

CHAPTER III

Day-Ahead Load Forecasting Using Supervised Deep Learning Techniques with Day-Aware Data Arrangement

3.1 Introduction

In the context of electrical load forecasting, choosing the right method for the highly complex data is an important issue, especially for electrical load data that has a non-linear relationship and changes significantly over time, such as weekdays, holidays, or school openings and closings, which all affect energy use behavior at the institutional or large organization level, such as universities.

Current research trends have begun to focus on the use of deep learning models, especially the RNN family of models such as LSTM, GRU, Bi-LSTM, and Bi-GRU, which are designed to efficiently handle time series data. The strength of these models is their ability to learn patterns of past data to predict the future. However, these models often have limitations in terms of transparency in explaining their results, which raises questions about their practical reliability. To solve this problem, research focuses on the process of selecting appropriate features, adding contextual information such as time, weather, calendar, and holidays, and using model testing and evaluation techniques that are consistent with time series data, such as data partitioning by the Expanding Window method, which does not leak future data to model training. In addition, the use of the SHAP technique allows for a clearer explanation of model decisions, both at the individual data point level and in general.

This chapter aims to describe the research process used to build and test the electrical load forecasting model, which will serve as an important basis for the comparative performance analysis in the next chapter. Parts of this thesis have been published in the proceedings of the 2025 13th International Electrical Engineering Congress (IEECON2025), is shown in the Appendix D.

3.2 Data Collection and Preparation

The datasets collected and used in the forecasting include electric load data from utility meters, which shows historical data from the Automatic Meter Reading (AMR) system maintained by the Provincial Electricity Authority (PEA), which records electricity usage according to load demand at the university, as well as solar energy production data, which records energy production from meters installed in the area with solar panels at Suranaree University of Technology in Thailand. Data regarding compensatory holiday long weekends is gathered from university announcements and the academic calendar. A reliable public website with an API for direct data extraction is the source of historical condition data. Every 15 minutes for a year, from January to December 2023, these statistics are collected and compiled. The electrical load forecasting utilizes 35,040 datasets in total. In the data cleaning process, which is an important process in preparing various data before using it for forecasting in the model, the following methods are used: Handling missing values (Handling Missing Values) will delete the missing data range or the data range with abnormalities and add new data from the method of calculating the average and transforming the data format to be appropriate (Data Formatting) using Timestamp to help create the Time Factor and Social Factors features, helping the model to learn the relationship of the data better and be more efficient in forecasting. Since the electrical load in the university is a high PV penetration load with a proportion of solar energy produced and sent to the total power system, the electric load data from utility meters or net load obtained is data that is mixed with solar energy production data or PV generation, making it possible to see electricity usage according to the load demand that is unrealistically low. Therefore, PV Generation is added back to obtain pure load or electrical load data that can be used to assess the actual electricity demand in forecasting.

The data is divided by taking the entire cleaned dataset for the year and splitting it into three sets including the training set, which uses 80% of the total data; the validation set, which uses 20% of the data for parameter tuning; and the test set, which consists of a separate portion of data that the model has never seen before, reserved for final evaluation.

3.3 Feature Extraction

Assigning features to existing data is very important in forecasting electrical load. Because features are variables that will be fed into the model to help the model learn relationships and generate accurate forecast results. Features used in forecasting this type of data are a collection of parameters that are significant to load forecasting and are created by considering the context of time and day. Creating features from data can be divided into two types as follows: The first type is creating features that are Day-Aware Data Arrangement, which is a method of arranging data that takes into account the context of time and day to allow the model to learn the characteristics of the data on each day or specific time period. It consists of Hour of Day, which is a value that specifies the hours in one year; Quarter-Hourly Interval, which is a value that specifies the minutes every 15 minutes in each hour; Weekdays and Weekend, which are values used to separate data between weekdays and weekends; Daytime and Nighttime, which are values used to separate data between day and night; Compensatory Holiday, which is a value that specifies whether that day is a holiday or not, Long Weekend, which is a value that specifies that period is a special holiday or festival. The second type is a feature that is weather data, consisting of ambient temperature data and relative humidity data. The number of data sets used to create and divide the feature data sets will be the same as the data used in forecasting electrical load.

3.4 Feature Engineering

Feature engineering is an important step in developing a load prediction model, playing a role in creating appropriate variables that accurately represent temporal behavior, which will improve both the accuracy and explainability of the model. From the collection and review of research, feature engineering techniques will be used. In this research, the techniques will be divided into two main groups: feature generation and external features. The first group is featuring generation, which is the creation of new features from existing temporal data and the transformation of temporal data that may not be easily understood by the model into logical information that is meaningful according to the context. as described as follows.

3.4.1 Feature Transformation / Binarization

Variable transformation and binarization are the processes of transforming the source data into a format that is easy to interpret or respond to the model, as follows: Create a feature that divides Monday to Friday to represent school days and Saturday and Sunday to represent weekends, with the numbers 0 and 1, so that the model clearly understands the difference between Monday and Sunday. Create a feature that divides day and nighttime with the numbers 0 and 1, such as 8:00 - 17:00, which is daytime and is 1, and the rest is nighttime, which is 0. Dividing day and nighttime into categories that are easier to understand. Creating a feature that does not just divide normal holidays and school days but can be adjusted according to special events, such as compensatory holidays or school days, helps the model better understand the behavior at each time period.

3.4.2 Time-based Feature Engineering / Ordinal Time Block

Segmenting data into meaningful timelines with 1 year of data every 15 minutes creates a feature that allows the model to learn the day and time, such as day 1, where 00:00 - 00:45 will be set as 0, time 01:00 - 01:45 will be set as 1, time 02:00 - 02:45 will be set as 2, and so on until 23:00 - 23:45 will be set as 23, considered day 1, and day 2 is set as the same. The method and advantages of this feature are as follows:

Ordinal Encoding: Use sequential numbers (0, 1, 2, ... 23) to represent time of day, which helps the model to recognize the continuity of the data.

Time Segmentation: Divide time periods into groups with different usage patterns, such as morning (Morning), afternoon (Afternoon), evening (Evening), and night (Night).

Pattern Recognition: The model can learn the trends of each time period, such as energy consumption may be higher in the morning or lower at night.

Use this feature to forecast energy usage by letting the model understand that different days can have different energy usage patterns, helping the model better understand the context of that time period.

3.4.3 Time Component Extraction

Decomposing time components into separate features creates a new feature that describes each sub-hour: 00.00 is 0, 00.15 is 15, 00.30 is 30, and 00.45 is 45. After 01.00, it is 0; 01.15 is 15; 01.30 is 30; and 01.45 is 45. After 02.00, it is 0. This continues so that the model can capture the relationship between time periods and different load behaviors. The model can understand the recurring behaviors during each sub-hour, and this helps to understand the trends in energy usage over time. For example, every hour from 0-45 may have a similar energy usage pattern.

