

Article

# Hybrid Ensemble Deep Learning-Based Approach for Time Series Energy Prediction

Pyae Pyae Phyoo  and Yung-Cheol Byun \* 

Department of Computer Engineering, Jeju National University, Jeju-si 63243, Korea; d6022300211@g.siiit.tu.ac.th

\* Correspondence: ycb@jejunu.ac.kr

**Abstract:** The energy manufacturers are required to produce an accurate amount of energy by meeting the energy requirements at the end-user side. Consequently, energy prediction becomes an essential role in the electric industrial zone. In this paper, we propose the hybrid ensemble deep learning model, which combines multilayer perceptron (MLP), convolutional neural network (CNN), long short-term memory (LSTM), and hybrid CNN-LSTM to improve the forecasting performance. These DL architectures are more popular and better than other machine learning (ML) models for time series electrical load prediction. Therefore, hourly-based energy data are collected from Jeju Island, South Korea, and applied for forecasting. We considered external features associated with meteorological conditions affecting energy. Two-year training and one-year testing data are preprocessed and arranged to reform the times series, which are then trained in each DL model. The forecasting results of the proposed ensemble model are evaluated by using mean square error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). Error metrics are compared with DL stand-alone models such as MLP, CNN, LSTM, and CNN-LSTM. Our ensemble model provides better performance than other forecasting models, providing minimum MAPE at 0.75%, and was proven to be inherently symmetric for forecasting time-series energy and demand data, which is of utmost concern to the power system sector.

**Keywords:** convolutional neural network (CNN); energy consumption; ensemble deep learning; long short-term memory (LSTM); multilayer perceptron; forecasting accuracy; time-series forecasting



**Citation:** Phyoo, P.P.; Byun, Y.-C. Hybrid Ensemble Deep Learning-Based Approach for Time Series Energy Prediction. *Symmetry* **2021**, *13*, 1942. <https://doi.org/10.3390/sym13101942>

Academic Editor: Tomohiro Inagaki

Received: 5 September 2021

Accepted: 11 October 2021

Published: 15 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The energy sector is one of the essential factors in modern society, and thus the required amount of energy between supply and demand should be balanced. As a result, energy forecasting plays a vital role in helping energy manufacturers. Additionally, it is helpful in the improvement of energy management systems, planning, and operation [1,2]. Energy forecasting can be categorized into three groups in term forecasting ranges: one hour–one week, one month–one year, and more than one year, correspondingly [3]. In this paper, short-term hourly energy forecasting is conducted because it is an effectively helpful tool for reducing energy generating and operating costs, ensure power system security, and perform short-term scheduling functions.

According to the benefits mentioned above, many researchers proposed numerous scientific models to achieve better performance on energy forecasting. Generally, forecasting models can be regarded as traditional statistical models and artificial intelligence (AI)-based models. Warren McCulloch and Walter Pitts firstly introduced the foundations of the AI network in 1943 [4]. Since then, AI-based machine learning (ML) models have been widely used in medicine, business, communications, and industrial process control as nonlinear time series problems can be solved. Nevertheless, deep learning (DL) models were established to handle the weakness of ML models. For instance, the training process of ML models could cost a longer computational time during the backpropagation process if there were multiple layers in the network. Moreover, there is no interconnection between

each layer in the traditional ML that causes the lack of information for time series data. Multilayer perceptron (MLP) [5], convolutional neural network (CNN) [6], and long short-term memory (LSTM) [7] are proposed to solve ML weakness for time series forecasting. Consequently, this article was applied these DL models to implement and improve forecasting accuracy based on energy time series data. Additionally, time series forecasting effectively supports demand management in the electric industry.

Naturally, complex seasonality patterns are exhibited in such time series data. For example, seasonal, calendar, and weather effects significantly influence energy consumption [8]. Firstly, seasonal effects involve four seasons: spring, summer, fall, and winter in Jeju. Secondly, calendar effects consist of day type, month, public holiday, and national holiday. Finally, weather effects are typically associated with meteorological circumstances such as temperature, cloudy, and humidity [9]. As a result, this study primarily considers the weather information, including temperature, dew point, humidity, wind speed, solar radiation, and other factors for better understanding the non-linear relationship between load patterns and influential variables that can enhance energy forecasting performance.

This research mainly uses the ensemble model of the latest advanced DL models for time series energy forecasting. Hence, four advanced DL models such as MLP, CNN, LSTM, and hybrid CNN-LSTM are applied and implemented for forecasting. These models can effectively handle the time series data by memorizing all sequences during the training process. This work compares the proposed ensemble model with the standard baseline forecasting models by preserving the feature engineering. Three error metrics, including mean square error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), are evaluated to make a performance comparison among all models.

### 1.1. Related Work

Lim and Zohren surveyed on developing DL architectures and hybrid DL models combining well-studied statistical models with neural network components for time series forecasting [10]. Furthermore, Langkvist et al. reviewed the development of DL and unsupervised feature learning for time series problems by suggesting ML and DL algorithms [11]. Compared to ML models, the experiment of DL algorithms provides the capability to cope with nonlinear relationships, model complexity, and computational efficiency. The clarification review of DL algorithms such as deep neural network (DNN), unsupervised learning, and reinforcement learning from prior works was conducted by Schmidhuber [12]. In the article of Hosein, the better results of DNN were compared with those of ML algorithms using periodic smart meter energy for short-term load forecasting [13]. The application of the multilayered DNN training along with different activation functions sigmoid, rectifier linear unit (ReLU), and exponential linear unit (ELU) to the Iberian electric market forecasting were investigated in work by Hossen et al. [14].

In the study of Cai et al., DL models including the recurrent neural network (RNN) and CNN models were compared with an autoregressive integrated moving average with exogenous inputs (ARIMAX) regarding the forecasting accuracy, computational efficiency, generalizability, and robustness [15]. Moreover, the effectiveness of the CNN algorithm was investigated by comparing experimental results to other ML and DL algorithms for energy load forecasting [16]. The LSTM prediction models with various configurations were constructed using France metropolitan's electricity consumption data for short- and medium-term load forecasting. They compared the relevant performance to ML models [17]. Besides, LSTM multi-input, multioutput models considering long-term historical dependencies for the cluster analysis of load trend were trained and opposed the performance to ML models [18]. Kong et al. also implemented the LSTM model on residential smart meter data, which outperformed backpropagation neural network (BPNN) in the task of short-term load forecasting [19]. These prior works provided reasonable and good forecasting results using the latest popular DL models.

