy rate. Therefore, contour map is used to describe the forecasting accuracy rate of DRCF in China. The accuracy distribution in the map can be seen in the contour map.

5.2 Data Set

The experimental data include 2015-2017 years' Micaps *surface mapping data*, *altitude (500hPa) mapping data* and *numerical forecasting results* released by the China Meteorological Administration. Thirteen physical quantities required by our experiments could be obtained from *surface mapping data* and *altitude mapping data*. Rainfall in forecasting area can be obtained from ground mapping data. Results of numerical models are used to compare with this model. Mapping data in 2015 and 2016 are used as training data of the model, and the data in 2017 are used as testing data of the model. Currently, there are more than 1000 county level meteorological stations in China. These meteorological stations are distributed over China. Because the data of some meteorological stations are incomplete, the meteorological observatory used in the experiment only includes 920 stations. Among these meteorological stations, 56 stations were selected as the target objects for forecasting. These 56 meteorological stations are usually the central stations of one certain area. They have radar and high altitude weather data. DRCF requires a central site and several surrounding sites. The central site is selected from 56 meteorological stations. The surrounding area is then selected from 920 meteorological stations. In Fig. 6, the blue points represent 56 central meteorological stations, and the smaller red dots represent 920 meteorological stations.

5.3 Baseline Approaches

To prove that DRCF outperforms other state-of-the-art approaches. The baseline approaches are described as follows:

- ECMWF: European Centre for Medium-Range Weather Forecasts. This is the official used atmospheric model in China.

3. https://en.wikipedia.org/wiki/Contour

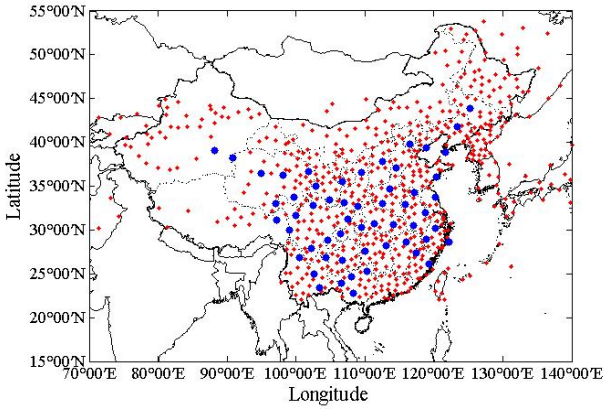

Fig. 6. Distribution of forecasting sites

- JAPAN: Japan Meteorological Agency numerical weather prediction model. This is also the official used atmospheric model in China.
- ARIMA: Auto-regressive Integrated Moving Average Model, which is a traditional and widely used time series model.
- BPNN: Back-Propagation Neural Networks, which is a traditional shallow neural network based on BP.
- RBFNN: Radial Basis Function Neural Network, which is another traditional shallow neural network based on RBF.
- DBN: Deep Belief Network, which is a common used deep learning model based on restricted Boltzmann machines.
- MLP: Multi-layer Perceptron, which is only based on the deep learning model MLP.
- DRCF: This is the model proposed in this paper, which is based on MLP and has a dynamical regional combined mechanism.

ECMWF and JAPAN are commonly used atmospheric models in meteorological operations in China. At present, the mainstream forecasting software in China is based on these two models. The forecasting results of them can be directly obtained from http://data.cma.cn/. ARIMA does not need to use physical quantity factors. It forecasts directly from the rainfall sequence. We use the same input for the other five approaches. Thirteen physical quantities are processed by PCA, and the comprehensive factors are used as the input of other five approaches at the same time.

5.4 Experimental Results

5.4.1 Relation between the number of MLPs and accuracy

For RQ1, we design a set of experiments to study the relation between the number of MLPs and accuracy. When the number of MLPs is different, the forecasting accuracy of the model is also different. First, an area (*Fuyang*) is used as an example to illustrate the effect on forecasting performance. Taking *Fuyang* as an example, we establish contacts with neighboring areas. The number of neighboring areas is equal to the number of MLPs. Fig. 7 shows three accuracy rates when the number of MLPs is different. With the increase in the number of MLPs, TS_1 decreases and TS_2 increases. If

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2018.2871151, IEEE Transactions on Big Data