3.4.4 Time Series Features / Lag Features

Building features from previous values (lag) is an important basic technique in the time series data analysis. It uses historical load values to build features such as load in the same period of the previous day (Load $t-1d$). This method allows the model to learn trends and seasonality efficiently.

In the second group there are external (exogenous) features, external variables that do not arise from the process of the target variable itself but have an impact or relationship with the forecast of the target variable and can be used as additional input to the forecasting model to increase its accuracy, consisting of the following techniques:

3.4.5 Ambient Temperature

Ambient temperature is an external factor that directly affects energy use behavior, such as air conditioning or heating use. This information is often used in conjunction with load data to help models understand the physical relationship between weather conditions and energy demand.

3.4.6 Dew Point

The Dew point is a measure of humidity in the air, which affects temperature perception and the use of cooling or heating equipment, especially in regions with high humidity. Incorporating dew point values into models allows for a more accurate capture of energy consumption behavior.

3.5 Advanced RNN Architectures

The design of the model architecture is the heart of developing the electrical load forecasting model. Due to the nature of electrical load data being a time series with significant variations over time and period, selecting a model that can accurately capture the characteristics of the data sequence is essential. In this section, we have selected the architecture of the Recurrent Neural Networks (RNNs) model that has been improved to better remember long-term data, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), as well as the Bidirectional version (Bi-LSTM and Bi-GRU), which can see the context from both the past and the future in the time sequence, allowing a deeper understanding of the relationships in the data.

3.5.1 LSTM Architecture

The Long Short-Term Memory (LSTM) model is a structure of Recurrent Neural Network (RNN) designed to solve problems and manipulate time-series data. It consists of a Forget Gate that determines which data should be forgotten from the original cell state. Let $\mathbf{x}_t$ be the input feature vector at time t , $\sigma(\cdot)$ be the sigmoid activation function, $\mathbf{h}_{t-1}$ be the hidden state from time $t - 1$, and $\mathbf{b}_f$ be the bias vector of the forget gate, $\mathbf{W}_f$ be the learnable weight matrix of the forget gate. The forget-gate output is computed as shown in Equations (3.1).

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad (3.1)$$

where $\mathbf{f}_t \in [0,1]$ is close to 1, indicating retention, while values close to 0 indicate erasure. This helps determine whether data should be retained. The next component, the Input Gate ($\mathbf{i}_t$) and the Candidate Cell State, decides which new data to add to the cell state, as shown in Equations (3.2) and Equations (3.3).

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad (3.2)$$

$$\tilde{C}_t = \sigma(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3.3)$$

where $\tilde{C}_t$ controls which new data should be imported, and $\tilde{C}_t$ represents the new data processed (candidate memory). The next component, the Update Cell State, combines the Forget Gate and the Input Gate to create a new cell state, as shown in Equations (3.4).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.4)$$

where the forgotten data is multiplied by f_t , and the new, important data is multiplied by i_t and added together. The final section is the Output Gate (o_t) and Hidden State. These define the data to output as hidden states for future forecasting, as shown in Equations (3.5) and Equations (3.6).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.5)$$

$$h_t = o_t * \tanh(C_t) \quad (3.6)$$

where h_t is the hidden state, used to predict the electrical load at the next point.

LSTM is a special kind of RNN that can identify long-term dependencies. LSTMs are purposefully designed to avoid the problem of long-term reliance. They essentially behave in such a way that their default state is long-term memory retention. For time series forecasting, they are hence the perfect parallel. Fig. 3.1 describes how LSTM cells are represented graphically.

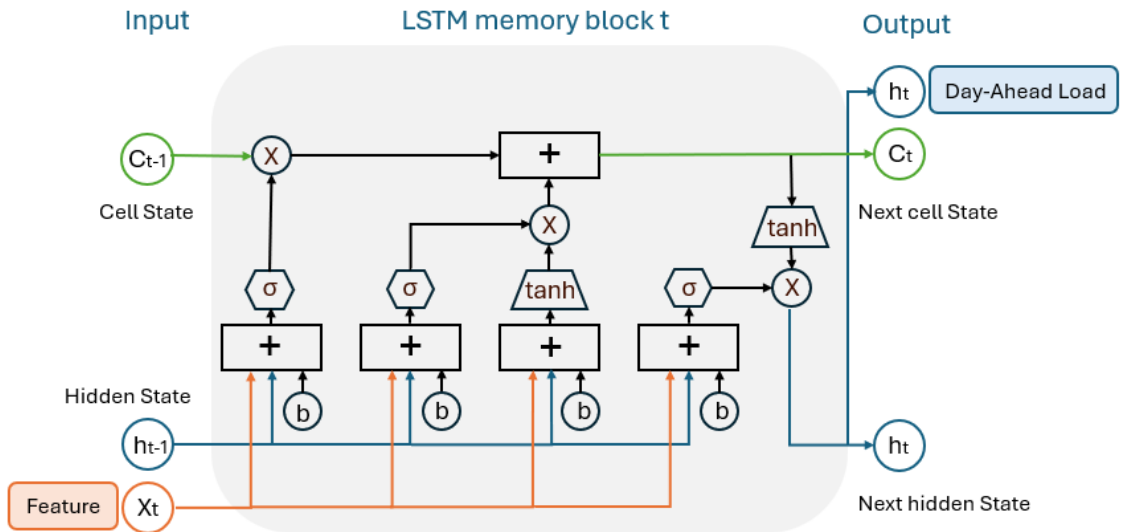


Figure 3.1 LSTM architecture

3.5.2 Bi-LSTM Architecture

The Bidirectional Long Short-Term Memory (Bi-LSTM) model is a structure that is developed from LSTM to increase the ability to learn sequences of data that have relationships in both the past (past dependencies) and the future (future dependencies) simultaneously. It consists of two LSTM layers that work together: a forward LSTM that processes data from the past to the present and a backward LSTM that processes data from the future back to the present at any time t . The results of both directions are combined into a single hidden state as shown in Equations (3.7).

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (3.7)$$

where $\vec{h}_t$ is the hidden state from the Forward LSTM, $\overleftarrow{h}_t$ is the hidden state from the Backward LSTM, and h_t is the combination of the two results.

Bi-LSTM is a sequence processing model that is made up of two LSTMs, one of which accepts input in a forward direction and the other in a backward manner. Bi-LSTMs give the network access to additional information, which improves the context of the algorithm. The primary advantage is that each input sequence piece includes both current and historical data. Therefore, Bi-LSTM can produce a more pertinent outcome by merging LSTM layers from both directions. The configuration of LSTM layers in Bi-LSTM is shown in Fig. 3.2.

