Study of power demand forecasting of a hospital by ensemble machine learning

Mayuka Nakai¹, Ryozo Ooka² and Shintaro Ikeda³

¹ Department of Architecture, The University of Tokyo, Japan
² Institute of Industrial Science, The University of Tokyo, Japan
³ Department of Innovation Science, Tokyo Institute of Technology, Japan

E-mail: nakai-mayuka448@g.ecc.u-tokyo.ac.jp

Abstract. To save energy in existing buildings, power demand can be predicted so a more efficient operation of equipment can be realized, like utilizing heat storage to lower the peak. Many attempts to predict building power consumption by machine learning have used simulation values in virtual buildings with no measurement errors or defects in the data. These models tend to have higher accuracy scores but have the risk of overfitting and possibly malfunction for missing data or outliers. To avoid the problems, this study proposes an ensemble machine learning algorithm to forecast power demand for a hospital building in Japan. Using the power consumption data, predictions were made by using algorithms such as Deep Neural Network (DNN) and Random Forest (RF). Each algorithm was combined to create ensemble models that take the weighted average of the predicted values. Consequently, we overcame the issues of each individual method, and achieved higher prediction accuracies. We selected the appropriate method for forecasting the power demand of real buildings based on accuracy. In future studies, we will apply the same methodology to predict cooling load.

1. Background
In order to prevent global warming and solve issues for a sustainable society, it is important not only to save energy in the case of new construction, but also to reduce energy consumption during operation of existing buildings. For that purpose, it is effective to reduce the peak demand and improve the operating efficiency of the equipment by predicting how much power demand is expected with respect to the schedule for operating the building and the load generated there. In recent years, there has been a movement to incorporate machine learning into forecasting demand for electric power. However, most of the research on building power consumption predictions by machine learning have used simulation values based on physical modeling of buildings. Although, the simulation values cannot have measurement errors or defects, it can include the system error or bias based on inaccuracies of modeling process. The statistical approach based on measurement is now evaluated again.

The purpose of this study is to clarify an appropriate machine learning method in power demand forecasting using only actual data.

2. Methodology
Machine learning algorithms used in this research includes, Deep Neural Network (DNN), Random Forest (RF) and Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). First, each algorithm is trained and predicted individually and the accuracy is compared. If there is no big difference in accuracy, it is judged by the difference in run time. Run time represents the time taken for learning.
Afterwards, we will try to further improve the accuracy by combining each machine learning method. Finally, by comparing the accuracy of each using Expected Error Percentage (EEP), we clarified an appropriate machine learning method for forecasting the power demand of buildings.

The root mean square error (RMSE) is the square root of the mean square error. In equation (1), the number of predicted values is n, the correct value is C, and the predicted output value is P, and when the prediction was started is i. EEP is RMSE divided by the maximum value $C_{\text{max}}$ in the power consumption test data.[1] However, this time, the average of the EEPs and RMSE of the entire test data is called the average EEP and average RMSE, and this is compared. The LSTM and GRU compare the prediction accuracy of only the prediction target time, and in the best model, the first prediction result is used as the latest power consumption for the next prediction, and the prediction is made for 30 hours. At this time, data such as the day of the week was not moved.

$$RMSE = \left( \frac{1}{n} \sum_{i=1}^{n} (C_i - P_i)^2 \right)^{1/2} \quad (1)$$

$$EEP = \frac{RMSE}{C_{\text{max}}} \times 100 \% \quad (2)$$

2.1. Data content
We used data for about four and a half years from 12:00 a.m. on April 1, 2016 to 5:00 p.m. on September 15, 2020 at a Hospital in, Ibaraki, Japan with a total floor area of 79,604m². However, the data with major defects are excluded (from 8:00 a.m. on August 11, 2017 to 11:30 p.m. on August 15, 2017, and from 11:00 p.m. on September 16, 2019 to October 2, 2019. Until 3:00 p.m.). It is randomly divided from all data so that train data: test data becomes 8: 2.

The raw data used for DNN and RF consists of the date, time, day of the week, whether it is a public holiday, the outside temperature, and the total value of power consumption at five observation points obtained every 30 minutes. Entering whether 12 hours later is a public holiday is aimed at improving the response to public holidays with a short number of days. The outside temperature is referenced to 30 hours after the predicted time, and the power consumption is referenced from 72 hours before the predicted time. Year data is omitted when studying. The processed data is by standardizing the data, outside temperature, and power consumption data and the day of the week is expressed as a one-hot vector.

