Big Data Research 5 (2016) 9–15

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Machine Learning with Big Data An Efficient Electricity Generation Forecasting System ☆

Mohammad Naimur Rahman, Amir Esmailpour *, Junhui Zhao

Department of Electrical & Computer Engineering and Computer Science, University of New Haven, West Haven, CT, 06516, United States

ARTICLE INFO

Article history:
Received 1 June 2015
Received in revised form 26 January 2016
Accepted 19 February 2016
Available online 24 March 2016

Keywords:
Artificial neural network
Backpropagation
Big Data
Electricity generation forecast
Hadoop
MapReduce

ABSTRACT

Machine Learning (ML) is a powerful tool that can be used to make predictions on the future nature of data based on the past history. ML algorithms operate by building a model from input examples to make data-driven predictions or decisions for the future. The growing concept "Big Data" has brought much success in the field of data science; it provides data scalability in a variety of ways that empower data science. ML can also be used in conjunction with Big Data to build effective predictive systems or to solve complex data analytic problems. In this work, we propose an electricity generation forecasting system that could predict the amount of power required at a rate close to the electricity consumption for the United States. The proposed scheme uses Big Data analytics to process the data collected on power management in the past 20 years. Then, it applies a ML model to train the system for the prediction stage. The model can forecast future power generation based on the collected data, and our test results show that the proposed system can predict the required power generation close to 99% of the actual usage. Our results indicate that the ML with Big Data can be integrated in forecasting techniques to improve the efficiency and solve complex data analytic problems existing in the power management systems.

1. Introduction

The United States (U.S.) is currently the second largest electricity producer and consumer in the world [1]. The U.S. enjoys a magnificent geographical diversity among states with a high amount of power consumption. This makes it challenging to deploy a centralized power management system that can control the power generation and regulate the consumption. The electricity is mostly generated from natural resources, such as coal, gas, nuclear, petroleum, oil, and renewable energy. The consumption sectors can be detailed in terms of commercial, industrial, residential and other user communities.

Due to lack of centralized control, there is a large disparity in the ratio of power consumption/power generation from one state to the next. This imbalance results in wasting large quantities of power generated in states where generation significantly exceeds consumption, while other states are suffering from insufficient amount of power generation. Due to the size and the geographical diversity of different states in the U.S., it is farfetched to prescribe centralized control over the power system. Merely, the interstate segments are regulated by the federal government [2,3], and the majority of the rest of the nation is delimited by individual states. Fig. 1 shows the electricity generation and consumption in the U.S. during 1980–2014. In this figure, the green line at the bottom shows the consumption, the red line in the middle represents the actual generation, and the blue line on top indicates total generation including net import (i.e. from neighboring countries). The difference between the generation (red line) and consumption (green line) is attributed to system losses, uncounted loads, and the lack of centralized control.

Fig. 2 shows electricity generation in the U.S., by state. States shown in lighter brown color are not producing enough electricity to meet their demand. Other states (shown in darker orange color) produce excess electricity, which could be used to compensate for the brown states lacking sufficient power generation. Further deficiencies are fulfilled by importing electricity from neighboring countries.

Power generation is in direct correlation with the amount of resources used to generate the electricity such as coal, gas, nuclear, petroleum, oil, and renewable energy. In Fig. 1, the red line in the middle (representing the power generation in the U.S.) provides two types of information: the amount of energy consumed and the quantity to be imported. Therefore, predicting power generation might provide vague information about power demand; hence

☆ This article belongs to Analytics & Applications.

* Corresponding author.
E-mail addresses: MRahm1@unh.newhaven.edu (M. Naimur Rahman), aesmailpour@newhaven.edu (A. Esmailpour), JZhao@newhaven.edu (J. Zhao).

http://dx.doi.org/10.1016/j.bdr.2016.02.002
2214-5796/

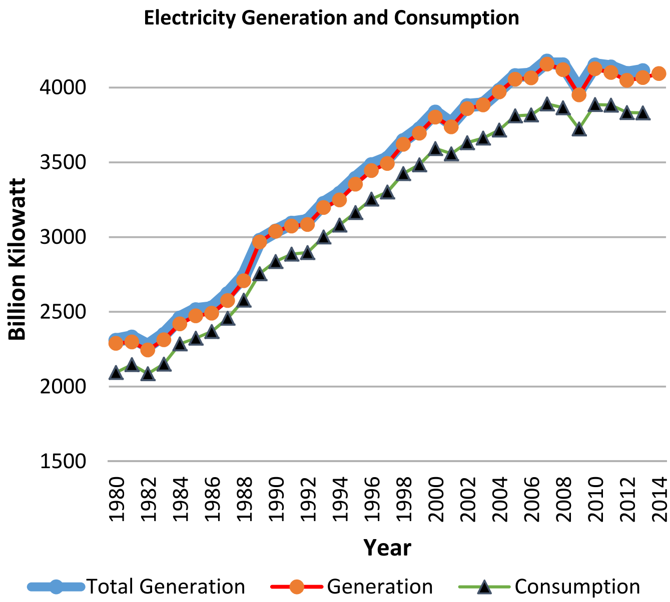


Fig. 1. Electricity generation and consumption graph. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 2. Electricity generation in the U.S., by state. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

increase the quantity to be imported from neighboring countries. It is critical to explore possibility of centralized power management and to determine the allocation of natural resources.

