Received October 15, 2021, accepted November 28, 2021, date of publication December 7, 2021, date of current version December 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3133702

A Short-Term Household Load Forecasting Framework Using LSTM and Data Preparation

DERNI AGENG, CHIN-YA HUANG, (Member, IEEE), AND RAY-GUANG CHENG, (Senior Member, IEEE)

Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 10605, Taiwan

Corresponding author: Chin-Ya Huang (chinya@gapps.ntust.edu.tw)

This work was supported in part by the AAEON under Contract NTUST-AAEON-9169; in part by the Ministry of Science and Technology (MOST), Taiwan, under Contract 110-2224-E-011-002; and in part by the ''Center for Cyber-physical System Innovation'' from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan.

ABSTRACT IoT devices are deployed in a building to instantly collect electricity load usage for next hour load consumption forecasting so that the operation of the building can be properly managed. However, due to the hardware or system error, data missing or overflow might occur. Noise might also be added into the collected data. Under this circumstance, the accuracy of next hour load forecasting would degrade which in turn decreases the building operation performance. In this paper, we propose an hourly load forecasting framework combining Data Preparation and LSTM, namely LSTM-DP, by considering data pre-processing, feature engineering and Long Short-Term Memory (LSTM). In LSTM-DP, the collected data is firstly processed by interpolation and Savitzky Golay filter, therefore the pattern of load consumption can be properly extracted by LSTM for next hour load forecasting. Moreover, we adopt two-stack LSTM to better determine the relationships among the time series information. We study the real data collected from three buildings of a company in Asia to investigate the performance of next hour load forecasting, and the results show the proposed LSTM-DP outperforms others.

INDEX TERMS Short-term load forecasting, long short-term memory (LSTM), smart meter, data pre-processing, feature engineering.

I. INTRODUCTION

Smart Meter Infrastructure (SMI) development enables bi-directional communication between the power company and individual users [1]. By integrating the SMI in a building, the collected data in the database can be applied for various applications. For example, short-term load forecasting is one of the popular applications utilizing SMI [2], [3]. More specifically, with accurate short-term load forecasting, individual customers can better plan their usage such as shifting their usage to off-peak periods whenever possible so that they can avoid massive consumption or overpayment. However, the short-term load forecasting, especially hourly load forecasting, highly relies on past historical data obtained from SMI because of the dynamic behavior of each user from one hour to another. Specifically, the dynamic behavior of each user affects the pattern of load usage which plays a main role in hourly load forecasting. Furthermore, transmission and technical errors also affect the load usage pattern which in turn may degrade the short-term load forecasting accuracy.

Due to transmission and technical errors in the system, the data might contain missing values, overflow values, noise and so on [4], [5] which would produce a lousy prediction result [2], when such data is directly used for load prediction. Hence, many studies have addressed the importance of data pre-processing. Due to the volatile nature of the load data, data smoothing and data imputation are performed in previous studies. In [2], authors apply a moving average filter to refine the load data. The moving average filter is quite helpful to refine the dataset. However, it only uses the mean value of the defined number of windows. If the sequence of the window contains mod of the data, the filled value will ruin the

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

sequence of the data. In [6], mean, median, Savitzky-Golay, Kalman, and Gaussian filter are applied to smooth the digital sensor data for further investigation.

Data imputation is also introduced in recent studies. Due to system/human error, the data sent to the database might be lost or wrong. In [7], authors introduce an improved random forest algorithm to fill the missing values aiming to improve the prediction performance. Because the filling result depends on the average vote from all the trees, the pattern of that sequence would be ruined. To fill the missing values, KNN is also used. However, as mentioned in [8], KNN requires the use of the training data in the application process of imputing unseen data. It takes more computation time on a large-scale dataset. Additionally, in [9], K. Liang *et al.* use selective ensemble learning to predict load consumption. They find many missing data for 1-3 months in their dataset, and they chose to fill the missing values with the average value. Otherwise, they will eliminate the value. They also use K-Means to make a cluster of load consumption behavior, but using the average value of the dataset to fill the missing value may break down the pattern of a certain sequence.

Previous studies introduce Long Short-Term Memory (LSTM) to perform load forecasting [3], [9]–[11]. In [3], [11], increasing number of layers in LSTM is introduced for the models to learn the sequence deeper so that load forecasting performance is improved. However, too many stacks of LSTM layers may overfit the data during the training process. Then, as [1], [2], [12], to increase the accuracy of load forecasting, hybrid models combining deep learning models with another machine learning model are proposed. Specifically, authors combine Convolutional Neural Network (CNN) and LSTM to construct a load forecasting framework. The purpose of using CNN is to perform feature extraction and facilitate various hidden features of the load sequences to provide accurate load forecasting. Ullah *et al.* in [2] present an intelligent hybrid technique that combines the CNN with a Multi-layer Bi-directional Long-Short Term Memory (M-BDLSTM). M-BDLSTM integrates the pre-processing and data organization mechanisms to refine the data and remove abnormalities. The refined data is fed to CNN first, and thus M-BDLSTM can learn the sequence pattern effectively which increases the prediction accuracy. However, M-BDLSTM only uses one feature to predict the load consumption which would limit the load forecasting performance. Under this condition, feature engineering would need.

