

Load Forecasting of Power SCADA Based on Spark MLlib

Tao Lin and Chong Jiang

No.8, Guangrong Road, Hongqiao District, Tianjin, China

Abstract—In order to improve the accuracy and speed of power forecasting in power SCADA system, a distributed real-time steaming forecasting model is designed based on K-means algorithm and Random Forest algorithm in the Spark machine learning library (MLlib). The model uses the sliding window mechanism to segment the incoming data stream. K-means Clustering is used to correct the abnormally data, and then the Random Forest Regression forecasting is performed. Model algorithms is implemented based on the Spark RDD, the performance of the algorithm is verified by sending the data through the daemon process which is a simulation of the message queue. The results show that the forecasting accuracy of the algorithm is superior to the traditional serial Random Forest forecasting and satisfies the real-time requirement.

Keywords—component; spark; decision tree; random forest; k-menas

I. INTRODUCTION

The prediction of the load of the power system can guide the rational economic of power generation and also maintain the stability of the power grid, which is the research hotpot of the data acquisition and monitoring system(SCADA). There are some classical methods of load forecasting, such as trend extrapolation method, time series method and so on, but they all based on historical data, the accuracy of prediction declines rapidly with the increase of time scale. With the development of non-linear tools such as neural network and fuzzy mathematics, the combination method with higher precision has emerged as well. At present, common prediction methods are decision tree, limit learning, genetic algorithm, support vector machine method, these methods provide a variety of effective ways for power load forecast. However, most of the above methods are single-machine prediction model which based on good historical data and small data amount.

As the data amount and dimensions of SCADA system rising, the above method may occur, such as over-fitting, local optimization and other shortcomings. Therefore, new data processing architecture with large data and distributed computing platform is becoming more and more urgent. New research methods are also emerging. In literatures 1-3, load forecasting algorithms based on distributed computing framework are proposed, show that the results are superior to traditional neural network and support vector machine.

Based on previous studies, this paper proposes a framework based on Spark computation framework, and combines the parallel K-means algorithm with parallel random forest regression algorithm for the characteristics of massive,

high-dimensional and real-time data in SCADA system. The model can effectively reduce the impact of abnormal data and at the same time improve the system's prediction accuracy, to achieve the desired effect of forecast.

II. DISTRIBUTED STEAMING COMPUTING FRAMEWORK

Spark is a typical distributed computing framework running on a cluster and compared to Hadoop, Spark can work entirely in memory, reducing the amount of system I / O. Ideally, Spark can operate 100 times of Hadoop. Spark Streaming is an extension of the Spark core API and is an important part of the current Spark ecosystem, which takes data from real-time message queues then converted to batches as they are accepted. RDD based on the various high-order functions described by the filtering algorithm to get the desired results. More and more machine learning algorithms are being improved in parallel and assembled as part of the Spark Machine Learning Library (MLlib), because it is fully working in the memory of the cluster, making it easy to iterate over multiple queries.

III. K-MEANS AND RANDOM FOREST ALGORITHM

A. K-means Clustering Algorithm and Its Improvement

The purpose of K-means clustering is to divide n points into k clusters, so that each point belongs to the cluster which nearest to its nearest mean. The algorithm flow are as follows:

Step 1. Initialize Center, k objects are selected randomly from N objects as the centroid of k clusters;

Step 2. Grouping, calculate the distance of each centroid to the remaining objects, and classify them into the cluster to which the nearest centroid belongs;

Step 3. Updating, according to the average of the various points in the cluster, select the new cluster center;

Step 4. Iteration, repeat the second and third steps, if the new centroid is equal to the original centroid or the difference is less than the given threshold, or the specified number of iterations is reached, stop the iteration and output the final classification result.

B. Decision Tress and Random Froest Algorithms

Decision tree is a classical prediction algorithm in machine learning. Compared with other regression models, the decision tree has good robustness to the outliers in the data set. The

algorithm is divided into three steps: feature selection, decision tree generation and pruning of decision tree.

