IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 11, 2018, accepted December 26, 2018, date of publication January 1, 2019, date of current version January 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2890339

Parallel Processing of Probabilistic Models-Based Power Supply Unit Mid-Term Load Forecasting With Apache Spark

WEI JIANG[1], (Member, IEEE), HAIBO TANG[1], LEI WU[1], HE HUANG[2], AND HUI QI[2]
[1]School of Electrical Engineering, Southeast University, Nanjing 210096, China
[2]State Grid Jiangsu Electric Power Company, Nanjing 210024, China

Corresponding author: Wei Jiang (jiangwei@seu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51877041 and in part by the Science and Technology Foundation of State Grid Jiangsu Electric Power Company, China, under Grant J2018088.

ABSTRACT Mid-term load forecasting (MTLF) of power supply unit (PSU) is an essential part of refined distribution network planning. By analyzing a large amount of historical data accumulated in electric automation systems, accurate MTLF result can be obtained with the help of big-data and parallel computing technology. In this paper, a dynamic bayes network (DBN)-based MTLF model is proposed to forecast the peak power load of next year of each PSU. In the first stage, we improve the accuracy of MTLF model by using dynamic radius DBSCAN algorithm to determine the optimal state division. In the second stage, to improve the computation efficiency, the calculations of multiple probability matrixes and the modified forward algorithm are implemented on an apache spark-based parallel computing platform. The experiment results indicate that the parallel processing of DBN-based MTLF model has superior performance in accuracy, efficiency, and versatility.

INDEX TERMS Dynamic bayes network, mid-term load forecasting, Apache Spark, parallel computing.

I. INTRODUCTION

State Grid Corporation of China has promoted, in Jiangsu Province, distribution network planning, whose approaches are based on power supply unit (PSU). As the essence of refined planning, these approaches divide the city into hundreds of PSU, and therefore the weakness of existed distribution networks and the power load development trends can be evaluated unit by unit. Meanwhile, more accurate and economical planning distribution network will come with more detailed information. In addition, mid-term load forecasting (MTLF) results are important to PSU-based planning, given that they represent the future power demand geographically and determine the necessity of upgrade projects [1]–[2].

Traditional regional load forecasting technologies focus on utilizing historical load profile and other factors such as weather, economic indicators and energy policies to predict the load electricity demand of next days or months. The MTLF methods can be classified into three main categories:

1) Time series approaches. These methods are more based on the internal trends of historical data in forecasting future load. The usage of time series methods such as linear regression, autoregressive integrated moving average (ARIMA), autoregressive moving average (ARMA) in MTLF are reported in [3]–[6].

2) Artificial intelligence (AI) based approaches. In [7], a clustering neural network consisting of logic operators is used in mid-long term load forecasting. A hybrid model based on dynamic and fuzzy time series for MTLF is proposed in [8] and proved to be superior to time series approach. Multiple ANN technologies are widely used because they do not require functional form of a forecasting model [9]–[11]. SVR is the application form of support vector machine (SVM), whose usage in load forecasting is reported in the literatures [12]–[15].

3) Conditional relationship approaches. The socio-economic factors in addition to other variables are considered in load forecasting in [16]. In [17], the non-linear influence of temperature on electricity demand is employed to generate robust dynamic patterns on the electricity demand-climate relationship. With no requirement of linearity and seasonality, conditional relationship approaches directly forecasting load by incorporating the related variables [18] and [19].

The main difference of PSU MTLF and regional MTLF resides in the fact that the forecasting range is restricted

in each PSU, typically including several residential districts or industrial parks. Limited user and the consequent significant load fluctuation. Whereas, the widely installed smart meters and the implement of advanced metering infrastructure (AMI) can make power consumption information collection system (PCICS) of State Grid Corporation of China (SGCC) store huge amounts of load data of each distribution transformer (DT). There are about 20000 DTs in a medium-sized city in Jiangsu Province, China. Multi-year and high-frequency annotated data are collected and stored in PCICS, which annually updates 5 Terabyte load data. Thus, big data analysis platform and parallel processing algorithm are needed to guarantee the forecasting efficiency. Meanwhile, utilizing these high-resolution data, the conditional relationship approaches based MTLF of each PSU is necessary [20].

In this paper, we focus on forecasting the peak load of each PSU in next year with the proposed Dynamic Bayes Network (DBN) model, which includes time series characteristic and conditional relationship. The rest of paper are organized as follows. Section 2 gives the description of observation variables. Section 3 explains MTLF model's construction and proposes a novel approach for state division of observation variables and forecasting variable. In Section

4, the large-scale historical data are employed to calculate the probability matrixes. The Apache Spark platform to improve the ability of parallel processing of large-scale data, and several algorithms to adapt the RDD based lazy computing. Experimental results and evaluations are shown in Section 5. Part 6 draws the conclusions.

II. DATASET AND FORECASTING SCENARIO

A. HISTORICAL LOAD DATA

The historical load dataset used in this work belongs to SGCC Jiangsu electric power company. Since the smart meters are generally deployed in recent years, data from 2015 to 2017 are selected to ensure the identical acquirement frequency.

As shown in Fig. 1, the original data from PCICS consists of the unique device identifier (UDID) of the DTs, indicator of phase sequence, failure flags and 96 points floats representing load sampling value for every 15 minutes. Possible communication failure, data collecting equipment failure and backup failure will generate null, zero and abnormal values in the raw dataset. Thus, the analysis should come after executing the data cleansing process, which normally contains removal of null and zero values, data completion with Lagrange interpolation formula and elimination of duplicated rows.

B. HISTORICAL WEATHER DATA

In this paper, we assume that the extreme weather conditions are related to peak load. Therefore, critical days when peak load or extreme weather occur are included in the forecasting model, which will be introduced in section III. The heating degree (HD) and cooling degree (CD) is employed to represent the linear relation between load and temperature. With temperature variables $T^C_{ref} = 26°C$ and $T^{\mathrm{H}}_{\mathrm{ref}} = 18°$, there are

$$HD\,(d) = \max[T^H_{ref} - T\,(d)\,, 0] \tag{1}$$

$$CD\,(d) = \max[0, T\,(d) - T^C_{ref}] \tag{2}$$

In the proposed forecasting algorithm, the critical days are selected for each PSU based on three standards:

1) The two days when peak load occurs in summer (June to September) and winter (December to February).

2) The two days when highest and lowest temperature occur.

3) Manually selected days when major events of weather extremes occur.

It should be noticed the defined peak load is the peak of loads summation of all DTs in one PSU. There is

$$L_{psu\_i} = \sum_{i=1}^{m} L_m \tag{3}$$

where $L_{\mathrm{psu\_i}}$ is the summation of all DT load in the i^{th} PSU.

