Sensors and Materials, Vol. 35, No. x (2023) 1–24
MYU Tokyo

S & M

IoT Data Collection and Short-Term Solar Power Forecasting Using Stacked Generalization Ensemble Model

Chun-Liang Tung*

Department of Information Management, National Chin-Yi University of Technology
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 411030, Taiwan (R.O.C.)

(Received May 29, 2023; accepted November 14, 2023)

Keywords: SVR, LASSO, RIDGE, stacking model, sensor

Accurate forecasting of solar power generation plays an important role in stabilizing power dispatch. Therefore, numerous machine learning models and deep learning models have been used for power generation forecasting. However, since a single independent model still has performance limitations, the prediction model of the ensemble model is adopted to aggregate the advantages of different independent models. Furthermore, its prediction error and generalization performance are both better than those of a single model. In this study, the stacking ensemble method was employed to gather four different base learners and then incorporated with the *k*-fold cross-validation for model training and testing. Meanwhile, a mobile IoT data collection system was also built, in which IoT sensors were applied to collect data of weather factors (solar radiation, ambient temperature, humidity, and wind velocity) as well as of solar power generation. Next, the real-time monitoring system for solar power generation developed in this study displayed real-time power generation and data storage. The experimental results showed that the root mean square error of the solar power generation prediction model, that is, the regression ensemble model (RGEM) proposed by this study, dropped by 6.24, 8.31, 9.94, and 4.21%, respectively, in model testing compared with those of the independent model support vector regression, least squares support vector regression, least absolute shrinkage and selection operator, and ridge. Besides, RGEM's testing mean absolute percentage error = 0.0966 indicated a prediction model of high accuracy.

1. Introduction

Renewable energy is the energy received from nature, such as solar energy, wind power, and geothermal energy. Resources of renewable energy are abundant and can be continuously supplemented by nature. In addition, in the process of energy transformation (such as to electric energy), they do not produce other natural products that contain pollutants. Therefore, they are viewed as the most important sources of clean energy for the future.(1) Solar energy is currently one of the most critical renewable energy sources. According to the report of the International Renewable Energy Agency, the power generation capacity of global solar panels rose from 40

*Corresponding author: e-mail: cltung@ncut.edu.tw

ISSN 0914-4935

MW to 217 MW between 2010 and 2015, representing an increase of 442.5%, and from 291 MW to 580 MW between 2016 and 2019, which was almost 13.5 times the power generation capacity of the previous 10 years.[2] In addition, according to the Renewables 2022 Global Status Report, the power generation capacity of global solar panels increased from 767 MW to 942 MW between 2020 and 2022.[3] Solar power plants will supply electricity into the power grid and operate in parallel with other power plants. Consequently, research issues on power dispatching as well as system power supply stability and security are becoming increasingly important. Although solar power can generate clean electricity, its greatest disadvantage is unstable power supply. For example, electricity cannot be generated on cloudy days, in rainy seasons, and at night. Stable power cannot be supplied at night in midsummer or during the peak time of power consumption. Thus, power scheduling between the solar power plants and other power plants is a key factor in stable power supply. However, the premise of power scheduling is that the power that the solar power plant can generate the next day must be known, making power generation forecasting of the solar power plant a very important research topic.[4]

The power generation capacity of solar panels is affected by climatic factors such as ambient temperature, solar radiation, and weather. Accordingly, the different degrees of importance of these factors must be taken into account when predicting the power generation capacity. The climatic factors that affect the power generation capacity of solar panels include solar radiation, ambient temperature, photovoltaic (PV) panel surface temperature, humidity, wind velocity, wind direction, and dustfall. Among them, three factors, sunlight, ambient temperature, and PV panel surface temperature, have the greatest impact on the power generation capacity of solar panels.[5–10] Because of the impact of climate factors, the power generation forecasting model must be able to provide accurate forecasting results and to avoid overfitting and underfitting of the model. For instance, Jebli *et al.*[11] used the Pearson correlation coefficient to select the meteorological data required by different models, thereby escalating the accuracy of power generation forecasting as well as avoiding overfitting of the model. In the solar PV power generation forecasting field, applying machine learning (ML) algorithms to power generation prediction has received extensive attention. The research results of many groups have shown that the forecasting models built on ML algorithms could provide accurate estimations of solar power generation. For example, Mishra *et al.* proposed the use of the wavelet transform (WT) to convert solar energy time-series data into different frequency series for statistical feature extraction, followed by deep learning (DL) to optimize the learning ability of the long short-term memory (LSTM) network model, ultimately resulting in the optimal forecasting accuracy of power generation.[12]

In the research category of solar panel power generation forecasting, on the basis of the length of the prediction period, the research types can be divided into four categories. The first category is very short-term forecasting (1 s to 1 h), which is applied to real-time electricity dispatch and maintaining the grid stability. The second type is short-term forecasting (1 h to 24 h), which is applied to energy planning and grid management as well as to increase the security of the grid. The third category is medium-term forecasting (1 week to 1 month), used in scheduling the maintenance of grid management and energy planning. The fourth category is long-term forecasting (1 month to 1 year), which can be applied to energy policy making.[4,5,12]

This study belongs to the second category of power generation forecasting research and is aimed at predicting future solar power generation using different regression models [such as support vector regression (SVR), least square SVR (LSSVR), least absolute shrinkage and selection operator (LASSO), and ridge regression (RIDGE)] and an ensemble model. The accuracy of power generation forecasting by the regression model depends on which regression algorithms and input factors are used. In addition, the quality and quantity of training data will also affect the forecasting accuracy of the model. Erten and Aydilek[13] used different regression algorithms (such as Linear, RIDGE, LASSO, and Elastic) and the principal component analysis (PCA) feature extraction method to compare predictions of the maximum power generation capacity. Their research results revealed that all regression models could accurately predict the capacity of solar power generation, with the Elastic model in particular performing better than the others.

The advantage of using the ensemble model for power generation forecasting is that it can combine different algorithms to produce more accurate and robust forecasting capabilities as well as overcome problems of high variance, low accuracy, and data noise better than a single prediction model.[14] Carneiro *et al.*[15] adopted multilayer perceptron (MLP), cascade forward back propagation (CFBP), self-organizing map (SOM), and radial basis function (RBF) network as the front-end predictors of the ensemble model. In addition, RIDGE regression was adopted to perform the linear combination for the output of each predictor and then carry out the final prediction output of the model. The research results have revealed that the model using the ensemble model was more accurate than the single prediction model in predicting either solar power generation or wind power. Aikandari and Ahmad[16] suggested that ML models should be combined with statistical models as the front-end predictors of the ensemble model, and ultimately, the prediction output of power generation of the model should be obtained by means of different ensemble methods. Their research results indicated that this type of ensemble method, where the variance combination of the inverse approach was used, had small errors and high accuracy.

As mentioned above, the accuracy of the power generation prediction model using the ensemble model is higher than that of the model using a single prediction algorithm. Therefore, herein, a solar power generation prediction model, the regression ensemble model (RGEM), which uses different regression models as the front-end predictors of the ensemble model followed by the use of a gradient boosting regressor (GBR) as the final estimator to predict the output of the model as well as adopting R^2, mean square error (*MSE*), root mean square error (*RMSE*), mean aboslute percentage error (*MAPE*), and *k*-fold cross-validation (CV) for the evaluation of its efficiency and accuracy, is proposed. The predictors used by RGEM include four independent models: SVR, LSSVR, LASSO, and RIDGE. In the experiment, the horizon intervals of power generation forecasting are 15 min and 1 day. These four independent models are all capable of accurate power generation forecasting. However, if ensemble learning is used, then the RGEM prediction model can output more highly accurate forecasts of solar power generation.

Furthermore, in terms of solar energy data collection, in this study, another mobile data collector (MDC) with IoT sensors is developed, facilitating the data collection of weather factors, solar panel surface temperature, and power generation capacity. Its structure is designed

to use IBM X3650 M5 as the system server and Ubuntu Server as the operating system, collect data of weather factors and power generation voltage through IoT sensors, solar panels, and solar inverters, and then send data back to the solar energy monitoring system (SEMS) via the Raspberry Pi Embedded System and the Internet. After referring to many practical studies,[17] the PHP Laravel programming language was selected for use in the SEMS proposed in this study to develop the monitoring system of the Internet of Things. In this SEMS, the real-time numerical changes of voltage and power generation will be displayed, and the data will be stored in the InfluxDB time series database (TSDB), facilitating downloading of the historical data as well as data analysis and management in the future, so as to achieve the purpose of monitoring solar power generation. In the proposed SEMS, the server uses the Docker Virtualization Technology to process tasks such as data storage, data distribution, and data inspection in different containers. Users can carry out remote monitoring with data transmission via the Internet to observe and control the system status anytime and anywhere. If something goes wrong, users can immediately receive a message through the LINE SNS, and someone can be sent to fix it.

As described above, this study is focused on enhancing the accuracy and the robustness of the prediction model. The major contributions of this paper are as follows.

1. Hardware construction: A MDC was built with IoT sensors to collect data of solar radiation, wind velocity, ambient temperature, and humidity. Also, a solar PV power generation monitoring system centered on Raspberry Pi, Docker container technology, and the InfluxDB database was developed for data collection and real-time monitoring to assist the future research on solar power generation.
2. Data collection: IoT sensors were installed in the solar PV power generation experimental field (702 kW) of the adiCET research center of Chiang Mai Rajabhat University in Thailand, and data on solar panel power generation and weather factors, including solar radiation, solar panel surface temperature, and ambient temperature, were also collected.
3. Data analysis and model evaluation: After the proposed ensemble model performed data preprocessing, model training, and testing using the adiCET solar database, the *MAPE* (15 min ahead) of the prediction model was found to be 0.0966, which indicated that the proposed model can accurately predict solar power generation.