In the learning of LSTM and GRU, I'm referring to the data on whether it is a day of the week, a public holiday, the outside temperature 30 hours ahead including now, and the total power consumption of 5 observation points 30 hours ago not including now is used.

2.2. Settings for each learning method

2.2.1. DNN model. DNN is a machine learning method modelled off the brain of an organism, and expresses neuron connection by a computational coefficient (weight) that can convey input and a function at the time of output.[2][3] Therefore, the input shape, the number of neurons, the weight, and the activation function determine the quality of the DNN model. For this study the DNN model predicts the power demand for 30 hours from the forecast target time at once. The input layer was 1 layer, the hidden layer was 2 layers, the output layer was 1 layer, the learning epoch was 30,000, and the early termination was when the minimum validation loss value was not updated during 500 epochs, the activation function was Scaled Exponential Linear Unit (SELU). The SELU is similar to the general activation function REctified Linear Unit (RELU).[2] The formula of SELU is:

$$SELU(x) = \begin{cases} \lambda x, & x > 0 \\ \lambda a e^x - \lambda a , & x \leq 0 \end{cases} \quad (3)$$
Where $\lambda \approx 1.050701$, $\alpha \approx 1.673263$. [4] By increasing the number of epochs to a huge number of 30,000, it continues until learning converges. Adam was used as the optimization function, and the learning rate was 0.005.

2.2.1.1. Check the effect of data pre-processing. Table 1 shows the cases comparing. Predictions made depending on whether raw data was used or processing data was used, assuming that the number of nodes in the hidden layer was 100 and the batch size was 32, and the improvement in accuracy due to pre-processing was confirmed.

2.2.1.2. Examination of the need for batch regularization. Using the processed data, with the number of nodes in the middle layer set to 100 and the batch size set to 32, Table 2 shows two cases, with or without batch regularization, and examine the necessity of batch regularization for power demand forecasting.

2.2.1.3 Determining the parameters. Reflecting the case with higher accuracy from 2.2.1.2, Table 3 shows the cases comparing the number of nodes in each layer by either 100 or 200, and the batch size by 32 or 64, and the optimum parameter is determined.

### Table 1. Case of checking the effect of data pre-processing.

| Case       | Node | Batch | Data          |
|------------|------|-------|---------------|
| DNNcase1-1 | 100  | 32    | Raw data      |
| DNNcase1-2 | 100  | 32    | Processed data|

### Table 2. Case of checking the effect of batch regularization.

| Case       | Node | Batch | Batch regularization |
|------------|------|-------|----------------------|
| DNNcase2-1 | 100  | 32    | With                |
| DNNcase2-2 | 100  | 32    | Without             |

### Table 3. Case of checking the effect of data pre-processing.

| Case       | Node | Batch |
|------------|------|-------|
| DNNcase3-1 | 100  | 32    |
| DNNcase3-2 | 200  | 32    |
| DNNcase3-3 | 100  | 64    |
| DNNcase3-4 | 200  | 64    |

2.2.2 RF model. The RF is a machine learning method that attempts to give a better answer by creating multiple decision trees, which are machine learning methods that simulate trees, and taking a majority vote. By randomly selecting the features so that the decision trees do not resemble each other, it is less susceptible to noise. Therefore, the parameters examined by RF include the number of decision trees, the maximum depth (how far the tree grows), and the maximum number of features (how many variables are used). There is a minimum number of samples (how many samples make a tree at the tip of a branch). [3][5] In this study, the number and maximum depth of decision trees are fixed at 50 and only max features and min samples split are examined. RF model also predicts the power demand for 30 hours from the forecast target time.

2.2.2.1 Check the effect of data pre-processing. As in 2.2.1.1, with max features set to 5 and min samples split set to 4, predictions are made when using raw data and when processing data is used, and the improvement in accuracy due to pre-processing is confirmed. Table 4 shows these cases.

2.2.2.2 Determining the max features. RF cases 2-1 to 2-15 are for max features 5 to 100. Table 5 shows these cases. Here, min samples split was fixed at 4.

