

Forecasting Electrical Energy Consumption using Artificial Neural Networks

Ayanangshu Das¹

¹Data Scientist

Cognizant Technologies Solutions United States

Anirban Sengupta²

²Asst. Professor

Sikkim Manipal Institute of Technology India

Abstract –Knowing how much the demand for energy would be for the coming months would be very useful for electricity producing companies. Electricity can't be stored and energy companies face a challenge where the demand is always more and the supply is less. Hence the companies look for factors that affect electricity consumption and accurately forecast its usage. In this paper Neural Network methodology is proposed as an effective methodology for load forecasting

Keywords:- Neural Network, Random Forest, LSTM

1. INTRODUCTION

In the contemporary energy sector, the prediction of energy consumption is done by traditional forecasting methods

such as machine learning models. This paper aims at establishing the fact that Artificial Neural Networks play a better role in making more accurate predictions provided that there is no restriction on the time to make the predictions and resources in terms of hardware to be used

The electrical energy consumption depends on the following factors/ loads

1.1 Social Load

The consumption varies during the holidays, the weekdays and the weekends

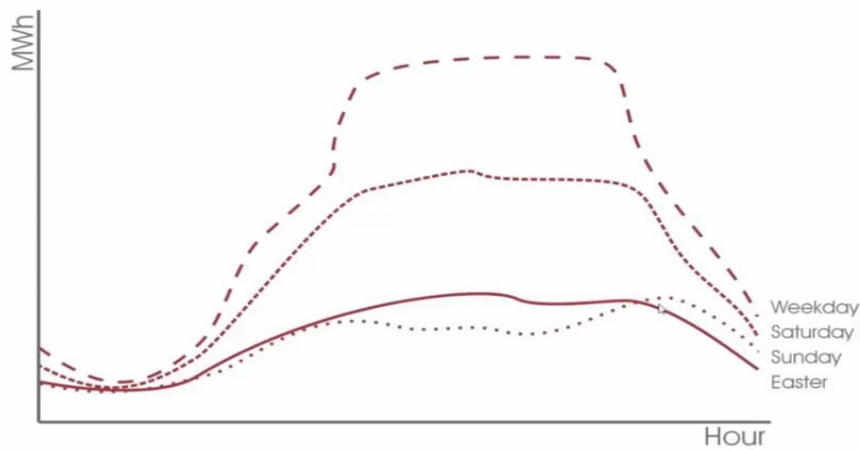


Fig 1: Variation of social load with time

1.2 Weather Dependent Load

Factors of weather are temperature, global radiation, wind, cloud etc. The load consumption increases due to high

temperature in summer or may increase due to more heater consumption during the winters

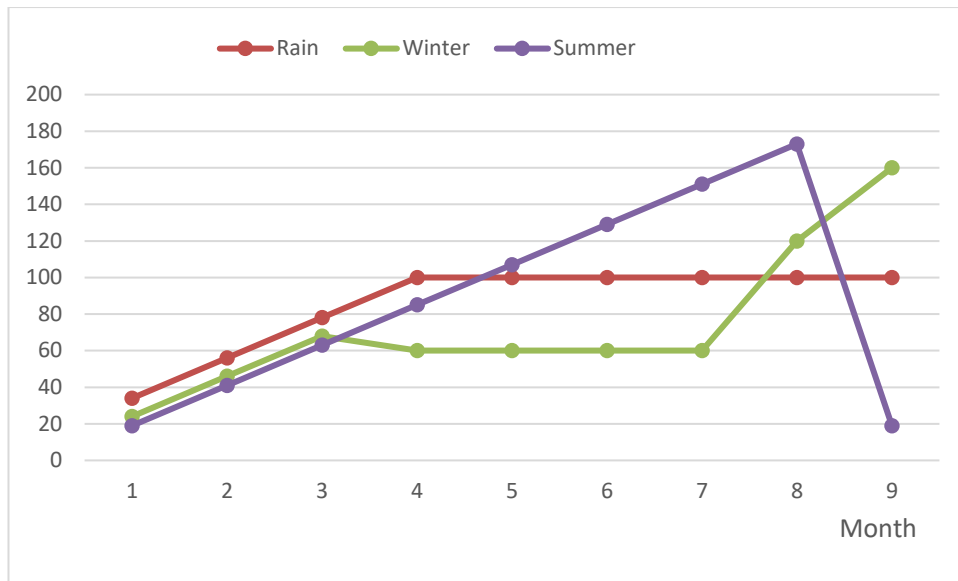


Figure 2-Variation of Weather Dependent Load

1.3 Stochastic Load

These types of loads are the ones that couldn't be accounted for e.g. a sudden storm has increased the load for a certain period of time. Now it would be necessary to understand that whether to take the load into consideration.

