processes

MDPI

Article

Development of a Two-Stage ESS-Scheduling Model for Cost Minimization Using Machine Learning-Based Load Prediction Techniques

Minsu Park [1], Jaehwi Kim [2], Dongjun Won [1] and Jaehee Kim [2,*]

[1] Department of Electrical Enginerring, Inha University, 100, Inha-ro, Michuhol-gu, Incheon 22212, Korea; zxcvbnmk7772@gmail.com (M.P.); djwon@inha.ac.kr (D.W.)

[2] Department of Statistics, Duksung Women's University, 33, Samyang-ro 144-gil, Dobong-gu, Seoul 01369, Korea; un7743@naver.com

* Correspondence: jaehee@duksung.ac.kr; Tel.: +82-02-901-8633

Received: 10 May 2019; Accepted: 9 June 2019; Published: 12 June 2019

Abstract: Effective use of energy storage systems (ESS) is important to reduce unnecessary power consumption. In this paper, a day-ahead two-stage ESS-scheduling model based on the use of a machine learning technique for load prediction has been proposed for minimizing the operating cost of the energy system. The proposed algorithm consists of two stages of ESS. In the first stage, ESS is used to minimize demand charges by reducing the peak load. Then, the remaining capacity is used to reduce energy charges through arbitrage trading, thereby minimizing the total operating cost. To achieve this purpose, accurate load prediction is required. Machine learning techniques are promising methods owing to the ability to improve forecasting performance. Among them, ensemble learning is a well-known machine learning method which helps to reduce variance and prevent overfitting of a model. To predict loads, we employed bootstrap aggregating (bagging) or random forest technique-based decision trees after Holt–Winters smoothing for trends. Our combined method can increase the prediction accuracy. In the simulation conducted, three combined prediction models were evaluated. The prediction task was performed using the R programming language. The effectiveness of the proposed algorithm was verified by using Python's PuLP library.

Keywords: energy storage system; two-stage algorithm; load prediction; machine learning; ensemble learning; bagging; random forest

1. Introduction

With the growing demand in global electricity, greenhouse gas (GHG) emissions have also increased significantly. Global efforts such as the Kyoto protocol and the Paris Agreement [1] have been made to reduce GHGs. Nevertheless, GHG emissions have continued to accelerate. Renewable energy source (RES) and ESS have are currently gaining attention as a solution to these problems. However, RESs adoption is delayed owing to high investment costs and the fact that uncertainties can affect the power system. In this regard, the use of ESS has increased dramatically in recent years. The reason is not only that it operates without uncertainty but also because it enables various services, such as peak reduction, arbitrage trading, demand response (DR), and frequency regulation (FR). However, accurate load forecasting [2–4] is necessary to provide these various services.

Thus, many studies on the utilization of ESS have been reported. Ke et al. [5] proposed a frequency regulation method with the battery life through the rain-flow algorithm, which is calculated regarding the depth of discharge (DOD). In a different study, the size, type, and location of ESS were assigned using a genetic algorithm (GA) and then ESS scheduling was performed by including a penalty cost for reverse flow. However, the working principle of the ESS regarding the predicted load was not described in detail [6]. Furthermore, a hybrid optimization of ESS in a photovoltaic (PV) integrated electric vehicle (EV) charging station has also been proposed. However, in each case, the load prediction method for ESS scheduling was not much discussed. In [7], the hybrid optimization was divided into rule-based and deterministic algorithm according to real-time prices. However, the proposed hybrid optimization method is only suitable for real-time plans and not for other prices. In [8,9], a cost minimization algorithm using hierarchical demand management was proposed. However, this algorithm does not use ESS and load prediction techniques.

With the development of sub-metering machines, a collection of load data has been made easy. Particularly, the development of the smart grid has made it possible to measure the load in real-time, enabling big data collection. Recently, many studies have been conducted to predict load using machine learning techniques because they have some advantages in predicting via computing-intensive learning. Hong [10] proposed a method of analyzing and predicting Taiwan regional electric load using support vector regression (SVR). Shi et al. [11] and Marino et al. [12] presented a method for estimating household load and building energy load using a recurrent neural network (RNN) as a deep learning technique. Abdel-Nasser and Mahmoud [13] proposed the use of long-short term memory recurrent neural network (LSTM-RNN) to predict the output power of photovoltaic systems accurately. Grmanová et al. [14] showed how to analyze and predict smart grid performance in Europe using ensemble learning.

The prediction method used in this paper is based on [14,15] with ensemble learning. Ensemble learning can compensate for a single learning problem by taking a representative value of the learning results in each base model.

Herein, we propose a day-ahead scheduling method that sequentially considers peak reduction and arbitrage trading using ESS. This method is especially suitable for environments where the demand charge and the energy charge are divided, as in the case of Korea's tariff system. Therefore, the proposed algorithm can be used to determine the hierarchical ESS output schedule based on predicted load data and market price. Optimal scheduling was calculated using mixed-integer linear programming (MILP) and simulated using actual Inha University predictive load data in South Korea.

Our work has a focus on use of a statistical model with Holt–Winters smoothing and ensemble learning. Based on this prediction including peaks, two-stage ESS operation is planned to reduce the cost using peak control and arbitrage trading, as shown Figure 1.

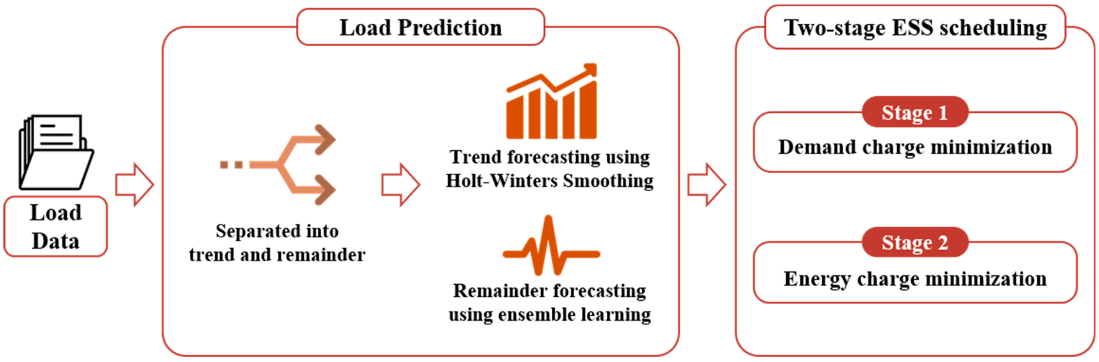

Load Data
Load Prediction
Separated into trend and remainder
Trend forecasting using Holt-Winters Smoothing
Remainder forecasting using ensemble learning
Two-stage ESS scheduling
Stage 1
Demand charge minimization
Stage 2
Energy charge minimization

Figure 1. Methodology diagram flow.

