ICIC Express Letters
Part B: Applications
Volume **15**, Number **2**, February **2024**

ICIC International ©2024 ISSN 2185-2766
pp. 179–186

FACTOR IDENTIFICATION AND PREDICTION OF POWER DEMAND USING SPARSE MODELING AI

Tatsuma Hachiya and Keiji Kamei*

Department of Integrated Systems Engineering
Nishinippon Institute of Technology
1-11, Aratsu, Kanda-machi, Miyako-gun, Fukuoka 800-0394, Japan
s202075@nishitech.ac.jp; *Corresponding author: kamei@nishitech.ac.jp

Received June 2023; accepted August 2023

Abstract. *We used sparse modeling AI to identify factors and predict in power demand. Our method is construction of weather condition clusters with SOM and learning with sparse modelization for that clusters. From the feature of sparse modeling AI, we can identify the factor and make highly accurate precision for power demand. As a result of experiment, sparse modeling AI succeeded in identifying the factors for power demand and achieving about 9% averaged absolute prediction error.*
Keywords: BPTT, eXplainable AI, SOM, RNN, Sparse modeling AI

1. **Introduction.** In recent years, as the number of applications of Artificial Intelligence (AI) increases, various AI problems have been pointed out. Among them is the problem of "unexplainable of basis for decision" in which the learned model becomes black box, so that the basis for AI decision when AI fails to decide for unknown data becomes invisible. This problem is a hindrance to the introduction of AI in various fields that require explanations of the basis for AI decisions, e.g., AI is introduced into the manufacturing process of industrial products because the basis for the occurrence of manufacturing defects becomes unclear. This black-boxing problem of hierarchical neural network is mentioned by the US DARPA in 2019, and focus is on eXplainable AI (XAI) research to solve. Ishikawa is one of the first researcher to address this problem in the 1990s, and he introduced the concept of "Forgetting" for learning of hierarchical neural networks as a method for construction of XAI [1]. This is called "Sparse Modeling" that introduces regularization term in the error (loss) function to attenuate weights and biases, so that only the necessary connections remain. This method, in which learning and sparsification with forgetting are performed simultaneously, is quite different from the recent AI sparsification by "pruning", in which small weights are eliminated after model construction. Loshchilov and Hutter examine the case of using the L2 norm of the weights for weights decaying [2]. They focus on the relationship between the optimizer and the L2 norm and show the effectiveness of the L2 norm in image recognition. On the other hand, we apply the L1 norm of weights for error functions in this research. Unlike the L2 norm, the L1 norm has the distinct meaning of a constant amount of decaying weights.

The goal of this research is to clarify demand factors and to improve prediction accuracy for power demand using the sparse modeling AI that eliminates unnecessary weights from general hierarchical AI model. We are provided power demand data for 2017 to 2019 years from power company. Therefore, demand data for 2017 and 2018 are used in this research. Weather condition data are also used for the corresponding time period. Because power demand is like time dependent wave, AI must be able to learn time series. In this research, we use Back Propagation Through Time (BPTT) that is type of Recurrent Neural Network

DOI: 10.24507/icicelb.15.02.179

(RNN), and it is one of time series AI. The results of former our research have shown that it is impossible to construct models that "learn and predict a certain period" such as one week [5]. For this, we focused on weather conditions, and we aimed to construct weather condition clusters using Self-Organizing Map (SOM). In previous research, hierarchical clustering by Ward's method was shown to be suitable for constructing weather condition clusters because of the absence of cluster labels. In this paper, we present clustering results, and to construct sparse modelized BPTT based on clustering results, we explain attributes such as humidity to affect power demand based on sparse modelized BPTT and show the results of power demand prediction.

Section 2 presents the learning method of sparse modeling. Section 3 presents the details of SOM. Section 4 explains the learning data for RNN, and then Section 5 shows the experiment and its results. Section 6 concludes and summarizes this paper.

2. **XAI: Sparse Modelized Deep RNN.**

In this section, we describe the structure and learning method of BPTT and sparse modeling method to construct XAI.