There are dozens of weather stations installed by SGCC in each city. These stations collect daily humidity, pressure, wind speed and temperature. Besides temperature, relative humidity is also an important parameter in load forecasting [19]. Therefore, in the present analysis we used daily average outdoor temperature taking HD, CD and humidity (%) as the basic meteorological parameters. The data from the closest weather station are selected for every PSU.

C. WEEKDAY TYPE

The weekday is transformed from the date field in Fig. 1 with Zeller formula. The weekday is coded as $0 \sim 6$ and 0 indicates Sunday. The public holidays are modified to Saturday,

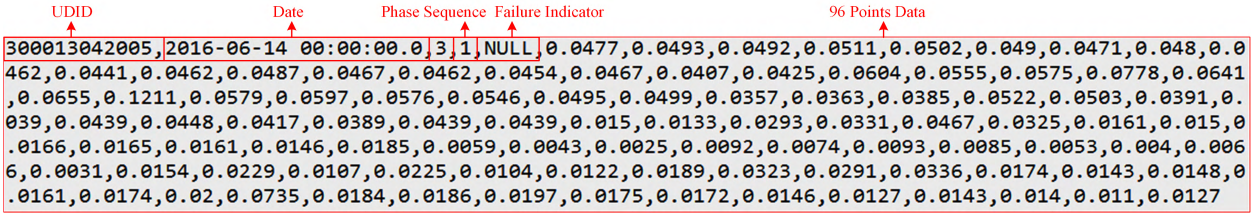

FIGURE 1. One row of raw load data from PCICS.

including Chinese New Year, National Day and Labor Day etc.

D. GEOGRAPHIC COORDINATES

The power production system of SGCC stored the geographic coordinates of each DT, whose functions in our forecasting are threefold:

1) With geographic coordinates, the DTs can be related to particular PSU with its margin information.

2) The distances between DTs can be calculated with geographic coordinates by Mercator projection and are used as another clustering factor of DTs.

3) The final load forecasting results have to be shown in a heat map. The geographic coordinates of DTs are used for renderer.

III. DYNAMIC BAYES NETWORK BASED FORECASTING MODEL

A. THE FORECASTING MODEL

In the MTLF scenario of this work, the load of more than 10000 DTs need to be forecasted at the same time. Meanwhile, large though the whole dataset is, the historical data profiles of each DT at the selected critical days are very limited. Based on above facts, an ideal model for PSU MTLF requires

1) The time series of historical load profile and condition relationship between observation variables and forecasting variable are both considered.

2) The parameters of the model are pre-calculated to accelerate the forecasting of large volume of DTs.

3) The structure of the forecasting model of single DT should be simple. Meanwhile, global dataset is employed to enhance the forecasting accuracy of individual DT.

4) The model can be realized efficiently on parallel computing platform.

To meet the requirements, a Dynamic Bayes Network (DBN) based MTLF model is proposed, which is the extension of BN on the time dimension [21]. The observation variable in our model refers to $Y = \{HD, CD, W, H\}$ which contains heating degree HD, cooling degree CD, weekday type W and humidity H. The forecasting variable $X = \{L\}$ is the load of DT at a certain time.

The target of the BN model is to reckon the probability distribution of forecasting variable X with certain observation variable Y. The probability of forecasting variable at certain state c is

$$P(X_t = c|HD_t, CD_t, W_t, H_t) = \frac{P(HD_t, CD_t, W_t, H_t|X_t = c) \cdot P(X_t = c)}{P(HD_t, CD_t, W_t, H_t)} \quad (4)$$

where X_t represents the load at discrete time t, W_t, T_t, H_t are observation variables at the same time.

The denominator of (4) can be further derived with total probability formula

$$P(HD_t, CD_t, W_t, H_t) = \sum_{d=1}^{v} P(HD_t, CD_t, W_t, H_t|X_t = d) \cdot P(X_t = d) \quad (5)$$

With (4) and (5), we have (6), as shown at the bottom of this page, where $P(HD_t|X_t = d)$, $P(CD_t|X_t = d)$, $P(W_t|X_t = d)$ and $P(H_t|X_t = d)$ are conditional probabilities of HD, CD, H and W when forecasting variable X is given. $P(X_t = d)$ is the priori probability of forecasting variable.

The BN based forecasting model only considers the causal relationship between observation variables and forecasting variables. The development of load between years also follows certain pattern, and therefore we can take the timing characteristics of forecasting variable into consideration by linking different BN models and building a DBN based forecasting model. The transition processes between loads in different years are assumed to be Markovian and causal [22]. Hence, there is

$$P(X_{t+1}|X_t, X_{t-1}, \ldots, X_1) = P(X_{t+1}|X_t) \quad (7)$$

where $P(X_{t+1}|X_t)$ is the transition probability of forecasting variable.

In this case, the three-node DBN based MTLF model is proposed, as shown in Fig.2*b*. The formation process of the model is shown in Fig. 3:

1) The critical dates of the PSU are selected according to the standards described in section II. B. Take the day when summer load peak occurs for example. If we want to forecast the peak load of 2018, the observation variable on that day of 2017 Y_{sp_2017} is first acquired from historical data:

$$Y_{sp\_2017} = \{HD = 0, CD = 12, W = Tuesday, H = 90\%\} \quad (8)$$

2) The absolute values of Y are converted to respective states. The conversion principles will be discussed in the next section. The three nodes in DBN model share structure and observation variables in order to investigate the load development trend with identical condition.

3) The three BN models $BN_y^{PSU_i}$, $BN_{y-1}^{PSU_i}$ and $BN_{y-2}^{PSU_i}$ represent the casual relationships between Y and X in the forecasting year and prior two years. The links between the models represent the development probability of load in subsequent years.