The prediction is challenging due to the accuracy requirement, and it becomes even more cumbersome when datasets are enormous in volume and have excessive noise and high volatility. Several forecasting methods using different species of Machine Learning (ML) algorithms, such as fuzzy neural network [15,22], gray algorithm [16], gray Markov model [17], and support vector regression [18] have been proposed to deal with electricity forecasting problems. Those models were showing impressive results in terms of forecasting accuracy. However, they might not be as effective dealing with Big Data, where more efficient schemes must be employed to deal with large volumes and complexity of datasets. On the other hand, large penetration of renewable energy sources, such as wind and solar systems, increases the uncertainty in generation [29,30].

It is important to forecast the power generation in order to allocate resources that produce the power and to calculate the demand and the quantity to be imported from neighboring countries. To reach this goal, ML methods based on Artificial Neural Network (ANN) algorithms have been developed. However, there still remains the problem of how to deal with large data size and complex mining process, and how to make the algorithms scalable and intact in their performance. In this study, a prediction method is developed based on a three step framework that incorporates Big Data analytics. First, raw data were processed and converted to suitable format; then, the data were normalized to get better performance from the ML algorithm; and finally, the data were fed into an ANN model for training purposes. The deployment begins by collecting past power generation data from all the states in the U.S., and storing it in a distributed database. Then Big Data tools are used to deal with the processing of the data. Data are first distributed to a group of computing nodes inside Hadoop cluster, and distributed algorithms are implemented in form of MapReduce to take advantage of distributed high performance computing paradigm in the laboratory environment. Afterwards, data are fed into the ANN algorithm to train the network. Finally, forecasted results from ANN are compared to the actual generation.

Fig. 3 depicts different steps in the framework for the proposed strategy. In the first step, the framework collects past power generation data from all U.S. states and stores them in a distributed database. This is the raw data with redundant information, some of which are in a completely unstructured format such as text files; others are not in any desired structured format such as csv formatted file. In the next step, Big Data tools are applied, MapReduce is implemented on top of Hadoop cluster to deal with such large datasets. Data are stored in multiple computing nodes, and distributed algorithms are implemented in the form of MapReduce. MapReduce is used to allocate assignment and to handle large datasets. Manipulated data is extracted from each computing node in the desired format. Then, data are normalized to increase the effectiveness of the ML algorithm. Finally, data from each node are used on ANN for training to predict the future power generation.

Forecasting electricity generation will eventually yield information on the demand, since there is a linear relationship between the two. Also, it is easier to deploy centralized control if we have enough information about generation and consumption for individual states as well as for the entire nation. Therefore, knowing the total generation eventually determines the amount of electricity to be imported from neighboring countries.

The remainder of this paper is organized as follows. Section 2 briefly introduces ML and ANN methods. Section 3 describes some related works, and Section 4 presents the detailed strategy and the design of the framework, followed by the results in Section 5. Finally, Section 6 concludes the paper with a discussion.

2. Background

Machine Learning (ML) and Artificial Neural Networks (ANN) are parts of cognitive science, initially evolved from two important concepts, pattern recognition and computational learning, both parts of Artificial Intelligence (AI) [4,7,8]. ML deals with analyzing algorithms that can be trained to make predictions for the future based on the past information. ANN is a learning process based on statistical models and human biological neural networks. ANN is used to estimate values based on a large number of inputs. ANN interconnects neurons with numeric values, adjustable based on experience, allowing them to use the inputs in the learning process. In this study we employ these concepts to build a framework for the electricity generation predictions with large volume of data.

ML and data mining processes have strong ties with mathematical optimization to build complex models, where designing and programming explicit and rule-based algorithms are infeasible. There are several ML algorithms, where the learning process can be supervised or unsupervised. ANN is one of the popular supervised learning process methods [7,26,27].

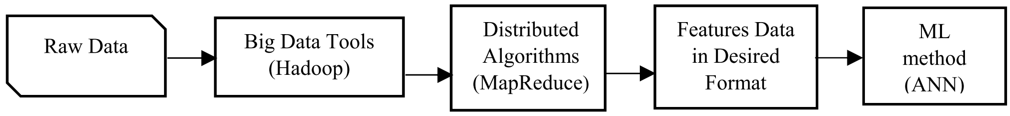


Fig. 3. Workflow for the proposed strategy.

