Senior Design Report

Graphic User's Interface Design
for Electricity Load Prediction of New York State

Yilan Zhu

Dr. Junhui Zhao
Dr. Bijan Karimi

April 2016

EXCUTIVE SUMMARY

This project demonstrates building a prediction model for electricity consumption of different areas of New York State with MATLAB. Artificial Neural Network is used as non-linear regression model for the electricity load prediction, based on hourly historical load and temperature information. A graphic user's interface is designed for user to determine the years to select for training, the area of forecast and the electricity load for a specific date. Accuracy percentage will be displayed after the network is trained and tested.

The network model is trained with hourly electricity load data from NYISO (New York Independent System Operations) of different zones. The entire New York state is divided up into 11 zones, only 6 of them can be chosen for prediction. Users can choose up to from year 2008 to 2014 as the input training data. Hourly temperature corresponding to the load zones are also considered as training information and have proved to be more effective.

To improve the performance of the prediction system, big data tools are also utilized when obtaining hourly temperature data. With ample built-in functions that MATLAB provide, the models are shown to produce highly accurate day electricity load, with accuracy up to 95-96%.

ACKNOWLEDGEMENTS

I would like to thank all those who have given their supported, and have contributed to my education as engineers. This project is based on all the support from the University of New Haven, Tagliatela College of Engineering, Department of Electrical & Computer Engineering and Computer Science. The department has generously provided much support in terms of the equipment and the outstanding faculties.

Special thanks to our supervisor Dr. Junhui Zhao for his guidance and constant supervision. I am grateful to Dr. Zhao as well as for providing the essential information for initiation of the project.

Dr. Bijan Karimi has also been helpful for helping me grasp a better understanding of neural network and how it should accurately apply to the forecasting system.

Acknowledgement to Rahman, Mohammad Naimur, an Electrical Engineering graduate student at the University of New Haven, who showed me the ropes on how to utilize Hadoop for effective data normalization.

Last but not least, I would also like to thank Mark Morton for the final printing of the project po

TABLE OF CONTENTS

1. INTRODUCTION

Compared to the lifestyles that humans had a few decades ago, one of the most obvious change would be electricity energy consumption. According to publications by the International Energy Agency (IEA), the total amount of electricity consumption in the world of 2012 was more than four times the amount of electricity consumed in 1971. With the massive demand of electricity for almost every area of life, it would not be difficult to predict that the total amount of electricity consumed for the future would still be growing.

Therefore, electricity load prediction would play a crucial role for utility planning and management. Though there exist inventory to store and buffer electricity in many forms, for most electricity consumers, electricity has to be generated and delivered as soon as it is consumed. To minimize unnecessary resource loss, electricity providing industries and utilities have to balance supply and demand for every second. Industries responsible for smooth operation of the grid need accurate load forecast, as when utilities and system operators feed into a number of their down steam components, predictions would be necessary in aspects such as resource scheduling, and transmission or distribution systems[1] .

The same concept applies for commercial electricity customers, such as households, restaurants, hospitals, hotels and educational institutes. These consumers' electrical consumption pattern depends mostly on annual schedules (holidays/ academic schedules etc.), environmental issues (weather/ temperature/extreme weather conditions), and revenue levels. On the financial side, electricity consumers doing electricity markets must be able to forecast the electricity load and prices, so generators and power markets need electricity load forecasts to represent a key or important element for a more accurate financial evaluation and management.

Compared to using commercial software, consulting a third party, and in house development using programming languages, MATLAB has the advantage of providing a more transparent construction of the prediction models. This indicates that more insights could be provided on how the predicted results are being generated and it would be easier to update or modify the prediction system to customize for the needs of the clients in a more economic manner.

2. PROJECT DEFINITION

2.1.Project Statement

In this project, a non-linear regression model is constructed to forecast the day electricity load for a given date, based on provided historical hourly loads of the entire New York state, divided into 6 parts. As weather is one of the prime factor that affects electricity usage, 6 corresponding weather observation stations were selected to apply the hourly temperature as another set of training dataset. Neural Network is the main algorithm used for this prediction model. The prediction network is trained with annual hourly data, up to from 2008 to 2014 and tested on real data from 2015. The models are shown to produce highly accuracy load forecasts with average accuracy up to 90%.

This design project contains a massive amount of research work, learning about new algorithms and getting familiar with them, at the same time applying the acquired knowledge to processing the data to develop a prediction mathematical model.

2.2.Summary of Literature Research

When referring to load forecasting, besides electricity consumption, the term is also used in electric power utilities, including generation, transmission and distribution. Due to the critical role of load forecasting in electrical industry plays, load forecasting provides a key input to power systems operations and planning, inaccurate load forecasts can lead to fatal accidents or even system-wide blackout [1]. Meanwhile, issues with electricity storage and the exponentially increasing necessity of electricity, plus the complex seasonal patterns, the need to be extremely accurate[1].

Models and algorithms based on Artificial Intelligence techniques, such as artificial neural network (ANN), fuzzy logic, and support vector machine, were black-box models that appealed to organizations unwilling to build an in-house team of forecasting analysts. Many utilities were still not comfortable with black-box approaches and instead developed forecasts using the classical methods such as the similar day method, and statistical techniques such as multiple linear regression. Some utilities with an in-house forecasting team also built black-box models or purchased forecasts for comparison purposes. Many companies turned to prediction software such as SAS, Itron and LoadSEER. However, these mentioned software tend to only typically offer 80-90% of the functionality necessary by a utility, such as considering regional loads, different weather patterns and so on.

