Journal of Signal Processing, Vol.24, No.4, pp.195-198, July 2020

SELECTED PAPER AT NCSP'20

Correction Method for Numerically Forecast Wind Speed Data of MSM-GPV Using CNN

Daichi Horihata[1], Hiroshi Suzuki[1], Takahiro Kitajima[1], Akinobu Kuwahara[1], Takashi Yasuno[1] and Kiyoshi Takigawa[2]

[1] Graduate School of Advanced Technology and Science, Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
E-mail:horihata-d@ee.tokushima-u.ac.jp
{suzuki.hiroshi, yasuno.takashi }@tokushima-u.ac.jp

[2] Shikoku Research Institute Inc.
2109-8 Yashima-Nishimachi, Takamatsu
761-0192, Japan
E-mail: k-takigawa@ssken.co.jp

Abstract

This paper describes a statistical correction model for wind speed data of the Meso-Scale Model Grid Point Value (MSM-GPV), which is one of the numerical weather forecasting systems. In the numerical forecasting system, there are calculation errors caused by both the physical modeling and estimation of initial values. Because numerical forecast data have two-dimensional spatial information, convolution with a convolutional neural network (CNN) is used to grasp and correct the two-dimensional features of errors contained in the forecast data. In the simulations, several MSM-GPV data used for the input data and various correction models are prepared and compared with the results of a fully connected neural network from the viewpoints of the error improvement rate and error distribution.

1. Introduction

Currently, power is mainly generated using fossil fuels, but fossil fuels have limited supply and emit CO_2 during power generation, causing global warming. For these reasons, renewable energy, which does not run out or emit CO_2 during power generation, has attracted attention in recent years. In particular, wind power is being introduced worldwide because it can generate electricity as long as the wind is blowing [1].

However, the output of wind power fluctuates greatly depending on the wind speed. For this reason, power companies adjust the power supply and demand balance using thermal and pumped-storage power generation. In order to operate a power system efficiently and stably, the prediction of wind power output using numerical forecasts is performed [2]. However, numerical forecasts are affected by the modeling errors of atmospheric and terrain models used in simulations. Therefore, in the prediction of wind power output using numerical forecasts, a correction is performed using a statistical method [3].

In this paper, we propose a statistical correction model for the wind speed data of the Meso-Scale Model Grid Point Value (MSM-GPV), which is one of the numerical forecast systems provided by Japan Meteorological Agency. The proposed model is composed of a two-dimensional convolutional neural network (2D-CNN)[4], which grasps the physical relationship between the grid points and the two-dimensional spatial information of errors included in GPV data, and corrects forecast wind speed data. Several models with different configurations are developed and compared on the basis of the error improvement rate and the error distribution of data from the simulation model.

2. Correction Model of MSM-GPV Wind Speed Data

In this paper, we propose a correction model of MSM-GPV by using a CNN and analyze the effects of different sizes of the input area by comparing the correction results. The target points of the correction are four grid points of the MSM-GPV, A, B, C, and D, as shown in Fig. 1, which surround the AMeDAS weather observation station located in Tokushima. Four forecasted data surrounding the prediction point are required to predict the wind speed at a point distant from the grid point in our future task. The target forecast data of MSM-GPV for correction is 6 hours ahead in consideration of the distribution delay time of MSM-GPV and an adjustment capability to valance the supply and demand on the power grid.

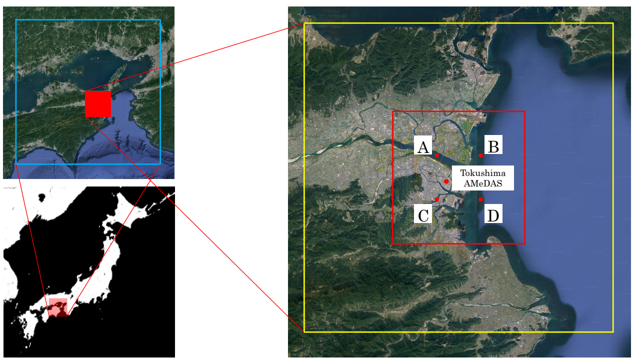


Figure 1: Correction points around Tokushima

All simulations using the correction models in this paper are constructed using Tensorflow [5]. The activation function in each model is ReLU. Weights and biases are adjusted by the back-propagation method with the Adam optimization function. The learning method is mini-batch learning with a batch size of 64. Correction models with 18 different layer configurations and three input sizes are compared to consider an appropriate model. Also, the proposed models are compared with a previously studied fully connected neural network[6] under several conditions.