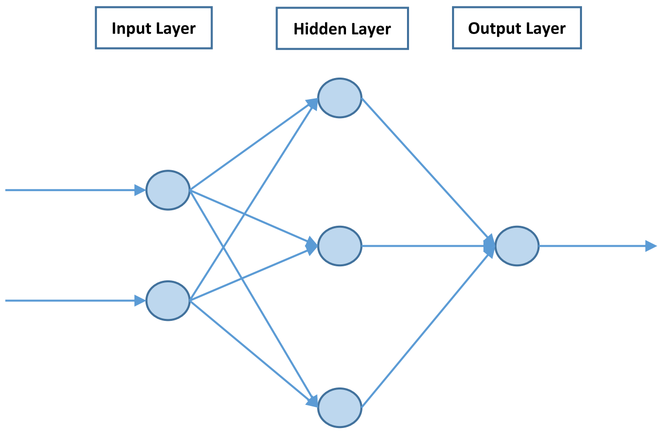


Fig. 4. Backpropagation Neural Network (BPNN).

ANN system acts like a human brain to process information and can be employed to determine the complex relationship between inputs and outputs of processes. A trained ANN system has the capability to predict the output as a set of previously unseen inputs once it is successfully trained. Numerous ANN algorithms have been proposed in the literature. Backpropagation (BP) algorithm was employed in this study. Fig. 4 shows a basic BP Neural Network (BPNN) consisting of three layers: input, hidden and output. There are two input, three hidden and only one output layer nodes (2-3-1) [14].

Forecasting electricity generation and consumption parameters is a difficult task because of the complex characteristics of data such as high volatility, inherent noise, hidden relationship and dependency on other parameters, such as climate, tariffs, and effort to uplift the energy conservation. However, much research has been done to deal with those difficulties. Among them, ANN is found to be more efficient than other intelligent forecasting systems. Several ANN algorithms and their modified versions were implemented [19,23], and [24] to predict the electricity demands. Many research projects have proposed the improved versions of ANN in solving forecasting problems, which are proven to be more efficient than native ANN. For instance, in reference [20] the authors used feed forward NN, and in [15] they used fuzzy logic NN. Other ML algorithms such as Support Vector Machine (SVM) [18] and Recurrent SVM with Genetic algorithm (RSVMG) [21] are also used to forecast electricity demand, which also outperform other ANN schemes.

Those systems proved to be efficient in analyzing small-scale datasets. Prediction of large datasets might not work as efficiently because of difficulties in the structure of large datasets and elimination of noise at the same time. Big Data tools can be used to deal with large electricity datasets, and ANN can be applied after processing those datasets. A similar approach was applied by D. Xian et al. [11], in which the authors predicted stock features using decision tree and SVM. They used Big Data tools to handle large datasets. Mining valuable data from a large volume of complex datasets is a challenge. However, several studies focus on overcoming the data mining challenges with Big Data tools [6,25].

3. Framework design for the proposed strategy

In this work, efficient electricity forecasting is built using the ML approach with Big Data to overcome the challenges related to large datasets. The proposed framework is designed not only to build an effective forecasting system, but also to solve the problems related to unstructured and semi-structured datasets that have noise, using distributed algorithms in the form of MapReduce [9,10]. The framework consists of three main phases including data collection and processing, data normalization, and prediction training.

Fig. 5 shows the design of the framework in detail. The three main stages of the framework are designed as follows: (i) process raw data and extract features, (ii) normalize the data in structured format, and (iii) train BP algorithm for ANN forecasting. There are also two additional stages that complete the entire process as shown in Fig. 5. Prior to the three stages, data have to be prepared by storing it in Hadoop Distributed File System (HDFS), and distributing it among appropriate nodes. Initially, data is stored in a database, then it is loaded into the HDFS, which distributes data to different nodes. Then to extract features from data using high performance distributed computing, an algorithm in the form of MapReduce is implemented. Featured data in structured format is saved again into HDFS. Then the data from each computing node is normalized before it is used to train the BP network. Finally, the fully trained network is used to forecast future electricity generation.

3.1. Process raw data and extract features

Datasets for monthly power generation in each state was collected for the past 15 years [5]. The data contained redundant information and texts in different sets since the consumption varies from one state to the next. Some sets were in a completely unstructured format. Those datasets form a typical Big Data problem in terms of complexity and noise related to size. In order to deal with such Big Data problem efficiently, the designed framework goes through several stages for raw data treatment.