2.2.2.3 Determining minimum samples split. Reflecting max features with short calculation time and high accuracy in the case of 2.2.2.1, RF cases 3-1 for min samples split as 4 and RF cases 3-2 for min...
samples split as 5. RF cases 3-3 for min samples split as 6 and RF cases 3-4 for min samples split as 7, and RF cases 3-5 for min samples split as 8 are designated. Table 6 shows these cases.

| Case            | Max features | Min samples split | Data          |
|-----------------|--------------|-------------------|---------------|
| RFcase1-1       | 5            | 4                 | Raw data      |
| RFcase1-2       | 5            | 4                 | Processed data|

Table 5. Case of determining the max features.

| Case      | Max features | Min samples split | Data       |
|-----------|--------------|-------------------|------------|
| RFcase2-1 | 5            | 4                 |            |
| RFcase2-2 | 10           | 4                 |            |
| RFcase2-3 | 15           | 4                 |            |
| RFcase2-4 | 20           | 4                 |            |
| RFcase2-5 | 25           | 4                 |            |
| RFcase2-6 | 30           | 4                 |            |
| RFcase2-7 | 35           | 4                 |            |
| RFcase2-8 | 40           | 4                 |            |
| RFcase2-9 | 45           | 4                 |            |
| RFcase2-10 | 50           | 4                 |            |
| RFcase2-11 | 60           | 4                 |            |
| RFcase2-12 | 70           | 4                 |            |
| RFcase2-13 | 80           | 4                 |            |
| RFcase2-14 | 90           | 4                 |            |
| RFcase2-15 | 100          | 4                 |            |

Table 6. Case of determining the min samples split.

| Case      | Max features | Min samples split | Data       |
|-----------|--------------|-------------------|------------|
| RFcase3-1 | 90           | 4                 |            |
| RFcase3-2 | 90           | 5                 |            |
| RFcase3-3 | 90           | 6                 |            |
| RFcase3-4 | 90           | 7                 |            |
| RFcase3-5 | 90           | 8                 |            |

2.2.3. **LSTM model.** The LSTM is based on the Recurrent Neural Network (RNN)'s idea.[6] It is a method proposed to prevent vanishing gradient and gradient explosion problems with long-term time dependence in RNN, and is generated by replacing the hidden layer of RNN with a circuit called LSTM block.[6] In the LSTM model, only the target time is predicted. The number of dimensions in the input data is 4, the number of dimensions in the output data is 1, the number of hidden layer units is 50, the length of time series is 60, the batch size is 100, the number of learning epochs is 10000, and the optimization function is Adam with the learning rate of 0.0005. The parameters consider two things, the early termination is the case where the minimum validation loss is not updated during how many epochs, and the activation function.

| Case      | Activation function |
|-----------|---------------------|
| LSTMcase1-1 | Tanh                |
| LSTMcase1-2 | SELU                |

2.2.3.1 Determining the activation function. First, patience is tentatively fixed at 25. The activation function is hyperbolic tangent(tanh) in LSTM case 1-1 SELU in LSTM case 1-2. Table 7 shows these cases.

2.2.3.2 Determining early termination time. The activation function is which had a short calculation time and high accuracy in 2.2.3.1. Table 8 shows LSTM cases 2-1 to 2-4 for 10 to 100 epochs.

| Case      | Epochs |
|-----------|--------|
| LSTMcase2-1 | 10     |
| LSTMcase2-2 | 25     |
| LSTMcase2-3 | 50     |
| LSTMcase2-4 | 100    |

2.2.4 **GRU model.** GRU is a simpler version of LSTM, which requires less computation due to the reduced number of parameters.[6] The GRU model also predicts only the time to be predicted. Also, as in 2.2.3, in GRU, we will consider the case where the minimum validation loss is not updated during
which epoch, and the activation function. Other parameters are the same as 2.2.3. Previous studies have suggested that GRU is compatible with SELU for the activation function.[4]

2.2.4.1 Determining the activation function. The patience, which had a short calculation time and high accuracy in 2.2.3.1, was used. The activation function was tanh in GRU case 1-1, and SELU in GRU case 1-2. Table 9 shows these cases.

2.2.4.2 Determining early termination time. Reflecting the activation function that was highly accurate in 2.2.3.1, Table 10 shows GRU cases 2-1 to 2-4 for 10 to 100 epochs.

Table 9. Case of determining the activation function.

| Case       | Activation function |
|------------|---------------------|
| GRUcase1-1 | Tanh                |
| GRUcase1-2 | SELU                |

Table 10. Case of determining the early termination.

| Case       | Epochs |
|------------|--------|
| GRUcase2-1 | 10     |
| GRUcase2-2 | 25     |
| GRUcase2-3 | 50     |
| GRUcase2-4 | 100    |

2.3. Details of ensemble learning

Ensemble of the best case predictions was determined by the weighted average of the DNN and the RF model or the LSTM and the GRU model. DNN : RF rates are as shown in Table 11. And LSTM : GRU rates are as shown in Table 12.