2.1 Correction period and input data

As the input data, the ground wind speed data in the areas enclosed in squares shown in Fig. 1 are used. MSM-GPV is data on grid points on a 5 km mesh, and the calculation is started every 3 hours. For each calculation start time, the forecast values for 1 hour intervals are calculated up to 39 hours ahead. Wind speed data are given as the east-west and north-south components. For the correction, the initial wind speed data at the calculation start time and the forecast value 6 hours ahead in the MSM-GPV wind speed data are used. By using the two wind speed data for the input, the model can consider time-series information, which is effective for the correction. To compare the effects of the input image size, corrections are conducted by inputting the three types of input data set enclosed by colored lines in Fig. 1 and listed in Table 1.

Table 1: Types of input data set

	Input data	Enclosed color (line)
Type 1	4×4 points	red
Type 2	8×8 points	yellow
Type 3	37×37 points	blue

Types 1, 2, and 3 use data from 16 (4×4), 64 (8×8), and 1369 grid points (37×37), respectively, surrounding the AMeDAS located in Tokushima. In Type 1 and 2, the AMeDAS is at the center of the input data, but in Type 3, the correction target is 100 km from the north end and 100 km from the west end of the data used as shown in Fig. 1. The periods of the input data sets are assigned as shown in Table 2.

Table 2: Periods of input data sets

	Period
Training	April 2015−−March 2016
Validation	April 2016−−March 2017
Target	February 2018

Each model is trained using the training data, and its generalization performance is examined using the validation data. The model with the smallest error for the validation data is used for correction and evaluated with the target data. When using the CNN, the data is input with time, and X and Y components of wind are superimposed in the channel direction. When using a fully connected neural network, the data is reshaped and input as one-dimensional data.

2.2 Correction method with 2D-CNN

The 2D-CNN in this study consists of convolutional, pooling, and fully connected layers as shown in Fig. 2. The CNN extracts spatial features using the convolutional and pooling layers. Multiple data with different time and wind direction components are used as input data, and input in the channel direction. In the 2D-CNN, the output of unit z_{xy}^{ij} at position (x, y) in the jth feature map in the ith layer with the filter of $P_i \times Q_i$ is given by

$$z_{xy}^{ij} = f\left(\sum_{k=0}^{K-1}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} z_{(x+p)(y+q)k}^{(i-1)j} h_{pq}^{kij} + b_{ij}\right) \quad (1)$$

where $f(\cdot)$ is the activation function, K is the number of channels of the previous layer, h is the pixel of the filter, and b is the bias. The convolution process is performed while sliding the filter with a stride of 1. After passing through the convolution–pooling layer, the created feature map is input to the fully connected layer. Here, the pooling method is max pooling and the loss function is the root mean square error (RMSE). The batch size for mini-batch learning is set to 64.

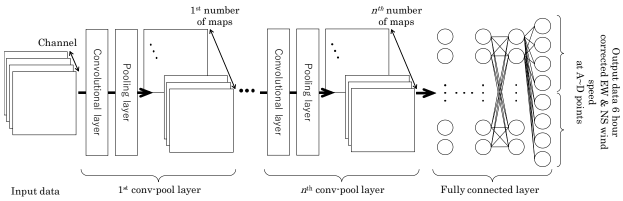


Figure 2: Overall structure of correction model

2.3 Fully connected neural network

In our previous research, the five-layered fully connected neural network shown in Fig. 3 was effective for correction with the Type 1 input [6]. This model is used as a comparison target for correction in this paper.

As input data, two-dimensional and time-series data of grid points are converted into one-dimensional data. Each layer is connected by weights $w_{ij}^{(k)}$ and has biases $b_i^{(k)}$. The output of the unit is calculated from the internal state by the ReLU activation function. The ith output of the kth hidden layer is given by

$$h_i^{(k)} = f\left(\sum_{j=1}^{M^{(k)}} w_{ij}^{(k)} x_j^{(k)} + b_i^{(k)}\right) \quad (2)$$

where M is the number of inputs and x is the input to the kth hidden layer. The output in the hidden layer becomes the input to the next layer. Here, the mean square error (MSE) is used as the loss function.