On the other hand, considering the benefits and drawbacks of ML and DL algorithms, various studies were employed either a hybrid method or an ensemble method

to execute more reliable and accurate forecasting outcomes [20–23]. The incorporation of an artificial neural network (ANN), a BPNN, a generalized regression neural network (GRNN), an Elman neural network, and a genetic algorithm optimized backpropagation neural network (GABPNN) was proposed for half-hourly electrical power prediction by Xiao et al. [24]. The hybrid algorithm, so-called SDEMD-LSTM, combining similar days (SD) selection, empirical mode decomposition (EMD), and LSTM networks, was conducted by Zheng et al. [25]. Their study evaluated the similarity between forecast and historical days by using the extreme gradient boosting-based weighted k-means algorithm. Furthermore, the ensemble method, including the EMD algorithm and the deep belief network (DBN) trained with two restricted Boltzmann machines (RBMs), were performed on electricity load demand [26]. This effectiveness experiment of the proposed EMD-based DBN model outperformed nine other forecasting methods in comparing simulated results. In the article by Zhang and Wang, another decomposition-ensemble method composed of SSA, SVM, ARIMA, and CS algorithms, was integrated for load forecasting. Their empirical outcomes ensured the importance of an ensemble model based on data input structure [27]. Regarding these prior works, the ensemble method combining two or more forecasting models could be essential to improve the forecasting performance in many areas, such as forecasting of the daily average number of COVID-19 patients, bitcoin price forecasting, household load forecasting, and typhoon formation forecasting [28–31].

### 1.2. Contribution

For the first time, this work handles time-series problems using the three latest advanced DL models and the hybrid CNN-LSTM model while considering the tuning parameters based on collected data. The ensemble combination of these four advanced DL models is a novelty proposed to improve energy forecasting performance. For better training, all DL models also consider weather features that affect energy consumption.

### 1.3. Paper Organization

This paper is organized using the following sections. The methodology, including DL architectures and the main framework of our proposed model, are described in Section 2. The results are then discussed and compared in Section 3. The conclusion of the complete work is written in the last section.

## 2. Methodology

In this section, deep neural networks-based models such as MLP, CNN, LSTM, hybrid CNN-LSTM, and our proposed ensemble model are described with a detailed explanation of their architectures and parameters applied in this research.

### 2.1. Multilayer Perceptron (MLP)

Like an ordinary neural network, an MLP has an input, output, and hidden layer. However, it can have many layers in its training process. There are two training processes: feedforward propagation and backward propagation in the MLP network. In the former approach, neuron nodes for input features are multiplied with respective weights and biases to execute the corresponding output values passing through nonlinear activation functions of all invisible layers. The latter process adjusts the weights to minimize the loss using backpropagation gradient descent after estimating the target and calculating loss in the forward direction. The structure of the MLP network training process is shown in Figure 1. The mathematical expression of the MLP is written as:

$$y_m = f\left(\sum_i^m w_{im}y_l + w_m\right) \quad (1)$$

where,  $y_m$  = the predicted output at the  $m$ th output layer,  $y_l$  = the output at the  $l$  hidden layer,  $w_{lm}$  = the weight between the  $l$ th hidden layer and the  $m$ th output layer,  $w_m$  = the

weight at the  $m$ th output layer,  $f =$  activation function,  $i =$  the input layer,  $m =$  the output layer.

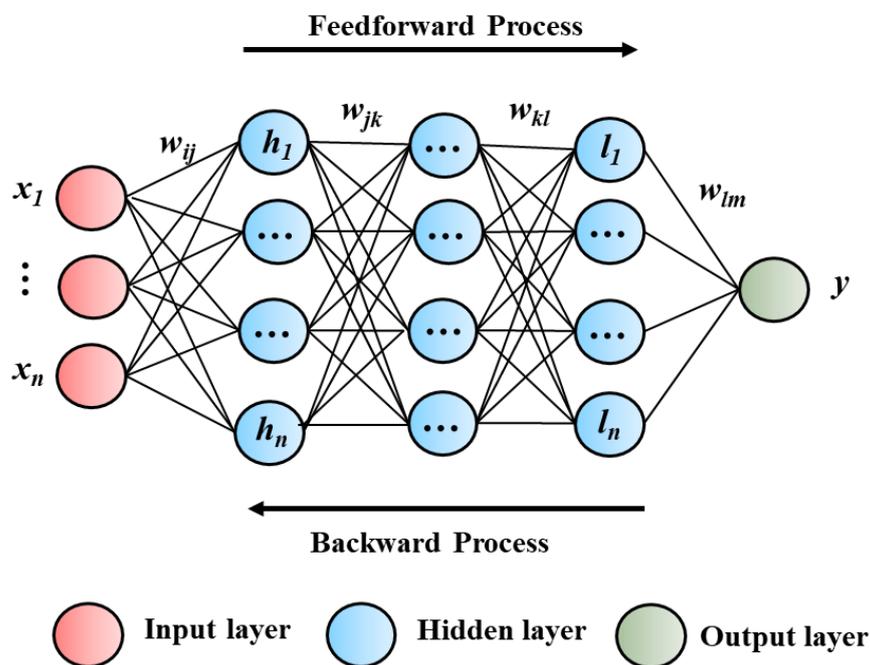


Figure 1. Basic structure of MLP network [5].

In this paper, MLP is firstly used as one of the DL models to execute our proposed ensemble DL model. Figure 2 indicates our proposed MLP model with one input layer, one hidden layer, and one output layer. The formation of all inputs is used as sequential data. Thus, the MLP network has the input for each sample in terms of the number of time steps. Ten sequential training inputs are initially loaded into the MLP training model. Afterward, the model is fitted with 100 dense layers and a Rectified Linear Unit (ReLU) activation function. Finally, the model predicted results based on the test data at the last dense layer.

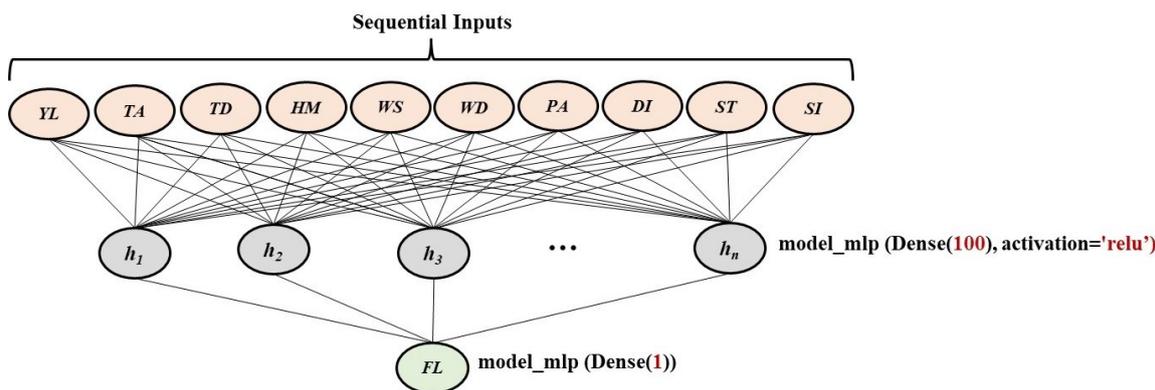


Figure 2. MLP model.

