

CHAPTER IV

Enhanced Data-Driven Load Forecasting Framework for High PV Penetration Electrical System: A Case Study of SUT's Campus

4.1 Introduction

This section proposes an approach to developing a one-day-ahead electrical load forecasting model using deep learning techniques. It compares deep learning models with basic models, namely Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM, to improve accuracy and reliability through Mutual Information (MI) and Shapley Additive Explanation (SHAP) analysis. This approach utilizes data collection and feature engineering to optimize the context of an electrical load with integrated solar power generation, Using Suranaree University of Technology (SUT) campus electrical system as a case study. Three accuracy indices are examined: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). Among the NN-based models, Bi-LSTM and LSTM tend to provide the best overall forecasting performance from the input data, and the Bi-LSTM model is the most accurate model with the lowest metric value among all models. The lag load feature, or the value of the electrical load in the past one day, is the feature that is most related to the forecasting target and has the most impact on the model's decision. Parts of this thesis have been published in ECTI Transactions on Electrical Engineering, Electronics, and Communications (2025), is shown in the Appendix D, and another set of simulated data is presented to further enhance the reliability of the framework, also shown in this section.

4.2 Data Collection and Preparation

Before entering the model for load forecasting, Suranaree University of Technology's electricity consumption data was collected for 1 year, collected every

15 minutes in 2023, totaling 35,040 data sets, which is considered high-resolution data for load forecasting. The data was collected from Automatic Metering (AMR) maintained by the Provincial Electricity Authority (PEA). In addition, solar energy generation data was collected from an Automatic Metering (AMR) maintained by an external company, ENOS, with the same resolution as the electricity load data. Weather data and academic calendar data for that year were collected and prepared to input all this data into the future feature engineering method. The required data is the actual load data. Therefore, the load obtained from the meter was combined with the solar energy generation data to obtain data showing the actual load of the university.

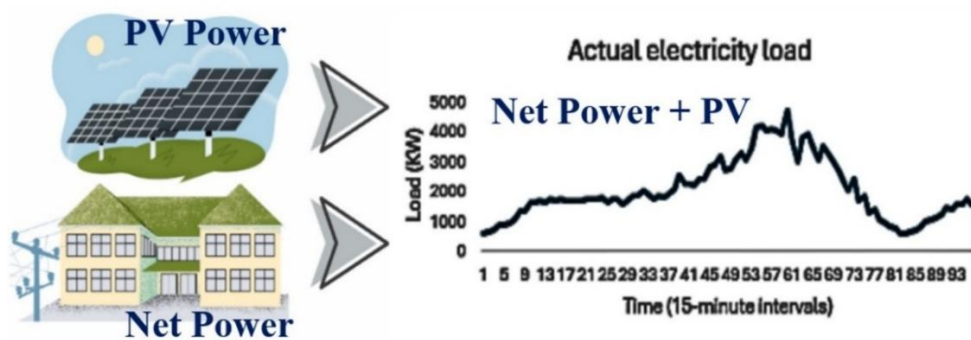


Figure 4.1 Preparation of Actual Electricity Load Data

The preparation of the real electricity load data can be shown on Fig. 4.1. All data used as features and output for the model to learn were divided into two parts: the first part was the data set to be trained, and the other part was partitioned for model validation. Time Series Cross-Validation (TSCV) was performed on this data set, which is a method of data partitioning to evaluate models that take into account temporal order and prevent future data leakage. The training and validation sets were generated using a Time Series Split with 5 folds. The second part was the data set tested on the model after training and validation were completed. It was a data set that the model had never seen before, so the results were unbiased and divided into 4608 data points. For testing, this group of data will be divided into 10 sets; each set

will have 3 days. Each set is divided according to the types of data groups, namely long holiday, weekdays during the school term, exam period, summer season, the start of the semester, compensation holiday period, rainy season, weekdays and weekends, term break, and winter season.

4.3 Feature Modeling and Representation of Electrical Load Patterns

Feature engineering has a group of data that has the characteristics of feature creation, which is creating new features from existing time data and converting time data that may not be easy for the model to logical data with contextual meaning.

4.3.1 SUT's Campus Data

It consists of the following techniques: In the feature transformation method, features that use this method include Weekdays and Weekend, Daytime and Nighttime, Compensatory Holiday, and Long Weekend, which are converting to variable values and binary representation with numbers 0 and 1. In the time-based feature engineering method, features that use this method include yearly hours, which divides the data into meaningful time sequences with 1 year's data every 15 minutes, creating features that allow the model to learn the day and time. The Time Component Extraction method features that use this method include Quarter-Hourly Interval, which extracts time components into separate features, creating new features that describe the sub-patterns that occur in each hour and the lag features method. Features that use this method include Lag Load, which creates features from previous values by taking historical load values to create features. In addition, there is a group of characteristics: external features. External variables affect the forecast of the target variable, such as ambient temperature, an external factor that directly affects energy use behavior, and dew point, an indicator of air humidity.

4.3.2 New England (ISO-NE) Data

Time-based feature generation plays a crucial role in enabling the model to efficiently learn periodic patterns and long-term trends of electrical load in the second dataset. Additional relevant features can be generated as follows:

The "Day" feature is used to reflect the order of days within a month, allowing the model to recognize changes in electricity usage behavior that may occur at the beginning, middle, or end of the month, such as during payday or before long holidays. The "Month" feature represents the seasonality of electrical load, particularly the impact of different weather conditions throughout the year, such as summer or rainy season, which directly influences electricity demand. The "Year" feature plays a role in reflecting long-term trends in electrical load, such as increases in the number of electrical appliances, infrastructure expansion, or changes in energy policy. This allows the model to appropriately differentiate electrical load behavior over several years. Furthermore, the "Weekday" feature is used to distinguish between weekdays and holidays, a key factor clearly influencing electricity usage patterns. Generally, electrical load patterns on weekdays differ from holidays, both in terms of load levels and peak times. Incorporating this feature allows the model to better learn about the differences in electricity usage behavior across different types of days.

In summary, adding time-related features—Day, Month, Year, and Weekday—along with Lag Load, Ambient Temperature, and Yearly Hours features in the ISO-NE dataset for further simulation enhances the model's ability to capture temporal relationships, long-term trends, and periodic patterns of electricity load—critical components for accurate and stable electricity load forecasting in the context of time series data.

4.4 Feature Selection and Mutual Information

Mutual Information (MI) captures the relationship of input features to a non-linear data model, shows their importance, and indicates how the feature reduces uncertainty with respect to the forecasting target. It is used as a tool for selecting appropriate forecasting features, especially in tasks where the number of features varies without necessarily being linear. It is calculated using the following equations (4.1).

$$i(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \cdot \log \left(\frac{p(x, y)}{p(x) \cdot p(y)} \right) \quad (4.1)$$

where Y is the target variable, i.e., the future load, X is the input variable, $p(x, y)$ is the joint probability of x and y , and $p(x)$ and $p(y)$ are the marginal probabilities of each variable. The value of MI is expressed as a real number greater than or equal to 0. The higher the value of MI, the more the feature is related to the forecast value. MI can help reduce the size of the data to control both the speed of training and the interpretation of the model. MI is a powerful tool for feature selection for time series models, especially in complex load forecasting tasks that change over time and in different contexts. Selecting appropriate features using MI can make the model more accurate and easier to interpret.

