

PAPER

Toward Predictive Modeling of Solar Power Generation for Multiple Power Plants

Kundjanasith THONGLEK^{†a)}, *Nonmember*, Kohei ICHIKAWA[†], *Member*, Keichi TAKAHASHI^{††},
Chawanat NAKASAN^{†††}, Kazufumi YUASA^{††††}, Tadatashi BABASAKI^{††††},
and Hajimu IIDA[†], *Nonmembers*

SUMMARY Solar power is the most widely used renewable energy source, which reduces pollution consequences from using conventional fossil fuels. However, supplying stable power from solar power generation remains challenging because it is difficult to forecast power generation. Accurate prediction of solar power generation would allow effective control of the amount of electricity stored in batteries, leading in a stable supply of electricity. Although the number of power plants is increasing, building a solar power prediction model for a newly constructed power plant usually requires collecting a new training dataset for the new power plant, which takes time to collect a sufficient amount of data. This paper aims to develop a highly accurate solar power prediction model for multiple power plants available for both new and existing power plants. The proposed method trains the model on existing multiple power plants to generate a general prediction model, and then uses it for a new power plant while waiting for the data to be collected. In addition, the proposed method tunes the general prediction model on the newly collected dataset and improves the accuracy for the new power plant. We evaluated the proposed method on 55 power plants in Japan with the dataset collected for two and a half years. As a result, the pre-trained models of our proposed method significantly reduces the average RMSE of the baseline method by 73.19%. This indicates that the model can generalize over multiple power plants, and training using datasets from other power plants is effective in reducing the RMSE. Fine-tuning the pre-trained model further reduces the RMSE by 8.12%.

key words: *fine-tuning, long short-term memory, solar power generation, time-series forecasting*

1. Introduction

Solar power is a prominent renewable energy source that transforms sunlight into electricity [1], emitting less pollution than traditional fossil fuels [2]. However, solar power generation faces challenges in energy efficiency, long-term adoption, cost reduction, and efficient battery utilization. Producing stable electricity using solar cells requires batteries to store and combine the electricity from the solar cells and the traditional fossil fuels, since the amount of solar power generated depends on the amount of solar radiation and does not always satisfy the electricity demand. There are several existing approaches to improve the power adjust-

ment of the electricity stored in the batteries for producing a stable power supply, such as increasing the capacity of the batteries [3], designing battery controllers [4], and building prediction models for the solar power generation [5]. However, improving battery capacity requires a large investment compared to developing prediction models. Therefore, there has been a considerable emphasis on the development of highly accurate prediction models. However, an acceptable prediction accuracy is not exactly defined, because it depends on the target facilities and the investment. Nevertheless, it is possible to evaluate how much the cost of battery systems can be saved from the improvement of the prediction accuracy.

Due to the limited amount of available datasets and the nature of large seasonal variations of the solar power generation, it remains difficult to develop models that accurately predict solar power generation [6]. Generally, training an accurate prediction model requires a large training dataset [7]. Scientists thus have to spend time in collecting and preprocessing datasets. In addition, predicting the solar power generation of multiple solar power plants is more challenging than a single power plant since each power plant has a different amount of power generation capacity.

In our previous work, we have developed a single model for predicting solar power generation [8]. This article extends our previous work from building a prediction model for a single power plant to multiple power plants. Previously, a prediction model was trained and tested on the dataset of a single power plant. However, this model is not directly available for other power plants. Therefore, we cannot immediately build prediction models for power plants that have recently started operation as collecting available training datasets is still insufficient.

In this article, we propose a method to train a general model for predicting the power generation of multiple solar power plants. The proposed model also supports predicting the solar power generation for newly constructed power plants of which there is little training data available. To achieve high model accuracy with small training dataset, the general prediction model trained on multiple power plant datasets is fine-tuned using the limited data from newly constructed power plants.

The remainder of this article is structured as follows. Section 2 reviews related literature on existing techniques for predicting solar power generation using machine learning. Section 3 explains the dataset used in this study. Section 4 describes our proposed methodology to develop a prediction

Manuscript received August 21, 2022.

Manuscript revised November 22, 2022.

Manuscript publicized December 22, 2022.

[†]The authors are with Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.

^{††}The author is with Tohoku University, Sendai-shi, 980-8578 Japan.

^{†††}The author is with Kasetsart University, Bangkok, Thailand.

^{††††}The authors are with NTT FACILITIES, INC., Tokyo, 108-0023 Japan.

a) E-mail: thonglek.kundjanasith.ti7@is.naist.jp

DOI: 10.1587/transcom.2022EBT0003

model of solar power generation for multiple power plants. Section 5 evaluates the proposed methodology. Section 6 concludes this article and discusses future work.

2. Related Work

This section gives a brief overview of existing machine learning approaches for predicting solar power generation, and also fine-tuning prediction models for enhancing the model accuracy.

2.1 Prediction of Solar Power Generation

Predicting solar power generation is an efficient approach to improve stability of solar power generation with less investment. Machine learning is commonly adopted to predict solar power generation [9].

Sharma et al. applied simple machine learning models including linear regression and kernel Support Vector Machine (SVM) to estimate solar power generation from weather forecasts [10]. The result indicated that an SVM-based prediction model outperformed linear regression by 27% in terms of Root Mean Square Error (RMSE). However, feature engineering is still required when employing these machine learning techniques. Feature engineering refers to the process of leveraging domain experts into feature extractor development [11]. By reducing the complexity of the input, the feature extractors make the patterns of input more visible to machine learning algorithms.

Deep learning has been introduced to eliminate the feature engineering process in building a prediction model, and also to increase the model accuracy. To forecast solar power generation, Rodriguez et al. developed a model based on artificial neural networks [12]. Simple artificial neural networks, however, are not ideal for predicting solar power generation since it is a type of time-series forecasting.

Long Short-Term Memory (LSTM) is a form of recurrent neural network (RNN) that can store historical data over a long period of time for time-series prediction. LSTM was developed to deal with the vanishing gradient problem that is often encountered when training traditional RNNs. Figure 1 illustrates the architecture of an LSTM block. An LSTM block has a memory cell, input gate, output gate, and a forget gate in addition to the hidden state in conventional RNNs [13]. The memory cell is the portion of the hidden unit that is only modified by addition or subtraction and scaling, and thus tends to preserve information for a relatively long time [14]. Because of its architecture, LSTMs can learn contexts from long-term time-series data and predict patterns in the future [15]. Zhang et al. presented an LSTM model to predict solar power generation [16]. Their results indicated that LSTM outperforms RNN in predicting solar power generation.

