# Short Term Load Forecasting based on Internet of Things and Machine Learning Algorithms

Mahdi Uz Zaman, Anisul Islam Fahad, Nahid Sultana, Amitabha Chakrabarty

Department of Computer Science and Engineering, BRAC University,
66 Mohakhali, Dhaka 1212, Bangladesh

{emahdi.zaman, kingfahad0802, nahid.sultana280}@gmail.com, amitabha@bracu.ac.bd

**Abstract - In the era of Internet every device is getting connected to the Internet. Devices that are connected to the Internet are called Internet of Things (IoT)** [1]**. In this paper we have assumed that IoT devices can share their power consumption history. Based on data points collected from real world environment we have conducted experiments to show that IoT can be used as a reliable backbone of short term load forecasting system. In the experiment four machine learning algorithms Long Short-term Memory (LSTM), Support Vector Machines Regression, Decision Forest Regression with AdaBoost and Nearest Neighbors Regression were used to analyze the performance of the load forecasting system. Result of the experiment have shown that short term load forecasting system based on LSTM will give comparatively better result with root mean square error of 1.82.**

**Keywords - IoT, Forecasting, Save electricity, LSTM, Machine Learning, SVM, kNN, Random Forest.**

## I. INTRODUCTION

Electrical energy generation and distribution is a complex and costly process. Efficient grid management plays a big role to reduce the cost of energy production. Grid management comprises of planning for load demand, maintenance of generation units, supply lines and efficient load distribution across the supply line. Therefore an accurate load forecast will increase the efficiency of planning process of a power generation company. Power generation companies do their plan based on data collected manually. Therefore real time prediction is not possible. If data can be collected in real time, forecasting in real time will be possible. Strong and reliable Internet infrastructure are already present. Every device we use in our daily life are gradually getting connected to the Internet to facilitate smart home technologies like Google Home [2], Amazon Alexa etc. A device connected to Internet usually treated as IoT. In general a device with sensors, microprocessor or microcontroller which can connect to the Internet, send and receive information through Internet is called Internet of Thing (IoT) [1]. If the devices are configured to send energy uses data to the Internet, these data can be used to give real time forecasting. In this paper we have shown that real time load forecasting is possible with the help of IoT and state of the art machine learning algorithms LSTM Network, Nearest Neighbors Regression, Support Vector Regression and Decision tree Regression with AdaBoost. First of all, increased demand of electricity is creating pressure on production companies as well as natural resources. We know natural resources which are used to produce to create electricity are limited in nature. Secondly, the byproduct of electricity generation is pollution. Again, cost for producers and consumers are increasing day by day. Therefore to ensure sustainable development research communities have shown great interest on how to reduce electricity demand by efficient use of electricity. One of the important of methods that is used to facilitate efficient use of electricity is load forecasting [3]. With the help of load forecasting producers can tune their production plan and consumer can optimized their electricity consumption. Existing forecasting system relies on data collected from production and distribution unit. We have shown that with the help of IoT load forecasting can be done in more easy, convenient and reliable way.

IoT can be used as a reliable backbone of a load forecasting system. To support our claim we have tested our system with real world datasets. Based on this dataset we have done empirical comparison and performance evaluation of four machine learning algorithm. This system will help home user to reduce their power consumption by early warning of future power use. This will also help the power generation company to meet their demand efficiently by planning ahead of time.

In this paper, section - 2 provides the literature review in brief description including and techniques used in the system. Proposed model including the algorithms and discussed in section - 3. Results and analysis are presented in section - 4. Lastly section - 5 gives the conclusion and future work.

## II. RELATED WORK

Load forecasting prediction was done by many others using different algorithms. Kong, W. worked on deep learning based method with appliance behavior learning using LSTM and was compared with the Feed Forward Neural Network (FFNN) and K-nearest neighbors (KNN) [4]. Ghulam and Angelos compared the performance of feed forward deep neural network (FF-DNN) and recurrent neural network (R-DNN) on the basis of accuracy and computational performance in the context of time wise short term forecast of electricity [5]. Papia Ray, Santanu Sen and A.K. Barisal granted two hybrid methodologies based on DWT in combination with ANN or SVR for short term load forecasting taking into consideration temperature, humidity dew point and load consumed a particular day at particular hour [6]. Taking in consideration of the above mentioned work we have implemented an IoT based load forecasting system. The core algorithm of the forecasting system is a machine learning algorithm. To select best performing algorithm we have tested performance of several machine learning algorithm with a new dataset called “The UK-DALE dataset” [7].

