



## Article

# Day-Ahead Electricity Market Price Forecasting Considering the Components of the Electricity Market Price; Using Demand Decomposition, Fuel Cost, and the Kernel Density Estimation

Arim Jin , Dahan Lee , Jong-Bae Park and Jae Hyung Roh \* 

Department of Electrical and Electronics Engineering, Konkuk University, Seoul 05029, Republic of Korea; doooj2@konkuk.ac.kr (A.J.); timeflaw@konkuk.ac.kr (D.L.); jbaepark@konkuk.ac.kr (J.-B.P.)

\* Correspondence: jhroh@konkuk.ac.kr

**Abstract:** This paper aims to improve the forecasting of electricity market prices by incorporating the characteristics of electricity market prices that are discretely affected by the fuel cost per unit, the unit generation cost of the large-scale generators, and the demand. In this paper, two new techniques are introduced. The first technique applies feature generation to the label and forecasts the transformed new variables, which are then post-processed by inverse transformation, considering the characteristic of the fuel types of marginal generators or prices through two variables: fuel cost per unit by the representative fuel type and argument of the maximum of Probability Density Function (PDF) calculated by Kernel Density Estimation (KDE) from the previous price. The second technique applies decomposition to the demand, followed by a feature selection process to apply the major decomposed feature. It is verified using gain or SHapley Additive exPlanations (SHAP) value in the feature selection process. In the case study, both showed improvement in all indicators. In the Korean Electricity Market, the unit generation cost for each generator is calculated monthly, resulting in a step-wise change in the electricity market price depending on the monthly fuel cost. Feature generation using the fuel cost per unit improved the forecasting by eliminating monthly volatility caused by the fuel costs and reducing the error that occurs at the beginning of the month. It improved the Mean Squared Percentage Error (MAPE) of 3.83[%]. Using the argument of the maximum PDF calculated by KDE improved the forecasting during the test period, where discrete monthly variations were not included. The resulting MAPE was 3.82[%]. Combining these two techniques resulted in the most accurate performance compared to the other techniques used, which had a MAPE of 3.49[%]. The MAPE of the forecasting with the decomposed data of the original price was 4.41[%].

**Keywords:** electricity market; price forecast; artificial intelligence; decomposition; feature selection; data preprocessing



**Citation:** Jin, A.; Lee, D.; Park, J.-B.; Roh, J.H. Day-Ahead Electricity Market Price Forecasting Considering the Components of the Electricity Market Price; Using Demand Decomposition, Fuel Cost, and the Kernel Density Estimation. *Energies* **2023**, *16*, 3222. <https://doi.org/10.3390/en16073222>

Academic Editor: Tiago Pinto

Received: 10 March 2023

Revised: 23 March 2023

Accepted: 30 March 2023

Published: 3 April 2023



**Copyright:** © 2023 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

By forecasting the prices of the electricity market, market participants can improve their profits. Especially for the participation in the electricity market through Virtual Power Plants (VPPs) or microgrids, which have less influence on the price determination of the electricity market, the strategies are simply based on price forecasting. Therefore, the accuracy of price forecasting, as described in [1], directly impacts profits and can significantly maximize profits. As a result, there have been various attempts to forecast the price of the electricity market.

There are six types of price forecasting models in the electricity market. First, the multi-agent-based model. Second, the model for the structural approach to the electricity market. Third, the probabilistic model characterizes statistical elements of electricity prices over time. Fourth, the statistical model is based on econometric approaches. Fifth, the artificial intelligence-based model can reflect dynamic systems. Combining these five types of models has also been considered [2].

There are several key features in the principles of electricity market price determination [3,4]. The centralized electricity market determines the electricity market price by optimizing it based on these factors. One is that demand and supply must match at certain time points. Second, electricity is difficult to store in large quantities. Third, demand is typically inelastic. Fourth, the flow of electricity is determined by the impedance of transmission lines and the load, and the source of the electricity used cannot be known. Finally, there is a capacity constraint on transmission lines, limiting the amount of electricity that can be supplied.

There have been many discussions on comparing univariate and multivariate forecasting for electricity market price forecasting [2]. Multivariate forecasting can reflect the structural characteristics of the electricity market. The factors that influence price determination in the day-ahead electricity market include the following: load, generation capacity, weather data that affect renewable energy generation and renewable energy generation itself, fuel cost per fuel type and costs due to emissions, the previous price in the electricity market, and weather data that affects the availability and efficiency of generators and power transmission lines.

Energy market prices are closely related to the overall load of the market. Since the cycle of energy consumption is constant, energy market prices also have a similar cycle. They show similar patterns every 24 h, weekly, and every season. In the case of the generators on the peak load, the marginal prices exhibit large fluctuations and have temporary high spikes. Therefore, to forecast energy market prices, previous energy market prices 24 h before or load data 24 and 168 h before are often used. The use of data decomposition techniques to extract valid data can also improve forecasting performance. Among various decomposition methods that aim to find the main characteristics of variables, there have been attempts to forecast univariate time series using time series decomposition. Methods for time series decomposition include the exponential smoothing state space model with Box-Cox transformation, Auto-regressive Integrated Moving Average (ARMA) errors, Trend and Seasonal components (BATS) and Trigonometric Exponential smoothing state space model with Box-Cox transformation, ARMA errors, Trend and Seasonal components (TBATS), Multiple Seasonal-Trend decomposition using Local regression (MSTL) based on Seasonal-Trend Decomposition (STL), Robust Seasonal-Trend Decomposition (RobustSTL), and Seasonal Adjustment of Daily time series (DSA) [5–8]. When variables are decomposed in this way, there have also been attempts to use them not only for univariate forecasting but also for multivariate forecasting [8–10].

On the other hand, if multivariate forecasting is desired, there are many potential variables to consider since various factors simultaneously affect energy market prices. For example, suppose every past time series feature of the exogenous variables is taken as input variables for a certain period. In that case, the curse of dimensionality may occur, reducing the forecasting performance as the scale increases. Therefore, various techniques have been studied to reduce the number of variables, including models that capture the significant characteristics of the data, dimensionality reduction methods, and feature selection [5,11–19].

Normalizing input variables can improve forecasting performances in AI-based models. Previous studies have attempted to improve forecasting by applying normalization or time-varying normalization with the AI method, which reflects the changes in the distribution of input data over time [20,21] or transforms the input data to mitigate noise or spikes [22,23]. For example, Passalis and Nikolaos (2020) proposed a normalization method for cases where data distribution is not following a normal distribution [20]. On the other hand, electricity market prices fluctuate in step-wise changes depending on the large-scale generator and type of fuel source. As the time unit determining the price increases, the demand fluctuation is large, resulting in the price changing discretely. Accordingly, research has been conducted on clustering electricity market prices to forecast them [2,24]. The Korean electricity market uses a cost-based pool, and fuel costs are submitted to the fuel cost evaluation committee of the Korea Power Exchange nine days before the start of

each month to calculate the cost function for each generator. In addition, the Korea Power Exchange publicly discloses the weighted average fuel cost per unit based on the capacity of each generator by the fuel type.

This paper focuses on the high correlation between electricity market prices and the load and the type of fuel source, not just the previous market price. Therefore, the value obtained by decomposing the electricity demand time series is verified to have a close relationship with the electricity market price. It is a feature variable that provides a new signal for forecasting electricity market prices. In addition, due to the characteristics of the electricity market, the distribution of electricity market prices varies depending on the large-scale generator or type of fuel source. It may have a different shape from the normal distribution but will maintain a certain form. Therefore, when scaling the input variables, it is suggested to use fuel cost or the argument of the maximum of the PDF calculated by KDE rather than a normal distribution. The comparison is made between forecasting the electricity market price and the electricity market price made by forecasting the new variables proposed in this paper, then inversely transforming it into the electricity market price.

Section 2 explains the process of the proposed algorithm. Section 2.1 provides an overview of the entire algorithm. Section 2.2 suggests feature generation that uses fuel prices or applies KDE to generate the forecasting label. Section 2.3 proposes a demand decomposition, and Section 2.4 explains the method for selecting the main features for the various features, including decomposed demand. Section 2.5 describes XGBoost, the forecasting model used in this study. In Section 3, the paper conducts a simulation of the South Korean electricity market as a case study. Using the forecasting environment and data described in Section 3.1, it performs and derives results using the performance indices described in Section 3.2. Section 3.3 analyzes the results obtained. Finally, in Section 4, the paper discusses the results and suggests future research directions.