Accordingly, we develop a neural network model based on LSTM since our dataset consists of hourly solar power generation and weather forecast data as time-series data for two and a half years. Our dataset is not large enough and

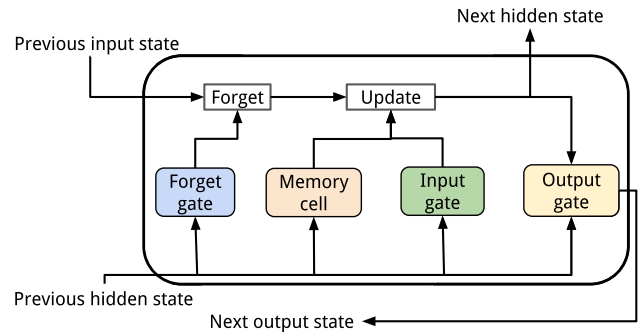


Fig. 1 A Long Short-Term Memory block.

has a lot of seasonal fluctuation, making it difficult to simply employ LSTM. Stratified k -fold cross-validation is therefore used throughout the training process to avoid the problems of overfitting and seasonal fluctuation [17].

2.2 Fine-Tuning of Prediction Models

Fine-tuning improves the accuracy of a pre-trained model by making small adjustments to the weights of the pre-trained model on a specific dataset [18]. While fine-tuning improves the accuracy of prediction models, it can only be applied when the datasets used for pre-training and fine-tuning processes are similar [19]. When fine-tuning a model, each layer in the model is treated as either *trainable* or *frozen*. The parameters in the trainable layers are updated during fine-tuning, whereas the parameters in the frozen layers are fixed. Tuning hyperparameters such as the learning rate and the design of the trainable layers is required for effective fine-tuning [20]. The learning rate determines how much the model changes in response to the loss each time the model weights are updated [21].

Wu et al. highlighted that choosing the optimal learning rate is challenging since using a too small value results in a long training time, while using a too large value results in skipping some significant weight adjustments that improves the model accuracy [22]. Thus, the appropriate learning rate depends on the model architecture and the training dataset. During the fine-tuning process, reducing the number of trainable layers with freezing some layers can shorten the training time since the number of parameters to be updated is reduced [23]. Xiao et al. demonstrated that freezing some layers during training process might improve the model accuracy if the less updated layers are frozen [24].

There are several existing works that train models using general datasets and then fine-tune the pre-trained models for specific datasets. For example, in the case of time-series prediction, Oyeleye et al. developed a prediction model for heart rates to prevent the risk of cardiovascular disease when a high heart rate was predicted [25]. They built a pre-trained model to predict heart rates, and then fine-tuned it for each person. Dhar et al. investigated the impact of adjusting learning rate on the model accuracy for predicting stock prices for each stock index [26]. A naive method to find the

optimal learning rate is to try with a few different learning rates and determine which one provides the highest accuracy. Konar et al. suggested starting with a large value, such as 0.1, and then trying with exponentially smaller values when adjusting the learning rate [27]. In this paper, we applied LSTM to build a pre-trained model of solar power generation on all available existing power plants. The pre-trained model is then fine-tuned for newly constructed plants to enhance its model accuracy. Furthermore, we studied on finding the suitable trainable layers and the learning rate during the fine-tuning process.

3. Dataset

This section describes the dataset used in this study. We built our dataset from two data sources: (1) weather forecasts provided by the Japan Meteorological Agency (JMA) [28], and (2) historical solar power generation at power plants maintained by NTT Facilities. We use the weather forecasts computed using the Meso-Scale Model (MSM), which provides 51- and 39-hour weather forecasts every 3 hours. NTT Facilities maintains 71 power plants across Japan as shown in Fig. 2[†]. Out of the 71 power plants, we excluded 10 power plants located in the *Kyushu* island. This is because the *Kyushu* island is located in the southern part of Japan where sunlight is intense, and power generation at power plants is sometimes capped due to over-generation.

Although each power plant has collected data on solar power generation over different periods, we have chosen to use the data collected from January 1, 2018, 00:00 AM to June 11, 2020, 00:00 AM. This is because solar radiation forecasts by JMA are only available from January 1, 2018, 00:00 AM, and most power plants had data on power generation between January 1, 2018, 00:00 AM and June 11, 2020, 00:00 AM. However, only 6 power plants had collected very little data during this period, so we excluded these 6 plants and use data from a total of 55 plants.

We acquire data from both sources every hour. Thus, 24 records are produced for each day. The dataset of each power plant has 24,960 records with 13 features is shown in Table 1. The first column is the timestamp, the second to twelfth columns are the weather forecast data, and the last column is the solar power generation. Due to errors in data collection, some rows contain values that exceed the capacity of the power plant, or that are less than zero. These rows are removed during the preprocessing process.

In this study, we aim to build a machine learning model to predict the amount of solar power generation for multiple power plants. Figure 3 shows the variability of hourly solar power generation at each power plant. It suggests that solar power generation is strongly skewed toward 0.0 kWh, because solar power cannot be generated if the solar radiation falls below a particular threshold. It is indeed difficult to create an accurate prediction model with such an imbalanced dataset. Furthermore, since each power plant has a different



Fig. 2 Location of NTT facilities' solar power plants in Japan.

Table 1 List of features in the dataset.

#	Feature	Data format
1	Time	YYYY/MM/DD hh:mm:ss
2	Atmospheric pressure	Pa
3	Wind speed (E-W)	$m s^{-1}$
4	Wind speed (N-S)	$m s^{-1}$
5	Temperature	K
6	Humidity	%
7	Low cloudage	%
8	Middle cloudage	%
9	High cloudage	%
10	Total cloudage	%
11	Precipitation	$kg m^{-2}$
12	Solar radiation	$W m^{-2}$
13	Solar power generation	kWh

generation capacity, training a prediction model for multiple power plants is challenging.