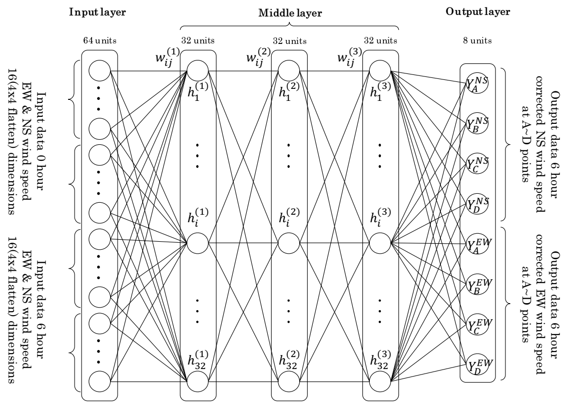


Figure 3: Structure of fully connected neural network

3. Network Configuration of the Correction Model

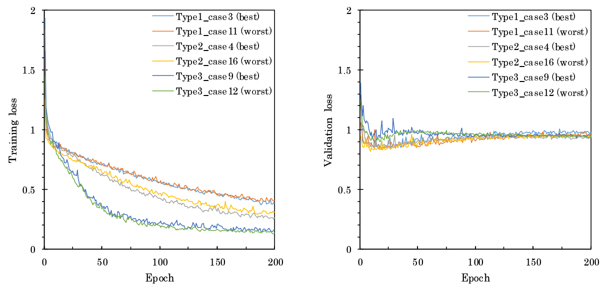
Table 3 shows the parameters of the developed correction models. The models are changed to cases 1 to 18, and their correction accuracies are compared. The correction accuracy of each model is evaluated on the basis of the improvement rate of the RMSE with respect to the uncorrected 6-hour forecast value of MSM-GPV. In addition, each correction model is evaluated on the basis of the distribution of the error.

Table 3: Types of correction model

	Number of maps				Fully connected layer			
case	1st	2nd	3rd	4th	1st	2nd	3rd	4th
1	32	64	128	-	128	8	-	-
2	16	32	64	-	128	8	-	-
3	32	64	128	-	128	64	32	8
4	16	32	64	-	64	32	16	8
5	32	32	32	-	128	8	-	-
6	128	128	128	-	128	8	-	-
7	16	32	64	-	128	64	32	8
8	32	64	128	256	256	128	32	8
9	16	32	64	-	2048	1024	512	8
10	128	128	128	128	256	128	32	8
11	32	128	-	-	128	8	-	-
12	32	64	-	-	128	8	-	-
13	16	32	-	-	128	8	-	-
14	64	128	-	-	256	8	-	-
15	64	-	-	-	256	8	-	-
16	128	-	-	-	512	8	-	-
17	16	32	64	-	256	228	32	8
18	32	64	128	-	64	32	16	8

3.1 Loss transition in training process

Figure 4 shows the transition of losses for the training and validation data in the training process with the best and worst improvement rates for each input data type. The loss for the training data decreases during training in all models. In contrast, the loss for the validation data start to increase from the early stage of training as shown in Fig.4(b) . In the case of 200 epochs, it can be seen that overtraining occurs in all cases. In the case of Type3, the loss for the training data fluctuates, the loss at 200 epochs is the smallest, and the output fits the training data very well as shown in Fig. 4(a). Therefore, the target data is corrected by each model with the smallest error for the validation data.



(a) Training data (2015)

(b) Validation data (2016)

Figure 4: Transition of loss in training process

3.2 Correction accuracy and improvement rate

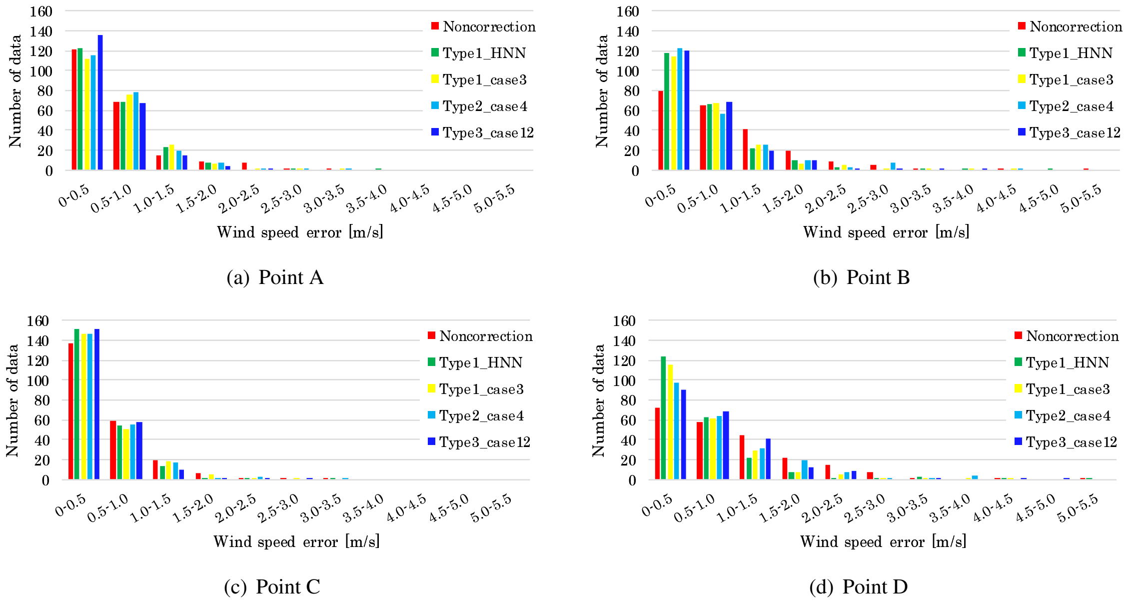
The error with respect to the validation data of each model is minimized at a certain number of epochs. In Table 4, red numbers indicate the best model and blue numbers indicate the worst model for each input data type. The models at these epoch numbers are used for correction.