Feature selection in load forecasting each feature plays a physical or behavioral role in load patterns. The Lag Load feature was selected because electrical loads are similar and have daily activities that often repeat within 24 hours, making it a very important representative of daily patterns in forecasting. The Yearly Hours and Quarter-Hourly Interval features help the model understand the time sequence of each day and allow the model to recognize period differences. When used in combination with other features, the model can better learn about daily trends. The Weekdays and Weekend features were selected because electricity usage by electricity consumers differs significantly during holidays and weekdays. The Daytime and Nighttime features were selected because people's behaviour shows higher electricity usage in the morning than at night, so the model should be able to see this difference. The Compensatory Holiday and Long Weekend features have different load patterns during long holidays and special holidays. The Ambient Temperature and Dew Point features are weather-related features that reflect the characteristics of electricity consumers when temperatures and humidity change. They also provide the model with sufficient forecasting capabilities when there is load variation caused by changing weather conditions.

4.5 Modeling Approaches and Baseline Methods

Accurate short-term electricity load forecasting requires models that can effectively capture temporal dependencies, nonlinear relationships, and variations in

consumption patterns under different operating conditions. To comprehensively evaluate forecasting performance and demonstrate the effectiveness of advanced deep learning techniques, this study investigates both baseline forecasting methods and a range of time-sequence deep learning models.

The models used to simulate results in the Recurrent Neural Network (RNN) category were compared and selected. The models with the highest potential for accuracy were LSTM and Bi-LSTM, as discussed in the previous chapter. The scale was then expanded to compare with other categories to cover a wider range of subjects. In the classical statistical time-series method category, the moving average method was selected. This is a traditional statistical forecasting method that calculates the average of the electrical load over a given period and uses that value as a forecast for the next period. It uses the average of the past 3 days. This method does not involve model learning and is used as a basic method. In the Feedforward Neural Network (FNN) category, the DNN model was selected. This is a multi-layered hidden neural network where data flows forward in only one direction from input to output, without memory of the past like RNNs. In the hybrid model category, the models selected were Attention-LSTM and CNN-LSTM.

4.5.1 Attention-LSTM Model

Attention-LSTM consists of an LSTM model that extracts deep features from time-dependent data and an attention component that determines which data in the sequence has the greatest impact on load forecasting. Features fed as input are given into the LSTM model, with the input data at each time point represented by variables X_{t-4} , X_{t-3} , X_{t-2} , X_{t-1} , and X_t . The model then generates a hidden state (h_t) that stores in-depth information about past electrical load behavior, as shown by the equation (4.2).

$$h_t = LSTM(X_t, h_{t-1}, C_{t-1}) \quad (4.2)$$

Each timestep yields the results h_{t-4} , h_{t-3} , h_{t-2} , h_{t-1} , h_t , where C_{t-1} is the cell state capable of retaining information for a long period. Since the LSTM model focuses only on the last h_t , an attention mechanism is added to emphasize (weight) each hidden

state. It creates weights ($\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$) indicating the importance of the information in that timestep and its impact on the forecast for each hidden state. A softmax function is then used to normalize the result so that the sum equals 1. This is shown in the equation (4.3).

$$\alpha_i = \frac{\exp(\text{score}(h_i))}{\sum_{j=1}^T \exp(\text{score}(h_j))} \quad (4.3)$$

Then, the hidden states are multiplied by the weight (α) and combined to create a Context Vector (C), which is the central value summarizing the historical load data that the model prioritizes most. This represents sequence data all the time and can be expressed as the follows equations (4.4).

$$C = \sum_{i=1}^T \alpha_i h_i \quad (4.4)$$

Once C is obtained, it is combined with the LSTM Output to create the Feature Final (Fea_{final}), which represents the features processed by both LSTM and Attention. It is then passed through the Dense Layer to predict the electrical load for the next day. The attention mechanism was implemented using Luong dot-product attention, where the attention weights (α) are normalized using the softmax function. Fea_{final} represents the combined feature vector generated from the LSTM hidden states and attention mechanism. The structural diagram of the model is shown in Fig. 4.2.

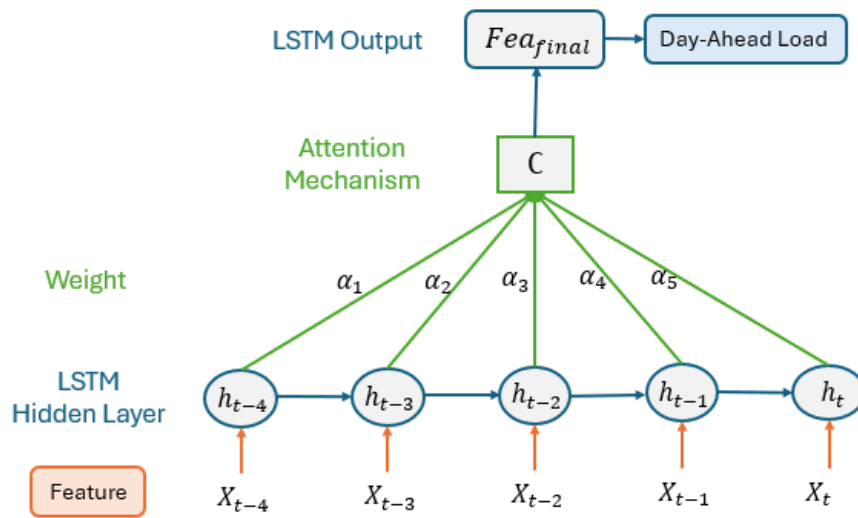


Figure 4.2 Attention-LSTM architecture

4.5.2 CNN-LSTM Model

The CNN-LSTM model is a combination of Convolutional Neural Network (CNN) and LSTM models for improved feature extraction and learning of temporal dependencies. CNN extracts feature from sliding window load data using 1D convolution to compute the feature map at position t . A kernel of size k is used, employing multiple filters to detect different patterns, as shown in the equation (4.5).

$$z_t = \sum_{i=0}^{k-1} w_i \cdot x_{t+i} + b \quad (4.5)$$

Where w_i is the filter weight (kernel weight), b is the bias, i is the index of the position within the kernel, x_{t+i} is the model input at time $t + i$, and z_t is the feature value obtained after convolution. A pooling layer is used to reduce the dimensionality of the data and extract only the important features, where p is the size of the pooling window, as shown in the equation (4.6).

$$p_t = \max(z_{t:t+p}) \quad (4.6)$$

Where p_t is the pooling result at time position t , $z_{t:t+p}$ is the sequence of feature map values obtained from the convolution step of the CNN from time t to $t + p$, and $\max(\cdot)$ is the max pooling function. After the CNN model finishes extracting features, it sends the obtained features to the LSTM model to learn the time sequence and finally sends them to the Dense Layer to forecast the electricity load for the next day. The predicted load for the next day is given by the equation (4.7).

$$\widehat{Y}_{t+1} = W_d \cdot H_t + b_d \quad (4.7)$$

This is the step for creating the output layer of the model, where $\widehat{Y}_{t+1}$ is the value, the model predicts at the next time point $t + 1$, and H_t is the hidden state of the LSTM at time t , which is the result of the time sequence learning process. This data has first featured using a CNN for short-term spatial or pattern extraction before being fed into LSTM to learn temporal relationships. It represents temporal memory that includes past load patterns and the influence of other features. W_d is the weight matrix of the Dense layer (fully connected layer), which transforms the hidden state of the LSTM into a dimension suitable for the desired output prediction, and b_d is the bias of the output layer, which adjusts the prediction value independently of the input,

allowing the model to be more flexible in estimating the true value. The CNN-LSTM model uses CNN to extract key patterns from historical data, then uses an LSTM to learn temporal relationships, and finally uses the Dense layer to transform the hidden state into the predicted electrical load value.