This paper is structured as follows. In Sect. 2, related literature concerning different types of algorithms are explored and the current research status of solar power generation forecasting is explained. In Sect. 3, the statistical approaches, ML algorithms, and the ensemble model adopted in this study are discussed. The evaluation and experimental results of the different models are presented in Sect. 4. Finally, the conclusions of this study are given in Sect. 5.

2. Related Works

Many studies have adopted statistical approaches, ML, and DL as models for predicting solar panel power generation. Frequently used statistical and ML algorithms include support vector machine (SVM)[18], SVR, LASSO, RIDGE, autoregressive (AR), AR integrated moving average (ARIMA), and ARIMA with exogenous inputs (ARIMAX) algorithms. Some commonly

applied DL algorithms include artificial neural network (ANN), and LSTM. AR, ARIMA, and ARIMAX all belong to the time series of forecasting models in statistical approaches, which are suitable for short-term forecasting and long-term forecasting. ARIMA is an extended model of AR because it takes into account the nonstationarity of the datasets and can handle data with trends and seasonal components. Therefore, ARIMA is suitable for long-term forecasting. ARIMAX is an extension of the ARIMA model since the model contains exogenous variables, such as weather data or external independent variables that can affect dependent variables, so that it can enhance the accuracy of predictions. Kim *et al.*[19] used seasonal autoregressive integrated moving average with exogenous factors (SARIMAX) and LSTM as the front-end predictors of a prediction model with the stacking ensemble technique to predict the power generation. The experimental results demonstrated that the *RMSE* of this ensemble model was 95.800, which was lower than that of other models (SARIMAX: 102.575, LSTM: 106.123, SVR linear: 109.130, deep neural network (DNN): 101.783, random forest: 106.226). Compared with the traditional time series of forecasting models, ANN can better handle and represent complex nonlinear relationships among variables. Recurrent neural network (RNN) is a neural network that is suitable for processing the time-series sequential data. However, in the training phase of the model, if the sequential data is too long, the vanishing gradient problem may arise. Since LSTM is an RNN-type neural network, which can retain or delete information by controlling the gate of information flow and a memory cell, it can solve the vanishing gradient problem as well as process longer sequential data.

As weather factors were adopted in this study to predict solar power generation, the applied ensemble model used Statistical Approaches and ML algorithms as the front-end predictors of the model. Finally, the prediction results were output through the meta-model. SVM is a supervised learning algorithm often1 applied to data classification. Its main principle is to project raw datasets to high dimensions via kernel functions, find a hyperplane with the maximum width for data classification, and use kernel functions to solve the problem of non-linearly separable data. Zeng and Qiao[20] used SVM as the basis of modeling and adopted RBF kernel functions and historical data of atmospheric transmittance in two-dimensional form, and related meteorological variables to conduct the predictions of atmospheric transmittance and power generation. The research results revealed that their prediction accuracy performed better than the AR model in the time series model and the RBF neural network model (RBFNN) in the neural network.

SVR is an extended model of SVM and is a commonly used nonlinear regression model. The SVR model establishes the nonlinear relationships between input variables and output variables through data conversion of kernel functions. The commonly used kernel functions include RBF and polynomial kernels. Alfadda *et al.*[21] used SVR and five different factors (outdoor temperature, solar irradiance, module temperature, wind velocity, and output power) to construct the prediction model and compared the *RMSE* error of its predicted values with that of the models such as linear regression, quadratic regression, and LASSO. The research results indicated that the SVR model had a lower *RMSE* value of power generation prediction. Fentis *et al.*[22] employed LSSVR and feed-forward neural network (FFNN) to build the power generation prediction model. The research results showed that LSSVR had a lower *RMSE* value than FFNN

(LSSVR, *MSE* = 0.0043, R^2 = 0.96). LASSO and RIDGE are two regularization techniques commonly used in regression models. The purpose is to avoid the problem of model overfitting caused by an overly complex model as well as to reduce generalization error without affecting the training error, so that the model can improve its generalization ability and prediction accuracy when facing new data in the future. Tang *et al.*[23] proposed a power generation forecasting model built with the LASSO regression model, including coefficient estimation and link function estimation, to carry out the coefficient estimates of the regression model and of the link function. The research results indicated that when the *RMSE* values of the LASSO-based model, the SVM model, and time-varying local linear estimation (TLLE) model were compared, that of the LASSO-based model was greatly reduced by 60.06%, and the lowest *MAPE* value was 3.3357, indicating that the LASSO-based model could accurately predict power generation. The difference between the RIDGE model and the LASSO model is that the RIDGE model uses L2 regularization to avoid the problem of model overfitting. In its loss function, the penalty term controls the size of the coefficient, and the coefficient value of the noninfluential variable is close to zero, thereby decreasing the SSE of the model and improving the generalization performance of the model.

The ensemble model lowers the errors and biases of a single prediction model by combining multiple statistical models or ML models, so that the accuracy and robustness of the prediction model can be increased. Numerous research results have shown that in solar power generation forecasting, the accuracy of the ensemble model is higher than that of a single forecasting model.[14,16] Amarasinghe *et al.*[24] came up with an ensemble model comprising a combination of three models: deep belief network (DBN), SVM, and random forest (RF). They first classified the data for the weather, then trained and tested multiple ensemble models, and finally output the power generation prediction (*RMSE* = 0.0591). Their research results revealed that the *RMSE* of the ensemble model compared with that of the three single DBN, SVM, and RF models (the training data of the three single models were not preprocessed by weather classification) was lowered by 8.74% on average (*RMSE* reduction: DBN 10.49%; SVM 7.78%; RF 7.95%). Sharma *et al.*[25] first decomposed the time series data, then constructed an ensemble model with multivariate LSTM for training and output the weight of each LSTM, and lastly, carried out weighted aggregation and obtained the final output of the prediction value. In the results of the 1-day-ahead experiment, the *MAPE* = 1.526 and *RMSE* = 0.1109 of the ensemble model were lower than those of other compared models (*MAPE*: DWT-LSTM 1.7423; LSTM 1.7744; RNN 2.5326; GRU 2.5321; neuro-fuzzy technique 1.5491) (*RMSE*: discrete wavelet transformation LSTM (DWT-LSTM) 0.2437; LSTM 1.2547; RNN 1.2467; GRU 1.2334; neuro-fuzzy technique 0.1146), proving that this ensemble model could accurately predict power generation.

GBR is a ML algorithm frequently used for dealing with regression tasks and comprises a combination of gradient descent and boosting algorithms. GBR is constructed as a strong learner using the prediction results of multiple weak learners, and maintains nonlinear relationships between data. For instance, Persson *et al.*[26] adopted gradient-boosted regression trees to predict the power generation of different solar plants and compared their power generation; in the 1–6-hour power generation forecast, the normalized root mean squared errors for all power plants were between 0.100 and 0.137. In addition, GBR uses the regularization techniques to avoid the problem of overfitting and has good robustness in processing outliers.

In this study, after we conducted an extensive literature research on solar power generation prediction, we found that in most of the studies, statistical models, ML models or hybrid techniques were adopted for power generation prediction, all of which have good prediction accuracy. In particular, the accuracy of power generation prediction using hybrid techniques or ensemble models is relatively high and superior to that of a single prediction model. The advantage of SVR is that data can be projected to high dimensions by the use of kernel functions, which can effectively grasp the nonlinear relationship between features, thereby improving the accuracy of model prediction. LSSVR employs a squared-loss function that can simplify the problem of model optimization and has a good ability to deal with noise data and outliers, enhancing the model robustness. LASSO is a regularization regression model, characterized by its ability to handle the multi-collinearity problem between features, and is suitable for features with low-dimensional dense characteristics. RIDGE is a linear regression technique that combines feature selection and regularization. Meanwhile, it can also deal with the phenomenon of model overfitting. Considering the above descriptions, in this study, we planned to use these four superior models as the base learners of the ensemble model. Therefore, after considering two factors—the overfitting problem and generalization performance of the prediction model—we tested the prediction abilities of SVR, LSSVR, LASO and RIDGE models with the *k*-fold CV, and then we constructed an ensemble model. Additionally, we used GBR as the final predictor as well as the output results of power generation prediction.

3. Methodology

In this section, first, we explain the structures and operating principles of the MDC and the real-time monitoring system for solar power generation (RMSP) built in this study. Next, we elaborate on the four independent models (including SVR, LSSVR, LASO, and RIDGE) and the solar power generation prediction model—RGEM—proposed in this study.

3.1 MDC and RMSP framework

The MDC mainly applies three communication protocols—RS485, Modbus RTU, and message queuing telemetry transport (MQTT)—to transmit information. Modbus RTU is used to receive data from different sensors (such as the PM2.5 sensor, temperature and humidity sensor, solar panel current sensor, solar panel surface temperature sensor, solar radiation sensor, and cup-type wind velocity sensor). RS485 is used as the interface between Modbus RTU and Raspberry Pi. Since the outputs of each sensor are the analog values of voltage and current, data exchanges among devices need to be carried out by Modbus RTU. MQTT, mainly used for the connection between Raspberry Pi and the recipient computer, is a machine-to-machine communication protocol. After the data are received by MQTT, the data are stored in the InfluxDB TSDB through Node-Red and JavaScript programs, and finally, the data are displayed through Grafana. The complete MDC hardware structure and the adopted sensors are shown in Fig. 1.