Moreover, machine learning based approaches are also commonly applied for modeling load consumption. Authors in [13] propose an improved random forest algorithm to perform load forecasting. They also introduce fuzzy clustering to improve the load forecasting performance. However, a prediction from random forest is an average of the predictions produced by the trees in the forest which might be biased by the presence of miss or overflow data. Meanwhile, authors in [14] use K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) to develop a load forecasting framework. The combination of KNN and SVM produces competitive performance. However, SVM is not suitable for large datasets. If the number of features for each data point exceeds the number of training data samples, the SVM would underperform. In addition to deep learning or machine learning based approaches, statistical approaches are also applied for load forecasting. In [15], authors propose Auto-Regressive Integrated Moving Average (ARIMA) to perform peak load forecasting, which incorporates the time series modeling with the knowledge of experienced human operators. It is shown that the proposed method can exactly forecast the daily peak load of a power system. Although ARIMA could improve performance, it is generally less flexible. If the data is generated by a process similar to ARIMA assumptions, it will work well. In [16], authors apply regression to perform load forecasting. The advantage of using regression to perform load forecasting is that it only needs a small amount of data. However, the performance would be limited since the pattern of the load usage might not be always linear. Due to dynamic user behavior, linearities might seldom occur. Also, the outliers in the dataset, such as missing or overflow values, would break the linearity of the data.

In this paper, we investigate the real data collected from three buildings of a company in Asia and propose an hourly load forecasting framework combining Data Preparation and LSTM, namely LSTM-DP, for hourly load forecasting. To handle the abnormal data, the Data Preparation (DP) strategy in LSTM-DP pre-processes the load consumption data through interpolation and filtering. The DP strategy also contains feature engineering to select several appropriate features to assist load forecasting so that the prediction accuracy can be improved. On the other hand, the LSTM for load forecasting (LSTM) strategy in LSTM-DP uses LSTM to identify short-term and long-term dependencies. We adopt two stacks of LSTM layers to let the model learn deeper along with the sequences to further enhance the forecasting performance.

The remainder of the paper is organized as follows. Exploratory data analysis and problem identification are provided in Section II. The proposed hourly load forecasting framework combining Data Preparation and LSTM (LSTM-DP) is presented in Section III. The case study and experimental results are elaborated in Section IV. Finally, the work is concluded in Section V.

II. EXPLORATORY DATA ANALYSIS AND PROBLEM IDENTIFICATION

The dataset contains the data from three buildings of a company in Asia from July until October 2020. The buildings are denoted as BU1, BU2 and BU3, and the data is collected every five minutes from smart power meters. We denote the load consumption of a building as p_t, where t is the t-th hour within a target time interval. We also denote the predicted load usage of a building the t-th hour from a target time interval as $\hat{p}_t$, and $\hat{p}_t$ is obtained at the (t-1)-th hour.

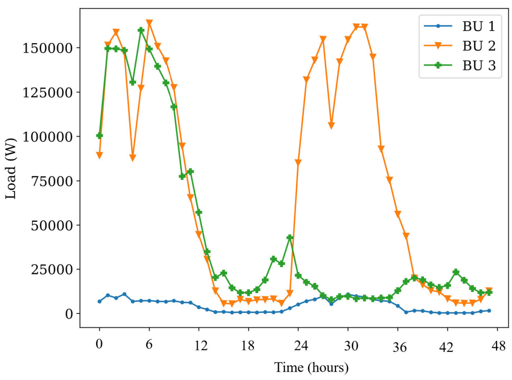

FIGURE 1. Hourly load data in three buildings.

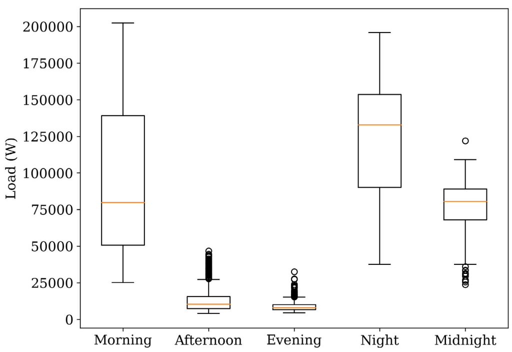

FIGURE 2. Distribution of load consumption in different session time.

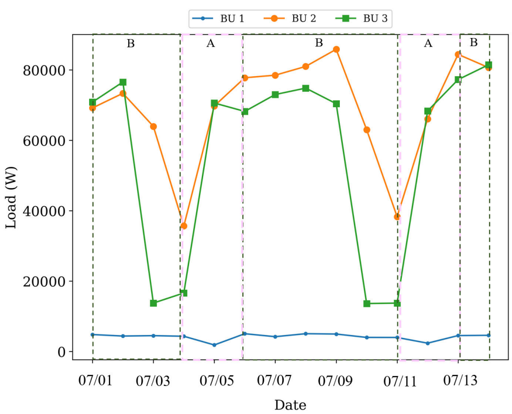

FIGURE 3. Load consumption pattern in weekdays/weekends.

Unlike smart grid, individual load data in a building lacks the obvious consistent patterns from others. Fig. 1 shows the load usages of three buildings at the same time interval over two consecutive days (i.e., 48 hours). It can be seen the load consumption patterns among these three buildings are totally different. Different numbers of people with different habits as well as different events in each building might make the load usage volatile and practically changing every time [17]. Therefore, simply using the data from one building to perform load forecasting for each building is challenging.

