Tutorial on time series prediction using 1D-CNN and BiLSTM: A case example of peak electricity demand and system marginal price prediction

Jaedong Kim [a], Seunghwan Oh [b], Hee-soo Kim [a], Woosung Choi [c,d *]

[a] *Power Generation Laboratory, KEPCO Research Institute, Daejeon, Republic of Korea*
[b] *DAPADA Inc., Suwon-si, Gyeonggi-do, Republic of Korea*
[c] *R&D Strategy Office, KEPCO Research Institute, Daejeon, Republic of Korea*
[d] *Electric Power Research Institute, Charlotte, NC 28262, USA*
[*] *Corresponding author*
E-mail addresses: zest.woosung@kepco.co.kr, WChoi@guestresearcher.epri.com

Abstract

Although research on time series prediction based on deep learning is being actively carried out in various industries, deep learning technology still has a high entry barrier for researchers who have not majored in computer science. This paper presents a tutorial on time series prediction using a deep learning-based model. The entire process of time series data prediction is presented—from data collection to evaluation of prediction results. The details of each step are shown through a case example of predicting peak electricity demand and system marginal price of Jeju Island in Korea using the 1D-CNN and BiLSTM model. To make it easier for readers to follow, the example uses only open public data, and the entire Python source code is shared via a GitHub repository. This tutorial is not limited to the energy industry but can be utilized for any application requiring time-series data prediction. This article is expected to be of great help to researchers who need to understand the process of time series prediction using deep learning and use it for application in their industry.

Keywords: Time series prediction, Peak electricity demand, System marginal price, 1D-CNN, BiLSTM, Tutorial

1. Introduction

With the rise in Industry 4.0 technologies such as big data and artificial intelligence, many industries are witnessing the rapid digitization of their manual tasks. In addition, a huge amount of data being captured via IoT is being stored in the cloud. When such data is collected for a certain period over time, it is called time-series data. While the concept of time series data itself is not new, long-term historical data has not been actively used until recently due to computational and technical limitations. However, as storage space and computational performance increase, time-series data analysis began to develop rapidly.

The main purpose of analyzing time-series data is to predict data for the future using historical data. In the past, there have been many attempts to predict time series data using stochastic and conventional machine learning approaches to predict features related to energy, such as wind speed, wind power, solar power, price, energy consumption, and so on [1–11]. However, as the predictive performance of deep learning algorithms has rapidly developed, research using such algorithms is also progressing rapidly[12–29], wherein most studies have used RNN-based models [19,20,24,26–29]. To create robust prediction models, domain knowledge of each industry is essential, and a process of sorting out necessary and non-necessary features is required. However, due to the initial entry barrier of deep learning technology, researchers who are not familiar with it face difficulty applying it. Many research articles apply deep learning in various fields; however, it is difficult to find a paper that explains the process in detail so that it can be easily adopted for other applications. This paper aims to present a guide and tutorial to beginners for a time series data prediction method using deep learning to apply the technology in their fields.

Explaining using a real-world example helps to understand the concepts more clearly. In this paper, the prediction process is explained through an example for predicting the peak electricity demand and system marginal price (SMP) of Jeju Island in Korea. For a stable power supply, it is important to predict demand in advance and maintain a proper reserve margin level. In particular, as the proportion of renewable energy on Jeju Island increases, the energy supply instability also increases. Therefore, more accurate prediction techniques are required. In this example, peak electricity demand and SMP were predicted by selecting features that considered regional characteristics such as climate features, number of tourists, and holiday information. A BiLSTM network was selected as the base model, improving upon the structure of conventional RNN, and a 1D-CNN was concatenated in front of the network for feature extraction.

It must be mentioned that while this paper uses an example from the energy industry for time series data prediction, the concepts discussed herein can be applied to any other similar application requiring time series data prediction.

The rest of the paper is organized as follows. In Section 2, 1D-CNN and BiLSTM neural networks used in the prediction model are explained briefly. In Section 3, the overall time series prediction process is introduced. In Section 4, the main part of this paper, detailed explanations are provided for each process—from data collection to prediction result evaluation—through case examples using real-world data. Finally, Section 5 concludes the work and discusses the scope for further study.

2. 1D-CNN and BiLSTM

In deep learning, the most popular and well-known algorithms with high performance are CNN and RNN [30–32]. Among the various RNN structures available, BiLSTM is the latest one. In recent studies related to time series prediction, prediction models with CNN or BiLSTM algorithms or algorithms that concatenate the two have been widely used. Those studies show higher performance than conventional statistical or machine learning models [33–36]. The basic theoretical background of those networks is explained in this section.

2.1 1D-CNN

The basic architecture of CNN cannot be applied for usual time series data prediction since the CNN structures are 2D-CNN, which only take 2D inputs. Therefore, the conventional 2D-CNN architecture is not directly applicable for 1D signal prediction. Some studies have converted 1D signals into 2D images to directly use the 2D-CNN architectures [37–41]. Such an approach may be useful in some application, but in most common cases, it increases the computational costs and decrease efficiency.

Apart from 2D-CNN, 1D-CNN has been developed and utilized in various applications [42–46]. Fig. 1 shows the basic structure of 1D-CNN. The significant advantages of 1D-CNN are that it requires much less computational complexity and time than 2D-CNN and takes 1D signal directly without 2D conversion. Due to these advantages, there are many studies and applications of 1D-CNN in various fields such as fault detection of rotating machinery [42], structural damage detection [44], real-time electrocardiogram monitoring [45], etc.

Using 1D-CNN, correlational properties of multivariate signals can be extracted without additional feature engineering.

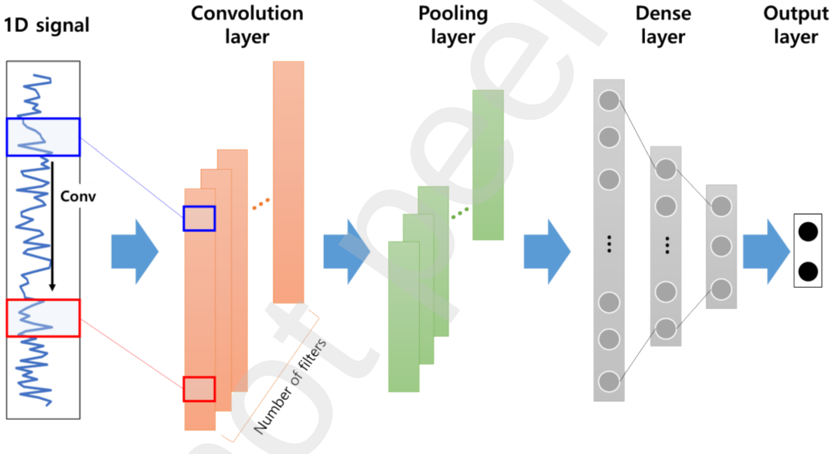


Fig. 1. A sample structure of 1D-CNN classification.

2.2 BiLSTM

Schuster et al. [47] first proposed bidirectional RNN (BRNN) to overcome the limitations of the conventional RNN. To construct a model that can be trained in both forward and backward directions, Schuster added backward layers, which represent a reversed copy of the input sequence. BiLSTM is the network that replaces the conventional RNN cells with LSTM cells in BRNN structures. Using these LSTM cells, the fundamental long-term dependency problem of conventional RNN has been resolved. Recent research has proved that the performance of BiLSTM is higher than the standard LSTM and conventional machine learning algorithms in time series data prediction [48–50]. The architecture of BiLSTM is shown in Fig. 2.

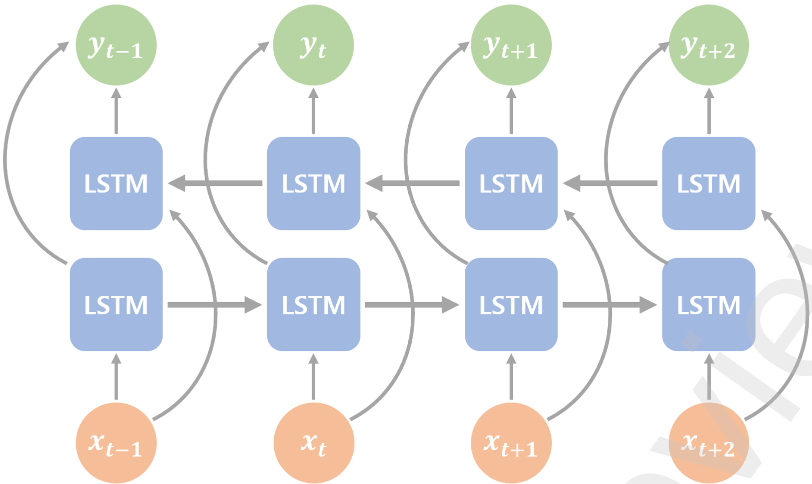


Fig. 2. Architecture of BiLSTM.

2.3 1D-CNN + BiLSTM

By concatenating 1D-CNN and BiLSTM, multivariate 1D time-series signals can be predicted with high performance. Several related studies have utilized CNN and LSTM networks together. Shi et al.[35] used the ConvLSTM model for precipitation nowcasting with 2D radar echo datasets. Sainath et al.[36] constructed an architecture by combining CNN, LSTM, and DNN to compare word error rates with English-spoken utterances. They found that CLDNN provides a 4-6% relative improvement in WER over a normal LSTM. From 2019, using a combination of networks caught the interest of researchers, and a variety of studies have used them for diverse applications, including stock price prediction [33], residential energy consumption [34], sentiment analysis [51], and life prediction of machinery components [52]. These studies mentioned used CNN and LSTM or BiLSTM for analysis or prediction and found better results than in the case of using a single CNN or LSTM architecture.