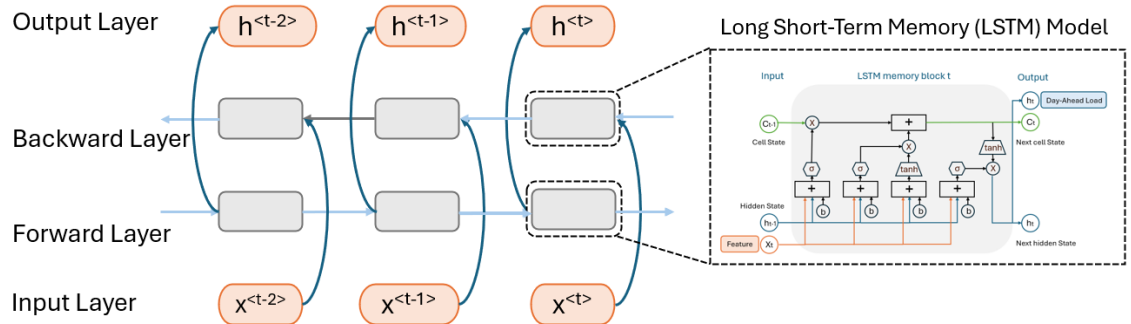


Figure 3.2 Bi-LSTM architecture

3.5.3 GRU Architecture

GRU resembles LSTM, which regulates data flow using gates. But with a simpler design, GRU provides an improvement over LSTM. GRU just has a hidden shape, not a distinct cell state like LSTM. The GRU has unrestricted access to all concealed states. Compared to LSTM units, GRUs are easier to train and do not require a memory unit to regulate the data flow. As illustrated in Fig. 3.3, GRU may train more quickly or employ less data to generalize since it has fewer parameters.

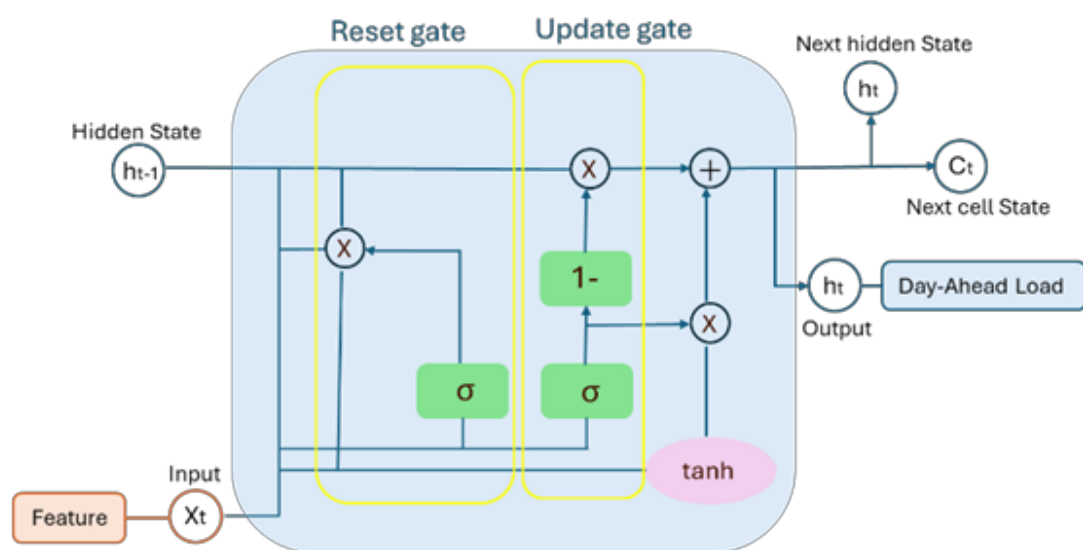


Figure 3.3 GRU architecture

3.5.4 Bi-GRU Architecture

Bidirectional GRU - A bidirectional GRU, often known as a Bi-GRU, a bidirectional GRU is designed to analyze univariate time series data. This enables the temporal dependencies found in the electrical load data to be accommodated. This integration becomes essential for precise forecasting by utilizing data from both previous and subsequent time periods. As illustrated in Fig. 3.4, this functionality plays a substantial role in improving forecasting precision.

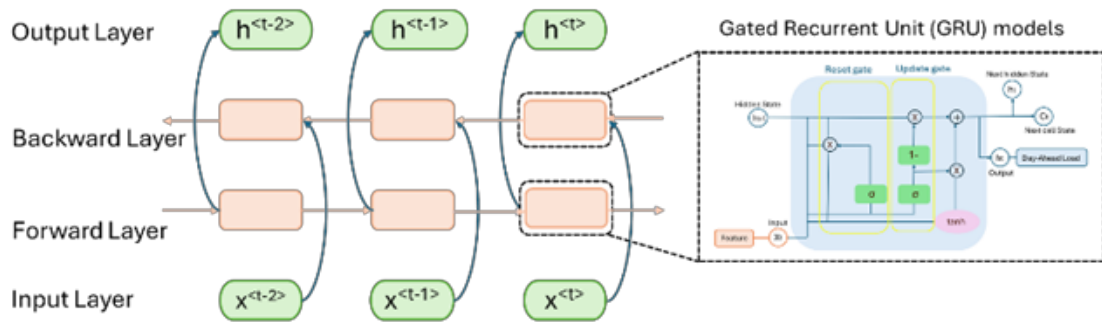


Figure 3.4 Bi-GRU architecture.