To further observe the load consumption based on the time interval of a day, we divide the timestamp into five time zones and call the time interval of each time zone as "Session Time". The session times consist of 'Morning', 'Afternoon', 'Evening', 'Night' and 'Midnight'. The session time 'Morning' is 6 AM - 12 PM, 'Afternoon' is 12 PM - 6 PM, 'Evening' is 6 PM - 12 AM, 'Night' is 12 AM - 6 AM, and 'Midnight' is 12 AM - 1 AM. Fig. 2 shows the relationship between the distribution of load consumption in a building and each session time in a day. For example, the load usage in the 'Night' session is mainly distributed between 100,000 and 150,000 (W) while those in 'Afternoon' and 'Evening' session are less than 25,000 (W). Moreover, in the 'Morning' session, the variation of the load consumption is much larger than others. For example, it can reach about 200,000 (W), but its distribution is mainly between 50,000 and 125,000 (W). Hence, this feature would be helpful to assist the hourly load forecasting.

On the other hand, based on Fig. 3, the box A and B represent the load consumption during the weekends and weekends, respectively. The load consumption is fluctuated significantly during the weekends. Further, it can be seen that more usage on weekdays than weekends which might be resulted from human behavior. Moreover, the data separated from the index of weekdays or weekends would also help hourly load forecasting.

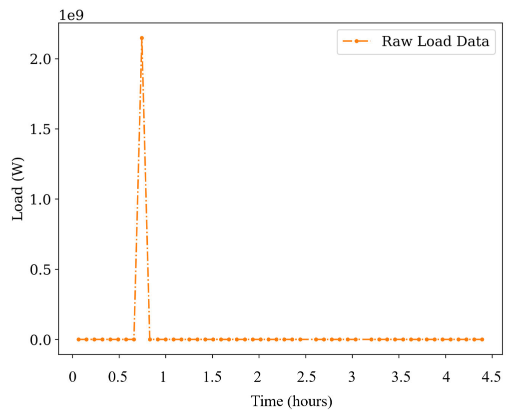

FIGURE 4. Original load data.

However, some issues are found in the collected dataset which cause the challenges in hourly load forecasting. Due to the system failure and human error, the data might be wrong or lost during transmission [5]. As shown in Fig. 4, the overflow value is sent to the database which is significantly larger than the load usage present in Fig. 2 and 3. Such anomaly data would produce lousy prediction if it is used to forecast load usage. Specifically, the distribution of the load usage would be influenced, and thus the accuracy in load forecasting decreases. Moreover, some of the load data collected from the smart power meter has missing values. We find each building contains missing data in Fig. 5, and more than 30% data of BU1 is missing. The missing value would also influence the load forecasting accuracy. As a result, a method is needed to pre-process such abnormal behaviors before we use the data for load forecasting. Additionally, due to system problems, individual mistakes or measurement errors, noise would also be introduced in the collected dataset. Similar to [2], the curve of load across several days shown in Fig. 6 contains some spikes on the top side of the data which might be introduced by the noise. The presence of noise would degrade the load forecasting performance, and thus a de-noising method would require to remove such advert effect.

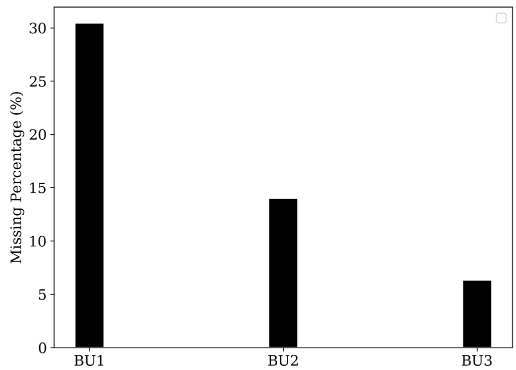

FIGURE 5. Missing percentage of the each building.

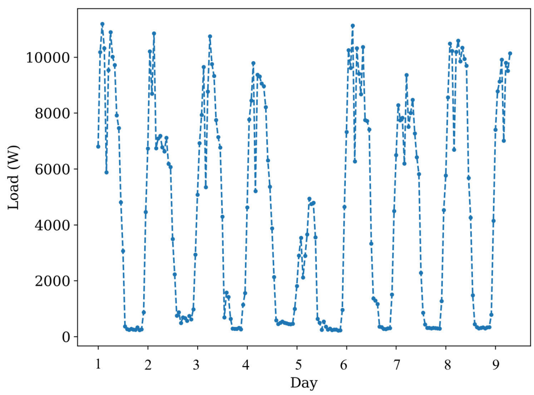

FIGURE 6. Illustration of noise data.

III. HOURLY LOAD FORECASTING FRAMEWORK COMBINING DATA PREPARATION AND LSTM (LSTM-DP)

To effectively predict the next hour load consumption, we propose hourly load forecasting framework combining Data Preparation and LSTM, namely LSTM-DP. LSTM-DP includes Data Preparation (DP) strategy and LSTM for load forecasting (LSTM) strategy. In DP strategy, we apply data pre-processing and feature engineering in order to prepare the input for load forecasting by LSTM. Specifically, data pre-processing aims to deal with missing, overflow and noisy data in the dataset to enhance the forecasting accuracy by interpolation, smoothing and de-noising approaches. On the other hand, feature engineering aims to select proper features to assist the forecasting. Moreover, we consider the model of two stacked LSTM with dropout and dense layers to forecast load consumption.