The design principle of our RMSP was developed using the Laravel web application framework and Vue.js framework. Firstly, the embedded system of Raspberry Pi collects data

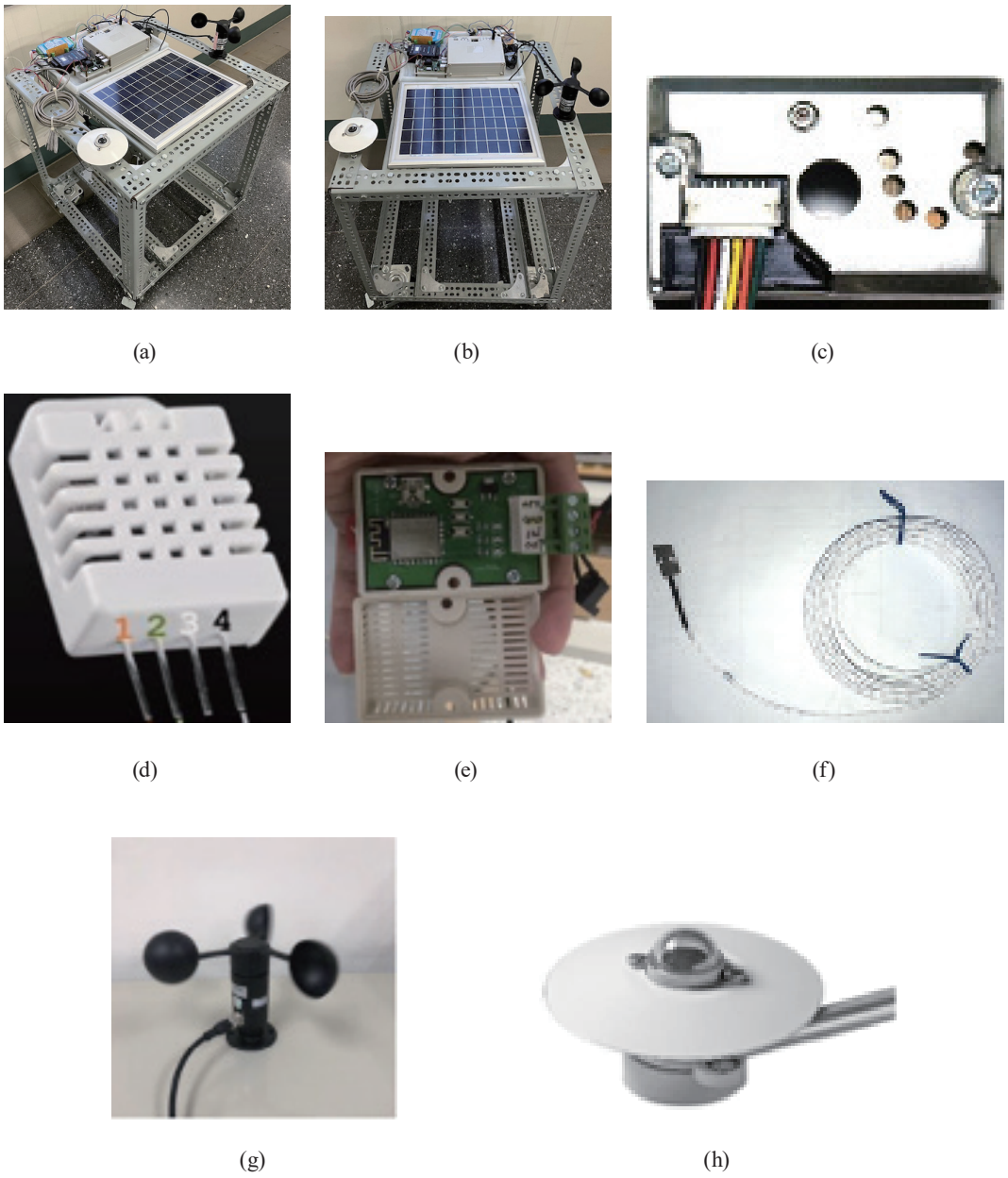


Fig. 1. (Color online) (a) Left view of MDC, (b) top view of MDC, (c) PM2.5 sensor, (d) temperature and humidity sensor, (e) solar panel current sensor, (f) solar panel surface temperature sensor, (g) wind velocity sensor, (h) solar radiation sensor.

from the inverter, and all data are stored through the system API. Next, the Docker container technology is employed to divide all assignments of the system and put each of them into respective containers. In each container, the data are stored in the form of Queue, and in the InfluxDB TSDB, the real-time data are broadcast to each user; then, real-time images are sent by video streaming (RTMP). Lastly, relevant information is given to the users. The above steps are shown in Fig. 2.

The front-end of RMSP mainly uses Vue for screen layout and design, while the back-end incorporates PHP Laravel with MySQL and InfluxDB for transfer and storage of the overall

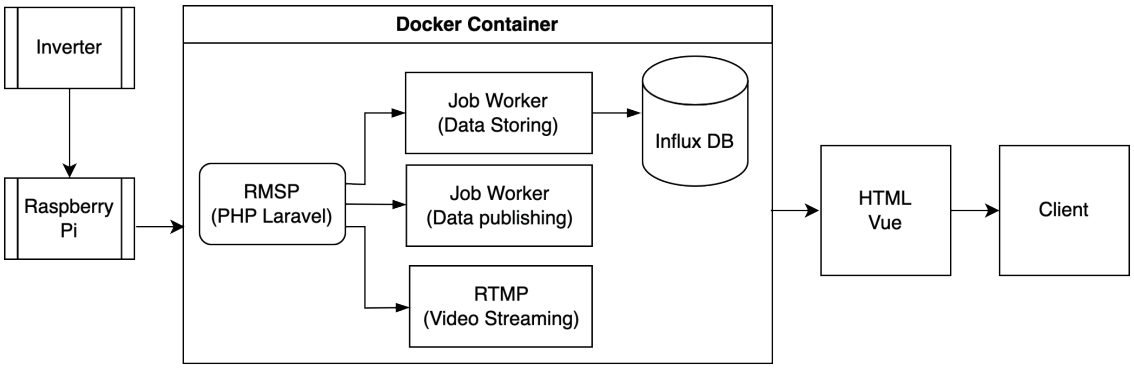


Fig. 2. RMSP framework, in which the Docker container technology can provide safer and faster information systems development and deployment.

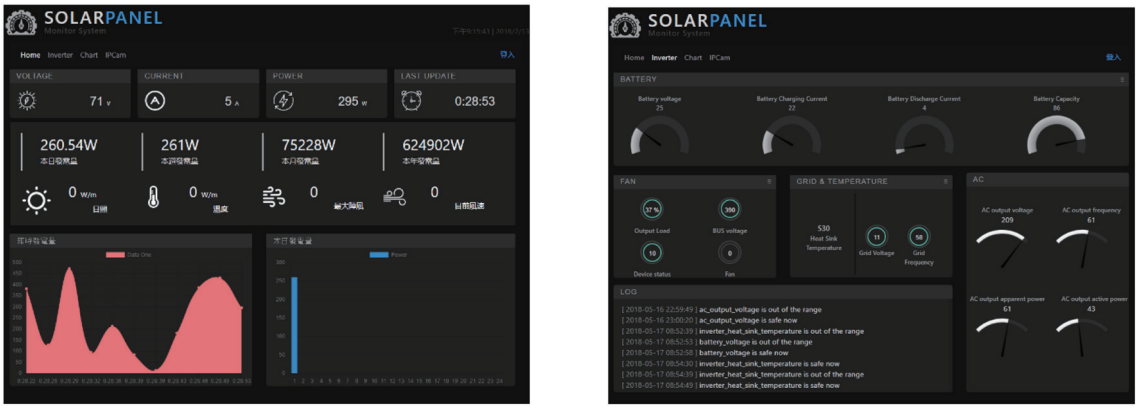
data. For the warning message prompt function, LINE API is applied for notification. When the system detects an abnormality or the hardware temperature becomes too high, the system will automatically send a warning message to the administrator through LINE. In addition, this system carries out real-time monitoring of the solar hardware equipment using the video streaming server, the Raspberry Pi embedded system, and the Raspberry Pi Camera V2 of the embedded system.