### 2.2. Convolutional Neural Network (CNN)

The second proposed DL model is CNN architecture which contains several layers, so-called multibuilding blocks. The detail of each layer in the CNN architecture is described in Figure 3. Firstly, the sequential data are imported to a convolutional layer convolved as a one-dimensional structure with 64 filters, two kernel sizes, and the ReLU activation function to generate the output feature map. The next layer is a pooling layer which shrinks large-size feature maps to create smaller feature maps. The one-dimensional maximum (max) pooling with two pool sizes is applied in our case. After that, the data are converted

into a one-dimensional array by a flattening layer. It is imported to the fully connected layer of the CNN model, having fifty dense layers and the ReLU activation function. Ultimately, the final layer executes the output on test prediction.

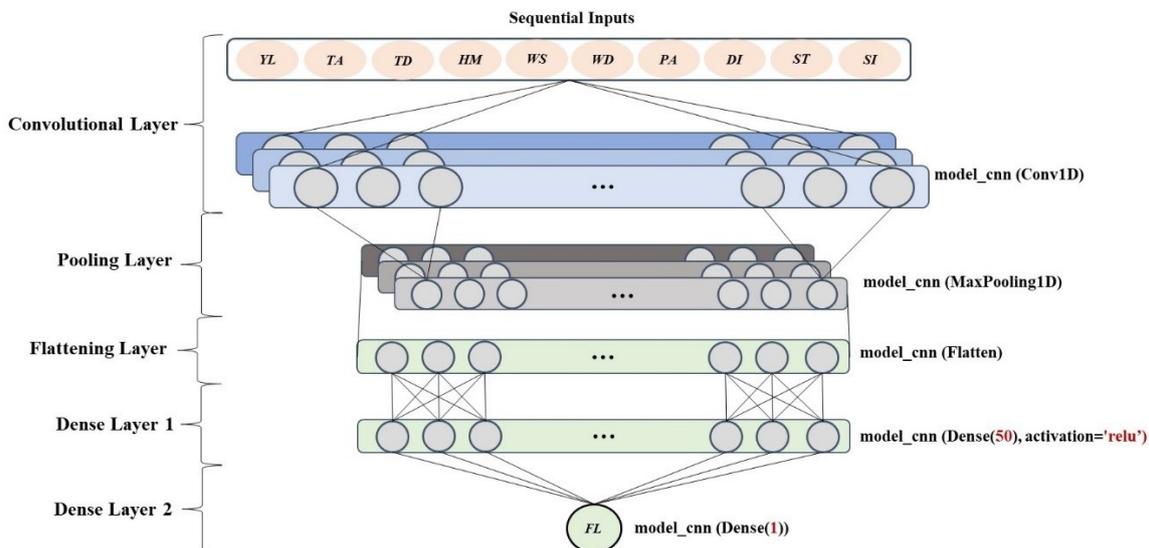


Figure 3. CNN model [6].

### 2.3. Long Short-Term Memory (LSTM)

The third DL model is the LSTM network which consists of a set of recurrently connected subnets, known as memory blocks. Each memory block includes a memory cell, input gate, forget gate, and output gate. Unlike the traditional recurrent unit, which overwrites its content at each step, the LSTM unit can decide whether to keep the existing memory via the introduced gates. LSTM avoids the long-term dependency problem explicitly. There are four interacting layers in the LSTM architecture instead of having a single neural network layer in a recurrent neural network. Figure 4 shows the structure of LSTM where each line carries an entire vector, from the output of one node to the inputs of the other.

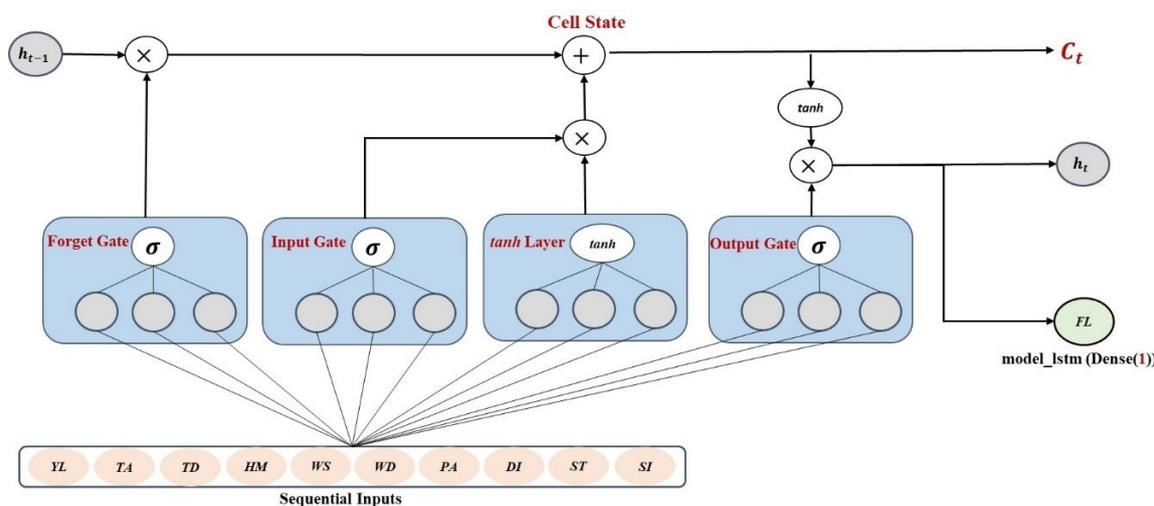


Figure 4. LSTM model [7].

Initially, the forget gate (sigmoid) layer takes an input that is needed to be kept and the previously hidden layer ( $h_{t-1}$ ) to give an output in the cell state. Afterward, an input gate (sigmoid) layer updates the input value, and then multiplies it with a tanh layer, creating a vector of new candidate values. The new cell state is then executed by combining the old

state and a new candidate value. Next, an output gate (sigmoid) layer performs an output using the new cell state passing through the tanh layer. Finally, the desired result is filtered by multiplying the output layer and the tanh layer. In this study, LSTM is implemented with the ReLU activation function.

### 2.4. Hybrid CNN-LSTM

A hybrid CNN-LSTM model is our last DL ensemble method. Very long input sequences can be handled as blocks or subsequences as the hybrid model contains both CNN and LSTM models. In this case, our sequential data are divided into further subsequences for each sample to train the hybrid model. A hybrid structure of CNN and LSTM models is represented in Figure 5. Primarily, the CNN model interprets each subsequence of sequential inputs. In this case, the CNN model is enveloped in Time Distributed wrapper layers of convolution, pooling, and flattening. Hereafter, the results are assembled by the LSTM layer before making a test prediction. The parameters of the hybrid model are adjusted in the same manner as stand-alone CNN and LSTM models.

