IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 4, 2021, accepted July 8, 2021, date of publication July 26, 2021, date of current version August 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3099981

An Extensible Framework for Short-Term Holiday Load Forecasting Combining Dynamic Time Warping and LSTM Network

JEFFREY GUNAWAN AND CHIN-YA HUANG, (Member, IEEE)
Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 106, Taiwan
Corresponding author: Chin-Ya Huang (chinya@gapps.ntust.edu.tw)

This work was supported in part by the Taiwan Building Technology Center, Taiwan Tech. from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education, Taiwan, and in part by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 108-2221-E-011-058-MY3.

ABSTRACT Due to the extreme change of human behavior, the load consumption in public holidays fluctuates more significantly compared to general weekdays resulting in the difficulty of hourly holiday load forecasting. The holiday load forecasting is even challenging because the forecast is practically predicted on the nearest workday which might be more than one days prior to the public holiday. In this paper, we propose a Joint Dynamic time warping and LSTM (JDL) framework, to predict the hourly holiday load consumption on the nearest workday which is at least one day before the incoming holiday. The proposed JDL is a hybrid short-term holiday forecasting framework which combines dynamic time warping (DTW) and long-short term memory (LSTM) network. The DTW predicts the load consumption of the nearest workday and any preceding compensatory holiday(s), if any, based on the similar holiday occurrence pattern. LSTM predicts the highly unpredictable load consumption of the target holiday by univariate and multivariate models. Current results show the proposed JDL outperforms others.

INDEX TERMS Holiday load forecasting, short-term load forecasting (STLF), long-short term memory (LSTM), dynamic time warping (DTW).

I. INTRODUCTION

Accurate electrical load forecasting in a power generation and distribution system can better assist the balance between the electricity production and demand. With accurate load forecasting, excess power reserve and redundant power generation cycle can be avoided so that the overall system efficiency will be enhanced [1]. Approximately 1% decline in prediction error is worth up to $1.6 million on a year-to-year basis [2]. However, the forecasting accuracy on holiday decreases significantly due to the drastic change in human behavior which in turn affects the electricity demand comparing to the weekdays/weekends.

A public holiday may be preceded by weekends and form a consecutive holiday period. We denote the weekends preceding this public holiday as compensatory holidays. Due to practical considerations, it is infeasible to only consider a day-ahead prediction in the presence of compensatory holidays. In addition, most previous works do not address the need to perform multiple days ahead prediction for holiday load forecasting. The behavior of load consumption on a public holiday preceded by compensatory holidays is even harder to predict due to complex relationship between day-type and the occurrence of the holiday itself. In this sense, the time horizon for holiday load forecast spans from one up to several days. Electric load forecasting is divided into several different groups based on their prediction horizon. Long-term load forecasting (LTLF) is designed to predict five years until a decade in the future to plan electricity capacity or grid while short-term load forecast (STLF) aims to predict a short period of time, such as twenty-four-hour ahead until a week [3]. In accordance with the chosen prediction horizon, we choose to perform STLF to enhance the power plant scheduling and demand response for hourly holiday load forecasting.

To better predict the short-term holiday load consumption, the challenges are listed as follows:

- Several holidays fall on different days of the week on a year-to-year basis. As a result, the resulting load usage is not the same between each year.

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

- The available data record of a given public holiday is insufficient, resulting in a sparse data problem for the short-term holiday load forecasting. Because a specific holiday occurs only once in a year, only one load consumption record for that holiday is available for each historical year.

Previous works propose a number of strategies to perform holiday load forecasting. Linear regression-based approaches are considered in [4], [5]. The advantage of statistics-based method, such as linear regression, is that it only needs a small amount of data. However, linear regression can only capture the linear relationship between input features. Alternatively, similarity-based methods are proposed as they take advantage of the fact that days with similar features, such as weather information, have similar load consumption patterns as well [6], [7]. However, since similarity-based methods heavily rely on the used feature, they are not a strong prediction model to predict the volatile load consumption of a holiday.

Then, machine-learning (ML) based approaches, such as support vector machine (SVM) [8] and artificial neural network (ANN) [9], are heavily favored since they can be trained to learn non-linear functions. Specifically, as ANN are the most commonly used model, it is continuously developed by adding more layers to improve its prediction performance. Nevertheless, increasing the number of layers causes over-fitting issues, limiting the performance of ANN. In response, deep learning (DL) methods are developed to improve the performance of ANN by modifying the inner structure of the network itself, such as deep residual networks [10]. In particular, long-short term memory network (LSTM) is a popular choice due to its capability to learn the non-linear relationship between input features of different past time periods [11]–[15]. However, DL methods require a sufficient amount of training data in order to perform accurately, but data of a specific holiday is insufficient due to the scarce data problem. To solve this issue, several ideas, such as transfer learning [16] and data augmentation [17], are proposed, where each one of them has their own limitations. However, among those works, only a few mention the need to perform multiple days ahead prediction [5], [11], [15].

In this paper, we propose a Joint Dynamic time warping and LSTM (JDL) framework to predict a consecutive holiday period. We assume the load forecasting task is executed at the start of the working day, and thus, given a target public holiday, JDL predicts not only the loads of the holiday but also compensatory holiday(s), if any, and the nearest workday.

To summarize, our contribution are as follows:

1) We propose a forecasting framework, JDL, based on a combination of similarity-based method using DTW and LSTM to predict holiday load consumption regardless of its occurrence.
2) We conduct a case study to predict nine holidays of a specific year from the dataset containing 9-year-load-consumption to validate and analyze the achieved prediction results among our proposed JDL and other approaches from the literature. The results show the prediction accuracy of JDL does not decline greatly between holidays with different length of prediction horizon.

The rest of the paper is organized as follows. Section III and IV describe the system model and the proposed JDL. Section V provides the prediction results, and this work is concluded in Section VI.

II. RELATED WORKS

To solve the scarce data issue, transfer learning is proposed [16]. By carefully selecting relevant historical load consumption information from neighbouring cities, the forecasting performance for a target city is improved. However, this method is viable only when additional data for multiple cities are available. Alternatively, to solve the scarce data issue, a data augmentation can be performed to increase the amount of available data by creating new, artificial data based on the available data. For example, Forestier *et al.* [17] design a data augmentation method for time-series classification by using weighted average from a set of time-series data to be used as a new, synthetic time-series data. However, said method needs at least two time-series data of the same class since an averaging process is performed whereas some power companies might not have more than one historical year for a certain holiday. To select past days to train a prediction model, a concept of similarity is commonly adopted, either by using temperature information or day-type information as a feature. Wi *et al.* [7] select weekdays data prior to the past holiday for each year in the historical data and the present year based on several weather features, such as temperature and humidity, to train their prediction model. However, their proposed data selection procedure relies heavily on weather features, where further optimizations might be required to adopt the procedure since different regions of the world have different climates.