The remaining part of this paper is organized as follows: In Section 2, an ensemble learning based load prediction technique is discussed. In Section 3, we present a mathematical description of the two-stage ESS optimization problem. Section 4 presents a case study for verifying the effectiveness of the proposed algorithm and prediction techniques. Discussions and conclusions are presented in Section 5.

2. Load Prediction Using Ensemble Learning

Ensemble learning is a method of deriving a representative value by combining the results of several base models, thereby preventing overfitting and enhancing prediction accuracy owing to its ability to lower the variance [14]. Ensemble learning can be categorized into two types regarding the base model. If the base models used in the ensemble model are all the same, it is referred to as homogeneous learning (or horizontal ensemble framework), and if they are different, the ensemble is referred to as heterogeneous learning (or vertical ensemble framework) [16]. In this paper, we used bagging and random forest, which are homogeneous learning methods involving the use of a decision tree as a base model. Since the decision tree is a supervised learning method, appropriate predictors are needed to predict the response (load) [17]. Thus, we generated predictors using the feature engineering approach.

2.1. Decision Tree and Feature Engineering

A decision tree is a technique for performing classification and prediction with respect to certain rules and it is also referred to as classification and regression tree (CART) [18]. A decision tree consists of nodes, such as a root node, intermediate nodes, and terminal nodes, according to a certain predictor rule. The estimate (or predictive value) is the average of the responses categorized as a terminal node [17]. We employed recursive partitioning regression trees (RPART) proposed in [19] and conditional inference trees (CTREE) proposed in [20]. RPART is a decision tree that uses recursive partitioning technique, making it possible to create more intuitive models and to predict more accurately than the conventional methods. CTREE solves the overfitting problem associated encountered when using recursive partitioning in a conditional inference framework.

Feature engineering is the process of handling initial data and creating features by using the knowledge of the data to increase the performance of the machine learning model or to generate the data to be used as input of the machine learning model [21,22]. Decision trees are not able to estimate the trend well, because they derive the appropriate response for predictors. Thus, we analyzed the load after separating it into the trend and the remainder. The trend part was estimated or predicted using Holt–Winters smoothing [23] technique while de-trend part was estimated using ensemble learning method. We used the period of the de-trend data to generate the predictor to be used in the decision tree. The period can be expressed using sine and cosine curves and can be used as a predictor along with the seasonal effect.

2.2. Bagging and Random Forest

Bagging is an ensemble learning technique which involves the use of bootstrap sampling. Bagging can be used to generate several bootstrap samples from one training data and the sample can be used to fit different decision trees [24]. The bagging result was calculated as an average, thus, reducing the variance of the model. This principle is the same as that the variance of the sample mean becomes σ^2/n when n samples are drawn from a population with variance σ^2 [17]. The value obtained using the bagging can be defined as follows:

$$\hat{f_{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f^{*b}}(x) \tag{1}$$

where x is the response, B denotes the number of bootstrap sample, and $\hat{f^{*b}}(x)$ denotes the value obtained by applying the bth bootstrap sample to the decision tree.

Bagging is an excellent way of reducing variance, although it has one problem: If there are predictors that have a much greater influence on the decision tree, the result is dominated by these influential predictors even if the decision tree is repeatedly generated many times. It is possible to prevent cases of all the three models made similar by making decision tree models using only a few predictors, which is referred to as a random forest [17,18]. Random forests use regression trees with only *p*/3 number of predictors when there is *p* number of predictors.

2.3. Load Predicton Model

In this paper, we discuss the three models that were used as a prediction model by using bagging with RPART, CTREE, or random forest. The three models can be summarized as follows:

1. Model 1: Holt–Winters smoothing + Bagging with RPART
2. Model 2: Holt–Winters smoothing + Bagging with CTREE
3. Model 3: Holt–Winters smoothing + Random Forest

We used the *rpart*, *party*, and *randomForest* R packages for the models, respectively. We also used a deep neural network (DNN) with 10 hidden layers as a comparison model to demonstrate the performance of ensemble learning. DNN was implemented through the *neuralnet* R packages. As the model accuracy measures regarding the test data, root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) were calculated. The three measures are defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|y_t - \hat{y}_t\right| \tag{3}$$

$$\mathrm{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| (\%) \tag{4}$$

where *t* denotes the time point, *n* denotes the number of the time points in test data in the model.

3. Two-Stage ESS-Scheduling Algorithm

The two-stage algorithm aims to minimize electric charges by using ESS and machine learning-based load prediction. In Korea's tariff system, electric charges are divided into demand charges and energy charges, which are caused by peak load and energy consumption, respectively. Generally, demand charges are 10 times more expensive than energy charges and can be maintained for one year when renewed [25] (e.g., demand charge is 6490 KRW/kW, and energy charges are 60.5, 86.3, and 119.8 for the off-peak, mid-peak, and on-peak, respectively). Therefore, the proposed algorithm is considered as a hierarchical structure, which minimizes demand charges by first reducing peak loads, and minimizing energy charges through the arbitrage trading with the remaining capacity of the ESS. Figure 2 shows the flow chart of the ESS optimization algorithm.

To use the 2-stage algorithm, it receives input data (electricity price, ESS information, load data, initial peak load) and makes predicted load data based on machine learning. Specifically, the loads used in this paper are fixed loads and load scheduling is not considered. the algorithm is operated between working time, i.e., $t = t_{work,s} - t_{work,e}$. Upon setting these parameters, peak of the predicted load is compared with the initial peak load. If the peak of the predicted load is greater than the initial peak load, perform stage 1 and stage 2 sequentially; otherwise, perform only stage 2. This process repeats until *t* = T. T means the number of time steps for which electricity charges are set.

In stage 1, ESS is scheduled for peak load minimization. And then updates the initial peak load. Stage 2 perform ESS scheduling to minimize the energy cost through arbitrage trading, while not exceeding the peak load updated in stage 1. Finally, calculate the economic benefits using the determined ESS power schedule.