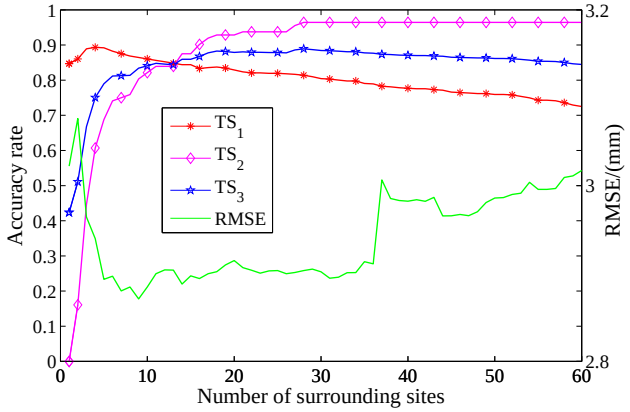

Fig. 7. Accuracy rate of model with different number of MLPs in *Fuyang*

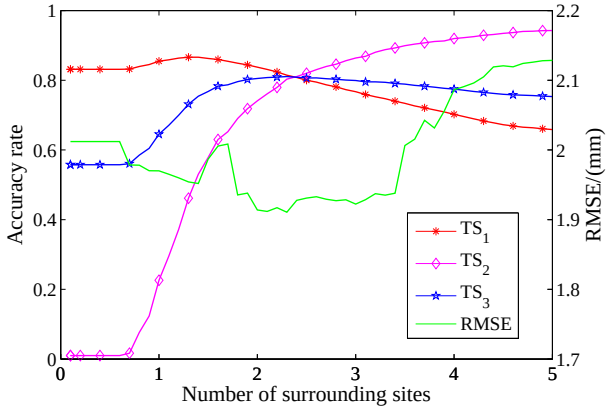

Fig. 9. Accuracy rate of model with different number of MLPs

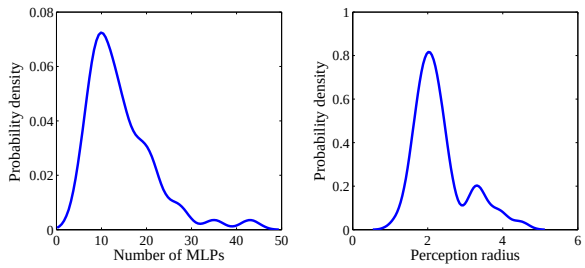

Fig. 8. Probability distribution of the optimal number of 56 regions

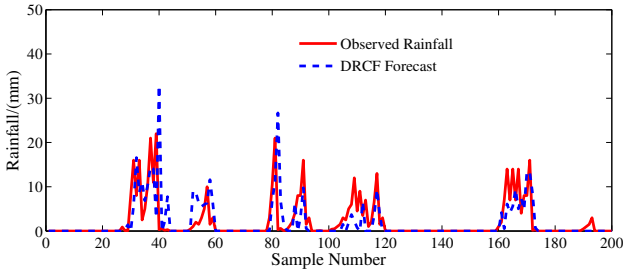

Fig. 10. DRCF's forecast result for *Fuyang* in August 2018

the error prediction is considered as clutter interference, the phenomenon in the picture can be explained as the accuracy of MLP will be higher when the number of MLPs increases. When the number of MLPs increases, the clutter interference is also stronger, and TS_1 also decreases. To comprehensively evaluate the model, the average of two accuracy rates is taken as the evaluation standard TS_3. The maximum TS_3 is achieved when the number of points in the perception station is 15. *RMSE* also gets the minimum value near 15, so it is considered that the forecasting effect is best when the number of sites is 15. Therefore, the number of MLP changes, the accuracy of the model also changes. For *Fuyang*, when the number of MLP is 15, the forecasting ability of the model is strongest.

The above example shows that an optimal number can make DRCF accuracy rate reach the highest. However, how does DRCF determine the optimal value of MLP's number is still a problem. For different regions, the best value may be different. Consequently, the experiments count the best values of all regions and hope to find the rules. The density of the sites around the region is different from each other. The best value may vary from one region to another.

For *Fuyang*, the number of MLPs is equal to 15, when the forecasting ability of the model is strongest. There are 56 optimal values in the 56 forecast areas. Probability distribution shows the best interval. There are two kinds of probability distributions in all. The first method is the direct statistics of 56 optimal numbers. The second method is calculated according to the perception radius. The different radius corresponds to different MLP numbers. There is 56 optimal radius in 56 forecasting sites. Fig. 8 represents two probability distributions. When the TS_3 of a forecasting area is the largest, the corresponding number of MLPs is called the optimal number, and its corresponding perception radius is called the optimal radius. The left picture in Fig. 8 gives the probability distribution of the optimal number of 56 regions. The right picture in Fig. 8 gives the probability distribution of the optimal radius of 56 regions. It is found that the optimal condition is more concentrated when perception radius is used as the parameter of the perception range. When the number of perception stations is used as the range parameter, the data is dispersed. Therefore, the model chooses the perception radius as the range parameter.