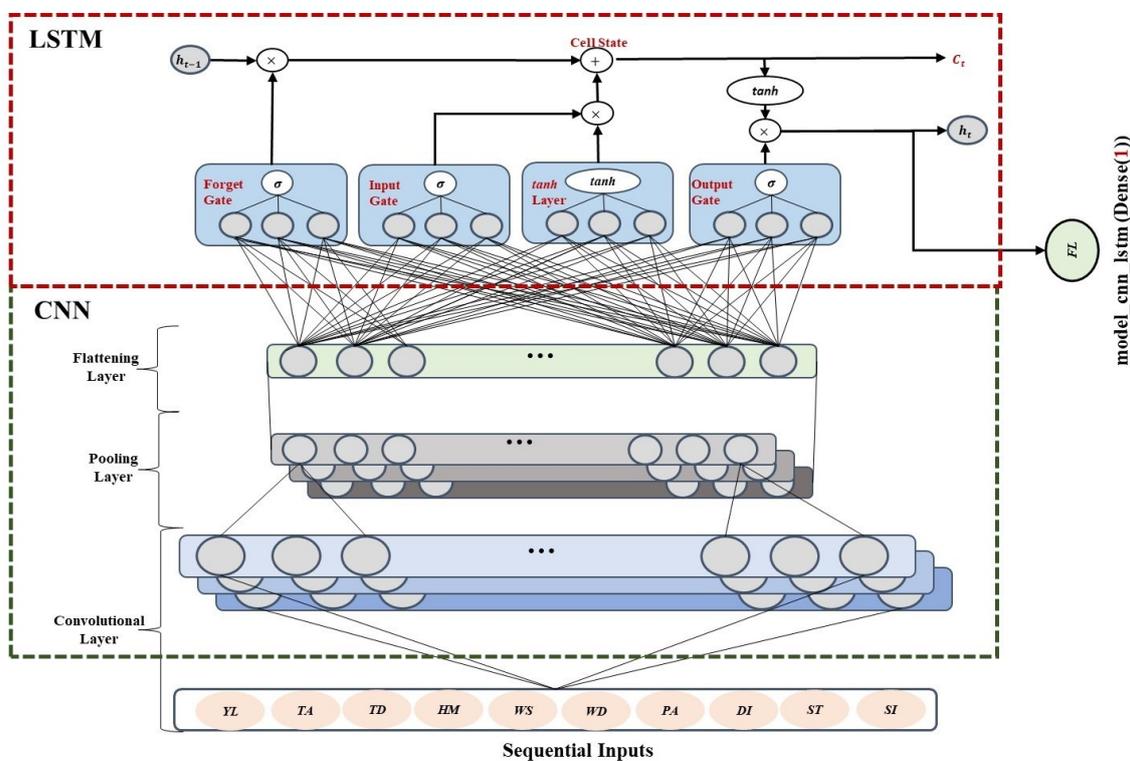


Figure 5. Hybrid CNN-LSTM model.

### 2.5. Ensemble Deep Learning Model

#### 2.5.1. Data Management

The hourly-based energy consumption data are gathered from four weather stations such as Jeju-si, Gosan, Seongsong, and Seogwipo of Jeju Island. The total load data of the whole Jeju are recorded from 2012 to June 2019, as indicated in Figure 6. It is obvious that energy consumption increases year by year due to the development of population and industries. Jeju has four seasons: spring from March to May, summer from June to August, fall from September to November, and winter from December to February. Generally, energy usage rises during summer when the peak load has occurred. Similarly, people consume higher electricity to keep warm in the winter season. During the spring and fall seasons of the specified years, the usage of energy varies approximately from 500 to 900 MW.

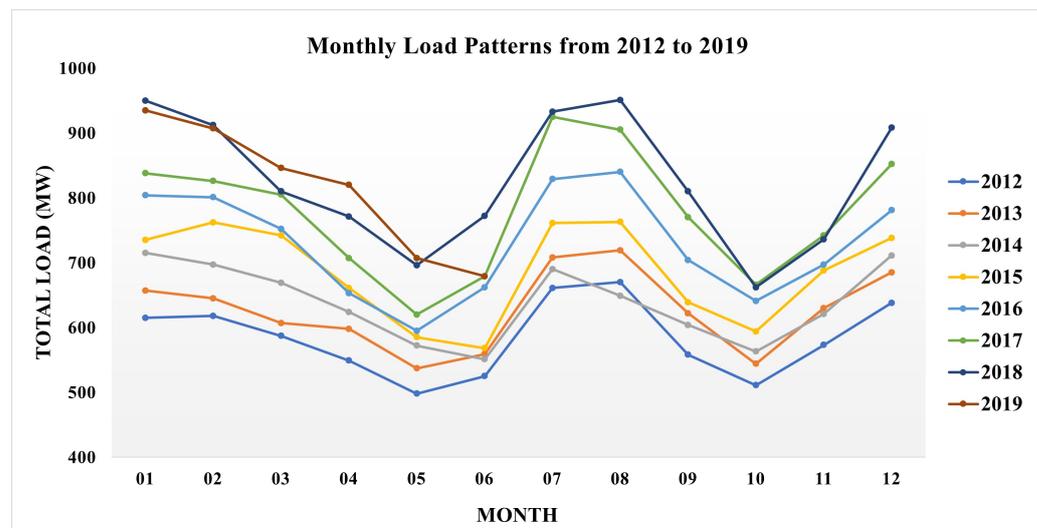


Figure 6. Monthly load patterns from 2012 to 2019.

Each station also collects the weather information separately. Collected hourly-based weather features are average temperature (TA), dew point temperature (TD), humidity (HM), wind speed (WS), wind direction degree (WD), atmospheric pressure on the ground (PA), discomfort index (DI), sensible temperature (ST), and solar irradiation quantity (SI). Our earlier work has widely explained the correlation between total load consumption and weather features from each station [20]. Some extra features are also added in this work, as demonstrated in Table 1. The table shows that positive values mean a strong correlation between two features, while negative values indicate less correlation. Therefore, the most affecting factor on load is the PA feature from all stations, showing around 0.23. The WS feature is ranked second at Jeju-si and Gosan, whereas WD is for Seongsan and Seogwipo. The WD is another influential feature on load for all stations, except Gosan. The rest features are given as negative correlation so that the impact of external factors is meager.

Table 1. Correlation between total load and weather features for each station.

	TA	TD	HM	WS	WD	PA	DI	ST	SI	Weather Station
Total Load	−0.18	−0.20	−0.16	0.14	0.13	0.23	−0.18	−0.21	0.02	Jeju-si
	−0.20	−0.18	−0.08	0.16	0.00	0.23	−0.18	−0.24	−0.06	Gosan
	−0.21	−0.23	−0.20	0.07	0.09	0.23	−0.20	−0.24	−	Seongsan
	−0.22	−0.18	−0.03	−0.14	0.08	0.22	−0.20	−0.23	−	Seogwipo

In this work, our dataset was collected from four geographic parts of Jeju (including Jeju-si, Seogwipo, Gosan, and Seongsan). Each region had its weather information containing nine features and the total load over the grid, and we used percentage-based division and aggregation approaches to reduce the dataset. The preprocessing method considered 50% of data from Jeju-si as it is a highly-populated area, and it matters the most. Similarly, we utilized 30% from Seogwipo, 10% from Gosan, and 10% from Seongsan. Therefore, ten sequential inputs, including load and nine weather features, are applied to train all DL models. The following expression is the vector of input features.

$$FL_t = [YL_t, TA_t, TD_t, HM_t, WS_t, WD_t, PA_t, DI_t, ST_t, SI_t] \quad (2)$$

where,  $FL_t$  = the forecasted load vector of input features,  $YL_t$  = the yesterday load at time  $t$ ,  $t = 1, 2, \dots, 24$ .