Figure 4 plots the average daily solar power generation in each month. It depicts how solar power generation fluctuates depending on the season throughout the year. The considerable seasonal variation of the dataset also makes developing a highly accurate prediction model difficult.

4. Methodology

This section explains the proposed methodology to build an accurate prediction model of solar power generation for multiple power plants. We first split the dataset of each power plant into training and testing datasets using stratified *k*-fold cross-validation. Next, we develop a model to predict solar power generation. The model is trained using the power

[†]<https://goo.gl/maps/1UkygW7yCpgvzm7QA>

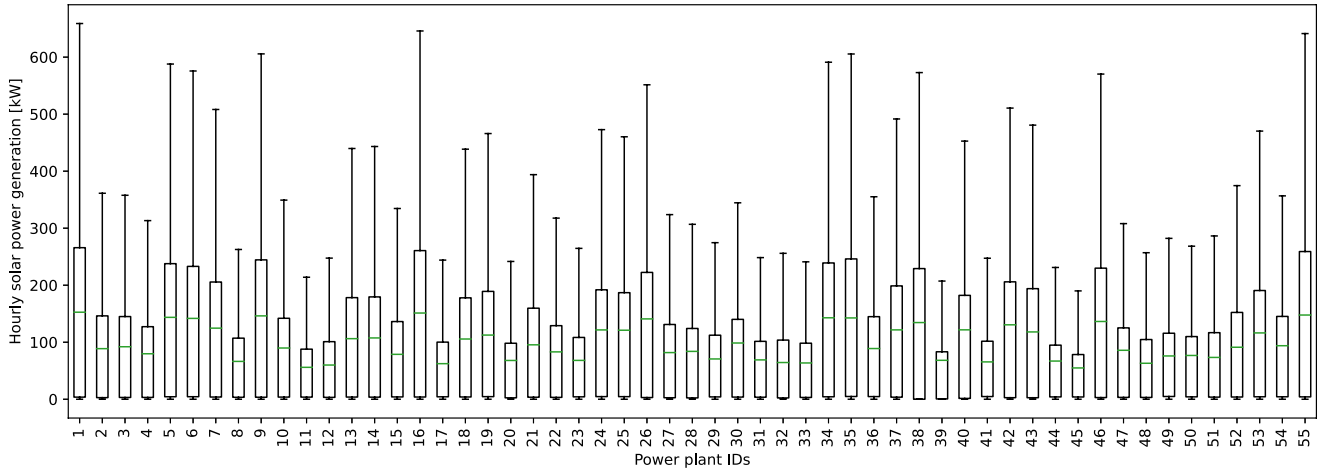


Fig. 3 Variability of hourly solar power generation at each power plant.

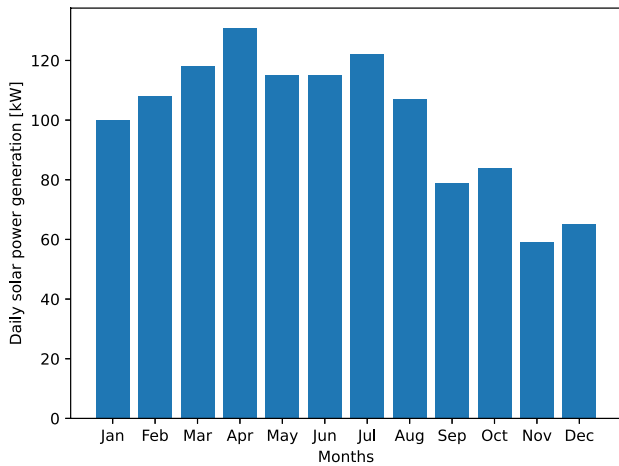


Fig. 4 Average daily solar power generation in each month.

generation data of all power plants except for a target power plant which is assumed as a newly deployed power plant. Subsequently, the model is fine-tuned for the target power plant used for evaluation.

4.1 Preparation of Training and Testing Datasets

Our model predicts the next 24 hours of solar power generation using (1) the last 48 hours of solar power generation, and (2) the weather forecasts for the last 24 hours and next 24 hours. Hence, the length of input and output sequences are 48 and 24, respectively.

The large imbalances and seasonal changes in the dataset could lead to learning problem including overfitting and underfitting problems. Hold-out is a common splitting method in which a dataset is split into two portions, one for training and the other for testing. Basic splitting approaches such as hold-out is not suitable because the training or testing dataset might have large imbalance and seasonal changes. Such a simple split causes an overfitting problem since the training dataset may only include data for a single season.

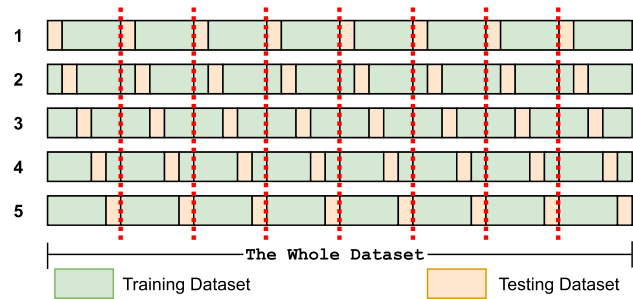


Fig. 5 Stratified 5-fold cross-validation.

The k -fold cross-validation is a cross-validation approach frequently used to enhance the generalization performance of prediction models for imbalanced datasets. Simple k -fold cross-validation is still insufficient for our dataset due to the small size and seasonal variation of the dataset. We may use k -fold cross-validation with a large k to enhance the generalization performance our model, but this would considerably increase the training time. As a consequence, we use a multi-level cross-validation approach called stratified k -fold cross-validation.

In stratified k -fold cross-validation, the dataset is split into equal-sized chunks, and each chunk is subject to k -fold cross-validation. To put it another way, each chunk is divided into k folds, with one k fold serving as the testing dataset and the other $k - 1$ serving as the training dataset. This procedure is performed k times more until each fold is utilized to test the model. Afterwards, the error is averaged over all k iterations to calculate the final accuracy of prediction model. The generalization performance of the model will increase compared to k -fold cross-validation because the training and testing datasets are dispersed over the whole dataset. Figure 5 illustrates an example of 5-fold stratified cross-validation.