## III. PROPOSED MODEL

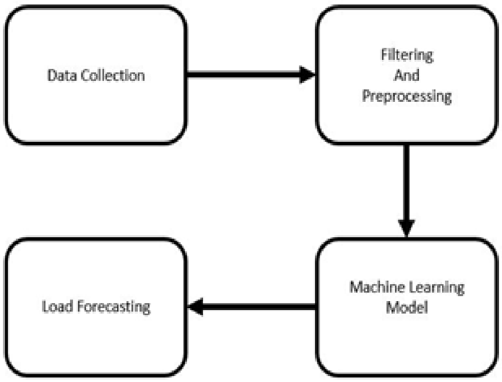


Figure 1. Proposed Model

The IoT based Short Term Load forecasting system is consist of IoT devices and a central processing unit. IoT devices are configured to upload power usage data to the server. The central processing unit is configured to do calculation based on a selected machine learning algorithm. Based on the calculation and learning process the unit will give prediction. In Figure 1 the workflow is shown. Main idea of this paper is to forecast total power consumption based on the data collected from the IoT devices. Due to limitations of time and resources we could not configure devices to upload power uses data to the internet. To determine which prediction algorithm works best for our system we have implemented four machine learning algorithms and compared their performance.

In particular, the following machine learning algorithms for forecasting were used:

• **Nearest Neighbor Regression:** KNN regression is used to calculate average of targeted value of k nearest neighbors. Using inverse distance weighted average of the k nearest neighbors can also be calculated. KNN classification and KNN regression uses the same distance functions. KNN regression is used to calculate average of targeted value of k nearest neighbors. Using inverse distance weighted average of the k nearest neighbors can also be calculated. KNN classification and KNN regression uses the same distance functions. With the help of the following functions distance between neighbors are measured [8]:

Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_{i-}y_i)^2} \tag{1}$$

Manhattan

$$\sum_{i=1}^{k}|x_i - y_i| \tag{2}$$

Minkowski

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q} \tag{3}$$

These equations can only be used for continuous variables. Inspecting the data the ideal value for K is chosen. With a large K value the noise is reduced but it becomes harder to detect the distinct features. To determine good K value using independent data set to validate K values, cross- validation is an ideal way. The ideal K for most datasets is 10 or more which produces better results than 1-NN

• **Support Vector Regression:** Support Vector Machine can also be used as a method of regression, keeping all the fundamental elements intact that designate the algorithm (maximal margin). With scarcely trivial distinction, the Support Vector Regression utilizes the same postulates as the SVM for categorization. Assuming a set of training points, $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $x_i$ is a feature vector and $x_i \in R^n$ and $y_i \in R^1$ is the target output. If the given parameters $C > 0$ and $\varepsilon > 0$ then according to Vapnik [9] Support Vector regression is

$$\sum_{i=0}^{n}(-\alpha_i + \alpha_i^*)K(x_i, x) + b \tag{4}$$

• **Decision Tree Regression with AdaBoost:** Decision tree establishes regression models in the form of a tree structure. It simultaneously jots down dataset onto smaller and smaller subsets and aligned decision is incrementally established. The end result is a tree with decision nodes and leaf nodes. A numerical target is illustrated by leaf node. The highest decision node in a tree which becomes equivalent to the leading predictor is called root node. Both categorical and numerical data can be conducted by decision trees [10]. To increase the accuracy of the model we boosted the model using AdaBoost. Assuming training vectors $x_i \in R^l, i = 1, \ldots, n$ and targeted values containing vector $y \in R^n$, partition of the space is made recursively such that the samples with the same labels are in a group. Suppose the data at node $k$ be represented by $P$. Splitting is done using $\theta = (j, t_k)$ having of an attribute $j$ and threshold $t_m$. After partitioning the data is kept into $P_{left}(\theta)$ and $P_{right}(\theta)$ subsets $P_{left}(\theta) = (x, y)|x_j \le t_k$ and $P_{right}(\theta) = P \setminus P_{left}(\theta)$. Noise at $k$ is calculated using a function $H(\,)$ which calculates impurity. The choice of function depends on the method of solving (classification or regression) [19]:

$$G(P, \theta) = \frac{n_{left}}{N_k} H\left(P_{left}(\theta)\right) + \frac{n_{right}}{N_k} H\left(P_{right}(\theta)\right) \tag{5}$$

To minimize impurity parameters are selected using

$$\theta^* = argmin_\theta G(P, \theta) \quad (6)$$

It is continued for subsets $P_{left}(\theta^*)$ and $P_{right}(\theta^*)$ until the maximum depth is reached where $N_k < min_{samples}$ or $N_k = 1$.