Ding *et al.* [6] combine the load consumption of the most similar day to the target holiday in terms of temperature information and annual load consumption growth between the present year and one previous year. Cai *et al.* [8] propose a support vector machine model optimized by genetic algorithm which takes load consumption of past days whose day-type are the same as the target holiday as inputs. However, both [6] and [8] do not consider the effect of which day of the week does the holiday falls has on the resulting load consumption. Fernandes *et al.* [4] propose a weighted addition method by combining the load consumption of day with the same day-type in the previous one week and the nearest Sunday prior to the target holiday. However, in their results, some holidays have much higher prediction error compared to the rest of the holidays, which indicates that statistics-based method, such as linear regression, is not the best solution to predict the load consumption of a public holiday. Based on regression analysis, Ziel *et al.* [5] propose two time-series approaches, which are univariate and multivariate models, to perform holiday load forecasting. Although the idea of univariate and multivariate model will be used in this study, the regression model itself is not used in our framework since, as we mention, it is not the best solution for holiday load prediction.

Artificial neural network (ANN) [9] is a classic prediction model in the field of STLF. However, one disadvantage of ANN is that the size of the networks will grow along with the increase in the number of input features, resulting in overfitting issues. Thus, deep learning methods are proposed to improve the performance of ANN by modifying the building block inside the network itself. For instance, to avoid stacking multiple hidden layers between the input and the output layer, which will increase the difficulty of the training process due to increase in the number of hidden layers, a deep residual network is proposed [10]. Specifically, LSTM has been widely used in load forecasting problems, especially when a traditional machine learning model fails to deliver, such as residential load forecasting and industrial power demand forecasting. Two LSTM layers are used by [11] and [13] to predict customer-level and region-level load consumption, respectively. However, both [11] and [13] only use a binary holiday indicator to anticipate an incoming holiday instead of further analyzing whether the incoming holiday is preceded by compensatory holidays or not. In order to further improve forecasting performance, an ensemble model is proposed by combining multiple prediction models. Tan *et al.* [12] propose an ensemble model by constructing multiple LSTM models, where each one is trained with a random subset of data and feature space and their outputs are combined using a weighted additive combination. However, said ensemble method is only possible when data with a large feature space is available. Tang *et al.* [15] propose a neural network layer to combine the prediction results from multiple LSTM models and auto-regressive integrated moving average (ARIMA) models which demonstrates an improvement over individual LSTM and ARIMA model. To avoid using another holiday occurring several weeks before the target holiday, the proposed methods in [15] require further modification.

Among existing approaches, only a few previous works address the need to perform multiple days ahead forecasting, which would be necessary for holiday load forecasting, as mentioned in the previous section. Although LSTM is popularly used in electric load forecasting, it is not an optimal solution for our considered multiple steps ahead forecasting because the prediction error for the forecast one day ahead load consumption will be propagated to the predicted load consumption of the next day [14].

Moreover, methods combing similarity-based approach, DTW, and deep learning-based approach, neural network are proposed to predict the time series information [18], [19]. Authors in [18] propose a combination of DTW-based fuzzy clustering algorithm and LSTM to predict traffic flow using data from multiple sensors. DTW is used to measure similarity between multiple sensor readings and form several clusters fed to the LSTM prediction model. However, multiple data sources are required to perform the clustering. In addition, the motivation in [18] is that sensors which are geographically close to each other are related by a traffic flow, whereas for holiday load forecasting, an initial data exploratory procedure to determine the relationship between load consumption curves of different cities is necessary. Thus, their proposed method cannot be directly adopted for short-term holiday load forecasting. In [19], DTW and gated recurrent neural network is combined to perform day-ahead peak load consumption forecasting of a target day. DTW selects the most similar peak load consumption of the same day of the week in the historical data. Since they focus on daily peak load forecasting, they consider daily peak load curves of one week due to the auto-regressive effect of a week whereas this paper focuses on the hourly holiday load forecasting which might need to predict the load consumption several days prior to the target holiday.

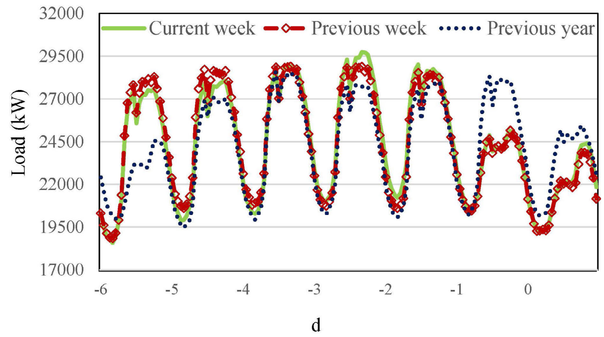
III. SYSTEM DESCRIPTION

We denote x as an indicator of a public holiday, in present year. Following [5], we consider hourly load information for one observed day. Thus, the length of observed load information for one day is twenty-four. Further, considering the target holiday with index x, we denote $|d|$ days before the target holiday x as d, $d < 0$ and $|N|$ years prior to present year as N, $N < 0$, assuming at least $|N|$ years historical data is available to assist the target holiday load forecasting. Further, h is denoted as the time of the day indicator, and $h = \{i\}$, where i indicates the i-th hour in one day, $i \in [1, 24]$. Then, we denote the twenty-four-hour load information for the day which is $|d|$ days before the target holiday x in the previous $|N|$-th year as (1).

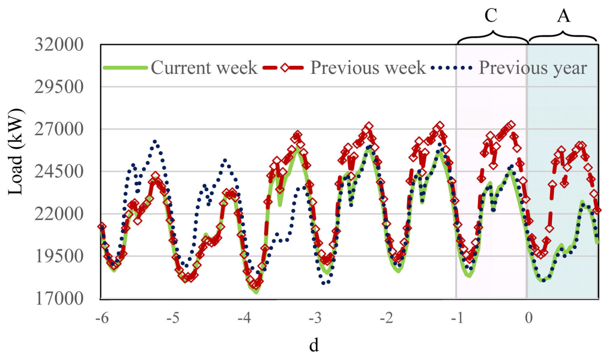
$$L^h_{d,N} = \{L^1_{d,N}, \ldots, L^{24}_{d,N}\}. \tag{1}$$

Hence, we denote the load consumption of the holiday x as $L^h_{0,0}$ and the prediction result of x as $\bar{L}^h$.