2.1. **Back propagation through time.**

Williams and Zipser's BPTT [3] uses Back Propagation (BP) proposed by Amari [4] with n time steps of time series data (called the number of time length n) and the output of the hidden layer one time step earlier as input. The connections take the form of sharing among time series data, recursively going back through the time series to learn the temporal relationship with output of the hidden layer one time ago. Therefore, this is a type of RNN that can learn time dependent "time series information". The structure of the BPTT with time length number n is shown in Figure 1.

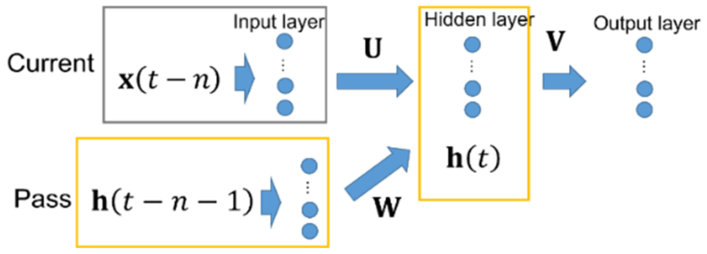


FIGURE 1. BPTT structure going back n time data

The BPTT in Figure 1 is represented as having a multi layer structure with one layer for each data for the number of data n to be traced back. As mentioned above, this has a structure in which the weights, $\mathbf{U}$ and $\mathbf{W}$, are shared among the time series data. The basic structure is identical to the 3 layer BP, and so this is also the factor that enables learning for each cluster of weather conditions in this research. On the other hand, Figure 1 shows that the BPTT has the same structure as the 3 layer BP, but is recursively expanded in time by the number of data going back. Therefore, as the number of time length increases, vanishing gradient problem occurs in deep layers of simple BP. So, BPTT can learn relatively short term time series data. Weights and biases are updated by Equations (1) to (5).

$$\mathbf{U}(t+1) = \mathbf{U}(t) - \eta \sum_{z=0}^{\tau} \mathbf{e_h}(t-z)\mathbf{x}(t-z)^{\mathrm{T}} \tag{1}$$

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta \sum_{z=0}^{\tau} \mathbf{e_h}(t-z)\mathbf{h}(t-z-1)^{\mathrm{T}} \tag{2}$$

$$\mathbf{V}(t+1) = \mathbf{V}(t) - \eta \mathbf{e_o}(t)\mathbf{h}(t)^{\mathrm{T}} \tag{3}$$

$$\mathbf{b}(t+1) = \mathbf{b}(t) - \eta \sum_{z=0}^{\tau} \mathbf{e_h}(y-z) \tag{4}$$

$$\mathbf{c}(t+1) = \mathbf{c}(t) - \eta \mathbf{e_o}(t-z) \tag{5}$$

where $\mathbf{x}(t)$ is the input at t, $\mathbf{b}$ is the bias of the hidden layer neurons (one for each neuron, expressed as a vector), $\mathbf{c}$ is the bias of the output layer neurons (as in the hidden layer), $\mathbf{e_h}$ is the output error of the hidden layer, $\mathbf{e_o}$ is the error of the output layer, $\mathbf{h}(t)$ is the output of the hidden layer at t, τ is the number of time evolutions, and η is the learning coefficient.

2.2. **Sparse modeling.** Sparse modeling is first introduced by Ishikawa in 1989 as "Structured learning with forgetting" [1] in connectionist models. That introduces the "L1 norm" which is the absolute sum of the weights and biases, as a regularization term in the error function (L1 regularization). The L1 regularization term forces unnecessary weights for logistic regression to approach zero, and that term keeps only necessary connections between layers; for this, L1 regularization is able to efficiently construct the network structure that is required for logistic regression. For sparse modelization using L1 regularization for BPTT, the L1 regularization term in Equation (6) is added to the error (loss) function.

$$l_1 = \lambda \left(\sum_{j,i} |u_{j,i}| + \sum_{j,j'} |w_{j,j'}| + \sum_{k,j} |v_{k,j}| + \sum_{j} |b_j| + \sum_{k} |c_k| \right) \tag{6}$$

where λ is the strength of regularization. u, w, v, b, c correspond to elements of $\mathbf{U}$, $\mathbf{W}$, $\mathbf{V}$, $\mathbf{b}$, $\mathbf{c}$, respectively. Minimizing the error function with L1 regularization term requires not only MSE minimization but also simultaneous minimization of the regularization term, so that an unregulated increase or decrease in weights is constrained. On the other hand, logistic regression uses gradient descent iteration, unlike linear regression which allows analytical regression with normal equations. Therefore, L1 regularization will become problems to MSE minimization. For this, after the L1 regularization, selective L1 regularization is performed only for connections that are smaller than the threshold as Equation (7).

