

Machine Learning for the Smart Grid

Iliana Voynichka

I. Setup

1. Introduction

The Smart Grid will be the power grid of the future. It will offer a two way communication flow, between consumers and providers, to ensure energy is distributed in the most efficient way. With the advance of the internet and technology in general, we have the ability to improve the traditional power grid and turn it into an intelligent, automated and distributed energy delivery network that will be able to assess the state of the grid in real time and adopt an appropriate mode of operation [1]. A key component of an efficient Smart Grid operation will be the accurate prediction of future supply and demand trends. Interest in developing accurate prediction models for energy demand has increased in recent years. Demand forecasting can be subdivided into three approaches – averaging models, statistical models and artificial intelligence models. Averaging models are simple models based on computing linear combinations of averages from similar days. These models are usually used by Utilities and Independent Service Operators (ISOs). Statistical models use regression analysis and time series analysis. Artificial intelligence (AI) models use techniques such as Artificial Neural Networks, pattern matching, expert systems, etc [2][3]. Effective and accurate prediction of energy demand is vital for the efficient and proper operation of the Smart Grid. The main goal of this project is to attempt to improve the current energy demand predicting techniques used by ISO New England (ISO NE) by exploring various statistical and AI models.

1.1 Data and Preprocessing

The data used for this project is publicly available from the Independent System Operator for New England (ISO NE) [4]. The selected dataset contains information pertaining to energy demand and pricing for the years 2013 and 2014 for the Boston area. The original dataset consists of 14 columns. For the purpose of the project, only the data pertaining to energy demand is used and the rest - which is related to pricing – is removed. In addition, the column representing time in the format *month/day/year*, is removed and replaced with two new columns – one for the month (1 (January) - 12 (December))and one for the day of week (1 (Monday) – 7 (Sunday)). The assumption is that energy usage will vary from month to month and also throughout the week. Also, energy usage throughout the year will exhibit different patterns based on the season. To account for this behavior an additional column is added representing the season (1- Winter; 2- Spring; 3-Summer; 4 Fall). Based on recent weather patterns in New England, the winter season is modeled to start in January and end in March and the fall season to start in October and End in December . The final format of the data consists of 7 columns and 14592 rows. Table 1 shows a breakdown of the available columns:

Table 1. Explanation of columns in the data set

Column	Meaning
Month	Number from 1 to 12 (1 – January, 12 – December)
Day Of Week	Number from 1 to 7 (1 – Monday, 7 – Sunday)
Hour	Number from 1 to 24 (1 – 1 am, 24 – 12 am)
Dew Point	Dew point(F) for the given hour as measured by Boston Weather Station
Demand	Actual Energy demand as determined by metering
Day Ahead Demand	The predicted demand for the specified time (determined by ISO NE)
Season	Number 1 to 4 (1 – Winter, 2 – Spring, 3- Summer, 4- Fall)

To get some intuition about the data, the actual energy demand is plotted as a time series for the whole year (Figure 1).

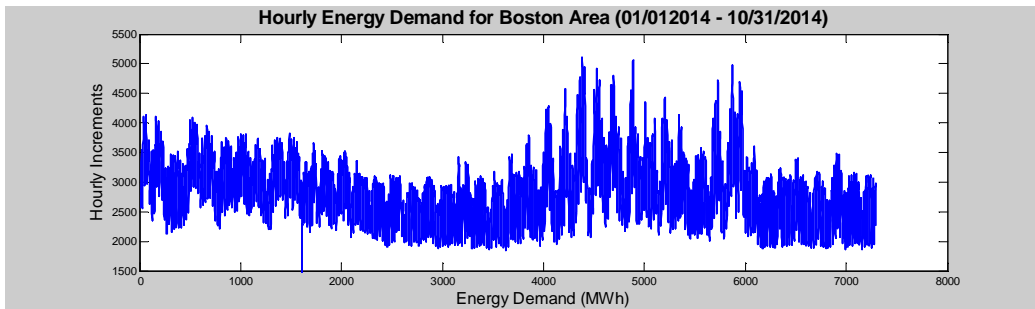


Figure 1. Actual energy demand for Boston Area (2014)

The energy demand seems to suddenly drop to 0 at $t=1610$. The rest of the data for that particular point turns out to be also 0. This is the only missing data in the dataset and the value for that point is generated by calculating the average for 5 points before that particular value and 5 points after the value. Figure 2. shows the resulting time series.