Compared to MATLAB, which provide the flexibility of building a completely customized load forecasting system. And due to the ample selection of functions that could be utilized, and the easiness to access to data needed, the time and cost to develop such a system is also dramatically lower than using commercial software for a customized prediction system.

3. EVALUATION OF ALTERNATIVES

In order to construct the prediction system, there were quite a few alternatives to choose from to obtain similar results. The reason for the selection of each method, technique or tool and their corresponding alternative would be explained in detail.

3.1.Alternative for programming software MATLAB

As the main outcome of this project is to deliver a Graphic User's Interface for users to obtain their desired forecast electricity load for a specific date, there were multiple platforms to use: Fireworks form Adobe, Sketch and Photoshop etc. However, the main issue for this project is to link the acquired data together and set as an input for an Artificial Neural Network algorithm. The mentioned software would construc t neat and pretty user interfaces, however, I would not be able to send data in and out of the User Interface design software easily as MATLAB could.

Another reason for choosing MATLAB would be, as mentioned before, its wide variety of functions that could be utilized simply by formatting the inputs as required. This would save me a lot of time and effort constructing my own algorithms.

3.2.Alternative for prediction algorithm Artificial Neural Network

Besides Artificial Neural Network, the most commonly used algorithm for pattern recognition and forecasting, Multiple Regression and Autoregressive Integrated Moving Average could also be used for the same scenario.

The advantage of using regression method are that regression methods are interpretable; easy to implement, update, and automate. However, this technique relies on explanatory variables (a designated functional form), which would require at least two years of history to make a prediction.

Autoregressive Integrated Moving Average (ARIMA) only has few parameters to estimate. To make an accurate this method does not require a long history. The ARIMA can be utilized for short term prediction, such as day ahead prediction. However, when it comes to long term prediction and including more inputs such as temperature, this algorithm will result in low accuracy forecast. Another drawback for this technique is its high cost to implement, update, and automate [2].

For Artificial Neural Network (ANN), we do not require too much statistical or domain knowledge to interpret how the system works. Though having disadvantages such as heavy computation; overparameterization and low accuracy during extreme weather conditions, ANN is certainly the best tool for computing the forecast of a particular 24 hour load during a typical day.

4. DESIGN APPROACH

The entire forecast system can be divided into parts: data collection, data import, Neural Network, MapReduce and GUI.

4.1.Overall Structure

Data collection is the very initial step for the entire project. Acquiring electricity load data from NYISO and Mesowest, arrays that contain month, date, day, hour and air temperature are formatted and normalized as the training set. Hourly electricity load values are set into arrays of the same size (number of rows) as the training sets.

After determine the type of ANN algorithm for evaluation and have the parameters set for best network performance, the network is initialized and configured.

The network would be trained multiple times using every year's training dataset as the input and the corresponding hourly electricity load as the target.

The trained network would predict a year's length of hourly electricity load. By comparing the predicted values with the real numbers of 2015, we could calculate the mean absolute percentage error (MAPE), which would be the main criterion for testing the accuracy of the prediction system.

Based on input information (area, the number of years for training and date for prediction) by the user, forecast and the effectiveness of the system is presented. This information would be displayed on the user interface. Detailed information about each section is explained below.

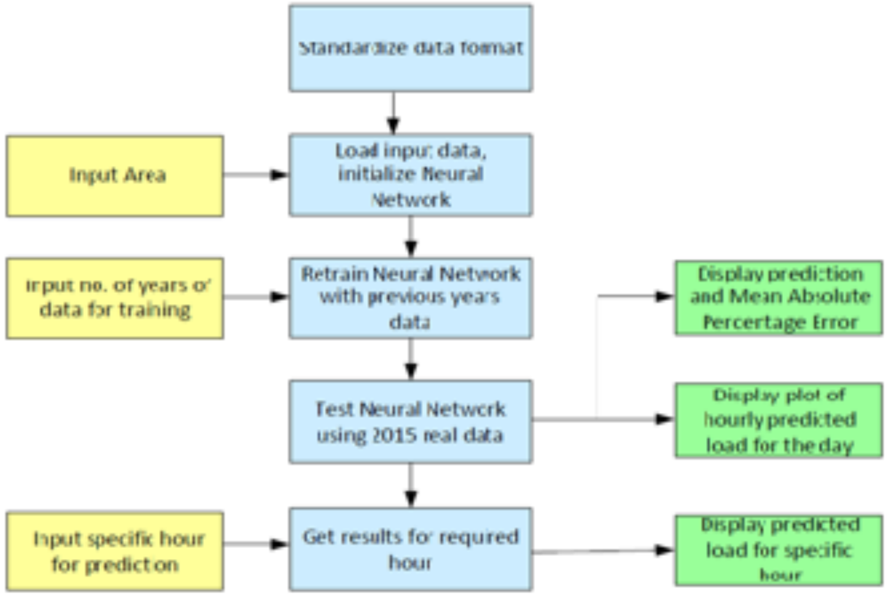


Fig 1. Overall structure of electricity load forecasting system

4.2.Data Acquisition

Electricity load hourly data is obtained from New York Independent System Operator (NYISO).

(Available at nyiso.com/public/markets_operations/market_data/load_data/index.jsp). Records of the dataset have been collected and downloaded from 2008 January to December 2015. This contains detailed data of the hourly consumption of eleven primary substations with corresponding electrical transmission owners (CAPITL, CENTRL, DUNWOD, GENESE, HUDVL, LONGIL, MHKVL, MILLWD, N.Y.C, NORTH and WEST).

