Electric Power Load Forecasting using Machine Learning Techniques

Vivek Vijaykumar Bajpai, Pooja Bhagwan Mankani
Illinois Institute of Technology

Abstract – The objective of this paper is to compare various machine learning techniques for the prediction of the electric power load in large buildings and residential apartments. Electric power demand prediction is a necessary process for efficient resource management in a smart grid. If performed accurately it can save millions of dollars for power companies. This paper helps to predict the electric power load using various regression techniques like Locally Weighted Regression (LWR), Multilayer Perceptron (MLP), Extreme Learning Machine (ELM) and Least Squared – Support Vector Machine (LS-SVM). All methods have been applied to the Comed Electricity historical data for a period of 90 days. The hour wise prediction is considered in each of the techniques. Moreover, this paper also discusses the interactive system model created which helps the user to view the prediction of power load consumption based on the inputs provided and the day for which the prediction needs to be made.

I. INTRODUCTION

Power Load prediction proves to be an economic way to manage resources required in power systems. With an accurate prediction it become very easy to know the electric power load required in a particular area at a particular time. Moreover, based on the climatic conditions and the nature of day (weekday / weekend) an accurate prediction of the power would retrieve correct results and also help to save millions of dollars by managing the resources efficiently based on the prediction made.

The paper discusses various regression techniques that are used on this problem. It compares the performance of each of the regression technique. Moreover, the paper also discusses the results obtained from the fresh testing data provided by the user. It discusses the interactive model created which helps the user to input the necessary data required to predict the power consumption. Based on the inputs provided by the user regression techniques are applied on the new data and results are predicted for the user.

II. DATA DESCRIPTION

The power load consumption majorly depends upon the climatic conditions of the local area, the day of the month (weekend / weekday) and the hour when the prediction needs to be made. The parameters that constitute the climatic conditions are temperature, wind chill, humidity, precipitation and wind speed.

Thus below are the parameters that are considered in this paper by each of the regression models to make the corresponding prediction:

Parameter	**Description**
temp	Current Temperature of the hour
wind_chill	Real feel temperature equivalent
hum	Humidity %
prep	Precipation of the hour
wind_speed	Corresponding wind speed during the hour
type_of_day	Day type either it could be weekday or weekend

Table 1: Input Parameters to create the feature matrix

The climatic data was extracted weather underground. A python script was created to call the weather API and the corresponding interval data was extracted. The hour wise climatic data was extracted for a period of 90 days from 1st Dec 2015 to 29th Feb 2016. The data was available over the internet in the below format:

https://www.wunderground.com/history/airport/KORD/2014/1/4/DailyHistory.html?req_city=&req_state=&req_statename=&reqdb.zip=&reqdb.magic=&reqdb.wmo=

However, bulk data for multiple days were required. Hence a script was created that called the API to extract the data as required. The data from the API was returned in JSON format which is was parsed and stored as required in a .csv file. A sample API is as follows:

http://api.wunderground.com/api/DEVELOPER_API_KEY/history_201501201/q/IL/Chicago.json

The climatic data along with the type_of_day data is the feature matrix along which the prediction will be made. The output data which is the power consumption for a period is extracted from the historical comed data. The data was extracted from the below link:

https://www.comed.com/customer-service/rates-pricing/retail-electricity-metering/Pages/historical-load-data.aspx

The following data patterns were seen between the climatic and power consumption data for a specific hour of the day:

Temperature v/s Power consumption: The power consumption is linearly dependent on the temperature for a particular hour. As the temperature increases the power consumption reduces since the electrical equipment used would be lesser which increase in temperature. Hence temperature is one of the dominant features in our prediction. This is relevant from the below graph.

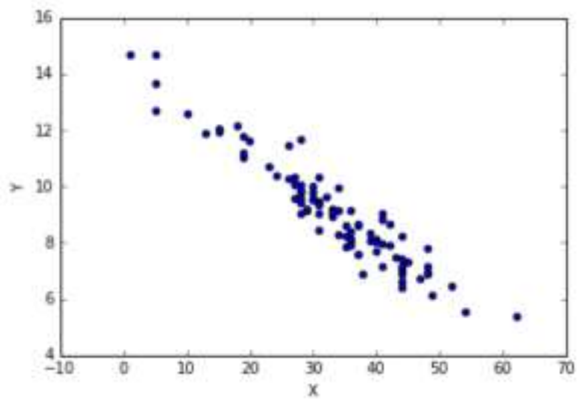

Fig 1. Temperature v/s Power Consumption

Wind Chill v/s Power consumption: Like temperature wind chill also follows the same trend wherein as the wind chill increases the power consumption reduces. Hence wind chill also is one of the dominant features in our prediction.