4.5.3 DNN Model

As part of the Artificial Neural Network (ANN) method, it learns deeper input-output relationships, especially in tasks with nonlinear data. The model consists of three parts: the Input Layer, which receives feature data and prepares it for encoding before sending it to the next layer; next are the Hidden Layers, which learn increasingly complex data and can be divided into multiple layers to accommodate more complex incoming data; and finally, the Output Layer, which predicts the continuous load (h^l). The calculation is shown in the equation (4.8).

$$h^l = f(w^l \cdot h^{l-1} + b^l) \quad (4.8)$$

Where w^l is the weight of the l layer, b^l is the bias constant, and $f(\cdot)$ is the activation function.

4.5.4 Moving average

This is a traditional, basic method used to forecast electrical loads with relatively stable patterns and minimal data fluctuations. The type used is the Simple Moving Average (SMA). Future forecasts depend on the average load usage over similar past periods, focusing on patterns and trends. The forecast value for the current day at time t will be equal to the average of the past three days at time t , as shown by the equation (4.9).

$$\widehat{Y}_{d,t} = \frac{Y_{d-1,t} + Y_{d-2,t} + Y_{d-3,t}}{3} \quad (4.9)$$

Where $\widehat{Y}_{d,t}$ is the forecast value for day d at time t and $Y_{d-1,t}$ is the actual load value for the previous days at time t .

4.6 Time Series Cross-Validation (TSCV)

Model performance evaluation is an important part of the data analysis process, especially in tasks involving time series data such as load forecasting, which are characterized by temporal order and temporal dependency, making it impossible to use traditional cross-validation techniques that are applied to independent and identically distributed data. To solve this problem, Time Series Cross-Validation (TSCV) is a method of data partitioning to evaluate models that take into account temporal order and prevent future data leakage. In conventional cross-validation, such as K-Fold Cross-Validation, data is randomly partitioned regardless of the order of the data. This will result in future data being present in the training set and historical data being present in the test set, resulting in data leakage that makes the model evaluation results do not reflect the true performance. TSCV solves this problem by preserving the order of the data and using a "walk forward" or "rolling window" approach instead. Here we will discuss the expanding window (walk-forward validation) approach, which starts with a small training data set and gradually adds new data to the training set, while the test set always moves in time. This is suitable for models that need to "learn" from the past and predict the future, especially in contexts where the data tends to change cumulatively, as shown in Fig. 4.3.

Fold	Train Index	Validation Index
1	[0, 1]	[2, 3]
2	[0, 1, 2, 3]	[4, 5]
3	[0, 1, 2, 3, 4, 5]	[6, 7]
4	[0, 1, 2, 3, 4, 5, 6, 7]	[8, 9]

Figure 4.3 Expanding Window

Time Series Cross-Validation is an important process that enables accurate model evaluation in the context of temporal data, especially in load forecasting applications where data are time-varying and have a seasonal structure. Proper selection of TSCV prevents data leakage, increases the reliability of evaluation results, and supports the creation of models that can be confidently put to practical use.

4.7 Shapley Additive Explanation (SHAP)

In a load forecasting system that uses complex models, users often need not only accurate predictions but also to understand why the model produces those predictions in order to build confidence, support business decisions, and verify the stability of the model when the environment changes. SHAP is one of the tools that can provide a rational and easily interpretable explanation of these models.

SHAP is a method based on the Shapley Value from Cooperative Game Theory, which measures how each feature contributes to the model's decision. SHAP transforms the model description problem into a game in which each feature is a player, and the model's outcome is a reward that must be shared (prediction). The Shapley Value calculation considers all subsets of features (feature coalitions) to assess the impact of each feature. The calculation principle (conceptually) can be described as the Shapley value of feature i is defined by the following equation (4.10).

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! (|N| - |S| - 1)!}{|N|!} \cdot [f(S \cup \{i\}) - f(S)] \quad (4.10)$$

where N is the set of all features, S is the subset of features excluding i , $f(S)$ is the predicted value of the model when using only the feature in S , and ϕ_i is the average of the effects that feature i has on the predicted value when combined with all other features in various orders.

In load forecasting, the input features can be complex. SHAP can analyze how much historical load (lag features) or temperature influence each time period, interpret which factors cause the model to overproduce or underproduce the forecast, and

identify globally and locally important features, or analyze both overall and case-specific important features. SHAP is the current model explainer, especially for load forecasting tasks that require both accuracy and transparent understanding of the model. It allows us to look into the black box and distinguish how each feature affects the outcome, which is in line with the goal of explainable Artificial Intelligence.

4.8 Forecasting Process Description

The development of an accurate and interpretable short-term load forecasting model involves a series of systematic processes, each contributing to the robustness and practicality of the final solution. This research was carried out in multiple stages, starting from the preliminary design and comparison of deep learning models and progressing through advanced techniques for feature selection, model validation, and explainability. The overall workflow can be divided into three main phases, each emphasizing different aspects of model enhancement: initial development, performance refinement, and interpretability.

Phase 1: Initial Development and Model Comparison The research began with data collection and preprocessing, focusing on historical load data from Suranaree University of Technology, complemented by contextual variables such as temperature, humidity, calendar effects, and time indicators. The raw data were cleaned, normalized, and structured into a time series format suitable for model training. Subsequently, feature engineering was employed to capture the temporal characteristics of the dataset. Manually crafted features, such as day of the week, hour of the day, holiday indicators, and seasonality factors, were added to enrich the dataset. These features are known to have a significant impact on electricity consumption in institutional settings. The core of this phase was the design and training of four deep learning models. These models were selected for their proven effectiveness in modeling time-dependent patterns. After training the models on the same dataset, their performance was evaluated using standard metrics, including Root

Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The comparison results consistently show that this model outperforms other models, especially during periods of high workload volatility. Therefore, it is selected as the most suitable model for further development.

Phase 2: Feature Selection and Time-Aware Validation To improve accuracy further and address potential overfitting, the next phase introduced automated feature selection using the mutual information (MI) method. MI assesses the non-linear dependency between each input variable and the target load, helping to retain only those features that contribute meaningfully to prediction performance. This step allowed for dimensionality reduction while preserving the complexity needed to reflect real-world electricity usage patterns. After refining the feature set, the model was retrained using the selected features. In this phase, model evaluation was conducted using Time Series Cross-Validation (TSCV) with an Expanding Window strategy. Unlike traditional k-fold cross-validation, which can lead to data leakage in time series data, TSCV respects the temporal order of the data, simulating real-world forecasting conditions where only past information is available to predict the future. This enhanced validation approach provided a more realistic and robust assessment of the model's generalization ability.