## 2. Proposed Algorithm

### 2.1. Overview of the Methodology

The following algorithm aims to forecast the day-ahead electricity market. The algorithm consists of four main parts: Feature generation, data preprocessing, model training and testing, and post-processing. Figure 1 illustrates the entire process of forecasting, including each of these parts. First, in the Feature generation part, there is the process of decomposition for the demand and feature generation for the label data. Decomposition divides the demand data into five features, trend, seasonal for 24 periods, seasonal for 168 periods, residual, and the original demand data. For feature generation of the label, the generated label variable used for forecasting is categorized into the following variables:  $P$ , which represents the System Marginal Price (SMP) that is the price of the electricity market,  $P_{fuel}$  which represents the difference between SMP and the capacity-weighted average fuel cost per unit, called "fuel cost", for the representative fuel type that is the most frequently used as the fuel source of marginal generators,  $P_{mean}$  which represent the mean point of the  $P$ , and  $P_{kdemax}$  which represents the argument of the maximum density point of the Probability Density Function (PDF), which is calculated by Kernel Density Estimation (KDE), and referred to as "kdeargmax" of  $P$ . For the label variable, feature importance was measured using the previous time-series dataset before the start points of the test period. When using previous price data, two methods were compared to determine the more accurate method, according to containing the previous variables of the two methods on the feature selection process: using the previous price as is and normalizing the price for the previous 7 days (1).  $\mu_T$  represents the mean value of the price for the previous 7 days, and  $\sigma_T$  represents the standard deviation. When comparing the two measurements, it was found that using the previous  $P$  was more accurate when forecasting  $P$ , and using

normalized variable by using the mean and standard deviation of  $P_{fuel}$  for the previous 7 days was more accurate when forecasting  $P_{fuel}$ .

$$P_{n,t} = \frac{P_t - \mu_T}{\sigma_T}, \quad T \in \text{previous 7 days} \tag{1}$$

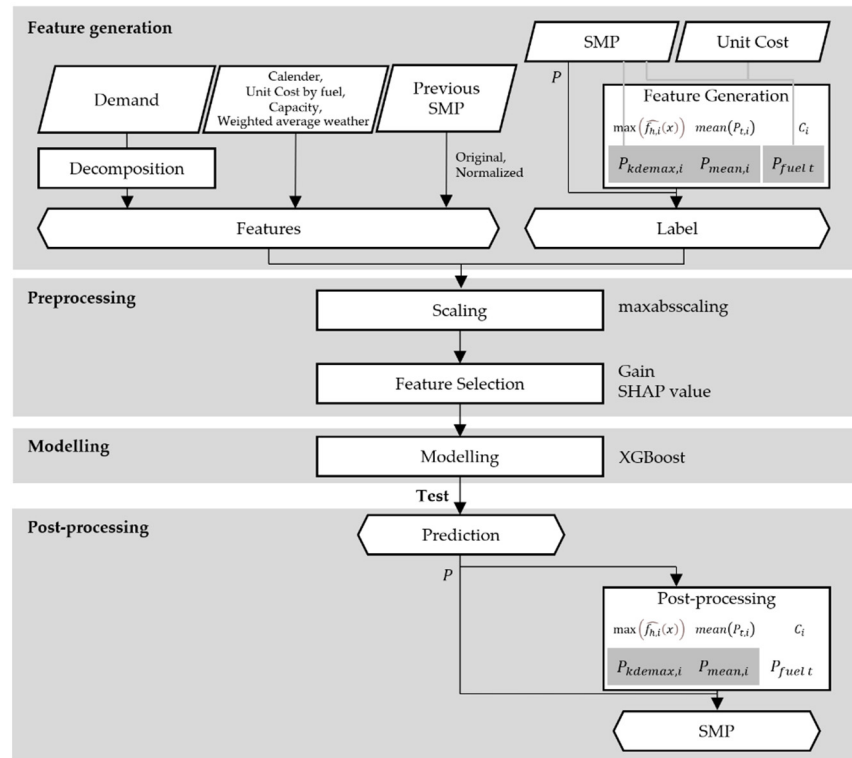


Figure 1. Process of forecasting.

At the data preprocessing part, scaling and feature selection are performed. The demand and the capacity are scaled by the maximum value of each feature. Then the important 10 features are selected by the feature selection process. The detail of the feature selection process is explained in Section 2.4. Finally, selected features are inputs for the forecasting model, which utilizes XGBoost. Subsequently, in post-processing, the label generated from feature generation is inversely transformed into an electricity market price.

## 2.2. Feature Generation of the Labels

### 2.2.1. Kernel Density Estimation

KDE is a type of non-parametric density estimation technique. It has been consistently used for data classification and clustering [25,26]. Density estimation is the process of estimating the PDF of all possible values of a variable based on a set of observed data. Parametric density estimation assumes that the PDF follows a specific distribution, while non-parametric density estimation does not make any assumptions about the distribution. KDE is expressed as (2), where  $\hat{f}_h(x)$  means the PDF calculated by KDE,  $K(x)$  means the kernel function and  $x$  represents the continuous random variable,  $x_i$  represents one observed data point,  $n$  represents the number of the dataset, and  $h$  represents the bandwidth of the kernel function.

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} K\left(\frac{x - x_i}{h}\right), \quad x \in \mathbb{R} \tag{2}$$

The process is as follows: a kernel function is created around each data value, all kernel functions on the data axis are summed, and then the result is divided by the number of data points. The data can be smoothed and represented as continuous values through this process. The shape of the KDE varies depending on the type of kernel function and bandwidth. A smaller bandwidth value leads to greater variability in the KDE. A kernel function refers to a function that satisfies the following where  $u \in \mathbb{R}$  and  $K(u)$  means the kernel function:

1.  $\int_{-\infty}^{\infty} K(u)du = 1$
2.  $K(-u) = K(u)$ , for all values of  $u$
3. *on – negative*

The Gaussian function is a representative kernel function (3).

$$K(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}, u \in \mathbb{R} \tag{3}$$

### 2.2.2. Feature Generation of the Labels and Post-Processing

The system marginal price in the electricity market is determined by the volatility and magnitude of the demand and the operational characteristics of each generator’s fuel source. Therefore, while we cannot know which generator determined the price, we can estimate the fuel source to some extent from the price. Two methods are proposed for generating new features by combining and subtracting other features from SMP.

First, (4) is the new variable  $P_{fuel}$ , which is created by the difference between  $P$ , which means SMP, and the fuel cost by the representative fuel type, which is represented by  $C$ . Figure 2 shows the monthly SMP determination frequency by the fuel types provided by the Korea Power Exchange over 4 years from 2018 to 2021. It shows the total number of SMP decisions and the number of decisions by fuel type. Nuclear has zero decisions and generators based on other more flexible fuel sources determine the SMP. Over 90% of the marginal power generators in 2021 were LNG generators. Therefore, we use  $P_{fuel}$  which is the difference between  $P$  and the fuel cost per unit of LNG. Generators with negative  $P_{fuel}$  values can be expected to use relatively stable demand and cheaper fuel sources, while generators with positive values are expected to use relatively volatile demand and more expensive fuel sources.

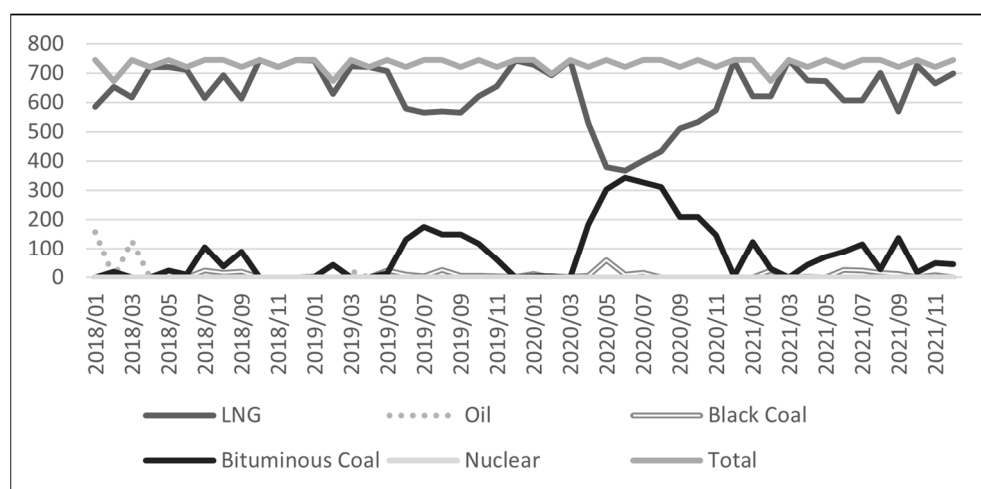


Figure 2. SMP determination frequency by fuel type.