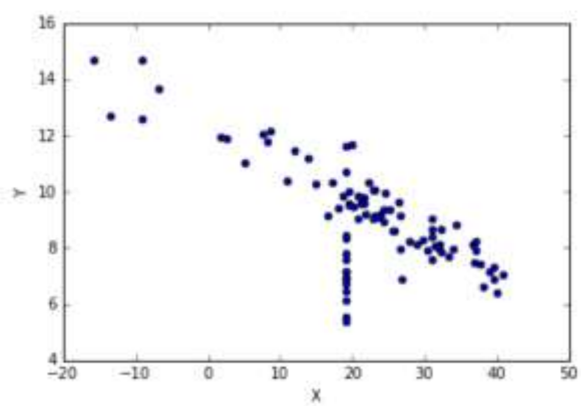

Fig 2. Wind Chill v/s Power Consumption

Humidity v/s Power consumption: There is no relation between the power consumption and humidity. This is because as we can see in the graph the power consumption is scattered and does not follow any trend with respect to humidity. Hence this is not a dominant feature for our prediction but one of the relevant features since such neutral features should also be taken into consideration while predicting real data.

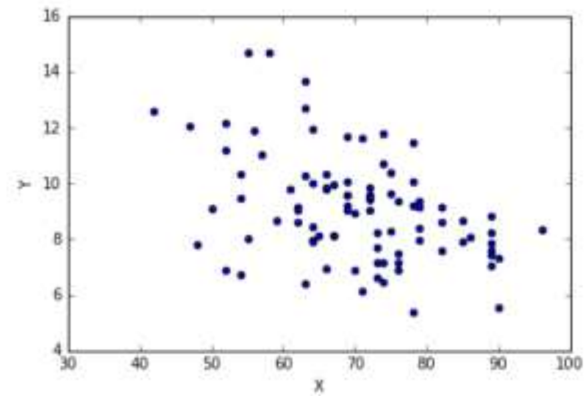

Fig 3. Humidity v/s Power Consumption

Like humidity a similar trend is observed for precipitation and wind speed.

Type of day v/s Power consumption: From the data it is observed that the average of power consumption is high on weekdays as compared to that on weekends.

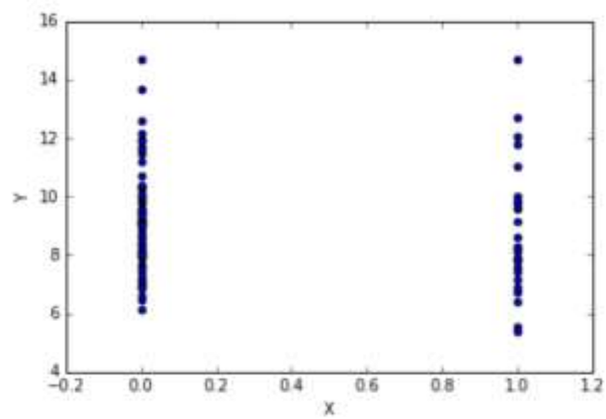

Fig 4. Type of Day v/s Power Consumption

The data patterns were also observed for the complete data set used. These are mentioned as follows with respect to the entire day:

Hour v/s Power consumption: It is noticed from the below graph that the power consumption on peak hours is the highest and lowest on non-peak hours.

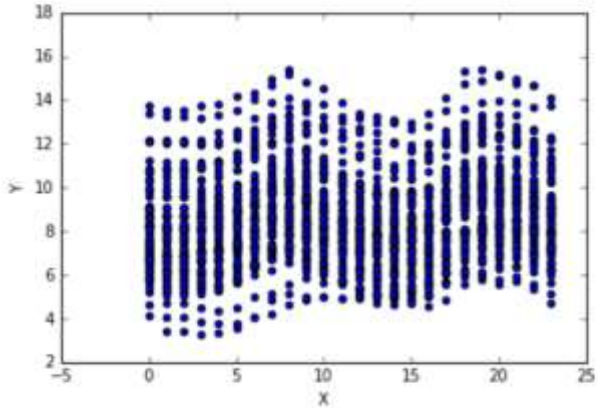

Fig 5. Hour v/s Power Consumption

Temperature v/s Power consumption for the entire dataset: It is seen that the power consumption is linearly dependent on temperature