Phase 3: Explainability and Model Transparency Although deep learning models can offer high accuracy, their black box nature often limits their practical utility in energy decision-making environments. To address this, the final phase introduced an explainability mechanism using SHAP. SHAP quantifies the contribution of each feature to a specific prediction (local explanation) and identifies the most influential features across the dataset (global importance). Insights from SHAP analysis were used to refine the feature set further. Features with low or inconsistent importance were reviewed, and the model was retrained with a final set of contextual and statistically relevant variables. This interpretability step not only improved the transparency of the forecasting system but also provided actionable insights for energy managers and

stakeholders, who could now better understand the reason behind the model's outputs.

The complete workflow—beginning with basic data preparation, progressing through model comparison, and culminating in interpretability—forms a robust framework for load forecasting that balances accuracy, transparency, and practical applicability. Each phase is built upon the previous one: initial model comparisons identify a strong baseline, advanced feature selection and time-aware validation enhance accuracy, and SHAP-based interpretability ensures model decisions can be trusted and understood. The overall research process is presented in Fig. 4.4, which illustrates the step-by-step process from data collection to descriptive load forecasting.

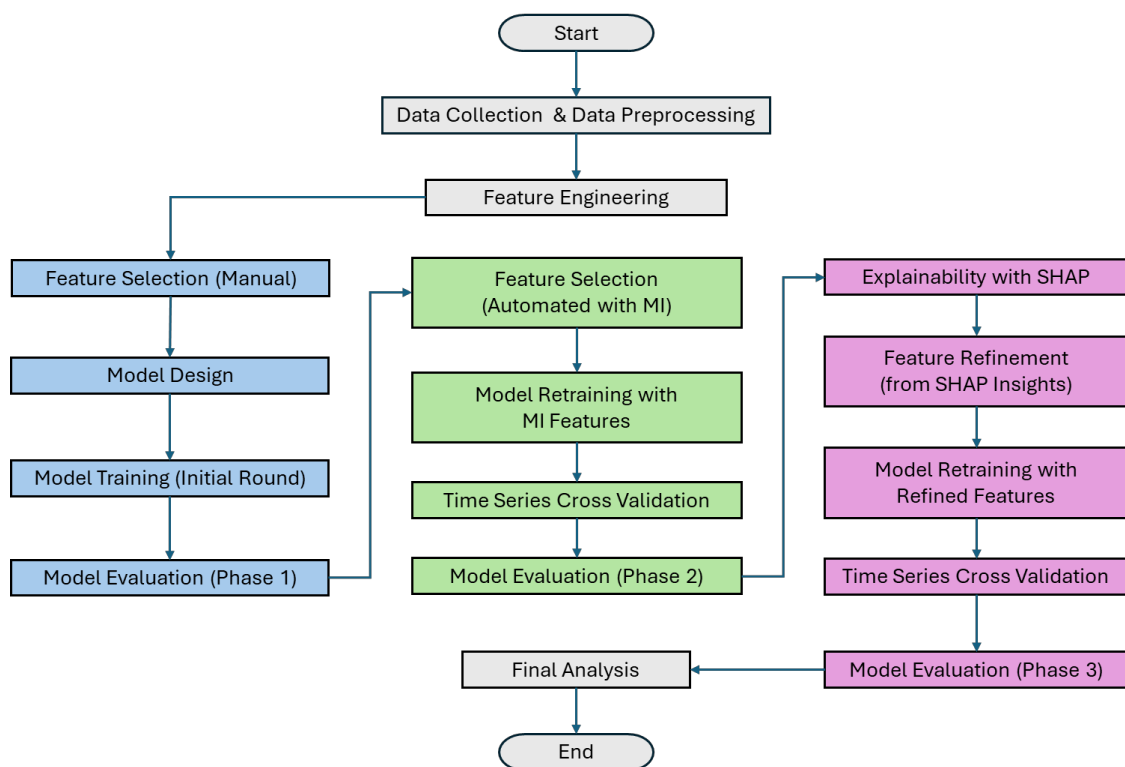


Figure 4.4 Process Description

4.9 Load Forecasting Framework

Electrical load forecasting requires the preparation of essential data, including aggregation and collection from reliable sources in the forecasting area, focusing on actual electricity load data from meters. This data is then fed into feature engineering. Two types of data are used: feature generation, which converts raw data into numbers that are easier for the model to understand and allows the model to better learn about data differences, and external feature data, which learns about data trends. This process takes place during the data collection and preparation process.

The next step in the simulation process is automatic feature selection using MI to assess the nonlinear relationship between each input variable and the target load. This helps preserve features that are important for forecasting. This reduces data complexity while retaining the most important features. The data is then grouped into three groups for training, validation, and testing, based on proportions. The resulting data is then fed into the model by designing the model, setting various parameters, and training deep learning models in various categories: Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM. After training the models on the same dataset for all models, then, in the results and analysis process, the performance is evaluated using standard metrics: root means square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

To enhance the transparency and clarity of the black-box model, the SHAP tool is used to guide decision-making on feature prioritization. This tool identifies the lowest and highest priority features for the final pre-load forecast. Figure 4.5 shows the entire forecasting process. The first step is the data collection and preparation process. The next step is the simulation process. The last step is the results and analysis process.

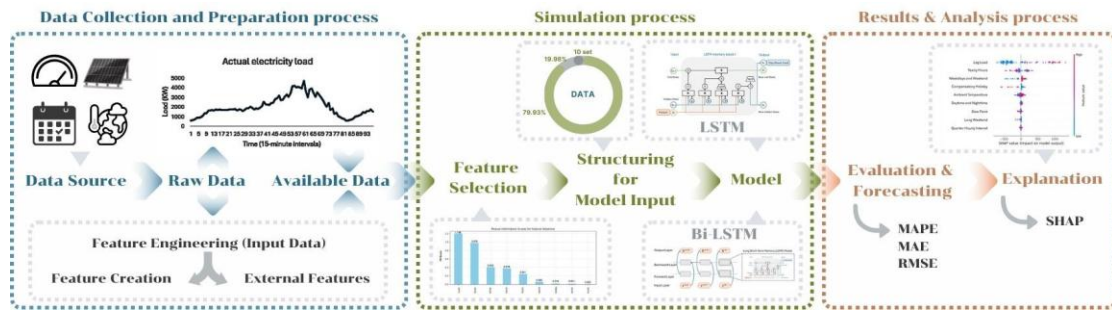


Figure 4.5 The Complete Forecasting Framework

4.10 Simulation Results and Discussion with SUT's Campus Data

The electrical load forecast from all the data obtained and the tools mentioned above will be used in simulation using the Python 3 program to show the results comprehensively and explain the accuracy and reliability of the model in the area and context of Suranaree University of Technology. It is a 1-day-ahead load forecast using historical data for 1 year of forecasting in 2023. The details are presented in this section. Using Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models.