The second method involves estimating the PDF of label data for a specific period, denoted as (5), and introducing a new variable representing the difference between the point that is maximum density and the label, denoted as (6). Here, the period  $I$  is considered arbitrary, and the label can be  $P$  or  $P_{fuel}$ . Since the PDF of the electricity market price may

not follow a normal distribution due to discrete changes in the marginal price depending on the fuel source and the large generator, a non-parametric Kernel Density Estimation (KDE) with a Gaussian function is employed to estimate the PDF. The new variable  $P_{kdemax}$  is created by subtracting the argument of maximum density, referred to as  $kdeargmax$ , from the original label data. In addition, a third method is suggested for comparing, which utilizes the variable,  $P_{mean}$ , representing the difference between the original label and the mean value of the label, expressed as (7).  $P_I$  refers to the dataset of  $P$  for an arbitrary period  $I$ .  $P_t$  means the price on the point  $t$ , while  $T$  represents the number of the time intervals within period  $I$ .  $P_{mean}$  is the normalized value of the Electricity Price obtained by subtracting the mean of the prices from the prices. This  $P_{mean}$  calculates the relative value of the Electricity Price in Electricity Price Forecasting and helps improve the accuracy of forecasting models, often used in previous studies [22].

$$P_{fuel} = P - C_i, \quad i \in Months \quad (4)$$

$$\hat{f}_{h,I}(x) = \frac{1}{T} \sum_{t \in I}^T K_h(x - P_t) \quad (5)$$

$$P_{kdemax} = P - \operatorname{argmax}(\hat{f}_{h,I}(x)) \quad (6)$$

$$P_{mean} = P - \operatorname{mean}(P_I) \quad (7)$$

After using the generated labels for forecasting, it is necessary to convert the labels back to their original values, as follows (8–10).  $\hat{P}_{fuel}$ ,  $\hat{P}_{kdemax}$ ,  $\hat{P}_{mean}$  means the forecasting of  $P_{fuel}$ ,  $P_{kdemax}$  and  $P_{mean}$ , and  $\hat{P}$  is the converted value, which represents the original label intended for forecasting.

$$\hat{P}_{fuel} + C_i = \hat{P} \quad (8)$$

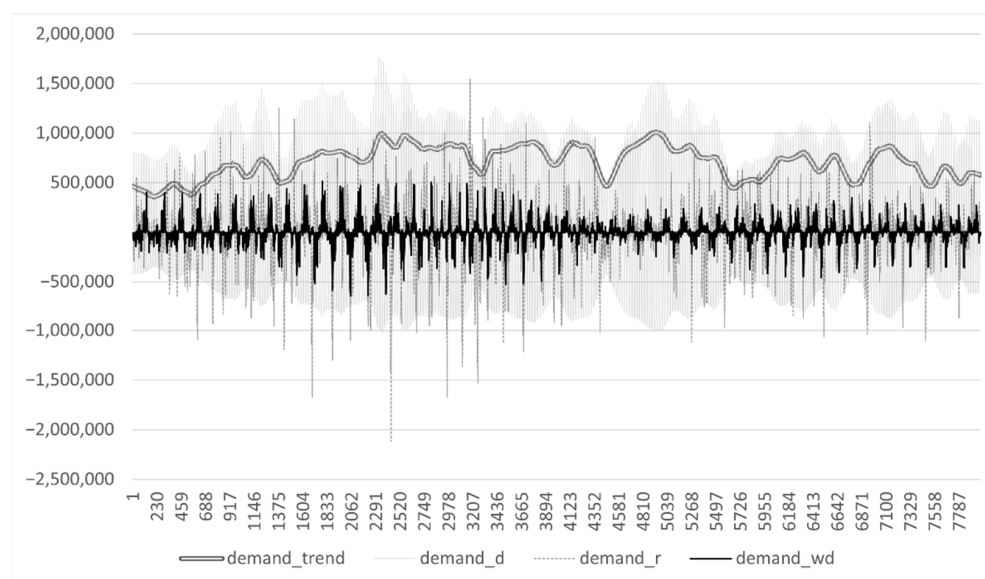
$$\hat{P}_{kdemax} + \operatorname{argmax}(\hat{f}_{h,I}(x)) = \hat{P} \quad (9)$$

$$\hat{P}_{mean} + \operatorname{mean}(P_I) = \hat{P} \quad (10)$$

### 2.3. Decomposition

The electricity demand data is decomposed and used for forecasting. Instead of the univariate forecasting of the decomposed data, multivariate forecasting using the main feature from the decomposition is selected for forecasting. The MSTL decomposition is based on the STL decomposition, which decomposes time series data into the trend, seasonal, and residual components. The MSTL decomposes multiple seasonal components. The preprocessing steps of MSTL are as follows: first, remove values that are less than half of the specified period and replace missing values while normalizing using Box-Cox transformation. For regression, the Loess regression technique is used to determine the weight by reflecting the  $x$ -axis and  $y$ -axis distances between the  $N$  nearest data points based on the data point. The size of  $N$  is determined using the period value in the case of seasonal components.

The MSTL was used for decomposing the demand data into two periods of 24 h and 168 h until the day before the forecasting day. Figure 3 shows the decomposed demand data from 1 January 2021 to 30 November 2021. To ensure rigorous validation, the demand until the day before the forecasting point in each of the 24-h tests is decomposed. The demand data and decomposed data from the previous 24 to 168 h are used for the feature selection process.



**Figure 3.** SMP decomposition output.

#### 2.4. Feature Selection

Using all of the decomposed data for forecasting can lead to an excessively large number of input variables in multivariate forecasting, which can negatively impact the forecasting performance. To address this issue, feature selection is performed. In this paper, two indicators of feature importance are used: the gain of the tree structure and the SHAP value [18,19].

##### 2.4.1. Information Gain

The importance of features in a tree structure can be calculated by Information Gain [27]. Information Gain is based on entropy, which represents the average information content of a random variable. As the probability approaches 1, the entropy becomes smaller, and as it approaches 0, it becomes larger, with a greater emphasis on terms with smaller probabilities. If a node in the decision tree does not split, the entropy is 0. Information Gain represents the difference in entropy when a node is split by each feature. A higher Information Gain means a greater decrease in entropy, indicating that the feature contributes significantly to accuracy. In other words, it means that the feature has a high degree of contribution.

The indicator of the ‘feature\_importances’ provided by the XGB regressor includes five types: weight, which is the number of times the variable was used to split the data at each branching point; cover, which is the weighted average of the number of data points that were split by the variable; total\_cover, which is the sum of the cover values over all branching points; gain, which is the weighted average of the decrease in training loss when using the variable; and total\_gain, which is the sum of the gain values over all branching points. The default feature importance is calculated based on the gain. The explanation for the gain of the XGBoost model will be supplemented in Section 2.5.

##### 2.4.2. SHAP

SHAP value is an eXplainable AI (XAI) that provides mathematical justification for AI based on Shapley Value, representing feature importance [28]. The SHAP value is an indicator that represents the contribution based on the Shapley value, inspired by game theory. The Shapley value calculates the difference in prediction error when comparing predictions based on the use of different features. The SHAP value proposes an indicator that possesses specific desirable properties that explainable AI should have, based on the Shapley value. SHAP proposes various approaches for the SHAP value, such as a kernel-based approach called kernelSHAP and a tree-based approach called TreeSHAP.

### 2.4.3. Average of the Rank of Each Feature Importance

Xgboost's 'feature\_importances\_' includes the weighting of the model for each feature depending on the training data, making it difficult to use as a general measure. Therefore, we use the SHAP value as the second measure of feature importance and the gain made by 'feature\_importances\_'. In this model, we ranked the features based on two important indicators and then selected the feature with the smallest average rank. Since the forecasted accuracy is influenced by the initial set of features, we repeated the process of selecting top features by reducing the number of features by two at a time until we reached a set of 14 features. Then, we used the feature importance of this set as a basis to determine the priority and selected the top 10 values. This takes into account that the Root Mean Squared Error (RMSE) tends to decrease when the number of features determined by gain and SHAP value is around 10 but then increases again or fluctuates.

### 2.5. XGBoost

An ensemble model is a method of adjusting weights for each dataset by forecasting the model in various ways. Bagging is a method of parallel forecasting and complementation while boosting is a sequential method. Bagging combines models using methods such as averaging, weighted averaging, and voting. Boosting increases weights for datasets with poor forecasting performance to perform learning for weak trees. Representative boosting models include Adaboost, Gradient Boost, XGBoost, and Catboost [29–32]. The objective function of the boosting model is expressed as (11). The objective is to minimize the sum of the loss function between,  $y_i$  and  $\hat{y}_i^{(t-1)}$ , and has a penalty function.  $f$  is the space of regression trees.  $y_i$  is the label value of the  $i^{th}$  leaf,  $\hat{y}_i$  is the forecasting by summing up the  $f_k$  that each  $f_k$  corresponds to an independent tree structure  $q$  and leaf weights  $\omega$ , expressed as (12). (13) shows the detail about the penalty function  $\Omega(f_k)$  to the complexity of the tree structure, preventing an excessive increase in the number of leaves  $T$  and the leaf weights  $\omega$ ,  $\gamma$  and  $\lambda$  are arbitrary values [30].

$$\mathcal{L}(\emptyset) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (11)$$

$$\hat{y}_i = \emptyset(x_i) = \sum_{k=1}^K f_k(x_i), \quad f_k \in \left\{ f(x) = \omega_{q(x)} \right\} \\ q : \mathbb{R}^m \rightarrow T, \quad \omega \in \mathbb{R}^T \quad (12)$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2 \quad (13)$$

XGBoost is an ensemble-based decision tree model that uses boosting technique [31]. XGBoost is a model based on Gradient Boosting Machine that enhances speed through parallel computing and constructs an objective function that minimizes not only residuals but also the number or weight of nodes to prevent overfitting. Furthermore, it creates weak trees based on the residuals of the model before the learning stage. In XGBoost, the objective function modified the boosting model to train to learn the residuals.