According to the above comparison, it is more appropriate to use perception radius as the criterion for determining the number of MLPs. When the perception radius is determined, the parameters of the model, including the structure and perception scope of MLP, have been determined. It is applied to all forecasting sites, and the average forecasting accuracy of all the sites is calculated. Fig. 9 is the forecasting accuracy of the 56 sites. When perception radius is 2, the overall accuracy rate reaches the maximum and the average $RMSE$ is also approximately the minimum.

Example 4. We still take *Fuyang* as an example to illustrate the difference between forecasting values and the observed values. Since August is the most rainfall concentration period in *Fuyang*, there should be 31*8 samples in August, and 199 samples are screened out to illustrate through time series diagram. Fig. 10 shows the prediction results of *Fuyang* in August 2018 using DRCF. It can be seen from the figure that the forecasting effect of DRCF is ideal in general. It can not only grasp

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2018.2871151, IEEE Transactions on Big Data

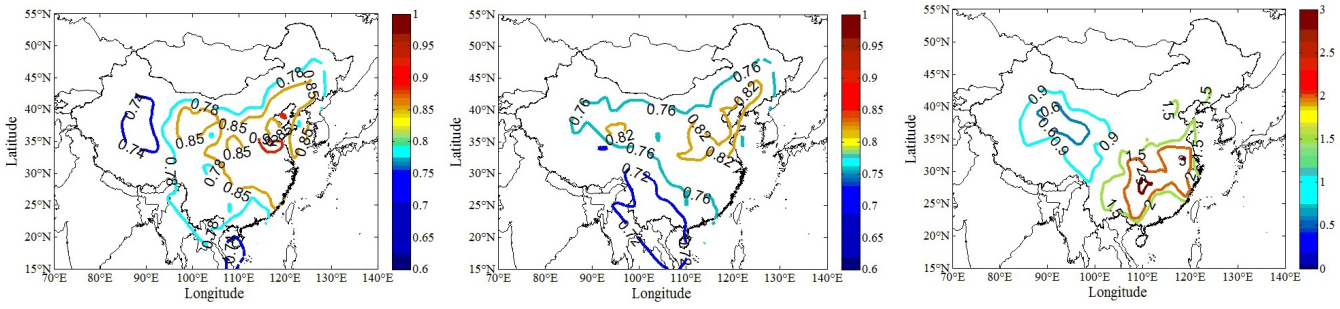

Fig. 11. DRCP forecast accuracy distribution contour: (a) TS_1, (b) TS_2, (c) *RMSE*.

the magnitude of rainfall but also can effectively filter the situation of no rainfall.

5.4.2 Accuracy

For RQ2, to clearly depict the accuracy of DRCF, three contour maps correspond to the distribution of three accuracy rates are shown in Fig. 11. Fig. 11a is the distribution of TS_1. TS_1 in eastern China is the highest, and their values generally reach 0.85. The accuracy rate of areas around Beijing can reach 0.92. TS_1 of areas in central China is generally near 0.78. TS_1 in the western region and Hainan province is the smallest, and their value is only about 0.75. In China, east areas are dominated by plain, and the economy is relatively developed. However, there are many mountains in the west, and the economy is relatively not developed. Therefore, there are more meteorological observation stations in the east than in the west, and there are more meteorological observation stations around every forecasting area in the east. This is the reason why $TS1$ in the east is higher than that in the west. Fig. 11b is the distribution of TS_2. TS_2 in eastern, northeastern and central of China is generally over 0.8. TS_2 in the southwest is relatively low, and their values are generally between 0.72 and 0.76. TS_2 is the forecast accuracy rate when the area is raining. The rainfall in southwest China is more frequent. There are more rainstorms in this area, and it is easy to produce unstable rainfall due to the influence of the typhoon from the Pacific Ocean. These reasons make it difficult to forecast rainfall in Southwest China. The most probable reason is that DRCF may need more physical factors, such as typhoon wind speed and typhoon location. Fig. 11c is the distribution of *RMSE*. *RMSE* in east areas is higher than that in west areas. This does not mean that the forecasting ability in west areas is stronger than that in the east are. In China, the frequency of rainfall is higher in east areas, especially in the Southeast, and the frequency of rainfall in west areas, especially in the Northwest, is very low. Strong rainfall usually occurs in the eastern and southern areas, which explains the distribution characteristics of *RMSE*.