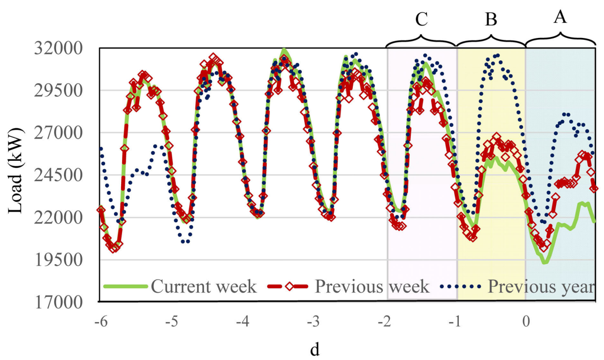
We assume that load forecasting task in a certain power system company is executed in the nearest workday prior to x. For the nearest workday of x, w, is denoted as the number of days before x, and thus $w \leq -1$. Specifically, the holiday might occur right after the nearest workday so that $w = -1$. Compensatory holiday(s) may exist between the nearest workday and the target holiday, and thus $w < -1$. We further assume the load forecasting execution time is at the δ-th hour of the nearest workday of x, where $\delta \in [2, 23]$. Thus, for the nearest workday, only $L^{i'}_{w,0}$, $i' \in [1, \delta - 1]$ is available when the incoming holiday load forecasting is performed. Hence, in the presence of compensatory holiday(s), not only part of the load consumption information on the nearest workday but also that on compensatory holiday(s) are unavailable for the holiday load forecasting. To better predict the holiday load usage, it is important to investigate load consumption patterns, in the viewpoint of the target holiday, workdays and compensatory holiday(s), if any.



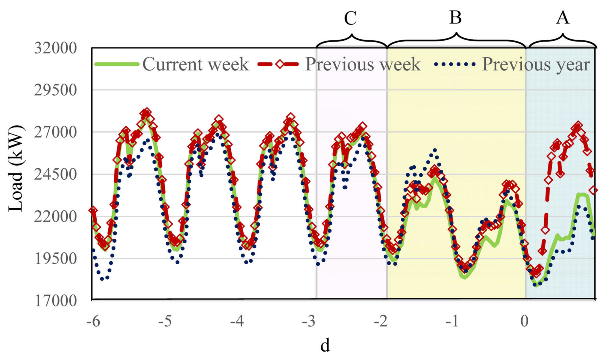
(a) No holiday.



(b) Holiday with no compensatory holidays (w=-1).



(c) Holiday with one compensatory holiday (w=-2).



(d) Holiday with two compensatory holidays (w=-3).

FIGURE 1. Load curves of two consecutive weeks under different condition.

Fig. 1 presents the load consumption curve for three different time zones, denoted as current week, previous week and previous year under different holiday occurrence patterns. In terms of the load consumption of current week, the curves of previous week and previous year indicate the load consumption a week and a year ago, respectively. Further, the x-axis indicates the consecutive seven days, ranging from -6 to 0 (i.e., $d = -6$ to 0). The y-axis illustrates the corresponding load consumption among twenty-four hours on each day. Obviously, it can be seen that the load pattern on each day is related, and within a day, the scale of the hourly load consumption depends on its previous hours and days. Further, as shown in Fig. 1a, in the absence of a holiday, the load consumption patterns between two consecutive weeks is almost identical. More specifically, for a target day, such as $d = 0$ of current week, the load consumption pattern is more close to that of the same day of previous week comparing to others. Although the load pattern of previous year is still similar to current year, certain variations occur which might be caused by human behavior. For example, when $d = -6$ or -1, the scales of load usage between current week and previous year are different. Further, the load consumption scales at $d = 0$ between current week and previous year are similar, but the patterns are different.

On the other hand, Fig. 1b - 1d present the load consumption curves when a holiday occurs on Day 0 of current week. It can be found in 'A' block, the load consumption patterns on Day 0 of current and previous week are significantly different, especially in Fig. 1d. It might be because the holiday falling on Day 0 of current week changes the human behavior which in turn causes the variation of load usage pattern. Furthermore, considering the load consumption pattern and scale of Day 0 in current week, no similarity can be found among Fig. 1b - 1d, even though a holiday occurs on that day. It also can be found in Fig. 1b and 1d, the load consumption patterns and scales of Day 0 between current week and previous year are similar, which implies historical data may help holiday load forecasting under certain conditions.

Additionally, one and two days compensatory holidays happen in Fig. 1c and 1d, respectively as shown the current week curves in 'B' block. It can be seen that the load consumption pattern between the current and previous week in 'B' block is smaller than 'A', but larger than other days in which no holidays or compensatory holidays present in current and previous week. Further, in Fig. 1c, in 'B' block, the curve of previous year is less similar to that of current week than the curves between current week and previous week. It might be because in previous year, the holiday occurs in the presence of no compensatory holiday, and the load curve of previous year in 'B' block is closer to that of working days such as the patterns of $d = -2$.

Moreover, in practice, the load forecasting is executed at the first hour of the nearest workday prior to the public holiday. In Fig. 1b - 1d, the current week curve falling the 'C' block indicates the load usage of the nearest workday. Obviously, the load consumption pattern on the nearest workday is similar to other working days such as its previous week, and even its previous year.

To sum up, the holiday load consumption might not only depend on the consecutive hours or days prior to the holiday, but also the same holiday in the past years. Further, the holiday load forecasting is executed on the nearest workday, which indicates that no complete load usage information is available on the nearest workday and compensatory holiday(s), if any, to assist the target holiday load forecasting. Hence, a framework is required to better predict the complex load consumption pattern, especially when the public holiday is preceded by multiple compensatory holidays.

IV. JOINT DYNAMIC TIME WARPING AND LSTM (JDL) FRAMEWORK

To effectively predict holiday load consumption, we propose a hybrid short-term holiday load forecasting framework, namely Joint Dynamic time warping and LSTM (JDL). Considering the load patterns in Fig. 1b - 1d, the proposed JDL combines two components, which are Dynamic time warping for the Nearest workday and Compensatory holiday load prediction (DynaNC) and LSTM for Holiday load prediction (LSTMH). Firstly, DynaNC utilizes DTW to forecast load consumption for the nearest workday and compensatory holiday(s), because the load consumption behavior on the nearest workday and the compensatory holiday(s), if any, between two consecutive week does not change drastically. Further, the load consumption pattern is time related, and thus the load consumption of the nearest workday and compensatory holidays are worth to be estimated aiming to assist the load forecasting for the target holiday. Secondly, LSTMH adopts LSTM to forecast the target holiday load consumption, which is denoted as $\bar{L}^h$, because the load consumption pattern on the target holiday behave more differently from its previous week and other holidays.