Table 11. Case of determining the DNN and RF ensemble rate.

| Case           | Rate(DNN:RF) |
|----------------|--------------|
| Ensemblecase1-1| 9:1          |
| Ensemblecase1-2| 8:2          |
| Ensemblecase1-3| 7:3          |
| Ensemblecase1-4| 6:4          |
| Ensemblecase1-5| 5:5          |
| Ensemblecase1-6| 4:6          |
| Ensemblecase1-7| 3:7          |
| Ensemblecase1-8| 2:8          |
| Ensemblecase1-9| 1:9          |

Table 12. Case of determining the LSTM and GRU ensemble rate.

| Case           | Rate(LSTM:GRU) |
|----------------|----------------|
| Ensemblecase2-1| 9:1            |
| Ensemblecase2-2| 8:2            |
| Ensemblecase2-3| 7:3            |
| Ensemblecase2-4| 6:4            |
| Ensemblecase2-5| 5:5            |
| Ensemblecase2-6| 4:6            |
| Ensemblecase2-7| 3:7            |
| Ensemblecase2-8| 2:8            |
| Ensemblecase2-9| 1:9            |

3. Results and Discussion

The prediction results of electricity demand of the best cases of these single models during the representative days are shown in Figure 1. The results of DNN, RF, and GRU shows fairly good agreement with observation, on the other hand the prediction accuracy of LSTM is low. Figure 2 compares the average values of EEP of the best cases of these single models. The result of LSTM shows the worst. In this study, the LSTM and GRU data are not standardized, so it is necessary to consider them in the future. The characteristics of each model is discussed below.
3.1. DNN

The results of DNN cases are shown in Table 13. DNN case 1-2 had an average EEP of 0.64 point lower than DNN case 1-1, indicating that pre-processing improves prediction accuracy. The average EEP of DNN case 2-1 was 0.09 point lower than that of DNN case 2-2, and it was found that batch regularization was not necessary. And in DNN Case 3-4, where the number of nodes in the middle layer was 200 and the batch size was 64, the average EEP was less than 1.55% and the calculation time was short, so this parameter is considered to be optimal for this DNN model.

Table 13. Average EEP and run time.

| Case      | Average EEP[\%] | Run time[s] |
|-----------|-----------------|-------------|
| DNNcase1-1| 2.32            | 9265.5      |
| DNNcase1-2| 1.68            | 8696.1      |
| DNNcase2-1| 1.77            | 19995.8     |
| DNNcase2-2| 1.68            | 8696.1      |
| DNNcase3-1| 1.68            | 8696.1      |
| DNNcase3-2| 1.50            | 13515.9     |
| DNNcase3-3| 1.73            | 2606.4      |
| DNNcase3-4| 1.52            | 3971.5      |
3.2. RF
The results of RF cases are shown in Table 14. Regarding the effect of data pre-processing, in RF, the average EEP decreased by 0.22 point, but there was almost no difference in the calculation time. It is thought that this is because the one-hot expression of the day of the week is effective, but the standardization of numerical values is not effective. RF cases 1-14 with max features of 90 had the shortest calculation time among cases with an average EEP of less than 1.9, so we found that 90 was appropriate for max features of this model. Even if min samples split was changed, the average EEP for the period did not change much, but the calculation time of RF case 2-3 which min samples split was 6 was shorter, so it was found that 6 was the most suitable for min samples split.

| Case        | Average EEP[%] | Run time[s] |
|-------------|----------------|-------------|
| RFcase1-1   | 3.62           | 69.91       |
| RFcase1-2   | 3.4            | 71.03       |
| RFcase2-1   | 3.4            | 71.03       |
| RFcase2-2   | 2.84           | 86.43       |
| RFcase2-3   | 2.52           | 190.7       |
| RFcase2-4   | 2.32           | 124.8       |
| RFcase2-5   | 2.20           | 213.9       |
| RFcase2-6   | 2.11           | 270.1       |
| RFcase2-7   | 2.05           | 303.7       |
| RFcase2-8   | 2.01           | 355.0       |
| RFcase2-9   | 1.98           | 360.9       |
| RFcase2-10  | 1.94           | 341.7       |
| RFcase2-11  | 1.92           | 400.3       |
| RFcase2-12  | 1.89           | 940.7       |
| RFcase2-13  | 1.88           | 962.8       |
| RFcase2-14  | 1.87           | 543.4       |
| RFcase2-15  | 1.86           | 1213        |
| RFcase3-1   | 1.87           | 543.4       |
| RFcase3-2   | 1.87           | 517.9       |
| RFcase3-3   | 1.87           | 509.3       |
| RFcase3-4   | 1.87           | 993.6       |
| RFcase3-5   | 1.87           | 1482        |