The decision tree determines a path from the root node to the leaf node as a category based on the input. The goal of this selection process is to make the sorted data pure, measured by node impurity. Node impurity is a measure of the homogeneity of the node data, and the variance is also called the least squares bias: $\frac{1}{N} \sum_{i=1}^N (y_i - \mu)^2$ for the regression tasks needed by

this paper. Where y_i is the label of the feature vector, N is the number of data sets, μ is mean given by $\frac{1}{N} \sum_{i=1}^N y_i$. The

information gain is the difference between the impurity of the parent node and the impurity purity of the two sub-nodes, which can be used to measure the effect of a feature on the classification result. Suppose an S is divided into the data set D of size N into two data sets D_l and D_r , whose data set sizes are N_l and N_r , respectively, then the information gain is: $IG(D,S) = \text{Impurity}(D) - \frac{N_l}{N} \text{Impurity}(D_l) - \frac{N_r}{N} \text{Impurity}(D_r)$.

Feature Selection: Select a suitable feature as the decision node, reducing the depth of the decision tree. At the time of feature selection, the feature with the largest gain should be selected as the splitting condition of the node. It is often the case that not all records can be accurately classified, making the construction of the decision tree difficult to stop. This requires setting the stop condition and ending the build.

Decision Tree Generation: A top-down tree-like classifier is generated by selecting different greedy algorithms to classify nodes.

In order to improve the generalization ability of the decision tree and avoid the over-fitting, Random Forest algorithm, which combines multiple decision trees, arises at the historic moment. Random forest algorithm training a set of decision tree, the training process can be parallel, the algorithm using the randomization of the decision tree are slightly different, the combination of these decision tree prediction results also better. The following two randomization processes run through each decision tree generation process.

Bootstrapping: Randomly put back samples from training data N for each decision tree;

Feature Subspace: The subspace m is chosen from all the attributes M (m should be much smaller than M), and the optimal attribute is selected from the subset. The condition is the maximum information gain in the decision tree.

Each decision tree thus obtained grows individually and the prediction results are weighted average (regression) or most (classified) to arrive at the final result of the random forest algorithm.

IV. LOAD FORECASTING BASED ON SPARK

A. Exception Handling Model Based on K-means++

Take the power load data of SCADA system as the ordinate, the corresponding time point is the abscissa, get the load curve, to identify whether a certain data on this curve is bad data, need to set a load characteristic curve as a reference. In order to calculate the load characteristic curve, the first assumption is the similarity: the load curves are similar within a few days. Consider the longitudinal similarity in Cartesian coordinates, quantified by the distance between the curves. This distance is the precision of clustering. The curve in a certain range can be labeled as a curve. The essence of the load characteristic curve is the centroid of each curve. The occurrence of bad data is accidental, assuming that the proportion of all the data is very small and does not affect the center of the heart.

For the rate of change of load on curve C , we examine whether these historical data are bad data or normal data. Before correction, first find the corresponding characteristics of the correct curve, each feature curve is a mass center, corresponding to a class of curves. After finding the bad data, the bad data can be corrected according to the horizontal similarity of the load curve. The revised formula is:

$$X_c(i) = X_i(i) \times \frac{[\frac{X_d(p-1)}{X_i(p-1)} + \frac{X_d(p+1)}{X_i(p+1)}]}{2}, \quad (1)$$

$$i = p, p+1, \dots, q$$

where X_d is the load curve with bad data, X_c is the load curve after repair, X_i is the characteristic load curve belonging to the load curve, and p to q are the bad data founded.

B. Load Forecasting Model Based on Random Forest

There are two main parameters that affect the performance of the decision tree: maxDepth and maxBins of the decision tree. The modest increase of these two parameters will increase the complexity of the model and make the model face the high dimension data of the performance better. As the random forest is a "random" decision tree composed of "forest", in addition to the above two parameters, there are the number of decision trees can affect the performance of the model.