Hourly Air Temperature data- Mesowest API. This is an online database for a collection of all the accessible weather observation stations throughout the United States, available at http://mesowest.utah.edu/. Mesowest's API is web services that provides data in response to specific requests (query string parameters). The requested dataset could be obtained by entering the accepted variables in a given format as an URL in a web browser's navigation bar. As electricity usage is closely related to air temperature, observation of hourly air temperature is acquired from Mesowest's API and utilized as a training set for the neural network.

Six areas are selected to test and train the prediction model. The corresponding weather observation station are also selected based on their location. The following states the names ‘KSYR’ for area ‘CENTRL’, ‘KALB’ for area ‘HDVL’, ‘KISP’ for area ‘LONGIL’, ‘KART’ for area ‘MHKVL’, ‘KNYC’ for area ‘NYC’, ‘KIAG’ for area ‘WEST’.

Load Area Name	Load Area ID in diagram	Weather Station Name/Location	Weather Station ID
WEST	A	Niagara Falls International Airport	KIAG
CENTRL	C	Syracuse Hancock International	KSYR
MHKVL	E	Watertown International Airport	KART
HDVL	G	Albany International Airport	KALB
LONGIL	K	Long Island Mac Arthur	KISP
NYC	J	New York City, Cent	KNYC

Table 1. Table for electricity load zones assigned by NYISO and their corresponding weather stations [3]