A. DATA PREPARATION (DP) STRATEGY

1) DATA INTERPOLATION

We apply interpolation to fill the missing or overflow data caused by system error to assist load forecasting. The missing value is interpolated using the average of the previous and next value. For example, if the value of the t-th point is missing or overflow, it is interpolated by $p_t = \frac{p_{t-1}+p_{t+1}}{2}$. After interpolating all missing points, we re-sample the dataset into hourly load consumption for next-hour load forecasting.

2) DATA SMOOTHING AND DE-NOISING

Data smoothing is further applied to remove the noise caused by climate, metering problems, and individual mistakes or measurement errors [2]. More specifically, we intend to refine the load data with Savitzky Golay Filter [6], and denote the refined value at t-th time as P_t.

$$P_t = \sum_{i=\frac{1-w}{2}}^{\frac{w-1}{2}} C_i \times p_{t+i}, \tag{1}$$

where w is the window size, C_i is the coefficient for weighting function which already declared in [18], p_{t+i} is the unfiltered value at the $(t+i)$ time, and $t = 1, \ldots, n$ within the target time interval.

3) FEATURE ENGINEERING

To select appropriate features for hourly load forecasting, we extract data from our dataset. Firstly, based on the dataset, the hourly load consumption correlates with time. As shown in Fig. 3, the pattern of load consumption behaves differently between weekdays and weekends. Further, the distribution of load usage varies from one session time to another as illustrated in Fig. 2. Thus, the hourly load consumption, weekdays/weekends indicator and the session time are firstly taken as features for load forecasting.

Moreover, we check all of the features retrieved from data analysis and feature engineering with Spearman Correlation. We select the feature that has a strong relationship with load data as additional features. Specifically, the total apparent power data is related to the load consumption since it is a combination of reactive power and load data. Total apparent power relates to the power triangle, which includes reactive power and load, therefore it is also selected as a feature. As mentioned in [19], weather temperature can also cause changes in load consumption. Hence, we collect temperature datasets from the government site, and then take an average temperature value of each hour as another feature.

Finally, given each selected feature, we consider the information of the past k consecutive timesteps (hours) prior to the target hour t, and thus the features fed into LSTM for load forecasting are summarized as below:

1) Load consumption: $P = \{P_{t-k}, \ldots, P_{t-2}, P_{t-1}\}$.
2) Total apparent power: $A = \{a_{t-k}, \ldots, a_{t-2}, a_{t-1}\}$, where a_i is the apparent power of the i-th point within the target time interval.
3) Temperature: $C = \{c_{t-k}, \ldots, c_{t-2}, c_{t-1}\} \in R^k$, where c_i is the temperature of the i-th point within the target time interval.
4) Weekdays/Weekends indicator: $H = \{h_{t-k}, \ldots, h_{t-2}, h_{t-1}\}$, where h_{t-k} is the indicator of the i-th point within the target time interval. It is 0 for weekdays and 0 for weekends.
5) Session time indicator: $S = \{s_{t-k}, \ldots, s_{t-2}, s_{t-1}\}$, where s_i is the session time indicator of the i-th point within the target time interval. It is set as 0, 1, 2, 3 and 4 for 'Night', 'Midnight', 'Morning', 'Afternoon' and 'Evening', respectively.

B. LSTM FOR LOAD FORECASTING (LSTM) STRATEGY

The architecture of the proposed LSTM based model for hourly load forecasting is shown in Fig. 7. The model consists of two stacked LSTM with dropout and dense layers. LSTM is commonly used for short-term load forecasting [1] as it is able to learn and capture the pattern along with the sequences. Previous studies prove that the additional stacks on LSTM could give benefits and disadvantages to the training model [11], and thus we adopt two stack for our model.

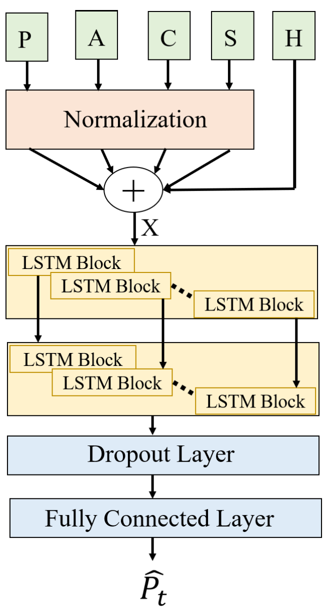

FIGURE 7. Process of LSTM strategy for load forecasting process in LSTM-DP.