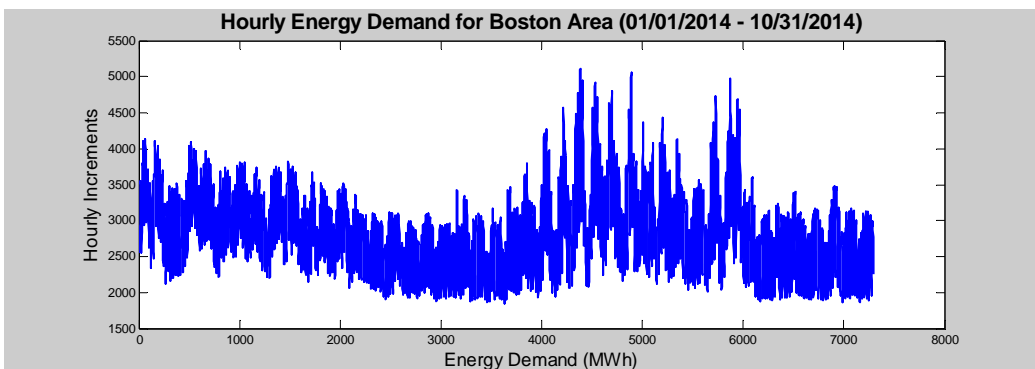


Figure 2. Actual energy demand for Boston Area (2014) with no missing values

The same process is applied to the data for the year 2013.

II. Method

1. Features Selection

Initially only four features are selected – month, day of week, hour and dew point. For particular models the season feature is added to evaluate its effect on the performance of the model. The target variable is the actual energy demand at a given point in time. The day ahead demand (refer to Table 1 for definition) is used to compare the models generated in this project to the model used by ISO NE.

2. Model Evaluation

The dataset is split in two – data containing 7296 observations for 2014 and data containing 7296 observations for 2013. The data from 2014 is used to train and evaluate the various models using 10-Fold Cross Validation. The models are scored based on average percentage error calculated by expressing the test error as a percentage of the actual value and calculating the average error across all 10 folds.

The model in each group that scores the highest is then trained on all the data from 2014 and applied to the data for 2013 to test how the model performs on a large and “unseen dataset”. Training on data from 2014 and testing it on 2013 might sound backwards, but it shouldn’t matter when it comes to evaluating models.

3. Models and Model Selection

Three main groups of models are evaluated against the ISO NE model – linear regression, linear regression with additive constant for each season, weighted linear regression and feed forward neural networks (NN).

2.1 Linear Regression

Seven linear regression models were evaluated on the 2014 dataset. The models are listed below starting with the simplest model that also has the worst performance. Figure 3 shows the comparison between the performances of the different models. None of the models outperform the ISO NE model. M1 is the best performing model among the group but also the most complex and runs the risk of over fitting. M3 is simpler and its performance is lagging by less than a percent (0.89%). M3 is chosen as the best performing model from the linear regression group.

Model 7 (M7) $Y = q_1x_1 + q_2x_2 + q_3x_3 + q_4x_4$

Model 6 (M6) $Y = q_1(x_1 * x_2) + q_1(x_3 * x_4)$

Model 5 (M5) $Y = q_1x_1 + q_3x_3 + q_4x_4 + q_2x_2^3 + q_5(x_1 * x_2) + q_6(x_3 * x_4)$

Model 4 (M4) $Y = q_1x_1 + q_2x_2 + q_4x_4 + q_4(x_1 * x_2) + q_3x_3^2$

Model 3 (M3) $Y = q_1x_1 + q_2x_2^2 + q_3x_3^3 + q_4x_4^4$

Model 2 (M2) $Y = q_1x_1 + q_2x_2 + q_3x_3 + q_4x_4 + q_5x_2^2 + q_6x_3^3 + q_7x_4^4 + q_8(x_1 * x_2) + q_9(x_1 * x_3) + q_{10}(x_1 * x_4) + q_{11}(x_2 * x_3) + q_{12}(x_2 * x_4) + q_{13}(x_3 * x_4)$

Model 1 (M1) $Y = q_1x_1 + q_2x_2 + q_3x_3 + q_4x_4 + q_5x_1^2 + q_6x_2^2 + q_7x_3^2 + q_8x_4^2 + q_9x_1^3 + q_{10}x_2^3 + q_{11}x_3^3 + q_{12}x_4^3 + q_{13}x_1^4 + q_{14}x_2^4 + q_{15}x_3^4 + q_{16}x_4^4$