A. DYNAMIC TIME WARPING FOR THE NEAREST WORKDAY AND COMPENSATORY HOLIDAY LOAD PREDICTION (DYNANC)

The proposed DynaNC predicts the load consumption of the nearest workday prior to a public holiday, x, and its compensatory holiday(s), if any. Specifically, based on the observation of Fig. 1, we apply similarity-based approach to perform load forecasting for the nearest workday and compensatory holiday(s), if any, aiming to provide more useful information to assist load forecasting of the target holiday.

Algorithm 1 presents the procedure of DynaNC to predict $L^h_{d,0}$, $w \leq d \leq -1$, and DynaNC firstly finds w. Based on $L^{h'}_{w,0}$, $h' \in \{i'\}$, $i' \in [1, \delta-1]$, DTW is used to select the most similar load pattern from the past two consecutive weeks, $L^h_{j,0}$, $j = \{w-14, w-7\}$. Specifically, we denote a distance function with DTW to be $DTW(\mathbf{a},\mathbf{b})$ which is shown as (3).

$$DTW(\mathbf{a},\mathbf{b}) = \mathbf{W}[24, 24], \quad \mathbf{a} \in \mathbb{R}^{24} \text{ and } \mathbf{b} \in \mathbb{R}^{24} \tag{3}$$

Algorithm 1 DynaNC

Find w;
$\bar{w} = \underset{j \in \{w-14, w-7\}}{\arg\min}\ DTW(L^{h'}_{w,0}, L^h_{j,0})$, $h' = \{i'\}, i' \in [1, \delta-1]$;
Obtain $L^i_{w,0}, \forall i \in [\delta, 24]$ by (2);
for $c = w+1; c < 0; c++$ **do**
  if $\bmod(D^1_{c,0}, 6) == 0$ **then**
    $\bar{c} = \underset{j \in \{c-14, c-7\}}{\arg\min}\ DTW(T^h_{c,0}, T^h_{j,0})$;
    Obtain $L^i_{c,0}, \forall i \in [1, 24]$ by (5);
  else
    Obtain $L^i_{c,0}, \forall i \in [1, 24]$, by (6);
  end if
end for

Each of the two vectors, **a** and **b**, contains twenty-four-hour load information as the input of DTW. We denote was the output of the DTW indicating the difference between each observed load consumption inside **a** and **b** and the minimum value from a set of differences between previous observations inside **a** and **b**. Further, the first row and column of **W** are initialized as infinity. Further, **W**[0,0] is set as zero. Following Bellman's principle [20], **W** can be obtained by (4), as shown at the bottom of the page.

Then, according to the load consumption pattern of x, the most similar week of the corresponding nearest workday, denoted as $\bar{w}$, is selected for the load forecasting of the nearest workday. Following (2), as shown at the bottom of the page, the load forecasting of the rest hours in the nearest workday, $L^i_{w,0}, \forall i \in [\delta, 24]$, can be obtained. Specifically, in (2), $u(.)$ is a step function, and O_1 is an offset which can be obtained by a trial and error to bring $L^i_{\bar{w},0}$ closer to $L^i_{w,0}$. Inspired by the idea of curve fitting, the predicted load curve for h_δ until h_{24} will be adjusted by considering the load consumption trend of $\bar{w}$ between the δ-th (δ-1)-th hour to maintain a consistent load consumption level between the predicted load curve and the known load curve of the nearest workday. In the nearest workday prior to the incoming public holiday, the load consumption of the δ-th hour is generally higher than that of the (δ-1)-th hour, due to the human behavior. Therefore, the step function is applied to adjust the trend of the load curve.

DynaNC also predicts the load consumption of the compensatory holidays. We adopt temperature information to assist compensatory holiday load forecasting, because the load consumption of the compensatory holidays are not available. More specifically, it is well known that the temperature

$$L^i_{w,0} = L^i_{\bar{w},0} + sgn\Big(\sum_{i=1}^{\delta-1} \frac{L^i_{w,0} - L^i_{\bar{w},0}}{\delta - 1}\Big) O_1 + u(L^{\delta-1}_{w,0} - \{L^{\delta}_{\bar{w},0} + sgn\Big(\sum_{i=1}^{\delta-1} \frac{L^i_{w,0} - L^i_{\bar{w},0}}{\delta - 1}\Big) O_1\})(L^{\delta}_{\bar{w},0} - L^{\delta-1}_{\bar{w},0}) \tag{2}$$

$$\mathbf{W}[j,k] = |\mathbf{a}[j] - \mathbf{b}[k]| + \min(\mathbf{W}[j-1,k], \mathbf{W}[j,k-1], \mathbf{W}[j-1,k-1]), \quad 1 \leq j \leq 24,\ 1 \leq k \leq 24 \tag{4}$$

would affect the electricity usage which in turn influence the load consumption. Therefore, the temperature information is adopted. Further, the twenty-four-hour temperature information for $|d|$ days before x, $|N|$ years ago is denoted as $T^h_{d,N} = \{T^i_{d,N}\}$. The temperature of the compensatory holidays can be obtained by the ten-day weather forecast or weekends forecast released by official weather service organizations.

Furthermore, c is denoted as the $|c|$-th compensatory holiday prior to a holiday x, and $c < 0$. We in further denote this compensatory holiday falling on weekends as weekends compensatory holiday and denote the compensatory holiday falling on weekdays a weekdays compensatory holiday. It is because normally user behavior is different between weekends and weekdays which in turn would affect the consecutive load patterns among the nearest workday, weekdays compensatory holiday(s) and weekends compensatory holiday(s). For weekends compensatory holiday, DTW is also used to measure the similarity between temperature information of the $|c|$-th compensatory holiday, $T_{c,0}$, and temperature information of one or two weeks prior to the $|c|$-th compensatory holiday, $T_{j,0}, \forall j \in \{c-14, c-7\}$. We select the past day with the most similar temperature as days with similar temperature tend to have similar load consumption patterns. The chosen past day is denoted with $\bar{c}$. Then, for weekends compensatory holiday, $L^i_{\bar{c},0}, i \in [1, 24]$ can be obtained by (5), O_2 is also an offset to bring the load consumption value of $\bar{c}$ closer to the load consumption of the target weekends compensatory holiday, and can be obtained by a trial and error.

$$L^i_{c,0} = L^i_{\bar{c},0} + sgn\Big(\sum_{i=1}^{24} \frac{T^i_{c,0} - T^i_{\bar{c},0}}{24}\Big)O_2 \quad (5)$$