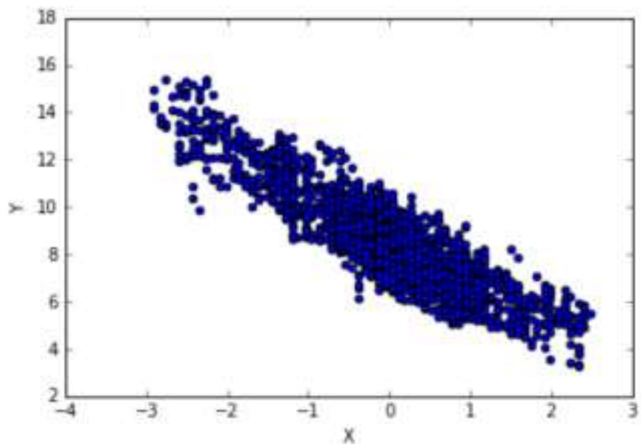

Fig 5. Temperature v/s Power Consumption for entire dataset

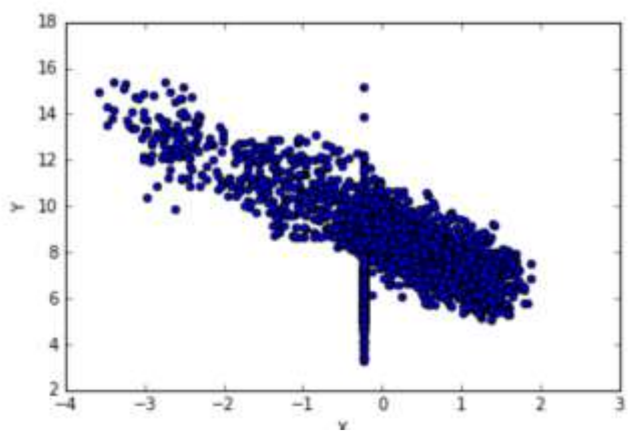

Fig 6. Windchill v/s Power Consumption for entire dataset

Data Normalization:

The data was normalized with respect to each columns standard deviation and average value. This is because the data used had different range of values. In case it would not be normalized the prediction would not be done accurately.

Missing Values:

In case the values of the any of the feature elements were missing, the average over that column is calculated and added to the missing feature. Thus this is handled in the implementation and the missing values would not affect the actual evaluation.

III. PROPOSED SOLUTION

Multiple regression techniques are available in order to make a prediction that is continuous. In the solution to predict power consumption using climatic data along with type of day in question we use simple regression techniques like Locally Weighted Regression (LWR) and multiple complex techniques like Multilayer Perceptron (MLP), Extreme Learning Machine (ELM) and Least Squared – Support Vector Machine (LS – SVM)

A. Locally Weighted Regression

In this regression technique, the objective function is defined similar to that of Multi Linear Regression, however, a window function "Wx" is added to the equation. This window helps to perform training only on the data that are local to that point where the prediction needs to be performed. LWR is most suited for real time control, however, it requires that the training be performed every time on the locally available data and this model is then applied to the corresponding testing data.

This regression technique was chosen since we had hour wise data. While applying a regression model on this kind of data it is optimal to use LWR since we have the local data available for each hour. Hence the training becomes faster and the results are available based on the local data which gives better performance.

The objective function of LWR along with the window function is derived as:

$$J(\theta x) = \sum_{i}^{m} Wx^{i}(\theta x.TX^{i} - y^{i})$$

The above equation can also be written as:

$$J(\theta x) = ((Z\theta - Y).T).Wx.(Z\theta - Y)$$

The theta is then calculated as:

$$\theta x = (Z.T.Wx.Z)^{-1} Z.TWxY$$

The Wx matrix is evaluated in the below form:

$$Wx = \begin{bmatrix} Wx^{(i)} & 0 & 0 & \dots. & 0 \\ \dots\dots. & & & & \\ \dots\dots.. & & & & \\ 0 & 0 & 0 & \dots. & Wx^{m} \end{bmatrix}_{m*m}$$

B. Multilayer Perceptron

This technique is a feedforward neural network model that helps to maps the input set into an output set using the hidden layer that is defined between the two. In our approach we have considered a single hidden layer fully connected to the input and output layer thus predicting the output as required.