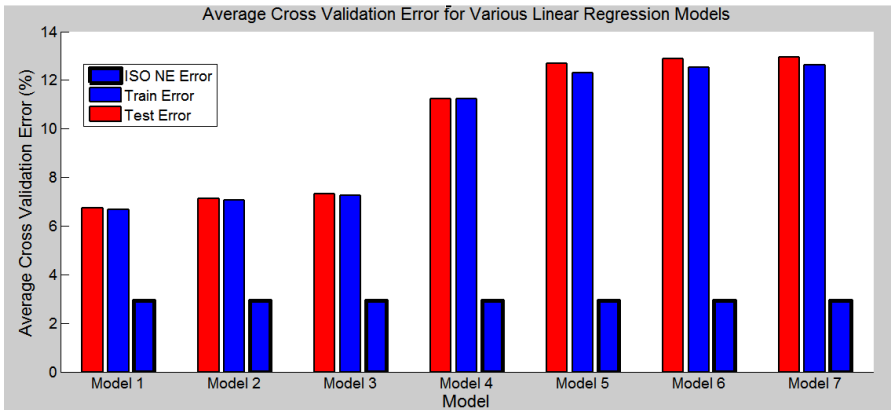


Figure 3. Average cross validation error for M1-M7

2.2 Linear Regression with Additive Constant for Each Season

In this test, the same seven linear regression models are used except now a constant is added to each model to account for the difference in season. Figure 4 shows the performance of each model. The added constant does not improve the performance of the high performing models M1, M2 and M3 but it improves the performance of the rest of the models – M4 – M7 by around 2%. This model also does not perform better than the existing ISO NE Model.

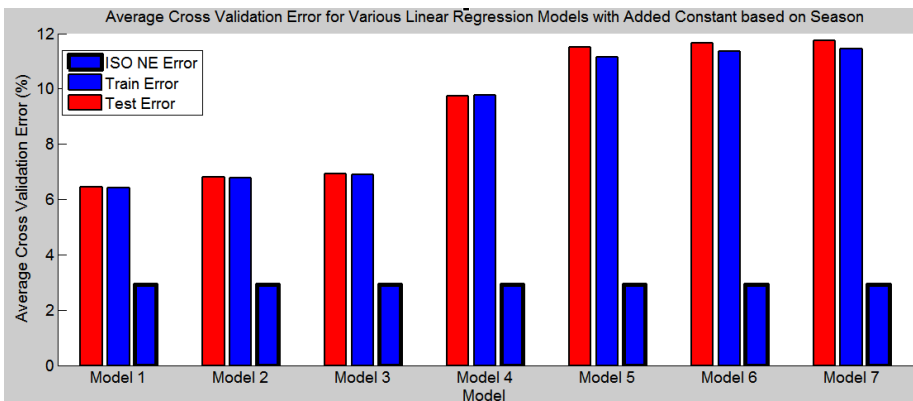


Figure 4. Average cross validation error for M1-M7 with added constant based on season

2.3 Weighted Linear Regression

Next, locally weighted linear regression (LWR) is applied to the 2014 dataset. The equation used is listed in (1).

$$\sum_i w^{(i)} (y^{(i)} - \theta^T x^{(i)})^2 \quad (1)$$

where

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right) \quad \text{and } \tau = .1, .3, .8, 2, 5, 10, 12$$

Figure 5 shows the performance of LWR for the different values of the bandwidth parameter. The best performing model is $\tau = .8$

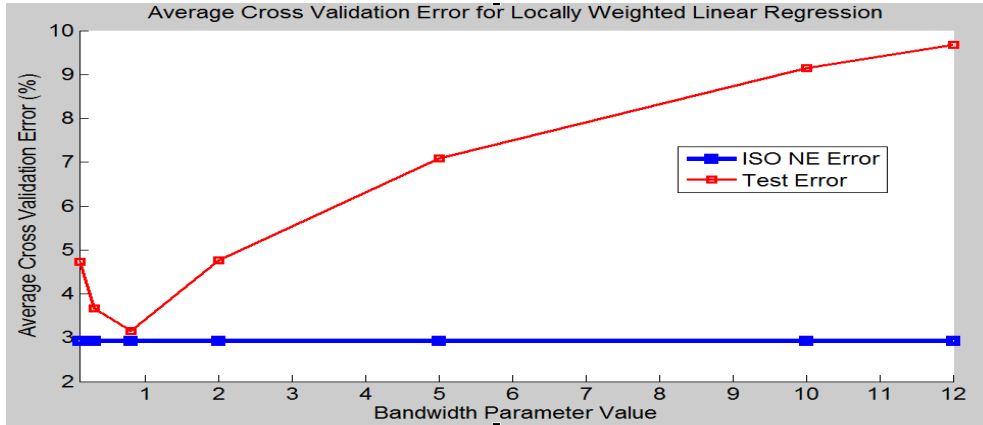


Figure 5. Average cross validation error for LWR with different values for the bandwidth parameter

The performance of the LWR model comes close to the performance of the ISO NE model but it still does not surpass it.