In this paper, the use of a concatenated model of 1D-CNN and BiLSTM is demonstrated for peak electricity demand and SMP prediction. The model architecture we used is shown in Fig. 3.

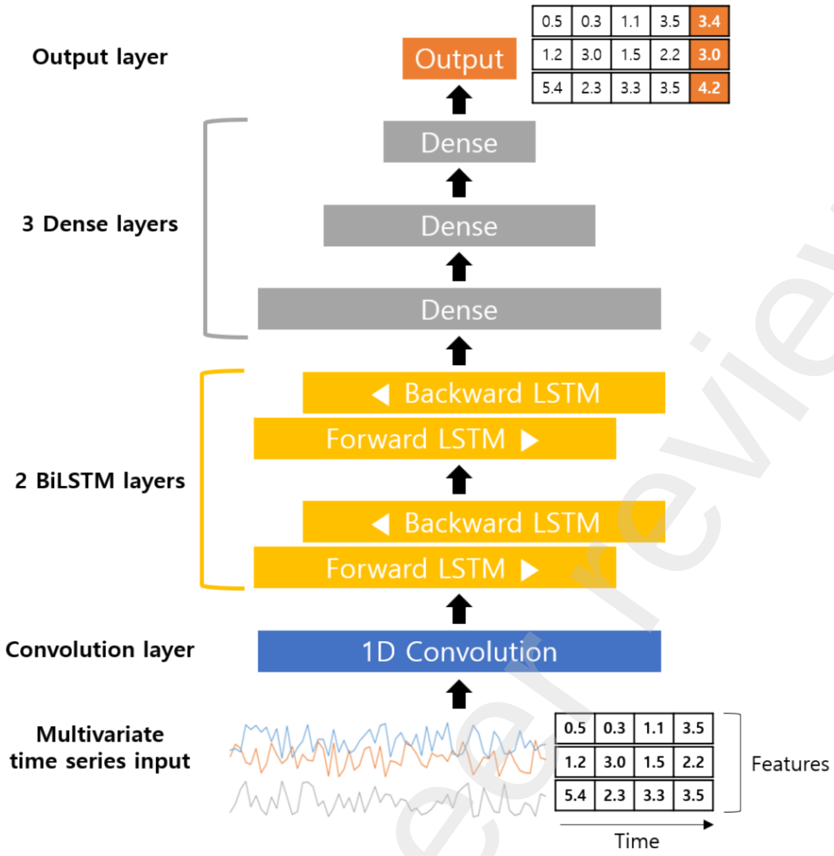


Fig. 3. Sample structure of 1D-CNN + BiLSTM prediction model.

3. Time series prediction procedure

In this article, data analysis and prediction procedure most suitable for generally used time series data in the energy field will be introduced. This overall procedure includes details from data preparation to final prediction and evaluation. The suggested procedure for time series prediction is illustrated in Fig. 4.

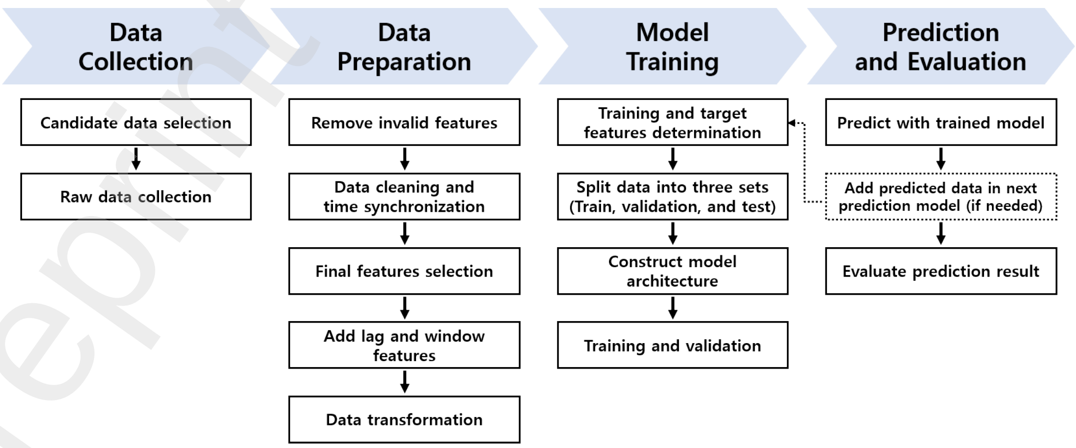


Fig. 4. The overall procedure for time series prediction.

There are four steps in this procedure: data collection, data preparation, model training, and prediction and evaluation. Brief descriptions for each step are as follows:

Data collection involves the selection and collection of raw data. This step requires strong domain knowledge to build a robust prediction model. If you are not familiar with the field you are trying to analyze, you may have to ask for help from related domain experts. After you choose the candidate data, you need to collect the data. The data type can be largely divided into open and non-open data. Open data includes freely available data that everyone can use and republish without restrictions [53]. On the contrary, non-open data has restricted access and limit the ability to reuse the data without permission. Only open data were used in this paper to demonstrate how to collect data.

Data preparation involves acquiring available data and processing it as valid data set. Ideally, collected time series data should have one valid data in one timestamp. In the real world, however, most data must be pre-processed for analysis since they contain a lot of invalid data. On the processed dataset, additional filtering should be performed in this step. The features in the dataset may all be usable, but unnecessary data may curtail prediction performance. Therefore, some features may be filtered in this step based on a logical decision, domain knowledge, and correlation of features. In addition, time series characteristics can be added by including lag and window features. Both features are widely used to analyze time series data and are reportedly effective in many problems. In addition, in the case of categorical features, it is necessary to convert the data into available data using techniques such as one-hot encoding or label encoding, etc.

Model training involves the construction and training of a robust prediction model. Herein, training includes validation since it is essential in creating a robust model. To train the model, the data frame created from feature engineering is split into three sets: training, validation, and test. The data is generally split as 70% for training, 20% for validation, and 10% for test. During training, the prediction loss of the validation set is calculated using the current model status. This validation set does not affect the training process. Therefore, it is possible to verify whether the model is robust for independent data not participating in training. Although the training loss converges, if the validation loss does not do so, the model is determined to suffer from overfitting or underfitting. Widely used validation methods include the hold-out validation, k-fold validation, and leave-one-out cross-validation (LOOCV) [54–56]. In addition, many different loss functions can be used in the training and validation processes, such as MAE, MAPE, and MASE. When the loss of the model is satisfactory, the prediction model can be used for time series prediction.

Prediction and evaluation constitute the final step and involve predicting time series data and evaluating the results. Predictions are performed using the model created in the previous step. In some cases, predicted features can be used to predict other features, which is the final target feature set. To do this, update the dataset with predicted values and create a new model to perform further predictions, as shown in Fig. 5. The predicted result should be evaluated through evaluation metrics such as MSE, MAPE, MASE, etc.

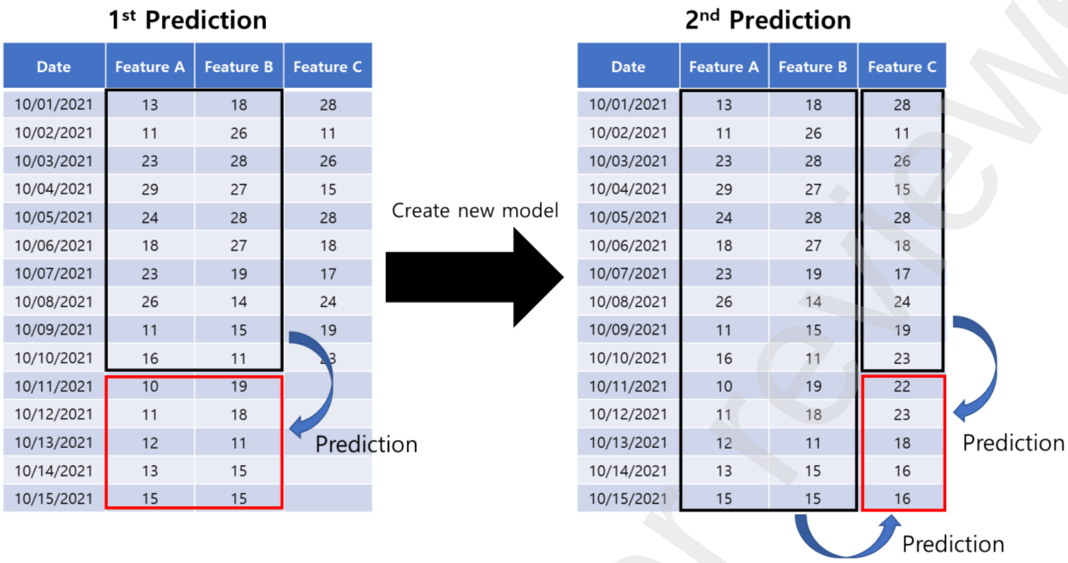


Fig. 5. Feature prediction based on previously predicted data.