In the main system of RMSP, Ubuntu is used as the operating system in the bottom layer of the server, and the system is divided into several subsystems and distributed to each Docker container, in order to protect the functioning of each system. In this way, when one of the systems malfunctions or stops, the operation of other systems will not be affected, and the systems can run more efficiently. The main system uses Nginx as the web server software. In addition to stability and high efficiency, Nginx has another feature, that is, there are many additional modules that provide a better structure for Nginx, allowing others to write modules for it as well as expand or strengthen its original function. In the main system, different tasks are divided into different subsystems through the Redis cache database software. As a medium for distributing tasks, Redis is a fast, open-source key-value data structure storage area in the random-access memory. The data broadcasting of RMSP is connected with users using the WebSocket protocol and is responsible for releasing the latest data to the users' front-end. In the Laravel module, the Laravel-Echo-Server, the server system for data broadcasting, not only works with Laravel's user authentication and private channels but also allows the bottom layer to be written by Node.js, so it supports the cross-platform setup. Users can obtain webpage content through HTTP and then connect the Laravel Echo Server with the WebSocket written in Javascript in the website. The Laravel Echo Sever will verify users' information with the Laravel main system, and finally, messages will be broadcast to Redis by the Laravel main system. Next, the Laravel Echo Sever will send the published messages to the users. The main duty of the job worker in RMSP is to perform the tasks proposed by the system, because the system will assign each job to a different queue and write it into Redis. Then, the job worker will read the assigned job from Redis and process it in the back-end. Different job workers can handle jobs such as data writing, data publication, data verification, and other data-related issues separately. The function

expansion of RMSP is very flexible. As long as more sensors are connected to the system, the data can be displayed in the front-end of the system in real time. The solar monitoring system is developed on the basis of the above design principles, as shown in Fig. 3, where the upper half of Fig. 3(a) shows the real-time examination of several important types of data (voltage, current, and power), the day's power generation, the week's power generation, the month's power generation, the year's power generation, and the weather conditions, such as solar radiation, temperature, and wind speed of the day. The lower half presents the real-time power generation and the day's power generation as a line graph and bar graph, respectively. Figure 3(b) reveals the detailed information of the inverter and other information of hardware equipment, such as AC power, battery, temperature, and status of the fan. At the same time, the system provides a warning reminder. Any abnormal state of the hardware device can be learned from the instant message displayed in the LOG block. If the hardware device is abnormal, we can know the content of the abnormal status through the message displayed by this function and immediately notify maintenance personnel to fix it.

3.2 Regression-based algorithm

Given that the prediction accuracy of a prediction model using a regression algorithm needs to be enhanced, feature selection must be carried out first, and then, the data dimension must be lowered before data preprocessing. Next, feature scaling or feature standardization is necessary. The main purpose of feature selection is to improve the performance of the ML model and reduce the complexity of the model. For example, the PCA can be used to reduce the dimensions of features. Feature scaling and feature standardization are two techniques widely used to convert features into common scales. Feature scaling can convert feature values into a specific numerical range, such as between 0 and 1, which is highly suitable for dealing with large changes in the range of feature values; this common technique resembles min-max scaling. Feature standardization can convert features into a mean of 0 and a standard deviation of 1 to ensure that all features are on the same scale; *Z*-score normalization is one of the common standardization



(a) (b)

Fig. 3. (Color online) Front-end screen displays of RMSP.

methods. After the features are processed by feature selection, feature scaling, or feature standardization, they can be adopted in different prediction models as independent variables or dependent variables for different purposes of forecasting.

SVR is an extended model of SVM. SVM is mainly used to solve the classification problem. Through the optimization of its objective function, the best hyperplane with the best boundary, or the maximum boundary, can be found to classify data into two categories, and its hyperplane can also maximize the margin of errors and minimize the training errors, thereby reducing the generalized errors as well as increasing the generalization performance of the model. What is commendable about SVR is that it projects data into a high dimension with the kernel function and then searches for the hyperplane, so it can handle the nonlinear relationship between independent variables and dependent variables, where data not linearly separable can be classified.

In this study, the independent variable of the training set $\{(x_i, y_i)\}_{i=1}^{N}$, $x_i \in R^d, y_i \in R$, was x = [*Solor irradiance*, *Ambient temperature*, *PV temperature*], the dependent variable was y = [*Solor Power*], and the linear function of the training set was $f(x) = w^T x_i + b$. The main purpose of the objective function of SVR is to find a regression model whose hyperplane has the minimum data error and the maximum margin of error, as shown in Fig. 4, where w means the weight of the feature, b represents the bias term, the upper and lower boundaries of the hyperplane are $f(x) = w^T x_i + b + \varepsilon$ and $f(x) = w^T x_i + b - \varepsilon$, $\{+\varepsilon, -\varepsilon\}$ means the acceptable data error of the linear function to the boundary, and $\{\xi_i, \xi_i^*\}$ refers to the deviation of the data outside the soft-margin range. The slack variable ξ can be used to determine the amount of data outside the hyperplane. The optimization problem and constraints of SVR are expressed as

$$\min \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i + \xi_i^*, \tag{1}$$

subject to

$$y_i - w^T \varphi(x_i) \leq \varepsilon + \xi_i^*, \; i = 1, ..., N,$$

$$w^T \varphi(x_i) - y_i \leq \varepsilon + \xi_i^*, \; i = 1, ..., N,$$

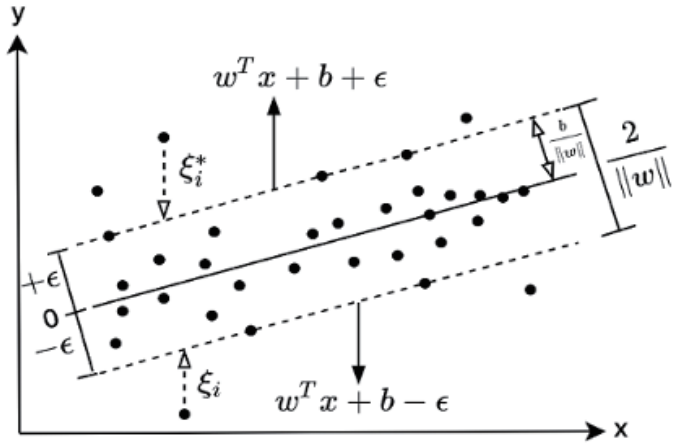


Fig. 4. Hyperplane of SVR, which can maximize the margin of error of the data.

$$\xi_i, \xi_i^* \geq 0, i = 1, ..., N,$$

where C denotes a regularization parameter, which can be used to adjust the weights between the margin and the error of the hyperplane. The larger the value of C, the greater the weight given to the model to diminish the error. $\varphi(x_i)$ is the kernel function. The optimization process of Eq. (1) can be derived using the Lagrangian function, Lagrange multipliers, and the quadratic optimization problem. The dual problem can also be derived by applying Karush–Kuhn–Tucker (KKT) conditions.[27] The derived result is expressed as

$$f(x) = \sum_{i=1}^{N_{SV}} (\alpha_i^* - \alpha_i) \langle \varphi(x), \varphi(x_i) \rangle + b,\ \alpha_i, \alpha_i^* \in [0, C], \tag{2}$$

where N_{SV} is the number of support vectors, $\{\alpha_i,\ \alpha_i^*\}$ refers to Lagrange multipliers, and $\alpha_i, \alpha_i^* \geq 0$. In accordance with Mercer's condition, the inner product of $\langle \varphi(x), \varphi(x_i) \rangle$ is calculated, and the dot product of the feature vectors in the high dimension can be computed using the kernel function $K(x, x_i)$.[28] Finally, the predicted value of the latest data can be obtained after the calculation by using the trained weight vectors and the error term b.

LSSVR is an extended model of LSSVM. The optimization problem and constraints of LSSVR are expressed as

$$\min_{w,b,e} J(w,e) = \frac{1}{2} \| w \|^2 + \frac{1}{2} \gamma \sum\nolimits_{=1} e^2, \tag{3}$$

subject to

$$y_i = w^T \varphi(x_i) + b + e_i,\ i = 1, ..., N.$$

In Eq. (3), the problem can be simplified by equality constraints and the least squares approach, in which $e_i \quad R$ represents the error variables of the data and γ denotes the regularization constant where $\gamma \geq 0$. If γ is large, it will lead to a decrease in the complexity of the model, which means that the low fitting level of the training data decreases. Similarly, the optimization process of Eq. (3) can solve the dual problem using Lagrangian function and Lagrange multipliers,[29] and the result is

$$f(x) = \sum_{i=1}^{N} \alpha_i \langle \varphi(x), \varphi(x_i) \rangle + b, \tag{4}$$

where α_i represents Lagrange multipliers and b denotes the bias term. The calculation of the inner product of $\langle \varphi(x), \varphi(x_i) \rangle$ can be replaced by the kernel function $K(x, x_i)$, in order to speed up the calculation efficiency. Common kernel functions include:

(1) Linear Kernel: $K(x, x_i) = \langle x, x_i \rangle$,

(2) Polynomial Kernel: $K(x, x_i) = \langle x, x_i \rangle^d$, $d \in N$,

(3) Gaussian Kernel (also called Radial Basis Function Kernel, RBF):

$K(x, x_i) = \exp\left(-\|x - x_i\|_2^2\right) / 2\sigma^2$,

(4) Sigmoid Kernel: $K(x, x_i) = \tanh(\alpha x^T x_i + C)$, $\alpha > 0$, $C > 0$,
(5) Chi-squared Kernel: $K(x, x_i) = \exp(-\gamma \sum_i (x - x_i)^2 / (x + x_i))$.

The efficiency of LSSVR in model training is higher than that of SVR. The reason is that the optimization problem is simplified by equality constraints and the least squares approach. As a result, faster model training efficiency can be achieved. The approach used to minimize the regression error is the least squares approach, instead of the margin-based approach adopted by SVR. In other words, LSSVR uses the least squares loss function rather than ε-insensitive loss function. Therefore, the training process of the LSSVR model is faster than that of the SVR model.