A hurricane might be a one of a phenomena in some place whereas it might be a regular phenomenon in some other place. As such a decision is necessary to be taken whether or not to account for these factors

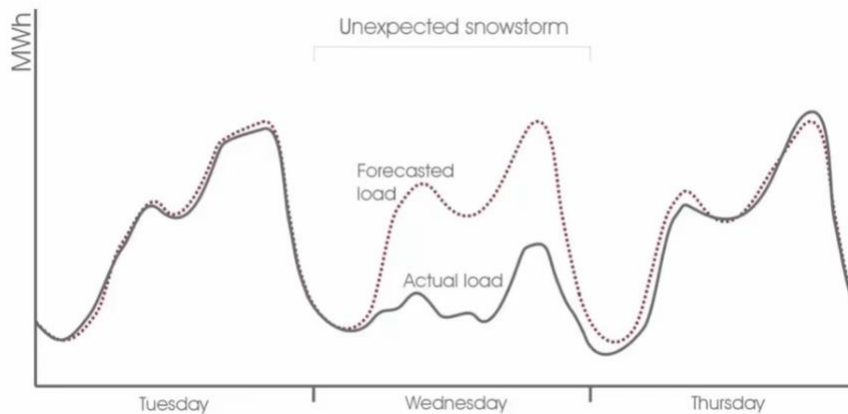


Fig 3: Variation of stochastic load

2. DATA USED

The data is fed into the machine learning model and was trained to identify the pattern so that the model is able to predict the usage amount for new databy date. The target variable is USAGE and the predictors are Date, Time, Global_active_power, Global_reactive_power, Voltage and Global_intensity.

If this energy consumption is analyzed over the days of the week, it can be seen that there is a difference in the pattern

of energy consumption on a weekday vs a weekend/holidays.

An additional column called IS_HOLIDAY is introduced to understand if the date is a holiday or not.

Now the usage values aggregated over one month, two months and so on till five months are introduced in the dataset

	TYPE	DATE	START TIME	END TIME	USAGE	UNITS	COST	NOTES	DATE_TIME	DAY_TYPE	IS_HOLIDAY	T_1	T_2	T_3	T_4	T_5
0	Electric usage	11/1/16	0:00	0:14	0.07	kWh	\$0.01	0	2016-11-01 00:14:00	0	0	0.00	0.00	0.00	0.00	0.00
1	Electric usage	11/1/16	0:15	0:29	0.05	kWh	\$0.01	0	2016-11-01 00:29:00	0	0	0.07	0.00	0.00	0.00	0.00
2	Electric usage	11/1/16	0:30	0:44	0.06	kWh	\$0.01	0	2016-11-01 00:44:00	0	0	0.05	0.07	0.00	0.00	0.00
3	Electric usage	11/1/16	0:45	0:59	0.05	kWh	\$0.01	0	2016-11-01 00:59:00	0	0	0.06	0.05	0.07	0.00	0.00
4	Electric usage	11/1/16	1:00	1:14	0.03	kWh	\$0.01	0	2016-11-01 01:14:00	0	0	0.05	0.06	0.05	0.07	0.00
5	Electric usage	11/1/16	1:15	1:29	0.03	kWh	\$0.01	0	2016-11-01 01:29:00	0	0	0.03	0.05	0.06	0.05	0.07
6	Electric usage	11/1/16	1:30	1:44	0.03	kWh	\$0.01	0	2016-11-01 01:44:00	0	0	0.03	0.03	0.05	0.06	0.05
7	Electric usage	11/1/16	1:45	1:59	0.06	kWh	\$0.01	0	2016-11-01 01:59:00	0	0	0.03	0.03	0.03	0.05	0.06
8	Electric usage	11/1/16	2:00	2:14	0.04	kWh	\$0.01	0	2016-11-01 02:14:00	0	0	0.06	0.03	0.03	0.03	0.05
9	Electric usage	11/1/16	2:15	2:29	0.03	kWh	\$0.01	0	2016-11-01 02:29:00	0	0	0.04	0.06	0.03	0.03	0.03