To provide a clearer understanding, a case example with a more detailed explanation of each step is included in the next section.

4. Tutorial: A case example

4.1 Problem description

Since electricity is typically consumed as it is produced, it is necessary to accurately predict peak demand in advance for a stable power supply. In the short term, overestimating power demand will increase prices in the power market and increase demand management costs. Conversely, underestimating will lead to an unstable power supply and increase the settlement cost of additional limited power generation in the power market. In addition, predicting the SMP can be beneficial for power generation market participants as well as power transmission companies, leading to stability in the electricity price. Especially, accurately predicting SMP is even more important in present times of increasing demand for decarbonization and distributed power generation.

For this reason, in this example, forecasting of peak electricity demand and SMP for Jeju Island in Korea from public data was demonstrated. This example will clarify the overall procedure of time series data prediction since this region is a tourist spot. The SMP for it is determined separately from that for the Korean mainland. Moreover, as shown in Fig. 6, the proportion of renewable energy in the total power generation capacity of Jeju Island is gradually increasing; in 2021, 67% of the total power generation capacity was represented by new and renewable energy. As the dependence on renewable energy increases, the need for electricity demand prediction increases simultaneously.

The prediction model was created using data from February 1st, 2018, to May 18th, 2020. Using this prediction model, peak electricity demand and SMP for May 19th, 2020, to June 8th, 2020, were predicted.

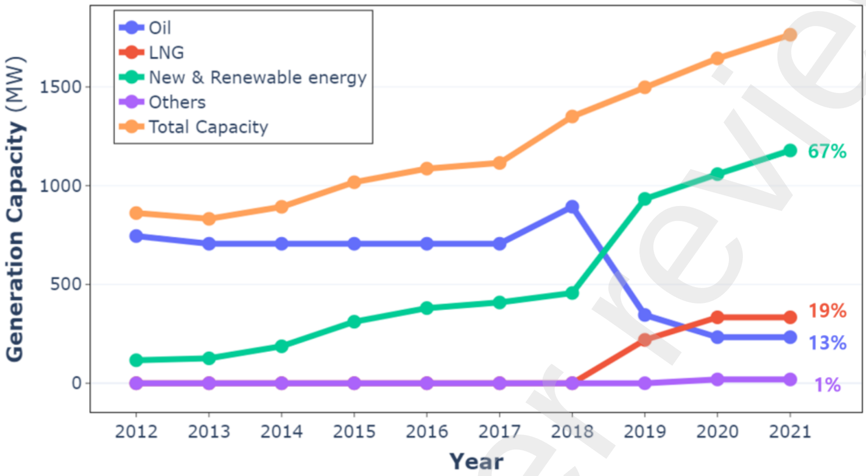


Fig. 6. Generation capacity by fuel for Jeju Island (2012–2021).

4.2 Data collection

4.2.1 Candidate data selection

The first step is selecting possible candidate data. First, historical data of target features, peak electricity demand, and SMP, in this case, should be used as training features. Then, selecting new features with domain knowledge is required. In general, electricity demand is known to be affected by the weather, while SMP is directly related to fuel costs; thus, these two main features were selected accordingly. Regional characteristics were also taken into account. Jeju Island is the most popular tourist spot in Korea, so the number of tourists may considerably affect energy consumption. In addition, holidays and weekends may affect energy consumption. Thus, these two features must be accounted for as well. Table 1 lists the six kinds of properties that were selected as possible training features in this step.

Table 1. List of candidate features.

#	Candidate feature
1	Peak electricity demand
2	System marginal price
3	Weather information
4	Fuel cost
5	National holiday information of South Korea
6	The number of tourists

In addition to the above data, if data related to social, environmental, or economic issues are used together, the prediction accuracy can be increased.

4.2.2 Raw data collection

In this example, two types of data collection methods were used, as shown in Fig. 7: file downloading and crawling. All the data used in this example are accessible data open to the public. We used these two data collection methods since they are the most common and generally used options. However, different methods can be used to deal with other data sources, such as a DB system or an in-house data format. The process details for gathering the six kinds of data will be demonstrated next. As mentioned above, all the data we used are open to the public so that the reader can follow all steps.

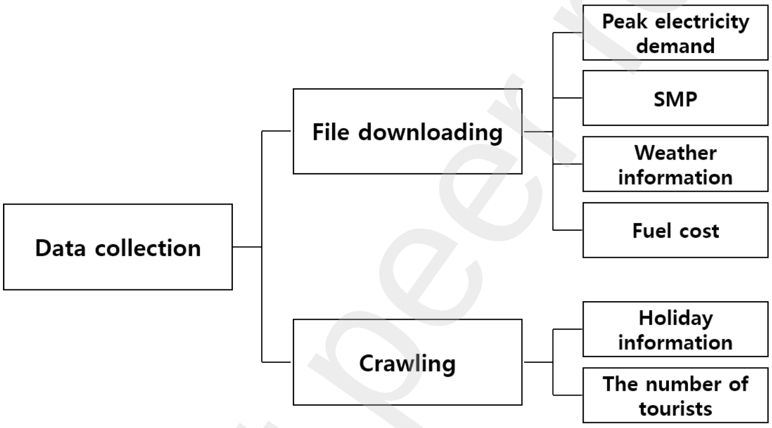


Fig. 7. Data collection methods for candidate data.

4.2.2.1 Peak electricity demand and system marginal price

Peak electricity demand and SMP are the most significant data since these are target features as well as training features. The historical data on peak electricity demand for Jeju Island is freely available on the Korea Power Exchange website (https://new.kpx.or.kr/powerDemandPerformJeju.es?mid=a10606070000). On this website, the daily peak demand and other relevant data for Jeju Island can be downloaded. SMP historical data for Jeju Island is also freely available on the Korea Power Exchange website (https://new.kpx.or.kr/smpJeju.es?mid=a10606080200&device=pc). In Korea, SMP values for mainland and Jeju Island are calculated separately. SMP is determined every hour, so hourly SMP data is provided. However, the prediction interval of this problem is a day and not an hour. Therefore, hourly recorded data were not used, and daily maximum, minimum, and mean values were calculated and used.

The peak electricity demand and SMP data trends for all periods are shown in Fig. 8. The electricity demand has a seasonal trend with increased demand in summer and winter and decreased demand in spring and autumn. On the contrary, SMP fluctuates from day to day. Even though this is a real-world value, such outliers should be eliminated to create a stable prediction model since our objective focuses on predicting features of usual days, not unusual and special occasions.

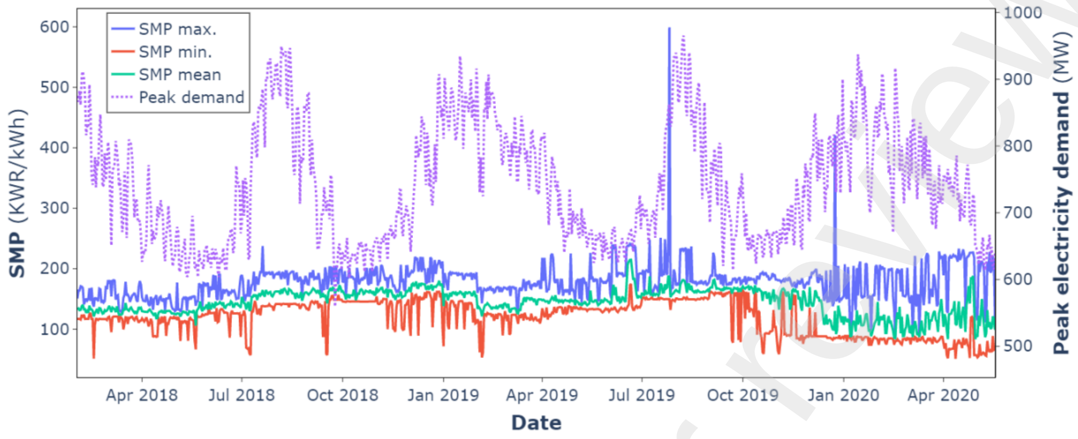


Fig. 8. Graph showing acquired peak electricity demand and SMP data.

4.2.2.2 Weather information

Weather information was sourced from Korea Meteorological Administration Weather Data Service (https://data.kma.go.kr/data/grnd/selectAsosRltmList.do?pgmNo=36). The data service provides weather information from two different stations: Automated Synoptic Observing System (ASOS) and Automatic Weather System (AWS). These two types of stations measure data automatically, but data from ASOS stations are known to be more reliable since they are crewed stations. For this reason, data from ASOS stations were used in this example. The weather information includes 34 kinds of weather properties, but not all those properties were used as training features. Only useful and usable features were selected for further steps.

4.2.2.3 Fuel cost

Fuel cost data are available on the web page of the Electric Power Statistics Information System, which is also operated by the Korea Power Exchange (http://epsis.kpx.or.kr/epsisnew/selectEkmaFucUpfChart.do?menuId=040100). Unfortunately, fuel cost information is provided monthly and not daily or weekly. Thus, for training, the same value was duplicated in the same month. For instance, the price of December 2019 was applied from the 1st to the 31st of December with the same value.