3.6 Workflow of the Proposed Method

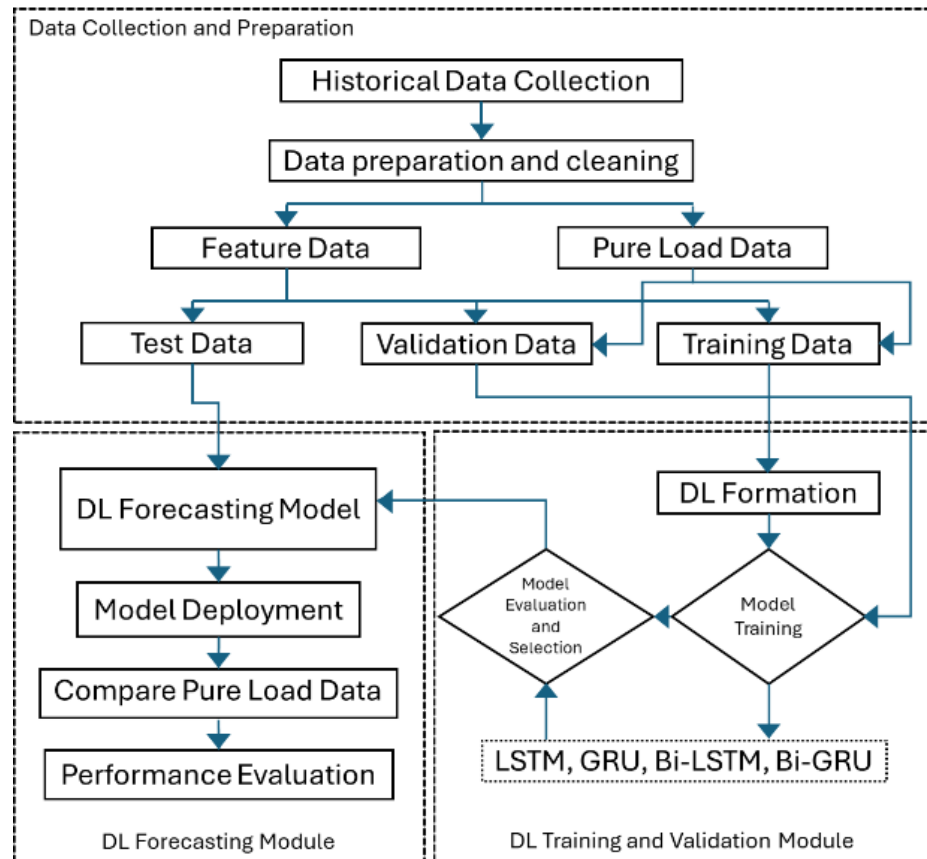


Figure 3.5 The Proposed Load Forecasting Workflow

The electrical load forecasting, including finding the model with the appropriate accuracy for the university's load data, can be done as follows. Start by collecting all relevant historical data, then clean the data with necessary techniques to facilitate

model learning. The data is divided into three groups: the training data group receives feature data as input and pure load data for the model to learn the relationship, then performs the deep learning formation step, which is to design the structure and select the function for the various models and train the model. The models used for training include LSTM, GRU, Bi-LSTM, and Bi-GRU, and they output the load forecast. The validation data group receives the same input as the first group, receiving various models for forecasting. When the output is the load forecast, it will help check the accuracy of model training and prevent overfitting in the model training process. It also helps evaluate the performance of the model during model training of the predicted values and actual load values. Then select the best-performing model as the deep learning forecasting model. The test data group receives features as input, inputs data into the selected model, and outputs daily load forecasts and obtains the developed model. By comparing the predicted values with the actual load values available, the performance of the tested model can be evaluated, as can be seen in Fig. 3.5.

3.7 Performance Factor

To examine the performance of the proposed forecasting model through evaluation metrics including percentage of MAPE, MAE and RMSE, as shown in Table 3.1. Let Y_i denote the actual load at time i , and $\hat{Y}_i$ denote the corresponding predicted load.

Table 3.1 Performance Factor

Name of Criteria	Formula
Mean Average Percentage Error (MAPE)	$\frac{1}{n} \sum_{i=1}^n \frac{(Y_i - \hat{Y}_i)}{Y_i}$
Mean Absolute Error (MAE)	$\frac{\sum_{i=1}^n Y_i - \hat{Y}_i }{n}$
Root Mean Square Error (RMSE)	$\sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$

3.8 Simulation Results

This section presents experimental results and performance analysis of the electrical load forecasting models developed in Chapter 3. It compares the forecasting results of LSTM, GRU, Bi-LSTM, and Bi-GRU models on pre-prepared test datasets, using RMSE, MAE, and MAPE performance indicators to measure model accuracy across multiple dimensions. The analysis is divided into three main parts: evaluating model learning and accuracy during the validation and training phases and comparing forecasting results between weekdays and weekends. This chapter aims to answer the key question of which model is best suited for electrical load forecasting in the context of real-world data in terms of accuracy, consistency, and operational capability under various scenarios, and to evaluate the performance of the best model on different datasets in terms of time and context.

3.8.1 Experimental setup

Python 3 software was employed to simulate the Suranaree University of Technology's daily electrical load forecast in order to create a model suitable for the university's electrical loads and evaluate and contrast efficiency using evaluation metrics including MAPE, MAE, and RMSE. The data selected for use was every 15 minutes throughout 2023 and contained, in addition to pure load and feature or electric load data used for forecasting, The models made use of to simulate the results were LSTM, GRU, Bi-LSTM, and Bi-GRU. To get this data ready for entry into the model, it was separated into three sections. This section will show the outcomes of the simulation and different model settings.

In preparing data for power load forecasting, this research addresses missing data using filler methods that correspond to the temporal structure of the data. However, data filler can affect model performance, potentially reducing power load fluctuations and causing bias in the learning process, especially when the filled data is large or concentrated over the same time period. Furthermore, filler values can impact the generation of temporal features, such as lag features, which rely on data continuity. To limit these effects, this research controls the proportion of filled data to

a low level and selects filler methods appropriate to the data characteristics. However, future research should develop more advanced methods for handling missing data or perform model sensitivity analysis to filler to improve forecasting reliability.

The selection of hyperparameters was carried out using a trial-and-error method within a systematic evaluation framework. It began by referencing commonly used values from relevant research, then adjusting parameters individually or in sets to study their impact on the model's learning ability and forecasting errors. This method allowed for observation of the model's behavior under different settings and clearly identified trends in values affecting model performance. The final hyperparameter selected is the set of values that yields the best accuracy results, consisting of the following: A small data set of size four is used to improve the diversity of learning, using 50 epochs to give the model enough time to learn from the training data, and the model structure is set to have 100 units as the number of units in the hidden layer to train different DL models. Two layers of each model are used, a dropout rate of 0.2 is used to prevent overfitting, and Adam Optimizer is used to adjust the model weights, and a learning rate of 0.001 is used as the learning rate for Adam Optimizer, as shown in Table 3.2.

Table 3.2 Model parameter settings

Methods	LSTM	GRU	Bi-LSTM	Bi-GRU
Layer	2	2	2	2
Units	100	100	100	100
Dropout rate	0.2	0.2	0.2	0.2
Learning rate	0.001	0.001	0.001	0.001
Batch Size	4	4	4	4
Epochs	50	50	50	50
Optimizer	Adam	Adam	Adam	Adam