$$sl_1 = \lambda \left(\sum_{|u_{j,i}|<\theta} |u_{j,i}| + \sum_{|w_{j,j'}|<\theta} |w_{j,j'}| + \sum_{|v_{k,j}|<\theta} |v_{k,j}| + \sum_{|b_j|<\theta} |b_j| + \sum_{|c_k|<\theta} |c_k| \right) \tag{7}$$

where θ is the threshold for selecting target. The other variables are the same as in Equation (6). In prediction for power demand, unlike the classification problem, real values must be predicted, so Mean Squared Error (MSE) in Equation (8) is used for the error.

$$MSE = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} \left(y_k^{(n)}(t) - t_k^{(n)}(t) \right) \tag{8}$$

where K is the size of the output layer, N is the number of data, $y_k(t)$ is the output of the output layer at time t, and $t_k(t)$ is the output target.

After L1 regularization and selective L1 regularization are applied to the model, only the essential connections for regression between input and hidden layer in $\mathbf{U}$ remain, so that it can be said that attributes of input layer for those connections indicate the factors for AI decisions. In addition, necessary connections in $\mathbf{W}$ and $\mathbf{V}$ remain as well as $\mathbf{U}$, so the method of sparse modeling can be expected to improve prediction accuracy.

3. **Self-Organizing Map.** Results of former research have shown that it is not possible to construct AI model that learns and predicts for a certain period. And that results have also shown that construction of weather condition clusters with SOM and modelization for weather condition clusters is effective. Figure 2 illustrates SOM structure. SOM has an input layer consisting of one neuron and competitive layer consisting of multiple neurons. In Figure 2, $\mathbf{x}_n$ is the data space vector element possessed by the input layer. $\mathbf{w}_{i_n}$ is the data space vector (reference vector) elements of the competing layer neurons.

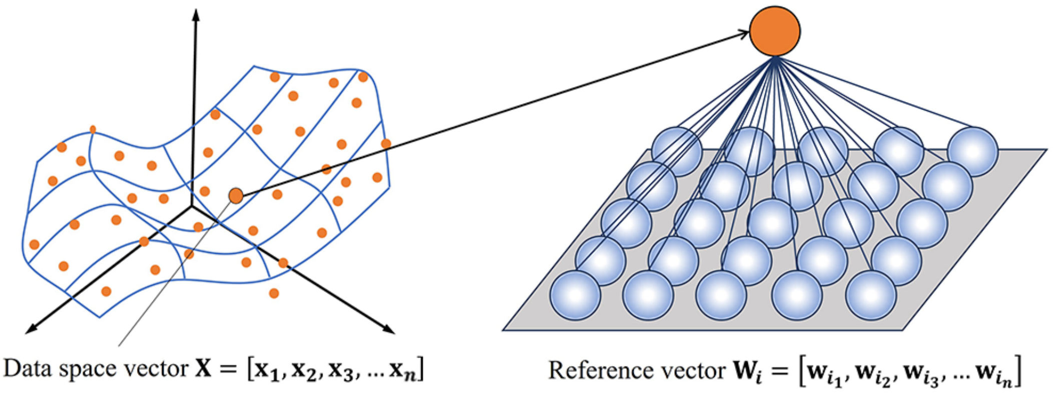

Data space vector $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots \mathbf{x}_n]$
Reference vector $\mathbf{W}_i = [\mathbf{w}_{i_1}, \mathbf{w}_{i_2}, \mathbf{w}_{i_3}, \dots \mathbf{w}_{i_n}]$

FIGURE 2. Structure of SOM

The neurons in competitive layer have reference (weight) vector with the same element size as data space. SOM is introducing the idea of neighborhood on feature space, so that SOM can map a training dataset to feature space with preserving topology among training datasets on data space. This is because neurons which are neighboring to the best matching neuron are also updated based on distance on feature space. Therefore, neurons that have similar reference vectors gather within local area on feature space; on the other hand, neurons that have big different reference vectors are never adjacent to each other. The SOM training is as below. Firstly, a training data is given to the input layer, and one competitive layer neuron with the best fit is selected as the winner neuron. Then, the winner neuron and its neighbors in the feature space update their own reference vector based on the training data. In this research, batch SOM as following training flow is applied.