4.2.2.4 Holiday information

So far, each data set was collected in the form of a *.csv file, which is a conventional way to source data. However, holiday information was acquired through web crawling. Korean holiday information can be directly obtained via web crawling on Korean Open Data Portal (https://www.data.go.kr/data/15012690/openapi.do). This is relatively easy since this website officially provides a detailed usage guide on accessing and getting data using OpenAPI. The data is open to the public; however, an authorization key should be obtained by signing up as a member before access. Using the Python library "BeautifulSoup," html information can be parsed easily.

4.2.2.5 Number of tourists

The annual number of visitors to Jeju Island is approximately 15 million, which is nearly one-third of the total population of South Korea. Since the tourism industry occupies a substantial portion of the economy of Jeju Island, it can be expected that electricity consumption and the number of visitors are closely related. Therefore, in this example, the number of tourists is an important feature.

Tourist information on Jeju Island can also be obtained through web crawling on the web page of Jeju Island (https://www.jeju.go.kr/open/open/iopenboard.htm?category=1035). This website does not require an authorization key but also does not officially provide data access API. In this case, considerable effort is needed to obtain the data. The web page is written in HTML, and the source code of the web page will need to be imported and parsed to extract the required information. It is highly recommended to use the following three Python modules to carry out these operations: "Requests," "Beautiful Soup," and "re" modules.

4.3 Data preparation

4.3.1 Remove invalid features

Before performing the data cleaning work, the data should be verified to remove invalid features. Fig. 9 shows the visualized nullity result for weather data. Most of the features are invalid since there are too much blank data. Between February 1st, 2018, and May 18th, 2020, 746,830 hourly recorded data points exist.

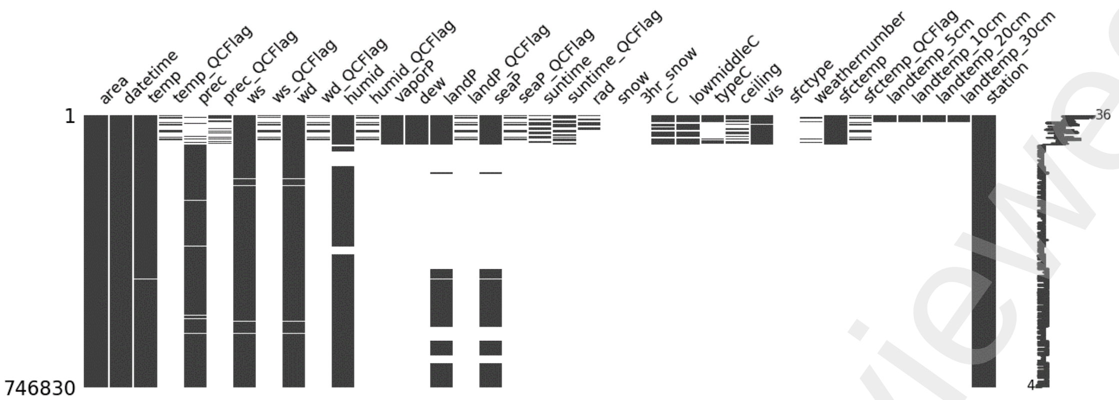


Fig. 9. Visualization of missing weather data.

Thus, except for location and temporal data (feature name area, datetime, and station), only three features (temp, ws, wd) have more than 95% data available, as shown in Table 2. The physical property itself of other features might be useful for prediction if you have complete data, but you should remove these missing features without hesitation to avoid unnecessary work.

Table 2. Available features in weather data.

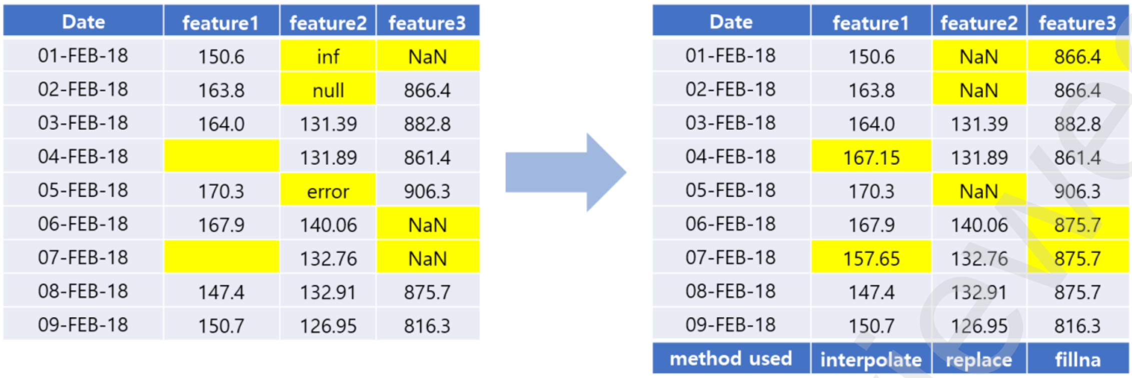
Feature name	Description	Available data ratio
area	Area code of observation location	1.000
datetime	Date and time of the measurement	1.000
temp	Air temperature (°C)	0.995
ws	Wind speed (m/s)	0.987
wd	Wind direction (16 cardinal points)	0.986
station	Kind of station (ASOS or AWS)	1.000

4.3.2 Data cleaning and time synchronization

4.3.2.1 Data cleaning

As shown in Fig. 9 and Table 2, most features contain missing or invalid data. These invalid data are called 'dirty data,' and Kim et al. [57] summarized a taxonomy for such data. In this section, the overall data cleaning theory is not considered; instead, only basic cleaning work used in this problem is explained.

The data we collected contains missing and NaN values, especially in weather data. There are three methods used to clean the data set: "interpolate," "fillna," and "replace." First, the blank data is filled using the interpolation method. However, some data fields could be filled with infinite values if interpolation is unavailable. In such cases, the "replace" method is used to replace all infinite values with NaN values. Finally, "fillna" replaces all NaN values in data fields with the next valid observation. Fig. 10 illustrates the result of data cleaning.



Date	feature1	feature2	feature3
01-FEB-18	150.6	inf	NaN
02-FEB-18	163.8	null	866.4
03-FEB-18	164.0	131.39	882.8
04-FEB-18		131.89	861.4
05-FEB-18	170.3	error	906.3
06-FEB-18	167.9	140.06	NaN
07-FEB-18		132.76	NaN
08-FEB-18	147.4	132.91	875.7
09-FEB-18	150.7	126.95	816.3

Date	feature1	feature2	feature3
01-FEB-18	150.6	NaN	866.4
02-FEB-18	163.8	NaN	866.4
03-FEB-18	164.0	131.39	882.8
04-FEB-18	167.15	131.89	861.4
05-FEB-18	170.3	NaN	906.3
06-FEB-18	167.9	140.06	875.7
07-FEB-18	157.65	132.76	875.7
08-FEB-18	147.4	132.91	875.7
09-FEB-18	150.7	126.95	816.3
method used	interpolate	replace	fillna

Fig. 10. Illustrated result of data cleaning.

Note that depending on the data characteristics, the appropriate data cleaning method should be applied.

4.3.2.2 Time synchronization

The data obtained from different sources may have different time intervals. In this case, time synchronization work should be performed to combine all features into one data set. In this example, the intervals of all data were synchronized with a day interval since we are required to predict daily demand and SMP.

The electricity demand and holiday features originally have daily intervals, so no further work is required. However, other features were recorded hourly or monthly; thus, time synchronization should be performed. If the feature has a narrower time interval than the target interval, a statistically representative value may be used. In this case, daily average values are used for both SMP and weather data. Additionally, for SMP data, daily minimum and maximum values are added as features. In contrast, fuel cost and the number of tourists data were recorded at monthly intervals. In this case, the same values should be duplicated for the whole month. Table 3 shows the feature set before and after time synchronization.

Table 3. Time interval of each feature.

Feature	Time interval of original data	Converted time interval
Electricity demand	Daily	As-is
SMP	Hourly	Daily max Daily min Daily mean
Weather information	Hourly	Daily max Daily min Daily mean Daily range Daily ratio

Fuel cost	Monthly	Same daily value in a month
Holiday information	Daily	As-is
Number of tourists	Monthly	Same daily value in a month

4.3.3 Final feature selection

This step involves finalizing the training features to use in the prediction model because variables with no relation may disturb training and produce poor prediction results. Among the weather features, highly correlated variables with target features will be selected. In this case, Pearson correlation is used. Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations [58]. The formulation for the Pearson correlation coefficient is as follows:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

where n is the sample size; x_i, y_i are the individual sample points indexed with i; and $\bar{x}$, $\bar{y}$ are the sample average values, respectively.

In Section 4.3.1, we sorted out the available three weather properties—temperature, wind speed, and wind direction—based on the nullity of data. The correlations between these three properties and the four target features—electricity demand and SMP (max, min, and mean)—were calculated. The calculated absolute correlation coefficients are shown in Table 4.

Table 4. Absolute correlation coefficients between features.

Feature	Electricity demand	SMP max.	SMP min.	SMP mean	**Average**
Air temperature	0.24148	0.14711	0.37898	0.39922	**0.30844**
Wind direction	0.25859	0.07228	0.17334	0.20446	**0.17717**
Wind speed	0.20182	0.10082	0.15494	0.17351	**0.15778**

Based on the results, the most correlated feature among the three features was temperature, with a correlation coefficient of 0.30844. Thus, to ensure a robust prediction model, the temperature feature, which has the highest correlation, was selected as the training feature among the weather properties.