Since LSTM is slightly sensitive to the value from the input, we normalize the input first, as shown in the Normalization block in Fig. 7. We use a min-max scaler to rescale P, A, C and S between 0 and 1, and denote the scaled feature as $\tilde{P}, \tilde{A}, \tilde{C}, \tilde{S}$, respectively. Then we denote X for the LSTM input matrix, and $X = \{\tilde{P}, \tilde{A}, \tilde{C}, H, \tilde{S}\}$. Each row of X is the scaled features for the corresponding timestep, which feeds sequentially in the LSTM block. Then, each LSTM block produces an output which is further fed into other LSTM blocks to determine which information is used for next sequence. The return sequence of the first stack of LSTM is set to true to pass the hidden state to the following stack. The return sequence is set to false in the following stack as it reaches the final step. The output of the learning is connected to the dropout layer. The dropout layer is used to keep essential nodes and avoid overfitting before the fully connected layer (Dense Layer) produces the forecasting result, $\hat{p_t}$.

Before performing prediction, we crate a training dataset to select best performing models. After running some initial tests, we find that the dropout layer with a 0.1 rate slightly improves the load forecasting model. We set 1000 epoch for the experiments with a stop condition. If the model fails to get a better result in 30 epochs, the training stops. We also create a validation dataset to test the trained model after the training process. Moreover, Adam is used to train the proposed forecasting framework with recommended default parameters (learning rate, momentum, and decay), since the Adam optimizer showed better results than the other candidates such as RMSProp and SGD in our experiments.

IV. EVALUATION RESULTS

In the case study, we evaluate the proposed LSTM-DP through the dataset collected from BU1, BU2 and BU3 to perform one hour ahead load forecasting. For each building, we pick 123 days data from July 1st, 2020 to October 31st, 2020 which covers a part of the summer season, an entire fall season and the beginning of the winter season so that seasonal factors can be reasonably ignored in this case study. For each building, we split the 123 days data into three subsets, which are the training set (July 1st to August 30th), validation set (August 31st to September 30th), and the testing set (October 1st to October 31st). Further, we set k as 9 indicating the previous 9 points of the load data prior to the target hour are applied to forecast next hour load usage. We also define several hyper-parameters in our load forecasting model. Table 1 shows a dropout layer rate, activation function, stopping condition, epoch, and optimizer function. All of the experiments are built on Google Colab notebooks. Furthermore, the mean absolute error percentage (MAPE) is used to evaluate the forecasting performance. Considering n is the total points of within the testing dataset, the MAPE is calculated as

$$MAPE = \frac{\sum_{t=1}^{n} |\frac{\hat{p_t} - p_t}{p_t}|}{n} \times 100\%. \tag{2}$$

A. RESULTS

1) THE IMPACT OF METHOD IN FILLING THE MISSING/OVERFLOW DATA

In this experiment, we compare LSTM-DP with other methods replacing the interpolation in LSTM-DP for load forecasting. Specifically, we investigate the prediction results using different methods to fill missing/overflow values. These methods include average value [9], random

TABLE 1. Hyper-parameter summary.

Parameter	Details
Dropout Layer	0.1
Activation on Dense Layer	Linear
Optimizer	Adam
Stopping Condition	40 Patience
Epoch	1000

forest algorithm [7], and KNN [20]. We name them as LSTM-DP(Average), LSTM-DP(Random Forest) and LSTM-DP(KNN), respectively, as only the method for filling missing value in LSTM-DP is changed.

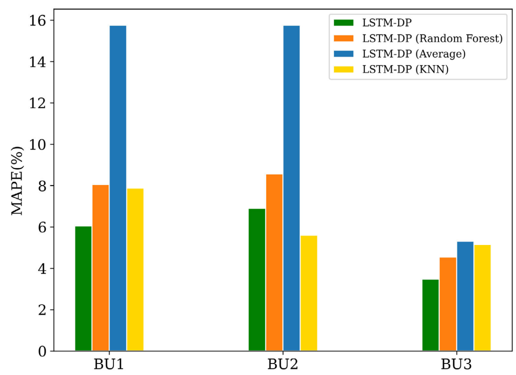

FIGURE 8. MAPE of various filling methods.

TABLE 2. Peak error of various filling methods.

	Peak Error (W)
LSTM-DP	12593.375
LSTM-DP (Average)	99649.972
LSTM-DP (Random Forest)	99703.956
LSTM-DP (KNN)	98478.93

As present in Fig. 8, LSTM-DP obtains the best performance. It is because the filled value from interpolation only involves the nearest value from the missing values, the filled value from interpolation certainly closes to the actual value. Meanwhile, LSTM-DP(Average) performs the worst among the others. As shown in Fig. 2, the existence of large load usage within certain time interval would dominate the average results which in turn would influence the load consumption pattern for forecasting. LSTM-DP(Random Forest) improves the performance comparing to LSTM-DP(Average) because of the working principle of the random forest algorithm. Under this condition, the filled data could be better correlated with the corresponding values for load forecasting. However, the performance is not as good as interpolation applied in LSTM-DP, since the interpolation by using the previous and next values could better capture the load consumption. On the other hand, LSTM-DP(KNN) generally performs better than LSTM-DP(Random Forest) and LSTM-DP(Average), because it fills the missing values by finding the most similar samples to assist estimation.

Moreover, as shown in Table 2, LSTM-DP also obtains the minimum prediction error in addition to the smallest MAPE. More specifically, approximately 80% improvement is achieved by the interpolation in LSTM-DP comparing to other filling methods. The missing points are interpolated by the nearest two points in LSTM-DP so that the load consumption pattern can be better reconstructed for prediction, since the load usage is normally time dependent due to human behavior.