The output can be predicted as:

$$\hat{y} = (V.T).Z = V_0 + V_1 Z_1 + V_n Z_n$$

where "Zj" is computed as:

$$Z_j = sigmoid((W_j.T).X)$$

The error function for the regression technique is calculated as:

$$E(V, W_j) = 1/2 \sum_{i=1}^{k} (\hat{y}_i - y_i)^2$$

Then gradient descent is applied to get the best value of V as:

$$V \leftarrow V - \eta \sum_{i=1}^{m} (\hat{y}_i - y_i) Z_i$$

The weights of the hidden layer are also calculated using the gradient descent as:

$$w_j \leftarrow w_j - \eta \sum_{i=1}^{m} (\hat{y}_i - y_i) V_j . Z_j^{(i)} (1 - Z_j^{(i)}) X^{(}$$

C. Extreme Learning Machine

The extreme learning machine is a single layer feedforward system using the neural networks similar to that of the Multilayer Perceptron. However, it was observed that multilayer perceptron takes large amount of time for execution with respect to training the hidden neurons. However, in extreme learning machine though the performance is not improved with respect to multilayer perceptron the amount of time taken to train and predict the values is less in extreme learning machine. This was evidently noticed in our program since compared to Multilayer Perceptron, Extreme learning machine took $1/10^{th}$ of the time for execution.

The coefficients of the hidden layer are randomly initialized. The optimal weights are then calculated using the pseudo-inverse of the hidden layer output matrix. This approach can be calculated using any of the incremental approaches like Gaussian, Iterative approach etc. Thus the ELM provided a good generalization performance at a very high learning rate.

The below equations were used for evaluating the performance of the model

$$min_{\beta \in R^{L*M}} \parallel H\beta - T \parallel^2$$

Where H is the hidden layer randomized matrix as:

$$H = \begin{bmatrix} h_1(x_1) & \dots\dots. & h_L(x_1) \\ \dots\dots. & & \\ \dots\dots.. & & \\ h_1(x_N) & \dots\dots.. & h_L(x_N) \end{bmatrix}$$

and T is the feature matrix as:

$$T = \begin{bmatrix} t_{11} & \dots\dots. & t_{1m} \\ \dots\dots. & & \\ \dots\dots.. & & \\ t_{N1} & \dots\dots. & t_{Nm} \end{bmatrix}$$

D. Least Squares – Support Vector Machine

In this regression technique the solution is solved using linear equations rather than quadratic programming. A kernel based method was applied for this approach to generate the results as required.

The optimization problem using the Lagrangian multipliers is defined as:

$$\mathcal{L}(w, b, e; \alpha) = \mathcal{J}(w, e) - \sum_{i=1}^{N} \alpha_i \{ y_i [w^T \varphi(x_i) + b] - 1 + e_i \}$$

Thus using the above optimization problem the dual problem can be resolved as:

$$\begin{bmatrix} 0 & | & y.t \\ y & | & \Omega + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_v \end{bmatrix}$$

where y is defined as:

$$y = [y1; \;\; .... \;\; ;yn]$$

and 1v is defined as:

$$1_v = [1; \;\; .... \;\; ;1]$$

IV. RESULTS AND EVALUATION

For each of the regression techniques the training root mean square error is evaluated and the actual and predicted data for each of the technique is plotted. The results are as follows:

A. *Locally Weighted Regression*

In LWR technique the model performs well for training data. The actual and the predicted values follow the trend and give the expected results. This is because for each of the training set the model is recalculated always with respect to the local data and the parameter calculation is done every time.

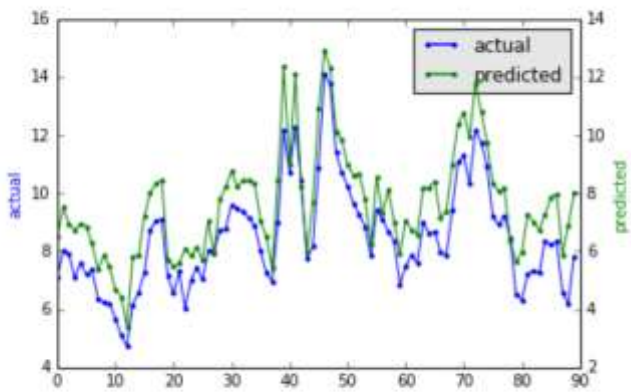

Fig 7. Actual v/s Predicted LWR

B. *Multilayer Perceptron*

This technique yields a better result compared to LWR. This is because the hidden layers are trained correctly and they are used to predict the output as required. The output layer is evaluated based on the tuned parameters which thus give better training results when compared to LWR.