On the other hand, if the compensatory holiday falls on weekdays, we adopt the load consumption information on its previous day to predict the load consumption of this weekdays compensatory holiday. It is commonly known that temperature would influence the load consumption, therefore the weekdays compensatory holiday usage, $L^i_{c,0}, \forall i \in [1, 24]$ can be obtained with the assistance of temperature, $T^i_{c,0}$, the load consumption of its prior day, $L^i_{c-1,0}$, and an offset, O_3 obtained by try and error.

$$L^i_{c,0} = L^i_{c-1,0} - DTW(T^h_{c,0}, T^h_{c-1,0})O_3 \quad (6)$$

B. LSTM FOR HOLIDAY (LSTMH)

Fig. 2 shows the flowchart of LSTMH. In LSTMH, given x, LSTM is applied to predict $L^h_{0,0}$, and the prediction result is $\bar{L}^h$. LSTMH firstly pre-processes the input features, which is a set of several features, including $L^h_{d,0}$, $T^h_{d,0}$, $D^h_{d,0}$, $I^h_{d,0}$, where $d \in [w, -1]$ and h. In addition to the historical load consumption data, to predict the target holiday load consumption, we apply the temperature information for twenty-four-hour observation, $T^h_{d,N}$, since the temperature is known to related to the electricity load consumption. We also consider the day of the week indicator for $|d|$ days before x, $|N|$ years ago, which is denoted as $D^h_{d,N} = \{D^i_{d,N}\}, i \in [1, 24]$ aiming

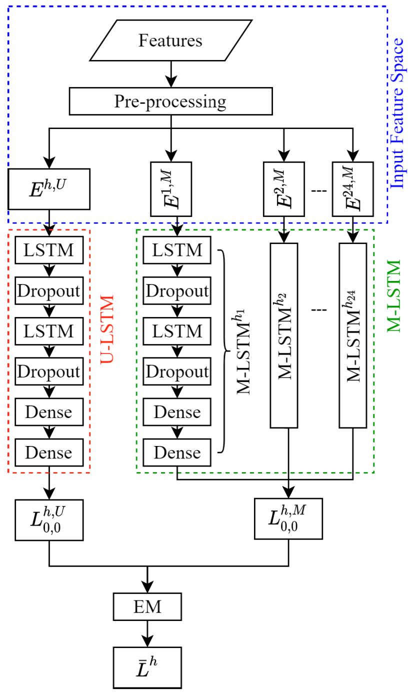


FIGURE 2. LSTMH flowchart.

to in further leverage the user behavior between weekdays and weekends for load forecasting. Specifically, $D^i_{d,N}$ can be obtained following (7). Moreover, $I^h_{d,N} = \{I^i_{d,N}\}, i \in [1, 24]$ is denoted as holiday occurrence indicator for $|d|$ days before x in the previous $|N|$-th year is also adopted to assist load forecasting. $I^i_{d,N}$ can be obtained by (8).

$$D^i_{d,N} = \begin{cases} 0, & |d| \text{ days prior to } x, |N|\text{-th years ago is Sun.} \\ 1, & |d| \text{ days prior to } x, |N|\text{-th years ago is Mon.} \\ 2, & |d| \text{ days prior to } x, |N|\text{-th years ago is Tue.} \\ 3, & |d| \text{ days prior to } x, |N|\text{-th years ago is Wed.} \\ 4, & |d| \text{ days prior to } x, |N|\text{-th years ago is Thu.} \\ 5, & |d| \text{ days prior to } x, |N|\text{-th years ago is Fri.} \\ 6, & |d| \text{ days prior to } x, |N|\text{-th years ago is Sat.} \end{cases} \quad (7)$$

$$I^i_{d,N} = \begin{cases} 1, & d = -1 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

The input features are pre-processed by min-max normalization and one-hot encoding. The former aims to scale number features, such as $L^h_{d,0}$ and $T^h_{d,0}$, between zero and one while the latter aims to transform categorical features, such as $D^h_{d,0}$, $I^h_{d,0}$, and h into binary values. Then, two time-series forecasting methods, U-LSTM and M-LSTM, are created by implementing univariate and multivariate method with LSTM, respectively. Considering multivariate approach, M-LSTM creates separate models to forecast the load consumption of each forecast period in h. Although multivariate approach allows fine-tuning for each forecast period in h, each model cannot learn the inter-dependency between different hour of the day because each model is created and trained independently. Additionally, multivariate method allows local normalization for features belonging on a specific forecasting period, so that the forecasting accuracy increases [9]. On the other hand, using univariate approach, U-LSTM creates a single model to predict over the twenty-four hour period. In this approach, the model can learn inter-dependency between each forecast period. However, it is less capable of accommodating volatility in the load consumption pattern because every hour of the day is predicted with a single model.

U-LSTM and M-LSTM are constructed using a uniform architecture since we focus on designing the forecasting framework, instead of optimizing the architecture itself. Specifically, as (6), these two models are trained by using the features selected from the historical data, including $h, L^h_{d,N}, T^h_{d,N}, D^h_{d,N}, I^h_{d,N}, d \in [w-14, w-1]$, in which year the holiday occurrence pattern is the same as the target holiday. Moreover, using multiple layers generally works better than a single layer, and thus we use two LSTM layers and two dense layers inside the model. Further, the difference between U-LSTM and M-LSTM lies in the shape of the input and output layer, while the depth of the model stays the same.

Additionally, we use dropout layers [21] as they help to reduce overfitting. M-LSTM and U-LSTM use a slightly different input features from each other. M-LSTM predicts $L^i_{0,0}$ using the features the i-th hour in the past $|p|$ days, where $i \in [1, 24]$ and $p < 0$. As shown in (9), $E^{i,M}$ is denoted as the set of the considered features for M-LSTM.

$$E^{i,M} = \begin{bmatrix} L^i_{-1,0} & T^i_{-1,0} & D^i_{-1,0} & I^i_{-1,0} \\ \vdots & \vdots & \vdots & \vdots \\ L^i_{p,0} & T^i_{p,0} & D^i_{p,0} & I^i_{p,0} \end{bmatrix} \tag{9}$$

On the other hand, U-LSTM considers twenty-four-hour holiday load for prediction all at once. Following [13], to predict the twenty-four-hour holiday load consumption, we use the past twenty-four-hour features, denoted as a matrix $E^{h,U}$ illustrated as (10).

$$E^{h,U} = \begin{bmatrix} L^h_{-1,0} & T^h_{-1,0} & D^h_{-1,0} & I^h_{-1,0} & h \end{bmatrix} \tag{10}$$

When $L^{h,M}_{0,0}$ and $L^{h,U}_{0,0}$, which are denoted as the twenty-four-hour predicted load consumption for the target holiday from M-LSTM and U-LSTM, respectively, are available, the ensemble layer (EM) combines $L^{h,M}_{0,0}$ and $L^{h,U}_{0,0}$ to further increase forecasting performance [22]. Specifically, EM, consisting of a neural network layer, is trained using $L^{h,M}_{0,N}$ and $L^{h,U}_{0,N}$, if the nearest workday of the $|N|$-th past year is the same as w. It is because the number of compensatory holidays would vary the load pattern, according to Fig. 1.