## 3. Case Study

### 3.1. Simulation Environment

The coding environment and data used were as follows. The decomposition was implemented in R, while other parts were implemented in Python. The 24-h forecasts were performed, and the test period is one month, from 1 December to 31 December 2021. For each forecasting, the model was trained using a 16-week dataset immediately before the test. The raw data used for feature selection was as follows.

#### 3.1.1. Database

As of December 2021, the total generation capacity of market participants in the Korean power system is 126.88 GWh, with nuclear power accounting for 18.32%, bituminous coal



at 29.86%, anthracite coal at 0.32%, oil at 1.64%, and LNG 32.73%. The maximum demand is 90.71 GWh.

To use the data effectively for practical 24-h forecasting, it is necessary to distinguish between data that can be used and data that is difficult to use at the point of forecasting. First, considering the submission time for the renewable energy generation forecasting is 17:00, we used the hourly weather forecasting data for +10 to +33 h, which is the 24-h dataset of the next day available for use at 14:00. Until the day before June 2021, only 3-h periodic weather forecasting data was provided, so it was processed into 1-h intervals using the piecewise linearization. In addition, we derived the weighted average value based on the population of the eight largest cities in South Korea [33]. The weather information data was collected from the Open MET Data Portal from Korean Meteorological Administration.

The monthly fuel cost per unit was obtained from the Electric Power Statistics Information System (EPSIS), and the 24-h demand forecast (one-day-ahead demand forecasting data for the price determination of the Korean Power Exchange) was collected from the Public Data Portal, operated by Ministry of the Interior and Safety. The fuel cost per unit [₩/kWh] provided by the Korea Power Exchange. The fuel cost per unit is evaluated monthly according to the Market Rule, so we used discrete values for each month instead of linear regression values. Additionally, we added calendar data to reflect the time series.

In the electricity market of South Korea, the unit generation cost is calculated and applied monthly. Therefore, when calculating in the training dataset, the KDE and the average value are calculated monthly. However, for the month containing the forecast day, if the forecast day is after the 8th day of the month, the SMP for the previous days of the same month is used, and if it is within the first 7 days of the month, the distribution for the previous 7 days is used, as follow (14) and (15).

As the algorithm described in Figure 1 in Section 2.1, the features and the label used in the case study are shown in Table 1. To add a further explanation, # means the previous hour of the forecasting point. On the use of the previous price, the original price is represented as 'h-#', while the normalized price distribution of the previous seven days is represented as 'hn-#'. In the case of using scaled price as label data, it is represented as 'hp-#', and when normalized, it is represented as 'hpn-#'.

$$P_{kdemax,t} = \begin{cases} P_t - \operatorname{argmax}(\hat{f}_{h,month\ i}(x)), & \text{if } count(P_{t \in month\ i}) \geq 24 \times 7 \\ P_t - \operatorname{argmax}(\hat{f}_{h,t-1,\dots,t-7}(x)), & \text{if } count(P_{t \in month\ i}) < 24 \times 7 \end{cases} \quad (14)$$

$$P_{mean,t} = \begin{cases} P_t - \operatorname{mean}(P_{t \in month\ i}), & \text{if } count(P_{t \in month\ i}) \geq 24 \times 7 \\ P_t - \operatorname{mean}(P_{t-1,\dots,t-7}), & \text{if } count(P_{t \in month\ i}) < 24 \times 7 \end{cases} \quad (15)$$

$$P_{t \in month\ i} = \{P_t \mid P_t \text{ that is the price corresponding to the month } i\}$$

Table 1. Features and the label used in the case study.

Feature Categories	Time Points for Using	Features	Feature Name	
Decomposed demand [MWh]	h <sup>2-24</sup> , h-25, h-48, h-72, h-168	<ul style="list-style-type: none"> <li>Trend</li> <li>Weekly-based seasonal</li> <li>Daily-based seasonal</li> <li>Residual</li> </ul>	<ul style="list-style-type: none"> <li>demand_trend-#<sup>3</sup></li> <li>demand_wd-#</li> <li>demand_d-#</li> <li>demand_residual-#</li> </ul>	
			(original)	(scaled)
Previous price [₩/kWh]	h-24, h-25, h-48, h-72, h-168	<ul style="list-style-type: none"> <li>Original Price</li> <li>7 days Normalized Price</li> </ul>	<ul style="list-style-type: none"> <li>h-#</li> <li>hn-#</li> </ul>	<ul style="list-style-type: none"> <li>hp-#</li> <li>hpn-#</li> </ul>

Table 1. Cont.

Feature Categories	Time Points for Using	Features	Feature Name
Calendar (integer)	h	<ul style="list-style-type: none"> <li>• # of the day</li> <li>• # of the day of the week (1 to 7)</li> <li>• # of the month</li> </ul>	<ul style="list-style-type: none"> <li>• day</li> <li>• day_week</li> <li>• month</li> </ul>
Fuel cost [¥/kWh] Capacity [MWh]	h	<ul style="list-style-type: none"> <li>• bituminous coal</li> <li>• oil</li> <li>• LNG</li> <li>• Total capacity</li> </ul>	<ul style="list-style-type: none"> <li>• price_coal</li> <li>• price_oil</li> <li>• price_lng</li> <li>• cap_coal</li> <li>• cap_oil</li> <li>• cap_lng</li> <li>• cap_total</li> </ul>
Weighted average weather	h	<ul style="list-style-type: none"> <li>• Humid (%)</li> <li>• Temperature [°C]</li> <li>• Wind speed [m/s]</li> </ul>	<ul style="list-style-type: none"> <li>• humid</li> <li>• temp</li> <li>• wind_speed</li> </ul>
price [¥/kWh] <sup>1</sup>	h	<ul style="list-style-type: none"> <li>• <math>P</math></li> <li>• <math>P_{fuel}</math></li> <li>• <math>P_{kdemax}</math></li> <li>• <math>P_{mean}</math></li> </ul>	

<sup>1</sup> The label data. <sup>2</sup> h is the forecasting time point. <sup>3</sup> # means the number of the hour before the forecasting time point.

### 3.1.2. The Case Studies

According to the methodology proposed in this paper, simulations were conducted by the following three case studies.

1. Forecasting  $P$  and  $P_{fuel}$  with or without decomposition data, and each of the four combinations contain the independent feature selection process.
2. Forecasting  $P$ ,  $P_{mean}$ ,  $P_{kdemax}$  and  $P_{fuel}$  with the same feature set that was used for the forecast  $P_{fuel}$  with decomposed demand.
3. Forecasting  $P_{fuel, mean}$  and  $P_{fuel, kdemax}$  using the same feature set as above.

### 3.1.3. Features and Parameters

After decomposing the demand and scaling demand and capacity data, feature importance is calculated for each case. The labels were split into two categories,  $P$  and  $P_{fuel}$ , and feature importance was evaluated separately for each category based on whether decomposition data was included for features or not, resulting in four cases being assessed. The four models can be summarized:

- forecasting  $P_{fuel}$  with decomposed demand
- forecasting  $P_{fuel}$  without decomposed demand
- forecasting  $P$  with decomposed demand
- forecasting  $P$  without decomposed demand

Figures 4–6 show the importance scores for features of each case for each importance indicator. Figure 4 shows the features ascend by the Feature score(F-score), which means a gain of each feature of the forecasting model. Figure 5 shows the features ascending by SHAP value. Figure 6 is the average rank of the two feature importance values. The features finally selected are in Table 2. Hyperparameter tuning was also performed using the grid search method for XGBRegressor, and Table 3. Shows the selected parameters.

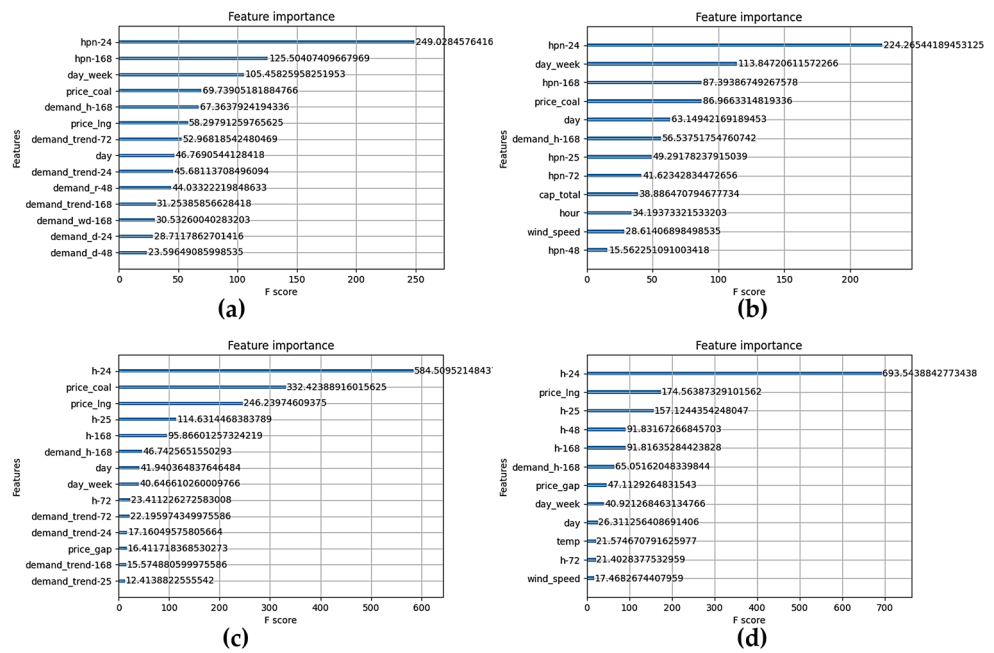


Figure 4. Calculate gains as forecasting (a)  $P_{fuel}$  (c)  $P$ . Without decomposed data, calculate gains as forecasting (b)  $P_{fuel}$  (d)  $P$ .