Since the generation capacity of each power plant is different, we normalize the dataset of each power plant into the same scale for training and testing a prediction model. Normalizing the dataset into the same scale for the different

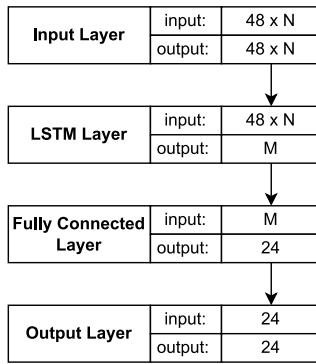


Fig. 6 Architecture of the proposed LSTM-based prediction model.

power plants to prevent distorting differences in the ranges of values. In this work, all features in the dataset of each power plant are normalized within the range between zero and one.

4.2 Pre-Training

The proposed prediction model is based on LSTM because LSTM is able to hold long-term context. This makes LSTM suitable for predicting time-series data like solar power generation. Since our preliminary experiment indicated that using multiple LSTM layers does not improve the prediction accuracy, we build a neural network model with a single LSTM layer.

Our prediction model is illustrated in Fig. 6. The model consists of an LSTM layer and a fully connected (FCN) layer. The LSTM layer has M hidden units. The size of the input layer is $48 \times N$ where N is the number of features and 48 is the length of the input sequence. Additionally, the size of the output layer is 24 since the proposed model predicts the next 24 time steps. In this work, we set the number of hidden units M to 100, and the number of features N to 100. The output layer is a FCN layer with ReLU as the activation function. ReLU is used since the predicted value (i.e., power generation) should always be positive. The model is pre-trained over the training dataset of every power plant except the training dataset of target power plant.

4.3 Fine-Tuning

After the model is pre-trained, it is fine-tuned for the target power plant. Since the proposed prediction model has two layers (i.e., an LSTM layer and a FCN layer), there are three options in choosing which layers to freeze: (1) freezing only the LSTM layer, (2) freezing only the FCN layer, and (3) not freezing any layer.

5. Evaluation

This section evaluates the proposed method to build a prediction model of solar power generation for multiple power plants. We measured the prediction accuracy of pre-trained

Table 2 Hardware specifications.

Hardware	Specification
CPU	Intel Xeon E5-2650 v2 (2.20 GHz, 12 cores) $\times 2$
Main Memory	256 GB
GPU	NVIDIA Tesla P100
GPU Memory	16 GB

and fine-tuned models with stratified k -fold cross-validation to find the learning configuration that achieves the best accuracy. We then compared the accuracy of the proposed method against a baseline method.

5.1 Experimental Environment

Table 2 shows the specifications of the hardware we used in our evaluation. Keras 2.4.3 was used to build the prediction model. The MinMaxScaler class in scikit-learn 0.24 was used to normalize the input data.

We used stratified 5-fold cross-validation and set the training batch size to 32. The numbers of training and testing samples were 19,968 and 4,992 for each power plant, respectively. Nesterov-accelerated Adaptive Moment Estimation (Nadam) was used as the optimizer for training, and the training was stopped when the validation loss did not decrease for five epochs. The accuracy of the prediction model was evaluated using the Root Mean Square Error (RMSE). The Normalized Root Mean Square Root Error (NRMSE) was used to evaluate the prediction model over the testing datasets of every power plant since each power plant has a different generation capacity. RMSE was used to compare different methods. On the other hand, NRMSE was used to compare different training configurations for the same method.

5.2 Pre-Training Results

In this evaluation, we find the suitable learning rate for pre-training considering both accuracy and training time. The model was trained on the training datasets of all power plants except for the target power plant, and then validated on the testing dataset of the target power plant. The learning rate was varied from 10^{-1} to 10^{-8} .

Figure 7 presents the NRMSE of the pre-trained models with different learning rates. Here, we compute the NRMSE for each power plant and show the variability using a box plot. The NRMSE was high and exhibited a large variance when the the learning rate was larger than 10^{-2} or smaller than 10^{-5} . Pre-training with the learning rate of 10^{-3} and 10^{-4} produced very low NRMSE.

Figure 8 shows the runtime for pre-training with different learning rates. Evidently, training with a large learning rate converged faster than with a small learning rate. The training time increased with the learning rate when the learning rate was between 10^{-1} to 10^{-4} , but remained almost constant when the learning rate was smaller than 10^{-4} since the training is not able to be fully converged when the learning

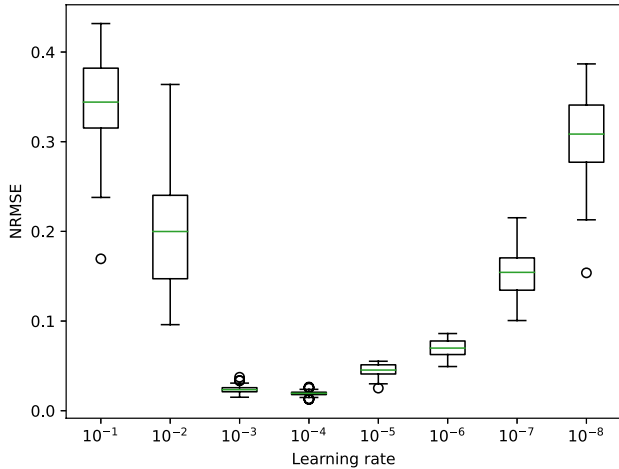


Fig. 7 NRMSE of pre-trained models (varying learning rate).

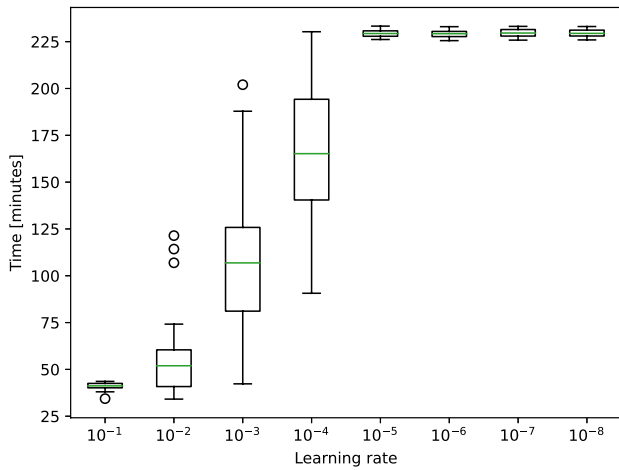


Fig. 8 Runtime for pre-training (varying learning rate).

rate is too small. Therefore, we chose 10^{-4} as the learning rate for pre-training because it produces the lowest NRMSE.