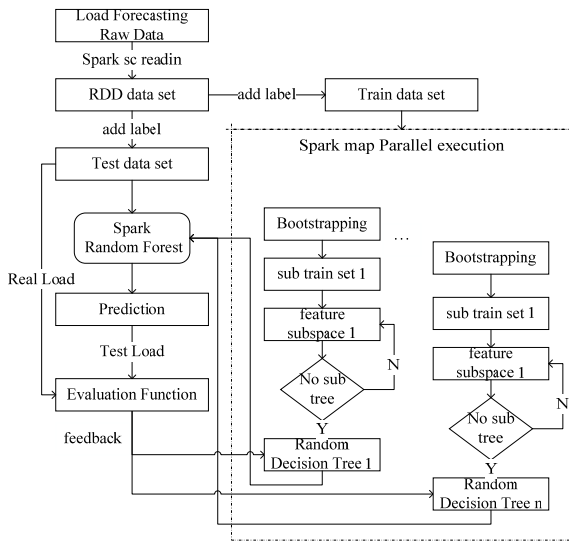


FIGURE I. RANDOM FOREST ALGORITHM'S FLOW CHART

There are many kinds of evaluation indexes of load forecasting results, such as Mean Square Error (MSE), Mean Absolute Error (MAE), Root Mean Square Logarithm Error (RMSLE) and Mean Absolute Percent Error (MAPE). Mean

$$\text{absolute percentage error (MAPE): } \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i' - y_i}{y_i} \right| \times 100\%$$

same as MAE as a general measure of the size of the error bias. MAPE was used to evaluate the performance of the model. The smaller the value of the evaluation index, the better the performance of the model.

V. VALIDATION

A. Data Preparation

In order to verify the effectiveness of the model and illustrate the versatility of the model more clearly, the validity of the model was verified by selecting real data from the EUNITE competition, which is widely used in power load forecasting. Integrate historical load with variables that influence load, such as temperature, holiday information, or periodic weeks, months, quarters, and even humidity, elevation, and other factors. This on the one hand is to the data collected by the SCADA system, on the other hand is related to the construction of the eigenvectors of the data set.

B. Exception Handling

The EUNITE competition data was selected as the input data for 24 load data records per day from January 1 to December 31, 1997. On March 10 of the 12,13,14 point of the original data to artificially increase, expanding 60%.

The clustering characteristic curve based on Spark's k-means ++ algorithm compared with the actual curve as follows:

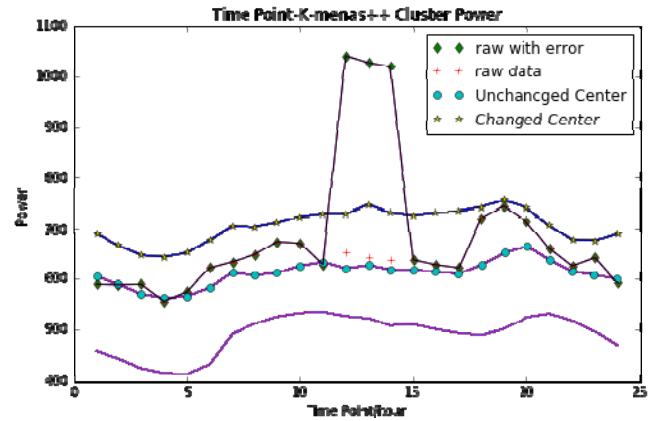


FIGURE II. K-MEANS CLUSTER

The smoothly changed three curves in the above graph are the result of algorithm clustering. The curve with obvious outliers is the abnormal curve belonging to the second cluster center. According to the above correction formula (formula 1), the outliers are corrected, the error percentage of the contrast is as follows:

TABLE I. TWO K-MEANS ERROR PERCENTAGE CONTRAST

Algorithm	Actual Value	Correction Value	Error Percentage
Traditional K-means ^[5]	652	629.12	3.51
	643	634.67	1.30
	638	626.15	1.86
Spark Based k-means++	652	629.26	3.49
	643	634.72	1.29
	638	626.33	1.83

It can be seen that the advantage of the algorithm is not obvious, because the data deviation is large, the core principle of the algorithm is the same, but after the parallel, the latter algorithm has an irreplaceable advantage in dealing with streaming and mass data.