V. EVALUATION RESULTS

A. EXPERIMENTAL SETUP

In this section, we analyze the prediction results for nine holidays in Asia. To ease of presentation, the holiday index (x) for each holiday is set according to the corresponding w. Specifically, $x \in \{1, 6\}$ refers to $w = -1$, $x = 7$ refers to $w = -2$, $x = 8$ refers to $w = -3$, and $x = 9$ refers to $w = -4$. Nine years hourly load consumption and temperature data are applied for performance evaluation. Note that the temperature forecasting is out of scope of this paper, and thus we assume the temperature is predicted correctly in the load forecasting. Further, mean absolute percentage error (MAPE) and peak prediction error are used to measure the prediction performance of a method.

$$MAPE = \frac{1}{24}\sum_{i=1}^{24} \left|\frac{\bar{L}^i - L^i_{0,0}}{L^i_{0,0}}\right| \times 100\%$$

We also consider the peak prediction error denoted as

$$\max_{i\in[1,24]} |\bar{L}^i - L^i_{0,0}|$$

since the peak prediction error is also an important factor.

In M-LSTM, following [5], we set p to be -7 to capture the auto-regressive effect of a week. To set the weight in dropout layers, following [12], we use 0.2 and 0.25 in M-LSTM and U-LSTM, respectively. Further, Adam optimizer is chosen as it is known to be the best performing optimizer for time-series prediction task [23]. In addition, to prevent over-fitting JDL, we set the number of epochs when training M-LSTM, U-LSTM, and ensemble layer to be 40, 40, and 20, respectively. Further, to investigate the gain from M-LSTM and U-LSTM, we introduce JDML and JDUL which only perform U-LSTM and M-LSTM, respectively while executing the LSTMH.

Moreover, we create two benchmark models to investigate the performance on holiday load forecasting by pure-LSTM based approaches. Firstly, we adopt the idea of [15] to create multi-model (MM) strategy to investigate the impact of using LSTM to predict the nearest workday and compensatory holiday(s). To deal with the data insufficiency problem, we adopt LSTM to create the augmented learning strategy (AL). AL selects two weeks before and after the public holiday as training data, for each historical year whose nearest workday is the same as that of the target holiday. The obtained augmented data will be used to train a U-LSTM model predicting the load consumption of the nearest workday, compensatory holiday(s), and the public holiday itself by a recursive manner.

B. RESULTS

Fig 3 shows the obtained MAPE of different holidays. Among the applied methods, the proposed JDL outperforms others, and the achieved MAPEs for all evaluated holidays are approximately less than 4%, regardless the number of compensatory holiday(s). Further, JDML obtains better results than JDUL for most cases. JDML predicts the i-th hour of the target holiday using the features at the same hour of the consecutive past seven days which better captures the trends of the load usage. JDUL takes the features in the previous day to predict the target holiday all at once. It would better capture the characteristics of the time series at the same day. However,

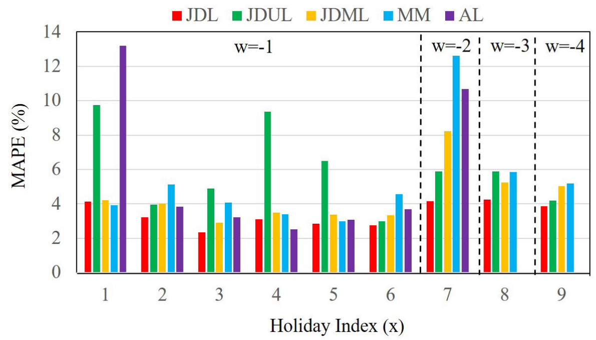


FIGURE 3. Prediction error for holidays.

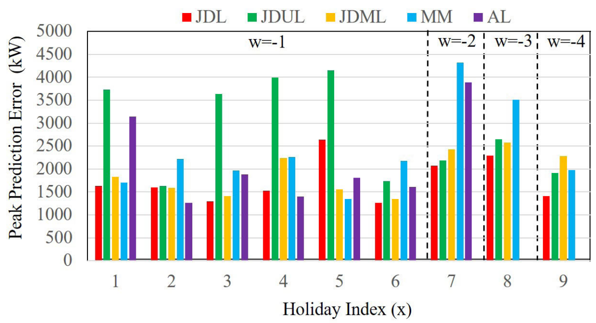


FIGURE 4. Peak prediction error for holidays.

the proposed JDL takes advantages of both U-LSTM and M-LSTM with the assistance of ensemble layer so that its MAPEs are low. The MAPEs of MM and AL fluctuate from a holiday to another. MM obtains 4-6% MAPEs when no compensatory holidays exist, but the MAPEs increase in the presence of compensatory holidays due to data contamination in predicting the compensatory holidays. On the other hand, in AL, the MAPEs are low when proper features are found for LSTM to capture the characteristic of load consumption pattern for the corresponding holiday. But, if the number of the same holiday occurrence pattern is insufficient, AL may obtain large MAPEs, such as $x = 1$ and 7. Additionally, the MAPE of AL is inapplicable, when $w \leq -3$, because the input data is insufficient for training due to the rare occurred holiday patterns of the corresponding holidays.

Fig. 4 presents the peak prediction error of each holiday. It can be found in both Fig. 3 and 4, for all methods, with the increasing of the compensatory holidays, both MAPE and the peak prediction error increase. The holiday load consumption is predicted at certain hour of the nearest workday, and the load pattern highly relies on the past consecutive days. Thus, the prediction error of the load consumption of the compensatory holiday(s) would propagate to the target holiday load forecasting. Under this circumstance, the MAPE and the peak prediction error increase when the number of compensatory holiday(s) increases.

Obviously, the proposed JDL achieves the lowest error for most holidays. Because of the careful design of DynaNC and LSTMH to predict the load consumption of compensatory holidays and target holiday, JDL is the most robust method.