3.8.2 Model performance in validation and model training

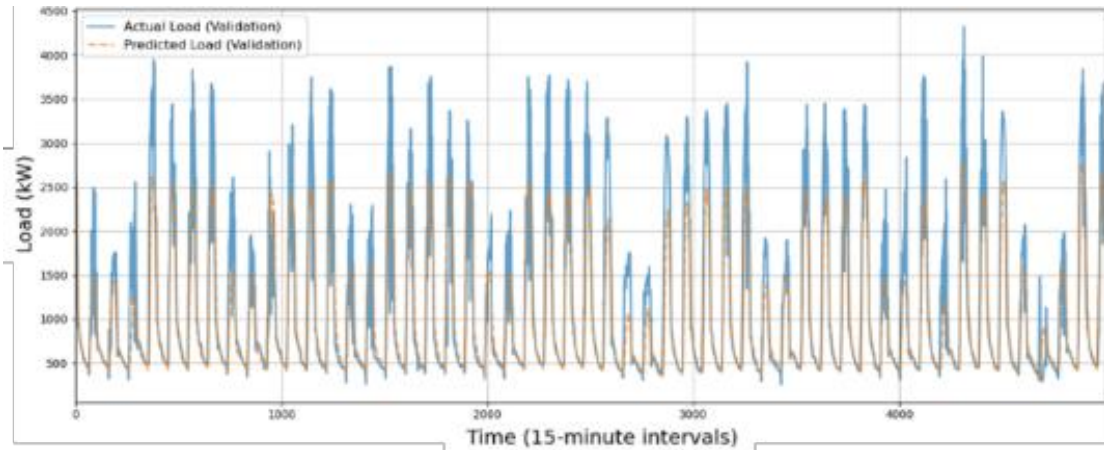
To evaluate the performance of the developed daily electrical load forecasting model in this research, the RNN family models, namely LSTM, GRU, Bi-LSTM, and Bi-GRU, were tested on the real load dataset to compare the model accuracy in the validation phase. In addition, the model training time was recorded to compare the time efficiency between different architectures. The results are shown in Table 3.3, reflecting the forecasting capability of each model under the same context and supporting the selection of the most suitable model in terms of accuracy and performance.

Table 3.3 Evaluation results of model performance in validation metrics and model training time.

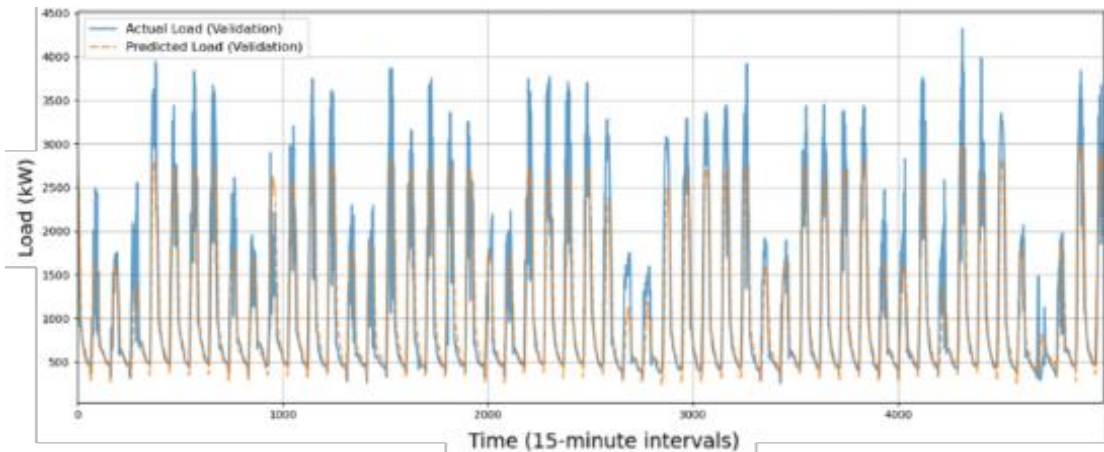
Methods	Validation Metrics			Model Training
	MAPE (%)	MAE (kW)	RMSE (kW)	Time (s)
LSTM	16.36	227.48	428.84	296.34
GRU	20.61	271.96	464.28	176.76
Bi-LSTM	16.86	241.22	439.58	589.14
Bi-GRU	17.71	252.18	453.94	318.95

The GRU model trained faster than the LSTM model, which took 296.34 seconds. Along with the evaluation of the model's performance in the validation set data, the model training time could indicate the device's usage of resources. The findings of the model's electrical load forecasts will be validated using this data. With MAPE = 16.36%, MAE = 227.49 kW, and RMSE = 428.85 kW, the LSTM model performs best after the performance evaluation. The Bi-LSTM model, with MAPE = 16.86%, MAE = 241.22 kW, and RMSE = 439.5889 kW, is the experimentally efficient model. The two models' training errors are comparable to those of the GRU model (MAPE = 20.61%, MAE = 271.96 kW, and RMSE = 464.29 kW) and Bi-GRU (MAPE = 17.71%, MAE = 252.19 kW, and RMSE = 453.94 kW), according to the three metrics. According to the validation dataset, results demonstrate the extent to which the LSTM and Bi-LSTM models learn

and capture relationships. With its speed and predicting accuracy, LSTM offers potential for real-world use. Table 3.3 displays the values in the validation set, which includes the model training settings or the simulation results of the validation metrics' performance.

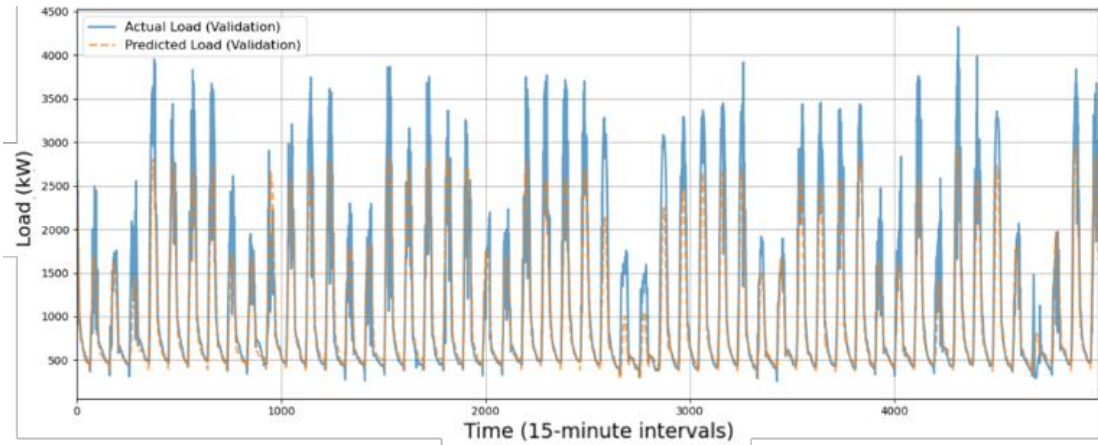


(a)