Initially, the raw data is stored in HDFS inside the Hadoop cluster. HDFS stores files in a distributed fashion, and it also replicates data blocks in different nodes (for this work the replication factor was set to default value of 3). Hadoop breaks the data into chunks or blocks to be stored inside HDFS. The data can be divided into blocks of 64, 128, and 256 MB. In this work, default block size of 64 MB is chosen. The data are first divided into blocks and then placed into HDFS; later the replication is performed. The reason for storing data in a distributed format is to perform parallel processing and computation of large data, while increasing reliability, flexibility, and scalability. Then we applied MapReduce, a low level language to retrieve desired features from data. We have implemented Mapper and Reducer algorithms in MapReduce to perform their tasks. The Mapper function tells the cluster which data points are required to be retrieved, and then the Reducer acquires and aggregates all the data, and converts it to a suitable format [32]. The Hadoop cluster contains one master and several slave nodes (NameNode acts as master and data nodes act as slaves). MapReduce has one master that is JobTracker, and the slave is TaskTracker. NameNode stores the metadata where the raw data are located, and data node stores the data. JobTracker keeps

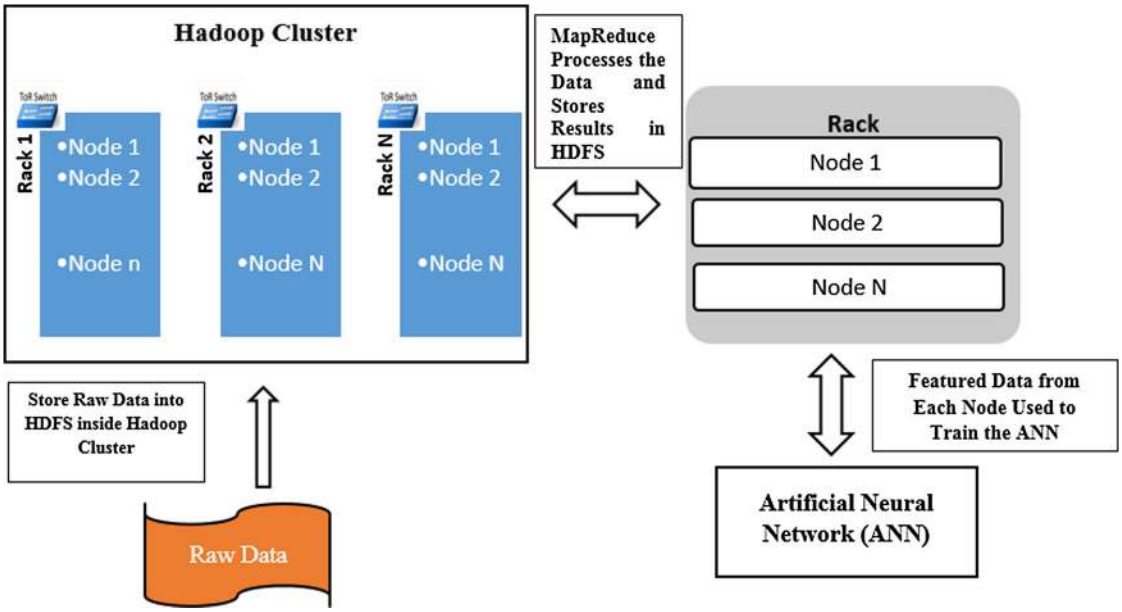


Fig. 5. The framework design stages.

track of the tasks to be performed and TaskTracker performs the task on the data. The master node distributes the assignment to the group of slaves. Slave nodes carry out the computation and are periodically monitored by the master. Once the computation is performed, the results from those nodes are submitted to the master and stored in HDFS. In the MapReduce, job is performed in a pseudo-distributed mode (that means all the Hadoop daemons are running under separate Java Virtual Machine (JVM) process), on both single and multi-node clusters (in our multi-mode cluster all the master and slaves have their own VMs).

Algorithm 1 *(MapReduce operational model to extract data).*

```
// Map Class
Input: (Key: name of the input; Value: Value of the input)
Output: (Key, Value)                  // Pair of Key & Value
Map (Key, Value) Start

Intermediate ('Key', Value);
// Reducer Class
Input: (Key: name of the Mapped data;
          Value: List of all map data with same key)
Output: Key of the Mapped data into row and column and
          save into CSV file
Reducer (key, Value) Start
    i = 0;
    While (values.hasNext () ){
        value = values.next().get();
        output = values + line.split(cvsSplitBy);
    i++
        If (i = 13){
        return; // for 12 months value and next month as target
        }
    }
Output (key, Row, Column);
```

Algorithm 1 shows the operation of the MapReduce process, concluded in three steps. Raw data are stored in HDFS and extracted to clean up and to be converted to structured format. The MapReduce operation is performed to mine the data from cluster and transform into structured format, which is suitable for ML process.

3.2. Normalize the data in structured format

The data features are extracted from raw data in order to be converted to structured format. Now a separate algorithm is constructed to do the normalization task. Feature extraction and normalization can be done in one MapReduce process. However, the framework demonstrates separate algorithms for the ease of feature extraction process. There are several normalization techniques in the literature [13]; their algorithm performance could be varied based on the normalization methods. Statistical column normalization is selected in this study for the structured data for its ability to diminish the error quickly and reduce the chance of local maxima and minima [13]. The normalization factor is calculated using Equation (1).