2.4 Feedforward Neural Network

In the final test, a feedforward neural network (FNN) is applied to the data. Different numbers of hidden layers are used to select the best performing model. As the number of hidden layers increases, the accuracy of the model improves. Also as the number of hidden layers increases, the training takes longer and the risk of overfitting increases. A FNN with 100 hidden layers is chosen as the best performing model in this group. The FNN models also has an error rate close to the error rate of the ISO NE model but it does not outperform it.

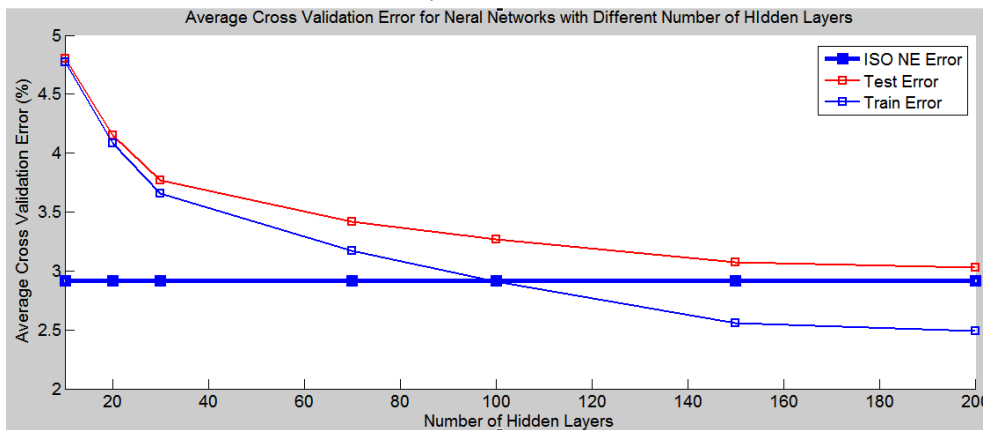


Figure 6. Average cross validation error for FNN with different number of hidden layers

3. Results and Discussion

Table 2 shows a comparison between the cross validation performance of the selected models from each group and Table 3 shows the performance of the same models against the 2013 dataset.

Model	Test Error (%)
Linear Regression (M1)	6.74
Linear Regression with an Additive Constant based on Season (M 1)	6.45
Weighted Linear Regression ($\tau = 0.8$)	3.15
Neural Network (hidden layer = 100)	3.27
ISO New England	2.92

Table 2. Comparison between the models that performed the best in each group

Based on the evaluation method used in this project, the best performing model across each group is Locally Weighted Linear Regression with bandwidth parameter = 0.8. The algorithm has the best accuracy performance but since it is a non-parametric learning algorithm, it runs very slow on large datasets.

Model	Test Error (%)
Linear Regression (M1)	6.86
Linear Regression with an Additive Constant based on Season (M 1)	6.55
Locally Weighted Linear Regression ($\tau = 0.8$)	5.77
Neural Network (hidden layer = 100)	5.99
ISO NEW England	3.93

Table 3. Comparison between the models that performed the best when tested on new data

When the models were run on the new and 'unseen' 2013 dataset, LWR again had the best performance. It didn't outperform the original ISO NE model but its accuracy is only 1.87% below the ISO NE Model.

III. Conclusion and Future Work

In this project, the performance of four groups of models is evaluated against the current model ISO NE uses to predict the energy demand for Boston Area. None of the algorithms explored in this project outperformed the existing ISO NE model but Locally Weighted Linear Regression comes close. Forwardfeed Neural Network model with 100 hidden layers has the second best performance.

In the future, more features should be added to try and improve the performance of the models. Features that should be looked into are daily temperature averages. Also non-linear regression models should be explored.

References:

- [1] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid—the new and improved power grid: A survey," IEEE Commun. Surveys Tuts., vol. 14, no. 4, pp. 944–980, Dec. 2011.
- [2] H. K. Alfares and M. Nazeeruddin, "Electric load forecasting: Literature survey and classification of methods," International Journal of Systems Science, vol. 33, no. 1, 2002.
- [3] F. Martinez-Alvarez, A. Troncoso, J. Riquelme, and J. A. Ruiz, "Energy time series forecasting based on pattern sequence similarity," IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 8, 2011. - Summary of models
- [4] <http://www.iso-ne.com/>