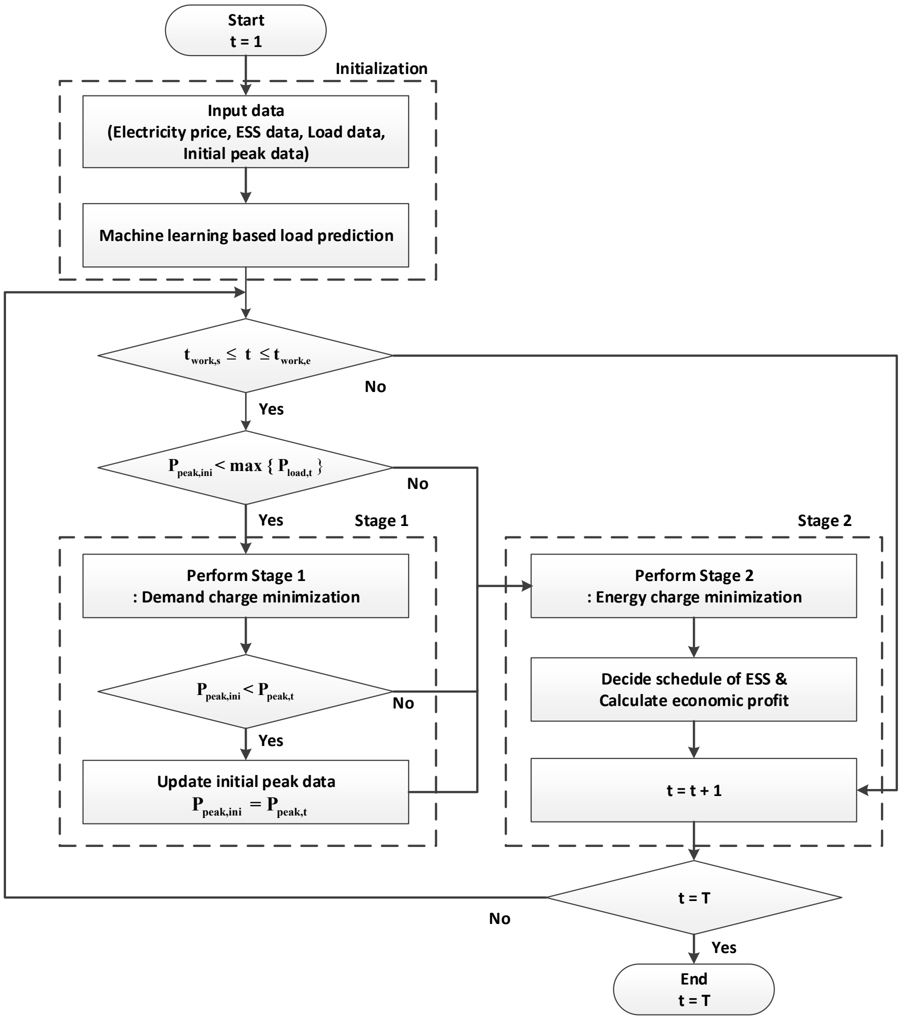


Figure 2. Flow chart of the ESS optimization algorithm.

3.1. ESS Optimization Model for Minimizing Demand Charges (Stage 1)

ESS can perform peak reduction service by charging at off-peak and discharging at on-peak. Alternatively, ESS may make arbitrage trading on time-of-usage (TOU). As mentioned earlier, demand charges in Korea's tariff system is a major part of electric charges paid. Therefore, the ESS optimization model performs stage 1 before stage 2. The mathematical model of stage 1 is as follows:

$$\min\left\{C_{dem} \times P_{peak}\right\} \tag{5}$$

Equation (5) is the objective function of stage 1 used for minimizing demand charges by reducing the peak load. In stage 1, the optimization model is implemented to obtain a lower peak load. The symbol C_{dem}, denotes demand charges that do not change with time while P_{peak} denotes the peak load. Demand charge is a constant determined only by magnitude of peak load. The objective function is subject to several constraints, as shown below:

$$P_{peak} = \max\left\{\left(P_{load,t} - P_{sch,t},\ P_{peak,ini}\right)\right\},\ \forall\ t \tag{6}$$

$$P_{grid,t} = P_{load,t} - (u_{dhc,t} \times P_{dch,t} - u_{cha,t} \times P_{cha,t)},\ \forall\ t \tag{7}$$

$$P_{sch,t} = u_{dhc,t} \times P_{dch,t} - u_{cha,t} \times P_{cha,t},\ \forall\ t \tag{8}$$

$$-P_{cha,max} \leq P_{sch,t} \leq P_{dch,max},\ \forall\ t \tag{9}$$

$$u_{dch,t} + u_{cha,t} \leq 1, \forall t \tag{10}$$

$$SOC_{t+1} = SOC_t - u_{dch,t} \times P_{dch,t} \times \eta_{dch}^{-1} + u_{cha,t} \times P_{cha,t} \times \eta_{cha}, \forall t \tag{11}$$

$$SOC_{min} \leq SOC_t \leq SOC_{max}, \forall t \tag{12}$$

$$SOC_{ini} = SOC_{fin} \tag{13}$$

where the time step Δt represents 15 min because electric charges are measured and determined every 15 min in Korea, $P_{peak,ini}$ denotes the initial peak load that determines the current demand charge. P_{peak} and $P_{load,t}$ refer to the peak load determined in Equation (6) and the actual load, respectively. $P_{dch,t}$, $u_{dhc,t}$, $P_{cha,t}$, and $u_{cha,t}$ denote the discharge power, discharge state variable, charging power, and charging state variable, respectively. These are decision variables for this problem and are used to calculate P_{peak} through Equation (6). $P_{sch,t}$ denotes the scheduled ESS power. Equation (6) implies that, after ESS operation, the peak load needs to be updated if it is larger than the initial peak load. $P_{grid,t}$ in Equation (7) is the active power injected to the network by independent system operator (ISO). The load except the scheduled ESS output is shown in Equation (7). Equation (8) represents the power of the scheduled ESS, whereas Equation (9) limits its range. Furthermore, Equation (10) denotes a constraint that prevents simultaneous charging and discharging. Specifically, $u_{dhc,t}$ and $u_{cha,t}$ are integer variables that determine the state of ESS for each time. The process in state-of-charge (SOC) can be evaluated as discussed in Equation (11). η_{dch} and η_{cha} refer to discharging efficiency and charging efficiency, respectively. Equation (12) denotes a constraint which ensures that ESS operates within the appropriate SOC range and ensures safe operation. As Equation (13) shows, the initial SOC and final SOC were set to have the same value. This implies that the constraint prevents excessive charge and discharge on a specific day.

3.2. ESS Optimization Model for Minimizing Energy Charges (Stage 2)

In stage 1, ESS produces the best performance in reducing the peak load. Thus, stage 2 is performed with values determined in stage 1 and remaining capacity. In stage 2, the ESS optimization model discharges when the electric price is low and charges when it is high to minimize energy charges. At this time, the peak load determined in stage 1 must not be exceeded. The peak load determined in stage 1 is added as a constraint, and the rest of constraint is the same as stage 1:

$$\min \sum_{t=1}^{T} \left\{ C_{ene,t} \times \left(P_{load,t} - P_{sch,t} \right) \times \Delta \mathrm{t} \right\} \tag{14}$$

$$\left(P_{load,t} - P_{sch,t} \right) \leq \mathrm{P_{peak}}, \forall \mathrm{t} \tag{15}$$

where $C_{ene,t}$ denotes the energy charges with respect to time-of-usage (TOU); T denotes the number of time steps for which the electricity charges are measured. Equation (14) shows the objective function of stage 2 which can be used to minimize energy charges through arbitrage trading. There is no peak update through arbitrage trading based on Equation (15). Therefore, the demand charges are not renewed, and the total electric charge is minimized. The other constraints follow Equations (6)–(13). The two-stage optimization model preferentially performs stage 1 on days when the predicted peak load is expected to be updated. Otherwise, only stage 2 is performed.