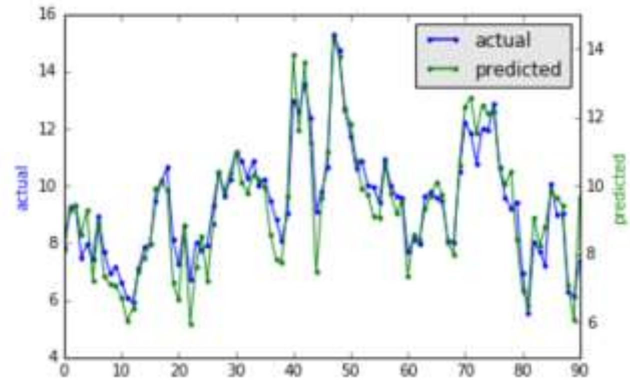

Fig 8. Actual v/s Predicted MLP

C. *Extreme Learning Machine*

The extreme learning machine performs similar to that of the Multiplayer Perceptron (MLP) since both are single layer neural network feedforward systems. However, in case of ELM it takes lesser time to execute and generate slightly better results. This is because initially random neurons are considered and the model learns at a very high learning rate.

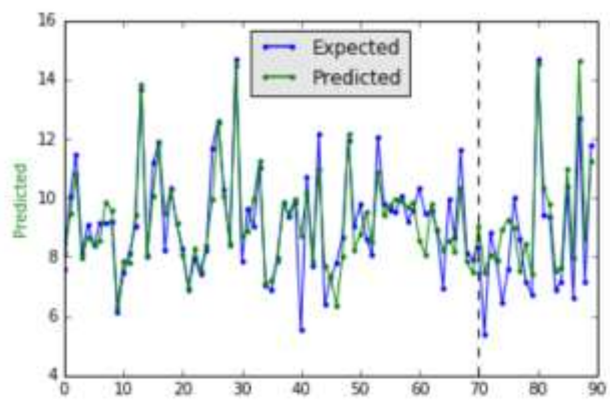

Fig 9. Actual v/s Predicted ELM

D. *Least Squares – Support Vector Machine*

The least squares support vector machine does not perform very well since this is a regression technique that is performed. The SVM performance is good during classification problems but does not work very well for regression.

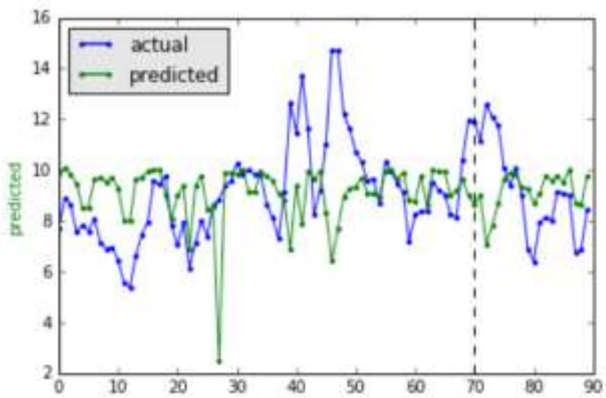

Fig 10. Actual v/s Predicted LS-SVM

Comparison of error (root mean square error) between models:

Model	RMSE
Locally Weighted Regression	0.975068906
Multilayer Perceptron	0.649443007
Extreme Learning Machine	0.550479168
Least Squares - Support Vector Machine	2.313839437

Table 2: Comparison of error between models

The performance of ELM is the best when compared to other models. This is because the ELM tries to improve the MLP model and since MLP works better than LWR the ELM works better than all of them.

Comparison of relative mean square error using 10 fold cross validation:

Model	Average Relative Mean Square error with Cross Validation
Locally Weighted Regression	1.111874532
Multilayer Perceptron	1.565121223
Extreme Learning Machine	0.285436659
Least Squares - Support Vector Machine	1.55329854

Table 3: Comparison of relative square error

Similarly ELM model works best over cross validation as well.