4) As shown in Fig.2*b*, there are three kinds of probability matrixes to be calculated to realize the forecasting process.

$$P(X_t = c|HD_t, CD_t, W_t, H_t) = \frac{P(HD_t|X_t = c) \cdot P(CD_t|X_t = c) \cdot P(W_t|X_t = c) \cdot P(H_t|X_t = c) \cdot P(X_t = c)}{\sum_{d=1}^{v} P(HD_t|X_t = d) \cdot P(CD_t|X_t = d) \cdot P(W_t|X_t = d) \cdot P(H_t|X_t = d) \cdot P(X_t = d)} \quad (6)$$

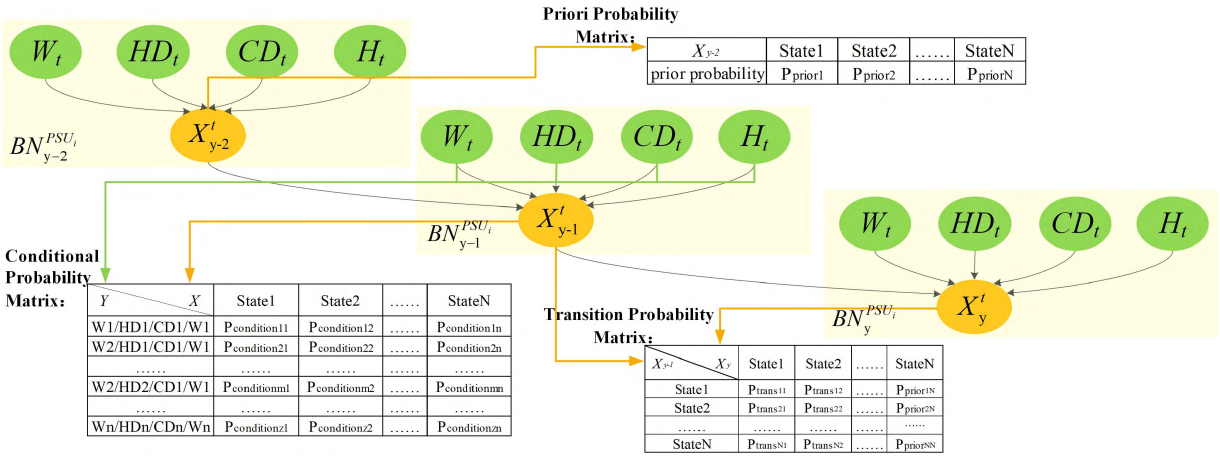

FIGURE 2. DAG of DBN based load forecasting model with multiple time slices.

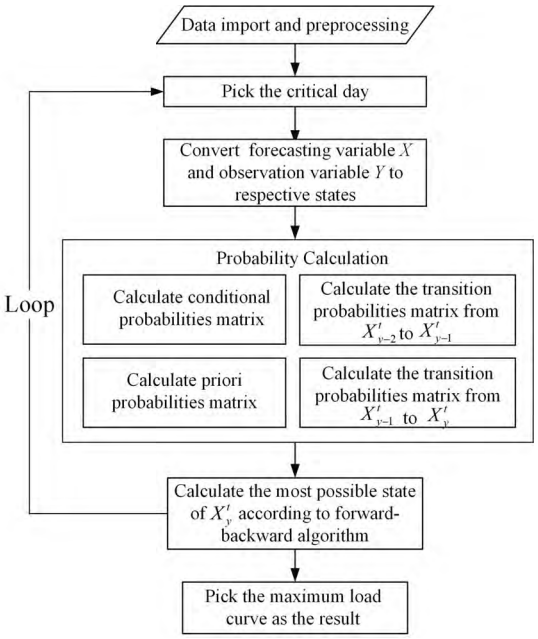

FIGURE 3. MTLF process based on DBN model.

First, the priori probability matrixes refer to the probability distributions of forecasting variables in different years. The conditional probability matrixes represent the probability distributions of forecasting variable with certain observation variables in different years.

The transition matrixes consist of state transfer probability of forecasting variable from year t to time $t+1$. All the three probability matrixes can be calculated from large-scale historical data by parallel processing.

5) With the DBN model, the probability matrixes, the distribution of load state of PSU_i on summer peak load day of the forecasting year can be derived with modified forward algorithm. The state with maximum probability $S^{PSU_i}_{sp\_y}$ is selected to represent the forecasting result on the summer peak load day.

6) All the distributions of load states on other critical days of PSU_i can be calculated with same process. The state with maximum value will be selected as the final MTLF result of that PSU:

$$S^{PSU_i}_{s\_y} = \max(S^{PSU_i}_{sp_y}, S^{PSU_i}_{wp_y}, S^{PSU_i}_{ht_y}, S^{PSU_i}_{lt_y} \ldots) \tag{9}$$

where $S^{PSU_i}_{sp_y}$, $S^{PSU_i}_{wp_y}$, $S^{PSU_i}_{ht_y}$ and $S^{PSU_i}_{lt_y}$ indicate the maximum load states on the four critical days we introduced.

The proposed three-node DNB model reflects the impact of observation variables on forecasting variables and the load development trend from year to year of each PSU. Compared to traditional multiple regression models, the proposed model have following advantages:

1) The probability of forecasting variable's state, rather than definite value is forecasted. Therefore, the computing scale of the forecasting algorithm is reduced.

2) The conditional and transition probabilities can be pre-calculated and reused in MTLF of different PSUs. Therefore, the real-time computing of a large number of regression coefficients can be avoided.

3) The model structure is fixed and simple. The parallel processing of the forecasting algorithm on Apache Spark platform is efficient.

B. VARIABLES STATE DIVISION WITH DYNAMIC RADIUS DBSCAN

The observation and forecasting variables of the DBN forecasting model are probability based, which means that the states of the variables need to be determined in advance. The division of variable states is a trade-off between precision and computing speed. If the ranges of states are too wide, the accuracy will be limited, if too narrow, the computing scale will increase exponentially. An efficient state division algorithm should divide the states of different variables asymmetrically according to the density of historical data. In order to obtain the asymmetric optimal division results, this paper

proposes a data driven variable state division approach which is based on density-based clustering algorithm (DBSCAN).

DBSCAN algorithm is a density-based clustering algorithm which considers a cluster as a region with high point density. The point density is defined as the number of points in the region around a specific point with pre-set radius. The points with a density above the threshold are treated as a cluster.

There are three basic steps of DBSCAN:

Step1: Select a point p from dataset arbitrarily;

Step2: Calculate the distance between p to all points within the radius;

Step3: If the number of the points in radius meets the requirement, a cluster is found.

In our model, the point is defined as the vector of observation/forecasting variables of a specific day. Take forecasting variable load as example, the point p on day d is

$$p_d = \{p_d^{t_1}, p_d^{t_2}, \ldots, p_d^{t_{96}}\} \tag{10}$$

The Euclidean distance is applied to measure the distance between two points of the same variable category

$$dist\,(p_{d1}, p_{d2}) = \left\|p_{d1} - p_{d2}\right\|_2 = \sqrt{\sum_{u=1}^{96} \left|p_{d1}^{t_u} - p_{d2}^{t_u}\right|^2} \tag{11}$$

The principles to determine a cluster C include

$$\begin{cases} dist\,(p_{d1}, p_{d2}) \le Eps\,(p_{d1}, p_{d2} \in C) \\ num\_C \ge MinPts \end{cases} \tag{12}$$

where Eps is the pre-set radius of a cluster, num_C is the number of points that are density-reachable in cluster C and $MinPts$ is the minimum required points within Eps. Based on these two principles, the algorithm can distinguish the noise from load data and ensure the high density of each cluster.