3.3. Optimization Method for Two-Stage ESS-Scheduling

Linear programming (LP) is an optimization method for finding the solution of a problem where the objective function and constraint appear as a linear function of state variables. In the case of LP, the problem constraint can be classified into equality and inequality constraints. Sometimes it is necessary to integerize state variables when solving practical problems. However, integerizing state variables may affect other variables and may not satisfy the constraints. Mixed-integer linear programming (MILP) is one of the optimization methods for overcoming this problem. Similar to LP, MILP can be used to obtain a solution to a problem where the objective function and the constraint appear as a linear function of the state variable, but it is useful when the state variable contains an integer [26].

In this paper, the proposed two-stage ESS-scheduling model includes integer variables. If these variables are not integers (if you solve the problem with LP), they may affect ESS scheduling. For example, variables that indicate the charge and discharge states of the ESS can affect ESS power when it is not an integer. These variables mean charging and discharging at 1, and not at 0. However, when it is not an integer, it does not know what state it represents. Therefore, the proposed model solves the problem using MILP.

4. Case Study

4.1. Load Data

Fifteen-minute interval load data was used to obtain the predicted load data learned by machine learning techniques. This load data was used during the summer (June–August) at Inha University in Incheon, South Korea from July 2015–August 2017. We removed weekends and holidays to increase the prediction accuracy because they have different consumption behaviors compared to weekdays. We used 20 days (3–31 August 2017) at the end of the load data as the test data while the rest were used as the training data. Figure 3 shows the training data used to predict the future load.

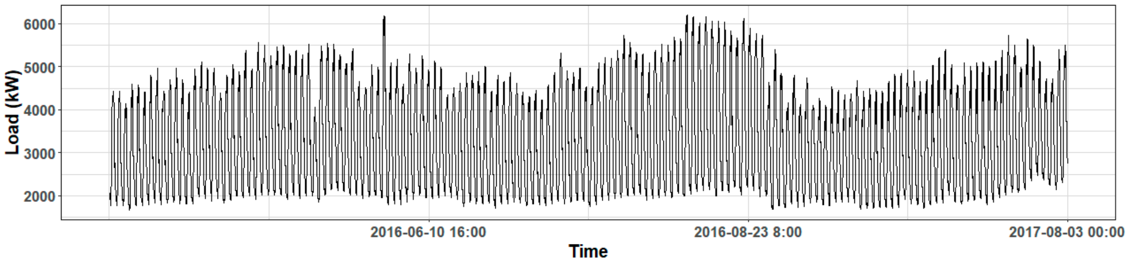


Figure 3. The 15-min interval train load data used during the summer at Inha University.

4.2. Ensemble Learning Based Load Prediction

We use 100 bootstrap samples for bagging and set the options for random forest as follows:

$$\text{ntree} = 100, \qquad \text{mtry} = 2, \qquad \text{nodesize} = 6$$

where ntree denotes the number of trees (the number of bootstrap samples), mtry denotes the number of predictors randomly used for each trees, and nodesize denotes the minimum size of terminal nodes. The parameters used in the trend prediction are as follows:

$$Period = 5904,\ \ \alpha = 0.9967874,\ \ \beta = 0.02922989,\ \ \gamma = 0.0632884$$

where *Period* denotes the period of the time series, and α, β, and γ refer to smoothing parameters. Smoothing parameters are automatically chosen by minimizing the model's training RMSE.

Figure 4 is a graph comparing the predicted values with the test data, and Table 1 is a comparison of the accuracy of the predicted values. Bagging based on CTREE (Model 2) has the lowest RMSE, MAE, and MAPE values and the best prediction among the three models. Surprisingly, the predicted value of the random forest does not fit well, which seems to be the performance difference between the general tree and RPART and CTREE. However, the prediction accuracy measure of the three models seems not quite different. Above all, the three methods show better effects than DNN.

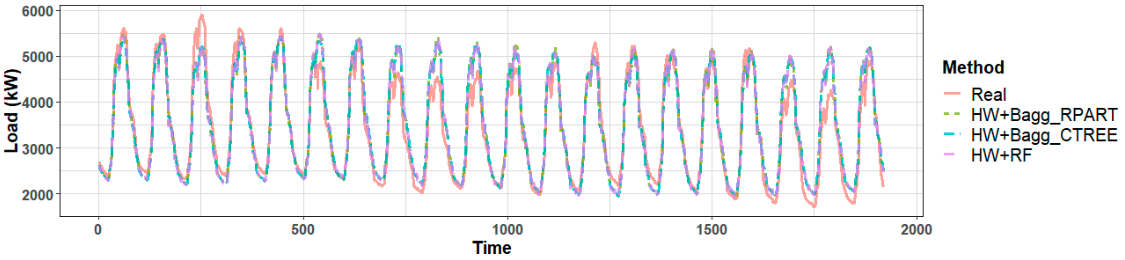


Figure 4. Comparison of ensemble learning forecast.

Table 1. Accuracy based on test data.

Model.	RMSE	MAE	MAPE (%)
Model 1	299.3747	227.1160	6.740471
Model 2	296.4163	225.8964	6.704895
Model 3	299.9798	227.8008	6.771545
DNN model	526.2574	419.8826	12.991492

4.3. Results of the Two-Stage ESS-Scheduling Model Using Machine Learning Load Prediction

4.3.1. Input Parameters for the Optimization Model

To validate the two-stage optimization model, data recorded from July 2015–August 2018 were used as raw data for various machine learning techniques. Therefore, the methods were used to obtain the predicted summer load data of Inha University. The rated power and rated capacity of the ESS are 250 kW and 250 kWh, respectively. Both charging and discharging efficiency were 90%. Table 2 presents the input parameters for the optimization model.

Table 2. Configuration data for the optimization model.

Parameter	Definition	Value	Parameter	Definition	Value
C_{dem}	Demand Charges	6980 KRW/kW	$P_{peak,ini}$	Initial Peak	5412.15
η_{cha}	Charging Efficiency	90%	SOC_{max}	Maximum SOC	0.9
η_{dha}	Discharging Efficiency	90%	SOC_{min}	Minimum SOC	0.1
$P_{cha,max}$	Discharge Rated Power	250 kW	SOC_{ini}	Initial SOC	0.5
$P_{dch,max}$	Charge Rated Power	250 kW	SOC_{fin}	Final SOC	0.5

The proposed optimization model was implemented and simulated in a MATLAB environment (R2018b, The MathWorks Inc., Natick, MA, USA). The “intlinprog” solver in MATLAB was used to solve the MILP optimization problem. In this problem, electric charges are the most important factor in determining ESS scheduling. In addition, the demand charges are updated if the average value of electricity usage for 15 min is greater than the initial peak load. Consequently, a simulation was carried out every 15 min. This means that T = 96. In other word, it means that every 15 min of simulation is repeated 96 times. In this case study, the demand charge was 6980 KRW/kW and energy charges incurred are presented in Figure 5. These charges consist of off-peak, on-peak, middle peak, 45.3 KRW/kW from 23:00–09:00, 155.9 KRW/kW from 10:00–12:00 and 13:00–17:00, other times were set at 90 KRW/kW, respectively, and they are based on tariffs provided by the Korea Electric Power Corporation (KEPCO) [10].