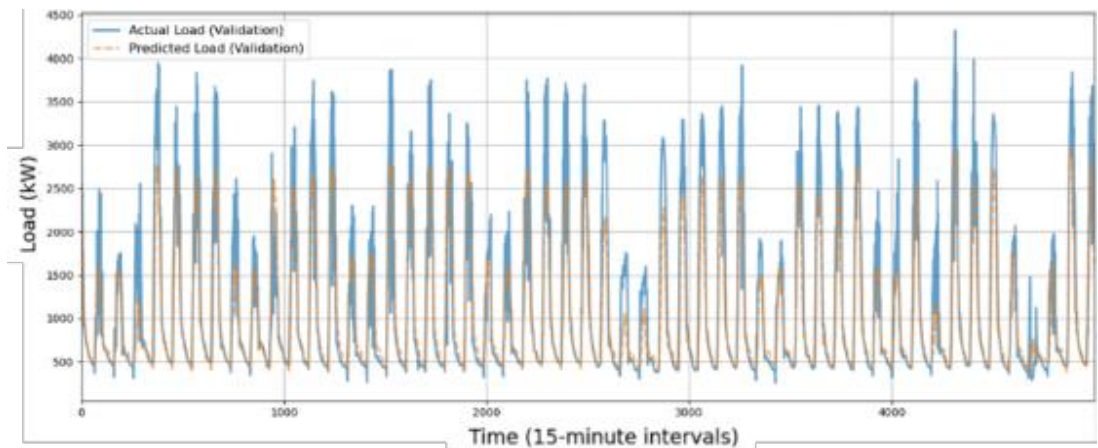


(b)

Figure 3.6 Electrical load forecasting on validation datasets of (a) LSTM, (b) GRU, (c) Bi-LSTM and (d) Bi-GRU models



(c)



(d)

Figure 3.6 Electrical load forecasting on validation datasets of (a) LSTM, (b) GRU, (c) Bi-LSTM and (d) Bi-GRU models

Fig. 3.6 presents the prediction results between the actual load and the model's predicted load on the validation data. It includes the predictions of the LSTM, GRU, Bi-LSTM, and Bi-GRU models, respectively. The results of the prediction align with the mentioned earlier metrics.

3.8.3 Performance of each model on weekday and weekend loads

After model training and validation, the model is tested in real use. The test set data is divided into two types: weekday load is the load from Monday to Friday, which has a higher total electricity demand than the holiday load. Additionally,

weekend load is the load on Saturday or Sunday with the largest variability in electricity demand. The following test days have been selected to evaluate the precision and capacity for forecasting each load model when the electrical load exhibits distinct characteristics: The weekday load is Wednesday, March 1, 2023. And the weekend load is Sunday, October 1, 2023. Forecasting the electrical load output for a day or 24 hours in the future is required, with data being displayed every 15 minutes. There will be a total of 96 data sets. The actual test date is simulated through each model. The results obtained from simulating the weekday load are shown on Fig. 3.7 and the results from simulating the holiday load are shown on Fig. 3.8. The forecast graph will appear.

The blue line represents the actual electrical load, or pure load, the yellow line, the green line, the red line, and the purple line. Shows the forecast of the LSTM, GRU, Bi-LSTM, and Bi-GRU models, respectively. In the weekdays load forecast, the LSTM model graph gives the forecast that is closest to the actual load graph. The graph is closely related to the actual load graph. The next order has the forecast that deviates from the actual load, which are the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively. However, other models can provide the forecast of the actual load trend. In the part of the weekend's load forecast, the LSTM model graph still gives the forecast that is closest to the actual load graph. However, there are still some load periods that do not give the forecast that is as close to the actual value as it should be because it is a day with a significant load fluctuation. Overall, the LSTM model still performs well. The models whose forecast graphs are closest to the Bi-LSTM model graph, the Bi-GRU model, and the GRU model, respectively.