There are four ASOS stations on Jeju Island, and these stations are in the north, east, west, and south of this island. For this case, the average value of these four stations' temperatures was used as the representative value.

4.3.4 Add lag and window features

The classical approach to improving model accuracy of time series data is to add a new feature from existing data. Two common features are lag features and window features. Both are known to be effective for time series data. There are three models to predict in this example. Each prediction model has its lag and window features. For instance, the temperature prediction model has temperature lag and window features.

Lag features are generated from past observations. However, selecting the optimal lag value is time-consuming. While there is some research on finding the best time-lag values of time series models [59], we used time-lag values that are known to be useful for general cases.

Window features are also created from past observations, and the window size needs to be set. In this example, the window size is selected in the same manner as lag features. Window features can also be easily created using the shift and rolling methods of the pandas library. The concept of lag and window features is illustrated in Fig. 11.

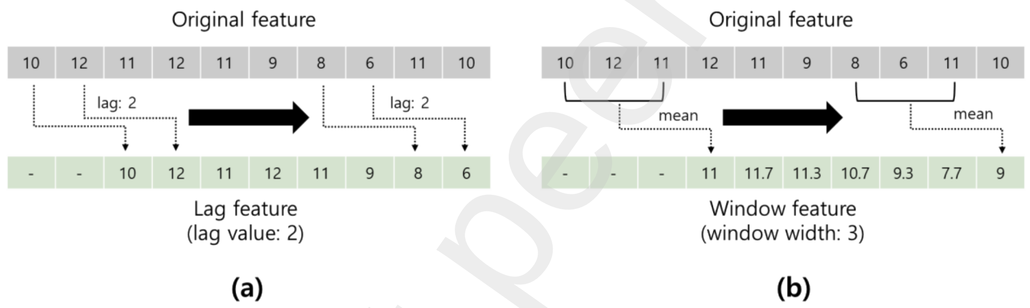


Fig. 11 Concept of (a) lag and (b) window features.

4.3.5 Data transformation

Since humans and computers have different ways of recognizing data, it is necessary to convert data so that it can be recognized by the computer in the same manner as the analyst's intention. One-hot encoding involves converting quantifying categorical features so that the prediction model can yield improved results. In this problem, the feature for which one-hot encoding is required is the holiday feature. Herein, the holiday feature can be treated as a categorical feature encoded as 1 in the case of holiday and 0 otherwise.

On the contrary, the date feature is encoded differently. The electricity demand and SMP have evident seasonality. Although the date is a cyclic feature, it is difficult for the predictive model to learn the periodic characteristics without conversion or transformation. For example, December 31st and January 1st have no significant difference in terms of electricity demand. However, if the months December (12) and January (1) are directly input to the model, they can be recognized as having very discontinuous and large differences. To enable the predictive model to learn periodic characteristics, the date and month features were encoded as a sinusoidal function. The cosine function was used because the demand was high in January, which is the beginning of the year. The variable "date" value was converted to the cosine function across the 365 days of a year (Fig. 12).

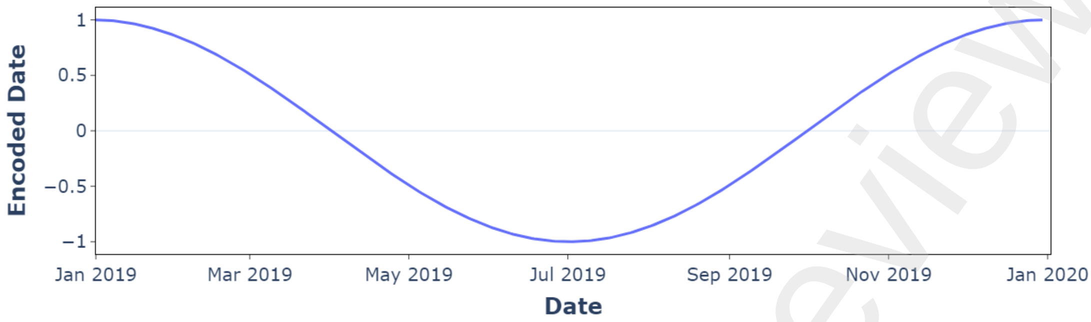


Fig. 12. Encoding date to the cosine function.

4.3.5.1 Data scaling

Every feature has different measuring units, implying that the range of change differs. If these features are used as inputs together without the scaling process, the sensitivity of the features can get distorted. This process is necessary for most machine learning algorithms, except for decision tree-based algorithms.

There are many scaling methods, such as standard scaling, min-max scaling, max-abs scaling, etc. [60,61]. In this problem, standard scaling was used, a popular and simple scaling method. The formulation of standard scaling is as follows:

$$x' = \frac{x - \bar{x}}{\sigma} \tag{2}$$

where x is the original feature vector, $\bar{x}$ is the mean of the feature vector, and σ is its standard deviation.

Finally, the data set for model training is complete. The prediction model training can now be carried out using this data set.

4.4 Model training

4.4.1 Training and target feature determination

The final target of this example is predicting electricity demand and SMP. To predict these two target features, other training features should be predicted in advance since they need to be included in the training. The temperature features were predicted using the prediction model, and other features were estimated in a reasonable manner, as below:

- The number of tourists was estimated to be 60% of the same month last year due to COVID-19.
- The fuel cost was estimated to be the same as last month since there were no ongoing geopolitical/economic events that could have led to a drastic change in price.
- The temperature features, considered the most important features in this example, were predicted using the prediction model.

In this example, there are three prediction models (temperature, demand, and SMP), so the training features of each model should be determined such that they can predict the target features properly. The training and target features of each prediction model are as below (Table 5):

Table 5. Training and target features of each prediction model.

Model	Temperature prediction model	Electricity demand prediction model	SMP prediction model
Training features	• Temperature (max, min, and mean) • Temperature range • Temperature ratio • Encoded date	• Electricity demand • Temperature (max, min, and mean) • Temperature range • Temperature ratio • Day of week • The number of tourists • Holiday • Encoded date	• SMP (max, min, and mean) • Electricity demand • Temperature (max, min, and mean) • Temperature range • Temperature ratio • Day of week • The number of tourists • Holiday • Encoded date • Fuel costs
Target features	• Temperature (max, min, and mean)	• Electricity demand	• SMP (max, min, and mean)

4.4.2 Split dataset

The final objective of a prediction model is to predict a new data set that is not included in training data. Therefore, not all the available data is used for training, which is split into validation and test datasets. The validation data confirms whether the model is correctly converging while training and the test data is data for evaluating prediction performance after training. Although there is no absolute rule for the split ratio, it generally is 70%/15%/15 or 70%/20%/10% % (train/validation/test). The data we had acquired was from February 1st, 2018, to June 8th, 2020, accounting for 859 days. We believe that the data is insufficient to learn seasonality because it includes only two years. Therefore, the data was divided into 30 days for validation, 21 days for test, and the remaining 808 days for training (Fig. 13).

Fig. 13. Data split (train/validation/test).

4.4.3 Construct model architecture

As shown in Fig. 14, there are three prediction models in this case. Although the same architecture can be used for all of them, it is desirable to construct a prediction model suitable for the characteristics of training and target features.

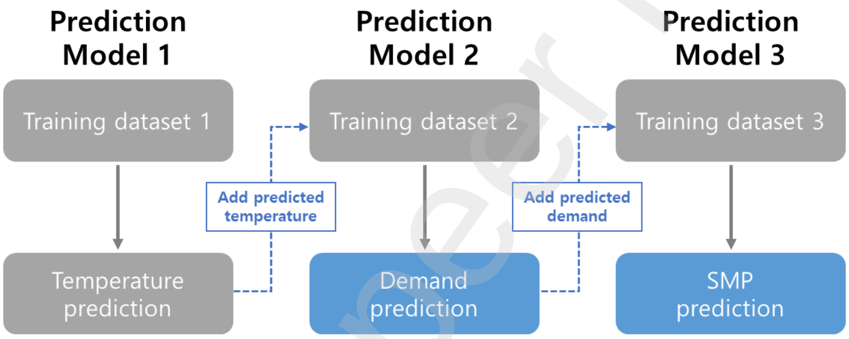


Fig. 14. Sequential prediction.

Essentially, the combined 1D-CNN and BiLSTM architecture was used for the lower layers, and some fully connected layers were added over them. The architectures were improved by modifying the FC layers and dropouts through hit and trial. In addition, hyperparameters such as learning rate, epochs, and dropout ratio can be optimized using kerastuner, a hyperparameter optimization framework. Finding optimal layers and hyperparameters is not covered here. For electricity demand prediction, one convolution layer, two BiLSTM layers, and four fully connected layers with some dropouts were used.

4.4.4 Training and validation

As the data, model architecture, and parameters have been determined; model training is now simple. However, the more important process than training is the model validation process. It is necessary to determine whether to use the model by checking whether the model is fitted properly after completion of training. There are several techniques for model validation. Most popular methods include hold-out validation, k-fold cross-validation, and leave-one-out cross-validation (LOOCV) [54–56]. In this example, the model was validated using the simplest hold-out validation method.

After training, the training and validation losses should be plotted to check for overfitting or underfitting. If training and validation losses converge in a similar trend with similar values, it can be considered that the training was successful and the model is robust.