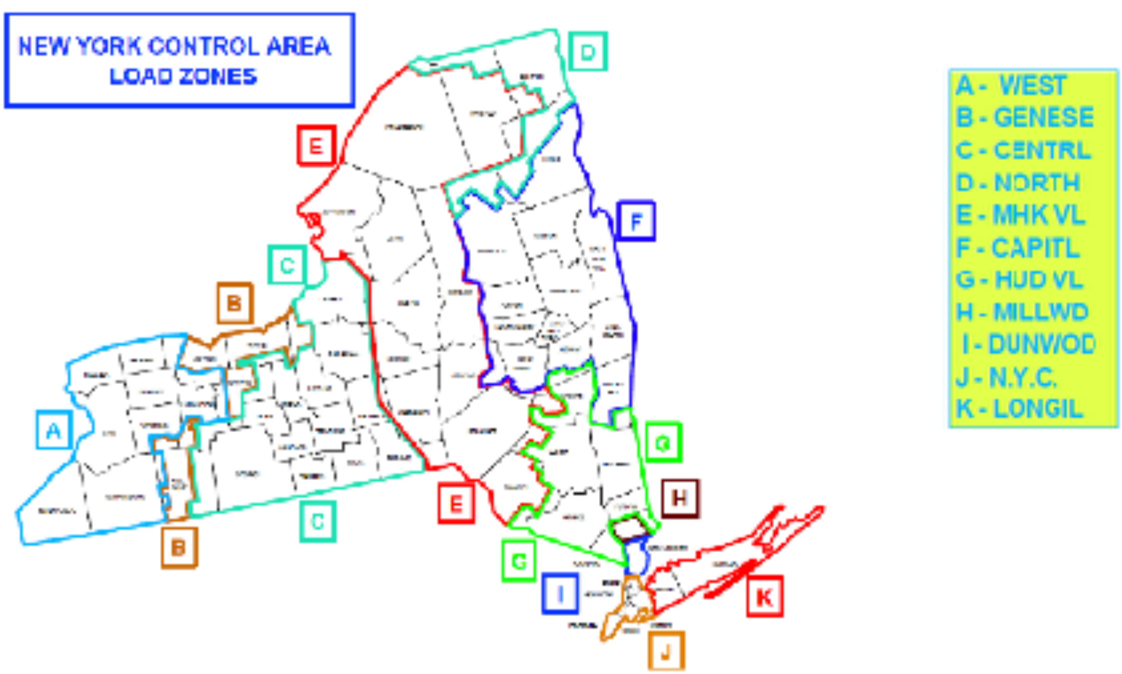


Fig 2. Load areas of New York State by NYISO

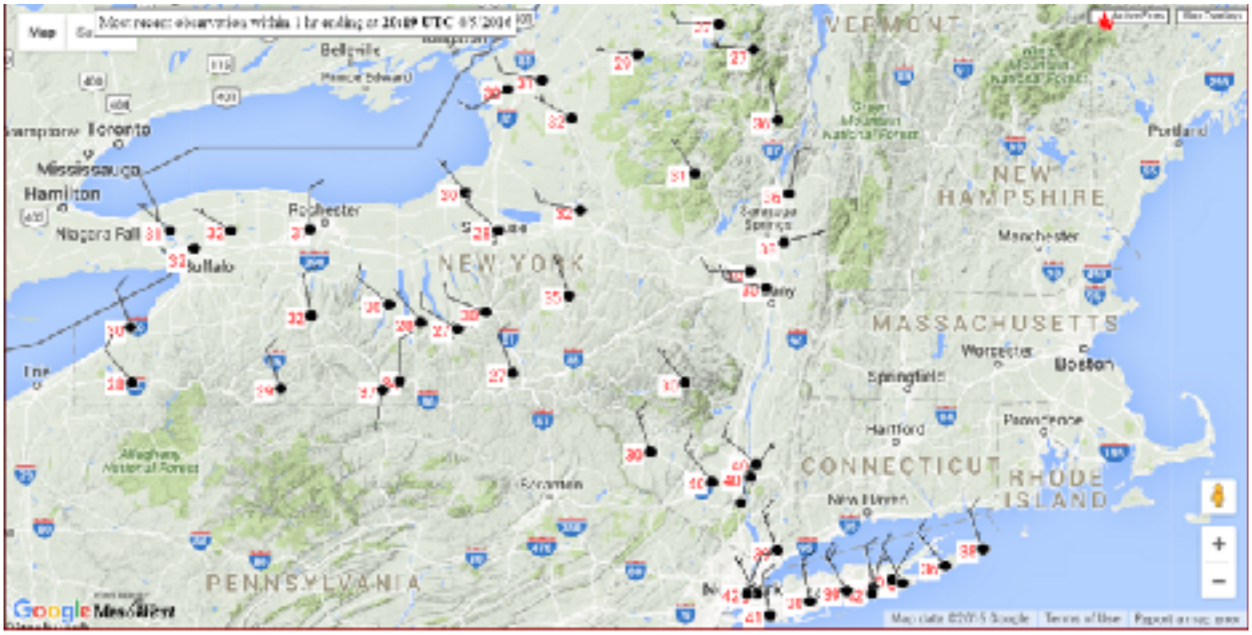


Fig 3. New York State with weather observation stations from Mesowest

4.3.Data Import

As mentioned, hourly electricity load data are downloaded to local drives via NYISO. As the acquired dataset are all stored in excel spreadsheets of CSV format, it could be easily imported into MATLAB using data importing functions such as ‘readtable’. As the provided information are given on a daily basis, each spreadsheet contains 24-hour electricity load for the 11 mentioned region. The ‘**LoadData.m**’ function is designed to extract the given information according to the names of the regions that is designated to be used as the training set. Multiple loop functions are also used to generate the month, date, day and hour arrays, that would be used for the Neural Network training set.

Air temperature data is acquired using the online database via the mentioned Mesowest API. Instead of downloading the datasets manually, the **‘weathernormal.m**’ function collects information directly from the API and store them as array variables in MATLAB. This could be accomplished by changing the query URL that is used to request the temperature dataset, and by the ‘websave’ function (only available in MATLAB version 2014b or higher) hour air temperature were extracted directly from the web service API. The obtained hourly temperature would then be collected and stored in a datastore specifically for large collections of data, then being normalized into values [-1,1] for better neural network performance using built-in MATLAB Big Data Tools MapReduce.

4.4.Training and testing Neural Network

The built-in MATLAB toolbox, Neural Network Toolbox, uses the network object to store all of the information that defines a neural network. The basic components of a neural network and s how they are created and stored in the network object for our electricity load forecast system are demonstrated as follows:

1. Collect data. Arrays that contain month, date, day, hour and air temperature are formatted and normalized as the training set. Hourly electricity load values are set into arrays of the same size (number of rows) as the training set as the target dataset.
2. Initialize and configure the network. Determine the type of algorithm used and have the parameters set for best network performance.
3. Train the network. The network would be trained using every year's training dataset as the input and the corresponding hourly electricity load as the target. For example, if the starting year is set to be 2010, the finishing year set to be 2014, then the network would be trained 5 times.
4. Validate the network. The trained network would predict a year's length of hourly electricity load. Adding the 24 hours' prediction we could have a day's load prediction. By comparing the predicted values with the real numbers of 2015, we could calculate the mean absolute percentage error (MAPE), which would be the main criteria for validating the accuracy of the trained network.
5. Use the network. Once the network is trained and validated with low error, we could make predictions for electricity load with a given month, date, day, hour and temperature.

4.5.MapReduce

MapReduce is a programming technique for analyzing data sets that do not fit in the memory. The map and reduce function is used inside the '**weathernormal.m'** for normalizing the extracted temperature data from the API and normalizing them to a range of [-1,1] to boost the accuracy and the training speed of the neural network.

Mapreduce includes a map.m function and a reduce.m function. The map function takes in chunks of data in the forms of datastore, and that outputs certain intermediate results. The reduce function would then read the intermediate results from the map function and produces a final result. For the map and reduce function that is used for this project (mapper.m and reducer.m), the normalized data is calculated by the following equation:

Nomalized array = ((array elements-minimum value of the entire array)/difference between the largest and smallest value of the array)-0.5)*2;

4.6.Graphic User Interface

There would be 4 required inputs, the starting and ending year selected for the training set ‘year_start’ and ‘year_end’, and the area code chosen from the six regions of New York State. A specific month/date would also be necessary. The output would be the accuracy of this prediction model and the corresponding electricity load daily for the given specific hour and temperature. A plot showing the predicted load and the real load would be displayed as well.

After choosing the specific years for starting, ending and the area for prediction, click on “Train Network” to training the Neural Network. After the network has been trained, by putting in the month and date and click on the “Show prediction” button, the accuracy and the predicted load will be shown on the text boxes on the right.