(i) Random initialization of reference vectors for competing layer neurons.

(ii) Determination of winner neurons:

$$k^*(i) = \arg\min \|\mathbf{w}_{\boldsymbol{k}} - \mathbf{x}_{\boldsymbol{i}}\| \tag{9}$$

(iii) Determination of the amount of update for each competing layer neuron:

$$\alpha_{k,i} = \frac{h\left(\|y_k - y_{k^*(i)}\|; \sigma(t)\right)}{\sum_{i'=1}^{N} h\left(\|y_k - y_{k^*(i')}\|; \sigma(t)\right)} \tag{10}$$

(iv) Update reference vector:

$$\mathbf{w}_{\boldsymbol{k}}(t) = (1-\varepsilon)\mathbf{w}_{\boldsymbol{k}}(t-1) + \varepsilon \sum_{i=1}^{N} \alpha_{k,i}\mathbf{x}_{\boldsymbol{i}} \tag{11}$$

(v) Return to determining the winner neuron

Here, k is the position of the competing layer neuron, k^* is the position of the winner neuron, $\alpha_{k,i}$ is the update amount of training data i for the competing layer neuron k, $h(\cdot;\sigma)$ is the neighborhood function, and ε is the training coefficient.

A neighborhood function is defined as monotonically decreasing according to distance in the feature space. In this research, the Gaussian function of Equation (12) is used. σ

in Equation (12) is a function defining the neighborhood radius, which decreases monotonically according to training step t. The combination of the neighborhood function and the neighborhood radius function results in a wide range of neighborhoods treated as winner neuron neighborhoods at beginning stage of training, and the neighborhoods become narrower as training progresses. In this research, $\sigma(t)$ is defined by Equation (13).

$$h(t) = \exp\left(-\frac{\|y_k - y_{k^*(i)}\|}{2\sigma(t)^2}\right) \tag{12}$$

$$\sigma(t) = \sigma_f + (\sigma_s - \sigma_f)\exp\left(-\frac{t}{\tau}\right) \tag{13}$$

where σ_s is the neighborhood radius at the start of training, σ_f is the neighborhood radius at the end of training, t is the number of training sessions, and τ is the coefficient of decrease of the neighborhood radius.

If the larger τ, the slower the decrease. Generally, training data for SOM have a class label that indicates the class to which the data belongs, and a neuron in feature space is given a class label based on the best matching training data which has a class label. The problem in this research is that weather conditions change slowly with the seasons and time, so making it impossible to give class labels. Mapping to feature space by SOM is possible; however, it is not possible to construct a clustering map based on class labels. For this reason, a hierarchical clustering method called the "Ward's method" is applied. In this method, each of the learned feature space neurons is considered one cluster. The clusters with the smallest inter cluster distance are merged at the first higher level. By repeating this process of hierarchization, larger clusters are gradually formed. The clusters distances are shown in Equation (14). And, E in Equation (14) is Equation (15).

$$D(C_1, C_2) = E(C_1 \cup C_2) - E(C_1) - E(C_2) \tag{14}$$

$$E(C_i) = \sum_{x \in C_i} (D(x, c_i))^2 \tag{15}$$

where x is the data belonging to cluster C_i and c_i is the center of gravity of cluster C_i.

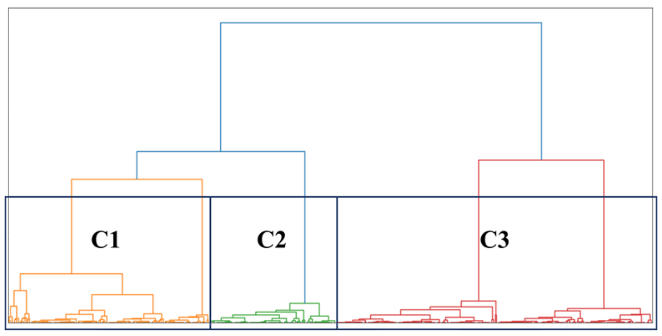
The result of hierarchical clustering by Ward's method is constructed as dendrogram. The vertical axis indicates the distance between clusters.