4.10.1 Implementation and Experimental Setup

The settings for each model are shown in Table 4.1. All models use similar parameters, namely the Dropout Rate parameter with a value of 0.2 to set the percentage of that unit. To prevent overfitting in the model, the learning rate is set to 0.001 to determine the speed of parameter tuning during learning. The optimizer uses Adam to automatically adjust the learning rate. The batch size, which is the number of data used to calculate the gradient per iteration of parameter tuning, is set to 32. The Epochs, which is the number of iterations to learn the whole data set, is set to 50. The loss function is used to evaluate the model error with MSE. In the different parts, each model is a layer part, which is the basic structure of the model that defines data processing. The DNN model has a Flatten Layer to transform the data and feed it into a Dense Layer, with the number of units in each Dense Layer having the units or the complexity of learning in each layer as 128, 64, and 1. The CNN-LSTM model has Conv1D filters, or the number of filters, as 64. The kernel size is the length of the

repeated pattern detection filter with the value of 3. The pool size for reducing the dimensionality is set to 2. The LSTM Units are set to 100 for learning the temporal relationship after Conv1D has extracted features and has a Dense value of 1 in the model part. The Attention-LSTM has an Attention Layer that prioritizes the timesteps that affect the prediction and returns 100 LSTM Units with a Dense of 1. In the LSTM model, the LSTM Unit Capacity is 100, and the Bi-LSTM model has a Bidirectional Wrapper to process two directions and uses 100 LSTM Units in each direction with a Dense of 1.

Table 4.1 Parameter Setting

Model	Parameters Layer
DNN	Flatten, Dense (128), Dense (64), Dense (1), and Units: 128, 64, 1
CNN-LSTM	Conv1D (64), kernel size (3), MaxPooling1D (pool size=2), Dense (1) and LSTM (100)
Attention-LSTM	LSTM (100), return sequences (True), Dense (1) and Attention
LSTM	LSTM (100) and Dense (1)
Bi-LSTM	Bidirectional (LSTM(100)) and Dense(1)

The selection of the appropriate hyperparameter values for each model was performed using a trial-and-error method. The parameters were adjusted within a specified range and then evaluated by evaluating the minimum MAPE value displayed when each parameter was adjusted within each range. Hyperparameter units were set in a range of 64 to 128, with a best value of 100. The dropout rate was set in a range of 0.1 to 0.3, with the best value of 0.2. The learning rate was set in a range of 0.005 to 0.0005, with a best value of 0.001. Timesteps were set in a range of 4 to 12, with a best value of 8. Batch size was set in a range of 16 to 64, with a best value of 32. Early stopping was set in a range of 5 to 20, with a best value of 10. As shown in Table 4.2, these hyperparameter sets were then used to simulate the results of each model. The details of the parameter adjustment in the Bi-LSTM model, evaluated using the MAPE

indicator, show a minimum value of 17.53% resulting in the optimal parameter range shown in Table 4.3. Example code for simulating the Bi-LSTM model is shown in Appendix B.

Table 4.2 Selecting the appropriate hyperparameter

Hyperparameter	Adjustment range	Best parameter values
Units	[64 128]	100
Dropout Rate	[0.1 0.3]	0.2
Learning Rate	[0.0005 0.005]	0.001
Timesteps	[4 12]	8
Batch Size	[16 64]	32
Early Stopping	[5 20]	10

Table 4.3 Details of parameter adjustment via indicators

Case	Units	Dropout Rate	Learning Rate	Timesteps	Batch Size	Early Stopping	MAPE (%)
1	64	0.2	0.001	8	32	10	21.28
2	128	0.2	0.001	8	32	10	19.69
3	100	0.1	0.001	8	32	10	18.64
4	100	0.3	0.001	8	32	10	21.56
5	100	0.2	0.005	8	32	10	19.95
6	100	0.2	0.0005	8	32	10	19.98
7	100	0.2	0.001	4	32	10	21.01
8	100	0.2	0.001	12	32	10	18.83
9	100	0.2	0.001	8	16	10	19.47
10	100	0.2	0.001	8	64	10	17.72
11	100	0.2	0.001	8	32	5	21.04
12	100	0.2	0.001	8	32	20	19.52
13	100	0.2	0.001	8	32	10	17.53

4.10.2 Mutual Information Scores of SUT Data

To prioritize the feature data before importing the model with the MI tool, the relationship of the features and the target electrical load dip data groups is measured as both linear and non-linear data. It shows the score ranking of each feature. The Lag Load feature has an MI score of 1.198, the Yearly Hours feature has an MI score of 0.976, the Daytime and Nighttime feature has an MI score of 0.405, the Ambient Temperature feature has an MI score of 0.378, the Dew Point feature has an MI score of 0.241, the Weekdays and Weekend feature has an MI score of 0.055, the Compensatory Holiday feature has an MI score of 0.018, the Long Weekend feature has an MI score of 0.014, and the Quarter-Hourly Interval feature has an MI score of 0.002, as shown in Fig. 4.6 The introduction of lag load as a feature tends to have a significant relationship between the input and output data. The characteristics of the Lag Load feature are most clearly related to the forecast value. The behaviour of electricity users today and yesterday is similar. The features that are tested with high scores are the Yearly Hours feature, which shows the time position that repeats patterns throughout the year, showing the relationship with the forecast value in a high proportion. These are 2 features that have significantly higher scores than other models. As for other features, the scores are closer to zero and are less relevant to the forecast.