3.3. LSTM
The results of LSTM cases are shown in Table 15. Since the average EEP of tanh was smaller than that of the SELU function, it is considered that tanh is suitable for the activation function of LSTM. It took a long time, but it was found that the larger the patience, the smaller the average EEP. If you want the calculation time to be kept short, patience = 10 is sufficient, but this time, patience = 100, which has the smallest average EEP, is considered to be optimal.

| Case        | EEP[%] | Run time[s] |
|-------------|--------|-------------|
| LSTMcase1-1 | 1.89   | 2148.0      |
| LSTMcase1-2 | 3.34   | 10462       |
| LSTMcase2-1 | 1.92   | 1841.2      |
| LSTMcase2-2 | 1.89   | 2148.0      |
| LSTMcase2-3 | 1.84   | 3332.5      |
| LSTMcase2-4 | 1.66   | 5132.6      |

| Case        | EEP[%] | Run time[s] |
|-------------|--------|-------------|
| GRUcase1-1  | 2.95   | 18796       |
| GRUcase1-2  | 1.83   | 3168.0      |
| GRUcase2-1  | 1.67   | 1284.5      |
| GRUcase2-2  | 1.83   | 3168.0      |
| GRUcase2-3  | 1.47   | 7502.2      |
| GRUcase2-4  | 1.25   | 11581       |

3.4. GRU
The results of GRU cases are shown in Table 16. Since SELU had a smaller average EEP than the tanh function, it is considered that SELU is suitable for the activation function of GRU. And as with LSTMs,
it takes a considerable amount of time, but it was found that the larger the patience, the smaller the average EEP. If the calculation time should be kept short, patience = 10 is sufficient, but this time, patience = 100, which has the smallest average EEP, is optimal.

3.5. Results after ensemble

The results of each case are shown in Table 17 and 18. The average EEP of DNN is 0.01 points, RF is 0.36 points, LSTM is 7.83 points, and GRU is 0.01 points lower than before the ensemble.

4. Conclusion

In this study, several machine learning algorithms are applied to forecast power demand for a hospital building in Japan. Using the power consumption data of almost four and a half years, initial predictions were made by algorithms of DNN, RF, LSTM, and GRU. The results of DNN, RF, and GRU shows fairly good agreement with observation, on the other hand the prediction accuracy of LSTM is low. Furthermore, some algorithms were combined to create ensemble models that take the weighted average of the predicted values. As a result, we overcome the issues of each individual method, such as the delay in responding to sudden changes in numerical values with DNN or long calculation time with LSTM and achieved higher prediction accuracies. When the equipment is expanded, it may be possible to create a model faster and easier than creating a model from scratch by performing transfer learning. In future studies, we will apply the same methodology to predict steam, hot water, heating, and cooling load and employ transfer learning algorithms to forecast demand for other buildings.

References

[1] Chengliang Fan, Yundan Liao and Yunfei Ding, Development of a cooling load prediction model for air-conditioning system control of office buildings (International Journal of Low-Carbon Technologies, Volume 14, Issue 1, March 2019, Pages 70–75)

[2] Yukinaga Azuma, Deep Learning Neural Network & Backpropagation with Python, in Japanese (Publisher:SB Creative Inc. Issue4, March 2019)

[3] Tatsuki Yamaguchi and Hiroyuki Matsuda from Aidemy Inc., Illustrated immediate force A textbook that clearly understands the mechanism and technology of machine learning & deep learning in this one book, in Japanese (Publisher:Gijutsu Hyouron Sha Inc., Issue 2 Dec 2019)

[4] X R ZU and R X SONG, Short-term Wind Power Prediction Method Based on Wavelet Packet Decomposition and Improved GRU (Journal of Physics: Conference Series, Volume 1087, Issue 2, 2018)

[5] Akiyoshi Suto, Clearly understand Introduction to Machine learning using Python, in Japanese (Publisher: Impress Inc., Issue1, Oct 2020)

[6] Yusuke Sugomori, Detailed explanation Deep learning Time series data processing with TensorFlow/Keras·PyTorch, in Japanese (Mynavi Shuppan Inc., Issue1, Nov 2019)