5.4.3 Comparison

For RQ3, we first compare the forecasting accuracy rate among six machine learning methods: ARIMA, SVM, BPNN, RBFNN, DBN, and MLP. Fig. 12a gives the *RMSE* distribution of the six approaches. The horizontal axis in the chart represents 56 forecasting areas, and the longitudinal axis represents the *RMSE*of a certain area. In contrast, it can be found that the *RMSE* of ARIMA, RBFNN, and SVM is the largest and the *RMSE* of MLP and DBN is the smallest. Similarly, Fig. 12b gives the TS_1 distribution of the six approaches. It can be found that the TS_1 of ARIMA and SVM is the smallest and the TS_1 of other approaches is not significantly different from each other. Fig 12c gives the TS_2 distribution of six methods. From the figure we can see that the TS_2 of ARIMA and SVM changes drastically. The TS_2 of BPNN and MLP is higher than DBN and RBFNN. To sum up, the forecasting ability of MLP is obviously better than SVM and ARIMA, and better than RBFNN, slightly better than DBN and BPNN. Table 3 is the comparison of three accuracy rates between six machine learning approaches, two numerical models, and DRCF. The values of TS_1 and TS_2 in the table represent the average of 56 areas. It can be found that MLP can achieve better prediction results among the six machine learning approaches. Furthermore, the average *RMSE* of DRCF is the smallest one among all the approaches.

TABLE 3
A comparison of three accuracy rates between DRCF and other approaches

Model	TS_1	TS_2	$RMSE$
RBFNN	0.86	0.37	16.42
BPNN	0.87	0.43	3.61
MLP	0.87	0.46	2.73
DBN	0.86	0.41	2.58
ARIMA	0.70	0.38	31.71
SVM	0.72	0.40	20.16
ECMWF	0.76	0.76	2.30
JAPAN	0.69	0.73	3.10
DRCF	**0.86**	**0.78**	**1.61**

Next, we compare the forecasting accuracy of DRCF, MLP and two atmospheric models. The forecasting interval of them is 3 hours. This provides convenience for the comparison of accuracy. The data of two atmospheric models are released by corresponding countries. The forecasting results are all lattice data. The forecasting data of each area is extracted directly from the text file according to latitude and longitude. Fig. 13a gives the *RMSE* distribution of MLP, DRCF and two atmospheric models. Fig. 13b gives the TS_1 distribution. Fig. 13c gives the TS_2 distribution. Similar to previous ones, the forecasting ability of DRCF

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2018.2871151, IEEE Transactions on Big Data

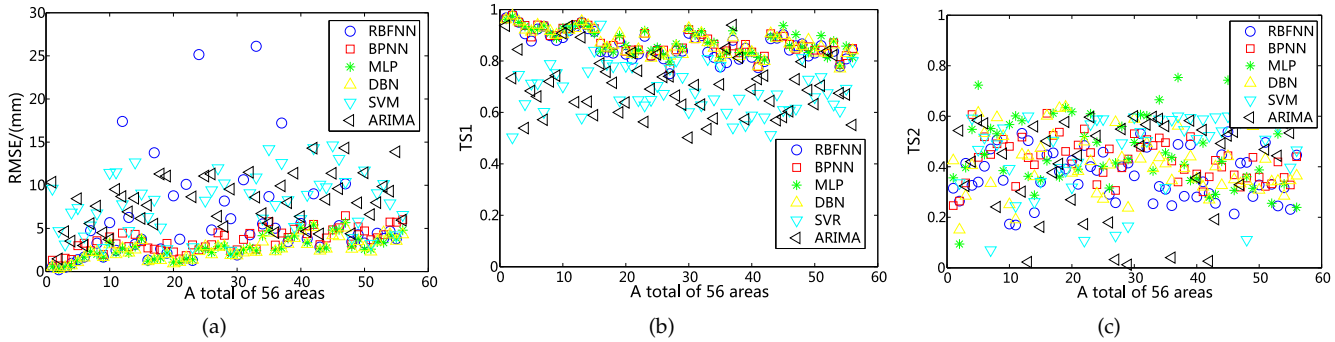

Fig. 12. Comparison results for six machine learning approaches: (a) *RMSE*, (b) TS_1, (c) TS_2.