### 2.5.2. Set Up Parameters

Data are separated into two sets: training and testing sets for all DL models. The training set is arranged from June 2016 to May 2018, while the testing set is employed from June

2018 to May 2019. The whole training set is applied to predict the value at the next hour at each time step, and this process continues until the end of the testing set. All sequential input data must be reshaped into the form of time series data based on each DL model. The parameters for all proposed models are suggested in Table 2. The proposed time series forecasting is evaluated on a desktop with specifications 11th Gen Intel Core i7 5.00 GHz processor, 16 GB RAM, 64-bit operating system, x64-based processor, fully loaded with Jupyter Python Language programming on Google Colab.

**Table 2.** Parameters for all trained models.

Parameters	MLP	CNN	LSTM	CNN-LSTM	Proposed Model
Input shape	[samples, timesteps]	[samples, timesteps, features]	[samples, timesteps, features]	[samples, subsequences, timesteps, features]	Based on each DL model
Epochs			50		
Batch size			256		
Learning rate			0.0003		
Optimizer			Adam		
Activation function			ReLU		
Loss function			MSE		

### 2.5.3. The Proposed System

The framework of the hybrid ensemble deep learning-based energy forecasting mechanism is represented in Figure 7. A detailed explanation of the six steps in the proposed system is described as follows:

#### Step 1: Data collection

The energy data and weather information are collected hourly from four regions in Jeju Island. The weather data are averaged according to the portion of each region. Therefore, this study uses the total load and total weather data of the whole Jeju.

#### Step 2: Data preprocessing

Some information is missing in our raw data, so we cleaned the original data using the specific data average. After cleaning and removing missing values, we selected ten inputs based on the data correlation and then arranged the input data.

#### Step 3: Data Splitting

The arranged data are split into two portions: training and testing. Two-year training from June 2016 to May 2018 and one-year testing from June 2018 to May 2019 are utilized for training and testing all proposed models.

#### Step 4: Training of forecasting module

After we get the training and testing sets, training data are given to each DL model. Before training, input data are transformed into sequential data to feed as sequencers and converted into a supervised learning format. Therefore, the sequential data are used for all proposed DL models. Four DL models such as MLP, CNN, LSTM, and hybrid CNN-LSTM are then defined and fitted in the training data. The parameters mentioned above in the last subsection are used for building these models. Successively, these four models are ensemble to make ensemble predictions.

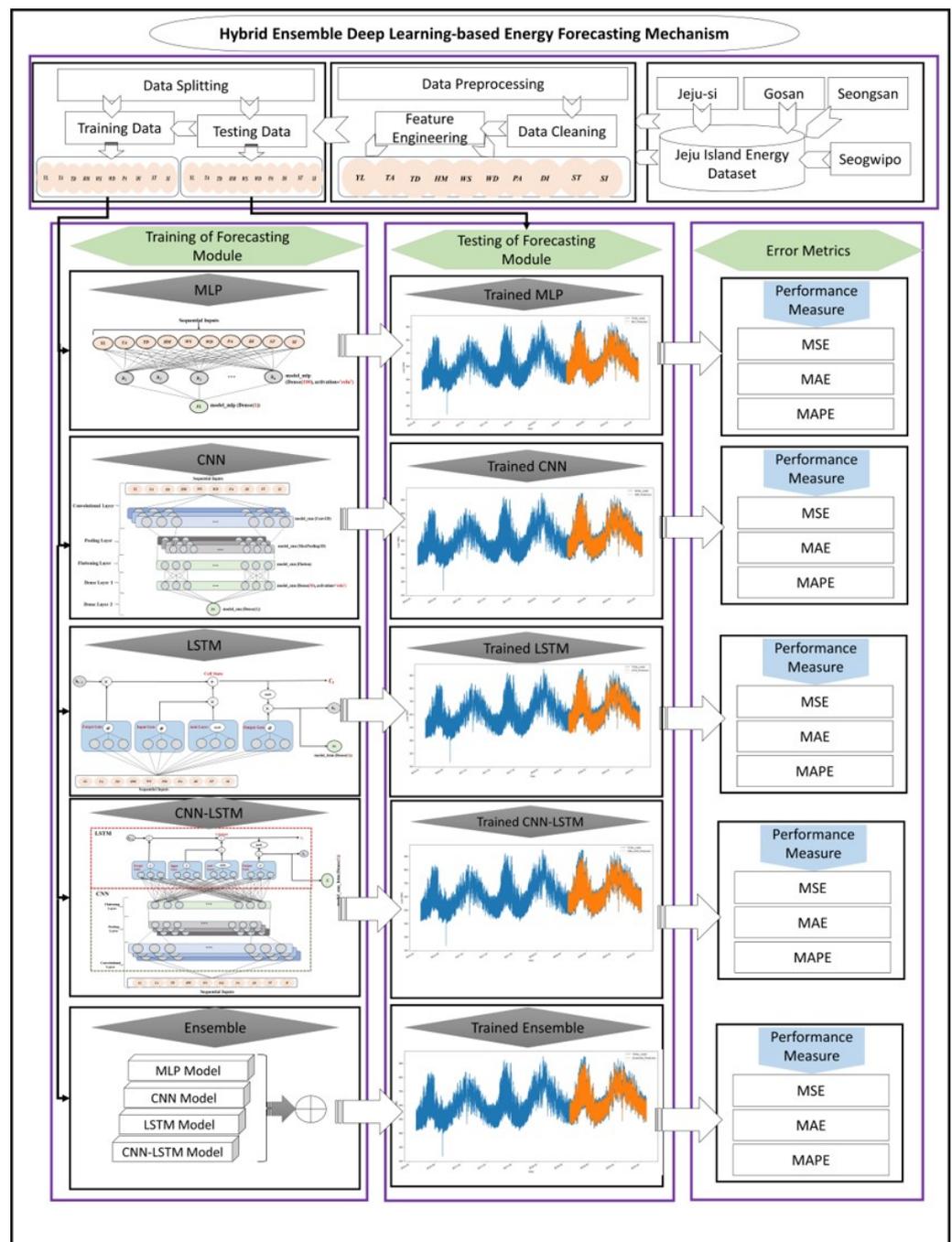


Figure 7. Framework of proposed ensemble model.

**Step 5: Testing of forecasting module**

Testing data is provided to the trained models once learned from the training data. Afterward, predictions on the testing data are completed for each DL model and the proposed ensemble model.