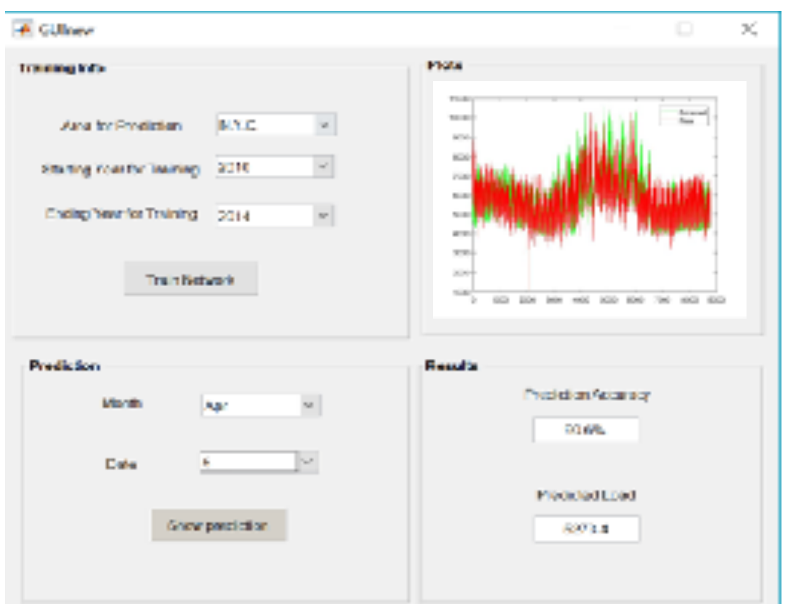


Fig 4. Completed GUI with predicted loads

MATLAB sub functions	Description

Main.m	Main function, input training variables, Neural Network parameters, output prediction and graph
Weathernormal.m	Hourly temperature data import
LoaData.m	
Mapper.m	Map function that break data into parts and produce intermediate results
Reducer.m	Reduce functions that take intermediate results and produce final results
GUI.m	Function that link variables with interface

Table 2. Table for different subfunctions and their functions

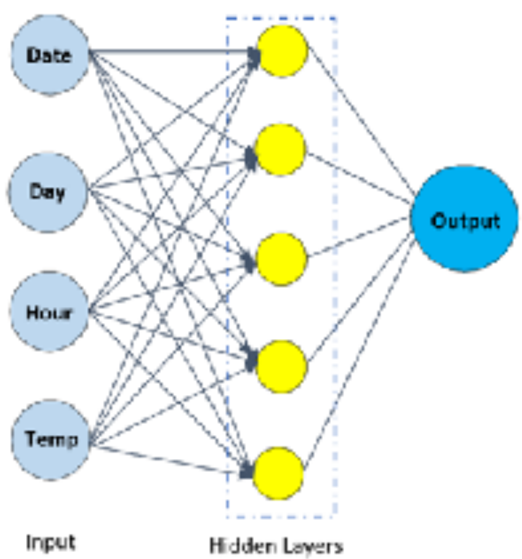
5. DESIGN VERIFICATION

5.1.Data normalization results

In order to save local space on the computer that the prediction model is running on, hourly air temperature data is collected via a compounded query string with the required variables and a 'websave' function built-in MATLAB. These numbers are then being normalized using the MapReduce function and minimizing it proportionally into a range of [-1, 1]. This is used to improve the performance of the Neural Network. At the early stage of the project, MapReduce was not implemented into the main code, and it could be observed and proved that not only the average error percentage were smaller after using MapReduce, the amount of time used to train the Neural Network was less by an average of 1 minute.

5.2.Prediction model performance result

The neural network algorithm structure used for this prediction model would be the feed-forward backpropagation network. For a given area, the input information consists of 3 elements: month, date, day, hourly air temperature. The target output is the hourly electrical load.



Input neurons – Month Date Hour Temperature

Number of neurons in hidden layers =

Empirical equations:

-

(*a* is an integral from 0 to 10)

- 2/3 input neurons + output neurons

Fig 5. Basic structure of neural network

1 output neurons –Hourly Electricity Load

In order to produce an accurate prediction model using Neural Network algorithm, the following two factors, number of years used as training set and the number of hidden layers, have been taken into major consideration and experimented on to achieve the best results.

a) Determining the number of years for training

As the pattern and the quantitate values for electricity consumption and changed greatly over the years, therefore we would not able to make the most accurate predictions if using too much information for the past. Meanwhile, using large data sets to train the model would be also more time consuming.

The parameters of the NN network used when determining the best duration for training data set are:

Maximum no. of iterations to train = 100

Number of validation checks = 50

Divide data randomly and ratio of training data/testing data = 85/15

No. of Hidden layers: 10

Error goal: 1e-5

No. of validation checks: 50

To test the accuracy of the prediction model, mean average error percentage (MAPE) is used as the criteria. The table below displays the mean average percentage error and the amount of time elapsed for each training and testing process. It could be concluded that using data sets of 2010-2014 would produce the most efficiency prediction system.

Electricity Load data included for training	Mean Average Percentage Error (%)	Time elapsed (s)
2008-2014	8.3374	78
2009-2014	7.6270	67
2010-2014	7.4473	62
2011-2014	7.4762	52
2012-2014	7.4610	47
2013-2014	7.8216	28

Table 3. MAPE of prediction network using different numbers of neurons in hidden layer

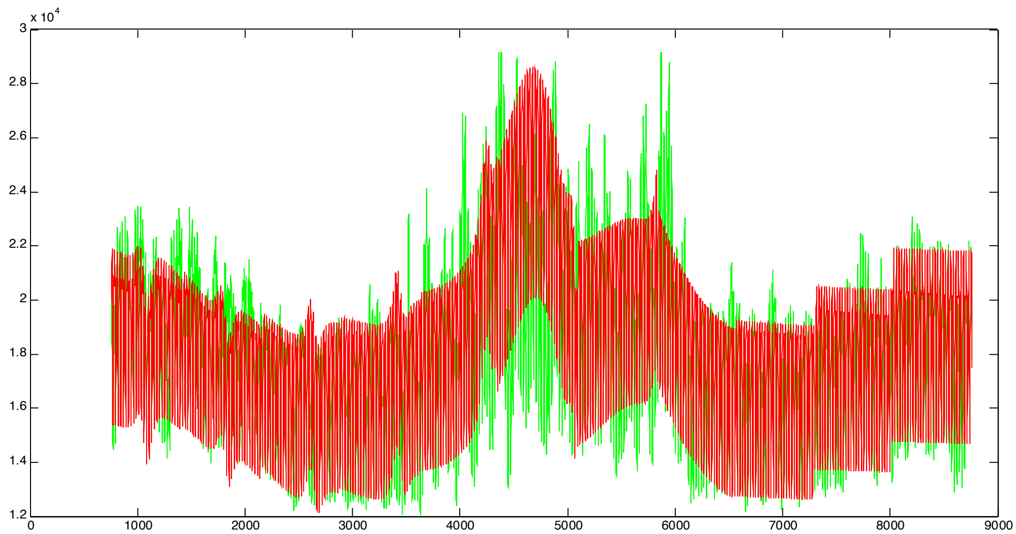


Fig. 6
Plot of actual electricity load vs. predicted electricity load of 2010-2014 (minimum MAPE)