2) THE IMPACT OF FILTER FOR DATA SMOOTHING AND DE-NOISING

In this experiment, we compare the Savitzky Golay filter in LSTM-DP with other filters for data smoothing aiming to investigate the gains using various filters for data smoothing. Specifically, Kalman and moving average filter replace the Savitzky Galoy filter in LSTM-DP, and thus they are denoted as LSTM-DP(Kalman Filter) and LSTM-DP(Moving Average Filter), respectively.

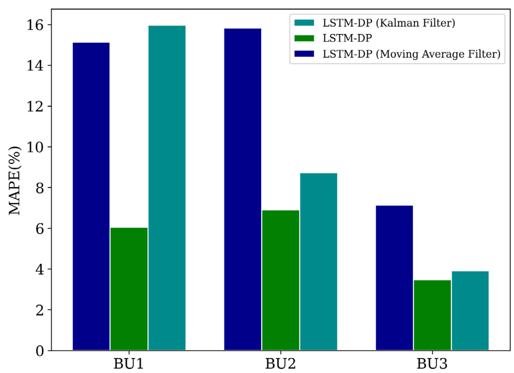

FIGURE 9. MAPE of various filters.

TABLE 3. Peak error of various filters.

	Peak Error (W)
LSTM-DP	12593.375
LSTM-DP (Kalman Filter)	14542.529
LSTM-DP (Moving Average Filter)	17458.440

Based on Fig. 9, LSTM-DP using Savitzky Golay filter for smoothing and de-noising outperforms the others. It is because Savitzky Golay filter tends to increase the precision of the load forecasting by better removing the spike and noise from the dataset. LSTM-DP(Kalman Filter) uses Kalman filter to estimate the load consumption and noise, but the performance is lower than LSTM-DP since Kalman filter is not specifically designed for recovering the spike or noise values of the data. LSTM-DP(Moving Average Filter) using the moving average filter performs worse than the others. Moving average algorithm makes a mean value of each window containing just several data, and thus it can not remove the noise efficiently.

Moreover, Table 3 shows LSTM-DP using Savitzky Golay filter introduces the least peak error among these three filters. LSTM-DP(Kalman Filter) approximately improves the peak error by 3000 (W) comparing to LSTM-DP(Moving Average Filter), but its peak error is higher than LSTM-DP by 2000 (W).

3) THE IMPACT OF MACHINE LEARNING MODEL FOR PREDICTION

In this experiment, we evaluate the performance of the load forecasting model in other deep learning model architectures. We pick other deep learning models and hybrid methods from the previous studies that show good performance. Specifically, 2CNN+LSTM [1], 3CNN+LSTM [2] and multi-layer bidirectional LSTM (MBLSTM) [21] are applied for prediction along with our proposed DP strategy, and thus we denoted them as 2CNN+LSTM-DP, 3CNN+3LSTM-DP and MBLSTM-DP, respectively. It is worth to noticed that 2CNN+LSTM and 3CNN+LSTM are hybrid model approaches while MBLSTM is multi-layer bidirectional LSTM.

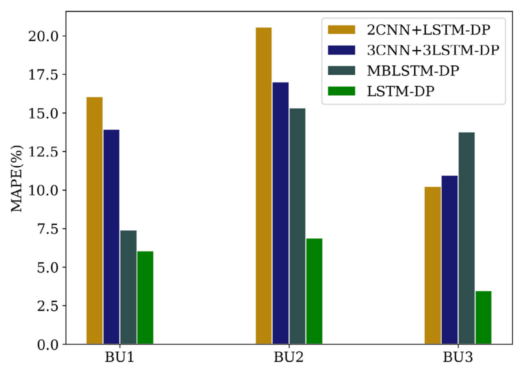

FIGURE 10. MAPE of various machine learning models.

TABLE 4. Peak error of various machine learning models.

	Peak Error (W)
LSTM-DP	12593.375
3CNN+3LSTM-DP	33013.38
2CNN+LSTM-DP	27611.41
MBLSTM-DP	24197.476

According to Fig. 10, LSTM-DP generally outperforms the others. In LSTM-DP, the DP strategy carefully selects appropriate dataset for the proposed LSTM strategy which effectively improves the load forecasting performance. The hybrid models, 3CMM+3LSTM and 2CNN+LSTM used in 3CMM+3LSTM-DP and 2CNN+LSTM-DP, obtain higher MAPEs comparing to LSTM-DP. In such hybrid models, CNN is generally used to extract and facilitate hidden features of the load sequences to provide accurate load forecasting which has been managed by DP strategy. Therefore, using a hybrid model with CNN and LSTM to replace LSTM for load forecasting may not further improve the MAPE. On the contrary, the extra processing might introduce more errors which in turn degrades prediction performance.

On the other hand, MBLSTM-DP using multi-layer bidirectional LSTM for load forecasting performs better than the hybrid model approaches. It is because the feedback mechanism helps improve the unidirectionality of traditional LSTM network prediction. However, using bidirectional will run the inputs in two ways, one from past to future and the other from future to past. Under this circumstance, MAPE increases due to overfitting. Although tuning the hyperparameters could improve the performance, it is out the scope of this paper.

In terms of peak error shown in Table 4, hybrid models (3CNN+3LSTM-DP and 2CNN+LSTM-DP) experience large peak error than rest models. More specifically, up to 60% peak error can be reduced by LSTM-DP comparing to hybrid models.