However, possible high variations in density of training dataset will prevent DBSCAN from distinguishing amongst clusters [23], [24]. If the Eps is too large, the resolution of the densely distributed region in the historical data will be insufficient, if too little, the less distributed region can never be clustered. Meanwhile, the geographical information is also important to the clustering algorithm because neighboring DTs are more likely to share the similar operation and meteorological conditions. Taken together, a dynamic radius DBSCAN is proposed with following improvement:

1) The geographical distances between DTs are considered in clustering algorithm by introducing the maximum radius of geographical distance between DTs Eps_2.

2) If the whole dataset is processed with the Eps and some load vectors are still not clustered, the initial maximum radiuses will be added with increments.

The steps of dynamic radius DBSCAN are described in Algorithm 1.

The clustering result of a DT with 100 historical load vectors is shown in Fig. 4*a*. Fig. 4*b* denotes that the load vectors with same state share similar modulus $\|p\|$. The key

Algorithm 1 Algorithm 1. Improved DBSCAN Algorithm

Input:

Eps_1 : Maximum non-spatial distance value

Eps_2 : Maximum geographical coordinate distance value

$MinPts$: Minimum number of points within Eps_1 and Eps_2 distance

ΔEps_1, ΔEps_2: Increment of Eps_1 and Eps_2 in each iteration

Output:

Labels: State division results of load dataset

Load daily load dataset of all DTs $\boldsymbol{RDD_X}$ and
create an empty list ***Labels***
Repeat
  for unmarked point $\boldsymbol{p}$ in $\boldsymbol{RDD_X}$
    calculate Euclidean distance e_1 between $\boldsymbol{p}$ and $\boldsymbol{q}$(another unmarked point in $\boldsymbol{RDD_X}$)
    calculate geographical coordinate distance e_2 in $\boldsymbol{RDD_Y}$ between the two DTs
    if $e_1 \le Eps_1$ && $e_2 \le Eps_2$
      append $\boldsymbol{q}$ to the cluster
      update the scale of cluster: n = n+1
    end if
    if $n \ge MinPts$
    mark all points in this cluster with current cluster label: ***Lables***[points] = n
    end if
  end for
  $Eps_1 \leftarrow Eps_1 + \Delta Eps_1$
  $Eps_2 \leftarrow Eps_2 + \Delta Eps_2$
Until all the points are assigned to their cluster
Terminate

of variable states division is to distinguish the border between neighboring clusters. The standard to define the borders is proposed as follows:

$$Lower\ border_i = \begin{cases} \left\|p_{d1} - p_{d2}\right\| \ i = 1 \\ Upper\ border_{i-1} i = 2, 3, \ldots \end{cases} \tag{13}$$

$$Upper\ border_i = \begin{cases} \|p\|_{max} \ i = 6 \\ \|p\|_{max}^{i} + (\|p\|_{min}^{i+1} - \|p\|_{max}^{i}) \\ \qquad \cdot \eta i = 1, 2, 3, 4, 5 \end{cases} \tag{14}$$

where $Lower\ border_i$ and $Upper\ border_i$ are the lower and upper borders of cluster i respectively, $\|p\|_{max}^{i}$ is maximum modulus of forecasting variable in state i, $\|p\|_{min}^{i+1}$ is minimum modulus of forecasting variable in state i, η is the balance factor of the gap between two divided state, which is defined as:

$$\eta = \frac{\|p\|_{max}^{i} - \|p\|_{min}^{i}}{\|p\|_{max}^{i+1} - \|p\|_{min}^{i+1}} \tag{15}$$

The state division result of forecasting variable with above standard is shown in TABLE 1.

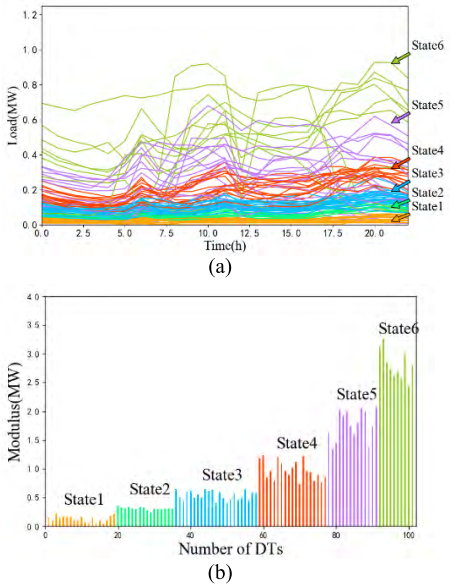

FIGURE 4. Clustering results of load data by different algorithm. *a* State division of load by improved DBSCAN. *b* Modulus of each state.

TABLE 1. Forecasting variable state division.

State	$\|p\|$ (MW)	State	$\|p\|$ (MW)
1	[0, 0.23]	4	[0.72,1.30]
2	[0.23,0.40]	5	[1.30,2.28]
3	[0.40,0.72]	6	[2.28,3.32]

C. PROBABILITY MATRIX

As described in previous section, the DBN model consists of priori, conditional and transition probability matrixes. The priori probability matrixes are determined by natural distribution of observation and forecasting variables. For example, the priori probability of X can be noted as

$$Priori_s = \frac{count(X = s)}{amount(X)} \tag{16}$$

where $count(X = s)$ is the count of occurrences when forecasting variable $X = s$, while $amount(X)$ is the total number.

$$\Lambda = P(HD_s^h : h \in XS, s \in HDS)$$

The conditional probabilities in Fig. 2 can be denoted as following

$$HD_s^h = P(HD = s|X = h) \tag{17}$$
$$CD_s^h = P(CD = s|X = h) \tag{18}$$
$$H_s^h = P(H = s|X = h) \tag{19}$$
$$W_s^h = P(W = s|X = h) \tag{20}$$

where HD_s^h, CD_s^h, H_s^h and W_s^h are the conditional probabilities of observation variables given the state of load $X = h$.

The conditional probabilities can be calculated from historical data with maximum likelihood estimation (MLE). For example, the MLE of heating degree given forecasting variable is the probability of the existence of HD_s^h. With given historical data set as D, if we defined the state set of heating degree and forecasting variable as HDSandXS, there is $HD_t \in HDS$ and $L_t \in XS\mathrm{L_t} \in LS$. The combined state Λ is defined as

$$\Lambda = P(HD_s^h : h \in XS, s \in HDS) \tag{21}$$

The MLE of Λ is

$$\Lambda^{MLE} = argmaxX(D, HD_s^h) \tag{22}$$
$$X(D, HD_s^h) = log\left[\prod_{i=1}^{n_h} P(HD_i = s|X_i = h)\right] \tag{23}$$

where n_h is the count of total time intervals of historical data.