C. Load Forecasting

Load prediction is more concerned about the maximum load power of the forecast, because it is very important for the determination of the power system power generation and transmission and distribution plan. The validation of this paper is to predict the maximum load power. There is not significant difference between predict the maximal load or the dense time series load, because the training process can fully convert the time period into one dimension feature vector.

The daily mean temperature is chosen as a numerical feature, and whether or not holiday and day of the week are selected as the two type features is selected. The daily maximum load is used as the label value to form a four-dimensional feature vector. For the linear model, since there are seven selections for the week and two for the vacation, the length of the feature vector of the category type is 9, the length of the feature vector is 2, and the total length of the feature vector is 11. However, because the decision tree itself supports

the class feature, the feature vector of the decision tree does not need to do binary processing, the length is still 4.

The following figure shows the effect of random forest on the impact of random forest parameters:

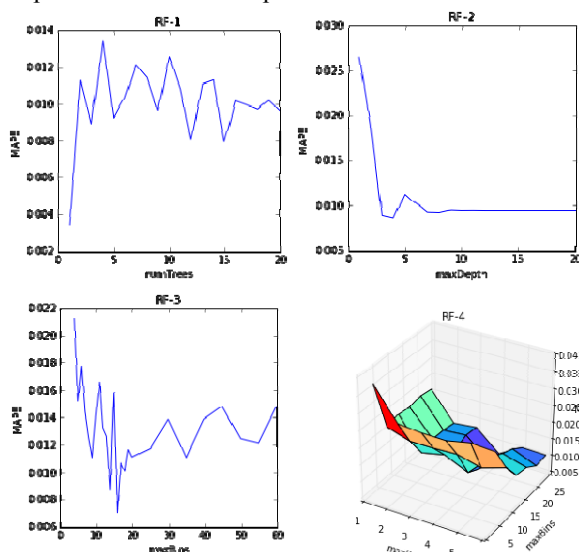


FIGURE III. RANDOM FOREST'S PERFORMANCE

RF-1, Fixed partition and tree depth, the number of decision trees on the impact of random forest performance curve. Because the complexity of the test data is relatively low, so the performance of the two decision trees is the best. Therefore, all the following comparisons hold the number of fixed decision trees to two.

RF-2, Fixed partitions and trees number, and the maximum depth of the tree on the performance of random forests. RF-3, Fixed depth and the number of trees, the maximum number of changes in the number of random forest performance curve. Although increasing the accuracy of each algorithm can improve the precision of the algorithm to a certain extent, increasing the complexity of the model does not increase the system performance.

From the three-dimensional graph in RF-4, it is known that the maximum depth and maximum partition of each decision tree will affect the performance of the random forest algorithm. When the number of trees is 2, the maximum tree depth is 6, and the partition is 16, the system performance is optimal, MAPE is 0.0071.

After traversing the parameters to obtain the optimal performance model, the model can be used to predict the test set. The following is a comparison of the predicted and actual results using random forest algorithm:

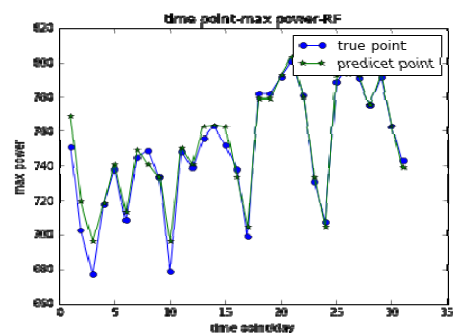


FIGURE IV. COMPARISON OF PREDICTIVE RESULTS AND ACTUAL VALUES OF RANDOM FOREST ALGORITHM WITH OPTIMUM PARAMETERS

D. Comparison of Prediction Accuracy

In order to quantify the performance of the contrast model, the following table shows the best performance comparisons for each model in other papers using the same training set and test set.