4. RNN Learning Data.

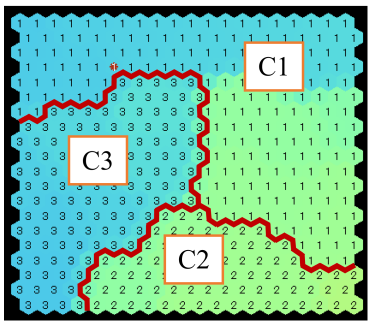
Generally, length of learning datasets for RNN is fixed among datasets. For example, number of time length is set to $n = 25$, and the 26th value is as the teacher signal. Then, start time for next time length and next teacher signal are shifted a time. In this research, it is impossible to construct AI model using time series learning with fixed time length, because the dates belonging to weather condition clusters are not continuous and the number of days that belong to a cluster varies. For example, if the weather conditions on January 1 and January 4 are very similar, they cannot be learned as continuous data because there is no temporal continuity between the two days. In addition, if a cluster has 5 days and the other has 10 days, learning with fixed time length is not possible because of the different length. To solve this problem, we focus on the structure of BPTT and use variable number of time length of data. It can be said that the BPTT is equivalent to a simple three layer hierarchical neural network. This is because it has a structure in which the weight, $\mathbf{U}$, between input and hidden layers and $\mathbf{W}$, between the hidden layers are shared among the time series data, as mentioned in Section 2.1. Therefore, the number of time length during model learning depends on the length of learning data. Since there is no dependence among learning data, it does not necessarily need to be fixed time length. So, we learn a model for a weather condition cluster by varying the number of time length for a day belonging to a certain weather condition cluster.

5. **Experiments.** We first construct weather condition clusters by SOM for the 45 days from July 30, 2017, and then identify the dates that fall into each cluster. Next, sparse modeling AI learning is performed using the data belonging to the certain weather condition cluster to construct sparse AI model. As mentioned in Section 2.2, construction for sparse modelized AI, it remains only the connections that are the important for regressing power demands, so that network connections between input and hidden layer allow discovery of power demand factors. Afterwards, 45 days from July 30, 2018 is given as unknown data to construct SOM with 2017 data to identify dates that apply to each weather condition cluster, and sparse modelized AI for the certain weather condition clusters is used to predict power demand.

5.1. **Construction of weather condition clusters by SOM.** The attributes of the data used to construct weather condition clusters are precipitation, maximum temperature, average humidity, extreme heat day code, midsummer day code, summer day code, and low temperature code, in units of one day. Each day code is set to 1 if applicable, 0 otherwise, and others are normalized to $N[0,1]$. The data for weather condition is Kitakyushu (Yahata). Some attributes are not observed in Kitakyushu, in which case Fukuoka is used. The number of clusters is 3 because enough days are not belonging to a cluster if that of clusters is more than 3. Figure 3(a) shows the results of Ward's method after SOM training, and 3(b) shows the 3 clustering results.



(a)



(b)

FIGURE 3. Construction of weather condition clusters by SOM: (a) Results of Ward's method, (b) 3 clustering map, C1 to C3 denote clusters 1 to 3

From Figure 3(b), cluster C1 is a sunny midsummer day with high average temperature, cluster C2 is cloudy and cool regardless of precipitation, and cluster C3 is a sunny day but with precipitation. The number of days belonging to cluster C1 is 31 in 2017 and 33 in 2018. Both of the number of data are close to each other and have a sufficient number of training and predict data. For this, prediction and factor identification by sparse modeling in the next section is performed in weather condition cluster C1.