By solving the limit of (17), we have

$$HD_s^h = \frac{count(s, h)}{count(h)} \tag{24}$$

where $count(s, h)$ is the count of occurrences when $HD_\mathrm{t} = s$ and $L_\mathrm{t} = h$ and $count(h)$ is when $L_\mathrm{t} = h$. (24) indicates that we can obtain the conditional probability by querying historical records with certain statements. The parallel query operations of large-scale historical data are suitable for big data analysis platform.

The conditional probability matrix is a $m \times v$ matrix where v is the number of states of forecasting variable's and m is the number of states of the observation variables. For example, the definition of heating degree conditional matrix HDM is

$$HDM = \begin{bmatrix} P(1|1) & \cdots & P(1|v) \\ \vdots & P(HD = s|X = h) & \vdots \\ P(q|1) & \cdots & P(q|v) \end{bmatrix} \tag{25}$$

In this work, the transition probabilities should reflect the load development trend between adjacent years. Therefore, the transition probability is defined as

$$TransM_k^h = P\left(X_{y\_d}^t = k|X_{y-1\_d}^t = h\right) = \frac{amount(k, h)}{amount(h)} \tag{26}$$

where $X_{y-1\_d}^t$ is the load at time t on day d in year y-1, e.g., at 9:45 2nd, August, 2016 while $X_{y\_d}^t$ is the load at the same time in year y, say, at 9:45 2nd, August, 2017 for this instance. $count(k, h)$ is the count of occurrence when $X_{y-1\_d}^t = h$ and $X_{y\_d}^t = k$ and $count(h)$ is when $X_{y\_d}^t = k$.

IV. IMPLEMENT OF DBN BASED MTLF ALGORITHM WITH APACHE SPARK

A. VARIABLES STATE DIVISION WITH DYNAMIC RADIUS DBSCAN

In this work, the Apache Spark platform is employed to fulfil the model training and forecasting tasks. As shown in Fig. 5, the Apache Spark based parallel processing platform is divided into three layers. The data source layer collects data from Electricity Information Collection System (EICS) and Road Weather Information System (RWIS). The raw datasets are imported into HDFS as data storage layer. In the parallel

data processing layer, Resilient Distributed Datasets (RDD) models are built on memory computing framework, which could save disk I/O operation time [25]. RDD is an abstract class defined in Apache Spark framework which includes *transformation* and *action* operators. *Transformation* operators realize the intermediate processing of the dataset while *action* operators trigger the submission of computing jobs on distributed computing cluster [26].

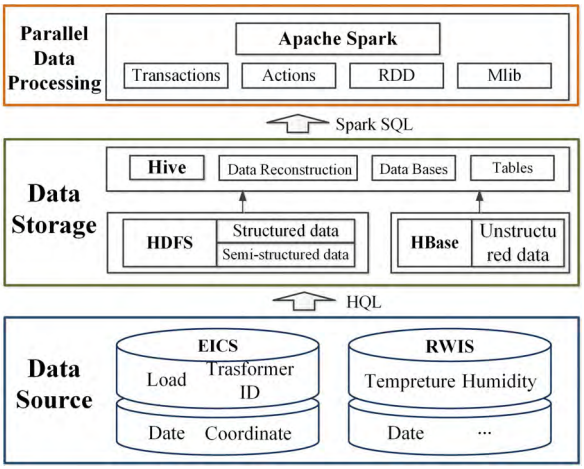

FIGURE 5. The Apache Spark based parallel processing platform.

Considering the merits of Spark, it is used as the data processing platform in our application to implement two main parallel computing tasks:

1) Parallel probability matrixes calculation. As discussed in section III. C. The elements of conditional and transition probability matrixes are calculated by filtering and counting large volume of historical data. The historical data are stored in DFS and the calculations are carried out on Spark.

2) Paralleled computation of modified forward forecasting algorithm. With DBN based load forecasting model and probability matrix, the load of thousands of PSU need to be parallel forecasted with modified forward algorithm, which is fulfilled on Spark to enhance the efficiency.

B. PARALLELIZATION OF PROBABILITY MATRIX COMPUTING ON SPARK

In the following experiments, the probability matrixes are calculated from 88841800 rows of historical load data across 3 years in advance. To handle large-scale model training task, the calculation needs to be accelerated by Apache Spark based parallel computing technology.

As shown in Fig. 6, the parallel probability matrixes calculation process consists of five steps:

1) The raw datasets are imported into Spark from HDFS as RDD objects. Each RDD object remains original data structure. For example, the load RDDs contain UDID of DTs and 96 points of daily load curves. The equipment RDDs contain UDID of DTs and their parameters, including geographical coordinates. Meteorological RDDs contain the historical temperature and humidity data of each PSU.

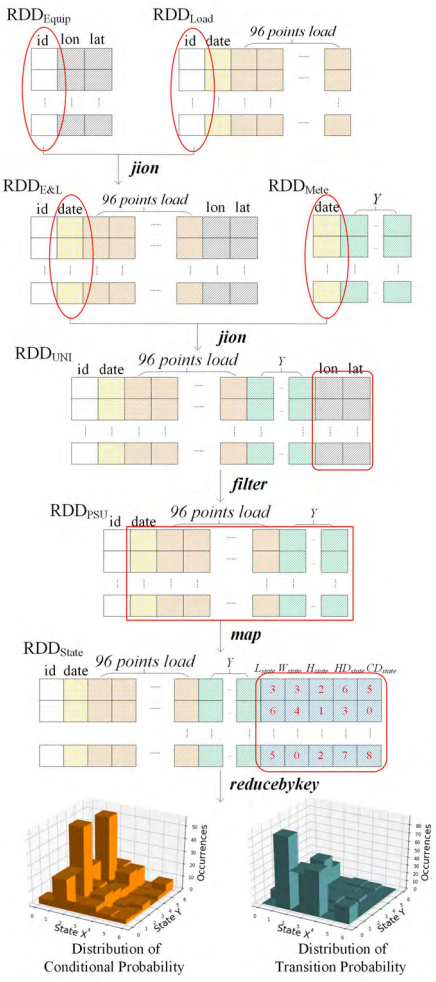

FIGURE 6. Process of calculating probabilities matrix based on Spark.

2) The load RDDs and meteorological RDDs both contains the attribute *date*. Meanwhile, the attribute UDID of DT is both included in the load RDDs and equipment RDDs. Avoiding querying data from multiple table can improve the efficiency of the matrix calculation, and this ask for utilizing the join transformation operator to generate a unified data structure RDD_{UNI}, which includes continuous time slices of load, temperature, humidity and date types of each DT joined with the geographical coordinates.

3) The *filter* transformation operator is applied to filter out the desired fields from $\mathrm{RDD_{UNI}}$ and form multiple RDD_{PSU} objects according to geographical coordinates of the DTs in different PSUs.