• **Long Short Term Memory:** LSTM networks are renowned for their ability to remember pattern and sequence [11]. Human behavior tends to be repetitive. From this intuition we used LSTM Network to learn the behavior pattern of power usages. LSTM network can give prediction based on the calculation of cell state, forget gate Equation 7, input gate Equation 8 and output gate Equation 11 [12].

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (7)$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (8)$$

$$\check{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (9)$$

$$C_t = i_t * \check{C}_t \quad (10)$$

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t * \tanh(C_t) \quad (12)$$

$W_f$: Weight for forget gate layer

$b_f$: Bias for forget gate layer

$i_t$: Output of input gate layer

$W_i$: Weight for input gate layer

$b_i$: Bias for input gate layer

$\check{C}_t$: Candidate value

$W_c$: Weight of tanh layer

$b_c$: Bias of tanh layer

$W_o$: Weight of output gate layer

$b_o$: Bias of output gate layer

## IV. Experiments and Results

### A. Experimental Protocol

In this paper we have used "The UK-DALE dataset" created by Jack Kelly & William Knottenbelt [7]. This datasets contains appliance level disaggregated power consumption record as well as aggregated whole house power consumption record. In Figure 2 whole house power consumption of last one year is given. We assumed that all the appliances in this datasets are IoT devices which records the power consumption of an appliances and uploads in the database. For training and testing we have used data of house_1 because it contains maximum number of appliances. Also they have given more emphasis on recording house_1 data. The main dataset contains five folder. Each folder correspond to each house. Under each house numbers of CSV files according to number of devices are given. Each CSV file contains records of power consumption with time. In each CSV file time is give in format of UNIX time epoch. Interval data recording is six seconds. Noise and misleading data in any dataset are bad for any model to train on. A misleading dataset will eventually produce a hypothesis which will not do well in unseen data. Therefore noise cancellation has done with great care. The dataset contains UML configuration file for every houses. The file has details description of meter devices and appliances. Each appliance has upper bound and lower bound of power consumption. Any power consumption beyond that limit is considered as noise or bad reading. In Table I upper and lower bound of few devices recorded in house_1 is given.

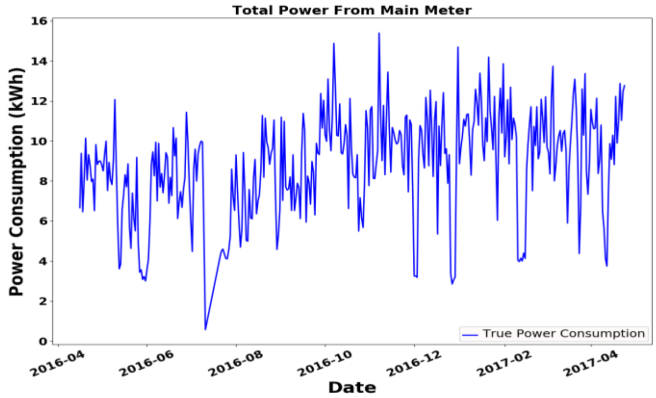


Figure 2. Total Power Consumption of last 1 year in KWh

.

TABLE I. List of few devices

| Channel ID | Name | Min Power (Watt) | Max Power (Watt) | Type |
|---|---|---|---|---|
| 2 | boiler | 70 | 4000 | Apparent |
| 3 | solar_thermal _pump | 43 | 4000 | Apparent |
| 4 | laptop | 70 | 4000 | Active |

The most frequently used measures of the differences between predicted values by an estimator and the actual values is the Root Mean Square Error (RMSE). The RMSE of predicted data $\hat{y}_i$ for survey of $i$ , for variables $y_i$ is calculated for $n$ numbers of cases using the following formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

The parameters for building the predictive models of the algorithms are:

• Nearest Neighbors Regression: $n\_neighbors$=15.