5.2. **Factor identification and prediction of power demand using sparse modeling AI.** Power demand is predicted in 30 minute increments from 0:00. However, prediction for 0:00 through 0:30 cannot be performed because the data belonging to the clusters are not continuous as mentioned in Section 4. Hence, learning for sparse modeling AI must be completed, and there are 47 predictions for a day. In this research, data is added every 30 minutes to compose the learning and prediction data, so that the time length of among data varies. Concretely, the data set for one day consists of data that the number of time length for data 0 is from 0:00 to 0:30, data 1 is from 0:00 to 1:00, data 2 is from 0:00 to 1:30, and the following in the same way. Therefore, the minimum and maximum number of time length of data are 1 and 47, respectively. These are prepared for each day belonging to a cluster and selected at random during learning. The learning attributes for the sparse modeling AI construction of weather condition are 16 dimensions: power demand value, 24 hour, weekday code, Saturday code, Sunday and holiday code, real temperature, humidity, dew point temperature, vapor pressure, 3 hour temperature forecast, daytime high temperature forecast, next day minimum day temperature forecast, next day maximum day temperature forecast, sunshine duration, wind speed, and amount of precipitation.

Sparse modeling is initially pre-learned (PL) using error function without L1 regularization term. Sparsification by L1 regularization (L1) is additionally performed to pre-learned model. Then, sparsification by selective L1 regularization (SL1) is additionally performed for elaborate for sparsification to L1 regularization model. Table 1 indicates parameters of PL, L1 and SL1. In addition, we use Adam for the optimizer. In SL1, weights with absolute values less than 0.1 are regularized. Figure 4 shows the number of connections for each attribute that is not subject to regularization after SL1, i.e., each attribute required for the regression of power demand.

Table 1. Parameters of PL, L1 and SL1

	PL	L1	SL1
Epochs	40,000	120,000	250,000
Learning late	0.001	0.001	0.001
Hidden layer size	200	200	200
β	0.9, 0.999	0.9, 0.999	0.9, 0.999
Regularization strength	–	$\lambda = 5 \times 10^{-6}$	$\lambda = 5 \times 10^{-6}$

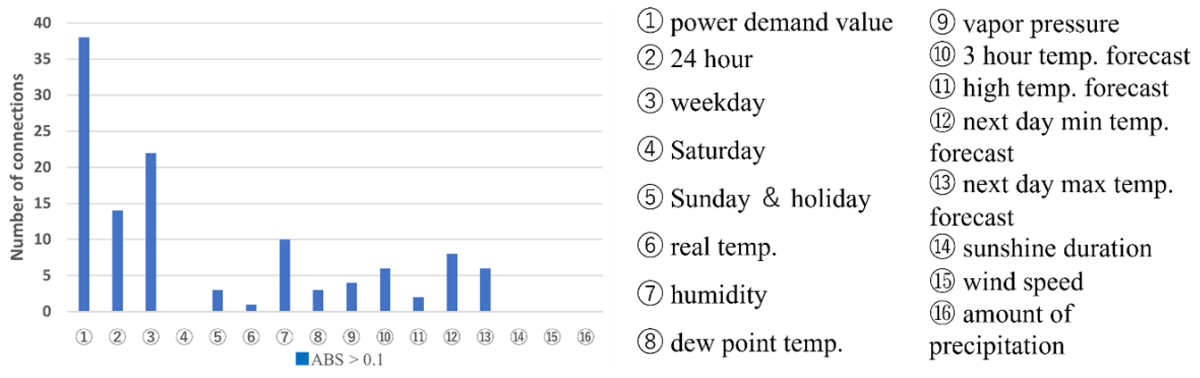


Figure 4. Factor identification of power demand for weather cluster C1

From Figure 4, the power demand value, weekday code, 24 hour, and humidity are important factors with more than 10 connections remaining. A small amount of connections remain, holidays, real air temperature, dew point temperature, vapor pressure, and weather forecast are of some importance. On the other hand, because connections do not remain, Saturday code, sunshine hours, wind speed, and rainfall are not important. Therefore, in a midsummer day with no rainfall and high average temperatures in weather condition cluster C1, humidity is the factors that affect power demand. Generally speaking, higher temperatures increase power demand for air conditioning, but humidity is more important than temperature. Next, sparse modeling AI learning on the 2017 cluster C1 data is used to predict power demand for days belonging to cluster C1 in 2018 and compare it to actual demand to verify generalization performance. Figure 5 shows the most successful prediction, July 15, 2018.