Normalized value of each column data (V_{nor}) is:

$$V_{nor} = Value * NF \tag{1}$$

where,
Normalization factor, $NF = V_{max}/F$.
V_{max} = Maximum value of the column
F = Convert the value of V_{max} to floating point

Algorithm 2 *(MapReduce operational model for normalization).*

```
// Map Class
Input: (Key: name of the input;
          Value: Value of the input)
Output: (Key, Value)                  // Key & Value Pair
Map (Key, Value) Start
Intermediate ('Key', Value);
// Reducer Class
Input: (Key: name of the Mapped data;
          Value: List of all map data with same key)
Output: Key of the Mapped data into normalization function
Reducer (key, Value) Start
          While (values.hasNext()){
              Normalization Factor = Normalization Equation;
              value = values.next().get() * Normalization factor;
          }
Output (key, Value);
```

Algorithm 2 shows the operation of the normalization process. The main goal of normalization is to increase the power and the quality of ML. Although normalization can be done by other algorithms, MapReduce is used to achieve high performance and scalability.

Statistical normalization is employed here since the BPNN performance depends on normalized value of the input data [12]. Normalization improves the quality of the ML and also the performance of the algorithm. Structured datasets are again saved to the Hadoop cluster after the normalization operation is performed.

3.3. Train BPNN for generation forecast

The output data in structured format stored in HDFS is retrieved for training the BPNN, which is an important part of the framework. Data is divided into two sets: 90% used for training the network and the remaining 10% for testing the network. For each prediction; In the input layer, there are 12 nodes; in the hidden layer, 6 nodes; and in the output layer one node (12-6-1). The size of the input layer contains the number of features in the data. Before setting the number of input nodes to 12, the forecasting results are evaluated using the 3rd, 4th, 6th and 8th input nodes. After this evaluation, the generation of next month is forecasted by the past 12 months' generation data, which has been included into the network. Hence, the algorithm outputs optimal results for the past 12 months as input into 12 nodes. The algorithm can recognize the pattern very well if the entire year is used. The size of output layer is also determined in a similar manner. BPNN can be run in two different ways: ML mode and Regression mode. ML mode determines the output as class label, and the regression mode returns values (e.g. predicting price). In this work, BPNN runs on regression mode and the output layer has a single node. There is one hidden layer with 6 nodes. Usually with the increase of hidden layer numbers, the performance improvement is very small, it also increases computation overhead. The size of hidden layer nodes depends on the size of input and output layers nodes. Empirical studies suggest the optimal size for hidden layer nodes lies between the size of input layer and the size of output layer [31]. The framework is tested for 6, 7 and 8 nodes in hidden layer. However, the performance is identical for any case above 5 nodes in hidden layer. In each node we have activation function, triggered after a certain level of inputs. The activation function is given in Equation (2).

$$f = \frac{1}{(1 + \exp^{-(net\ input)})} \tag{2}$$

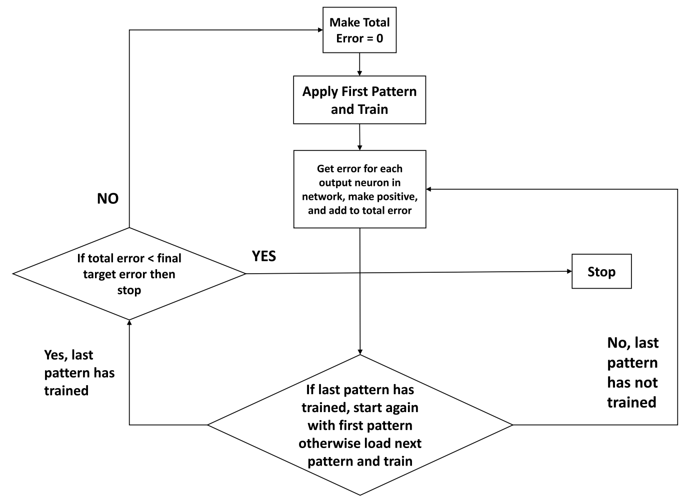
where the net input is the total generation for 12 months.

Fig. 6 shows the error graph of the BPNN training process. It can be observed from the figure that error value decreases as the training iteration number increases due to Backpropagation process. Error values shown in Fig. 6 are in normalized form. The algorithm is trained by setting the iteration value to 4000. Each set of input data is trained 4000 times, but the designed framework is tested by varying the iteration number from 2600 to 12000. Error value remains at a constant level after training is done 2800 times. Iteration value is kept at 4000 by performing an optimum tradeoff between simulation accuracy and speed. Although the algorithm has the risk of overtraining and/or local maxima and minima, the statistical normalization would reduce these kinds of risks.

Algorithm 3 shows the flow chart for BPNN. Datasets are divided into two sets: training set and testing set.