The LASSO model is a linear regression technique that combines feature selection and regularization and can handle the phenomenon of model overfitting; this is suitable for features with high-dimensional sparse characteristics. In the linear regression model, ordinary least square (OLS) is the most commonly used estimation method of model coefficients, but the major problem for OLS is that overfitting easily occurs with this model. Consequently, LASSO adds a penalty term to the objective function of OLS to adjust the complexity of the model. The objective function of OLS is shown as

$$\min_{\beta_0, \beta} \left(\frac{1}{N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 \right), \tag{5}$$

subject to

$$\sum_{j=1}^{P} \left| \beta_j \right| \le t,$$

where y_i is the dependent variable, x_i^T means features, and β is the vector of coefficients of the model. OLS employs a method of minimizing the data error to carry out the estimation of the model parameters, so that overfitting easily occurs. If multi-collinearity exists among features, then there is a great impact on the prediction accuracy of the model. LASSO is based on OLS and adds penalty items to adjust the complexity of the model and reduce the feature dimension. The objective function of LASSO is shown as

$$\min_{\beta_0, \beta} \left\{ \left(\frac{1}{N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 \right) + \lambda \sum_{j=1}^{P} \left| \beta_j \right| \right\}, \tag{6}$$

where the first term refers to the OLS loss function, the second term is the L1 penalty term, and λ is the regularization parameter, which is used to control the strength of the penalty term. If λ is larger, it indicates a stronger penalty for coefficients, which means that more coefficients will be forced to be zero and features that have a stronger influence on the model can be selected. In other words, it will reduce the complexity of the model. The adjustment of the λ value can be accomplished by means of CV or using information criteria such as the Akaike information criterion (AIC) or Bayesian information criterion (BIC). An optimal λ value can lead to better generalization performance.[30]

RIDGE is a type of regularization regression model and is characterized by its ability to handle the multicollinearity problem among features as well as its suitability for features with

low-dimensional dense characteristics. RIDGE is very similar to LASSO. However, since the penalty term is calculated as the sum of the squared coefficients, the selection of features cannot be performed. RIDGE's difference from LASSO is that an L2 penalty term is added to the objective function to avoid overfitting and overcomplexity of the model,[15] as shown by

$$\min_{\beta_0,\beta}\left\{\left(\frac{1}{N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2\right)+\lambda\sum_{j=1}^{P}\beta_j^2\right\}, \tag{7}$$

where the first term refers to the OLS loss function, and the second term denotes the L2 penalty term. The L2 penalty term is the sum of squared feature coefficients, and λ is used to control the strength of the penalty term. When $\lambda = 0$, it means that only OLS is employed to estimate the coefficients, which is the coefficient estimation procedure applied by general regression models. Nevertheless, when $\lambda = \infty$, it indicates that the coefficient estimation procedure will s*et all* coefficients to zero. The smallest residual sum of squares (RSS) can be estimated through OLS, which implies that when the RSS is relatively large, the strength of the penalty term must be increased to achieve a balance between the RSS and the penalty term. The optimization process of RIDGE is intended to minimize the OLS and the L2 penalty term as well as estimate better coefficients of $\hat{\beta}$ so as to increase the accuracy of the model.

3.3 Proposed model

In this study, we proposed a solar power generation prediction model, namely, RGEM, which is a stacked generalization model. This model uses the meta-learning algorithm as the learning algorithm. The stacking model combines the prediction results of different base learners, and finally, the meta-model outputs the final prediction result. RGEM employs a two-layer architecture to conduct model training and testing. Level One performs training and k-fold CV of four base learners (SVR, LSSVR, LASSO, and RIDGE). We adopted GBR as the meta-model, so the meta-model was trained and tested through Level Two. The framework of RGEM is depicted in Fig. 5.

In Fig. 5, the original data $\{(x_i, y_i)\}_{i=1}^{N}$, $x_i \in R^d, y_i \in R$ after data preprocessing is divided into training dataset D and testing dataset E; in Level One, the k-fold CV is used to train and test four base learners, respectively, and the generated prediction results, called Meta-X, are retained and can be used as the training dataset of the meta-model. The calculation process is shown by Eq. (8). In Level One, the trained base learners test the model with the original testing dataset, and the generated prediction results, called Meta-Y, are retained after the mean is calculated, and can be used as the testing dataset of the meta-model. The calculation process is shown by Eq. (9). Finally, the meta-model will use Meta-X for model training and Meta-Y for model prediction.

$$f_{Meta-X}(D)=\sum_{m=1}^{M}\sum_{k=1}^{K}f_m(D_{(k)}), \tag{8}$$

Here, D indicates that the training dataset is divided into k-fold data groups, for example, in 5-fold CV,$D = \{d_1, ..., d_k \mid k = 5\}$; $D_{(k)}$ refers to the data of the validation set in the 5-fold CV. m

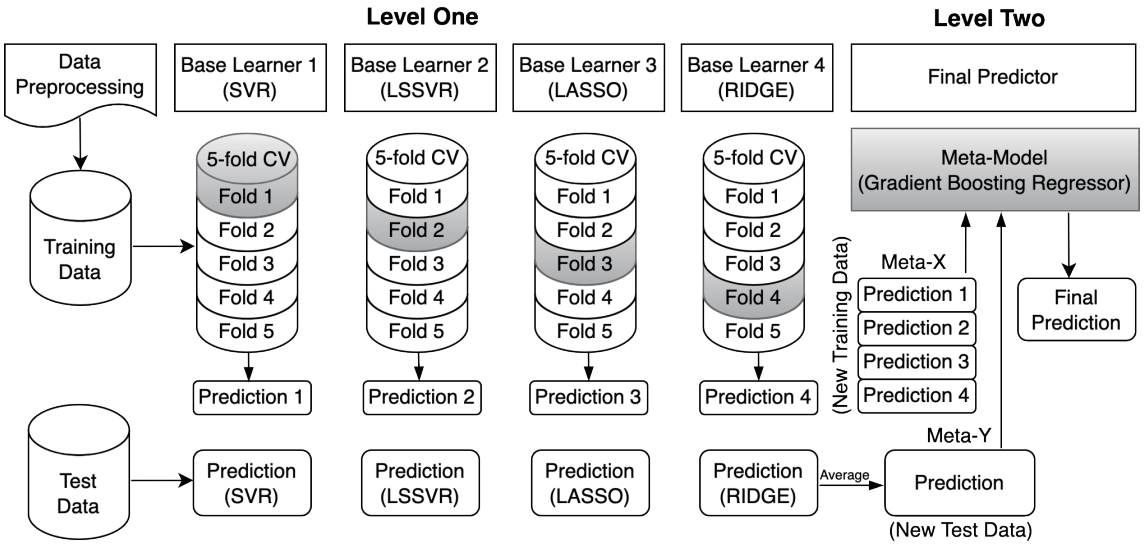


Fig. 5. The two-layer stacked generalization model adopted in this study uses the datasets (Meta-X and Meta-Y) generated by Level One for training and testing of the meta-model.

denotes the base learner. Therefore, through the calculation of Eq. (8), the predicted values and feature values of all base learner validation sets can be obtained and used in the training of the meta-model.

$$f_{Meta-Y}(E) = \frac{1}{M}\sum_{m=1}^{M} f_m(E), \tag{9}$$

E represents the testing datasets provided to each base learner. Consequently, via the calculation of Eq. (9), the average predicted values and feature values of all base learner testing datasets of can be obtained and regarded in the testing of the meta-model.

3.4 Model performance evaluation

After the ML model is built up, evaluation metrics are usually required to test its overall prediction error or classification error, in order to verify and ensure the performance of the model. In this study, *MSE*, *RMSE*, the coefficient of determination (*R*-square or R^2), and *MAPE* were used as the performance indicators of four independent models (SVR, LSSVR, LASSO, and RIDGE) and the RGEM stacked generalization model, as expressed below:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{10}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{11}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}, \tag{12}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{13}$$

where y_i, $\hat{y}_i$, and $\bar{y}_i$ respectively represent the observed (or actual) value of the target variable, the predicted value of the target variable, and the average of the target variable. N refers to the number of instances of features. Both *MSE* and *RMSE* are commonly used evaluation metrics of model prediction error. However, since *RMSE* is the square root of *MSE*, it is sensitive to large errors. R^2 is widely applied as a performance index of the regression model. In statistics, it implies that the proportion of the variances of the dependent variables can be explained by the independent variables in the model. In other words, R^2 can be used to evaluate the explanatory power of the model, and the value of R^2 ranges from 0 to 1; the larger the value, the better the goodness of fit of the model. *MAPE* is an indicator (metric) of the prediction accuracy of the model. It is expressed as a percentage in the range from 0 to infinity. Generally speaking, when the *MAPE* value of the model is less than 10%, the prediction ability of the model is "highly accurate forecasting"; when the *MAPE* value is between 10% and 20%, the prediction ability of the model is "good forecasting"; when the MAPE value is between 20% and 50%, the prediction ability is "reasonable forecasting"; when the *MAPE* value is greater than 50%, the prediction ability of the model is "inaccurate forecasting".[31]

4. Experimental Results

4.1 Data description

Since 2017, this study has be a part of the solar data transmission and analysis cooperation project of adiCET (Asian Development College for Community Economy and Technology), Chiang Mai Rajabhat University, Thailand. Therefore, we use the data of the 702 kW solar power experiment field (latitude 19.024293°N, longitude 98.940272°E) built by the adiCET for model training and testing.

The data were collected from 09:00 to 16:00 every day (the interval of data sampling was 15 min) between January 1 and November 30, 2021, for a total of 9,687 raw data values. The specifications for the data collection were solar power (kW), solar radiation (W/m^2), ambient temperature (°C), and PV panel surface temperature (°C) (the significant features influencing power generation forecasting have been explained in the first section), as shown in Fig. 6. Figure 6(a) shows the raw data sets of solar power generation used in this study, and Figs. 6(b) to 6(d) indicate raw data sets of solar radiation, ambient temperature, and PV panel surface temperature, as well as their correlations with solar power generation.