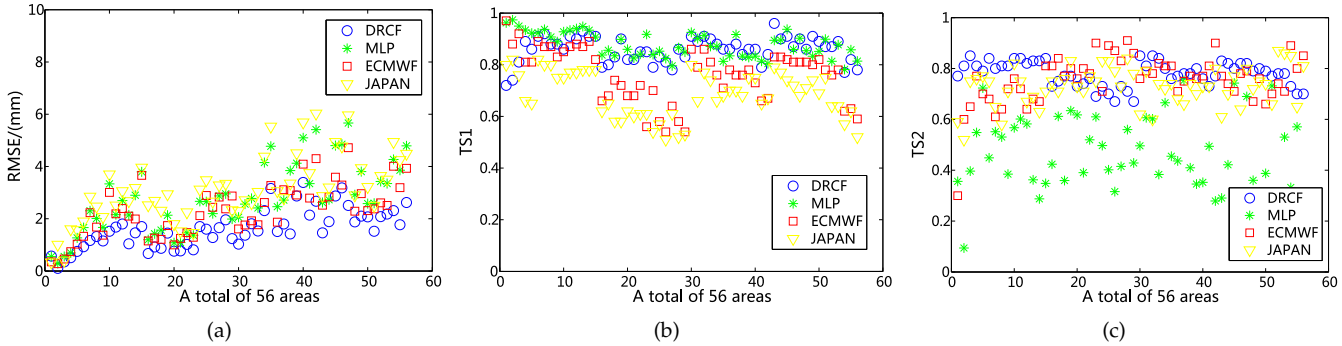

Fig. 13. Comparison results for MLP, DRCF, ECMWF and JAPAN: (a) *RMSE*, (b) TS_1, (c) TS_2.

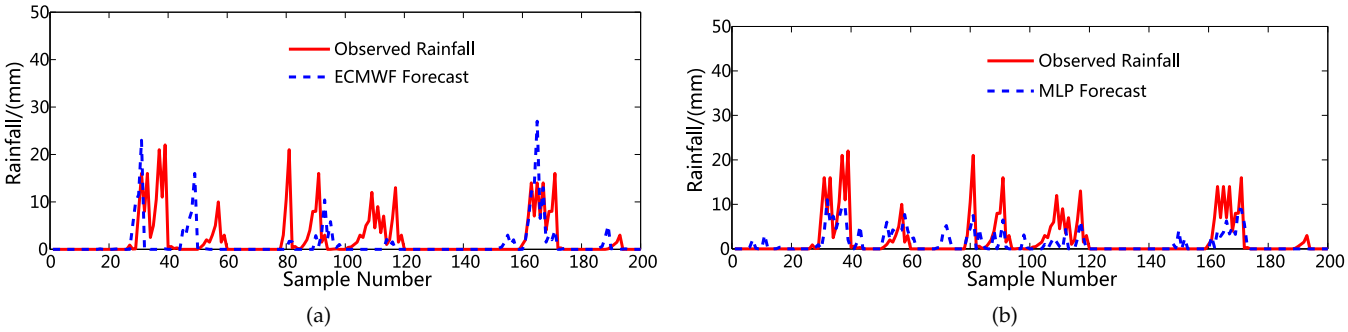

Fig. 14. Rainfall forecasting results for *Fuyang* in August 2018: (a) using ECMWF, (b) using MLP

is obviously higher than that of MLP. At the same time, the forecasting accuracy rate of DRCF is higher than two atmospheric models and MLP. The average *RMSE* of two atmospheric models are also shown in Table 3. After two kinds of comparisons, the forecasting ability of DRCF is quite convincing.

Example 5. One machine learning approach (MLP) and one atmospheric model (ECMWF) are used to compare with DRCF. Fig. 14a and Fig. 14b are forecasting results for *Fuyang* in August 2018 using ECMWF and MLP, respectively. From these two figures, we can compare the forecasting results of MLP and ECMWF. ECMWF can predict the overall rainfall process but cannot predict rainfall volume and duration. MLP could not predict larger rainfall, and it would appear many empty reports. In general, the overall forecasting performance of DRCF outperforms that of MLP and ECMWF.

5.5 Discussions

Here we first measure the time consumption of DCRF. Second, we discuss the barriers and good prospects for the official application of machine learning approaches.