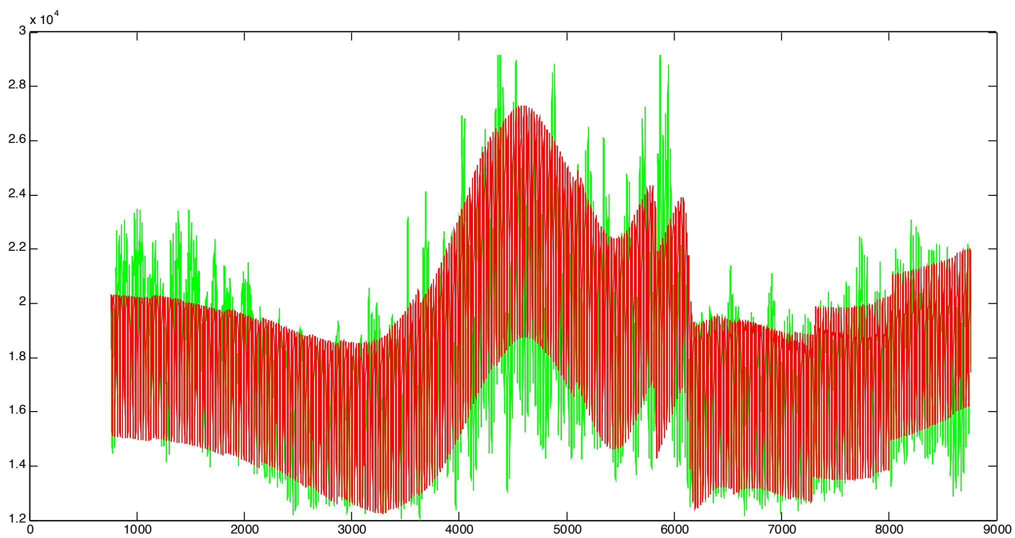


Fig. 7 Plot of actual electricity load vs. predicted electricity load of 2008-2014 (maximum MAPE)

b) Determining the Number of hidden layers for the Neural Network

Deciding the number of neurons in the hidden layers is a crucial factor to determine the overall neural network architecture. Though these layers do not directly interact with the external environment, they have a tremendous influence on the final output. Both the number of hidden layers and the number of neurons in each of these hidden layers must be carefully considered.

Using too few neurons in the hidden layers will result in underfitting, which occurs when there are not enough neurons in the hidden layer to adequately detect the signals in a complicated data set.

Correspondingly, using too many neurons in the hidden layers can result in overfitting. When the neural network has so much information processing capacity that the limited amount of information contained in the training set is not enough to train all of the neurons in the hidden layers. Moreover, using more neurons in the hidden layer will take more time to train the prediction network[2].

The amount of training time can increase to the point that it is impossible to adequately train the neural network. Obviously, some compromise must be reached between too many and too few neurons in the hidden layers.

Thus after deciding the best amount of data required for the best performance (2010-2014), this training dataset is then used to determine the numbers of neurons on the hidden layer should be used to produce predictions with lowest error percentage. The results for this experiment suggests that the most suitable number of neurons in the hidden layer would be 10.

No. of neurons in hidden layer	Mean Average Percentage Error (%)	Time elapsed (s)
1	10.2895	34
2	9.9312	36
4	8.1951	47
5	8.1201	56
6	7.9576	61
7	7.6842	65
8	7.8702	73
10	7.3157	84
11	7.5237	78
13	7.4380	88
15	7.6220	86
50	7.8638	142

Table 4. MAPE of prediction network using different numbers of neurons in hidden layer

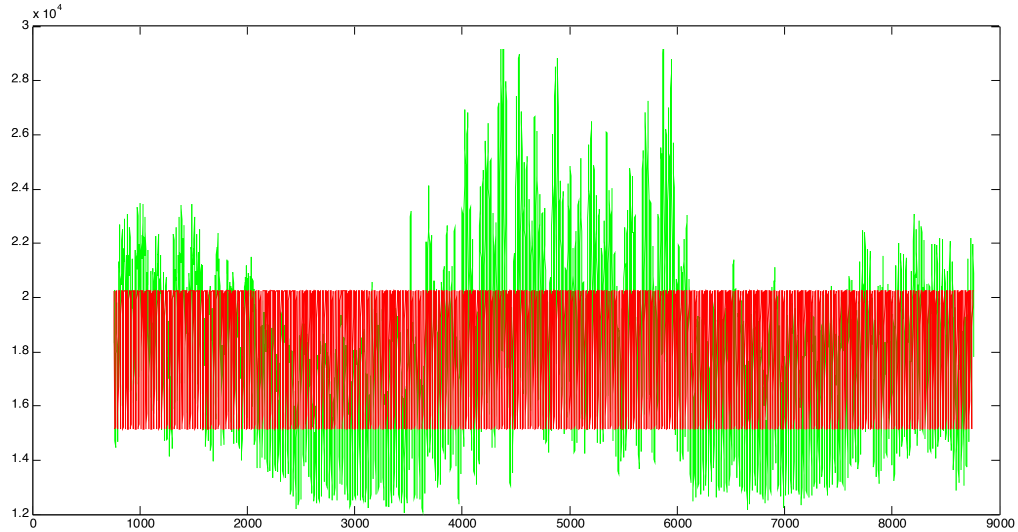


Fig. 8 Plot of actual electricity load vs. predicted electricity load for 2014 when using hidden layer n=1 (underfitting)