• Support Vector Regression: Penalty parameter C=1.0, epsilon=0.2 (specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value).

• Decision Tree Regression with AdaBoost: max_depth=16, n_estimators=300.

• LSTM: the network contains on LSTM cell. Input of every time step contains a sequence of 7 samples.

### B. Results

As our goal is to predict the power consumption of the house the next day depending on the power usage of the appliances of the present day. Model is evaluated on test set. The error result is obtained from test set. For preprocessing we scaled the data using the library function scale(). We used Kfold splits for splitting the dataset as it's a time series dataset. Later we trained our model on train set and the error rate is acquired by evaluating the models against our test set. We used all the default settings of the library and few changes in parameters of few algorithms. We used an API out of 3 APIs of the library to measure our trained model's performance. The API scoring parameter contains model-evaluating tools using cross-validation depends on an internal scoring strategy. We used this API to find the MSE of our models then used square root to find out RMSE and we compared the algorithms based on the result of the API.

Nearest Neighbors has RMSE of 1.9331727. As we are to predict the total power consumption of a house of the next day depending on the usage of the today's power consumption of the appliances, for which we are getting this much higher RMSE value. In Figure 3 the comparison of the real value and the predicted value of the trained model Nearest Neighbors Regression is given where x-axis is the date and y-axis is power consumption of the house in KWh.

Using the kernel RBF performs better than other kernels in SVR. RBF kernel performs better in this context because of the data. The higher the degree, the performance of other kernels are worse than RBF. SVR using kernel polynomial and Gaussian Radial Basis function with degree of 3 has RMSE of 2.1229618 and 1.8341087 respectively. But kernel linear performs slightly lesser than RBF but better than the polynomial as it has a degree of 1. It has RMSE of 1.8474361. For having the lowest RMSE among the kernel function, we preferred the kernel function RBF for SVR. In Figure 4 the performance of Support Vector regression (kernel RBF) is shown by comparing the true value with predicted value where x-axis is date and y-axis is the power consumption of the appliances in KWh.

The predicted power consumption is the output of the model. The RMSE of Decision Tree Regression with AdaBoost algorithm based model is 1.9202281. As we restricted the depth to 16 and we selected all the features to train the model we are getting much high RMSE. The deeper the tree is the model tend to over fit. To avoid overfitting we specified the maximum depth of the tree. In Figure 5 Evaluation of the model Decision Forest Regression with AdaBoost is shown by comparing predicted and real value. In Figure 5 x-axis is the date and y-axis is power consumption of the house in KWh.

In experiment we used single cell LSTM network. We have tested with LSTM network with up to 3 LSTM cell stacked top of one another. Stacking more than one LSTM cell made computation heavier but did not give better result. In some cases it went bad. We have also experimented with length of look back. Here look back is how many samples is given as an input in each time step. We have tested variable length of look back. Most significant were 7 for 7 days, 15 for 15 days, 30 for 1 month. Length of look back (between 7 and 15), have given better result than longer look back like 30.