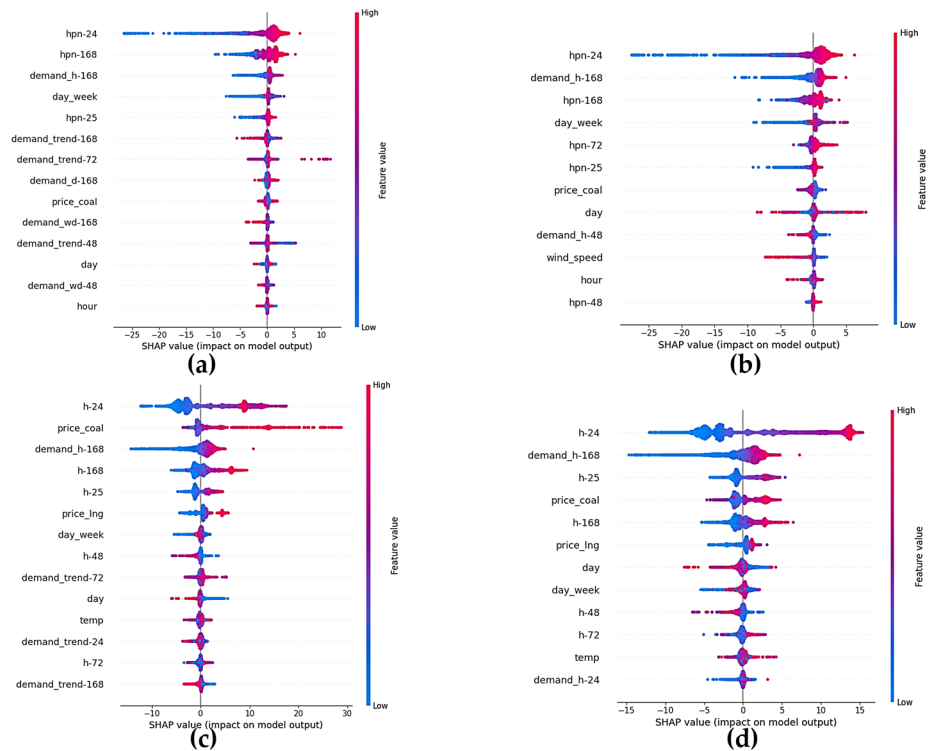


Figure 5. Calculate shapely value as forecasting (a)  $P_{fuel}$  (c)  $P$ . Without decomposed data, calculate gains as forecasting (b)  $P_{fuel}$  (d)  $P$ .

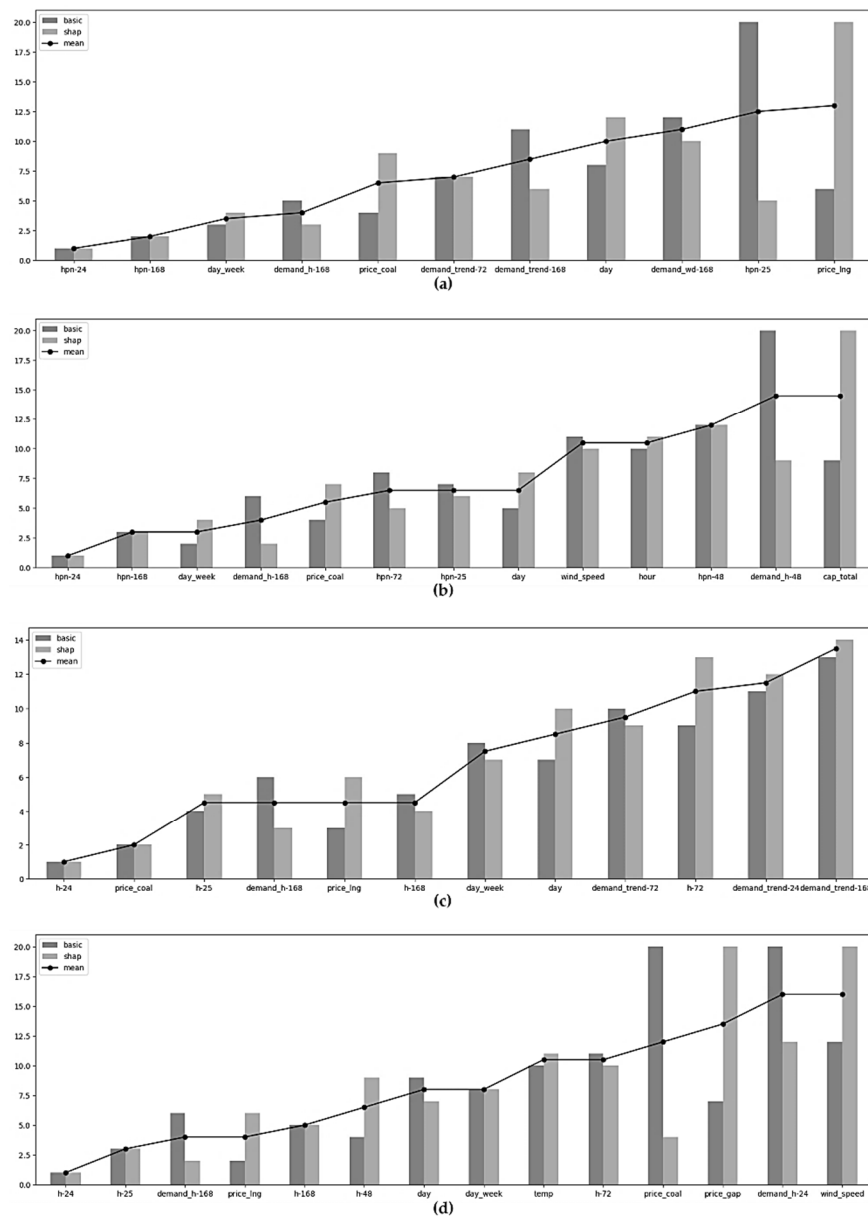


Figure 6. Calculate rank as forecasting (a)  $P_{fuel}$  (c)  $P$ . Without decomposed data, calculate gains as forecasting (b)  $P_{fuel}$  (d)  $P$ .

Table 2. Selected Features for each case at case study 1.

		Categorical	Numerical	Decomposed	Past SMP
$P_{fuel}$	Decomposition	(a) 'day_week', 'day'	'price_coal'	'demand_h-168', 'demand_wd-168', 'demand_trend-168', 'demand_trend-72'	'hpn-24', 'hpn-25', 'hpn-168'
	-	(b) 'day_week', 'day', 'month'	'wind_speed'	'demand_h-168', 'demand_h-48'	'hpn-24', 'hpn-25', 'hpn-72', 'hpn-168'
$P$	Decomposition	(c) 'day_week', 'day'	'price_coal', 'price_lng'	'demand_h-168', 'demand_trend-72'	'h-24', 'h-25', 'h-72', 'h-168'
	-	(d) 'day_week', 'day'	'price_coal', 'price_lng', 'price_gap'	'demand_h-168'	'h-24', 'h-25', 'h-72', 'h-168'

**Table 3.** Selected parameters for the model of XGBRegressor.

Parameters
scale_pos_weight = 0.1, n_estimators = 1000, colsample_bytree = 0.9, learning_rate = 0.01, alpha = 0, gamma = 0.001, max_depth = 15, min_child_weight = 3, objective = 'reg:squarederror', subsample = 0.8

### 3.2. Performance Indices

Error metrics are indicators used to compare accuracy by measuring the error between forecasted and actual values in a dataset. They are commonly used to evaluate accuracy in point forecasting.

#### 3.2.1. Error Metric

In this paper, Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Daily Error (MDE) are used as metrics to measure the magnitude of forecasting errors [34]. Equations (16)–(18) represent each metric.  $P_h$  denotes the actual price for the point of  $h$  of a 24-h test, while  $\hat{P}_h$  represents the forecasted price. The labels generated through feature generation are inversely transformed to represent electricity market prices.

RMSE calculates the square root of the mean of the squared differences between forecasted and actual values. This metric considers the variance of forecasting, allowing for the comparison of accuracy while considering the variance of errors. This paper uses Daily Root Mean Squared Error (DRMSE), which is the average RMSE for 24-h forecasting.

MAPE calculates the average percentage difference between the forecasted and actual values using the absolute values of the differences. The result is multiplied by 100 to express the error as a percentage.