4) THE IMPACT OF LOOKBACK CONFIGURATION IN LSTM

In this experiment, we apply various lookbacks to the LSTM architecture of LSTM-DP for performance evaluation. We obtain the best performance when 9 lookbacks in LSTM is used. Thus, we select 2, 4, 6 and 10 lookbacks denoted as LSTM(2 Lookback)-DP, LSTM(4 Lookback)-DP, LSTM(6 Lookback)-DP, LSTM(10 Lookback)-DP, respectively for performance comparison.

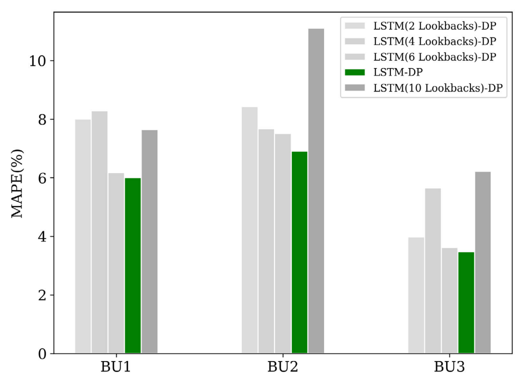

FIGURE 11. MAPE of various number of lookbacks in LSTM.

TABLE 5. Peak error of various number of lookbacks in LSTM.

	Peak Error (W)
LSTM(6 Lookbacks)-DP	12426.710
LSTM-DP	12593.375
LSTM(10 Lookbacks)-DP	34287.019

As illustrated in Fig. 11, some improvements can be found by adding the number of lookback in LSTM. In this case, LSTM-DP which uses 9 lookbacks outperforms the others.

More specifically, the number of lookbacks impacts the load forecasting results. For example, LSTM(2 Loopbacks)-DP uses 2 lookbacks which indicates only 2 hours historical information from the current time is used for load forecasting. Thus, some references to predict the next hour load usage may miss which reduce the prediction accuracy. Generally, when the number of lookbacks is less than 10, the load forecasting accuracy increases with the increasing of the number of lookbacks, especially for BU2. In addition, the performance of LSTM(6 Lookbacks)-DP using 6 lookbacks is better than LSTM(4 Loopbacks)-DP using 4 lookbacks for all buildings. However, the performance of LSTM(10 Lookbacks)-DP using 10 lookbacks decreases comparing to LSTM-DP using 9 lookbacks for all buildings. It is because overfitting/overtrained might occur when 10 lookbacks is applied which is similar to the conclusion in [22]. Moreover, the effect of overfitting/overtrained also can be found in Table 5 in which LSTM(10 Loopbacks)-DP has about 3 times more peak error than others.

V. CONCLUSION

In this paper, we firstly elaborate the challenges and characteristic of the load consumption dataset, and then propose the hourly load forecasting framework combining Data Preparation and LSTM, namely LSTM-DP, to predict the hourly load consumption of each household. The DP strategy of LSTM-DP is designed to handle the problems of the data caused by error or noise and select appropriate features while the LSTM strategy is in charge of load forecasting. Specifically, through the DP strategy, an input dataset containing several features is created after anomaly data is reconstructed by interpolation and smoothing. In this way, the LSTM strategy can utilize the prepared datset to better capture the load consumption pattern on each household for hourly load forecasting. The evaluation results show the proposed LSTM-DP outperforms the others in terms of various viewpoints. In the future, hyper-parameter tuning methodologies can be developed to enhance forecasting accuracy. The minor effects that influence the load consumption include the number of people, seasonal, and period of holiday can also be investigated in further.

REFERENCES

[1] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid CNN-LSTM model for short-term individual household load forecasting," *IEEE Access*, vol. 8, pp. 180544–180557, 2020.

[2] F. U. M. Ullah, A. Ullah, I. U. Haq, S. Rho, and S. W. Baik, "Short-term prediction of residential power energy consumption via CNN and multi-layer bi-directional LSTM networks," *IEEE Access*, vol. 8, pp. 123369–123380, 2020.

[3] 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.

[4] N. Q. Uy and V. H. Nam, "A comparison of AMQP and MQTT protocols for Internet of Things," in *Proc. 6th NAFOSTED Conf. Inf. Comput. Sci. (NICS)*, Dec. 2019, pp. 292–297.

[5] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, May 2019.

[6] P. Kowalski and R. Smyk, "Review and comparison of smoothing algorithms for one-dimensional data noise reduction," in *Proc. Int. Interdiscipl. PhD Workshop (IIPhDW)*, May 2018, pp. 277–281.

[7] W. Deng, Y. Guo, J. Liu, Y. Li, D. Liu, and L. Zhu, "A missing power data filling method based on improved random forest algorithm," *Chin. J. Electr. Eng.*, vol. 5, no. 4, pp. 33–39, Dec. 2019.

[8] B. Al-Helali, Q. Chen, B. Xue, and M. Zhang, "GP-based feature selection and weighted KNN-based instance selection for symbolic regression with incomplete data," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2020, pp. 905–912.

[9] K. Liang, F. Liu, and Y. Zhang, "Household power consumption prediction method based on selective ensemble learning," *IEEE Access*, vol. 8, pp. 95657–95666, 2020.