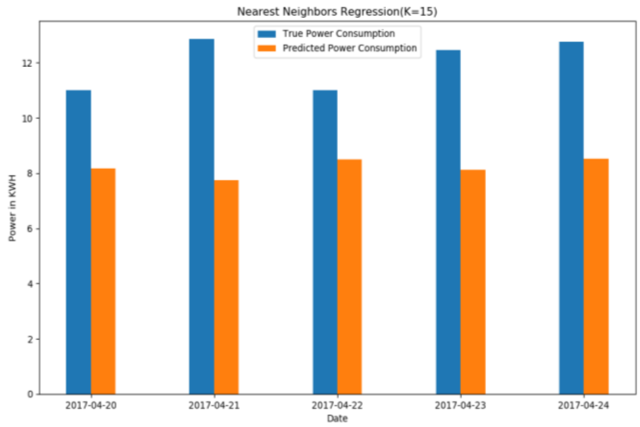


Figure 3. Nearest Neighbors Regression, empirical comparison

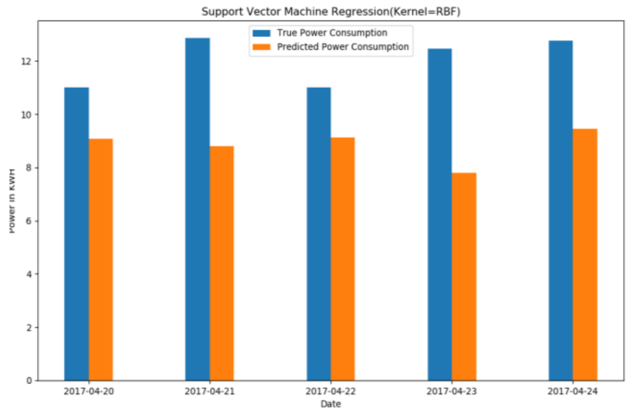


Figure 4. Support Vector Regression, empirical comparison

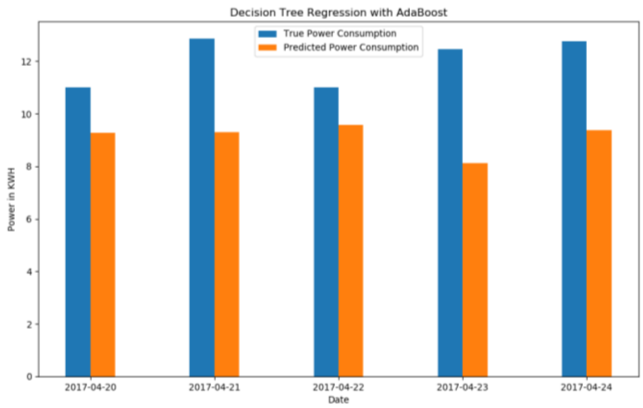


Figure 5. Decision Tree Regression with AdaBoost, empirical comparison

We have found best result in look back length 7. In Table - II we can see LSTM has given lowest RMSE score. Reason of the lowest score is ability of a LSTM to process sequence of samples rather than a single sample. In Figure 6 we can see that predicted data points by a LSTM almost catches the pattern of electricity usages. In Figure 7 point by point comparison of 10 test and predicted data point is given.

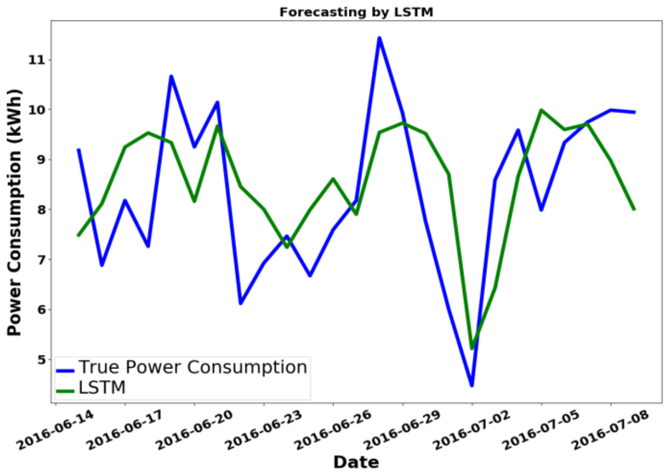


Figure 6. Load forecasting by LSTM

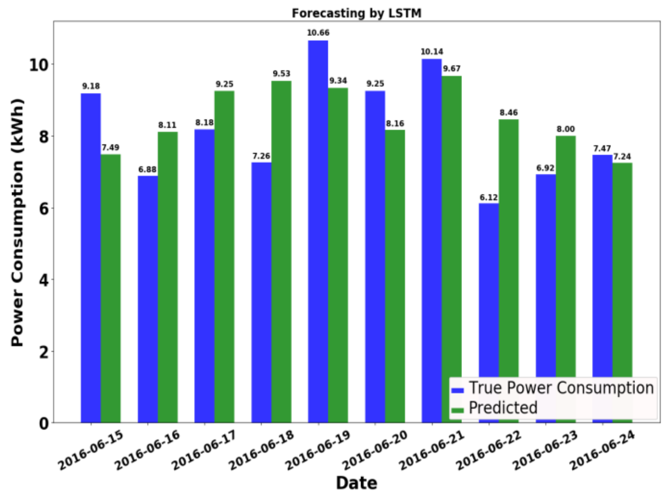


Figure 7. Load forecasting of LSTM, empirical comparison

TABLE II. RMSE OF ML MODELS

| Algorithms | Error |
|---|---|
| | *Root Mean Square Error* |
| Nearest Neighbors Regression | 1.93 |
| Support Vector Regression | 1.83 |
| Decision Tree Regression with AdaBoost | 1.86 |
| Long Short Term Memory | 1.82 |