5.3 Fine-Tuning Results

In this evaluation, we find the suitable learning rate and layers to train for fine-tuning. In addition, we quantify the impact of the size of the fine-tuning dataset on the fine-tuned model accuracy. The fine-tuning dataset of the target was split into 12 months where each month is considered 30 days.

We first investigate which layers should be fine-tuned to produce the lowest NRMSE. Here, a pre-trained model was fine-tuned with the learning rate of 10^{-3} . Figure 9 shows the average NRMSE of the fine-tuned models with different set of fine-tuned layers. Although the figure indicates that the choice of the fine-tuned layers did not have a significant impact on the NRMSE, we chose to retrain both the LSTM and FCN layers because it produced the lowest NRMSE when the size of fine-tuning dataset was 12 months. Since solar power generation shows large seasonal variations as described in Sect. 3, the fine-tuning dataset should cover all

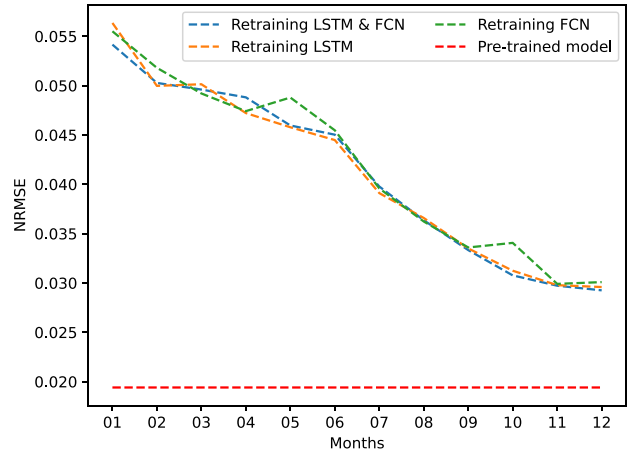


Fig. 9 NRMSE of fine-tuned models (varying fine-tuning dataset size).

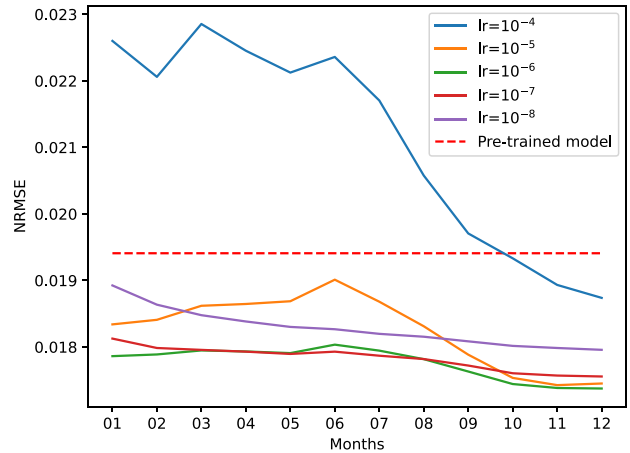


Fig. 10 NRMSE of fine-tuned models (varying learning rate and fine-tuning dataset size).

seasons throughout a year to capture the seasonal variations. We thus focus on the performance when using 12 months of data for fine-tuning.

Nevertheless, the model fine-tuned with a learning rate of 10^{-3} produced higher NRMSE than the original pre-trained model. Thus, we varied the learning rate from 10^{-4} to 10^{-8} to find the suitable learning rate for fine-tuning. The result is shown in Fig. 10. The result indicates that the model accuracy increased with the fine-tuning dataset size. When the learning rate is less than or equal to 10^{-5} , the NRMSE after fine-tuning was consistently lower than the pre-trained model for all dataset sizes.

In addition, we measured the training time for fine-tuning using 12 months of training data. The result is shown in Fig. 11. The training time for fine-tuning does not significantly change depending on the learning rate when the learning rate is in a range between 10^{-1} and 10^{-6} . We therefore chose 10^{-6} as the learning rate for fine-tuning because it produced the lowest NRMSE.

5.4 Comparison to Baseline

We compared our proposed method with a baseline method, where the prediction model is trained only on the dataset of the target power plant. We used 12 months of training data for fine-tuning, and used the hyperparameters chosen in Sects. 5.2 and 5.3 for pre-training and fine-tuning. That is, we set the learning rate to 10^{-4} and 10^{-6} for pre-training and fine-tuning, respectively, and fine-tune both the LSTM and FCN layers. The proposed and baseline methods were

evaluated on the testing dataset of the target power plant. The RMSE was used to compare the prediction errors of each method.

The result is shown in Fig. 12. The pre-trained models of our proposed method significantly reduced the average RMSE of the baseline method by 73.19%. This indicates that the model can generalize over multiple power plants, and that training with datasets from multiple power plants is effective in reducing the RMSE. Fine-tuning the pre-trained models further reduced the RMSE by 8.12%. Moreover, the variance of RMSEs of the proposed method was smaller

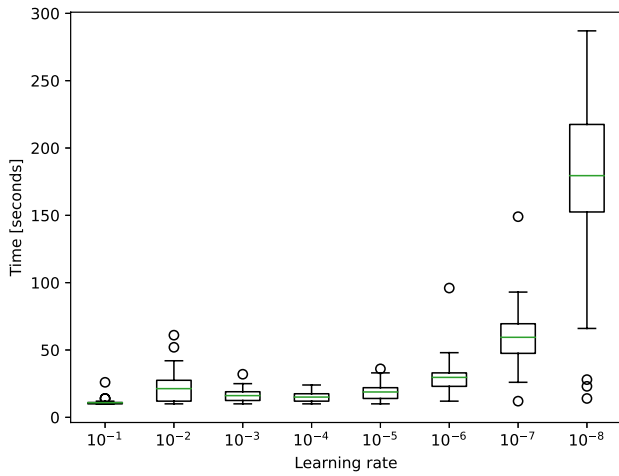


Fig. 11 Runtime for fine-tuning (varying learning rate).