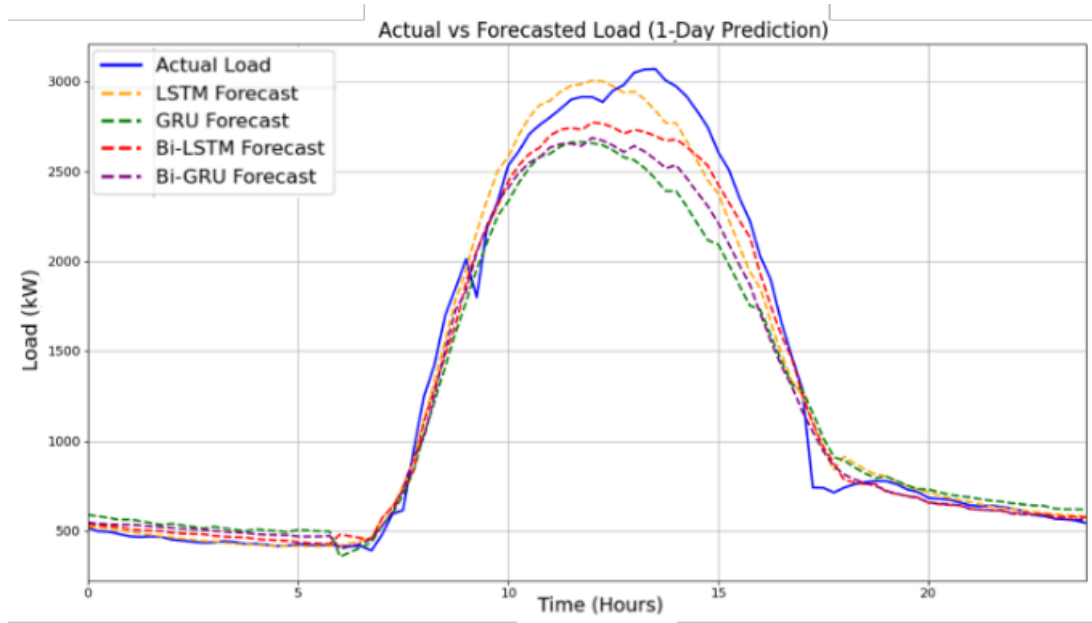


Figure 3.7 One-day-ahead electricity load forecast for each model on weekdays

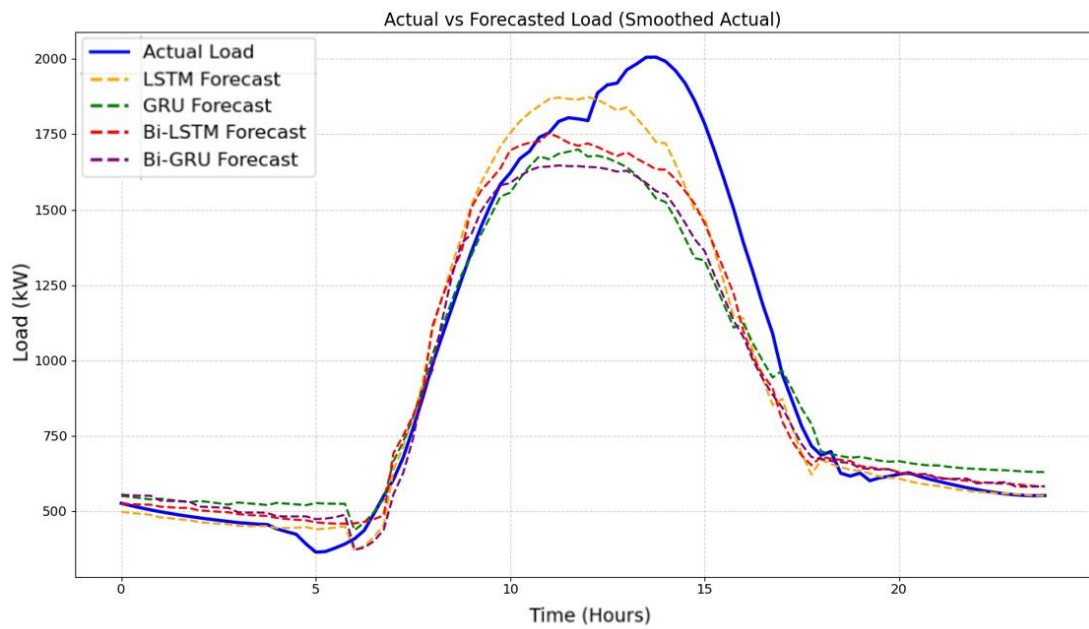


Figure 3.8 One-day-ahead electricity load forecast for each model on weekends

Table 3.4 Performance of each model on weekday load

Methods	Model Performance of Weekday Load		
	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	6.09	81.51	122.84
GRU	13.29	170.33	242.46
Bi-LSTM	7.38	91.38	131.30
Bi-GRU	9.88	135.16	196.28

Table 3.5 Performance of each model on weekend load

Methods	Model Performance of Weekend Load		
	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	9.62	115.12	214.53
GRU	15.51	157.18	254.17
Bi-LSTM	11.21	127.00	219.68
Bi-GRU	11.42	138.39	243.42

In addition, the performance of each model was evaluated through all three indicators or through evaluation metrics. The results of the weekday load evaluation are as follows: The LSTM model has a MAPE = 6.09%, MAE = 81.51 kW, and RMSE = 122.84 kW. The GRU model has MAPE = 13.29%, MAE = 170.33 kW, and RMSE = 242.46 kW. The Bi-LSTM model has a MAPE = 7.38%, MAE = 91.38 kW, and RMSE = 131.30 kW. And the Bi-GRU model has a MAPE = 9.88%, MAE = 135.16 kW, and RMSE = 196.28 kW. Shown in Table 3.4. The results of the holiday load evaluation are as follows: The LSTM model has a MAPE = 9.62%, MAE = 115.12 kW, and RMSE = 214.53 kW; the GRU model has a MAPE = 15.51%, MAE = 157.18 kW, and RMSE = 254.17 kW; the Bi-LSTM model has a MAPE = 11.21%, MAE = 127.00 kW, and RMSE = 219.68 kW; and the Bi-GRU model has a MAPE = 11.42%, MAE = 138.39 kW, and RMSE = 243.42 kW. Shown in Table 3.5. From the evaluation of the model performance, the results of the weekday load, the LSTM model shows the lowest forecast error when compared to the Bi-LSTM model, Bi-GRU model, and GRU model. For the results of holiday load,

the LSTM model still has the lowest error of all metrics when compared to other models.

However, it still has a higher error than the weekday load because it is a day with high volatility, especially in hours 13 to 18, which has the highest volatility, resulting in the highest forecast error during that period. Overall, the LSTM model still has the best power demand prediction and gives the lowest error for that day. The results show that the most efficient and accurate model for university load data is the LSTM model because the model can learn the patterns and relationships of long-term periods well, including managing the time series data in the data set most efficiently. The weekday load pattern has slightly higher forecast accuracy than the holiday load because the holiday load is more volatile. The next most efficient models are the Bi-LSTM and Bi-GRU models because the type of data that needs to be forecasted may not give enough weight to consider the data from the past and the future, causing the model to learn the relationship and not be able to use its full potential. The last model is the GRU model because it is a simpler model than the LSTM model, which makes it not as good at learning the relationship of the data set, and this type of data may not be suitable for the model. While Suranaree University's power load can be reliably predicted by LSTM and Bi-LSTM models, there are still some restrictions that should be considered when using these models in other situations.

To properly understand long-term correlation patterns, the models need a lot of high-quality data. In the case of data-constrained environments, such as small-scale power systems in rural areas or organizations with data shortages, the model performance may be reduced. In addition, Bi-LSTM and Bi-GRU models are complex and require higher computing resources than GRUs, which may not be suitable in systems with limited computing resources, such as IoT devices or edge computing systems. However, the models can be applied in different environments, both in the large-scale energy industry and in distributed energy systems, by adjusting the model structure and parameter settings to suit the data complexity in each context or adding variables such as weather data or sunrise-sunset times to increase the accuracy of predicting fluctuating energy usage. The use of GRU models with higher model training speed in real-time systems may be a suitable option. The results of this

study have important implications for the energy sector in many dimensions. The developed electricity load forecasting model can help enhance the efficiency of energy management strategies in various contexts. By precisely predicting energy demand ahead of time, energy suppliers may effectively plan power generation and distribution, minimize overloading, and minimize transmission line losses through integration with smart grid energy management technologies. These models can be used in industrial energy management to predict the energy demand for pieces of equipment or production processes. This helps energy managers plan when to use energy (demand-side management), lower electricity costs during periods of high energy prices, and make the production system more sustainable. In addition, accurate forecasting models can be used to forecast energy demand in conjunction with renewable energy sources such as solar and wind power, which can help reduce energy imbalances and improve the efficiency of renewable energy use, as well as support the reduction of unnecessary energy use as part of the strategy to reduce greenhouse gas emissions in the energy sector.

3.8.4 Performance of LSTM model on 12 test sets

In power forecasting for systems with continuous load variability, the use of machine learning models, especially Long Short-Term Memory (LSTM) models, is a popular approach that shows good performance in many contexts. This is because LSTMs have the ability to remember data sequences and detect long-term patterns in data. Testing the LSTM model on datasets with different characteristics in terms of energy consumption behavior, such as holidays, exam days, or different seasons, can help to clearly assess the model's flexibility and accuracy. The datasets were managed by performing feature engineering before importing the model. Therefore, the analysis in this section focuses on comparing the performance of the LSTM model on 12 test datasets, which reflect different temporal characteristics, along with the use of performance indicators such as MAPE, MAE, and RMSE. The performance evaluation results of the LSTM model on 12 test datasets, each of which represents a variety of time periods or situations, such as long holidays, school openings, exam days, or