If the training data is insufficient, the final model can be trained with the validation data after initial training with the training data. In this example, the final model was updated by additional training with 30 days of validation data. The model training and validation for this example are shown in Fig. 15.

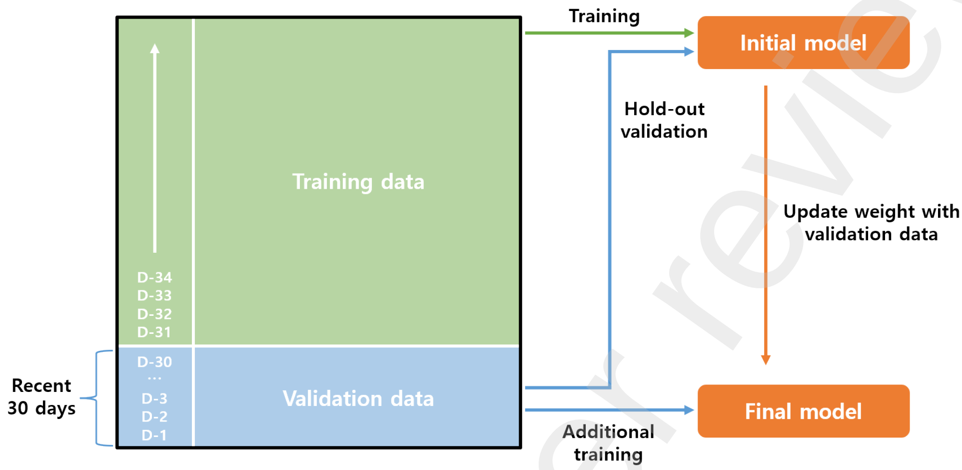


Fig. 15. Model training and validation.

4.5 *Prediction and evaluation*

4.5.1 Prediction with a trained model

Using the model trained in previous steps, temperature, peak electricity demand, and SMP were predicted. As shown in Fig. 16, data for the 8th day was predicted by inputting the last seven days. Then, the predicted 8th day was added as input for the next prediction, and the 9th day was predicted. The prediction was repeated for the next three weeks (21 days). There are three prediction models, and the prediction was performed in the order of temperature, demand, and SMP.

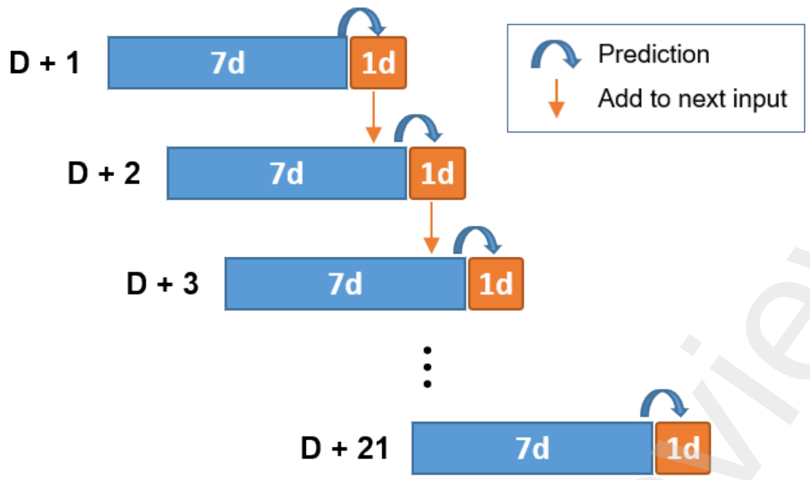


Fig. 16. Day-ahead prediction with last seven days.

Since all the data outputs of the prediction model were standardized, the output data should be rescaled after prediction to compare with the real scale result. The final prediction results are shown below in Fig. 17 and Table 6. Since demand has relatively little volatility, it can be seen that the prediction is close to the actual value. Similarly, the SMP max feature also stably predicted the trend due to little volatility during the period. On the contrary, the SMP min and SMP mean features are volatile during the corresponding period, so the prediction model could not predict the fluctuating value for each day. However, the overall trend was seemingly well-predicted.

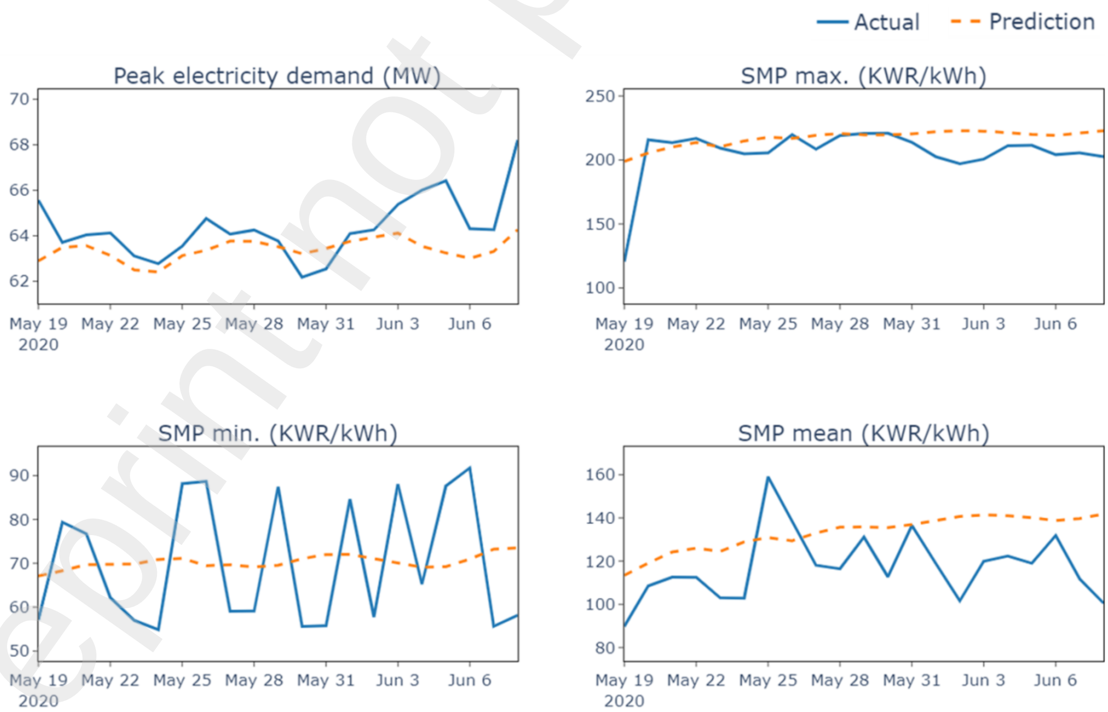


Fig. 17. Graph of prediction results versus actual values.

Table 6. Prediction results and actual values (the first and last five days).

Date	Electricity demand (MW)		SMP max. (KRW/kWh)		SMP min. (KRW/kWh)		SMP mean (KRW/kWh)	
	Actual	Prediction	Actual	Prediction	Actual	Prediction	Actual	Prediction
19-May-20	65.56	62.89	120.73	198.90	57.13	67.10	89.83	113.43
20-May-20	63.71	63.49	215.79	205.50	79.36	68.30	108.52	119.10
21-May-20	64.04	63.56	213.63	210.12	76.71	69.71	112.59	124.22
22-May-20	64.12	63.14	216.87	213.66	62.18	69.70	112.51	125.97
23-May-20	63.11	62.50	209.24	210.25	56.95	69.81	103.00	124.49
04-Jun-20	66.00	63.54	211.16	221.41	65.26	69.10	122.33	141.01
05-Jun-20	66.42	63.24	211.46	219.99	87.61	69.21	119.01	140.12
06-Jun-20	64.31	63.01	204.19	219.16	91.74	70.96	131.86	138.74
07-Jun-20	64.27	63.31	205.63	220.94	55.65	73.23	111.79	139.73
08-Jun-20	68.20	64.27	202.62	222.90	58.19	73.49	100.56	141.81

4.5.2 Evaluation of prediction results

There are several metrics for evaluating the prediction results of the time series prediction model. The simple, conventional, and commonly used metrics are mean absolute error, mean squared error, and root mean square error, which are all scale-dependent. In addition, the mean absolute percentage error is often used to express the percentage error, which is more intuitive. To get rid of the scale of the data, mean absolute scaled error and root mean squared scaled error have been proposed [62]. A brief description of these evaluation metrics is provided in the following subsections.

1) MAE, MSE, and RMSE

These three metrics are widely used as regression result evaluation metrics. These metrics do not imply a normalization process in their formula, so the error size is scale-dependent. Therefore, they should never be used when comparing data errors with different scales.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{3}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{4}$$

$$RMSE = \sqrt{MSE} \tag{5}$$

where y_i is the true value, $\hat{y}_i$ is prediction value indexed with i, and n is the number of data points, respectively.

2) MAPE

MAPE is an advanced version of MAE that removes its scale dependency. Because MAPE shows percentage error, you can compare the results of different data sets. As shown in Eq. (6), however, since the actual value is included in the denominator, when the actual value closes to 0, the value diverges to infinity.