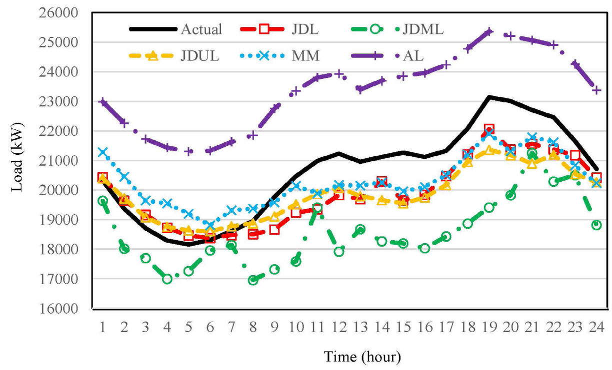


FIGURE 5. Holiday load prediction when no compensatory holidays occur.

It not only achieves small MAPEs (2-4%) but also small peak prediction error (1500-2500 (kW)). Further, JDL adopts the ensemble layer to balance the time and day series correlations, and thus both MAPEs and peak prediction errors are less than 4%. Although other methods can achieve less than 4% MAPEs and 2500 (kW) peak prediction error in some cases, they are not as robust as JDL. Existing methods are mainly designed to perform one day ahead load forecasting. When compensatory holidays present, additional prediction error would be introduced, because the holiday load consumption is predicted more than one days ahead.

To further investigate the behavior of holiday load forecasting, we individually select a holiday for different number of compensatory holidays and evaluate the corresponding load usage of the chosen holiday. Fig. 5 - 8 show the twenty-four-hour holiday load forecasting along with the actual load usage under different number of compensatory holidays. As seen in Fig. 5, the x-axis indicates the i-th hour of the holiday, and the y-axis is the corresponding load usage when no compensatory holidays present prior to the target holiday (i.e., $x = 1$). The black line without markers is the actual load usage, and the rests are the predicted load consumption though different methods. The load pattern of JDL between the 1st and the 6th hour is very close the that of actual load. From the 10th to the 24th hour, the load prediction curve of JDL is the closest one to the actual load. Therefore, JDL obtains the smallest MAPE and peak prediction error. The load forecasting curves of JDUL and MM are also close to actual load, and thus their MAPEs and peak prediction errors are small. Although the pattern of AL is similar to the actual load, large load prediction offset exists between AL and the actual load resulting in large MAPE and peak prediction error. JDML does not capture the pattern and the load prediction offset properly. Because it only uses features of the same hour for prediction, its MAPEs and peak prediction error are larger than JDL.

Fig. 6 shows the holiday load usage for $x = 7$ in which one compensatory holiday exists. As seen in Fig. 6, the overall load forecasting curve of each method has certain offsets comparing to the actual load. In the presence of one compensatory holiday, the prediction error increases

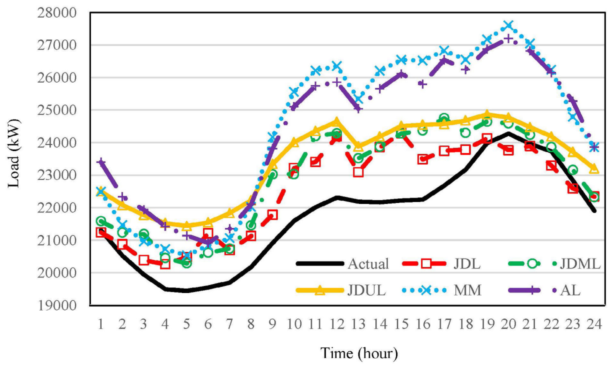


FIGURE 6. **Holiday load prediction when one compensatory holiday occurs.**

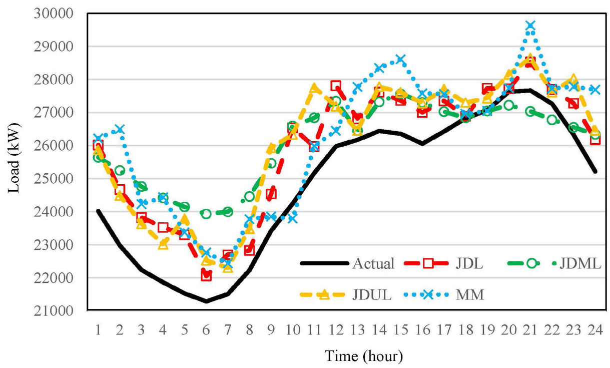


FIGURE 7. **Holiday load prediction when two compensatory holidays occur.**

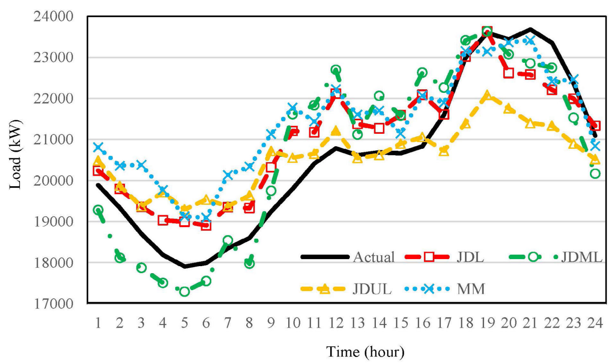


FIGURE 8. **Holiday load prediction when three compensatory holidays occur.**

in comparison with the absence of compensatory holidays. More specifically, JDL obtains the closest load forecasting curve to the actual load which indicates the most precise load forecasting result. The load forecasting curves of JDML and JDUL are similar, and JDML predicts load usage better than JDUL before the 8th hour. It may because JDML takes more days information of the same hour for prediction, and then the MAPE could be improved. Further, MM and AL better predict the load consumption before the 8th hour, and the prediction errors significantly increase since the 8th hour. Therefore, the MAPEs and peak prediction errors are large.

Similarly, Fig. 7 and 8 present the holiday loads for $x = 8$ and 9, respectively. It can be observed that in Fig. 7, the prediction curves of all methods are close to each other. In the presence of two or three compensatory holidays, the prediction errors increase for both MM and JDUL because they lack of accurate one day ahead load usage. It is worth to notice that JDUL achieves better forecasting results at certain hours on both holidays, and thus its performance is better than MM. JDML consider multiple days ahead load usage to better capture the characteristic of the load pattern for most time intervals resulting in better performance than JDUL. Moreover, JDL take advantages of both JDML and JDUL to achieve the best forecasting results.

Overall, JDL is more reliable in terms of MAPE, the peak prediction error and the twenty-four-hour load forecasting pattern under different holiday status. The performance can be enhanced in further by considering more complicated ensemble layers or features to capture more characteristic of the holiday load consumption to assist prediction.