The consumption of energy over time is analyzed as follows

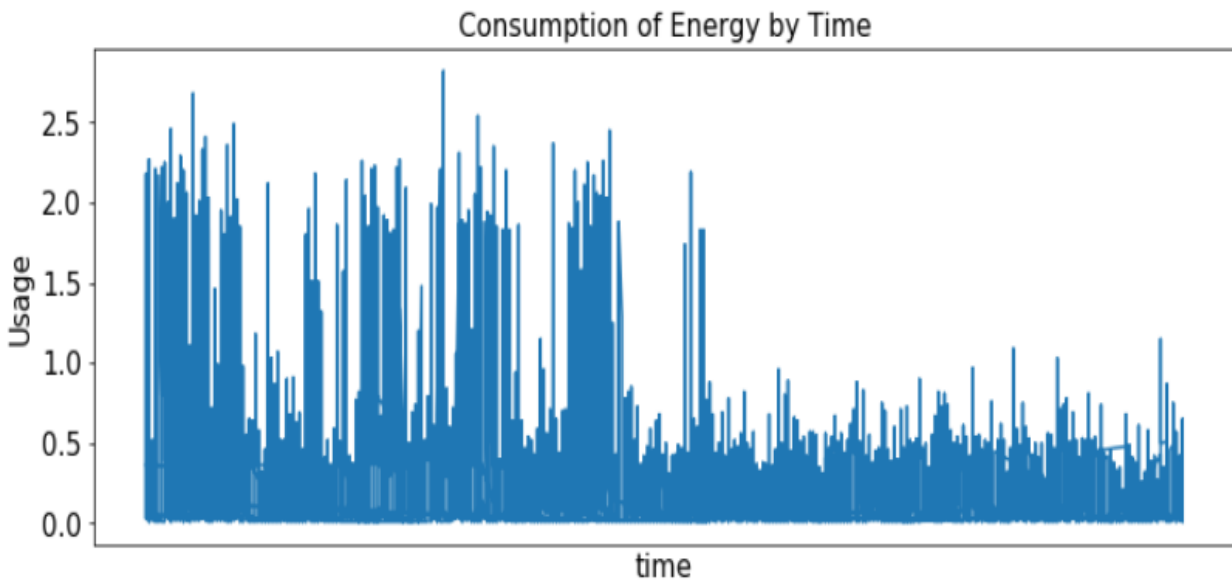


Fig 4: Consumption of energy over time

3. RANDOM FORESTSMETHODOLOGY

A random forest is an ensemble of decision trees. The final value of random forest is obtained by taking the average of each of the values of the decision tree.

The Random Forest algorithm adds extra randomness when growing trees. Instead of searching for the very best feature when splitting a node it searches for the best feature among

a random subset of features. As a result it creates a greater tree diversity, which introduces higher bias and a lower variance, generally yielding an overall better model.

The features derived as explained above are used for making the predictions using the random forest algorithm.

With Random Forest Regressor, the following actual against predicted result is obtained