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{6}$$

3) MASE, RMSSE, and WRMSSE

To measure the scale-independent error without the infinity problem, MASE was proposed [62]. As shown in Eq. (7), the numerator is the same as the MAE of predicted values, but the denominator is different. The denominator can be considered MAE of training data set with naïve forecasting results. MASE is scale-independent, so it can be used to compare prediction results with different scales. Moreover, if the training data are not all of the same value, the denominator cannot be 0. The MASE has stable and robust metrics compared to other methods. RMSSE is the square root version of MASE, and WRMSSE is the weighted error when there are multiple prediction features.

$$MASE = \frac{\frac{1}{h}\sum_{i=n+1}^{n+h}\left|y_i - \hat{y}_i\right|}{\frac{1}{n-1}\sum_{i=2}^{n}\left|y_i - y_{i-1}\right|} \tag{7}$$

$$RMSSE = \sqrt{\frac{\frac{1}{h}\sum_{i=n+1}^{n+h}\left(y_i - \hat{y}_i\right)^2}{\frac{1}{n-1}\sum_{i=2}^{n}\left(y_i - y_{i-1}\right)^2}} \tag{8}$$

$$WRMSSE = \frac{\sum_{i=1}^{m} w_i RMSSE_i}{\sum_{i=1}^{m} w_i} \tag{9}$$

where y_i is the true value, $\hat{y}_i$ is prediction value, w_i is the weight of each feature, n is the number of data points in training data, h is the number of data points in test data, and m is the number of prediction features. In this example, m is four.

To calculate WRMSSE, weights must be assigned to each feature according to their importance. In this example, electricity demand was considered a more important feature than SMP, so the weight was assigned as 6:4; in SMP, the average value was deemed more important than the maximum and minimum values, so it was assigned 2:1:1. In short, the assigned weight for WRMSSE is 6:2:1:1. The prediction results were evaluated using the seven metrics mentioned above. Table 7 shows the prediction error evaluated by each evaluation metric.

Table 7. Prediction error calculated by each evaluation metric.

Evaluation metric	*Electricity demand*	*SMP mean*	*SMP max.*	*SMP min.*	*Average*
MAE	1.14	19.18	13.29	13.84	11.86
MSE	2.35	468.81	439.75	210.95	280.47
RMSE	1.53	21.65	20.97	14.52	14.67
MAPE	0.017	0.171	0.077	0.202	0.117
MASE	0.413	2.889	0.845	1.673	1.455
RMSSE	0.42	2.08	0.63	0.85	0.994
WRMSSE					0.814

For the scale-dependent metrics, it is difficult to judge whether the prediction was successful based on the size of the value, while the scaled metrics can be used to determine the prediction performance based on the size of the value.

The error calculated by MASE showed the lowest error in demand and the highest in SMP mean value. As shown in Eq. (7), a MASE value greater than one means that the forecast was poorer than naïve forecasting calculated from the training data, and a value less than one means that the forecast was relatively good, that is, better than naïve forecasting. The MASE of SMP min and mean were 1.67 and 2.88, which can be considered poor prediction performance, while those for demand and SMP max can be considered relatively robust.

RMSSE and WRMSSE are also evaluated based on 1.0. Unlike MASE, the error for the SMP min value reduced from 1.67 to 0.85 due to the influence of the square root.

When two or more features are predicted, the representative value to evaluate the overall prediction performance can be averaged and expressed as a single value. WRMSSE is suitable for evaluating the overall predictive performance because it gives weight to more important features and yields a weighted average. Because the forecasting accuracy of demand was higher than that of SMP, RMSSE was 0.99, and WRMSSE was 0.81.

Overall, for the case of electricity demand with small fluctuations, prediction performance was robust regardless of the evaluation metrics; however, SMP with large fluctuations had low prediction performance. Because prediction performance varies depending on which evaluation metric is used, it is important to choose an evaluation metric according to the time series characteristics of the features to be predicted.

5. Conclusion

This tutorial demonstrated the process of predicting time series data using a deep learning model. By training features using a model combining 1D-CNN and BiLSTM, the peak electricity demand and SMP for Jeju Island in South Korea were predicted for three weeks forward. The overall process—from data preparation to result evaluation—was shown in detail. The prediction results were evaluated and comparatively analyzed using seven evaluation metrics. For electricity demand with relatively low volatility, the prediction error values of the metrics were small; however, for SMP with high volatility, the prediction accuracy was low. Nonetheless, the prediction results did not deviate from the trend of the actual values. Although the model introduced here is not the best among its peers, the methods introduced here can be an easy primer for researchers who have not majored in data analysis but need to apply AI/ML in their respective industries.

In a future study, we will study more effective models and evaluation metrics for long-term predictions that are more accurate.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit author statement

Jaedong Kim: Data curation, Visualization, Writing-original draft preparation, **SeungHwan Oh**: Data curation, Investigation, Resources, **Hee-soo Kim**: Writing-review, Funding acquisition, **Woosung Choi**: Writing-review and Editing, Methodology, Supervision, Conceptualization

Acknowledgments

This research was supported by Korea Electric Power Corporation [Grant number: R17GA08]. This study was conducted as part of the AI Friends activity, which is focused on the Daedeok Research Complex to expand the base of AI. We also thank the editors and anonymous reviews for their valuable support and comments. Certainly, all remaining errors are our own.

Data availability

The entire Python source code is available at the GitHub repository (https://github.com/jdkim6413/Tutorial-for-Time-Series-Prediction.git).

References

[1] Liu H, Shi J, Erdem E. Prediction of wind speed time series using modified Taylor Kriging method. Energy 2010;35. https://doi.org/10.1016/j.energy.2010.09.001.

[2] Ahmed NK, Atiya AF, Gayar N el, El-Shishiny H. An Empirical Comparison of Machine Learning Models for Time Series Forecasting. Econometric Reviews 2010;29:594–621. https://doi.org/10.1080/07474938.2010.481556.

[3] Amjady N, Keynia F. A new prediction strategy for price spike forecasting of day-ahead electricity markets. Applied Soft Computing 2011;11:4246–56. https://doi.org/10.1016/j.asoc.2011.03.024.

[4] Prema V, Uma Rao K. Development of statistical time series models for solar power prediction. Renewable Energy 2015;83. https://doi.org/10.1016/j.renene.2015.03.038.

[5] Hu J, Wang J. Short-term wind speed prediction using empirical wavelet transform and Gaussian process regression. Energy 2015;93. https://doi.org/10.1016/j.energy.2015.10.041.

[6] Maté A, Peral J, Ferrández A, Gil D, Trujillo J. A hybrid integrated architecture for energy consumption prediction. Future Generation Computer Systems 2016;63. https://doi.org/10.1016/j.future.2016.03.020.

[7] Taslimi Renani E, Elias MFM, Rahim NA. Using data-driven approach for wind power prediction: A comparative study. Energy Conversion and Management 2016;118. https://doi.org/10.1016/j.enconman.2016.03.078.

[8] Voyant C, Motte F, Fouilloy A, Notton G, Paoli C, Nivet M-L. Forecasting method for global radiation time series without training phase: Comparison with other well-known prediction methodologies. Energy 2017;120:199–208. https://doi.org/10.1016/j.energy.2016.12.118.

[9] Lin L, Wang F, Xie X, Zhong S. Random forests-based extreme learning machine ensemble for multi-regime time series prediction. Expert Systems with Applications 2017;83. https://doi.org/10.1016/j.eswa.2017.04.013.

[10] Liu C, Sun B, Zhang C, Li F. A hybrid prediction model for residential electricity consumption using holt-winters and extreme learning machine. Applied Energy 2020;275. https://doi.org/10.1016/j.apenergy.2020.115383.

[11] Lu H, Cheng F, Ma X, Hu G. Short-term prediction of building energy consumption employing an improved extreme gradient boosting model: A case study of an intake tower. Energy 2020;203. https://doi.org/10.1016/j.energy.2020.117756.

[12] Frank RJ, Davey N, Hunt SP. Time series prediction and neural networks. Journal of Intelligent and Robotic Systems: Theory and Applications 2001;31. https://doi.org/10.1023/A:1012074215150.

[13] Kim B, Velas JP, Lee J, Park J, Shin J, Lee KY. Short-term system marginal price forecasting using system-type neural network architecture. 2006 IEEE PES Power Systems Conference and Exposition, PSCE 2006 - Proceedings, 2006. https://doi.org/10.1109/PSCE.2006.296178.

[14] Guo Z, Zhou K, Zhang X, Yang S. A deep learning model for short-term power load and probability density forecasting. Energy 2018;160. https://doi.org/10.1016/j.energy.2018.07.090.

[15] Tong C, Li J, Lang C, Kong F, Niu J, Rodrigues JJPC. An efficient deep model for day-ahead electricity load forecasting with stacked denoising auto-encoders. Journal of Parallel and Distributed Computing 2018;117. https://doi.org/10.1016/j.jpdc.2017.06.007.

[16] Chen J, Zeng GQ, Zhou W, Du W, Lu K di. Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. Energy Conversion and Management 2018;165. https://doi.org/10.1016/j.enconman.2018.03.098.

[17] Yin H, Ou Z, Huang S, Meng A. A cascaded deep learning wind power prediction approach based on a two-layer of mode decomposition. Energy 2019;189. https://doi.org/10.1016/j.energy.2019.116316.

[18] Hao X, Guo T, Huang G, Shi X, Zhao Y, Yang Y. Energy consumption prediction in cement calcination process: A method of deep belief network with sliding window. Energy 2020;207. https://doi.org/10.1016/j.energy.2020.118256.