**Step 6: Error Measurement**

The last phase measures the forecasting performance that indicates how much forecast value differs from the corresponding observation. To do so, we select three error metrics, including mean squared error (MSE), mean absolute error (MAE), and mean absolute

percentage error (*MAPE*), which are commonly used for measuring the forecasting accuracy. The mathematical formulas for these error metrics are expressed as:

$$MSE = 1/t \sum_{t=1}^{24} (AL_t - FL_t)^2 \quad (3)$$

$$MAE = 1/t \sum_{t=1}^{24} |AL_t - FL_t| \quad (4)$$

$$MAPE = 1/t \sum_{t=1}^{24} |AL_t - FL_t| / |AL_t| \quad (5)$$

where,  $AL_t$  = an actual energy at time  $t$ ,  $FL_t$  = a predicted energy at time  $t$ ,  $t = 1, 2, \dots, 24$ .

### 3. Experimental Results

The generated results between the proposed ensemble model and other DL stand-alone models are compared monthly error metrics on test predictions from June 2018 to May 2019. Monthly MSE, MAE, and MAPE are evaluated to make a comparison among all models, as revealed in Tables 3–5, respectively. In general, our proposed model outperforms other DL models, showing 1472.76 MSE, 28 MAE, and 4.15 MAPE. All models provide reasonable forecast values, with errors varying from 3% to 5% of MAPEs in all months, except February. Our proposed DL ensemble model is ranked first, showing better accuracy in June, August, September, March, and April, followed by the MLP model, which gives better performance in October and December. The hybrid CNN-LSTM and CNN models execute almost similar percentages, at 4.26% and 4.27% in the total average of test forecasts. The LSTM model is ranked last with the average 4.34% MAPE, 29.13 MAE, and 1459.68 MSE, while there are better accuracies of LSTM in July and February.

Four different groups consisting of weekdays, weekends, Mondays, and holidays are divided to further MAPE comparison among all models. Table 6 is referred to the comparison of average MAPEs of each category for both proposed and baseline models. In general, the group on weekdays performs lower MAPE than other groups, with around 3% in all models. Considering the weekday category, the proposed model provides the lowest MAPE with 3.39% among all DL models. It also generates lower MAPE than others in the group of weekends, amounting to 4.04%. However, the ensemble model contributes approximately 0.1% higher than the LSTM model in the groups of Mondays and holidays.

**Table 3.** Monthly MSE comparison between proposed model and baseline models.

	MLP	CNN	LSTM	CNN-LSTM	Proposed Model	Count Holidays
June	818.21	922.69	741.19	826.75	734.45	2
July	1869.12	1581.35	1281.58	1975.15	1558.18	
August	2287.82	1828.29	1638.69	2260.11	1848.72	1
September	1335.28	1298.10	1134.87	1408.95	1227.61	4
October	606.81	660.69	687.55	596.11	604.30	2
November	704.69	713.32	817.32	679.21	699.32	
December	1596.63	1653.35	1659.93	1710.96	1600.28	2
January	2374.31	2275.50	2320.60	2315.49	2287.95	1
February	3811.47	3686.27	3258.94	3705.23	3563.36	3
March	1779.04	1815.36	1834.50	1812.24	1745.12	2
April	1164.04	1151.37	1276.48	1200.46	1144.15	
May	792.16	810.01	978.31	764.01	794.73	4
Average	1583.21	1520.93	1459.68	1593.60	1472.76	

**Table 4.** Monthly MAE comparison between proposed model and baseline models in MW.

	MLP	CNN	LSTM	CNN-LSTM	Proposed Model	Count Holidays
June	19.87	22.02	20.46	20.57	19.25	2
July	32.05	29.70	27.79	32.85	29.38	
August	34.05	31.38	32.86	33.85	31.34	1
September	26.02	26.91	26.21	27.16	25.70	4
October	17.18	19.37	20.16	17.44	17.84	2
November	18.72	19.51	21.97	18.42	19.15	
December	31.05	31.80	32.33	32.80	31.48	2
January	36.94	35.89	37.43	36.02	36.26	1
February	47.21	46.35	44.04	46.07	45.57	3
March	32.28	32.55	33.16	32.17	31.90	2
April	26.81	26.70	28.71	26.99	26.60	
May	22.22	22.71	25.30	21.96	22.52	4
Average	28.61	28.65	29.13	28.78	28.00	

**Table 5.** Monthly MAPE comparison between proposed model and baseline models in percent.

	MLP	CNN	LSTM	CNN-LSTM	Proposed Model	Count Holidays
June	3.28	3.65	3.39	3.44	3.18	2
July	4.38	4.11	3.94	4.46	4.04	
August	4.37	4.15	4.45	4.30	4.07	1
September	4.21	4.37	4.27	4.41	4.17	4
October	3.00	3.39	3.47	3.03	3.09	2
November	3.10	3.23	3.62	3.05	3.16	
December	4.43	4.55	4.62	4.70	4.50	2
January	4.90	4.76	4.93	4.78	4.80	1
February	6.58	6.46	6.10	6.45	6.35	3
March	4.73	4.74	4.79	4.69	4.64	2
April	4.24	4.21	4.47	4.28	4.19	
May	3.73	3.79	4.17	3.69	3.75	4
Average	4.23	4.27	4.34	4.26	4.15	

**Table 6.** MAPE comparison of four different groups for all models in percent.

	MLP	CNN	LSTM	CNN-LSTM	Proposed Model
Weekdays	3.42	3.52	3.78	3.47	3.39
Weekends	4.30	4.09	4.07	4.16	4.04
Monday	7.11	7.42	7.08	7.40	7.21
Holidays	4.61	4.63	4.38	4.63	4.47

The actual energy versus forecast energy of the proposed model on testing data is indicated in Figure 8. The forecast of the proposed model closely follows the actual energy. The  $x$ -axis of the figure represents the hourly period from June 2018 to May 2019, whereas the  $y$ -axis indicates the load in the megawatt units (MW). The blue line shows the actual energy, whereas the red line is indicated as the forecast results. As shown in the figure, there is a big gap between actual and forecast in July, ranging in time from 721 to 1440 due to the high temperature. Apart from this, the proposed model predicts the actual load accurately.

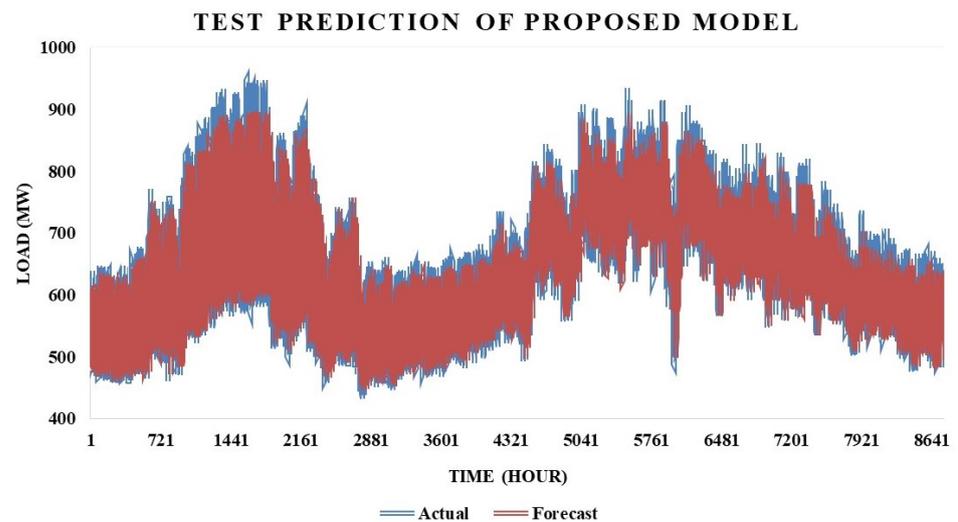


Figure 8. Comparison between actual and forecast values by proposed ensemble model.