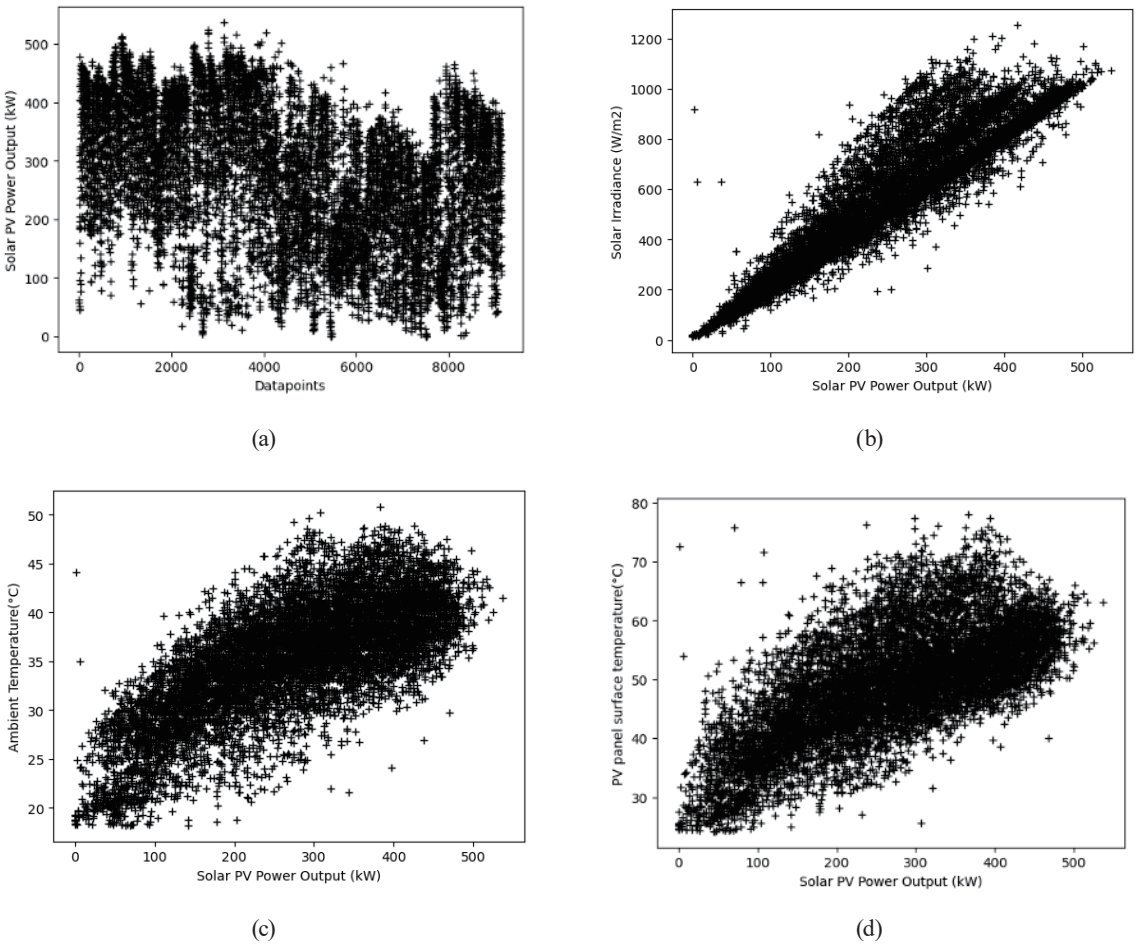
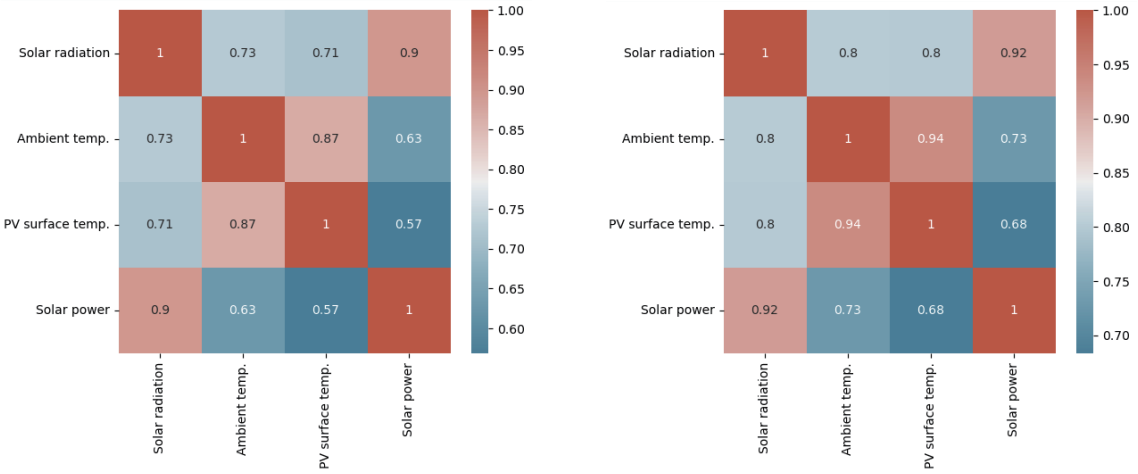


Fig. 6. Solar power generation data and three significant features affecting the power generation of solar PV panels.

4.2 Data preprocessing

To enhance the training and testing accuracy of the ML model, data preprocessing is a necessary task, including data cleaning, feature selection, and data normalization. The purpose of data cleaning is to eliminate noisy data, missing values, and outliers as well as to avoid incorrect data analysis results. When outliers appear in the data, the data can only be deleted from the statistical point of view or by judgment based on professional experience. Feature selection can reduce the complexity of the model and improve computing performance. The most commonly used feature selection technique is Pearson correlation analysis, which evaluates the linear relationship between variables by calculating the covariance and standard deviations between variables. In this study, after the data were preprocessed, the amount of data in the solar power generation database used in this study decreased from 9,687 to 9,147. Figure 7 displays the results of the Pearson correlation analysis on the significant factors adopted in the prediction model of this study. The results of Pearson correlation analysis of the raw data are illustrated in Fig. 7(a), in which solar power and solar radiation show a positive correlation of 0.90, solar power



(a) (b)

Fig. 7. (Color online) Pearson correlation analysis of significant factors: (a) Pearson correlation of the raw data; (b) Pearson correlation after the raw data were processed by data cleaning.

and ambient temperature a positive correlation of 0.63, and solar power and PV panel surface temperature a positive correlation of 0.57. The Pearson correlation analysis after the raw data were processed by data cleaning is displayed in Fig. 7(b), where solar power and solar radiation show a positive correlation of 0.92, solar power and ambient temperature a positive correlation of 0.73, and solar power and PV panel surface temperature a positive correlation 0.68. The results of comparison revealed that all the dependent variables and the independent variables have strong positive correlations. In particular, solar power and solar radiation have stronger linear relationships than other variables.

The purpose of data normalization is to convert the raw data into a standard format or scale, so that the data of different scales or units can be standardized and the consistency of the data can be retained as well. By doing so, the performance of the model can be boosted to ensure the reliability of the output results of the model. Common data normalization techniques include Min–Max scaling and *Z*-score standardization. We adopted the Min–Max scaling technique to convert the data to the range between 0 and 1. The mean and the standard deviation were applied to the data conversion process of *Z*-score standardization. When the data has outliers, converting the data range using the *Z*-score standardization is not recommended.

4.3 Results and discussion

The solar power generation database adopted in this study had a total of 9,147 data after data preprocessing, and the 5-fold CV was used for model training and evaluation, so that CV could more accurately explain the generalization performance and robustness of the model. During the experiment, R^2 (actual solar power versus predicted solar power) of each base learner in the model training phase and testing phase was higher than 0.84, indicating a better model performance of RGEM than those of the four independent models, as shown in Fig. 8, where (a)