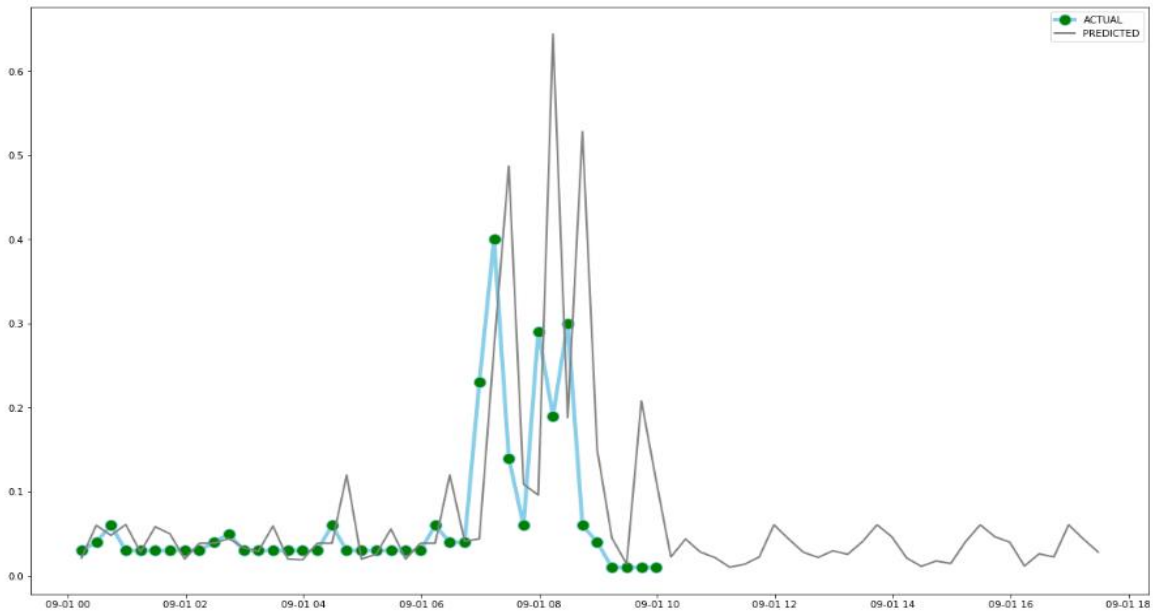


Fig 5: Result from Random Forest Prediction

It can be seen that the model predicted the values accurately to a certain extent. The grey line is the prediction value on new data. The validation score on the validation data is

$$RMSE = 0.18412616700088405$$

4. SOLVING WITH NEURAL NETWORK APPROACH

Neural networks is another approach and has been widely used for time series forecasting. These are feed-forward

networks which employ a sliding window over the input sequence. Common examples of this approach are stock market predictions and predicting telecommunication load [1, 2]. The neural network based prediction can be considered as a model in which function f is assumed to be a non-linear combination of a fixed number of previously input values [3].

The neural network is constructed as a three layer architecture as depicted in the following diagram

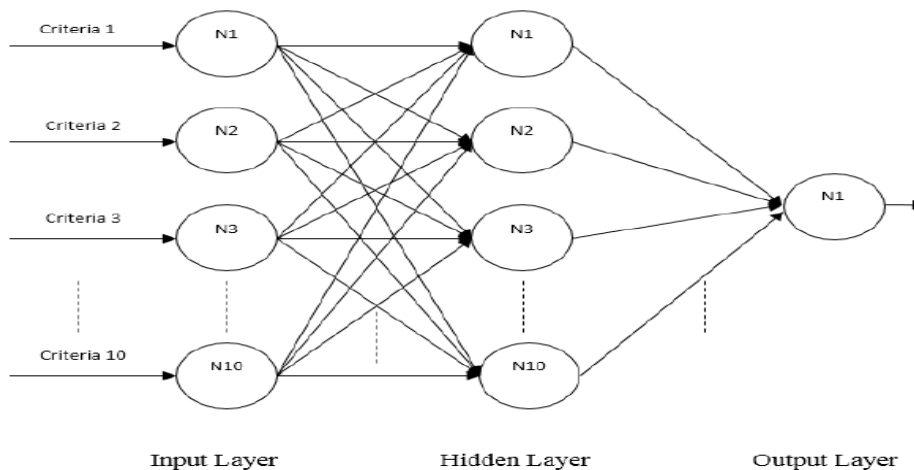


Fig 6: Neural Network Model

Loss function: Mean Squared Error (MSE) Activation function: ReLU for hidden layers and linear for the output layer Number of epochs=1000

Batch size (for batch gradient descent) =250 Optimizer=adam

On training the network on the train data, the following validation error is obtained