To the extent of better visualization in predicted results, the best predicted week and the best predicted day from test predictions are visualized in Figures 9 and 10, correspondingly. Both figures have a primary  $x$ -axis in an hour, a  $y$ -axis in MW, and a secondary  $y$ -axis in the percentage of MAPE. Regarding lines in both figures, blue represents the actual load, while red is the forecasted MW load. Moreover, another green line refers to the MAPE measurement. The best predicted week is the second week of October 2018, as seen in Figure 9. The load fluctuations between actual and forecast are almost similar, ranging from 400–640 MW. The proposed model conducts under forecast, which is suitable for forecasting. Although there are high errors for some points on 8 October 2018, the model predicts better values for the other days of the week. MAPEs vary from 1.24% to 1.64%, except 6.23% on October 8 and 3.28% on October 10.

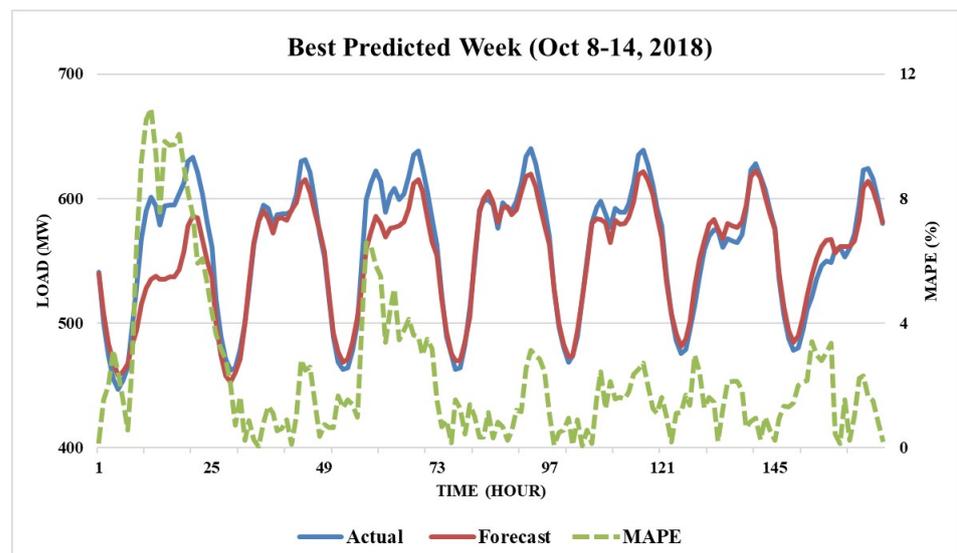


Figure 9. Best predicted week by proposed ensemble model.

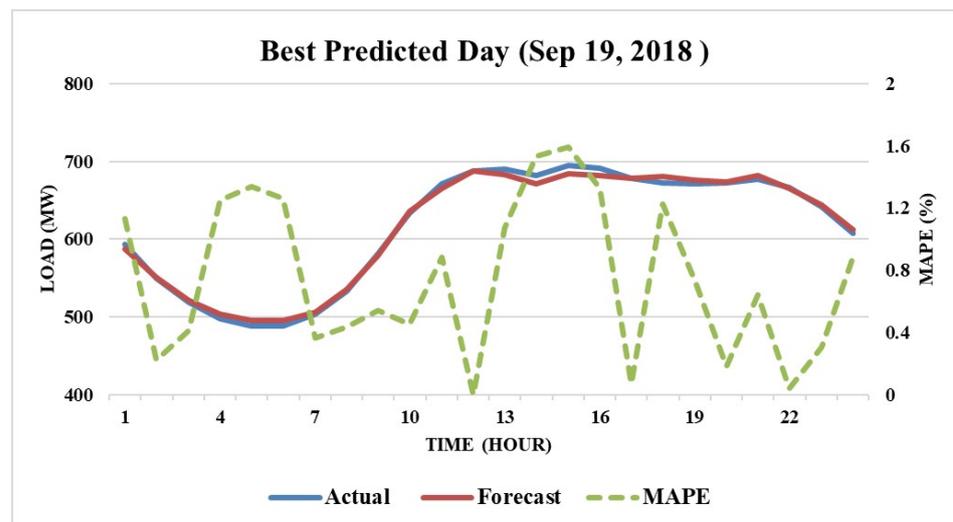


Figure 10. Best predicted day by proposed ensemble model.

Similarly, the best predicted day is selected to realize how much minimum error the proposed model can forecast. Like Figure 9, three lines show actual, forecast, and MAPE in Figure 10. The minimum MAPE at 0.75% is executed on 19 September 2018. The errors range from 0.001% at 11 AM to 1.593% at 2 PM in a day. The pattern of forecast values fluctuates similarly to that of actual values. Therefore, our proposed ensemble model is a promising model providing an acceptable and reliable result for time series prediction.

For further comparison, we chose the existing paper conducted similar to our work and compared our results and their results [32]. Their research proposed the stochastic ensemble model framework formulated as a two-stage random forest problem with a series of homogeneous prediction models. In their study, three load consumption datasets such as the Korean Electric Power Company (KEPCO) substation building, the Korean Research Institute (KEPRI) building, and testbed were applied to train their proposed ensemble model. However, we select the MAPE results from their first two datasets to compare with our results. Table 7 indicates the seasonal MAPE comparison between their proposed model and our proposed model. According to previous work, MAPEs for each season are computed to make further comparisons. Our proposed DL ensemble model outperforms the cited proposed ensemble model by providing 4.20% in Spring, 3.77% in Summer, 3.48% in Fall, and 5.22% in Winter. Nevertheless, we can reveal that the proposed ensemble model from both works could predict the load consumption as accurately as the actual energy by comparing it with other stand-alone forecasting models.

Table 7. MAPE comparison between existing research and our proposed model in percent.