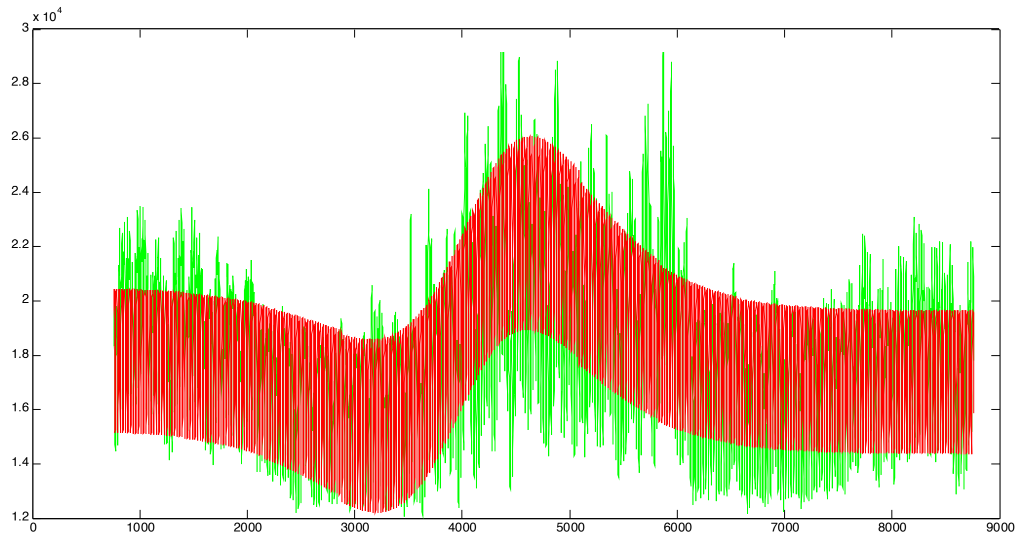


Fig 9 Plot of actual electricity load vs. predicted electricity load for 2014 when using hidden layer n=4 (starting to take shape)

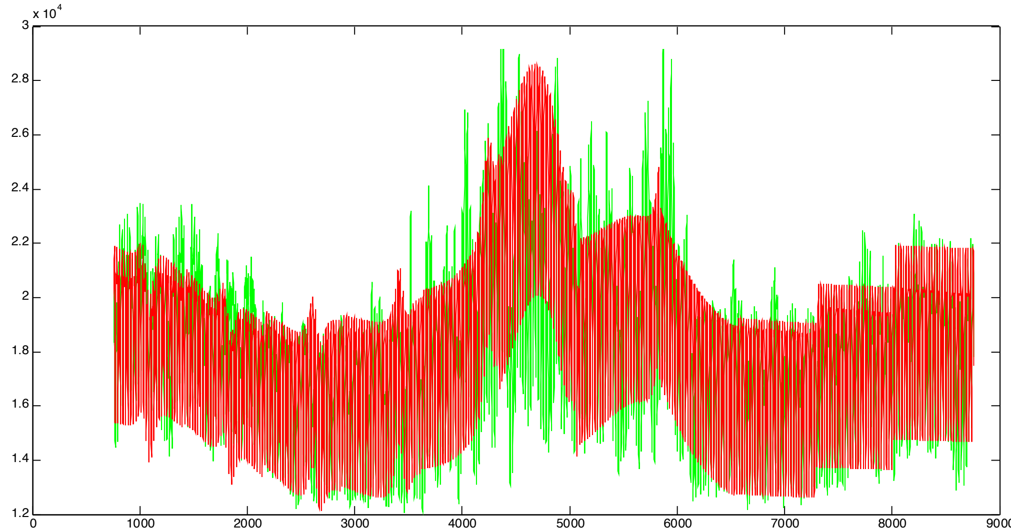


Fig 10 Plot of actual electricity load vs. predicted electricity load for 2014 when using hidden layer n=10 (minimum MAPE)

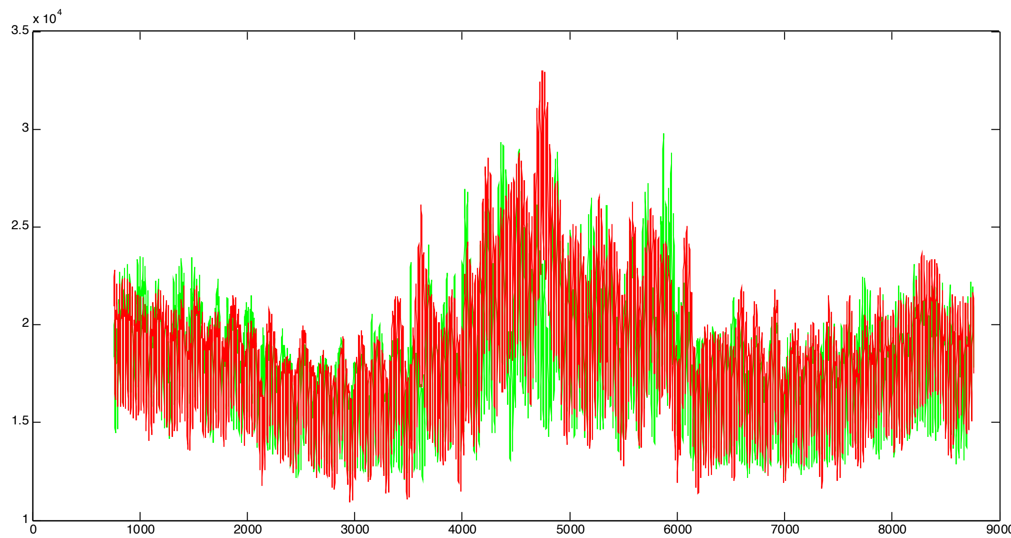


Fig 11 Plot of actual electricity load vs. predicted electricity load for 2014 when using hidden layer n=50 (overfitting)