Algorithm 3 *(Flow Chart for BPNN algorithm).*

1: ***Input:*** *Training Datasets in structured format*
2: ***Output:*** *Electricity Generation Forecast Model: BPNN*
// ***Procedure***



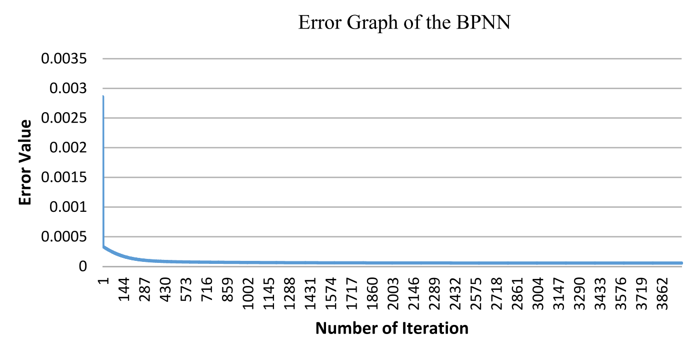


Fig. 6. BPNN error graph.

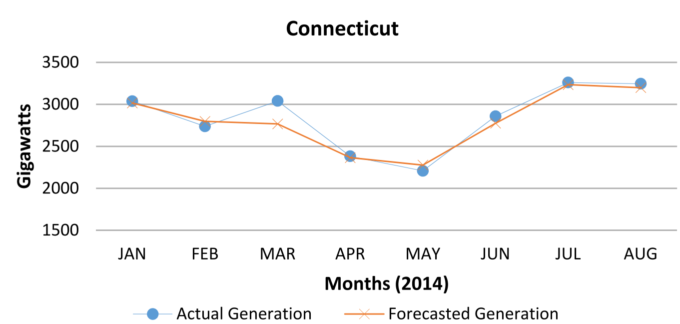


Fig. 7. Actual and predicted forecast for Connecticut. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Although the BP algorithm has problems such as local minima and overtraining, yet it is a good choice for its outstanding performance, and once it is successfully trained, it has the ability to detect patterns with excessive noise.

4. Results and analysis

In this section performance of the algorithm is evaluated and the output forecasts are compared with actual generation. The forecasts are performed for individual states and the collected values are used to find the total generation. The total generation is also separately forecasted using net generation data. The results are presented for three states with different climates and different energy demands. Figs. 7, 8, 9, and 10 show forecast with actual generation for the three states and total U.S. generation.

Figs. 7, 8, and 9 show results for Connecticut, Texas and California respectively; the blue line shows actual generation, and the

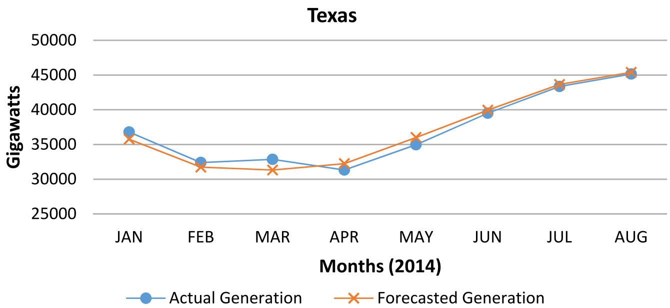


Fig. 8. Actual and predicted forecast for Texas. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

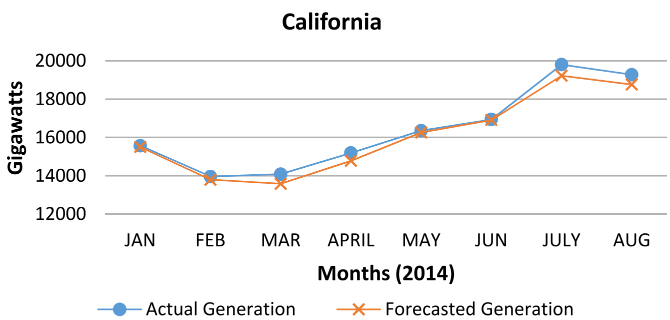


Fig. 9. Actual and predicted forecast for California. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

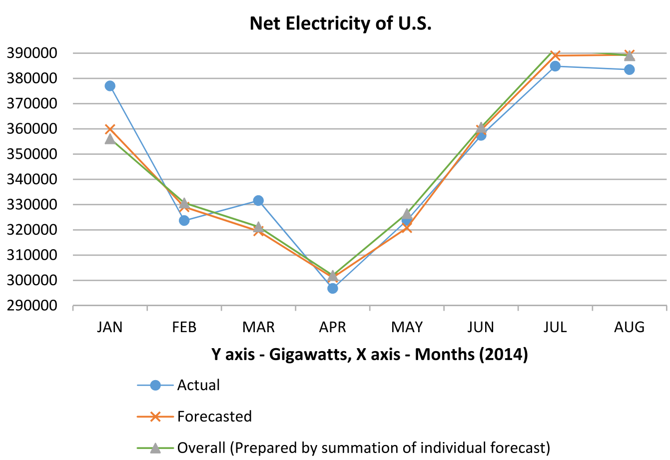


Fig. 10. Actual and predicted total electricity forecast in the U.S.A. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

orange line indicates the forecast. It is observed from the figures that the forecasted values closely match those of the actual generations.