[19] Zhang L, Wang J, Wang B. Energy market prediction with novel long short-term memory network: Case study of energy futures index volatility. Energy 2020;211. https://doi.org/10.1016/j.energy.2020.118634.

[20] Xue G, Qi C, Li H, Kong X, Song J. Heating load prediction based on attention long short term memory: A case study of Xingtai. Energy 2020;203. https://doi.org/10.1016/j.energy.2020.117846.

[21] Wang JQ, Du Y, Wang J. LSTM based long-term energy consumption prediction with periodicity. Energy 2020;197. https://doi.org/10.1016/j.energy.2020.117197.

[22] Zheng J, Zhang H, Dai Y, Wang B, Zheng T, Liao Q, et al. Time series prediction for output of multi-region solar power plants. Applied Energy 2020;257. https://doi.org/10.1016/j.apenergy.2019.114001.

[23] Lu H, Ma X, Azimi M. US natural gas consumption prediction using an improved kernel-based nonlinear extension of the Arps decline model. Energy 2020;194. https://doi.org/10.1016/j.energy.2020.116905.

[24] Le T, Vo MT, Vo B, Hwang E, Rho S, Baik SW. Improving electric energy consumption prediction using CNN and Bi-LSTM. Applied Sciences (Switzerland) 2019;9. https://doi.org/10.3390/app9204237.

[25] Ustundag BB, Kulaglic A. High-Performance Time Series Prediction with Predictive Error Compensated Wavelet Neural Networks. IEEE Access 2020;8. https://doi.org/10.1109/ACCESS.2020.3038724.

[26] Laib O, Khadir MT, Mihaylova L. Toward efficient energy systems based on natural gas consumption prediction with LSTM Recurrent Neural Networks. Energy 2019;177. https://doi.org/10.1016/j.energy.2019.04.075.

[27] E J, Ye J, He L, Jin H. Energy price prediction based on independent component analysis and gated recurrent unit neural network. Energy 2019;189. https://doi.org/10.1016/j.energy.2019.116278.

[28] Chang Z, Zhang Y, Chen W. Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. Energy 2019;187. https://doi.org/10.1016/j.energy.2019.07.134.

[29] Cai X, Zhang N, Venayagamoorthy GK, Wunsch DC. Time series prediction with recurrent neural networks trained by a hybrid PSO–EA algorithm. Neurocomputing 2007;70:2342–53. https://doi.org/10.1016/j.neucom.2005.12.138.

[30] Emmert-Streib F, Yang Z, Feng H, Tripathi S, Dehmer M. An Introductory Review of Deep Learning for Prediction Models With Big Data. Frontiers in Artificial Intelligence 2020;3. https://doi.org/10.3389/frai.2020.00004.

[31] Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. Neurocomputing 2017;234. https://doi.org/10.1016/j.neucom.2016.12.038.

[32] Salehinejad H, Sankar S, Barfett J, Colak E, Valaee S. Recent Advances in Recurrent Neural Networks 2017. https://doi.org/10.48550/arxiv.1801.01078.

[33] Lu W, Li J, Wang J, Qin L. A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications 2020. https://doi.org/10.1007/s00521-020-05532-z.

[34] Kim TY, Cho SB. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 2019;182. https://doi.org/10.1016/j.energy.2019.05.230.

[35] Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in Neural Information Processing Systems, vol. 2015- Janua, 2015.

[36] Sainath TN, Vinyals O, Senior A, Sak H. Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 2015- Augus, 2015. https://doi.org/10.1109/ICASSP.2015.7178838.

[37] Wang R, Liu F, Hou F, Jiang W, Hou Q, Yu L. A Non-Contact Fault Diagnosis Method for Rolling Bearings Based on Acoustic Imaging and Convolutional Neural Networks. IEEE Access 2020;8. https://doi.org/10.1109/ACCESS.2020.3010272.

[38] Liu Q, Huang C. A Fault Diagnosis Method Based on Transfer Convolutional Neural Networks. IEEE Access 2019;7. https://doi.org/10.1109/ACCESS.2019.2956052.

[39] Neupane D, Kim Y, Seok J. Bearing Fault Detection Using Scalogram and Switchable Normalization-Based CNN (SN-CNN). IEEE Access 2021;9. https://doi.org/10.1109/ACCESS.2021.3089698.

[40] Ding X, He Q. Energy-Fluctuated Multiscale Feature Learning with Deep ConvNet for Intelligent Spindle Bearing Fault Diagnosis. IEEE Transactions on Instrumentation and Measurement 2017;66. https://doi.org/10.1109/TIM.2017.2674738.

[41] Lu C, Wang Z, Zhou B. Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification. Advanced Engineering Informatics 2017;32. https://doi.org/10.1016/j.aei.2017.02.005.

[42] Ince T, Kiranyaz S, Eren L, Askar M, Gabbouj M. Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks. IEEE Transactions on Industrial Electronics 2016;63:7067–75. https://doi.org/10.1109/TIE.2016.2582729.

[43] Eren L, Ince T, Kiranyaz S. A Generic Intelligent Bearing Fault Diagnosis System Using Compact Adaptive 1D CNN Classifier. Journal of Signal Processing Systems 2019;91. https://doi.org/10.1007/s11265-018-1378-3.

[44] Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M, Inman DJ. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. Journal of Sound and Vibration 2017;388. https://doi.org/10.1016/j.jsv.2016.10.043.

[45] Kiranyaz S, Ince T, Gabbouj M. Personalized Monitoring and Advance Warning System for Cardiac Arrhythmias. Scientific Reports 2017;7:9270. https://doi.org/10.1038/s41598-017-09544-z.

[46] Eren L. Bearing Fault Detection by One-Dimensional Convolutional Neural Networks. Mathematical Problems in Engineering 2017;2017:1–9. https://doi.org/10.1155/2017/8617315.

[47] Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 1997;45. https://doi.org/10.1109/78.650093.

[48] Das R, Bo R, Ur Rehman W, Chen H, Wunsch D. Cross-market price difference forecast using deep learning for electricity markets. IEEE PES Innovative Smart Grid Technologies Conference Europe, vol. 2020- Octob, 2020. https://doi.org/10.1109/ISGT-Europe47291.2020.9248867.

[49] Siami-Namini S, Tavakoli N, Namin AS. The Performance of LSTM and BiLSTM in Forecasting Time Series. Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019, 2019. https://doi.org/10.1109/BigData47090.2019.9005997.

[50] Huang CG, Huang HZ, Li YF. A Bidirectional LSTM Prognostics Method Under Multiple Operational Conditions. IEEE Transactions on Industrial Electronics 2019;66. https://doi.org/10.1109/TIE.2019.2891463.

[51] Rhanoui M, Mikram M, Yousfi S, Barzali S. A CNN-BiLSTM Model for Document-Level Sentiment Analysis. Machine Learning and Knowledge Extraction 2019;1. https://doi.org/10.3390/make1030048.

[52] Wang M, Cheng J, Zhai H. Life Prediction for Machinery Components Based on CNN-BiLSTM Network and Attention Model. 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), IEEE; 2020, p. 851–5. https://doi.org/10.1109/ITOEC49072.2020.9141720.

[53] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A nucleus for a Web of open data. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4825 LNCS, 2007. https://doi.org/10.1007/978-3-540-76298-0_52.

[54] Stone M. Cross-Validatory Choice and Assessment of Statistical Predictions. Journal of the Royal Statistical Society: Series B (Methodological) 1974;36:111–33. https://doi.org/10.1111/j.2517-6161.1974.tb00994.x.

[55] Allen DM. The Relationship Between Variable Selection and Data Agumentation and a Method for Prediction. Technometrics 1974;16:125–7. https://doi.org/10.1080/00401706.1974.10489157.

[56] Geisser S. The Predictive Sample Reuse Method with Applications. J Am Stat Assoc 1975;70:320–8. https://doi.org/10.1080/01621459.1975.10479865.

[57] Kim W, Choi BJ, Hong EK, Kim SK, Lee D. A Taxonomy of Dirty Data. Data Mining and Knowledge Discovery 2003 7:1 2003;7:81–99. https://doi.org/10.1023/A:1021564703268.

[58] PEARSON K. NOTES ON THE HISTORY OF CORRELATION. Biometrika 1920;13:25–45. https://doi.org/10.1093/BIOMET/13.1.25.

[59] Surakhi O, Zaidan MA, Fung PL, Hossein Motlagh N, Serhan S, AlKhanafseh M, et al. Time-Lag Selection for Time-Series Forecasting Using Neural Network and Heuristic Algorithm. Electronics (Basel) 2021;10:2518. https://doi.org/10.3390/electronics10202518.

[60] Ahsan MM, Mahmud MAP, Saha PK, Gupta KD, Siddique Z. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. Technologies (Basel) 2021;9. https://doi.org/10.3390/technologies9030052.

[61] Raju VNG, Lakshmi KP, Jain VM, Kalidindi A, Padma V. Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020 2020:729–35. https://doi.org/10.1109/ICSSIT48917.2020.9214160.

[62] Hyndman RJ, Koehler AB. Another look at measures of forecast accuracy. International Journal of Forecasting 2006;22:679–88. https://doi.org/10.1016/J.IJFORECAST.2006.03.001.