[10] R. K. Agrawal, F. Muchahary, and M. M. Tripathi, "Long term load forecasting with hourly predictions based on long-short-term-memory networks," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, Feb. 2018, pp. 1–6.

[11] S. Chakraborty, J. Banik, S. Addhya, and D. Chatterjee, "Study of dependency on number of LSTM units for character based text generation models," in *Proc. Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA)*, Mar. 2020, pp. 1–5.

[12] S. H. Rafi, Nahid-Al-Masood, S. R. Deeba, and E. Hossain, "A short-term load forecasting method using integrated CNN and LSTM network," *IEEE Access*, vol. 9, pp. 32436–32448, 2021.

[13] H. Yiling and H. Shaofeng, "A short-term load forecasting model based on improved random forest algorithm," in *Proc. 7th Int. Forum Electr. Eng. Autom. (IFEEA)*, Sep. 2020, pp. 928–931.

[14] M. Ali, Z. A. Khan, S. Mujeeb, S. Abbas, and N. Javaid, "Short-term electricity price and load forecasting using enhanced support vector machine and *K*-nearest neighbor," in *Proc. 6th HCT Inf. Technol. Trends (ITT)*, Nov. 2019, pp. 79–83.

[15] N. Amjady, "Short-term hourly load forecasting using time-series modeling with peak load estimation capability," *IEEE Trans. Power Syst.*, vol. 16, no. 3, pp. 498–505, Aug. 2001.

[16] A. D. Papalexopoulos and T. C. Hesterberg, "A regression-based approach to short-term system load forecasting," *IEEE Trans. Power Syst.*, vol. 5, no. 4, pp. 1535–1547, Nov. 1990.

[17] C. Guan, P. B. Luh, L. D. Michel, and Z. Chi, "Hybrid Kalman filters for very short-term load forecasting and prediction interval estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3806–3817, Nov. 2013.

[18] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.

[19] C. S. Chen, M. S. Kang, J. C. Hwang, and C. W. Huang, "Temperature effect to distribution system load profiles and feeder losses," *IEEE Trans. Power Syst.*, vol. 16, no. 4, pp. 916–921, Nov. 2001.

[20] B. Sun, L. Ma, W. Cheng, W. Wen, P. Goswami, and G. Bai, "An improved k-nearest neighbours method for traffic time series imputation," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 7346–7351.

[21] C. Cai, Y. Tao, Q. Ren, and G. Hu, "Short-term load forecasting based on MB-LSTM neural network," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 5402–5406.

[22] K. A. Koparanov, K. K. Georgiev, and V. A. Shterev, "Lookback period, epochs and hidden states effect on time series prediction using a LSTM based neural network," in *Proc. 28th Nat. Conf. Int. Participation (TELECOM)*, Oct. 2020, pp. 61–64.

DERNI AGENG received the B.Eng. degree from the Electrical Engineering Department, University of Indonesia, Jakarta, Indonesia, in 2019, 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.

CHIN-YA HUANG (Member, IEEE) received the B.S. degree in electrical engineering from the National Central University, Taiwan, in 2004, the M.S. degree in communication engineering from the National Chiao-Tung University, Taiwan, in 2006, and the M.S. degrees in both electrical & computer engineering and computer science and the Ph.D. degree in electrical & computer engineering from the University of Wisconsin–Madison, USA, in 2008, 2010, and 2012, respectively. She also worked as an Assistant Professor with National Central University, from 2015 to 2018. She worked as a member of technical staff in Optimum Semiconductor Tech. Inc., NY, USA, from 2013 to 2015. In 2012 and 2013, she was a Senior Software Engineer at Qualcomm Technology Inc., San Diego, CA, USA. She also worked as an Assistant Research Fellow at Information and Communication Technology Laboratory, National Chiao-Tung University, in 2013. 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.

RAY-GUANG CHENG (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in communication engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1991, 1993, and 1996, respectively. From 1997 to 2000, he was a Researcher and the Project Leader with the Computer and Communication Laboratories, Advanced Technology Center, Industrial Technology Research Institute (ITRI), Hsinchu. He led the 3G Protocol Project and his team was named Top Research Team of the Year by ITRI, in 2000. From 2000 to 2003, he was the Senior Manager with the Research and Development Division, BenQ Mobile System Inc., Hsinchu. He is currently a Distinguished Professor with the Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan. His current research interests include massive machine-type communications and open radio access networks (O-RAN). He is a member of Phi Tau Phi Scholastic Honor Society. He was a recipient of the IEEE Wireless Communication Professional Certification. He has been holding IEEE Wireless Communication Professional (WCP) certification, since 2013; holds 18 U.S. patents; and has published more 30 IEEE/3GPP standard contributions. He led the 3G Protocol project and his team was named Top Research Team of the Year by ITRI, Taiwan, in 2000. He received the Best Industrial-based Paper Award from the Ministry of Education, Taiwan, in 1998; Advanced Technologies Award from the Ministry of Economic Affairs, Taiwan, in 2000; Y. Z. Hsu Scientific Paper Award, Taiwan, in 2016; and ‘‘World Class Professor Program’’ from the Ministry of Research, Technology and Higher Education, Indonesia, in 2018.