V. APPLICATION USAGE

This implementation can be used to predict the power consumption based on the climatic conditions and the day provided. The user can input the test data in a file called "test_input.txt" in the below format:

Parameter	Description
hour_of_the_day	Hour to be predicted
temp	Current Temperature of the hour
wind_chill	Real feel temperature equivalent
hum	Humidity %
prep	Precipation of the hour
wind_speed	Corresponding wind speed during the hour
weekend/weekday	Day type either it could be weekday or weekend

Table 4: Format for test data

The test was performed for one of the test input data on all the models and the following are mentioned below:

The prediction shows that we notice that the results are predicted well. The power consumption is least at around 3 to 4 AM and highest at 10 PM. This is relevant to our daily lives wherein the electricity usage is less at the non-peak hours and high and peak hours.

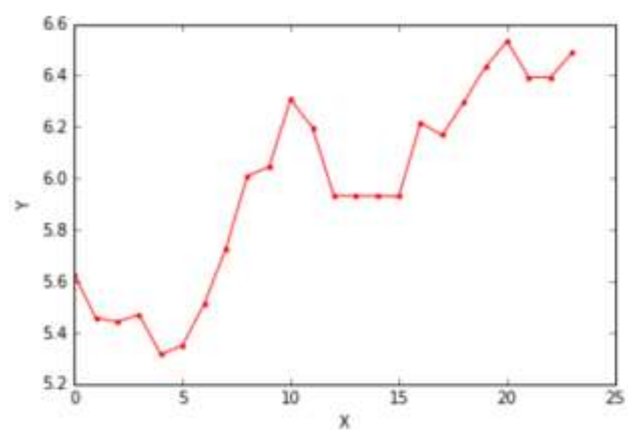

Fig 11. Prediction of test data

VI. CONCLUSION

Power consumption prediction is important and if used efficiently can save huge amount of money in terms of resource management required for the electricity supply.

Various regression techniques can be applied to predict the power consumption. This will help various energy power plants to generate and supply optimal energy. The various regression models work as required to predict the data with each of them using the best fit parameters to predict the power consumption.

VII. FUTURE WORK

These models can be applied to the synchrophasor data that is generated in IIT buildings. This will help to predict the power consumption of the data generated from the data grid. Also this will be very useful to the electrical department at IIT to manage resources required for the energy generation and supply.

Different kernel methods can be applied to Least Square Support Vector Machine in order to improve the performance of the model. Also the best fit parameters must be obtained using various hyperparameterization and the must be applied to the model.

The time taken by the Multilayer Perceptron needs to be improved by using additional kernel tricks to calculate the weights of the hidden layer. This can also be done using cython to execute the various for loops required for the program

VIII. REFERNCES

[1] Machine Learning Techniques for Short-Term Electric Power Demand Prediction- [Fernando Mateo, Juan J. Carrasc, M´onica Mill´an-Giraldo, Abderrahim Sellami, Pablo Escandell-Montero, Jos´e M. Mart´ınez-Mart´ınez and Emilio Soria-Olivas]

[2] Machine learning based switching model for electricity load forecasting – [Shu Fan, Luonan Chen, Wei-Jen]

[3] An Event-Oriented Method for Online Load Modeling Based on Synchrophasor Data- [Yinyin Ge, Student Member, IEEE, Alexander J. Flueck, Senior Member, IEEE, Dae-Kyeong Kim, Jong-Bo Ahn, Jae-Duck Lee, and Dae-Yun Kwon]

[4] Ge, Y., Flueck, A.J., Kim, D.-K., Ahn, J.-B., Lee, J.-D, and Kwon, D.-Y, " An Event-Oriented Method for Online Load Modeling Based on Synchrophasor Data," in Smart Grid, IEEE transactions, 2015, volume: PP, Issue: 99.

[5] Ge, Y., Flueck, A.J., Kim, D.-K., Ahn, J.-B., Lee, J.-D, and Kwon, D.-Y, "Power System Real-time Event Detection and Associated Data Archival Reduction Based on Synchrophasor," in Smart Grid, IEEE transactions, 2015, volume: PP, Issue: 99.

[5] Navarro, I.R., "Dynamic Power System Load: Estimation of Parameters from Operational Data", Medya-Tryck, Sweden, 2005.

[6] http://elm.readthedocs.org/en/latest/

[7] https://en.wikipedia.org/wiki/Least_squares_support_vector_machine

[8] ftp://ftp.esat.kuleuven.be/sista/kdebaban/11-82.pdf