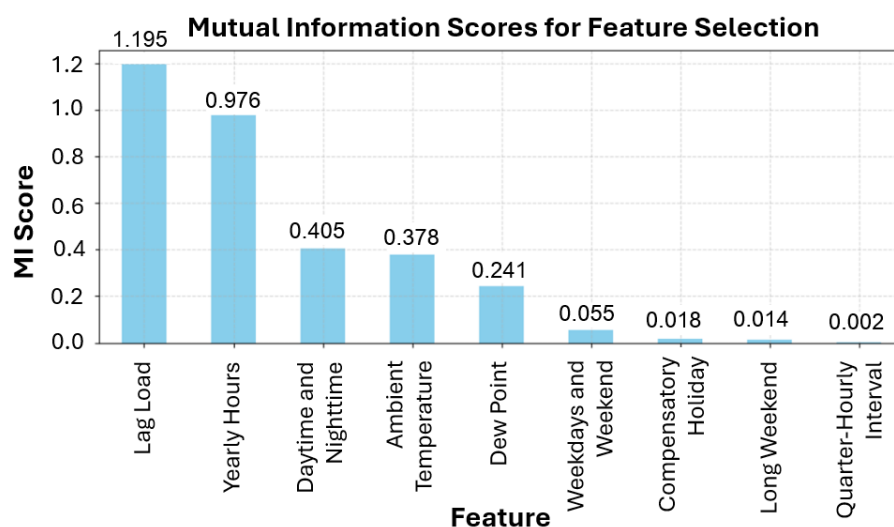


Figure 4.6 Mutual Information Scores of SUT Data

4.10.3 Daily Load Forecasting Performance

Table 4.4 Performance of Each Model on 10 Test Sets

Models	Important dates		Metric Statistic		
	Minimum	Maximum	Metric	Min.	Max.
Moving average	2023-08-01	2023-08-03	MAE (kW)	68.60	675.72
	2023-05-01	2023-08-03	RMSE (kW)	87.39	945.33
	2023-12-27	2023-09-10	MAPE (%)	5.78	78.34
Attention-LSTM	2023-04-16	2023-12-25	MAE (kW)	73.10	385.39
	2023-04-16	2023-12-25	RMSE (kW)	114.97	563.92
	2023-03-01	2023-09-10	MAPE (%)	5.99	35.40
DNN	2023-03-02	2023-12-25	MAE (kW)	97.47	403.80
	2023-03-02	2023-12-25	RMSE (kW)	122.82	615.58
	2023-09-04	2023-09-10	MAPE (%)	10.16	37.38
CNN-LSTM	2023-03-02	2023-05-02	MAE (kW)	68.33	368.25
	2023-03-02	2023-05-02	RMSE (kW)	90.55	556.91
	2023-03-02	2023-09-10	MAPE (%)	7.21	39.63
LSTM	2023-03-02	2023-05-02	MAE (kW)	83.60	362.97
	2023-03-02	2023-05-02	RMSE (kW)	105.69	528.20
	2023-09-03	2023-09-10	MAPE (%)	9.82	33.84
Bi-LSTM	2023-03-02	2023-12-25	MAE (kW)	46.75	373.37
	2023-03-02	2023-12-25	RMSE (kW)	68.35	566.79
	2023-03-02	2023-09-10	MAPE (%)	4.82	38.74
Bi-LSTM (Top 3 Features by SHAP)	2023-12-27	2023-09-11	MAE (kW)	63.52	394.71
	2023-03-03	2023-09-11	RMSE (kW)	90.11	612.98
	2023-03-02	2023-09-10	MAPE (%)	4.92	36.79
Bi-LSTM (Top 3 Features by MI)	2023-03-03	2023-08-03	MAE (kW)	103.28	469.44
	2023-03-03	2023-09-10	RMSE (kW)	141.87	722.92
	2023-03-02	2023-09-10	MAPE (%)	7.51	63.60

One-day-ahead electrical load forecasting at Suranaree University of Technology can demonstrate the performance of each of the six models through three metrics, compiled into statistical metrics that show the highest, lowest, and average values of each metric from ten test data sets. Important days in the forecast, including the days showing the lowest and highest values of each metric, are also included in the test data set for the model. It is shown in Table 4.4. The forecasting results show

that the moving average forecasting method gives the highest value of all 3 indicators from the tested data set. MAE has a value of 675.72 kW, and RMSE has a value of 945.33 kW on 2023-08-03, which is a date in the set of semesters start dates, as shown in Fig. 4.7 And MAPE has a value of 78.34% on 2023-09-10, which is a date in the rainy season set, as shown in Fig. 4.8 The forecasting method with the Bi-LSTM model gives the lowest values of MAE, RMSE, and MAPE indicators from the tested data set, with values of 46.75 kW, 68.35 kW, and 4.82%, respectively, which are a date in the exam set, as shown in Fig. 4.9 The figures show the load forecast for 1 day in advance for 3 consecutive days. Each model is graphed compared to the actual load graph to show the forecast trend.

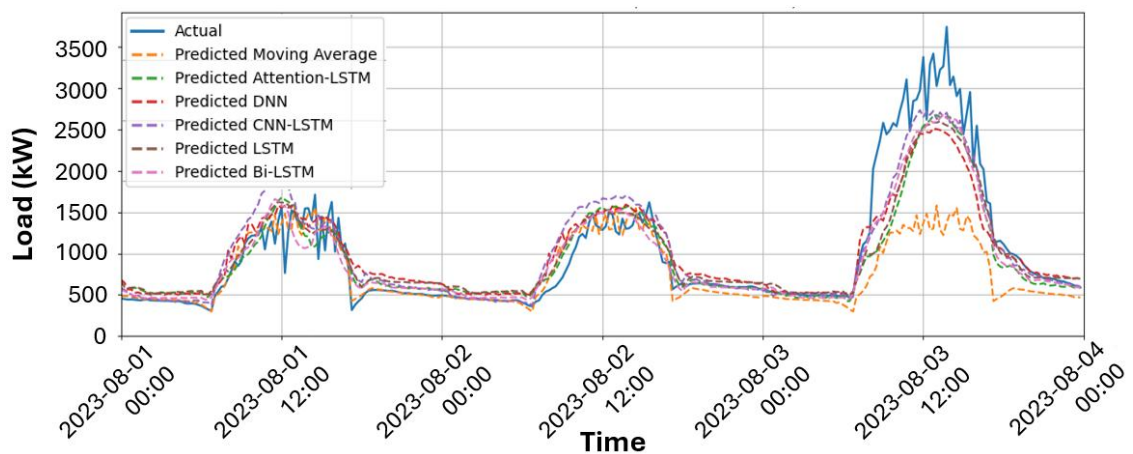


Figure 4.7 The Lowest-Performing Forecast Results in the Initial Semester Test Set

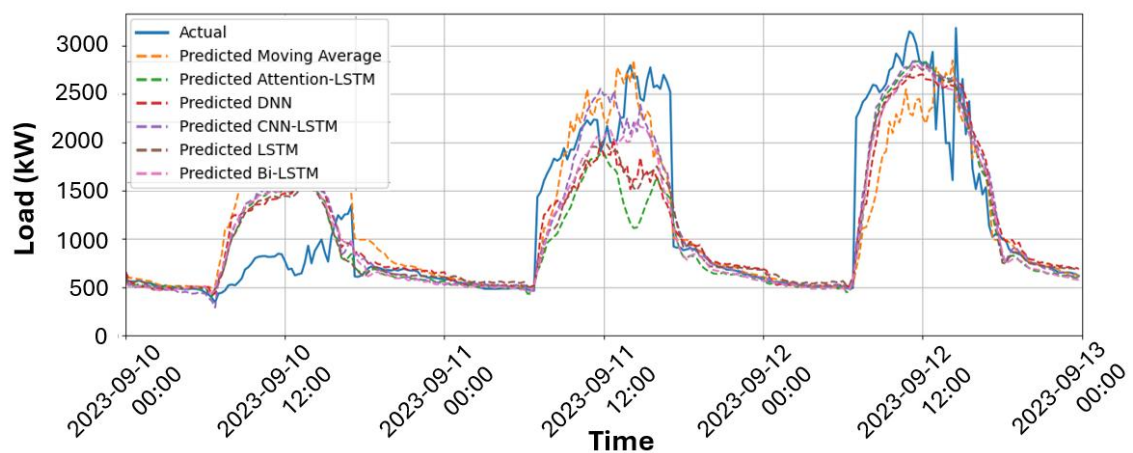


Figure 4.8 The Lowest-Performing Forecast Results in the Rainy Season Test Set

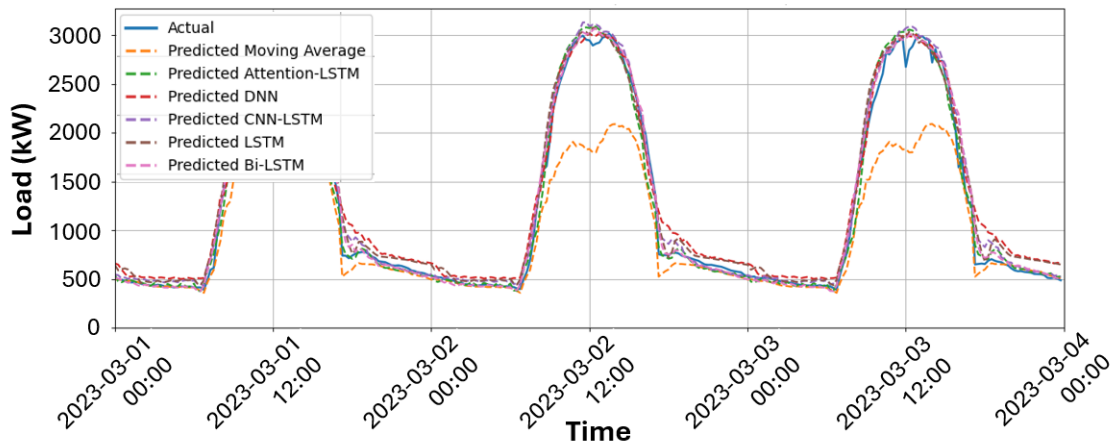


Figure 4.9 The Best-Performing Forecasting Results in the Test Set during the Exam Period