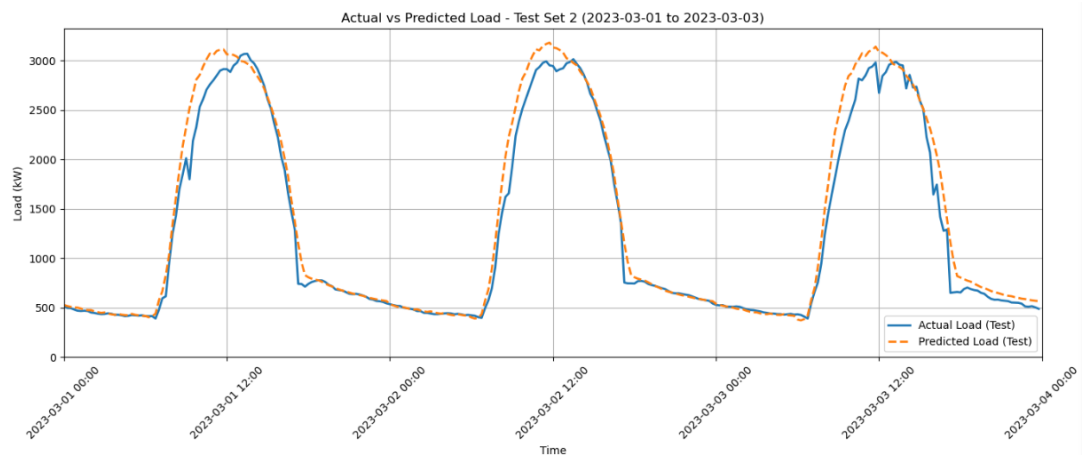


Figure 3.9 Forecasting Result – Test Set 3: Exam Period

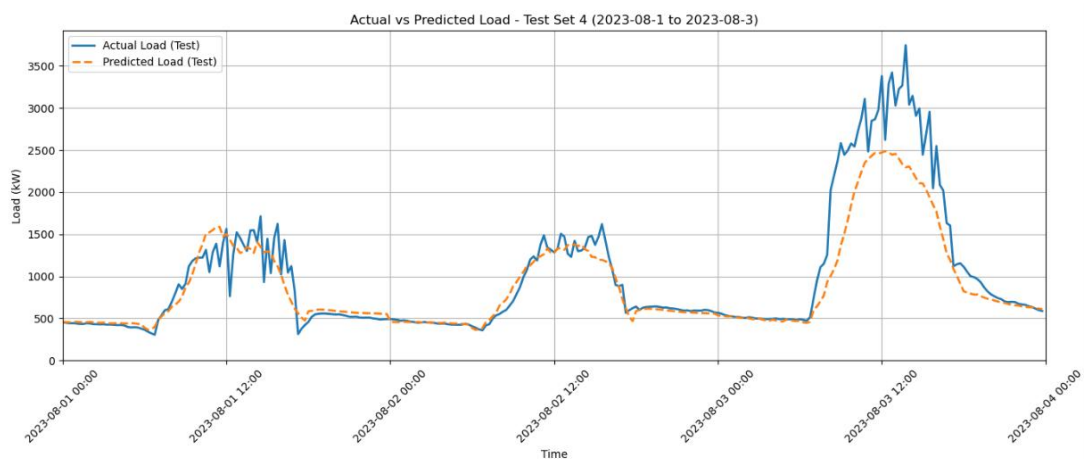


Figure 3.10 Forecasting Result – Test Set 6: Start of the Semester

different seasons, are presented. Each data set consists of 3 days of data to clearly show the trends in energy consumption behavior in each period, The results are shown in Appendix A.

From the evaluation results, it was found that Set 3 (exam period 2023-03-02) gave the lowest MAPE value at 5.93%, indicating the highest forecast accuracy with MAE equal to 83.05 kW and RMSE equal to 145.60 kW, which showed good results in terms of percentage of error. Meanwhile, set 6 (the start of the semester, 2023-08-02) gave the lowest MAE and RMSE values, namely MAE = 59.18 kW and RMSE = 93.66 kW, indicating that in terms of absolute error and root mean square error, the model

can predict the closest to the actual value. For analysis, the forecast results of Set 3 and Set 6 are presented in graph form. The graph shows the difference between the actual value and the value predicted by the model, helping to show the level of model consistency in each period shown in Fig. 3.9 and Fig. 3.10.

Table 3.6 Model Performance on Validation and Average Performance on 12 Test Sets.

Methods	Model performance of the validation set			Average performance of the model on 12 test sets		
	MAPE (%)	MAE (kW)	RMSE (kW)	MAPE (%)	MAE (kW)	RMSE (kW)
LSTM	13.71	177.58	315.67	14.07	188.77	302.04

In addition to the evaluation of each dataset, Table 3.6 shows the overall evaluation results in two aspects: the prediction results on the validation set and the average model results from testing on the 12 datasets. On the validation set, the LSTM model yielded MAPE = 13.71%, MAE = 177.58 kW, and RMSE = 315.67 kW. The average of the model on the 12 test data sets yielded MAPE = 14.07%, MAE = 188.77 kW, and RMSE = 302.04 kW. These results reflect that the model can maintain a stable level of accuracy when comparing the validation set and the test set, with an acceptable level of error for use in short-term forecasting systems. From the evaluation of the LSTM model's performance on various data sets in terms of time and context, it was found that the model can forecast accurately in some time periods, especially during exams and school openings, where the forecasting results are more accurate than other periods. The use of lag features and clustering data by time characteristics enhances the model's potential to capture the patterns of load data. In addition, the overall evaluation of the validation set and the average values from the test sets confirm that the model is stable and can be effectively applied to load forecasting in various contexts.

3.9 Summary of the daily forecast

This article presents an alternative approach to developing short-term electric load forecasting models and improving their accuracy through deep learning techniques, including the Day-Aware Data Arrangement method. The most widely used forecasting methods in this field were simulated and compared to evaluate their performance and accuracy. The daily electric load forecasting for Suranaree University of Technology often correlates with past time intervals, allowing LSTM to effectively capture these patterns better than other models. As a result, its load predictions are the closest to the actual values. Adding bidirectional layers to the models did not significantly enhance the results for electric load data. The GRU model, while easier to implement and faster to process, demonstrated lower performance and accuracy compared to the LSTM model for sequential data of this nature. Notwithstanding the encouraging outcomes, this study also discusses significant shortcomings and challenges for further applicability. For maximum performance, high-quality and adequate data are necessary, which might be difficult in contexts with limited resources. Nonetheless, similar models can be used in various energy sectors by reorganizing them, adding new variables, and adjusting the parameter values. The study's findings have important ramifications for the energy industry overall. Planning energy production and distribution effectively, reducing energy losses, and maximizing the integration of renewable energy sources are all made possible by the inclusion of accurate forecasting models, particularly LSTMs, into energy management techniques. Faster GRU models may be suitable for real-time applications where computational resources are limited. And adding Feature Engineering to evaluate the stability of the LSTM model on 12 test sets to study the variability of the results that may be caused by different data characteristics at different times.