	Ensemble Model Proposed in Citation		Proposed Model
	KEPCO Dataset	KEPRI Dataset	Jeju Dataset
Spring	17.09	7.34	4.20
Summer	16.46	7.52	3.77
Fall	22.02	5.63	3.48
Winter	17.57	10.25	5.22

#### 4. Conclusions

This research mainly contributes to the ensemble DL model with the combination of four architectures: MLP, CNN, LSTM, and hybrid CNN-LSTM. As all these algorithms are suitable for time series forecasting, hourly-based energy data are applied in this paper. Four regions provide the collected energy data in Jeju Island, South Korea. Moreover, weather information is attached and used as external features. Both energy and weather data are transformed into time series data to train all DL models. The training set from June 2016 to

May 2018 is applied to predict the testing set from June 2018 to May 2019. All forecasting results of the proposed model are evaluated using three error metrics. Consequently, these are compared with individual DL algorithms. The ensemble model outperforms other DL models, with an average MAPE of 4.15% on the whole test. Our model is an acceptable and promising model that provides the relevant forecasting results in the forecasting field using another dataset.

**Author Contributions:** Conceptualization, P.P.P.; methodology, P.P.P.; software, P.P.P.; validation, P.P.P.; formal analysis, P.P.P.; investigation, P.P.P.; writing—original draft preparation, P.P.P.; writing—review and editing, Y.-C.B.; visualization, P.P.P.; supervision, Y.-C.B.; funding acquisition, Y.-C.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** Following are the results of a study on the “Leaders in Industry-University Cooperation and Project”, supported by the Ministry of Education and National Research Foundation of Korea. Moreover, this work was supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0016977, The Establishment Project of Industry-University Fusion District).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Barić, I.; Grbić, R.; Nyarko, E.K. Short-Term Forecasting of Electricity Consumption Using Artificial Neural Networks-an Overview. In Proceedings of the 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 20–24 May 2019; pp. 1076–1081.
- Waqas Khan, P.; Byun, Y.C.; Lee, S.J.; Park, N. Machine learning based hybrid system for imputation and efficient energy demand forecasting. *Energies* **2020**, *13*, 2681. [[CrossRef](#)]
- Goia, A.; May, C.; Fusai, G. Functional clustering and linear regression for peak load forecasting. *Int. J. Forecast.* **2010**, *26*, 700–711. [[CrossRef](#)]
- McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [[CrossRef](#)]
- LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
- LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* **1989**, *2*, 396–404.
- Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
- Chapagain, K.; Kittipiyakul, S.; Kulthanavit, P. Short-Term Electricity Demand Forecasting: Impact Analysis of Temperature for Thailand. *Energies* **2020**, *13*, 2498. [[CrossRef](#)]
- Phyo, P.P.; Jeenanunta, C.; Hashimoto, K. Electricity load forecasting in Thailand using deep learning models. *Int. J. Electr. Electron. Eng. Telecommun.* **2019**, *8*, 221–225. [[CrossRef](#)]
- Lim, B.; Zohren, S. Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200209. [[CrossRef](#)] [[PubMed](#)]
- Längkvist, M.; Karlsson, L.; Loutfi, A. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognit. Lett.* **2014**, *42*, 11–24. [[CrossRef](#)]
- Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [[CrossRef](#)]
- Hosein, S.; Hosein, P. Load forecasting using deep neural networks. In Proceedings of the 2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 23–26 April 2017; pp. 1–5.
- Hossen, T.; Nair, A.S.; Chinnathambi, R.A.; Ranganathan, P. Residential load forecasting using deep neural networks (DNN). In Proceedings of the 2018 North American Power Symposium (NAPS), Fargo, ND, USA, 9–11 September 2018; pp. 1–5.
- Cai, M.; Pipattanasomporn, M.; Rahman, S. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Appl. Energy* **2019**, *236*, 1078–1088. [[CrossRef](#)]
- Amarasinghe, K.; Marino, D.L.; Manic, M. Deep neural networks for energy load forecasting. In Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 19–21 June 2017; pp. 1483–1488.
- Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M.A. Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies* **2018**, *11*, 1636. [[CrossRef](#)]
- Bedi, J.; Toshniwal, D. Deep learning framework to forecast electricity demand. *Appl. Energy* **2019**, *238*, 1312–1326. [[CrossRef](#)]

19. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. Smart Grid* **2017**, *10*, 841–851. [[CrossRef](#)]
20. Khan, P.W.; Byun, Y.C. Genetic algorithm based optimized feature engineering and hybrid machine learning for effective energy consumption prediction. *IEEE Access* **2020**, *8*, 196274–196286. [[CrossRef](#)]
21. Ferdoush, Z.; Mahmud, B.N.; Chakrabarty, A.; Uddin, J. A short-term hybrid forecasting model for time series electrical-load data using random forest and bidirectional long short-term memory. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 763–771. [[CrossRef](#)]
22. Liu, Z.; Hara, R.; Kita, H. Hybrid forecasting system based on data area division and deep learning neural network for short-term wind speed forecasting. *Energy Convers. Manag.* **2021**, *238*, 114136. [[CrossRef](#)]
23. Pannakkong, W.; Pham, V.H.; Huynh, V.N. A novel hybridization of ARIMA, ANN, and K-means for time series forecasting. In *Research Anthology on Artificial Neural Network Applications*; IGI Global: Hershey, PA, USA, 2022; pp. 1532–1558.
24. Xiao, L.; Wang, J.; Hou, R.; Wu, J. A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting. *Energy* **2015**, *82*, 524–549. [[CrossRef](#)]
25. Zheng, H.; Yuan, J.; Chen, L. Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. *Energies* **2017**, *10*, 1168. [[CrossRef](#)]
26. Qiu, X.; Ren, Y.; Suganthan, P.N.; Amaratunga, G.A. Empirical mode decomposition based ensemble deep learning for load demand time series forecasting. *Appl. Soft Comput.* **2017**, *54*, 246–255. [[CrossRef](#)]
27. Zhang, X.; Wang, J. A novel decomposition-ensemble model for forecasting short-term load-time series with multiple seasonal patterns. *Appl. Soft Comput.* **2018**, *65*, 478–494. [[CrossRef](#)]
28. Zain, Z.M.; Alturki, N.M. COVID-19 Pandemic Forecasting Using CNN-LSTM: A Hybrid Approach. *J. Control Sci. Eng.* **2021**, *2021*, 8785636. [[CrossRef](#)]
29. Li, Y.; Dai, W. Bitcoin price forecasting method based on CNN-LSTM hybrid neural network model. *J. Eng.* **2020**, *2020*, 344–347. [[CrossRef](#)]
30. Alhussein, M.; Aurangzeb, K.; Haider, S.I. Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access* **2020**, *8*, 180544–180557. [[CrossRef](#)]
31. Chen, R.; Wang, X.; Zhang, W.; Zhu, X.; Li, A.; Yang, C. A hybrid CNN-LSTM model for typhoon formation forecasting. *Geoinformatica* **2019**, *23*, 375–396. [[CrossRef](#)]
32. Agyeman, K.A.; Kim, G.; Jo, H.; Park, S.; Han, S. An ensemble stochastic forecasting framework for variable distributed demand loads. *Energies* **2020**, *13*, 2658. [[CrossRef](#)]