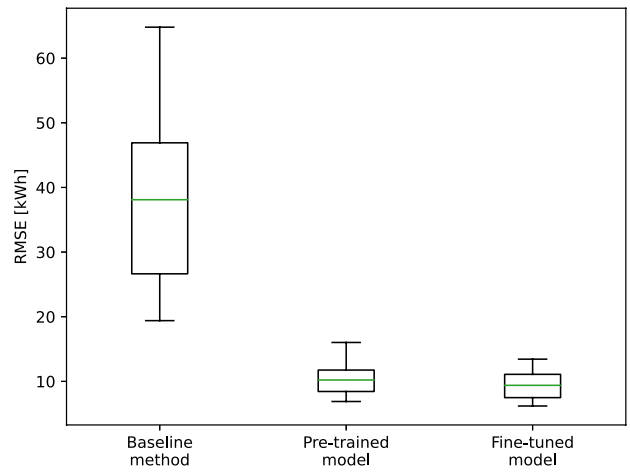
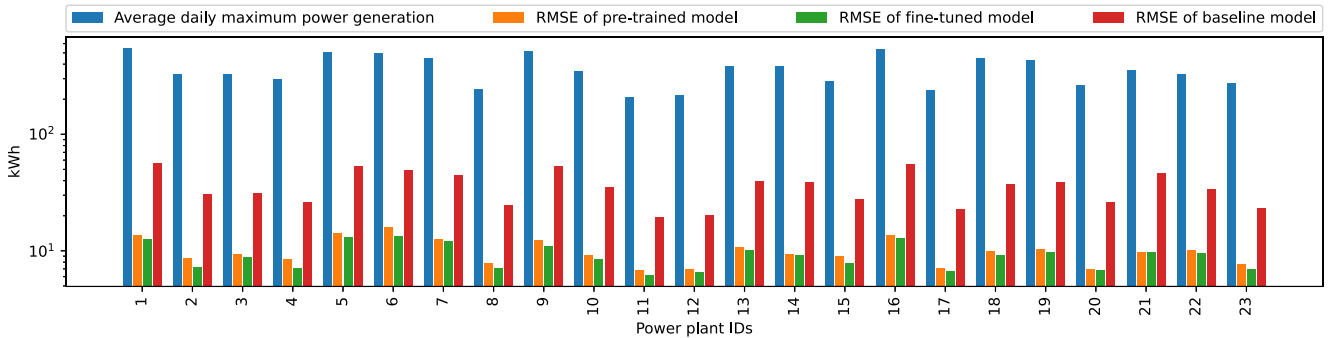
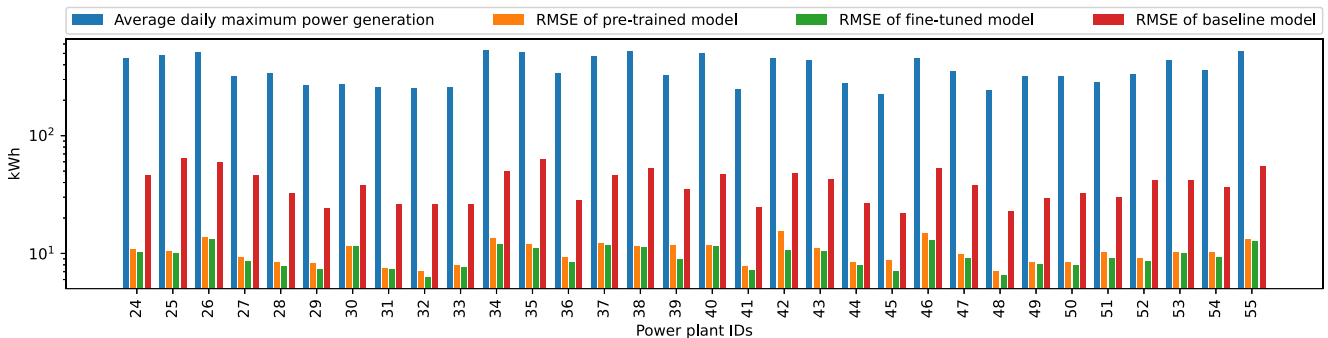


Fig. 12 RMSE of the proposed and baseline methods.



(a) Power plants #1 to #23



(b) Power plants #24 to #55

Fig. 13 RMSE of the proposed and baseline methods.

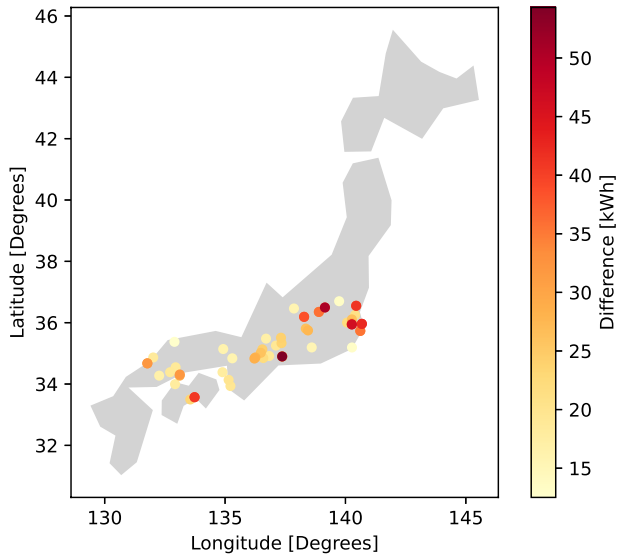


Fig. 14 Differences in RMSE of baseline and pre-trained models.

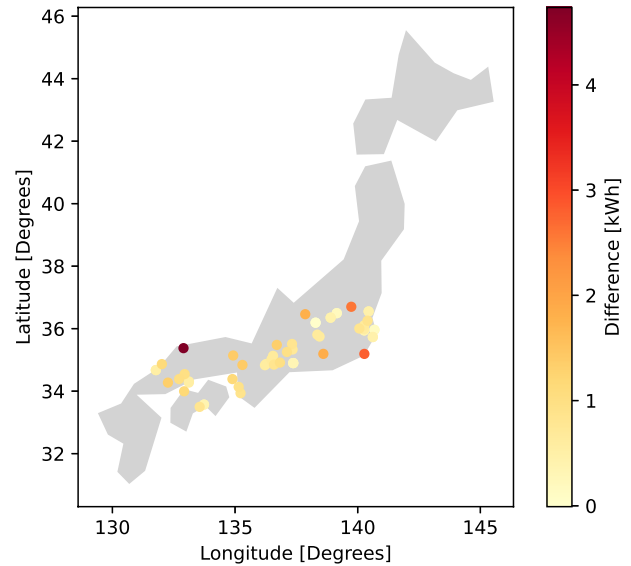


Fig. 15 Differences in RMSE of pre-trained and fine-tuned models

than that of the baseline method, indicating that our method generated more robust models than the baseline method.