4) Each RDD_{PSU} is processed with *map* transformation operator to handle the load and weather data fields in data rows and to determine their state according to the pre-set state division scheme. This process generates RDD_{State} which contains the state division information.

5) Finally, the *reducebykey* action is implemented to count the number of occurrences of different observation variables Y and forecasting variable X. The Priori probability matrix, which refers to the probability distribution of X of

$$\alpha_t(i) = P(X_t = i|Y_1, Y_2, \cdots, Y_t) = \frac{P(X_t = i, Y_1, Y_2, \cdots, Y_t)}{P(Y_1, Y_2, \cdots, Y_t)} = \eta_n^t \cdot \sum_{j=1}^{n-1} \tau_{t-1}^{ji} \cdot P(X_{t-1} = j|Y_1, Y_2, \cdots, Y_{t-1}) \cdot P(Y_t|X_t = i)$$
$$= \eta_n^t \cdot \sum_{j=1}^{n-1} \tau_{t-1}^{ji} \cdot P(X_{t-1} = j|Y_1, Y_2, \cdots, Y_{t-1}) \cdot \prod_{v}^{W,H,T} \left(\sum_{k} P(Y_t^v = y_t^{vk}|X_t = i) \cdot E(Y_t^v = y_t^{vk}) \right) \quad (27)$$

each state can be figured out by *reduce* action on RDD_{State} with (16). Meanwhile, the conditional probabilities and transition probabilities matrixes can be calculated according to (24) and (25) with these statistics.

With the lazy computing strategy of Spark, the probability matrixes computing tasks are finally triggered by reduce action operator and are then distributed to different nodes in the computing cluster based on the data scale and the system configuration. All the data partition and task scheduling operations are automatically managed by Spark platform. Therefore, the algorithm can be implemented in parallel.

C. IMPLEMENT OF MODIFIED FORWARD ALGORITHM ON SPARK

With the objective of minimizing execution time of DBN model, multiple probability matrixes and modified forward algorithm are computed in parallel on Apache Spark.

At time t, the probability of forecast forecasting variable of state i can be calculated with the modified forward operator $\alpha_t(i)$ which indicates the effect of the previous observation variable states on the forecasting variable state at time t, as given in (27), shown at the top of this page, where $\tau_{\text{t-1}}^{ji}$ is the transition probability of forecasting variable from state j to state i, τ_{t}^{ji} is the transition probability of forecasting value from state i to state j and η_n^t is the normalization factor. With (27), the probability of each forecasting variable state at t can be figure out.

Using forward algorithm is preconditioned by the fact that the states of observation variables in critical days are known. Given thatthe state of observation variables in close dates is usually shared by the same division, the state divisions of observation variable on close dates in prior years are applied to simulate the observation variables in forecasting date. The outcome of the algorithm is the forecasting variable state with maximum probability of the forecasting year of a DT on one critical day. The summation of the forecasting result of all DTs in a PSU can represent its one possible load development trend. When the load development trend of all critical days are calculated, the maximum value can be used as the final forecasting result.

Since all critical days' load of each DT are needed to be calculated separately, the calculation tasks of MTLF are executed by distribution executor in parallel based on Apache Spark platform, as shown in Fig. 7.

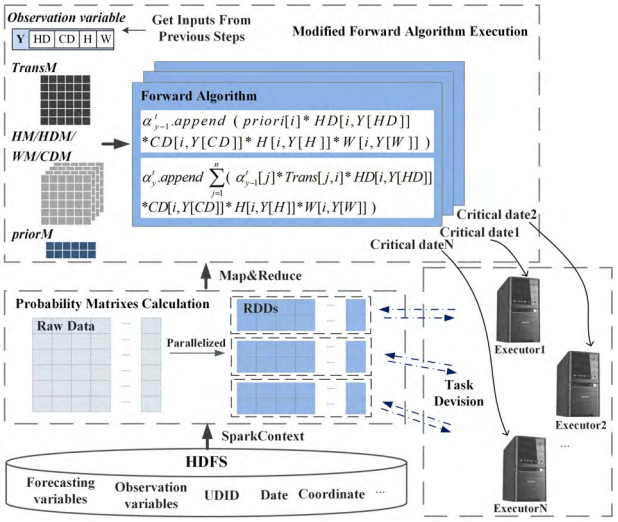

FIGURE 7. Parallel process of forward algorithm based load forecasting on Spark.

V. CASE STUDY

A. EXPERIMENT SETUP

The Spark platform, where all the experiments are performed, consists of one master node and three slave nodes. Each node executes in Ubuntu 16.04.2 and has 32 GB memory. All nodes are connected by a high-speed intranet. Hadoop 2.8.0 and Spark 2.2.0 are installed on both master and slave nodes. The algorithm is implemented in Python 3.6.4.

B. FORECASTING ACCURACY EVALUATION

An accuracy analysis is conducted to investigate the performance of proposed DBN model on forecasting peak load of 2017. In our experiments, the historical dataset is split into two subsets: 1) training set for DBN model (from 2011 to 2016), 2) testing set (maximum load of forecasting objects in 2017). Algorithms used in latest researches are used for performance comparison, which include the least squares multiple linear regression model, ARMA model and feed-forward neural network (FNN) model. The number of hidden layer neurons for FNN is optimized according to [27]. Besides, to validate the proposed DBN model which uses improved DBSCAN for variables state division, the forecasting results from DBN model which uses symmetric variable state divisions are also used for forecasting accuracy comparison. The forecasting result is shown in Fig. 8.

Fig. 8*a* shows the comparison of MTLF results of a typical DT between proposed model and other three models. It can be

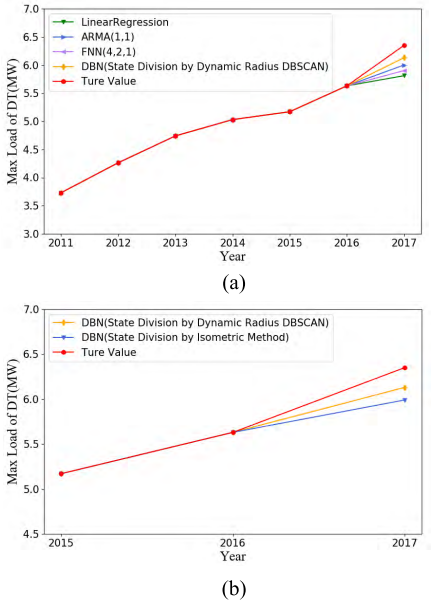

FIGURE 8. MTLF result of a typical distribution transformer. *a* MTLF results of different models. *b* MTLF results of DBN model with different variables state division method.