Table 4: Epoch with minimum loss

case	1	2	3	4	5	6	7	8	9
Type1	14	24	**21**	24	24	19	16	10	10
Type2	10	15	23	**19**	19	6	14	13	4
Type3	16	23	24	21	24	17	14	193	**15**
case	10	11	12	13	14	15	16	17	18
Type1	22	**20**	21	31	20	22	32	32	27
Type2	11	16	12	20	11	13	**9**	22	12
Type3	143	15	**13**	16	16	16	7	20	21

As a result of the correction, using the network for all input types and models, the average RMSE of the four correcting points was improved from that of the uncorrected data. Table 5 shows the maximum and minimum improvement rates for each input type. The improvement rates at points B and D were largest because these points are grid points on the sea where the effect of the terrain is small. Therefore, the tendency of the error is easy to grasp. On the other hand, points A and C are grid points on land where the effect of the terrain is large; therefore, the tendency of the error is considered to be complex. Regardless of the input type, the average improvement rate of the four points was positive; therefore, the correction was successfully made in terms of RMSE. The correction result with a 37×37 data input to the network model of case 12 was the best.

Figures 5(a) to 5(d) show the distribution of the error on each model before and after correction. The variance of the error decreases after correction, and it can be seen that the correction is performed in all models. In particular, a significant improvement was seen at points B and D. In addition, no significant difference was observed between the CNN and the HNN when the correction was performed using Type 1 data.



(a) Point A

(b) Point B

(c) Point C

(d) Point D

Figure 5: Error distribution for each point and model

Table 5: Improvement of RMSE

Input	Type 1				
Point	A	B	C	D	Average
case3	8.07	28.59	6.67	30.31	21.43
case11	14.93	23.20	-0.23	23.29	14.41
5–layer HNN	9.69	27.97	10.5	32.60	22.96
Input	Type 2				
Point	A	B	C	D	Average
case4	12.11	27.25	5.36	22.24	18.96
case16	16.67	21.20	-20.16	2.51	6.95
Input	Type 3				
Point	A	B	C	D	Average
case12	25.63	33.36	13.83	23.71	25.41
case9	21.44	11.58	-21.24	-4.83	2.69

Compared with the CNN for each input, at a points A and C on land, a larger input size was a better improvement rate. On the other hand, for points B and D on the sea, a smaller input size has a better improvement rate.

4. Conclusions

We corrected the 6-hour forecast values of MSM-GPV at four grid points around Tokushima City by using a CNN and confirmed the improvement rate for the MSM-GPV forecast. Forecasting accuracy can be improved by correction using a wide range of input data. Comparing the error distributions obtained before and after the correction, there was a small difference between the CNN and the HNN.

Future tasks should consider the relationship between the prediction time and the size of input data, and evaluate the effect of MSM-GPV correction on wind speed prediction.

Acknowledgment

MSM-GPV data are provided by the Research Institute for Sustainable Humanosphere (RISH), Kyoto University, Japan.

References

[1] Agency for Natural Resources and Energy: FY2018 Annual Report on Energy (Energy White Paper 2019), pp208-209, 2019.

[2] K. Takigawa: Development of wind power prediction system, J. IEIE Jpn., Vol.29, No. 11, pp.911-913, 2009.

[3] S. Kadokura et al.: Development of the prediction system for wind power generation over a wind farm, CRIEPI Research Report, No.V08065, 2009.

[4] H. Nakayama: Image feature extraction and transfer learning using deep convolutional neural networks, IEICE Technical Report, Vol. 115, No.146, pp. 55-59, 2015.

[5] M. Abadi et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv: 1603.04467, 2016.

[6] D. Horihata et al.: Correction of numerical forecast wind speed data using deep neural network, Record of SJCIEE 2019, No.3-1, p.46, 2019.