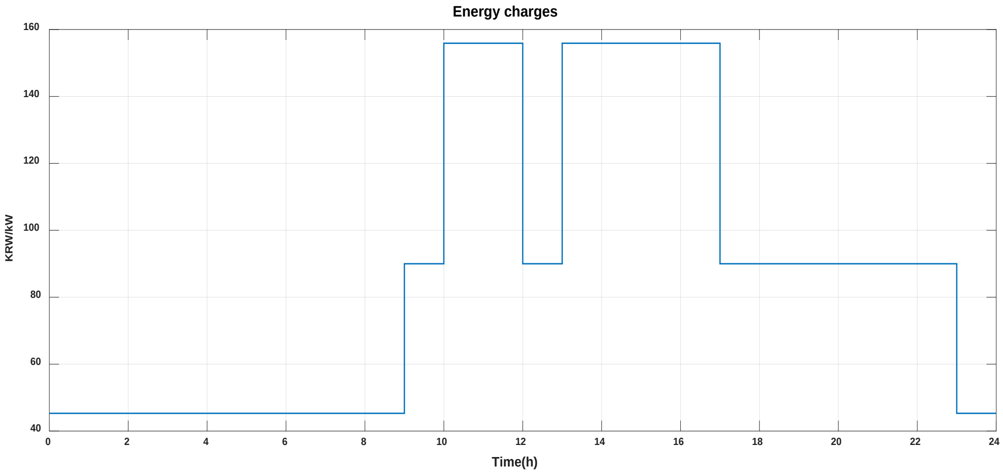


Figure 5. Energy charges.

4.3.2. Two-Stage ESS-Scheduling Simulation Results

Figure 6 shows the predicted and actual loads on August 3, when the initial peak was updated. These loads were used for optimization model simulations that were conducted. On this day, the actual peak load is 5614.08 kW and load prediction models 1, 2, and 3 are 5474.60, 5302.70, and 5520.52 kW, respectively, except for model 2, which exceeds the initial peak value of 5412.15 kW. Therefore, the load profile excluding model 2 would be applied to both stage 1 and stage 2, but only stage 2 would be applied to model 2. Table 3 summarizes the simulation cases.

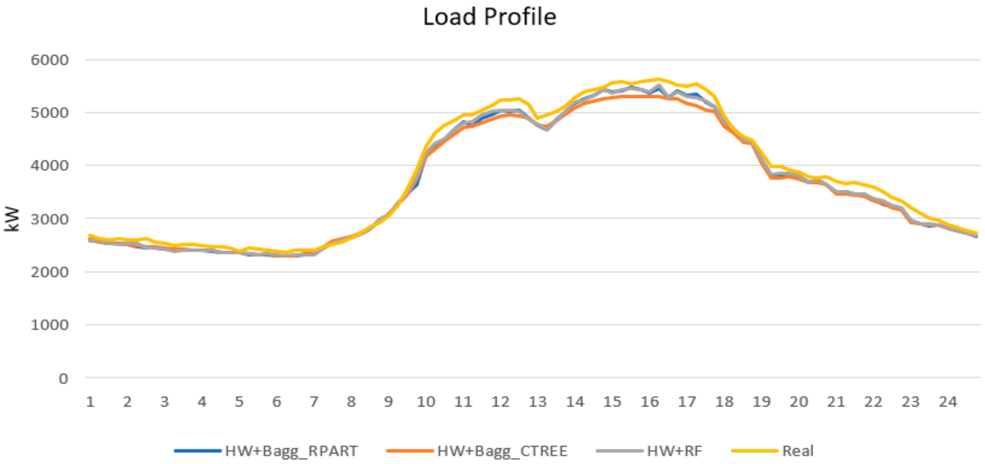


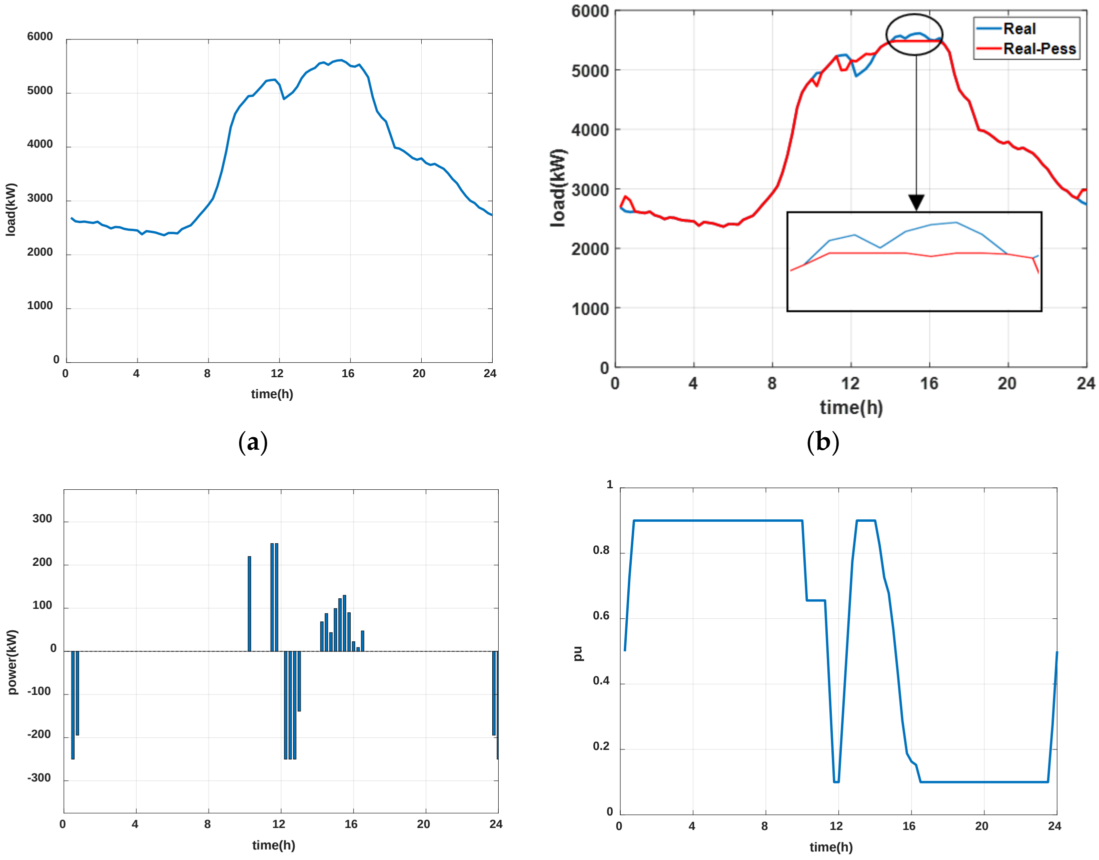
Figure 6. Load profile used in simulation.