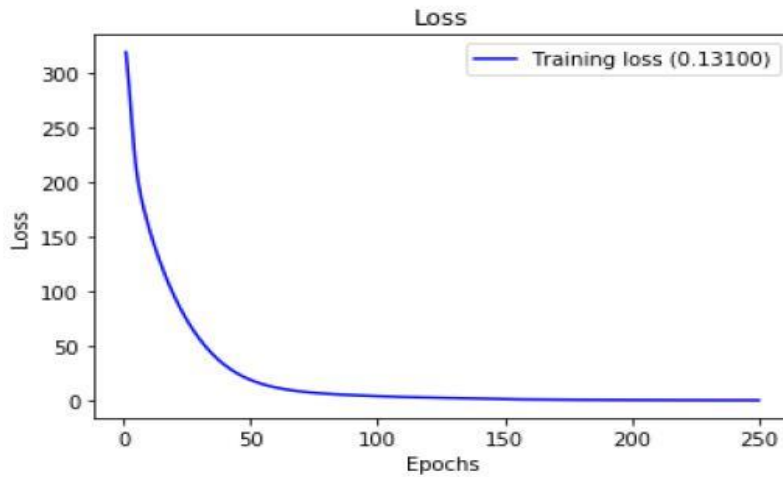


Fig 7: Variation of train data

It is seen that the training loss decreased over time

The predicted data

The below graph shows the result on how the model predicted the train data and the grey line represents the prediction on new unseen data

The accuracy of the model is RMSE= 0.02003

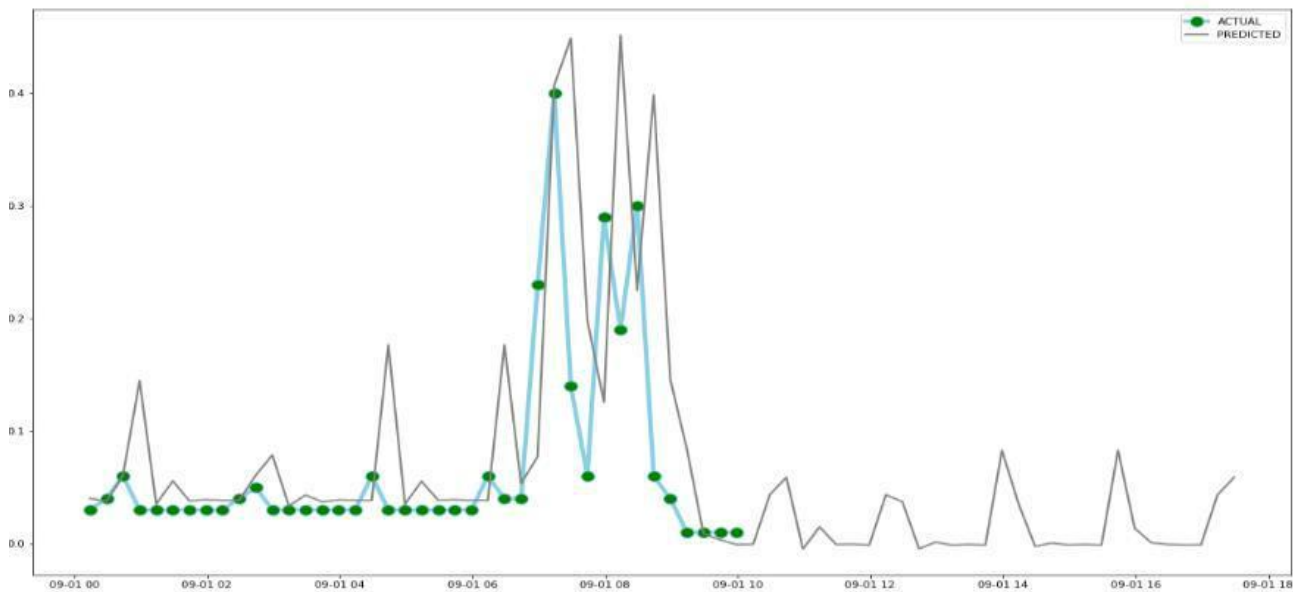


Figure 8-Result from Neural Network

Artificial Neural Networks: LSTM

Ho and Xie performed a comparative study between ARIMA and RNN showing that RNN has performed better than ARIMA[4] in modelling time series prediction. Connor and Martin also discussed the effectiveness of Neural Networks in time series forecasting [6]. The Long-Short-Term-Memory (LSTM) cell proposed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber [5] proved to be more effective than traditional recurrent neural networks in time series predictions. LSTM cell can be considered as a black box. It can be used very much like a basic cell, except it will perform much better; training will converge faster and it will detect long-term dependencies in the data. LSTM is a kind of Recurrent Neural Networks which is depicted by the following architecture:

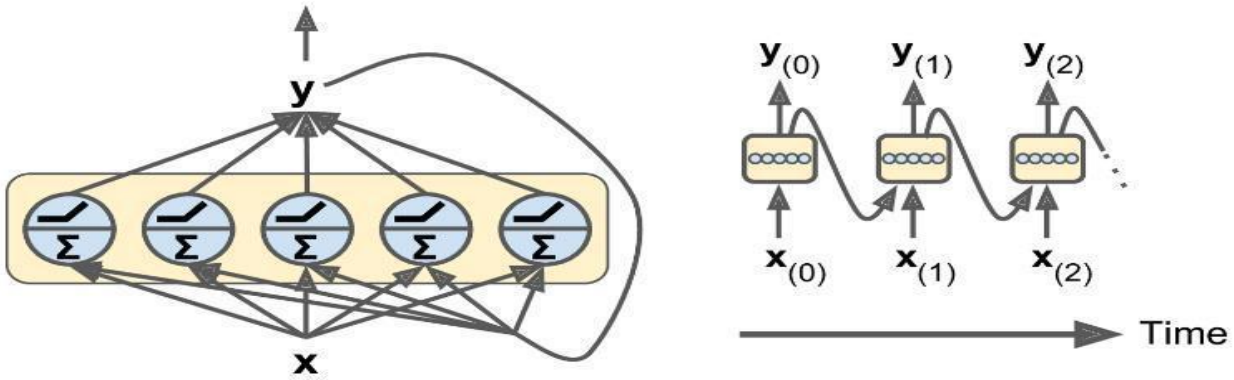


Figure 9-LSTM Architecture

In a Recurrent Neural Network, the output of the previous layer is fed into itself and also to the next layer. This type of architecture is very effective in time-series forecasting. The equations for a RNN architecture is as follows:

$$y_{(t)} = \phi(\mathbf{W}_x^T \cdot \mathbf{x}_{(t)} + \mathbf{W}_y^T \cdot \mathbf{y}_{(t-1)} + \mathbf{b})$$

$$\begin{aligned} \mathbf{Y}_{(t)} &= \phi(\mathbf{X}_{(t)} \cdot \mathbf{W}_x + \mathbf{Y}_{(t-1)} \cdot \mathbf{W}_y + \mathbf{b}) \\ &= \phi([\mathbf{X}_{(t)} \quad \mathbf{Y}_{(t-1)}] \cdot \mathbf{W} + \mathbf{b}) \text{ with } \mathbf{W} = \begin{bmatrix} \mathbf{W}_x \\ \mathbf{W}_y \end{bmatrix} \end{aligned}$$

- $\mathbf{Y}_{(t)}$ is an $m \times n$ neurons matrix containing the layer's outputs at time step t for each instance in the mini-batch (m is the number of instances in the mini-batch and n neurons is the number of neurons).
- $\mathbf{X}_{(t)}$ is an $m \times n$ inputs matrix containing the inputs for all instances (n inputs is the number of input features).
- \mathbf{W}_x is an $n \text{ inputs} \times n \text{ neurons}$ matrix containing the connection weights for the inputs of the current time step.
- \mathbf{W}_y is an $n \text{ neurons} \times n \text{ neurons}$ matrix containing the connection weights for the outputs of the previous time step.
- \mathbf{b} is a vector of size n neurons containing each neuron's bias term.
- The weight matrices \mathbf{W}_x and \mathbf{W}_y are often concatenated vertically into a single weight

matrix \mathbf{W} of shape $(n \text{ inputs} + n \text{ neurons}) \times n \text{ neurons}$

□ The notation $[\mathbf{X}_{(t)} \mathbf{Y}_{(t-1)}]$ represents the

horizontal concatenation of the matrices $\mathbf{X}_{(t)}$ and $\mathbf{Y}_{(t-1)}$

The basic LSTM architecture is depicted as follows:

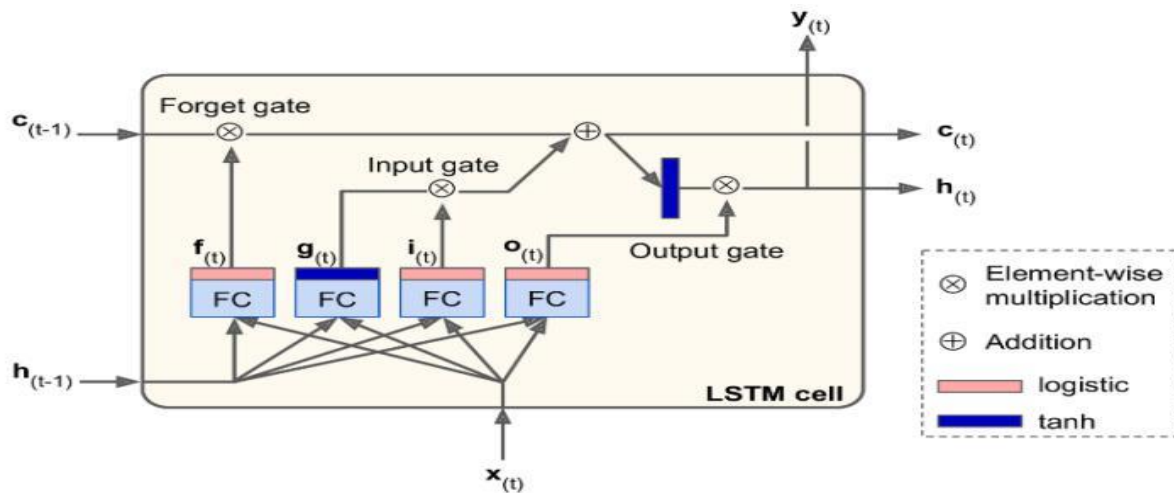


Figure 10-LSTM Gates

The LSTM computations are as follows:

$$\begin{aligned}
 i_{(t)} &= \sigma(W_{xi}^T \cdot x_{(t)} + W_{hi}^T \cdot h_{(t-1)} + b_i) \\
 f_{(t)} &= \sigma(W_{xf}^T \cdot x_{(t)} + W_{hf}^T \cdot h_{(t-1)} + b_f) \\
 o_{(t)} &= \sigma(W_{xo}^T \cdot x_{(t)} + W_{ho}^T \cdot h_{(t-1)} + b_o) \\
 g_{(t)} &= \tanh(W_{xg}^T \cdot x_{(t)} + W_{hg}^T \cdot h_{(t-1)} + b_g) \\
 c_{(t)} &= f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \\
 y_{(t)} &= h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)})
 \end{aligned}$$

- $W_{xi}, W_{xf}, W_{xo}, W_{xg}$ are the weightmatrices of each of the four layers for their connection to the input vector $x(t)$.
- $W_{hi}, W_{hf}, W_{ho},$ and W_{hg} are the weightmatrices of each of the four layers for their connection to the previous short-term state $h(t-1)$.
- $b_i, b_f, b_o,$ and b_g are the bias terms for each of the four layers.

The training and validation losses using LSTM came down to as follows:

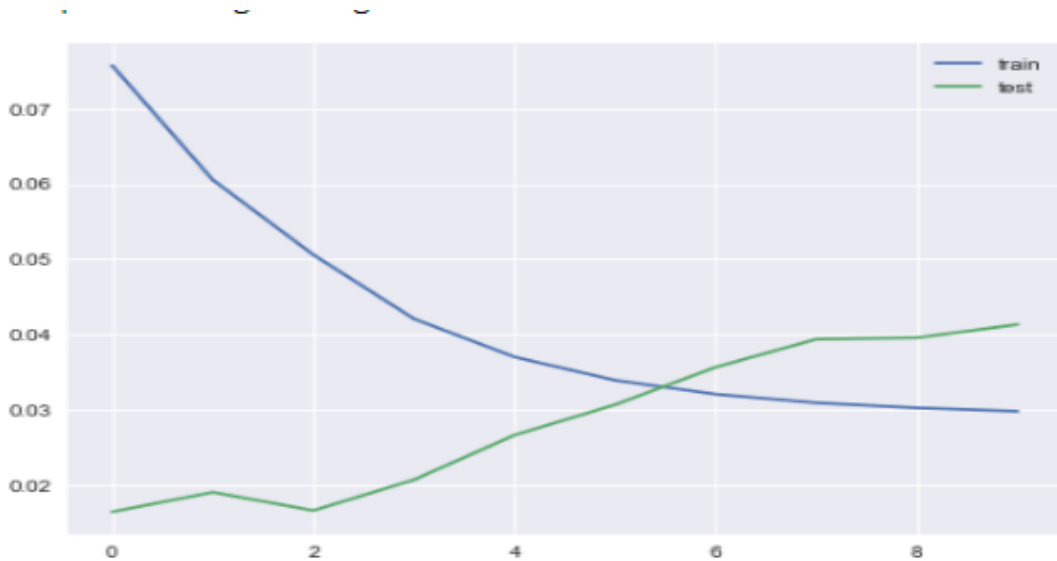


Figure 11-Train and Test Loss

The performance of the model is shown as below. The grey line is an approximation of how the model predicted the value in the training data against the actual value in the green line.