Time consumption is also an important criterion for evaluating a model. For DRCF, the time consumed is divided into two parts: *training time* and *prediction time*. Training time is the time spent to train the model and its corresponding parameters. For the sample data of two years, the training time of DRCF usually takes about half an hour. However, the whole training process can be performed offline. When the actual physical quantities of surrounding areas are received, the prediction time could be completed within several minutes. Consequently, the prediction time is very short which is less than forecasting interval (3 hours).

Currently, atmospheric methods are still the main guide for official forecasting in China. Some meteorological bu-

reaus try to use machine learning approaches as references or research purposes. The main barriers for this phenomenon are described as follows:

- *Lack of knowledge*. Since atmospheric approaches have been widely used in official forecasting, most meteorological officials are unfamiliar with machine learning approaches and also unwilling to adopt them.
- *Few successful experiences*. The existing machine learning approaches are only used in one certain area. Few successful experiences of widely used examples have been achieved yet.

However, we still believe that machine learning approaches have certain advantages and good prospects. We summarized in the following:

- *Less cost*. Atmospheric models, including numerical forecasts, require large hardware resources. They cannot achieve very high prediction accuracy in all the regions. On the contrary, machine learning approaches usually requires much fewer resources, and the generalization is better.
- *Large amounts of data*. The number of observed meteorological data is increasing. Large amounts of data provide a nice base for machine learning approaches.
- *Technology progress*. Deep learning (DL) algorithms are developing rapidly. More efficient DL algorithms have been proposed, which may improve the overall prediction ability of machine learning approaches.

6 Conclusions and Future Work

To solve the limitations of existing short-term rainfall forecasting approaches, a novel model called DRCF is proposed based on MLP. The experimental results show that the overall forecasting performance of DRCF outperforms state-of-the-art approaches.

Future work focuses on the following aspects. First, the forecasting interval of DRCF is limited on 3 hours. Longer forecasting interval is further needed. Second, the greedy algorithm can only decide the optimal structure of local MLP, but cannot get the global optimal structure. The selection of MLP structures is still worth studying.

Acknowledgments

The work is supported by the National Key R&D Program of China under Grant No. 2018YFC0407901, the Fundamental Research Funds for the Central Universities under Grant No. 2018B16014, the National Natural Science Foundation of China under Grant Nos. 61572171, 61761136003, and the Natural Science Foundation of Jiangsu Province under Grant No. BK20171427.

References

[1] M. D. Crown, "Validation of the NOAA space weather prediction center's solar flare forecasting look-up table and forecaster-issued probabilities," *Space Weather*, vol. 10, no. 6, pp. 1–4, 2012.

[2] K. R. Moran, G. Fairchild, N. Generous, K. Hickmann, D. Osthus, R. Priedhorsky, J. Hyman, and S. Y. Del Valle, "Epidemic forecasting is messier than weather forecasting: The role of human behavior and internet data streams in epidemic forecast," *The Journal of infectious diseases*, vol. 214, no. suppl_4, pp. S404–S408, 2016.

[3] C. W. Zheng, C. Y. Li, X. Chen, and J. Pan, "Numerical forecasting experiment of the wave energy resource in the China sea," vol. 2016, no. 3, pp. 1–12, 2016.

[4] K. C. Luk, J. E. Ball, and A. Sharma, "An application of artificial neural networks for rainfall forecasting," *Mathematical Computer Modelling*, vol. 33, no. 6, pp. 683–693, 2001.

[5] K. W. Appel, R. C. Gilliam, N. Davis, A. Zubrow, and S. C. Howard, "Overview of the atmospheric model evaluation tool (MET) v1.1 for evaluating meteorological and air quality models," *Environmental Modelling & Software*, vol. 26, no. 4, pp. 434–443, 2011.

[6] N. Bartoletti, F. Casagli, S. Marsili-Libelli, A. Nardi, and L. Palandri, "Data-driven rainfall/runoff modelling based on a neuro-fuzzy inference system," *Environmental Modelling & Software*, vol. 106, pp. 35–47, 2018.

[7] L. Cozzi, "Weather models as virtual sensors to data driven rainfall predictions in urban watersheds," *EGU General Assembly Conference Abstracts*, vol. 15, pp. 111–115, 2013.

[8] K. Sudheer, A. Gosain, and K. Ramasastri, "A data-driven algorithm for constructing artificial neural network rainfall-runoff models," *Hydrological processes*, vol. 16, no. 6, pp. 1325–1330, 2002.