TABLE II. ACCURACY COMPARISON WITH OTHER ALGORITHMS

Prediction Algorithms	MAPE/%
Spark Random Forest	0.71%
Spark Decision Tree	0.14%
IPPSO-LS-SVM ^[6]	1.7302%
MR-OSELM-WA ^[1]	1.9498%
Functional Networks ^[7]	3.4300%
SVR ^[8]	2.1498%

VI. CONCLUSION

Support vector regression (SVR) and generalized neural networks (functional networks) have shown very good predictive power in EUNITE. These two algorithms are batch learning mode, cannot be extended in mass data mode.

The on-line serialization algorithm based on map-reduce (MR-OSELM-WA) solves the problem of mass data expansion by parallelizing the limit learning algorithm in the Hadoop framework. However, disk-based iterations produce large numbers of I/O and can not meet the real-time requirements. LS-SVM algorithm based on improved Particle Swarm Optimization (PSO) algorithm based on Spark platform not only realizes the expansion of data quantity, but also realizes the memory-based operation, and improves the efficiency of the operation significantly. However, the algorithm of the model implementation is complex and cannot be modified for stream processing.

Based on the parallel algorithm in MLlib, the original data is first modified by the k-means algorithm and then passed to the decision tree and the random forest module for prediction, along with the sliding of the Spark Streaming on the DStream. The prediction results in each time window can be given. As can be seen from Table 3, both the decision tree based on Spark and the random forest algorithm have better prediction accuracy than other algorithms. Although the optimal

performance of the decision tree is higher than that of the random forest, it is difficult to appear when the data with high complexity. The distributed memory and streaming framework not only meet the requirements of scalable and high-dimensional scalability, but also improve the accuracy of power load forecasting.

ACKNOWLEDGMENT

I'd like to extend my sincere gratitude to my supervisor, Lin Tao, for his useful guide and suggestions. Special thanks should go to my division brothers and sisters who help me a lot with the draft. Finally, thanks my family for their continuous support and encouragement.

REFERENCES

- [1] Wang Baoyi, Zhao Shuo, Zhang Shaomin, "A Distributed Load Forecasting Algorithm Based on Cloud Computing and Extreme Learning Machine" *Power System Technology*, vol. 38, No.2, pp. 526-531, Feb 2014.
- [2] Wang Dewen, Sun Zhiwei, "Big Data Analysis and Parallel Load Forecasting of Electric Power User Side" *Proceedings of the CSEE*, vol 35, pp. 527-537, Feb 2015.
- [3] Ma Tiannan, Niu Dongxiao, Huang yali, and Du Zhendong, "Short-Term Load Forecasting for Distributed Energy System Based on Spark Platform and Multi-Variable L2-Boosting Regression Model" *Power System Technology*, vol. 40, No.6, pp. 1642-1649, Jun 2016.
- [4] Meng Jianliang, Liu Dechao, "A new method for identifying bad data of power system based on Spark and clustering analysis" *Power System Protection and Control*, vol. 44, No3, pp. 85-91, Feb 2016.
- [5] Wang Baoyi, Wang Dongyang, Zhang Zhaomin, " Distributed short-term load forecasting algorithm based on Spark and IPPSO_LSSVM" *Electric Power Automation Equipment*, vol. 61, No.1, pp. 117-122, Jan 2016.
- [6] Castillo E, Guijarro B, Alonso A. Electricity load forecast using functional networks[C]//EUNITE Symposium-Competition on Electricity Load Forecast Using Intelligent Technologies. 2001.
- [7] Chen B J, Chang M W. Load forecasting using support vector machines: A study on EUNITE competition 2001[J]. *IEEE Transactions on Power Systems*, 2004, 19(4), pp. 1821-1830.