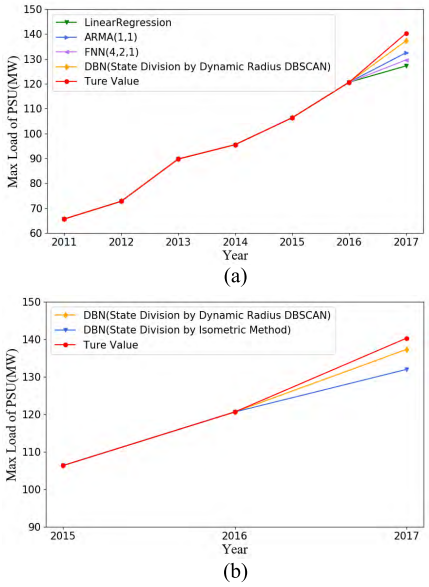

FIGURE 9. MTLF result of a typical power supply unit. *a* MTLF results of different models. *b* MTLF results of DBN model with different variables state division method.

seen that compared with linear regression, ARMA and FNN model, the DBN model improves the forecasting accuracy significantly, for both casual relation and development trend between years are considered in the model. Fig. 8*b* demonstrates that DBN model which uses improved DBSCAN for variables state division outperforms DBN model which uses symmetric variable state divisions, because the improved DBSCAN reduces the interval of forecasting variable state with maximum possibility. However, because most of DTs in our experiment are light-loaded, there is still a gap between true value and forecasting result of DBN model. Elimination of the random errors of single DT will lead to more accurate result in forecasting the maximum load of PSU. Fig. 9 shows the MTLF result of a typical PSU with different models.

Table 2 compares the performance of the proposed DBN based MTLF model in terms of peck load forecasting accuracy with four other techniques. All the presented metrics in the table take the averaged values across all the tested DTs or PSUs. The results show that improved DBN based MTLF model outperforms linear regression by 6.951%, FNN by 5.265%, ARMA by 3.430% and DBN model with symmetric variable state divisions by 3.616% in terms of PSUs' peak load forecasting.

C. PERFORMANCE EVALUATION

In this section, experiments are conducted to evaluate the performance of parallel processing DBN model. The speed of parallel processing DBN model is compared with other algorithms executed on common dataset. The number of experimental samples is ranging from 1 million to 64 million. The average execution times of the tested MTLF algorithms are demonstrated in Fig. 10.

TABLE 2. Performance comparison.

Algorithm	Forecasting Accuracy (%) on DTs	Forecasting Accuracy (%) on PSUs
Linear Regression Model	88.329%	89.995%
Feed Forward Network	90.468%	91.681%
ARMA	93.051%	93.516%
Classic DBN	92.579%	93.330%
Improved DBN	95.669%	96.946%

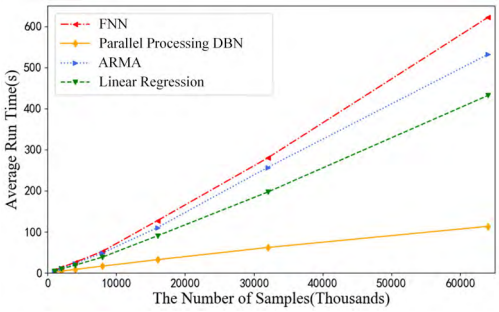

FIGURE 10. Average execution time of the algorithms for different datasets.

When the number of experiment samples grows from 1 million to 64 million, the average execution time of FNN model increases from 5.97 to 621.51 seconds, the average execution time of linear regression model increases from 4.05 to 531.51 seconds, the average execution time of ARMA model increases from 4.82 to 432.83 seconds while the average execution time of parallel processing DBN model increases from 3.35 to 113.12 seconds.The probability matrixes can be re-used, thus, the parallel processing DBN

model achieves a faster processing speed than FNN, linear regression and ARMA do. Taking advantage of RDD based cluster computing, the performance of proposed DBN model will be more significantwhen the data size increases.

Additionally, the proposed MTLF algorithm is implemented in the stand-alone environment and the Spark platform to evaluate the performance of parallel processing. Fig. 11 presents the comparison of different computing environment analysis with number of experiment samples ranging from 1 million to 64 million.

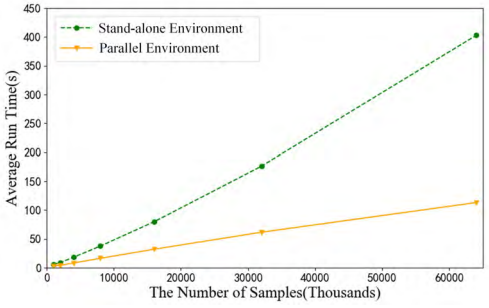

FIGURE 11. Average execution time of DBN model in diffierent environments for different datasets.

When the number of experiment samples grows from 1 million to 64 million, the average execution time of DBN model in stand-alone environment increases from 5.68 to 402.76 seconds, while that of Spark-DBN model increases from 3.35 to 113.12 seconds. Hence, it can be seen that DBN algorithm executed in parallel environment achieves a faster processing speed than serial computing environment does, especially when the volume of historical dataset increases.

VI. CONCLUSION

This paper proposes a DBN model based power supply unit mid-term load forecasting model. And several factors that affect the load curve are also taken into considertation.An improved DBSCAN clustering algorithm is proposed to obtain rational variable state division, and a modified forward algorithm is employed to optimize the calculation of probability model. Given that high volume of historical data of large number of DTs needs to be processed, the parallel processing platform based on Apache Spark is deployed to improve the forecasting performance and speed. Thus, the probabilistic matrixes in DBN model and the modified forward algorithm can be calculated by parallel computing of RDDs with transformation and action operators. The experiments results indicate that parallel processing of DBN model for MTLF on Apache Spark shows superiority in terms of forecasting accuracy and calculation speed.

REFERENCES

[1] S. R. Khuntia, J. L. Rueda, and M. A. M. M. van der Meijden, "Forecasting the load of electrical power systems in mid- and long-term horizons: A review," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 16, pp. 3971–3977, Dec. 2016.

[2] A. Jarndal, "Load forecasting for power system planning using a genetic-fuzzy-neural networks approach," in *Proc. 7th IEEE GCC Conf. Exhib. (GCC)*, Doha, Qatar, Nov. 2013, pp. 44–48.

[3] S.-J. Huang and K.-R. Shih, "Short-term load forecasting via ARMA model identification including non-Gaussian process considerations," *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 673–679, May 2003.

[4] R. Torkzadeh, A. Mirzaei, M. M. Mirjalili, A. S. Anaraki, M. R. Sehhati, and F. Behdad, "Medium term load forecasting in distribution systems based on multi linear regression & principal component analysis: A novel approach," in *Proc. 19th Conf. Elect. Power Distrib. Netw. (EPDC)*, Tehran, Iran, 2014, pp. 66–70.