Table 3. Simulation case.

Case	**Simulation Conditions**	
	Load Data	**Peak Load**
Case 1	Real	5614.08 kW
Case 2	HW + Bagg_RPART (Model 1)	5474.60 kW
Case 3	HW + Bagg_CTREE (Model 2)	5302.70 kW
Case 4	HW+RF (Model 3)	5520.52 kW

Case 1

Case 1 shows the result of applying the two-stage algorithm to the actual load. In other words, since the actual load is assumed to be the predicted load, the prediction error is 0%, which is the most ideal simulation result. Before applying the algorithm, the peak load on August 3 is 5614.08 kW, and peak load after optimization is 5484.1 kW, which is 130 kW less than the peak load before optimization. The peak load was distributed over a wide range of time; therefore, it was not possible to reduce the rated output of ESS by 250 kW. These results are shown in Figure 7a,b. Figure 7c indicates that the scheduled ESS operates with rated power. During times of peak load and high electricity rates, ESS was discharged and at other times it was charged. Finally, Figure 7d shows SOC variation due to ESS operation. Per unit (pu) is the current value divided by the nominal value. In other words, it has a value between 0 and 1 under steady state conditions.



(**a**) (**b**)

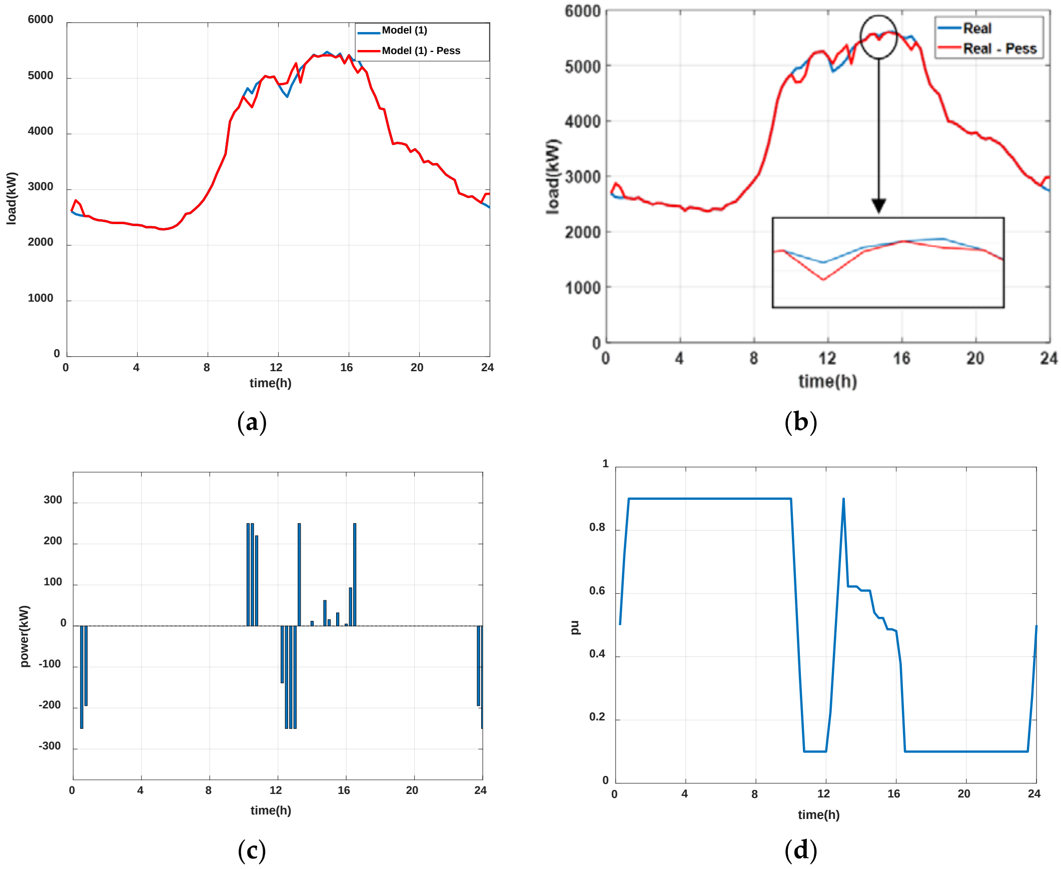
(**c**) (**d**)

Figure 7. Case 1 simulation results. (**a**) Actual load on August 3; (**b**) comparison actual load with load after applying the two-stage algorithm; (**c**) ESS power; and (**d**) SOC.

Case 2

Cases 2–4 show the results of applying the two-stage algorithm to the predicted load. Among them, the predicted load used in Case 2 is 6.74% for MAPE and 5474.60 kW for the estimated peak load. The two-stage algorithm was applied because the predicted peak load is larger than the initial peak load. Therefore, it has both the potential of reducing demand charges due to peak reduction and energy charges through arbitrage trading.

Figure 8a shows the load before and after the algorithm was applied to the predicted load. On the other hand, the result of applying the scheduled ESS output according to the predicted load to actual load occurring on August 3 is shown in Figure 8b. When applied to the predicted load, it was expected to be reduced by 62.5 kW to 5412.1 kW; however, it was found to be 5606.4 kW when applied to the actual load. Scheduled ESS power and SOC according to the predicted load are shown in Figure 8c,d, respectively.



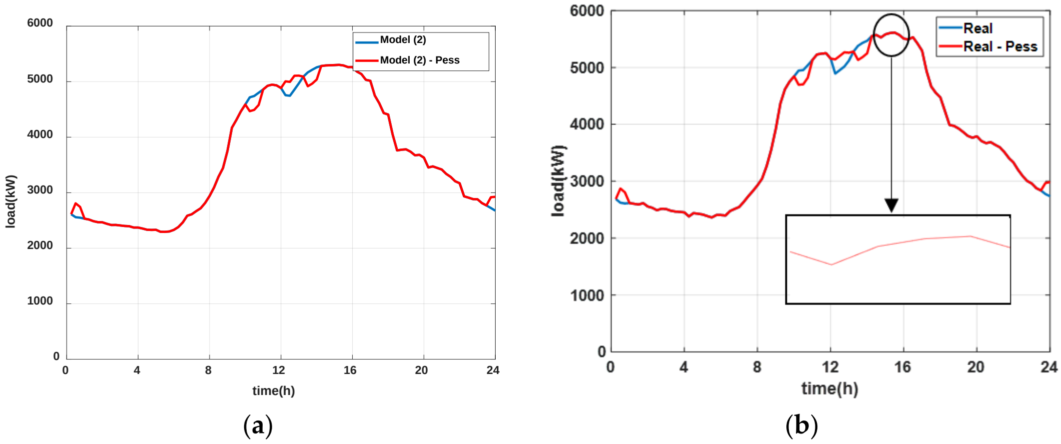
(**a**) (**b**)

(**c**) (**d**)

Figure 8. Case 2 simulation results. (**a**) Comparison of the predicted load with the actual load after applying the two-stage algorithm; (**b**) comparison of the actual load with the recorded load after applying the two-stage algorithm; (**c**) ESS power; and (**d**) SOC.