Since the moving average model relies on the average of historical load patterns, they must be similar at the same time point for accurate forecasting. However, data from the beginning of the semester, the rainy season, or the test data used for forecasting are volatile and unstable. This results in higher forecast error rates than all other models when analyzed for the maximum possible forecast error. The Bi-LSTM model has two layers that process historical data and predict future load trends, enabling it to effectively learn from data patterns that fluctuate with weather and time. Furthermore, the test data during exams or similar data have lag load and yearly hours features that significantly affect the relationship with the forecast at that time, accounting for a significant portion of the MI score. This factor occurs both before and after the forecast point. Therefore, the bidirectional learning method demonstrates the highest forecasting performance compared to all other models when analyzed for the maximum possible model accuracy. The Bi-LSTM model exhibits a lower average error (MAE) value, a percentage error compared to the actual value (MAPE), and a lower proportion of large errors (RMSE) than other models. The model can reduce errors at each time point, reduce forecasts with small errors, and

forecast peak load periods more accurately than other models. Therefore, the resulting model is the best-performing model in this context.

The LSTM model showed the lowest values of all three indicators as follows: MAE was 83.60 kW, RMSE was 105.69 kW, and MAPE was 9.82%. The model has a learning characteristic from the past to the present. There is a layer that remembers various values, which is different from the DNN model in that the features fed into the model are not chronologically ordered, which makes the indicator values higher than the LSTM model. MAE was 97.47 kW, RMSE was 122.82 kW, and MAPE was 10.16%. For the hybrid CNN-LSTM model, CNN was used to capture short-term data patterns, resulting in lower errors. The lowest values of all three indicators were as follows: MAE was 68.33 kW, RMSE was 90.55 kW, and MAPE was 7.21%. For the Attention-LSTM model, developed from a simple LSTM structure, it prioritizes the most effective timestep, showing an MAE of 73.10 kW, an RMSE of 114.97 kW, and an MAPE of 5.99%. Both models exhibit higher forecasting accuracy than the simple LSTM model. The moving average method yields the best indicators, with an MAE of 68.60 kW, an RMSE of 87.39 kW, and an MAPE of 5.78%. These are more effective than some of the models mentioned above. However, when examining the worst-day recordings, the model achieves higher indicators than all other models. Furthermore, the average test data also has the lowest overall performance.

In addition, the Bi-LSTM model was investigated to obtain the most three influential features from SHAP: Lag Load, Yearly Hours, Weekdays, and Weekend. The lowest values of the metrics were obtained as follows: MAE was 63.52 kW, RMSE was 90.11 kW, and MAPE was 4.92%. Compared to the same model set using all 9 features, the results show that the model using only the most important features provides acceptable forecast performance, although its performance is slightly reduced compared to using all features. The Lag Load feature has the highest weight in describing the daily electrical load characteristics. The Yearly Hours, Weekdays, and Weekend features still help the model better understand hourly and weekly patterns.

While selecting only 3 important features allows the model to perform well, using all features yields the lowest metric values and the best performance.

The Bi-LSTM model was re-simulated using the top three features identified as having the greatest influence by MI: Lag Load, Yearly Hours, Daytime, and Nighttime. The lowest metrics were MAE at 103.28 kW, RMSE at 141.87 kW, and MAPE at 7.51%. The MI methodology indicates that the Lag Load feature has a strong and consistent relationship with output throughout the year, demonstrating model continuity. The Yearly Hours feature can segment the load by time of year. The daytime and nighttime features can show more detailed time relationships, but they still don't yield results as good as using all nine features. For daily or one-day forecasting, the data structure requires more context than using only three features to allow the model to understand more complex details. Using only three features identifies the most important individual features. Operating alone or without the context of other features fails to recognize true relationships and can lead to higher forecasting errors.

4.10.4 Test Set Load Forecasting Performance

Table 4.5 presents the complete test dataset of the university load data, simulating the results for each of the six models. It also compares simulation cases using the top three most important features from the SHAP method and the MI method, showing the results of the overall test metrics. It was found that the model with the lowest MAE, RMSE, and MAPE metrics was the one selected using the top three most important features from the SHAP method, simulated with the Bi-LSTM model. The MAE was 177.90 kW, the RMSE was 310.29 kW, and the MAPE was 15.59%. These results indicate that the model can efficiently learn the structure of the electricity load data by selecting the most relevant features that truly impact the model. The SHAP method selects features that directly affect the model's output, thus eliminating some less relevant features. Furthermore, the memory of the Bi-LSTM

model is not used for unnecessary features; instead, more weight is given to time-related features, as the model is sensitive to temporal dependency. This results in lower overall metric values compared to other models and methods. The graphs showing the complete test suite for each model are presented in Appendix C.

Table 4.5 Overall Test Set Performance Metrics

Model	Test Set		
	MAE (kW)	RMSE (kW)	MAPE (%)
Attention-LSTM	183.72	323.18	16.68
Bi-LSTM	184.05	321.23	17.53
CNN-LSTM	195.24	332.14	19.21
LSTM	200.78	326.22	20.47
Moving Average	260.56	465.05	20.58
DNN	207.97	330.07	18.23
Bi-LSTM (Top 3 Features of SHAP)	177.91	310.29	15.59
Bi-LSTM (Top 3 Features of MI)	221.87	383.06	19.77

4.10.5 SHAP-Based Feature Importance Analysis

The SHAP tool was imported to describe the importance of features that affect the decision of the Bi-LSTM model, respectively, as shown in Fig. 4.10 The results of the remaining models are shown in the Appendix D. The x-axis shows the SHAP value, with a positive value indicating a higher prediction value. The y-axis sorts the features by cluster importance from highest to lowest, and the color of the dots indicates the actual feature value, with red having a high value and blue having a low value. The top three features of the NN model with the highest importance are as follows: The Lag Load feature is an important variable that has a great impact on the model's decision because it has the highest SHAP value distribution. There are a lot of red dots to the right, meaning that if yesterday's load is higher, it will also push today's

forecast value higher. The Yearly Hours feature is distributed in clusters and has most red dots to the left, meaning that features with high values will lower the forecast value. The Weekdays and Weekend features show that the closer the SHAP value distribution is to zero, the less the feature's impact on the model's decision. In the non-linear features, the SHAP values are widely distributed, with both positive and negative values and a mixture of red and blue dots at the same point. The moving average method is a basic statistical method for forecasting using features Lag 1, Lag 2, and Lag 3 as the electrical load values for the past 1, 2, and 3 days, respectively. The time-related data, namely the Hour and Minute features, are the time data for all 3 days in every hour and minute at the same time of each day. Feature Lag 1 is the feature with the most dispersion and has the most impact on the model's decision.