MDE calculates the sum of the absolute difference between the forecasted and actual values, divided by the 24-h mean of the actual values, which is considered for the daily period.  $\overline{P_{24}}$  refers to the daily 24-h mean of the actual values.

$$\text{RMSE} = \sqrt{\frac{\sum_{h=1}^{24} (P_h - \hat{P}_h)^2}{24}} \quad (16)$$

$$\text{MAPE} = \frac{1}{24} \sum_{h=1}^{24} \frac{|P_h - \hat{P}_h|}{P_h} \quad (17)$$

$$\text{MDE} = \frac{1}{24} \sum_{h=1}^{24} \frac{|P_h - \hat{P}_h|}{\overline{P_{24}}} \quad (18)$$

#### 3.2.2. Statistical Test: Diebold-Mariano Test

The D-M test (Diebold-Mariano test) is a widely used indicator for comparing predictive accuracy, which was introduced in 1995 [35,36]. This method is a validation test that determines which of the two forecasting models is superior. The null hypothesis  $H_0$  and alternative hypothesis  $H_1$  in the D-M test are as follows: (19), where  $e_{1t}$  represents the forecasting error of model 1 at time  $t$ ,  $e_{2t}$  is of model 2 for the comparison, and  $g(e)$  represents the function representing the forecasting error. If the null hypothesis is accepted, the loss function values for the forecasting error are equal, indicating no significant difference between the forecasting results of the two models. The significance level is typically set to 0.05. If the  $p$ -value is less than 0.05, it's determined that there is a significant difference between the two forecasting results, and the two models are considered different.

$$\begin{aligned} H_0 &: E[g(e_{1t}) - g(e_{2t})] = 0 \\ H_1 &: E[g(e_{1t}) - g(e_{2t})] \neq 0 \end{aligned} \quad (19)$$

### 3.2.3. Dynamic Time Warping (DTW) Distance

The DTW algorithm is a method for measuring the similarity of the shape of data over time [37]. It has been used to calculate a distance matrix (20) to cluster time series data based on their similarity [38,39]. For two time series  $x$  and  $y$  with lengths  $M$  and  $N$ ,  $D(i, j)$  represents the cumulative distance matrix between  $x_i$  and  $y_j$ . The cumulative value starts at 0 at (1,1), and  $d(x_i, y_j)$  represents the Euclidean distance between  $x_i$  and  $y_j$ , so the first value is equal to  $d(x_i, y_j)$ . The next neighboring values are calculated by adding the Euclidean distance to the previous cumulative value in the matrix [40]. The forecasting could be evaluated using the total minimum distance, the DTW distance calculated by minimizing the path value in the DTW matrix, as shown in (21).  $P$  represents the set of paths formed by the  $D$  matrix, generated by  $x$  and  $y$  with lengths  $M$  and  $N$ .  $p$  represents a single path, and  $L$  is the number of elements corresponding to the path  $p$ . The size of a path is defined as the sum of  $D$  over the number of elements in the path. The DTW distance is the size of the path when it is minimized. The smaller the value, the more similar the time series are. For demand management or management of energy storage resources, it is important to establish operational strategies based on the price patterns of a 24-h market, so this paper calculated the DTW distance.

$$D(i, j) = d(x_i, y_j) + \min \begin{cases} D(i-1, j-1) \\ D(i-1, j) \\ D(i, j-1) \end{cases} \quad (20)$$

$$DTW \text{ distance} = \min \left\{ \sum_{l=1}^L D_{p,l}(i, j), p \in P^{M \times N} \right\} \quad (21)$$

### 3.3. Simulation Results

Tables 4 and 5 means the result. Since the test was conducted for 31 days, each error metric represents the average of the metrics for the 31 days.

Table 4 shows the results of case study 1. It was confirmed that forecasting could be improved by using decomposed data of the previous demand as a feature. The  $p$ -value of the D-M test is  $3.7627 \times 10^{-7}$  for forecasting  $P$  and  $8.2567 \times 10^{-6}$  for forecasting  $P_{fuel}$ . It was also confirmed that the forecasting can be improved by using the fuel cost per unit. The  $p$ -value of the D-M test is  $5.0678 \times 10^{-7}$  when using decomposed data. Combining the use of decomposed data and the use of fuel cost per unit improved forecasting, resulting in the  $p$ -value of  $5.6803 \times 10^{-9}$  in the D-M test.

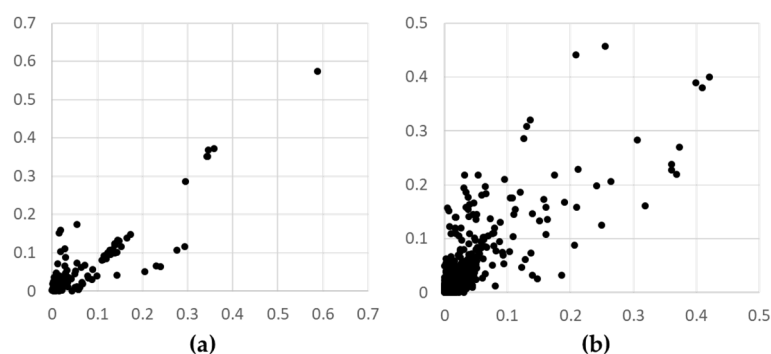
**Table 4.** The comparison of the forecasting on the 1<sup>st</sup> case study.

	$P_{fuel}$	$P_{fuel}$ (No Decomposition)	$P$	$P$ (No Decomposition)
Mean of RMSE	7.1544	7.5764	8.0916	8.3355
Mean of MAPE	3.8308	4.1759	4.5900	4.7564
D-M test statistic	-	-4.4899	-5.0678	-5.8951
[ $p$ -value]	5.0678 [5.0842 $\times 10^{-7}$ ]	[8.2567 $\times 10^{-6}$ ] 2.8786 [4.1096 $\times 10^{-3}$ ]	[5.0842 $\times 10^{-7}$ ] -	[5.6803 $\times 10^{-9}$ ] -5.1268 [3.7627 $\times 10^{-7}$ ]
Mean of MDE	3.4016	3.7335	4.1780	4.1841
Mean of DTW	91.4516	97.7729	118.8496	122.1072

**Table 5.** The comparison of  $P$  forecasting of the 2<sup>nd</sup> case study using the same feature with  $P_{fuel}$ , divided by the test dates.