In Figure 8 an overall summary of the outputs of four algorithm is given. Day by day our processors are getting stronger and less power hungry. Also in recent days processors are coming with dedicated core for neural net and artificial intelligence. As a result cost and computational power required for training a neural network will not be problem in the future. Neural networks like LSTM has the ability to adopt with a great variety of patterns and the ability of recognize those pattern. In our experiment LSTM has given better result with compared to Support Vector Machines Regression, Decision Forest Regression with AdaBoost and Nearest Neighbors Regression, Table-II.

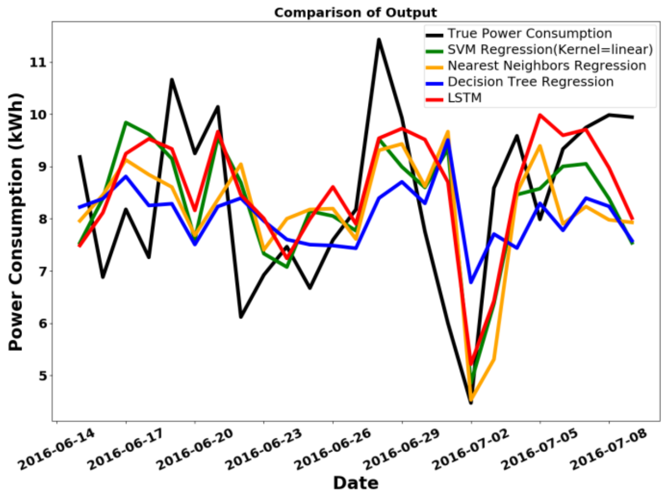


Figure 8. Comparison of four ML Algorithms based on their outputs on same test set.

## V. CONCLUSION

In this paper we have presented a system which can give prediction based on data collected from IoT devices. To prove the reliability of the system we have tested the system with real world data sets. We have conducted several experiments to evaluate the performance of four machine learning algorithms and concluded the experiment with a comparison of RMSE loss score. Long Short Term Memory network has given lowest RMSE in the experiment. The system described in this paper worked on data collected from IoT devices. This system can be implemented for home managements and grid managements. We performed analysis using power consumption data using data clustering based on the time (day). Models can be improved if the data clustering is based on the time interval of hours. This might reduce the error of the models. In future we are looking forward to compare models using different regression based ML algorithms. Privacy is a big concern here. In future we also want to work on the security side of this system. Predictions can be improved selection of features and changing the parameters.

## REFERENCES

[1] B. Dorsemaine, J.-P. Gaulier, J.-P. Wary, N. Kheir, and P. Urien, "Internet of Things: A Definition & Taxonomy," in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, 2015, pp. 72–77.

[2] P. Dempsey, "The Teardown: Google Home personal assistant," *Eng. Technol.*, vol. 12, no. 3, pp. 80–81, Apr. 2017.

[3] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?," *Energy Effic.*,

vol. 1, no. 1, pp. 79–104, Feb. 2008.

[4] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, “Short-Term Residential Load Forecasting Based on Resident Behaviour Learning,” *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1087–1088, Jan. 2018.

[5] G. M. U. Din and A. K. Marnerides, “Short term power load forecasting using Deep Neural Networks,” in *2017 International Conference on Computing, Networking and Communications (ICNC)*, 2017, pp. 594–598.

[6] P. Ray, S. Sen, and A. K. Barisal, “Hybrid methodology for short-term load forecasting,” in *2014 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, 2014, pp. 1–6.

[7] J. Kelly and W. Knottenbelt, “The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes,” *Sci. Data*, vol. 2, p. 150007, Mar. 2015.

[8] N. S. Altman, “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, Aug. 1992.

[9] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer New York, 2000.

[10] J. Elith, J. R. Leathwick, and T. Hastie, “A working guide to boosted regression trees,” *J. Anim. Ecol.*, vol. 77, no. 4, pp. 802–813, Jul. 2008.

[11] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[12] F. A. Gers, F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” *NEURAL Comput.*, vol. 12, pp. 2451--2471, 1999.