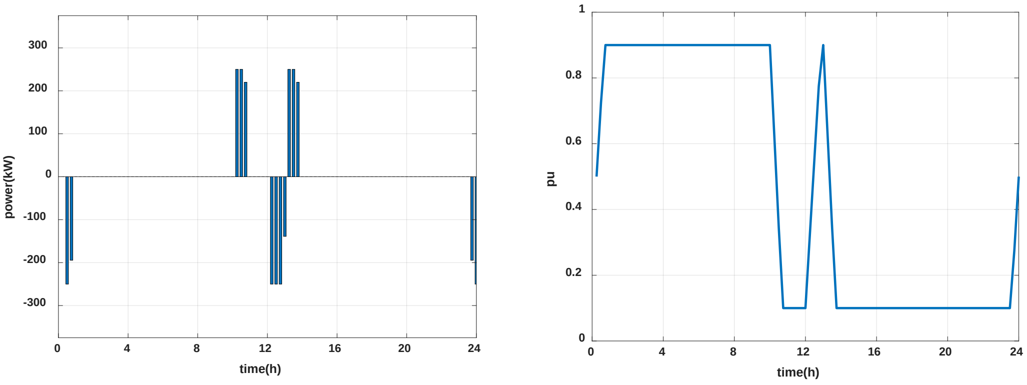
Case 3

Case 3 was simulated for the predicted load using Model 2. At this moment, the MAPE is expected to be 6.7% and the peak load to be 5302.7 kW. However, contrary to Case 2, only Stage 2 was applied because the predicted peak load was smaller than the initial peak load. Therefore, the ESS was scheduled for arbitrage trading. Figure 9a shows the result of applying the scheduled ESS value to the predicted load while Figure 9b shows the result when the scheduled ESS value was applied to the actual load. Figure 9c,d show ESS power and SOC, respectively.



(**a**) (**b**)

Figure 9. *Cont.*

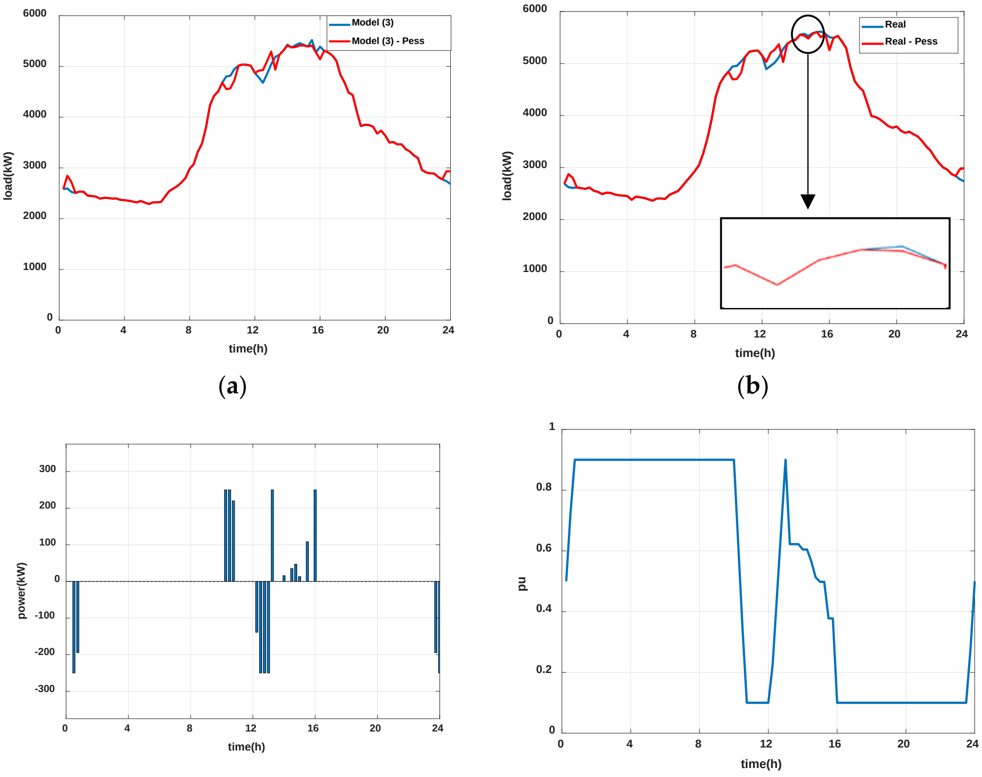


(**c**) (**d**)

Figure 9. Case 3 simulation results. (**a**) Comparison of the predicted load with the recorded load after applying the two-stage algorithm; (**b**) comparison of the actual load with the recorded load after applying the two-stage algorithm; (**c**) ESS power; and (**d**) SOC.

Case 4

The peak of the predicted load used in Case 4 is 5520.52 kW. Case 2 and two-stage algorithms were applied. Figure 10a shows the results obtained before and after the two-stage algorithm was applied to the predicted load. Figure 10b shows the result of applying the scheduled ESS value to the actual load. After applying the algorithm, the peak load was expected to be 5412.2 kW. However, when applied to the actual load, it was found to be 5606.4 kW. As in Cases 1–3, Figure 10c,d show the ESS output and SOC.



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Case 4 simulation results. (**a**) Comparison between the predicted load and the recorded load after applying the two-stage algorithm; (**b**) comparison between the actual load and the recorded load after applying the two-stage algorithm; (**c**) ESS power; and (**d**) SOC.

Table 4 summarizes the simulation results. The base case means electric charges on August 3 when the ESS did not work. Case 1, which assumes a prediction error of 0%, is the lowest rate. In addition, Case 3, which predicted that the peak load would be lower than the initial peak load, has the highest demand charges. However, as Table 4 shows, the energy costs are all the same. Since ESS capacity is smaller than the load, there is a limit to its participation in Stage 2.

Table 4. Simulation results.

Case	Simulation Results		
	Demand Charges	**Energy Charges**	**Total Reduction Rate**
Base Case	39,186,000 KRW	9,132,015 KRW/kW	-
Case 1	38,279,000 KRW	9,106,000 KRW/kW	933,015 KRW
Case 2	39,132,672 KRW	9,106,000 KRW/kW	79,343 KRW
Case 3	39,186,000 KRW	9,106,000 KRW/kW	26,015 KRW
Case 4	39,132,672 KRW	9,106,000 KRW/kW	79,343 KRW

5. Conclusions and Future Works

This paper proposes a two-stage ESS scheduling model based on machine learning for predicting electricity load. The proposed algorithm minimizes the ESS operator's electricity bill by reducing demand charges generated due to peak loads and energy charges caused due to electricity consumption. From the results of the simulations conducted, we observed that the algorithm is dependent on the predicted load and the electric rates. In addition, reducing demand charges is more profitable than reducing energy charges. Hence, an economical two-stage ESS-scheduling operation requires an accurate peak prediction.