Test Dates: 12.01–12.31	$P$	$P_{mean}$	$P_{kdemax}$	$P_{fuel}$	$P_{fuel, mean}$	$P_{fuel, kdemax}$
Mean of RMSE	7.7059	7.3693	7.0126	7.1544	6.9925	6.4846
Mean of MAPE	4.4054	4.1295	3.8200	3.8308	3.8681	3.4855
D-M test statistic [ $p$ -value]	−1.6959 [0.0903]	-	5.3007 [1.5225 × 10 <sup>−7</sup> ]	2.2326 [2.5875 × 10 <sup>−2</sup> ]	4.4798 [8.6467 × 10 <sup>−6</sup> ]	7.3816 [4.1985 × 10 <sup>−13</sup> ]
	−3.8738 [1.1660 × 10 <sup>−4</sup> ]	−5.3007 [1.5225 × 10 <sup>−7</sup> ]	-	−0.0884 [0.9296]	−1.1494 [0.2508]	6.3962 [2.8154 × 10 <sup>−10</sup> ]
	−3.7154 [2.1810 × 10 <sup>−4</sup> ]	−2.2326 [2.5875 × 10 <sup>−2</sup> ]	0.0884 [0.9296]	-	−0.3469 [0.7288]	3.2639 [1.1491 × 10 <sup>−3</sup> ]
	−3.6879 [2.4260 × 10 <sup>−4</sup> ]	−4.4798 [8.6467 × 10 <sup>−6</sup> ]	1.1494 [0.2508]	0.3469 [0.7288]	-	8.5620 [6.3485 × 10 <sup>−17</sup> ]
	−6.1395 [1.3485 × 10 <sup>−9</sup> ]	−7.3816 [4.1985 × 10 <sup>−13</sup> ]	−6.3962 [2.8154 × 10 <sup>−10</sup> ]	−3.2639 [1.1491 × 10 <sup>−3</sup> ]	−8.5620 [6.3485 × 10 <sup>−17</sup> ]	-
Mean of MDE	4.0419	3.7479	3.4451	3.4016	3.4896	3.1148
Mean of DTW	117.2682	110.8884	100.7169	91.4516	100.3044	88.5130
12.01–12.07	$P$	$P_{mean}$	$P_{kdemax}$	$P_{fuel}$	$P_{fuel, mean}$	$P_{fuel, kdemax}$
Mean of RMSE	9.6569	11.6741	11.0114	9.2406	10.2170	8.9655
Mean of MAPE	5.4366	6.7260	6.1647	5.0623	5.7129	4.8689
D-M test statistic [ $p$ -value]	3.3877 [8.7907 × 10 <sup>−4</sup> ]	-	3.0695 [2.5028 × 10 <sup>−3</sup> ]	4.8458 [2.8631 × 10 <sup>−6</sup> ]	−4.6920 [5.6043 × 10 <sup>−6</sup> ]	6.3787 [1.6863 × 10 <sup>−9</sup> ]
	2.3822 [1.8336 × 10 <sup>−2</sup> ]	−3.0695 [2.5028 × 10 <sup>−3</sup> ]	-	4.2597 [3.4098 × 10 <sup>−5</sup> ]	4.4040 [1.8920 × 10 <sup>−5</sup> ]	7.0852 [3.7051 × 10 <sup>−11</sup> ]
	−1.4907 [0.1379]	−4.8458 [2.8631 × 10 <sup>−6</sup> ]	−4.2597 [3.4098 × 10 <sup>−5</sup> ]	-	−3.3772 [9.1084 × 10 <sup>−4</sup> ]	0.7559 [0.4508]
	0.8637 [0.3890]	4.6920 [5.6043 × 10 <sup>−6</sup> ]	−4.4040 [1.8920 × 10 <sup>−5</sup> ]	3.3772 [9.1084 × 10 <sup>−4</sup> ]	-	7.5808 [2.2594 × 10 <sup>−12</sup> ]
	−2.2433 [2.6191 × 10 <sup>−2</sup> ]	−6.3787 [1.6863 × 10 <sup>−9</sup> ]	−7.0852 [3.7051 × 10 <sup>−11</sup> ]	0.7559 [0.4508]	−7.5808 [2.2594 × 10 <sup>−12</sup> ]	-
Mean of MDE	4.8351	6.0611	5.5053	4.4443	5.0602	4.2253
Mean of DTW	150.9208	191.8587	169.6297	114.5523	150.8459	122.3397
12.08–12.31	$P$	$P_{mean}$	$P_{kdemax}$	$P_{fuel}$	$P_{fuel, mean}$	$P_{fuel, kdemax}$
Mean of RMSE	7.4111	6.3271	6.0976	6.8896	6.3271	6.0976
Mean of MAPE	4.2184	3.4493	3.2428	3.6474	3.4493	3.2428
D-M test statistic [ $p$ -value]	−4.6369 [4.3802 × 10 <sup>−6</sup> ]	-	5.0688 [5.4061 × 10 <sup>−7</sup> ]	−1.7033 [0.0896]	1.1100 [0.2675]	5.0688 [5.4061 × 10 <sup>−7</sup> ]
	−5.7786 [1.2351 × 10 <sup>−8</sup> ]	−5.0688 [5.4061 × 10 <sup>−7</sup> ]	-	−3.2329 [1.2954 × 10 <sup>−3</sup> ]	−5.0688 [5.4061 × 10 <sup>−7</sup> ]	−0.0522 [0.9584]
	−3.4886 [5.2278 × 10 <sup>−4</sup> ]	1.7033 [0.0896]	3.2329 [1.2954 × 10 <sup>−3</sup> ]	-	1.7033 [0.0891]	3.2329 [1.2954 × 10 <sup>−3</sup> ]
	−4.6370 [4.3802 × 10 <sup>−6</sup> ]	−1.1100 [0.2675]	5.0688 [5.4061 × 10 <sup>−7</sup> ]	−1.7033 [0.0891]	-	5.0688 [5.4061 × 10 <sup>−7</sup> ]
	−5.7786 [1.2351 × 10 <sup>−8</sup> ]	−5.0688 [5.4061 × 10 <sup>−7</sup> ]	0.0522 [0.9584]	3.2329 [1.2954 × 10 <sup>−7</sup> ]	−5.0688 [5.4061 × 10 <sup>−7</sup> ]	-
Mean of MDE	3.8849	3.1118	2.9115	3.2313	3.1118	2.9115
Mean of DTW	109.6864	88.4524	82.9930	90.0540	88.4524	82.9930

Table 5 shows the results of case studies 2 and 3, which are the forecasting using the same features to compare the cases of forecasting  $P_{fuel}$ ,  $P_{mean}$  which is the normalized value obtained by subtracting the mean of the prices often used in the existing Electricity Price Forecasting, and  $P_{kdemax}$ . We used the features used in forecasting  $P_{fuel}$  with decomposed data. We divided the test horizon into two parts: one including data that varies discretely month by month, called “first week of the month,” and the other excluding it, called “rest of the month,” and analyzed the results. When forecasting the first week of the month, the forecasting performance was the best in all indicators when forecasting  $P_{fuel}$ , and the  $p$ -value of the D-M test was within 0.001 except for  $P$  forecasting. When comparing the results with  $P$  forecasting, the  $p$ -value of the D-M test was not significantly better, with 0.1379. However, when using the combination of the fuel cost per unit and argument of the maximum of PDF calculated by KDE to forecast at the beginning of the month, the forecasting performance improved significantly, with a  $p$ -value of 0.02619 compared to  $P$ . The feature generation is performed only by subtracting, not dividing, resulting in the same forecasting for  $P_{kdemax}$  and  $P_{fuel, kdemax}$ , and  $P_{mean}$  and  $P_{fuel, mean}$  at the rest of the month. The performance of forecasting  $P_{kdemax}$ , which has the same value as  $P_{fuel, kdemax}$ , was the best, with a  $p$ -value of within 0.002 in the D-M test compared to using others. Figure 7 compares the performance of the forecasting between using  $P$  and using  $P_{kdemax}$ . It shows a tendency to have fewer errors in the rest of the months, using  $P_{kdemax}$ .



**Figure 7.** The variance of forecasting errors of the case of forecasting  $P$  based on the forecasting error of the case of forecasting  $P_{kdemax}$ , where (a) represents the result of the dates between 12.01 and 12.07, and (b) between 12.08 and 12.31.

To summarize, the following improvements were quantitatively evaluated by forecasting with the proposed techniques:

- The forecasting can be significantly improved by using decomposed data of the previous demand as a feature.
- Combining the use of decomposed data and the use of fuel cost per unit improves forecasting significantly compared to each case.
- Using fuel cost per unit improves forecasting significantly for all periods.
- For forecasting at the beginning of the month, using fuel cost per unit resulted in the most accurate but not significant forecasting. However, when using the argument of PDF calculated by KDE together, the forecasting was significantly improved.
- For forecasting the rest of the month, using  $kdeargmax$  improves the forecasting significantly compared to other methods.

#### 4. Discussion and Conclusions

This paper proposed three methods based on two main approaches for forecasting electricity market prices. As the first approach, it demonstrates that decomposing the previous load and separating fuel prices by generation type in electricity market prices can improve forecasting accuracy. As the second approach, the separation of fuel prices was concretized through the two methods: scaling by the fuel cost and the argument of the maximum value of the PDF calculated by KDE. The forecasting performance was compared



using four indices, RMSE, MAPE, MDE, and DTW, to evaluate the forecasting accuracy. The D-M test was conducted based on MAPE as the criterion to verify the forecasting performance. The three methods improved the forecasting accuracy. We confirmed the improvement in forecasting accuracy for both methods through case studies. When using the decomposed data, the forecasting was improved. When calculating the distribution of electricity market prices without including the previous month's data, scaling with *kdeargmax* performed the most accurately. For the early month, with large fluctuations in electricity market price due to discrete changes in monthly fuel costs, scaling with fuel cost resulted in the smallest error. For the early month, when forming a distribution by subtracting the fuel cost from the price of that month and subtracting the *kdeargmax* of the previous month, then scaling with the calculated *kdeargmax* value, which is a mixture of fuel cost and *kdeargmax*, the forecasting was most improved.

The decomposition of the demand was specified using seasonal-trend decomposition. Then, the demand was decomposed into several signals using decomposition and used for forecasting. Decomposed features were treated equally with other features, and major past decomposed load data was used through feature selection. To show the effectiveness of the decomposed data, the feature importance and Shapley value of the decomposed data and the original data with all features were compared, and the decomposed data was shown to improve forecasting accuracy.

When information on the fuel cost per unit is publicly available, using it can significantly improve forecasting accuracy. Even in cases where fuel costs are not directly available, a new variable called "*kdeargmax*" was proposed. The use of *kdeargmax* in forecasting *P* was expected to improve forecasting accuracy. Compared to scaling with the mean of *P*, scaling with the *kdeargmax* of *P* resulted in improved performance. It seems to be because electricity market prices are determined discretely due to the different fuel types of each power plant and the large-scale generators, even when using the same fuel type. The separation of fuel types at generation occurs when subtracting the argument of the maximum value of the PDF calculated by KDE from the price for cases where it takes positive, negative, or values close to 0.

To forecast the electricity market price, this paper used the characteristics that the demand is closely related. The forecasting can be improved by normalizing the electricity market price and showing improved performance. The results of feature selection confirmed that past demand and past electricity market prices are important features. In a further study, forecasting by designing a separate algorithm instead of the feature selection process might be compared. Forecasting could be improved by splitting the dataset based on price or by forecasting the type of fuel sources and large-scale generators based on price and using them to forecast electricity market prices. The accumulation of forecasting errors should be considered by classifying past prices.