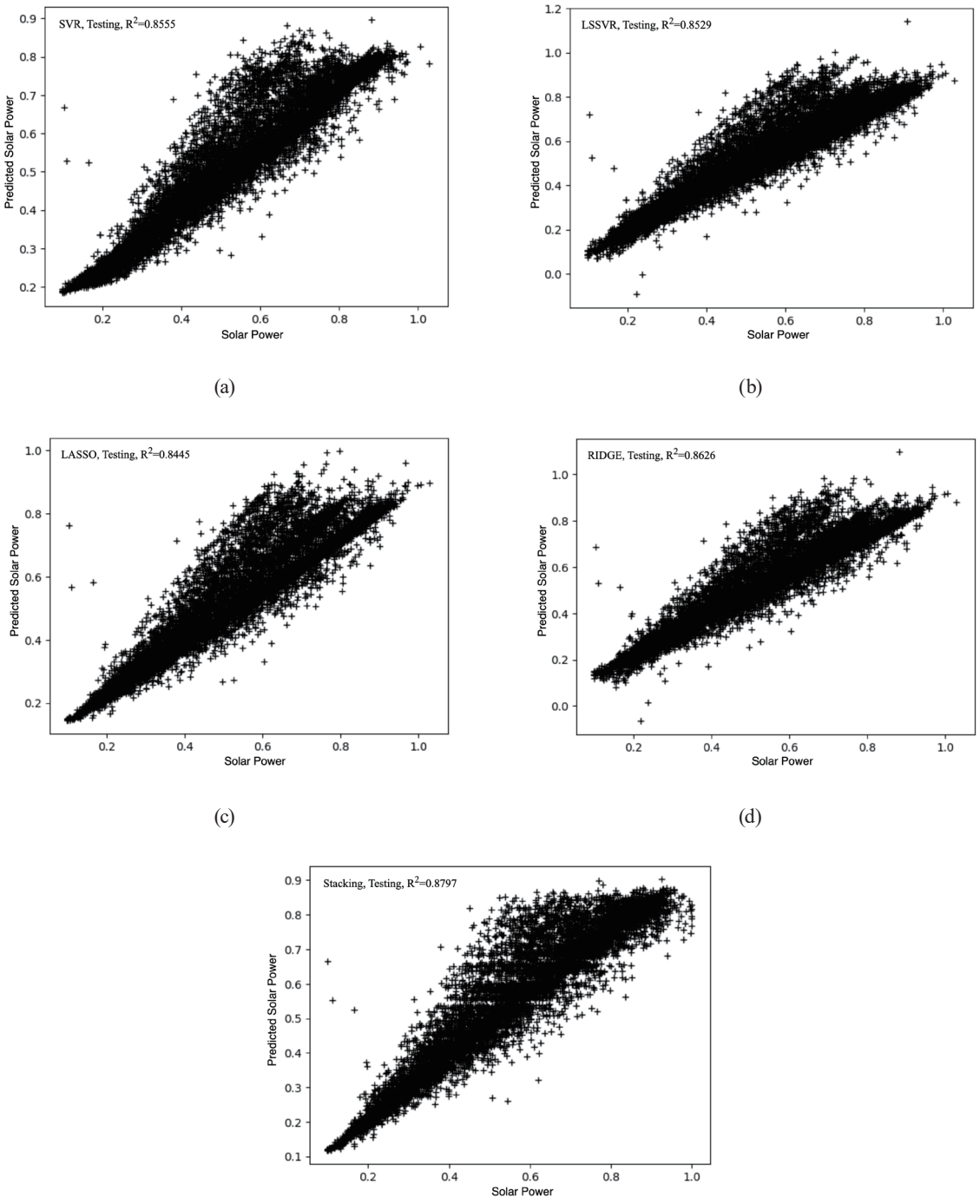


Fig. 8. RGEM stacked generalization model and R^2 of base learners in the model testing phase: (a) SVR, (b) LSSVR, (c) LASSO, (d) RIDGE, and (e) RGEM.

shows $R^2 = 0.8555$ for SVR in model testing, (b) shows $R^2 = 0.8529$ for LSSVR in model testing, (c) shows $R^2 = 0.8445$ for LASSO in model testing, (d) shows $R^2 = 0.8626$ for RIDGE in model testing, and (e) shows $R^2 = 0.8797$ for the stacked generalization model (the proposed RGEM) in model testing.

In the RGEM architecture, the base learners in Level One performed model training and testing by 5-fold CV, and the prediction results using validation sets for model prediction were retained as model training conducted by the meta-model in Level Two. After each base learner completed model training and testing, it carried out model forecasting using the test data kept in the solar database again, and its prediction results were retained to compute its mean for model testing performed by the meta-model in Level Two. In other words, the data evaluated by the base learners in Level One of RGEM became more accurate, and were then used for model training and testing of the meta-model in Level Two, and the final prediction results were output. Therefore, RGEM was able to calculate more accurate forecasting results for solar power generation. Figure 9 shows the residual plot of the final prediction results of RGEM using the solar power generation database of this study. There is no obvious trend or pattern in the residual plot. This means the model has a high goodness of fit, but there are two possible outliers that must be re-evaluated. Figure 10 shows the prediction results of RGEM (comparisons between observed values and predicted values), where $R^2 = 0.8797$, $MSE = 0.0050$, $RMSE = 0.0706$, and $MAPE = 0.0966$.

The benefit of the stacked generalization model is to combine several base learners to make predictions and generate new data. Consequently, if a base learner cannot provide more accurate data, the influence of its error value will be diminished by the data generated by other base learners. In addition, looking upon the stacked generalization model from the point of view of data search, each base learner generates new data in the process of model testing after model training, which means it is possible to avoid finding a local solution. Better predicted values are then provided to the meta-model for model training, giving the meta-model the opportunity to find a global solution, thereby advancing the generalization performance of the final model. In

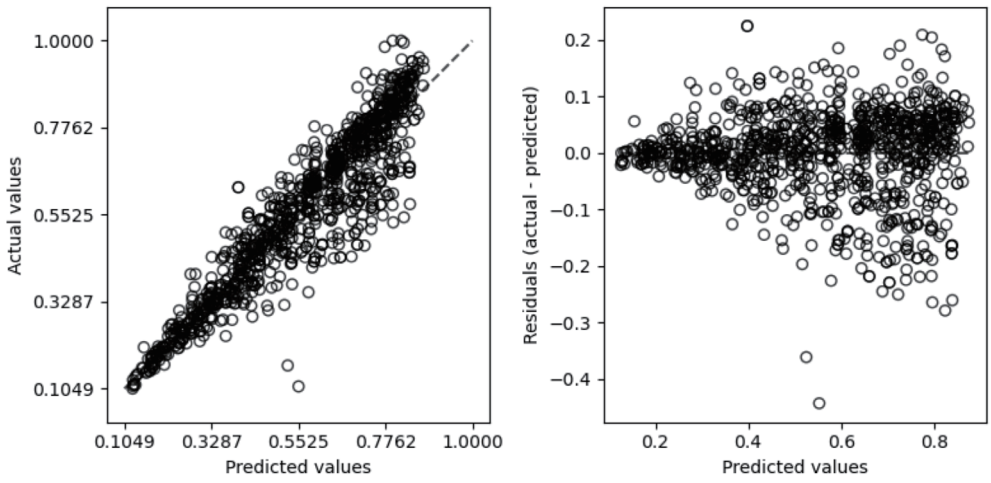


Fig. 9. Residual plot of the final prediction results of RGEM, indicating goodness of fit.

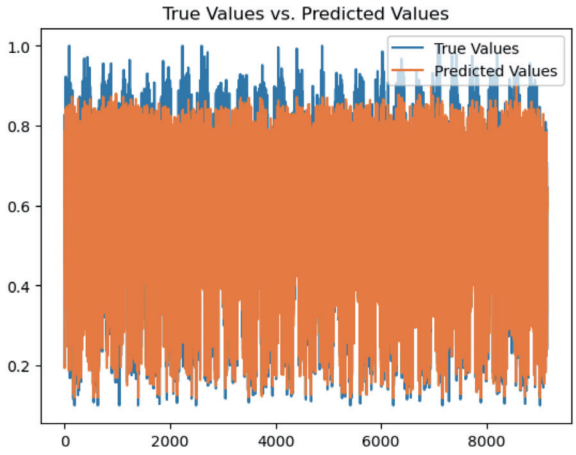


Fig. 10. (Color online) True values and final prediction results of RGEM, where R^2 = 0.8797, *MSE* = 0.0050, *RMSE* = 0.0706, and *MAPE* = 0.0966.

Table 1, the prediction performance of all base learners is good; among them, RIDGE performs the best in the model training and testing phases (training *MAPE* = 0.1075 and testing *MAPE* = 0.1076). After the different kernel functions used by SVR and LSSVR are tested, the results demonstrated that the *MAPE* value of the RBF kernel function is lower than other kernel functions, indicating that the model has a better prediction ability. Thus, the base learners (SVR and LSSVR) in RGEM adopt the RBF function for data conversion. With the help of base learners, the influence of data errors is reduced. Therefore, after combining the advantages of base learners, RGEM can have better prediction performance (training *MAPE* = 0.0916 and testing *MAPE* = 0.0966) as well as reduced prediction error (training *MSE* = 0.0044 and testing *MSE* = 0.0050).We also compared RGEM with other stacking models. For example, Rahimi *et al.*[14] pointed out that, in their research, the *RMSE* prediction error values of the ensemble model using WD-ANN and WD-BCRF were 0.1966 and 0.3212, respectively. In addition, Amarasinghe *et al.*[24] adopted DBN, SVR, and random forest as base learners, used DBN again as a meta-model, and applied 30 weather parameters as input features of the model (e.g., relative humidity, total cloud cover, and solar radiation) through the procedure of feature selection. The experimental results of this study indicated that while the power generation was being predicted in 21 different solar power plants, the *RMSE* prediction error values of the stacking model ranged from 0.0393 to 0.1046, and the *RMSE* average error was 0.0636. However, when different weather factors (e.g., clear, partly cloudy, and overcast) were also considered, the experimental results of this study showed that when the weather factor was “clear”, the average *RMSE* error of the prediction model was 0.0592, which is a significantly reduced forecast error. Consequently, variations in weather factors can be incorporated into the prediction model in future research on solar panel power generation prediction in order to raise the prediction accuracy of the model.

Table 1
Performance of stacking and each base learner.