MSE = 0.1586783446921463

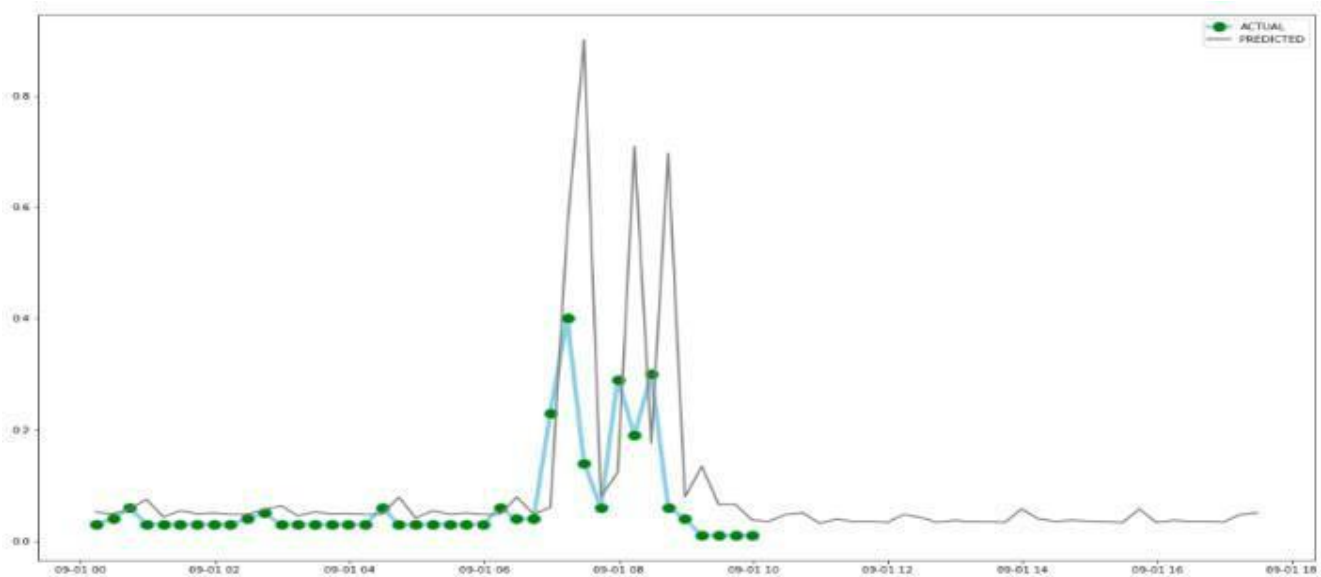


Figure 2-Result from LSTM

CONCLUSION

It can be concluded that the Artificial Neural Networks did a better job in terms of predicting the actual values than the random forest. If the emphasis is on predicting accurate value, the NN models to be used. However, if the data volume is huge and there is a constraint on how many servers can be used and how quickly the predictions can be made, a simpler machine learning algorithm such as lasso regression or Random Forest can be used.

REFERENCES

- [1] Bengio, S., Fessant, F., and Collobert, D.: A connectionist system for medium-term horizon time series prediction, in: Proc. of Internat. Workshop Application Neural Networks to Telecoms, 1995, pp. 308–315
- [2] Edwards, T., Tansley, D. S. W., Davey, N., and Frank, R. J.: Traffic trends analysis using neural networks, in: Proc. of the Internat. Workshop on Applications of Neural Networks to Telecommunications, Vol. 3, 1997, pp. 157–164
- [3] Dorffner, G.: Neural networks for time series processing, Neural Network World 4/96 (1996), pp. 447–468
- [4] Ho, Xie, Goh, April 11, Singapore, A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction, V42, pp. 371-375
- [5] SeppHochreiter and Jürgen chmidhuber, Nov, 1997, MIT, Neural Computation V9, pp. 1735-1780
- [6] Connor, Martin, Atlas, Mar 94, NJ, Recurrent neural networks and robust time Seriesprediction, pp.240-254