c) Overall performance of prediction model

The overall accuracy of the prediction model when forecasting a 24-hour electricity load is 92%-93%. The reason for the fluctuation of the accuracy is that MATLAB randomly assign weights of the neuron of the neural network.

6. PROFESSIONAL AND ECONOMIC EVALUATION

The forecasting system is a useful tool when provided with historic data and relevant information that can help determine and recognize the pattern of the data to be predicted. However, neural network, after all, cannot be used for prediction. On the one hand, there is no guarantee that what we predict for the future would be absolutely correct: even if our prediction system can make predictions up to 99.9999%, there is still a 0.0001% possibility that things might not go as what we predict. On the other hand, the neural network toolbox that was utilized in this project had a “target input”. In other words, the neural network had the “solutions” to the questions called “future electricity load”, then it went backwards to do the math to figure out how to get from the solutions to the questions. Neural network is more of a pattern recognition algorithm than a prediction algorithm. However, it is still being widely used for its high accuracy and easy implementation.

From an economic stand of view, this project did not cost anything in terms of material. The MATLAB software was provided to me at the beginning of the project, and the it was only the labor cost that took up the entire project.

7. DISCUSSION

Overall, the project was successfully completed and reached my initial goals. As the entire project was done only on MATLAB, I gained more experience and knowledge working with MATLAB and I am in awe of how amazing this piece of software is and its capabilities of more than just making plots of equations. Before I worked on this project with MATLAB, I had no idea that MATLAB could import data and spreadsheets straight from the internet, without having to download and save the files on the local drives. With so many built-in function that was available for MATLAB users, I could accomplish any sort of calculations based on the input formats. I also realized that importing data was more than simply putting down the number of columns and rows. With different format of data sheets to start with, notifying the "read" function which are the lines to skip, what is the symbol of the delimiter, which variable I should choose for my input and how many rows and columns do I need. These details build on and one can easily miss out one or two which would cause the entire MATLAB script to display twenty lines of errors.

Another detail I learned from this project is the effect of the Daylight saving time (DST), typically involves advancing clocks by an hour near the start of spring and adjusting clocks back in the fall, had on people's lives. The primary purpose of daylight saving time is to adjust human activity schedule during daytime to take advantage of the daylight hours during summer time. As I had to keep my hourly electricity load data and hours data consistent, it took me quite a while to figure out where the data was missing for the one hour on March 8 of 2015. I resolved this issue manually by duplicating the numbers OF THE 1AM to the 2AM. The same thing goes for leap years (2008/2012), where data could not be automatically input as arrays without MATLAB showing up errors.

There was another issue with reading the data from the Mesowest API directly to the MATLAB workspace. The API decided to add two lines of their data summary in string forms right after I completed my main code, causing my MapReduce code to be failing as I could not get the temperature numbers in due to the last two lines of characters. Again, this took me a long time to figure out what exactly was going on, and this issue was resolved again by just setting the starting symbol of the second last line as a comment symbol and tell MATLAB to ignore when reading the data from API.

I realize that the key of success to software development is the patience to sit down and go through every line of code even if one is confident that it would not be wrong. I have been wronged so many times and frustration do occur when the red lines in the MATLAB command window. Nevertheless, patience is the crucial element to writing a successful code.

8. CONCLUSIONS AND RECOMMENDATIONS

The completion of the project in time adds credit to the time and effort contributed to the electricity load forecasting system. The project is completed alone, which, on one hand, had brought on a lot of workload for one single person, but on the other hand I had more flexible schedule to work on the project whenever available. To mention it again, MATLAB is a powerful tool for industrial and academic use and I would definitely recommend it to any student and or researchers that have to deal with simulations based on any load of theorical calculations.

From the prediction and observing past years, it would not be difficult to conclude that the peak hour usage of the entire year would be in the summer in July, where air conditional has to be turned on during the highest temperature of the day. Although we would expect more electricity consumption during winter in New York where the temperatures are mostly below 0 degree Celsius, where heating electrical appliances should be used more, it was not even half of the peak hourly electricity consumption of a hot day in July. This might be because in New York state or most of the other states in America, electricity might not be the only source of energy. The increasing use of renewable energy such as solar power, wind power and nuclear power, might bring down the demand for electricity in the future. Therefore, when we were to do predictions for electricity consumption in future years, we might need to consider the amount of energy that these alternative resources provide, in order to get a more accurate prediction.

9. REFERENCES

[1] Tao Hong, January 15 2015, Load Forecasting Case Study, Eastern Interconnection States' Planning Council.

[2] Mirza Cilimkovic, "Neural Networks and Back Propagation Algorithm", Institute of Technology Blanchardstown. Blanchardstown Road North. Dublin 15. Ireland.

[3]New York Independent System Operations. Federal Energy Regulatory Commission. Website: http://www.nyiso.com/public/markets_operations/market_data/load_data/index.jsp