Figure 5 shows that AI predicted that demand would rise around 2:00 (No. 4), but real demand started around 4:30 (No. 9), resulting in a discrepancy of more than 15%. Around 5:00 (No. 10), the prediction is lower than the real demand, deviating by more

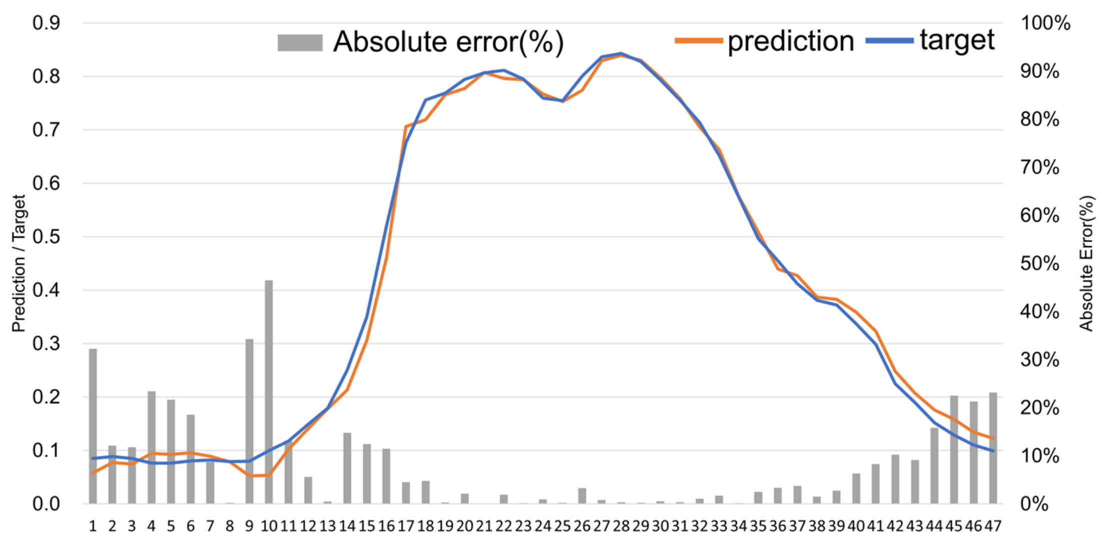


FIGURE 5. Generalization performance test: Prediction of unknown data (Blue is actual values, Orange is predicted value, Gray bars indicate absolute error)

than 35%. On the other hand, after that, the predictions have been made well. From 14:00 (No. 28) to 19:00 (No. 38), the error is within 2%, and in some cases the error is 0%. However, the average daily error is 9%, which does not reach the target of 3%. Based on the above results, it is necessary to adjust the optimal number of clusters and the hyper parameters for learning.

6. Conclusions.

In this research, we propose to construct weather condition clusters by SOM and to predict power demand based on those clusters by sparse modeling AI. From the experimental results, our proposal succeeded in constructing weather clusters by SOM. Moreover, sparse modeling AI showed that humidity affects power demand when midsummer day with no rainfall and high average temperatures. In addition, prediction error by sparse modeling AI is about 9%. Our target in this research is average daily absolute prediction error of 3% or less, so it is necessary to consider the number of weather condition clusters for SOM and optimization of hyper parameters for sparse modeling AI, and those are for further study.

Acknowledgment. This research was supported by power demand data from Kitakyushu Power Company and past weather forecast data from Fukuoka Meteorological Observatory.

REFERENCES

[1] M. Ishikawa, Structural learning with forgetting, *Neural Networks*, vol.9, no.3, pp.509-521, 1996.
[2] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, *arXiv.org*, arXiv: 1711.05101v3, 2019.
[3] R. J. Williams and D. Zipser, Gradient based learning algorithm for recurrent networks, *Back Propagation: Theory Architectures and Applications*, pp.433-486, 1995.
[4] S. Amari, A theory of adaptive pattern classifiers, *IEEE Transactions on Electronic Computers*, vol.EC-16, no.3, pp.299-307, 1967.
[5] K. Osawa, K. Kamei and M. Ishikawa, Efficient cluster mapping for conditions of weather based on combination of self-organizing map and hierarchical clustering, *IEICE Tech. Rep.*, vol.119, no.453, pp.213-218, 2020.