[5] E. H. Barakat, "Modeling of nonstationary time-series data. Part II. Dynamic periodic trends," *Int. J. Elect. Power Energy Syst.*, vol. 23, pp. 63–68, Jan. 2001.

[6] Ü. B. Filik, Ö. N. Gerek, and M. Kurban, "A novel modeling approach for hourly forecasting of long-term electric energy demand," *Energy Convers. Manage.*, vol. 52, pp. 199–211, Jan. 2011.

[7] W.-J. Lee and J. Hong, "A hybrid dynamic and fuzzy time series model for mid-term power load forecasting," *Int. J. Elect. Power Energy Syst.*, vol. 64, pp. 1057–1062, Jan. 2015.

[8] H. Yu and Q. Zhang, "Application of variable structure artificial neural network for mid-long term load forecasting," in *Proc. 2nd IEEE Int. Conf. Inf. Manage. Eng.*, Chengdu, China, Apr. 2010, pp. 450–453.

[9] C.-I. Kim, I.-K. Yu, and Y. H. Song, "Kohonen neural network and wavelet transform based approach to short-term load forecasting," *Electr. Power Syst. Res.*, vol. 63, no. 3, pp. 169–176, May 2002.

[10] A.-U. Asar and J. R. Mcdonald, "A specification of neural network applications in the load forecasting problem," *IEEE Trans. Control Syst. Technol.*, vol. 2, no. 2, pp. 135–141, Jun. 1994.

[11] I. Arora and M. Kaur, "Unit commitment scheduling by employing artificial neural network based load forecasting," in *Proc. 7th India Int. Conf. Power Electron. (IICPE)*, Patiala, India, Nov. 2016, pp. 1–6.

[12] L. Zhu, Q. H. Wu, M. S. Li, L. Jiang, and J. S. Smith, "Support vector regression-based short-term wind power prediction with false neighbours filtered," in *Proc. Int. Conf. Renew. Energy Res. Appl. (ICRERA)*, Madrid, Spain, Oct. 2013, pp. 740–744.

[13] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load forecasting using support vector machines: A study on EUNITE competition 2001," *IEEE Trans. Power Syst.*, vol. 19, no. 4, pp. 1821–1830, Nov. 2004.

[14] J. Jose, V. Margaret, and K. U. Rao, "Impact of demand response contracts on short-term load forecasting in smart grid using SVR optimized by GA," in *Proc. Innov. Power Adv. Comput. Technol. (i-PACT)*, Vellore, India, Apr. 2017, pp. 1–9.

[15] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4356–4364, Nov. 2013.

[16] C. W. Fu and T. T. Nguyen, "Models for long-term energy forecasting," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, vol. 1, Jul. 2003, pp. 235–239.

[17] M. D. Ghiassi, D. K. Zimbra, and H. Saidane, "Medium term system load forecasting with a dynamic artificial neural network model," *Electr. Power Syst. Res.*, vol. 76, no. 5, pp. 302–316, 2006.

[18] Q. Song and A. Wang, "The prediction of the medium term power load based on combined model of the Bayes theory and LS-SVM," in *Proc. 6th Int. Conf. Natural Comput.*, Yantai, China, Aug. 2010, pp. 940–944.

[19] S. Mirasgedis *et al.*, "Models for mid-term electricity demand forecasting incorporating weather influences," *Energy*, vol. 31, pp. 208–227, Feb. 2006.

[20] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 456–462, Jan. 2014.

[21] N. Yodo, P. Wang, and Z. Zhou, "Predictive resilience analysis of complex systems using dynamic Bayesian networks," *IEEE Trans. Rel.*, vol. 66, no. 3, pp. 761–770, Sep. 2017.

[22] D. X. Niu, B. E. Kou, and Y. Y. Zhang, "Mid-long term load forecasting using hidden Markov model," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Appl.*, Nanchang, China, Nov. 2009, pp. 481–483.

[23] A. Bryant and K. Cios, "RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1109–1121, Jun. 2018.

[24] S. Babichev, V. Lytvynenko, and V. Osypenko, "Implementation of the objective clustering inductive technology based on DBSCAN clustering algorithm," in *Proc. 12th Int. Sci. Tech. Conf. Comput. Sci. Inf. Technol. (CSIT)*, Lviv, Ukraine, Sep. 2017, pp. 479–484.

[25] J. Chen *et al.*, "A parallel random forest algorithm for big data in a spark cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 919–933, Apr. 2017.
[26] I. Chebbi, W. Boulila, N. Mellouli, M. Lamolle, and I. R. Farah, "A comparison of big remote sensing data processing with Hadoop MapReduce and spark," in *Proc. 4th Int. Conf. Adv. Technol. Signal Image Process. (ATSIP)*, Sousse, Tunisia, Mar. 2018, pp. 1–4.
[27] K. Shin-Ike, "A two phase method for determining the number of neurons in the hidden layer of a 3-layer neural network," in *Proc. SICE Annu. Conf.*, Taipei, Taiwan, Aug. 2010, pp. 238–242.

WEI JIANG (M'12) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 2004, 2008, and 2012, respectively. He is currently a Lecturer with the School of Electrical Engineering, Southeast University.

His research interests include the application of power electronics in distributed generation systems, energy storage systems, and power quality control.

HAIBO TANG was born in Jiangsu, China, in 1995. He received the B.S. degree from Chien-Shiung Wu College, Southeast University, Nanjing, China, in 2017, where he is currently pursuing the M.S. degree in electrical engineering.

His research interests include machine learning and its application in smart grid. His gitHub account is: SEU_TangHaibo.

LEI WU was born in Anhui, China, in 1997. He received the B.S. degree from Chang'an University, Xian, China, in 2018. He is currently pursuing the M.S. degree in electrical engineering with Southeast University, Nanjing, China.

His research interests include power electronics and its application in smart grid.

HE HUANG was born in Nanjin, Jiangsu, China, in 1978. He received the bachelor's degree in electrical engineering from Southeast University, Nanjin, in 2000. He is currently with State Grid Jiangsu Electric Power Company. His technical title is Senior Engineer.

His previous research interests are the application of big data in power system and big data transactions in the electricity market.

HUI QI was born in Taizhou, Jiangsu, China, in 1981. He received the bachelor's degree in electrical engineering from the Harbin Institute of Technology, Harbin, Heilongjiang, China, in 2000. He is currently with State Grid Jiangsu Electric Power Company. His technical title is Senior Engineer.

His previous research interests are the application of big data in power systems and the impact of Internets accelerated speed technology on grid development.