**Author Contributions:** Conceptualization and methodology, A.J. and D.L.; software, investigation, data curation, validation, formal analysis, writing—original draft preparation and visualization, A.J. resources, writing—review and editing, and funding acquisition, J.-B.P. and J.H.R. supervision, J.H.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE), grant number 20204010600220, and by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE), grant number 20194310100060.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This work was supported by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20204010600220). This work was partly supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry, and Energy (MOTIE), Republic of Korea, under Grant 20194310100060.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the study's design; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Gao, C.; Bompard, E.; Napoli, R.; Wan, Q.; Zhou, J. Bidding Strategy with Forecast Technology Based on Support Vector Machine in the Electricity Market. *Phys. A Stat. Mech. Its Appl.* **2008**, *387*, 3874–3881. [[CrossRef](#)]
2. Weron, R. Electricity Price Forecasting: A Review of the State-of-the-Art with a Look into the Future. *Int. J. Forecast.* **2014**, *30*, 1030–1081. [[CrossRef](#)]
3. Stoft, S. *Power System Economics: Designing Markets for Electricity*; IEEE Press: New York, NY, USA, 2002.
4. Kirschen, D.S.; Strbac, G. *Fundamentals of Power System Economics*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2018.
5. Petropoulos, F.; Apiletti, D.; Assimakopoulos, V.; Babai, M.Z.; Barrow, D.K.; ben Taieb, S.; Bergmeir, C.; Bessa, R.J.; Bijak, J.; Boylan, J.E.; et al. Forecasting: Theory and Practice. *Int. J. Forecast.* **2022**, *38*, 705–871. [[CrossRef](#)]
6. Ollech, D. Seasonal Adjustment of Daily Time Series. *Dtsch. Bundesbank Discuss. Pap.* **2018**, *41*. [[CrossRef](#)]
7. Bandara, K.; Hyndman, R.J.; Bergmeir, C. MSTL: A Seasonal-Trend Decomposition Algorithm for Time Series with Multiple Seasonal Patterns. *Int. J. Oper. Res.* **2021**. [[CrossRef](#)]
8. Cleveland, R.B.; Cleveland, W.S.; McRae, J.E.; Terpenning, I. STL: A Seasonal-Trend Decomposition Procedure Based on Loess. *Off. Stat.* **1990**, *6*, 3–73.
9. Gao, T.; Niu, D.; Ji, Z.; Sun, L. Mid-Term Electricity Demand Forecasting Using Improved Variational Mode De-composition and Extreme Learning Machine Optimized by Sparrow Search Algorithm. *Energy* **2022**, *261*, 125328. [[CrossRef](#)]
10. Bandara, K.; Bergmeir, C.; Hewamalage, H. LSTM-MSNet: Leveraging Forecasts on Sets of Related Time Series with Multiple Seasonal Patterns. *IEEE Trans. Neural. Netw. Learn. Syst.* **2021**, *32*, 1586–1599. [[CrossRef](#)] [[PubMed](#)]
11. Jurgen, A. Doornik Autometrics. In *The Methodology and Practice of Econometrics*; Castle, J., Shephard, N., Eds.; Oxford University Press: Oxford, UK, 2009; pp. 88–121. [[CrossRef](#)]
12. Zou, H.; Hastie, T. Regression Shrinkage and Selection via the Elastic Net. *R. Stat. Society. Ser. B (Stat. Methodol.)* **2005**, *67*, 301–320. [[CrossRef](#)]
13. Tibshirani, R. *Regression Shrinkage and Selection via the Lasso*; Wiley: Hoboken, NJ, USA, 1996; Volume 58.
14. Elliott, G.; Gargano, A.; Timmermann, A. Complete Subset Regressions. *J. Econom.* **2013**, *177*, 357–373. [[CrossRef](#)]
15. Aue, A.; Norinho, D.D.; Hörmann, S. On the Prediction of Stationary Functional Time Series. *J. Am. Stat. Assoc.* **2015**, *110*, 378–392. [[CrossRef](#)]
16. Kelly, B.; Pruitt, S. Market Expectations in the Cross-Section of Present Values. *J. Financ.* **2013**, *68*, 1721–1756. [[CrossRef](#)]
17. Box, G.E.P.; Cox, D.R. An Analysis of Transformations. *Source J. R. Stat. Society. Ser. B (Methodol.)* **1964**, *26*, 211–252. [[CrossRef](#)]
18. Effrosynidis, D.; Arampatzis, A. An Evaluation of Feature Selection Methods for Environmental Data. *Ecol. Inform.* **2021**, *61*, 101224. [[CrossRef](#)]
19. Zhang, R.; Nie, F.; Li, X.; Wei, X. Feature Selection with Multi-View Data: A Survey. *Inf. Fusion* **2019**, *50*, 158–167. [[CrossRef](#)]
20. Passalis, N.; Tefas, A.; Kannianen, J.; Gabbouj, M.; Iosifidis, A. Deep Adaptive Input Normalization for Time Series Forecasting. *IEEE Trans. Neural. Netw. Learn. Syst.* **2020**, *31*, 3760–3765. [[CrossRef](#)]
21. Ogasawara, E.; Martinez, L.C.; De Oliveira, D.; Zimbrão, G.; Pappa, G.L.; Mattoso, M. Adaptive Normalization: A Novel Data Normalization Approach for Non-Stationary Time Series. In Proceedings of the International Joint Conference on Neural Networks, Barcelona, Spain, 18–23 July 2010.
22. Uniejewski, B.; Weron, R.; Ziel, F. Variance Stabilizing Transformations for Electricity Spot Price Forecasting. *IEEE Trans. Power Syst.* **2018**, *33*, 2219–2229. [[CrossRef](#)]
23. Nayak, S.C.; Misra, B.B.; Behera, H.S. Impact of Data Normalization on Stock Index Forecasting. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **2014**, *6*, 257–269.
24. Guo, J.J.; Luh, P.B. Selecting Input Factors for Clusters of Gaussian Radial Basis Function Networks to Improve Market Clearing Price Prediction. *IEEE Trans. Power Syst.* **2003**, *18*, 665–672. [[CrossRef](#)]
25. Aliamiri, A.; Stalnakar, J.; Miller, E.L. Statistical Classification of Buried Unexploded Ordnance Using Nonpara-metric Prior Models. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 2794–2806. [[CrossRef](#)]
26. Van Hulle, M.M. Density-Based Clustering with Topographic Maps. *IEEE Trans. Neural. Netw.* **1999**, *10*, 204–207. [[CrossRef](#)] [[PubMed](#)]
27. Tangirala, S. Evaluating the Impact of GINI Index and Information Gain on Classification Using Decision Tree Classifier Algorithm. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 612–619. [[CrossRef](#)]
28. Lundberg, S.M.; Allen, P.G.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4768–4777.
29. Natekin, A.; Knoll, A. Gradient Boosting Machines, a Tutorial. *Front. Neurorobot.* **2013**, *7*, 21. [[CrossRef](#)]
30. Freund, Y.; Schapire, R.E. Experiments with a New Boosting Algorithm DRAFT-PLEASE DO NOT DISTRIBUTE. Experiments with a new boosting algorithm. *icmi* **1996**, *96*, 148–156.

31. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794.
32. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased Boosting with Categorical Features. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 6638–6648.
33. Kwon, B.S.; Park, R.J.; Song, K. Bin Short-Term Load Forecasting Based on Deep Neural Networks Using LSTM Layer. *J. Electr. Eng. Technol.* **2020**, *15*, 1501–1509. [[CrossRef](#)]
34. Misiolek, A.; Trueck, S.; Weron, R. Point and Interval Forecasting of Spot Electricity Prices: Linear vs. Non-Linear Time Series Models. *Stud. Nonlinear Dyn. Econom.* **2006**, *10*, 3. [[CrossRef](#)]
35. Diebold, F.X.; Mariano, R.S. Comparing Predictive Accuracy. *J. Bus. Econ. Stat.* **2002**, *20*, 134–144. [[CrossRef](#)]
36. Harvey, D.; Leybourne, S.; Newbold, P. Testing the Equality of Prediction Mean Squared Errors. *Int. J. Forecast.* **1997**, *13*, 281–291. [[CrossRef](#)]
37. Senin, P. *Dynamic Time Warping Algorithm Review*; Information and Computer Science Department, University of Hawaii: Honolulu, HI, USA, 2008.
38. Hsu, H.H.; Yang, A.C.; Lu, M. Da KNN-DTW Based Missing Value Imputation for Microarray Time Series Data. *J. Comput.* **2011**, *6*, 418–425. [[CrossRef](#)]
39. Shen, S.K.; Liu, W.; Zhang, T. Load Pattern Recognition and Prediction Based on DTW K-Medoids Clustering and Markov Model. In Proceedings of the IEEE International Conference on Energy Internet, ICEI 2019, Nanjing, China, 27–31 May 2019; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2019; pp. 403–408.
40. Laperre, B.; Amaya, J.; Lapenta, G. Dynamic Time Warping as a New Evaluation for Dst Forecast With Machine Learning. *Front. Astron. Space Sci.* **2020**, *7*, 39. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.