[9] R. J. Schalkoff, *Artificial neural networks*, vol. 1. McGraw-Hill New York, 1997.

[10] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," vol. 9, pp. 249–256, 2010.

[11] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[12] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pp. 3642–3649, IEEE, 2012.

[13] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, ACM, 2008.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[15] S. Ergezinger and E. Thomsen, "An accelerated learning algorithm for multilayer perceptrons: optimization layer by layer," *IEEE Transactions on neural networks*, vol. 6, no. 1, pp. 31–42, 1995.

[16] R. Salakhutdinov and H. Larochelle, "Efficient learning of deep boltzmann machines," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 693–700, 2010.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] G. Hepner, T. Logan, N. Ritter, and N. Bryant, "Artificial neural network classification using a minimal training set- comparison to conventional supervised classification," *Photogrammetric Engineering and Remote Sensing*, vol. 56, no. 4, pp. 469–473, 1990.

[19] E. Hernández, V. Sanchez-Anguix, V. Julian, J. Palanca, and N. Duque, "Rainfall prediction: A deep learning approach," in *International Conference on Hybrid Artificial Intelligence Systems*, pp. 151–162, Springer, 2016.

[20] S. Gope, S. Sarkar, P. Mitra, and S. Ghosh, "Early prediction of extreme rainfall events: a deep learning approach," in *Industrial Conference on Data Mining*, pp. 154–167, Springer, 2016.

[21] P. Zhang, L. Zhang, H. Leung, and J. Wang, "A deep-learning based precipitation forecasting approach using multiple environmental factors," in *Big Data (BigData Congress), 2017 IEEE International Congress on*, pp. 193–200, IEEE, 2017.

[22] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," in *Advances in Neural Information Processing Systems*, pp. 5622–5632, 2017.

[23] J. Boehm, B. Werl, and H. Schuh, "Troposphere mapping functions for GPS and very long baseline interferometry from European centre for medium-range weather forecasts operational analysis data," *Journal of Geophysical Research: Solid Earth*, vol. 111, no. B2, 2006.

[24] Y. Honda, M. Nishijima, K. Koizumi, Y. Ohta, K. Tamiya, T. Kawabata, and T. Tsuyuki, "A pre-operational variational data assimilation system for a non-hydrostatic model at the Japan meteorological agency: Formulation and preliminary results," *Quarterly*