To examine in more detail which power plants benefit from the proposed method, we compared the RMSEs of the proposed and baseline methods for individual power plants. Figure 13 presents the average daily maximum power generation and RMSEs of each method. The average daily maximum power generation is used to represent the scale of the power generation for each power plant. The RMSEs of the baseline models were essentially proportional to the scale of the power plants. As already shown in Fig. 12, the accuracy of the pre-trained and fine-tuned models was significantly improved over the baseline models. However, the improvement rates were not the same across all power plants. In some power plants, the pre-trained model already provided high accuracy, and the accuracy did not improve after file-tuning. On the other hand, in some other power plants, the fine-tuned models showed significant improvements over the pre-trained models. On average, the proposed method saves 28.7 kWh for every power plant. According to a report [29], the cost of a utility-scale Li-ion energy storage system ranges from \$379/kWh to \$907/kWh as of Q1 2021, depending on the duration of energy discharge. The proposed method can therefore reduce the cost of battery systems by approximately \$10,877 to \$26,031.

We further investigate the differences in accuracy improvement between power plants. Figure 14 plots the difference in RMSE between the baseline and pre-trained models for each power plant along with the geographical locations of the power plants. The plants plotted as darker-colored points indicate that the improvements by the pre-trained models are greater than the plants plotted as lighter-colored points. Similarly, Fig. 15 plots the difference in RMSE between the pre-trained and fine-tuned models for each power plant. The pre-trained model provided high accuracy for power plants where many other power plants are located

nearby. Since the pre-trained model is a generalized model trained on the datasets of all power plants, the pre-trained model provided high accuracy for groups of plants sharing similar characteristics. Therefore, it is expected to perform better for power plants deployed in the same geographical region. The fine-tuned models showed large improvements over the pre-trained models for power plants where there are no other power plants nearby. Since the generalized model does not capture the characteristics of those isolated power plants, fine-tuning using the dataset of each plant seems to improve the accuracy.

6. Conclusion

In this article, we studied how to build a model to predict future solar power generation using past solar power generation and weather forecasts for multiple power plants. We designed a neural network model based on LSTM to predict solar power generation. In addition, we fine-tune the pre-trained model on the new training dataset to improve accuracy. As a result, the pre-trained models of our proposed method significantly reduced the average RMSE of the baseline method by 73.19%. This indicates that the model can generalize over multiple power plants, and training using datasets from other power plants is effective in reducing the RMSE. Fine-tuning the pre-trained model further reduced the RMSE by 8.12%.

One direction for future work is to investigate other time series forecasting techniques to further improve the prediction accuracy. To validate the generality of the proposed prediction method, solar power generation data and weather forecasts from other sources should be used. Also, clustering the power plants based on some features and then fine-tuning the pre-trained model using data from plants within the same cluster might increase the model accuracy.