VI. CONCLUSION

In this paper, we propose JDL to perform holiday hourly load forecasting on the nearest workday prior to the incoming public holiday. To react to the characteristic of different prediction horizon on the nearest workday, JDL combines the similarity-based method and deep learning. The similarity-based method, DynaNC, utilizes DTW to predict the load consumption of the nearest workday and compensatory holiday(s), if any. The deep learning-based method, LSTMH, utilizes LSTM to predict the load consumption of a public holiday.

Nine holidays in Asia are applied to evaluate our proposed JDL along with two methods from the literature. The results show the proposed JDL is most reliable and robust in forecasting the holiday load consumption. The prediction errors of JDL fall on the range of 4%. In the future, we will investigate the design of temperature prediction to enhance the load forecasting accuracy and reliability. Moreover, the design of a better ensemble layer would also be studied to improve the performance of load prediction.

ACKNOWLEDGMENT

This work was partially supported by the Taiwan Building Technology Center, Taiwan Tech. from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education, Taiwan, and the Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. MOST 108-2221-E-011-058-MY3.

REFERENCES

[1] M. A. Ortega-Vazquez and D. S. Kirschen, "Economic impact assessment of load forecast errors considering the cost of interruptions," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Jun. 2006, pp. 1–8.

[2] B. F. Hobbs, S. Jitprapaikulsarn, S. Konda, V. Chankong, K. A. Loparo, and D. J. Maratukulam, "Analysis of the value for unit commitment of improved load forecasts," *IEEE Trans. Power Syst.*, vol. 14, no. 4, pp. 1342–1348, Nov. 1999.

[3] A. Almalaq and J. J. Zhang, *Deep Learning Application: Load Forecasting in Big Data of Smart Grids*, vol. 865. Springer, 2020, pp. 103–128.

[4] R. S. S. Fernandes, Y. K. Bichpuriya, M. S. S. Rao, and S. A. Soman, "Day ahead load forecasting models for holidays in Indian context," in *Proc. Int. Conf. Power Energy Syst.*, Dec. 2011, pp. 1–5.

[5] F. Ziel, "Modeling public holidays in load forecasting: A German case study," *J. Modern Power Syst. Clean Energy*, vol. 6, no. 2, pp. 191–207, Mar. 2018.
[6] Q. Ding, H. Zhang, T. Huang, and J. Zhang, "A holiday short term load forecasting considering weather information," in *Proc. Int. Power Eng. Conf.*, Nov. 2005, pp. 1–61.
[7] Y.-M. Wi, S.-K. Joo, and K.-B. Song, "Holiday load forecasting using fuzzy polynomial regression with weather feature selection and adjustment," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 596–603, May 2012.
[8] Y. Cai, Q. Xie, C. Wang, and F. Lu, "Short-term load forecasting for city holidays based on genetic support vector machines," in *Proc. Int. Conf. Electr. Control Eng.*, Sep. 2011, pp. 3144–3147.
[9] A. Ahmad, N. Javaid, A. Mateen, M. Awais, and Z. A. Khan, "Short-term load forecasting in smart grids: An intelligent modular approach," *Energies*, vol. 12, no. 1, pp. 1–21, Jan. 2019.
[10] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943–3952, Jul. 2019.
[11] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.
[12] M. Tan, S. Yuan, S. Li, Y. Su, H. Li, and F. H. He, "Ultra-Short-Term industrial power demand forecasting using LSTM based hybrid ensemble learning," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 2937–2948, Jul. 2020.
[13] M. S. Hossain and H. Mahmood, "Short-term load forecasting using an LSTM neural network," in *Proc. IEEE Power Energy Conf. Illinois (PECI)*, Feb. 2020, pp. 1–6.
[14] J. Zheng, C. Xu, Z. Zhang, and X. Li, "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6.
[15] L. Tang, Y. Yi, and Y. Peng, "An ensemble deep learning model for short-term load forecasting based on ARIMA and LSTM," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2019, pp. 1–6.
[16] P. Zeng, C. Sheng, and M. Jin, "A learning framework based on weighted knowledge transfer for holiday load forecasting," *J. Mod. Power Syst. Clean Energy*, vol. 7, pp. 329–339, Dec. 2019.
[17] G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, and E. Keogh, "Generating synthetic time series to augment sparse datasets," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 865–870.
[18] R. Asadi and A. C. Regan, "A spatio-temporal decomposition based deep neural network for time series forecasting," *Appl. Soft Comput.*, vol. 87, pp. 1–13, Feb. 2020.
[19] Z. Yu, Z. Niu, W. Tang, and Q. Wu, "Deep learning for daily peak load forecasting—A novel gated recurrent neural network combining dynamic time warping," *IEEE Access*, vol. 7, pp. 17184–17194, 2019.
[20] S. Theodoridis and K. Koutroumbas, *Template Matching*. San Diego, CA, USA: Academic, 2009, pp. 324–325.
[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[22] F. Ziel and R. Weron, "Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks," *Energy Econ.*, vol. 70, pp. 396–420, Feb. 2018.
[23] S. Liu, H. Ji, and M. C. Wang, "Nonpooling convolutional neural network forecasting for seasonal time series with trends," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2879–2888, Aug. 2020.

JEFFREY GUNAWAN received the B.Eng. degree from the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 2018, and the master's degree (M.S.) in electronics and computer engineering from the National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, in 2021, under the supervision of Prof. Chin-Ya Huang. He is currently working as a software engineer for a startup company specializing in applied artificial intelligence for industrial applications.

CHIN-YA HUANG (Member, IEEE) received the B.S. degree in electrical engineering from National Central University, Taiwan, in 2004, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 2006, and the M.S. degree in electrical and computer engineering, the M.S. degree in computer science, and the Ph.D. degree in electrical and computer engineering from the University of Wisconsin-Madison, USA, in 2008, 2010, and 2012, respectively. She worked as an Assistant Research Fellow at Information and Communication Tech. Laboratory, National Chiao Tung University, Taiwan, in 2013. From 2012 to 2013, she was a Senior Software Engineer at Qualcomm Tech. San Diego, CA, USA. She worked as a member of Technical Staff with Optimum Semiconductor Tech. Inc., NY, USA, from 2013 to 2015. She worked as an Assistant Professor with National Central University, from 2015 to 2018. Since 2018, she has been working as an Assistant Professor with the National Taiwan University of Science and Technology (NTUST), Taiwan. Her research interests include next-generation wireless networking, software-defined networking, and real-time computing.