Fig. 10 depicts the total generation forecast for the U.S. The blue line at the top shows actual generation and the red line in the middle indicates our forecasted generation, by summing all individual state generation forecasts. The green line at the bottom represents the overall forecast, prepared by sum of all the states' actual generation using the summed data to forecast the overall generation. In both cases (red and blue lines) we can see our forecast provides a close match to the actual generation.

The results show that the forecasted values for power generation closely match those of the actual measurements. This indicates that the system can recognize the data pattern properly and forecast the values accurately. The BPNN performance is reliable once it has been successfully trained. In the proposed framework, the network is properly trained, and the error rate is minimized. The network can accurately forecast from noisy input, and it has the capability to detect abnormal demand from forecast results after learning from examples.

The results show a close proximity between forecasted and actual data. In order to verify the results, an analytical approach is applied. The Mean Absolute Percentage Error (MAPE) is calculated for the developed BPNN model results. MAPE is a measure of accuracy of a method to construct forecasting values of a certain time series. It expresses accuracy as a percentage, defined by Equation (3).

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{3}$$

where A_t is the actual value and F_t is the forecasted value.

MAPE is calculated for both the total generation forecast and some individual state's forecasted results. MAPE percentage was calculated to be 4.13% for total generation forecast and the individual forecast values are in the range of 4–9% for all the states. The normalization process has an impact on MAPE percentage. Normalization reduces the error and helps ML methods to learn more quickly. It also reduces the mean absolute error. Hence, the performance of BPNN algorithm satisfies both analytical and forecasted results.

The main contributions of this work are as follows: 1) A BP algorithm is implemented and shown to be able to efficiently deal with large datasets by means of simulation results; 2) The problems with dealing with large datasets are solved with the Big Data approach; 3) The ML approach with Big Data is integrated and shown to be a viable forecasting solution when dealing with large datasets having complex noise.

5. Conclusion

In this study, a ML scheme is implemented to deal with Big Data analytics. Big Data has the ability to deal with large datasets in different formats, hence a suitable solution for analytics. ML combined with Big Data is a novel approach to solve a complex problem related to power generation prediction. Electricity generation forecasting is a challenging issue, especially when one is dealing with a large dataset complemented with noise. Experimental results of this work have been compared to predicted future power generations, and it provides a close match between their respective values. The role of Big Data approach is to extract the desired statistical features from the data using a distributed algorithm in the form of MapReduce on high performance platform and applied to ANN to find a relationship or specific patterns in the data. This relationship is used to forecast future generations. The results show a close proximity between the forecasted and the actual power generation values.

In future we plan to add other metrics such as load in our analysis. There are numerous studies in the literature on load forecasting as it is extremely important for the operation and planning of utility companies [28]. Considering accuracy of integrating the ML with Big Data analytics in forecasting, the proposed strategy will be further developed to predict the load demands in future. Accurate prediction will provide a clear picture for power system operators to effectively dispatch the electricity generation, reduce power losses, and enhance the energy security.

References

[1] Wikipedia contributor (2013, Dec 21), List of countries by electricity consumption (version ID: 634536654) [online], available: http://en.wikipedia.org/wiki/List_of_countries_by_electricity_consumption.

[2] Wikipedia contributor (2014, Oct 15), Electricity sector of the United States (version ID: 623351676) [online], available: http://en.wikipedia.org/wiki/Electricity_sector_of_the_United_States.

[3] Wikipedia contributor (2014, Jan 5), Energy in United States (version ID: 633059979) [online], available: http://en.wikipedia.org/wiki/Energy_in_the_United_States.

[4] Wikipedia contributor (2014, April 20), Artificial Neural Network (version ID: 174595685) [online], available: http://en.wikipedia.org/wiki/Artificial_neural_network.

[5] Electricity datasets from Energy Information Administration, available on web link: http://www.eia.gov/beta/api/qb.cfm?category=0, Web – 20 Jan, 2014.

[6] Wikipedia contributor (2014, Jan 10), Big data (version ID: 409751071) [online], available: http://en.wikipedia.org/wiki/Big_data.

[7] Wikipedia contributor (2013, Nov 23), Machine Learning (version ID: 175534373) [online], available: http://en.wikipedia.org/wiki/Machine_learning, 2013, Web – 12 Jan, 2014.

[8] Wikipedia contributor (2016, Jan 16) [online], Cognitive Science (version ID: 699869928) [online], available: https://en.wikipedia.org/wiki/Cognitive_science.