	Hyperparameters	Training				Testing			
		R^2	*MSE*	*RMSE*	*MAPE*	R^2	*MSE*	*RMSE*	*MAPE*
SVR	kernel: RBF *C* = 10.0 *gamma* = 0.1 *tol* = 0.01	0.8560	0.0056	0.0752	0.1200	0.8555	0.0057	0.0753	0.1202
LSSVR	kernel: RBF *C* = 10.0 gamma = 0.1	0.8531	0.0059	0.0769	0.1097	0.8529	0.0059	0.0770	0.1099
LASSO	*alpha* = 1.0 *tol* = 0.01	0.8446	0.0061	0.0783	0.1193	0.8445	0.0061	0.0784	0.1193
RIDGE	*alpha* = 1.0 *tol* = 0.01	0.8628	0.0054	0.0736	0.1075	0.8626	0.0054	0.0737	0.1076
Stacking (RGEM)	*Loss* = 'squared_error'	**0.8934**	**0.0044**	**0.0665**	**0.0916**	**0.8797**	**0.0050**	**0.0706**	**0.0966**

5. Conclusions

The research topics of renewable energy have been widely valued in various countries, especially in the research field of solar PV power generation. However, the power dispatch between the power generated by traditional power plants and the power generated by solar PV power plants has played a relatively important role in effective power distribution. Efficient solar power dispatch is strongly dependent on accurate solar power generation forecasting techniques. Driven by this knowledge, we was aimed at developing an accurate power forecasting model and a mobile data collection device with IoT sensors. Different regression models were tested, and finally, the architecture of RGEM was built on the basis of the stacked generalization model to boost the forecasting accuracy of power generation. The contribution of this study is in (1) constructing a MDC to collect the data of weather factors using different sensors and developing a RMSP for real-time data display and data storage of solar power generation, and (2) presenting and building a stacked generalization model combining four different base learners for solar power generation prediction. The solar power generation database employed by this study came from the solar power plant of adiCET of CMRU, Thailand. The total power generation of the power plant was 702 kW, and the data of power generation and weather factors were recorded every 15 minutes. When the data preprocessing was carried out at the power plant, null values and abnormal values were found. It was judged that these were generated by bad sensors. Therefore, the values were deleted, and the final number of data sets was 9,147. The RGEM model proposed in this study combined four different base learners, SVR, LSSVR, LASSO, and RIDGE, for collaborative training and prediction. Compared with the traditional linear regression model, SVR has greater robustness in dealing with outlier problems, and LSSVR has better computational efficiency; LASSO pushes unimportant feature coefficients toward 0 via L1 regularization, automatically performs feature selection, and can handle the problem of multi-collinearity. Also, RIDGE improves the stability and generalization performance of the model via L2 regularization and can also deal with the problem of multi-collinearity. As a result,

after integrating the advantages of these four base learners, RGEM had improved overall prediction accuracy and better model generalization performance. In the evaluation and comparison of single prediction models, RIDGE was found to perform the best in the model training and testing phases (training *MAPE* = 0.1075 and testing *MAPE* = 0.1076). The average 15-min-ahead *MSE* error of the RGEM architecture was 0.0011 lower than those of other single prediction models in the model training phase and 0.0008 lower than those of other single prediction models in the model testing stage. Moreover, *MAPE* in both model training and testing phases was less than 0.1, showing that RGEM is a prediction model with high accuracy.

Acknowledgments

The solar power generation database employed in this study is supported by the Asian Development College for Community Economy and Technology (adiCET), Chiang Mai Rajabhat University, Thailand.

References

1 M. Diesendorf and B. Elliston: Renewable Sustainable Energy Rev. **93** (2018) 318. https://doi.org/10.1016/j.rser.2018.05.042
2 International Renewable Energy Agency: https://www.irena.org/publications/2020/Mar/Renewable-Capacity-Statistics-2020 (accessed March 2023).
3 Renewable Energy Policy Network for the 21st Century: https://www.ren21.net/wp-content/uploads/2019/05/GSR2022_Full_Report.pdf (accessed March 2023).
4 R. Ahmed, V. Sreeram, Y. Mishra, and M. D. Arif: Renewable Sustainable Energy Rev. **124** (2020) 109792. https://doi.org/10.1016/j.rser.2020.109792
5 M. N. Akhter, S. Mekhilef, H. Mokhlis, and N. M. Shah: IET Renewable Power Gener. **13** (2019) 7. https://doi.org/10.1049/iet-rpg.2018.5649
6 L. Al-Ghussain, O. Taylan, M. Abujubbeh, and M. A. Hassan: Sol. Energy. **249** (2023) 67. https://doi.org/10.1016/j.solener.2022.11.029
7 A. Djaafari, A. Ibrahim, N. Bailek, K. Bouchouicha, M. A. Hassan, A. Kuriqi, N. Al-Ansari, and E. S. M. El-kenawy: Energy Rep. **8** (2022) 15548. https://doi.org/10.1016/j.egyr.2022.10.402
8 E. Garazi, L. Asier, A. Naiara, and R. Fermin: Energy Sustainable Dev. **68** (2022) 1. https://doi.org/10.1016/j.esd.2022.02.002
9 M. Alshawaf, R. Poudineh, and N. S. Alhajeri: Renewable Sustainable Energy Rev. **134** (2020) 1. https://doi.org/10.1016/j.rser.2020.110346
10 A. Abdulhadi, R. Opoku, C. K. K. Sekyere, S. Boahen, K. O. Amoabeng, F. Uba, G. Y. Obeng, and F. K. ForsonO: Case Stud. Therm. Eng. **35** (2022) 1. https://doi.org/10.1016/j.csite.2022.102133
11 I. Jebli, F. Z. Belouadha, M. I. Kabbaj, and A. Tilioua: Energy **224** (2021) 1. https://doi.org/10.1016/j.energy.2021.120109
12 M. Mishra, P. B. Dash, J. Nayak, B. Naik, and S. K. Swain: Measurement **166** (2020) 1. https://doi.org/10.1016/j.measurement.2020.108250
13 M. Y. Erten and H. Aydilek: Int. J. Eng. Res. Dev. **14** (2022) 333. https://doi.org/10.29137/umagd.1100957
14 N. Rahimi, S. Park, W. Choi, B. Oh, S. Kim, Y. H. Cho, S. Ahn, C. Chong, D. Kim, C. Jin, and D. Lee: J. Electr. Eng. Technol. **18** (2023) 719. https://doi.org/10.1007/s42835-023-01378-2
15 T. C. Carneiro, P. A. C. Rocha, P. C. M. Carvalho, and L. M. Fernandez-Ramirez: Appl. Energy. **314** (2022) 1. https://doi.org/10.1016/j.apenergy.2022.118936
16 M. Aikandari and I. Ahmad: Appl. Comput. Inf. **16** (2020) 1. https://doi.org/10.1016/j.aci.2019.11.002
17 D. Dutta, S. D. Arpan, T. Hossain, S. A. Mamun, D. K. Shah, and S. Mishra: 2022 international Virtual Conference on Power Engineering Computing and Control: Developments in Electric Vehicles and Energy Sector for Sustainable Feture (PECCON, 2022) 1–4. https://doi.org/10.1109/PECCON55017.2022.9851048
18 A. Zendehboudi, M. A. Baseer, and R. Saidur: J. Cleaner Prod. **199** (2018) 272. https://doi.org/10.1016/j.jclepro.2018.07.164

19 B. Kim, D. Suh, M. O. Otto, and J. S. Huh: Remote Sens. **13** (2021) 2605. https://doi.org/10.3390/rs13132605
20 J. Zeng and W. Qiao: Renewable Energy. **52** (2013) 118. https://doi.org/10.1016/j.renene.2012.10.009
21 A. Alfadda, R. Adhikari, M. Kuzlu, and S. Rahman: 2017 IEEE Power & Energy Society Innovative Smart Grid Technology Conference (IEEE, 2017) 1–5. https://doi.org/10.1109/ISGT.2017.8086020
22 A. Fentis, L. Bahatti, M. Mestari, and B. Chouri: 2017 15th IEEE International New Circuits and Systems Conference (NEWCAS, 2017) 405–408. https://doi.org/10.1109/NEWCAS.2017.8010191
23 N. Tang, S. Mao, Y. Wang, and R.M. Nelms: IEEE Internet Things J. **5** (2018) 1090. https://doi.org/10.1109/JIOT.2018.2812155
24 P. A. G. M. Amarasinghe, N. S. Abeygunawardana, T. N. Jayasekara, E. A. J. P. Edirisinghe, and S. K. Abeygunawardane: AIMS Energy. **8** (2020) 252. https://doi.org/10.3934/energy.2020.2.252
25 N. Sharma, M. Mangla, S. Yadav, N. Goyal, A. Singh, S. Verma, and T. Saber: Comput. Electr. Eng. **96** (2021) 1. https://doi.org/10.1016/j.compeleceng.2021.107484
26 C. Persson, P. Bacher, T. Shiga, and H. Madsen: Sol. Energy **150** (2017) 423. https://doi.org/10.1016/j.solener.2017.04.066
27 Z. Guo and G. Bai: Chin. J. Aeronaut. **22** (2009) 160. https://doi.org/10.1016/S1000-9361(08)60082-5
28 H. Wang and D. Hu: 2005 International Conference on Neural Networks and Brain. (ICNNB, 2005) 279–283. https://doi.org/10.1109/ICNNB.2005.1614615
29 L. J. Zhang, X. Y. Wei, J. Q. Lu, and J. H. Pan: Adv. Psychol. Sci. **28** (2020) 1777. https://doi.org/10.3724/SP.J.1042.2020.01777
30 A. M. khan and M. Osinska: Expert Syst. Appl. **212** (2023) 1. https://doi.org/10.1016/j.eswa.2022.118840
31 A. J. Smola and B. Scholkopf: Stat. Comput. **14** (2004) 199. https://doi.org/10.1023/B:STCO.0000035301.49549.88