References

- [1] S. Jose and R.L. Itagi, "Smart solar power plant," Proc. IEEE International Conference on Communications and Signal Processing (ICCSP), pp.0850–0854, 2015.
- [2] J. Vasita, Q. Shakhya, and J. Modi, "Feasibility study and performance evaluation of a grid-connected rooftop solar pv system," Proc. IEEE International Conference on Information, Communication, Instrumentation and Control (ICICIC), pp.1–7, 2017.
- [3] A. Mills and J. Zambreno, "Towards scalable monitoring and maintenance of rechargeable batteries," IEEE International Conference on Electro/Information Technology, pp.624–629, 2014.
- [4] Z. Hradílek, P. Moldřík, and R. Chválek, "Solar energy storage using hydrogen technology," Proc. IEEE International Conference on Environment and Electrical Engineering (EEEIC), pp.110–113, 2010.
- [5] V. Prema and K.U. Rao, "Development of statistical time series models for solar power prediction," Renewable Energy, vol.83, pp.100–109, 2015.
- [6] J. Liu, J. Wang, Z. Tan, Y. Meng, and X. Xu, "The analysis and application of solar energy PV power," 2011 International Conference on Advanced Power System Automation and Protection, pp.1696–1700, 2011.
- [7] S. Vinod, M. Naveen, A.K. Patra, and A.A.R. John, "Accelerating towards larger deep learning models and datasets – A system platform view point," 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp.1–7, 2020.
- [8] K. Thonglek, K. Ichikawa, K. Yuasa, and T. Babasaki, "LSTM-based neural network model for predicting solar power generation," Proc. Technical Committee on Energy Engineering in Electronics and Communications (IEICE-EE), pp.7–12, May 2021.
- [9] O. Obulesu, M. Mahendra, and M. ThirilokReddy, "Machine learning techniques and tools: A survey," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), pp.605–611, 2018.
- [10] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," Proc. IEEE International Conference on Smart Grid Communications (Smart-GridComm), pp.528–533, 2011.
- [11] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," Proc. IEEE SoutheastCon, pp.1–6, 2016.
- [12] F. Rodríguez, A. Fleetwood, A. Galarza, and L. Fontán, "Predicting solar energy generation through artificial neural networks using weather forecasts for microgrid control," Renewable Energy, vol.126, pp.855–864, 2018.
- [13] M. Sajjad, Z.A. Khan, A. Ullah, T. Hussain, W. Ullah, M.Y. Lee, and S.W. Baik, "A novel CNN-GRU-based hybrid approach for short-term residential load forecasting," IEEE Access, vol.8, pp.143759–143768, 2020.
- [14] D. Wei, "Prediction of stock price based on lstm neural network," 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM), pp.544–547, 2019.
- [15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol.9, no.8, pp.1735–1780, Nov. 1997.
- [16] J. Zhang, Y. Chi, and L. Xiao, "Solar power generation forecast based on LSTM," Proc. IEEE International Conference on Software Engineering and Service Science (ICSESS), pp.869–872, 2018.
- [17] M.R. Islam, M.A.M. Hasan, and A. Sayeed, "Transfer learning based diabetic retinopathy detection with a novel preprocessed layer," Proc. IEEE Region 10 Symposium (TENSYP), pp.888–891, 2020.
- [18] D. Sarkar, R. Bali, and T. Ghosh, Hands-On Transfer Learning with Python: Implement Advanced Deep Learning and Neural Network Models Using TensorFlow and Keras, Packt Publishing, 2018.
- [19] T. Alshalali and D. Josyula, "Fine-tuning of pre-trained deep learning models with extreme learning machine," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), pp.469–473, 2018.
- [20] S. Chhabra, P. Majumdar, M. Vatsa, and R. Singh, "Data fine-tuning," Proc. AAAI Conference on Artificial Intelligence, pp.8223–8230, July 2019.
- [21] A. Senior, G. Heigold, M. Ranzato, and K. Yang, "An empirical study of learning rates in deep neural networks for speech recognition," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp.6724–6728, 2013.
- [22] Y. Wu, L. Liu, J. Bae, K.H. Chow, A. Iyengar, C. Pu, W. Wei, L. Yu, and Q. Zhang, "Demystifying learning rate policies for high accuracy training of deep neural networks," 2019 IEEE International Conference on Big Data (Big Data), pp.1971–1980, 2019.
- [23] K. Goutam, S. Balasubramanian, D. Gera, and R.R. Sarma, "Layer-out: Freezing layers in deep neural networks," SN Computer Science, vol.1, p.295, Sept. 2020.
- [24] X. Xiao, T. Bamunu Mudiyansele, C. Ji, J. Hu, and Y. Pan, "Fast deep learning training through intelligently freezing layers," 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp.1225–1232, 2019.
- [25] M. Oyeleye, T. Chen, S. Titarenko, and G. Antoniou, "A predictive analysis of heart rates using machine learning techniques," International Journal of Environmental Research and Public Health, vol.19, no.4, 2022.
- [26] S. Dhar, T. Mukherjee, and A.K. Ghoshal, "Performance evaluation of neural network approach in financial prediction: Evidence from indian market," 2010 International Conference on Communication and Computational Intelligence (INCOCCI), pp.597–602, 2010.
- [27] J. Konar, P. Khandelwal, and R. Tripathi, "Comparison of various learning rate scheduling techniques on convolutional neural network," 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pp.1–5, 2020.
- [28] K. Saito, T. Fujita, Y. Yamada, J. Ishida, Y. Kumagai, K. Aranami, S. Ohmori, R. Nagasawa, S. Kumagai, C. Muroi, T. Kato, H. Eito, and Y. Yamazaki, "The operational JMA nonhydrostatic mesoscale model," Monthly Weather Review, vol.134, no.4, pp.1266–1298, 2006.
- [29] V. Ramasamy, D. Feldman, J. Desai, and R. Margolis, "U.S. solar photovoltaic system and energy storage cost benchmarks: Q1 2021," Technical Report, National Renewable Energy Laboratory, Nov. 2021.



Kundjanasith Thonglek received the bachelor's degree in software and knowledge engineering from the Faculty of Computer Engineering, Kasetsart University, Thailand, in 2018. He received the master's degree in information science and engineering from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include edge machine learning, federated learning, neural network model com-

pression and acceleration.



Kohei Ichikawa is an Associate Professor in the Division of Information Science at Nara Institute of Science and Technology (NAIST) Japan. He received his B.E., M.S., and Ph.D. degrees from Osaka University in 2003, 2005, and 2008, respectively. He was a postdoctoral fellow at the Research Center of Socionetwork Strategies, Kansai University from 2008 to 2009. He also worked as an Assistant Professor at the Central Office for Information Infrastructure, Osaka University from 2009 to 2012. His current research

interests include distributed systems, virtualization technologies, Software Defined Networking and distributed machine learning.



Tadatoshi Babasaki was born in Nagasaki prefecture in Japan on October 5, 1965. He received his B.S. and M.S. degrees in electronic engineering, and Ph.D. degree in systems science from Nagasaki University, Nagasaki, Japan in 1988, 1990, and 2011, respectively. He previously worked at the R&D department of NTT Corp. He has been a senior manager with the Service Development Division of NTT Facilities, Inc. His research interests include DC power supply systems for telecommunication and lead

acid batteries. He is a member of IEEE, IEEJ, and IEICE.



Keichi Takahashi received his M.Sc. and Ph.D. degrees in information science from Osaka University, Osaka, Japan in 2016 and 2019, respectively. In 2018, he was a visiting scholar at the Oak Ridge National Laboratory, Oak Ridge, TN, USA. He was an assistant professor at the Nara Institute of Science and Technology, Nara, Japan from 2019 to 2021. Since 2021, he has been an assistant professor at Tohoku University, Sendai, Japan. His research interests include high performance computing and parallel distributed computing.

distributed computing.



Hajimu Iida received his B.E., M.E., and Dr. of Eng. degrees from Osaka University in 1988, 1990, and 1993, respectively. From 1991 to 1995, he worked for the Department of Information and Computer Science, Faculty of Engineering Science, Osaka University as a research associate. Since 1995 he has been with the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. His current position is a Professor of the Laboratory of Software Design and Analysis. His research interests include modeling and analysis of software and development process.

development process.



Chawanat Nakasan received his B.Eng. in Computer Engineering from Kasetsart University, Thailand, in 2012, followed by Master and Doctor of Engineering degrees from Nara Institute of Science and Technology, Japan, in 2015 and 2018. He currently works as a lecturer at Kasetsart University in Thailand, focusing on high-performance computing, software-defined networking, and operating systems.



Kazufumi Yuasa was born in Okayama prefecture in Japan on July 27, 1986. He received his B.S. and M.S. degrees in electrical and electronic engineering, and Ph.D. degree in life science and systems engineering from Kyushu Institute of Technology, Fukuoka, Japan in 2009, 2011, and 2022, respectively. He has been an Assistant Manager with the Service Development Division of NTT Facilities, Inc. His research interests include the integration of renewable energy sources and energy control optimization. He is a member of IEEE, IEEJ, IEICE, and IEIEJ.

optimization. He is a member of IEEE, IEEJ, IEICE, and IEIEJ.