We conducted load prediction using ensemble learning and Holt–Winters smoothing. It is widely agreed that the best approach is to estimate the load by employing bagging which uses CTREE as the base model (Model 2). It is a generally known fact that random forests are better than bagging. Therefore, this result seems to be the result of the difference in the base model. Using CTREE model as a base model in a random forest may lead to better results than this. Additionally, the load is a variable that is highly influenced by the environment. Using weather variables such as temperature and humidity and other specific variables relating to the load can be used to obtain a better model. This is a more in-depth issue to be considered in a study on load prediction.

In this study, the specification of ESS was assumed to be 250 kW/250 kWh. As a future study, we will further look into the economics of the two-stage algorithm by adding a technique for estimating the optimal capacity of the ESS for the load profiles.

Author Contributions: M.P. carried out the main research tasks, proposed the two-stage algorithm and wrote the full manuscript, and J.K. (Jaehwi Kim) provided technical support to verify the machine learning algorithm for load prediction in simulation software. J.K. (Jaehee Kim) and D.W. validated and double-checked the proposed strategy, the results, and the whole manuscript.

Funding: This research received no external funding.

Acknowledgments: This research was supported by Korea Electric Power Corporation (grant number: R18XA01). This work is supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry and Energy (MOTIE) of the Republic of Korea (no. 20161210200610). The authors would like to thank Enago (www.enago.co.kr) for the English language review.

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

References

1. UNFCCC. United Nations Climate Change. 2019. Available online: http://unfccc.int (accessed on 4 February 2019).
2. Choi, S.; Choi, S.W.S.; Min, S.W. Optimal scheduling and operation of the ESS for prosumer market environment in grid-connected industrial complex. *IEEE Trans. Ind. Appl.* **2018**, *54*, 1949–1957. [CrossRef]

3. Dang, J.; Seuss, J.; Suneja, L.; Harley, R.G. SOC feedback control for wind and ESS hybrid power system frequency regulation. *IEEE J. Emerg. Sel. Top. Power Electron.* **2014**, *2*, 79–86. [CrossRef]
4. Faisal, M.; Hannan, M.A.; Ker, P.J.; Hussain, A.; Mansor, M.B.; Blaabjerg, F. Review of energy storage system technologies in microgrid applications: Issues and challenges. *IEEE Access* **2018**, *7*, 35143–35164. [CrossRef]
5. Ke, X.; Lu, N.; Jin, C. Control and Size Energy Storage Systems for Managing Energy Imbalance of Variable Generation Resources. *IEEE Trans. Sustain. Energy* **2015**, *6*, 70–78. [CrossRef]
6. Rangel, C.A.S.; Canha, L.; Sperandio, M.; Severiano, R. Methodology for ESS-type selection and optimal energy management in distribution system with DG considering reverse flow limitations and cost penalties. *IET Gener. Trans. Distrib.* **2017**, *12*, 1164–1170. [CrossRef]
7. Chaudhari, K.; Ukil, A.; Kumar, K.N.; Manandhar, U.; Kollimalla, S.K. Hybrid optimization for economic deployment of ESS in PV-integrated EV charging stations. *IEEE Trans. Ind. Inf.* **2018**, *14*, 106–116. [CrossRef]
8. Tang, Y.; Chen, Q.; Ning, J.; Wang, Q.; Feng, S.; Li, Y. Hierarchical control strategy for residential demand response considering time-varying aggregated capacity. *Electron. Power Energy Syst.* **2018**, *97*, 165–173. [CrossRef]
9. Zhang, J.; Zhang, P.; Wu, H.; Qi, X.; Yang, S.; Li, Z. Two-stage load-scheduling model for the incentive-based demand response of industrial users considering load aggregators. *IET Gener. Trans. Distrib.* **2018**, *12*, 3518–3526. [CrossRef]
10. Hong, W.C. Electric load forecasting by support vector model. *Appl. Math. Model.* **2009**, *33*, 2444–2454. [CrossRef]
11. Shi, H.; Xu, M.; Li, R. Deep learning for household load forecasting—A novel pooling deep RNN. *IEEE Trans. Smart Grid* **2018**, *9*, 5271–5280. [CrossRef]
12. Marino, D.L.; Amarasinghe, K.; Manic, M. Building energy load forecasting using deep neural networks. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–25 October 2016; pp. 7046–7051.
13. Abdel-Nasser, M.; Mahmoud, K. Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Comput. Appl.* **2017**, *28*, 1–14. [CrossRef]
14. Grmanová, G.; Laurinec, P.; Rozinajová, V.; Ezzeddine, A.B.; Lucká, M.; Lacko, P.; Návrat, P. Incremental ensemble learning for electricity load forecasting. *Acta Polytechnica Hungarica* **2016**, *13*, 97–117.
15. Laurinec, P. Ensemble Learning for Time Series Forecasting in R. 2017. Available online: https://petolau.github.io/Ensemble-of-trees-for-forecasting-time-series/ (accessed on 10 August 2018).
16. Zang, W.; Zhang, P.; Zhou, C.; Guo, L. Comparative study between incremental and ensemble learning on data streams: Case study. *J. Big Data* **2014**, *1*, 5. [CrossRef]
17. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. (Eds.) *An Introduction to Statistical Learning*; Springer: New York, NY, USA, 2013.
18. Hastie, T.; Tibshirani, R.; Fridman, J. (Eds.) *The Elements of Statistical Learning*; Springer: New York, NY, USA, 2001.
19. Therneau, T.M.; Atkinson, E.J. *An Introduction to Recursive Partitioning Using the RPART Routines*; Technical Report; Mayo Foundation: Rochester, MN, USA, 1997.
20. Hothorn, T.; Hornik, K.; Zeileis, A. Party: A laboratory for recursive part(y)itioning. 2010. Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.151.2872 (accessed on 10 February 2019).
21. Deshpande, B. Learning Data Science: Feature Engineering. 2016. Available online: http://www.simafore.com/blog/learning-data-science-feature-engineering (accessed on 10 February 2019).
22. Brownlee, J. Discover Feature Engineering, How to Engineer Features and How to Get Good at It. 2014. Available online: https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/ (accessed on 10 February 2019).
23. Cowpertwait, P.S.P.; Metcalfe, A.V. (Eds.) *Introductory Time Series with R*; Springer: New York, NY, USA, 2009.
24. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
25. KEPCO. Electric Rates Table. 2019. Available online: http://cyber.kepco.co.kr/ckepco/front/jsp/CY/E/E/CYEEHP00201.jsp (accessed on 11 January 2019).
26. Rao, S.S. (Ed.) *Engineering Optimization: Theory and Practice*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2009.