[9] H.C. Yang, A. Dasdan, R.L. Hsiao, D.S. Parker, Map-Reduce-merge: simplified relational data processing on large clusters, in: Proc. 2007 International Conference on Management of Data (SIGMOD), 2007, pp. 1029–1040.

[10] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large cluster, Commun. ACM 51 (2008) 107–113.

[11] D. Wang, X. Liu, M. Wang, A DT-SVM strategy for stock futures prediction with Big Data, in: Proc. 2013 IEEE 16th International Conference on Computational Science and Engineering, 2013, pp. 1005–1012.

[12] T. Jayalakshmi, Dr.A. Santhakumaran, Statistical normalization and back propagation for classification, Int. J. Comput. Theory Eng. 3 (1) (February 2011) 1793–8201.

[13] Data normalization for statistical analysis, web link: http://www.damienfrancois.be/blog/pivot/entry.php?id=8, 13 Jan 2014.

[14] Wikipedia contributor (2014, Jan 10), Backpropagation (version ID: 634418284) [online], available: http://en.wikipedia.org/wiki/Backpropagation.

[15] P.C. Chang, C.Y. Fan, J.C. Hsieh, A weighted evolving fuzzy neural network for electricity demand forecasting, in: 2009 First Asian Conference on Intelligent Information and Database System, vol. 9, 2009, pp. 330–350.

[16] Y. Xie, M. Li, Application of gray forecasting model optimized by genetic algorithm in electricity demand forecasting, in: Second International Conference on Computer Modeling and Simulation, vol. 10, 2010, pp. 275–277.

[17] D.X. Niu, Y. Wei, J. Li, J. Wu, Analysis of electricity demand forecasting in Inner Mongolia based on gray Markov model, in: International Conference on E-Business and E-Government, vol. 10, 2010, pp. 5082–5085.

[18] A. Setaiwan, I. Koprinska, V.G. Agelidis, Very short-term electricity load demand forecasting using support vector regression, in: International Joint Conference on Neural Network, Atlanta, Georgia, USA, vol. 1, 2009, pp. 2888–2894.

[19] H. Wang, S. Zhu, J. Zhao, G. Li, An improved combined model for the electricity demand forecasting, in: International Conference on Computational and Information Sciences, vol. 6, 2010, pp. 107–111.

[20] J.N. Fidalgo, M.A. Matos, Forecasting Portugal global load with Artificial Neural Network, LNCS 4669 (2) (2007) 728–737.

[21] P.F. Pai, W.C. Hong, Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms, Electr. Power Syst. Res. 74 (3) (2005) 417–425.

[22] P.C. Chang, Y.W. Wang, C.H. Liu, Fuzzy Delhi and Backpropagation model for sales forecasting in PCB industry, Expert Syst. Appl. 30 (4) (2006) 715–726.

[23] M.C. Su, C.W. Liu, S.S. Stay, Neural Network based Fuzzy Model and its application transient stability prediction in power system, IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. 29 (1) (1999) 149–157.

[24] R.H. Liang, Application of gray relation analysis to hydroelectric generation scheduling, Int. J. Electr. Power Energy Syst. 21 (5) (1999) 357–364.

[25] X. Wu, X. Zhu, G. Wu, W. Ding, Data mining with Big Data, IEEE Trans. Knowl. Data Eng. 26 (1) (Jan 2014).

[26] Wikipedia contributor (2015, May 10), List of Machine Learning concepts (version ID: 685871451) [online], available: http://en.wikipedia.org/wiki/List_of_machine_learning_concepts.

[27] Wikipedia contributor (2015, April 12), Data Science (version ID: 7000450339) [online], available: http://en.wikipedia.org/wiki/Data_science.

[28] T. Senjyu, H. Taakara, K. Uezato, T. Funabashi, One-hour-ahead load forecasting using neutral network, IEEE Trans. Power Syst. 17 (1) (2002) 113–118.

[29] K. Orwig, M. Ahlstrom, V. Banunarayanan, J. Sharp, J. Wilczak, J. Freedman, S. Haupt, J. Cline, O. Bartholomy, H. Hamann, B. Hodge, C. Finley, D. Nakafuji, J. Peterson, D. Maggio, M. Marquis, Recent trends in variable generation forecasting and its value to the power system, IEEE Trans. Sustain. Energy (2015) 1–10.

[30] C. Wan, Z. Xu, P. Pinson, Z. Dong, K. Wong, Probabilistic forecasting of wind power generation using extreme learning machine, IEEE Trans. Power Syst. 29 (3) (2014) 1033–1044.

[31] Jeff Heaton, Introduction to Neural Network for Java, 2nd edition, Heaton Research, ISBN 1604390085, October 2008.

[32] MapReduce tutorial (2015, Dec 18), The Apache Software Foundation [online], available: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html.