Journal of the Royal Meteorological Society, vol. 131, no. 613, pp. 3465–3475, 2005.
[25] M. Xue, K. K. Droegemeier, and V. Wong, "The advanced regional prediction system (ARPS)–a multi-scale nonhydrostatic atmospheric simulation and prediction model. part i: Model dynamics and verification," *Meteorology and atmospheric physics*, vol. 75, no. 3-4, pp. 161–193, 2000.
[26] P. Chan and K. Hon, "Performance of super high resolution numerical weather prediction model in forecasting terrain-disrupted airflow at the Hong Kong international airport: case studies," *Meteorological Applications*, vol. 23, no. 1, pp. 101–114, 2016.
[27] W. Yu, E. Nakakita, S. Kim, and K. Yamaguchi, "Assessment of ensemble flood forecasting with numerical weather prediction by considering spatial shift of rainfall fields," *KSCE Journal of Civil Engineering*, pp. 1–11, 2018.
[28] P. Narayanan, A. Basistha, S. Sarkar, and S. Kamna, "Trend analysis and arima modelling of pre-monsoon rainfall data for western India," *Comptes rendus geoscience*, vol. 345, no. 1, pp. 22–27, 2013.
[29] J. Farajzadeh, A. F. Fard, and S. Lotfi, "Modeling of monthly rainfall and runoff of Urmia lake basin using feed-forward neural network and time series analysis model," *Water Resources and Industry*, vol. 7, pp. 38–48, 2014.
[30] P.-S. Yu, C.-J. Chen, and S.-J. Chen, "Application of gray and fuzzy methods for rainfall forecasting," *Journal of Hydrologic Engineering*, vol. 5, no. 4, pp. 339–345, 2000.
[31] J. Abbot and J. Marohasy, "Application of artificial neural networks to rainfall forecasting in Queensland, Australia," *Advances in Atmospheric Sciences*, vol. 29, no. 4, pp. 717–730, 2012.
[32] J. Abbot and J. Marohasy, "Input selection and optimisation for monthly rainfall forecasting in Queensland, Australia, using artificial neural networks," *Atmospheric Research*, vol. 138, pp. 166–178, 2014.
[33] Z. Wu, B. Wang, J. Li, and F.-F. Jin, "An empirical seasonal prediction model of the East Asian summer monsoon using ENSO and NAO," *Journal of Geophysical Research: Atmospheres*, vol. 114, no. D18, 2009.
[34] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.
[35] G.-F. Lin and M.-C. Wu, "A hybrid neural network model for typhoon-rainfall forecasting," *Journal of Hydrology*, vol. 375, no. 3-4, pp. 450–458, 2009.
[36] P. Zhang, Y. Jia, L. Zhang, J. Gao, and H. Leung, "A deep belief network based precipitation forecast approach using multiple environmental factors," *Intelligent Data Analysis*, vol. 22, no. 4, pp. 843–866, 2018.
[37] F. Mekanik, M. Imteaz, S. Gato-Trinidad, and A. Elmahdi, "Multiple regression and artificial neural network for long-term rainfall forecasting using large scale climate modes," *Journal of Hydrology*, vol. 503, pp. 11–21, 2013.
[38] S. Ramanujam, R. Chandrasekar, and B. Chakravarthy, "A new pca-ann algorithm for retrieval of rainfall structure in a precipitating atmosphere," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 21, no. 8, pp. 1002–1025, 2011.
[39] A. George, "Anomaly detection based on machine learning: dimensionality reduction using PCA and classification using SVM," *International Journal of Computer Applications*, vol. 47, no. 21, 2012.
[40] T. Shimobaba, N. Kuwata, M. Homma, T. Takahashi, Y. Nagahama, M. Sano, S. Hasegawa, R. Hirayama, T. Kakue, A. Shiraki, *et al.*, "Deep-learning-based data page classification for holographic memory," *arXiv preprint arXiv:1707.00684*, 2017.
[41] D. Imseng, P. Motlicek, P. N. Garner, and H. Bourlard, "Impact of deep mlp architecture on different acoustic modeling techniques for under-resourced speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pp. 332–337, IEEE, 2013.
[42] S. Lange, K. Gehmlich, A. S. Lun, J. Blondelle, C. Hooper, N. D. Dalton, E. A. Alvarez, X. Zhang, M.-L. Bang, Y. A. Abassi, *et al.*, "Mlp and carp are linked to chronic pkcα signalling in dilated cardiomyopathy," *Nature communications*, vol. 7, p. 12120, 2016.
[43] T.-Y. Kwok and D.-Y. Yeung, "Constructive algorithms for structure learning in feedforward neural networks for regression problems," *IEEE transactions on neural networks*, vol. 8, no. 3, pp. 630–645, 1997.

Pengcheng Zhang received the Ph.D. degree in computer science from Southeast University in 2010. He is currently an associate professor in College of Computer and Information, Hohai University, Nanjing, China, and was a visiting scholar at San Jose State University, USA. His research interests include water Informatics, software engineering, service computing, and data mining. He co-authored more than 70 peer-reviewed conference and journal papers, and has served as technical program committee member on various international conferences.

Yangyang Jia is a master student in College of Computer and Information, Hohai University, Nanjing, China. He received his bachelor degree in atmospheric science from Nanjing University, Nanjing, China in 2013. His current research interests include water Informatics and data mining.

Jerry Gao is a professor at the Department of Computer Engineering at San Jose State University. He has published over hundreds (180) publications in IEEE/ACM journals, magazines, International conferences and workshops. His current research areas include cloud computing, TaaS, software engineering, mobile computing and big data.

Wei Song received the Ph.D. degree from Nanjing University, China, in 2010. He is currently an associate professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China, and was a visiting scholar at Technische Universität München, Germany. His research interests include data sciecne, software engineering, services and cloud computing, program analysis, middleware, and process mining. He was invited to the Schloss Dagstuhl Seminar "Integrating Process-Oriented and Event-Based Systems" held in August 2016. He is a member of the IEEE.

Hareton Leung is a director of the Lab for Software Development and Management. He serves on the Editorial Board of Software Quality Journal. He is a fellow of Hong Kong Computer Society, chairperson of its Quality Management Division (QMSID) and chairperson of HKSPIN. He is also an accomplished industry consultant, giving advice on software testing, quality assurance, process and quality improvement, system development, and providing expert witness and litigation support.