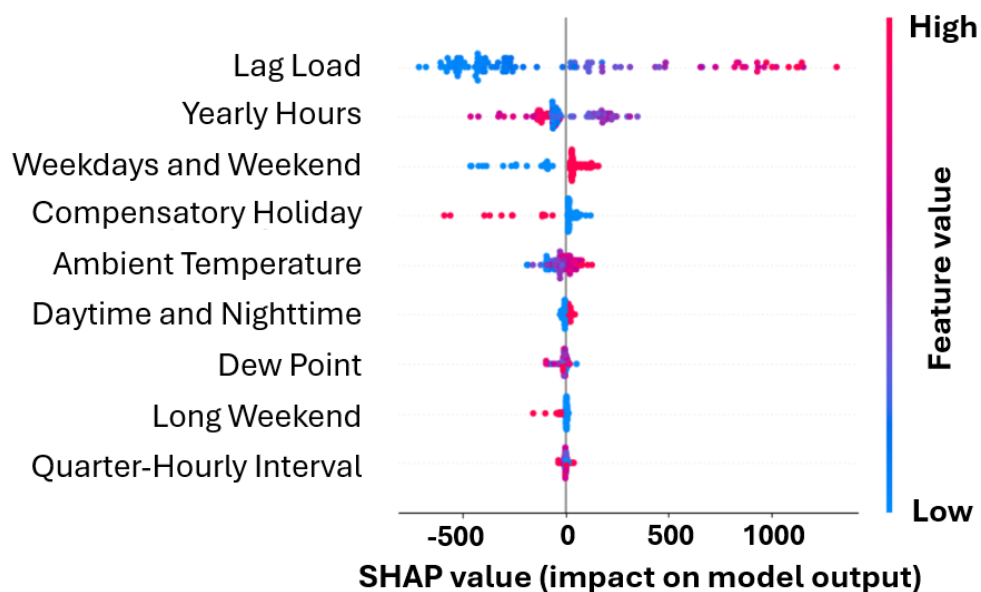


Figure 4.10 SHAP Value of Each Model: Bi-LSTM

4.10.6 Comparative Analysis of Feature Importance Models and Methods

The importance of ranking of features of NN models by SHAP and MI, where model A is LSTM, B is Bi-LSTM, C is CNN-LSTM, D is Attention-LSTM, E is DNN, and F is MI. From Table 4.6, number 1 shows the ranking of the most important features, and the higher the ranking number, the importance will decrease until

number 9, which has the lowest feature importance to the model. The Bi-LSTM model forecasts have the highest accuracy. The ranking of features from the highest to the lowest is shown as follows: Lag Load, Yearly Hours, Weekdays and Weekend, Compensatory Holiday, Ambient Temperature, Daytime and Nighttime, Dew Point, Long Weekend, and Quarter-Hourly Interval, respectively, were compared for all 6 models. The Bi-LSTM and LSTM models are models that learn the time sequence of data. The 4th feature is Compensatory Holiday, which shows the compensatory holidays of the university reflect the overall electricity consumption behavior. There are red and blue dots clearly separated from the blue ones. The model can recognize clear patterns with this data and associate it with the load drop. This differs from the fifth-ranked feature, ambient temperature. Since alternating points in the same location have a non-linear relationship, fluctuations can make it more difficult for models to capture these patterns.

Table 4.6 Feature Importance Ranking of NN Model by SHAP and MI

Feature	Feature importance ranking					
	A	B	C	D	E	F
Lag Load	1	1	1	1	1	1
Yearly Hours	2	2	2	2	2	2
Weekdays and Weekend	3	3	3	3	3	6
Compensatory Holiday	4	4	5	5	5	7
Ambient Temperature	5	5	4	4	4	4
Daytime and Nighttime	9	6	7	8	6	3
Dew Point	6	7	6	6	7	5
Long Weekend	8	8	8	7	8	8
Quarter-Hourly Interval	7	9	9	9	9	9

A CNN-LSTM model, which captures short-term and long-term patterns well, or an Attention-LSTM model that emphasizes time steps, will better understand this feature. Based on feature priorities, CNN-LSTM and Attention-LSTM models prioritize ambient temperature over compensatory holidays. DNN models can easily learn by finding association weights, do not remember event sequences, and process data without considering time sequences. The SHAP value of the Ambient Temperature feature, used to describe load changes, is higher than that of the Compensatory Holiday feature, which relies on time sequences and previous behaviors. The Quarter-Hourly Interval feature, also a later feature, is considered a fundamental feature, as it does not directly correlate with load and is a simple feature for the model to learn. From the simulation results, it was found that the MI method gives more importance to the Daytime and Nighttime, Ambient Temperature, and Dew Point features of each model than the SHAP method because these features have more raw data statistics related to the forecasting target, which reflects the actual behavior of electrical loads without learning the model mechanism and not understanding the relationship between the features like SHAP. When using SHAP, the importance is reduced. For Weekdays and Weekend and Compensatory Holiday features, which have lower MI scores, SHAP gives higher importance to both features. This is because the model learns the effect of these features through the more important Lag Load feature, not just learning the features alone. Using SHAP, one can learn features with complex relationships better.

The third feature of the MI method, Daytime and Nighttime, is shown to be highly correlated with load. However, compared to the SHAP method, this feature is ranked low due to the model's inability to extract patterns well. However, the key features, Lag Load and Yearly Hours, ranked first and second in both methods, showed consistent results, indicating that this feature set can definitely improve accuracy and efficiency. Using load data, the MI method screens out potentially important features related to electrical loads, screening relevant and related data before importing into

the model and isolating truly irrelevant features. The trend in importance of features is evident, eliminating bias from the model for optimal forecasting performance. However, since it is not yet possible to determine how features affect model decisions, SHAP values are used at each data point in real-world situations to explain the relationship between features and model decisions. This ensures that the model is indeed using those features and provides a deeper explanation of model behavior, demonstrating the importance of features that directly influence the model. Furthermore, under different load conditions, the direction of their impact on forecasting can be identified. Regarding the reliability of black-box models, this can be explained.

4.11 Simulation Result and Discussion with ISO New England Data

In electricity load forecasting, understanding the context of the data and the influence of features on the results of various models is crucial. Improving forecasting performance and model description are key issues, as are the repeatability and reliability of the research framework. Therefore, electricity load forecasting was simulated using hourly electricity demand data from January 3, 2003, to December 31, 2014, compiled and published by ISO New England (ISO-NE), an independent agency responsible for regulating and managing electricity systems in the New England region of the United States, as an example to illustrate the analysis of the models and methods proposed above. Using Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models.

4.11.1 Mutual Information Scores of ISO-NE Data

Prioritizing feature data before importing them into the model using the MI tool measures the correlation between features and dip data sets of electricity demand, compiled and published by ISO New England (ISO-NE) of the United States. Both linear and non-linear data are used, showing the ranking scores of each feature. The Lag Load feature has an MI score of 0.977, Yearly Hours 0.487, Ambient

Temperature 0.386, Month 0.171, Weekdays 0.0516, Year 0.023, and Day 0.011, as shown in Fig. 4.11. It can be seen that including lag load as a feature shows a significant correlation between input and output data. The characteristics of the Lag Load feature are most clearly related to the forecast values, with electricity user behavior today and yesterday being very similar. The features with the lowest scores are yearly hours, which show recurring time patterns throughout the year and demonstrate a high correlation with the predicted values. These two features show significantly higher scores than other models. Features with scores close to 0 show decreasing correlation with the predictions in descending order.

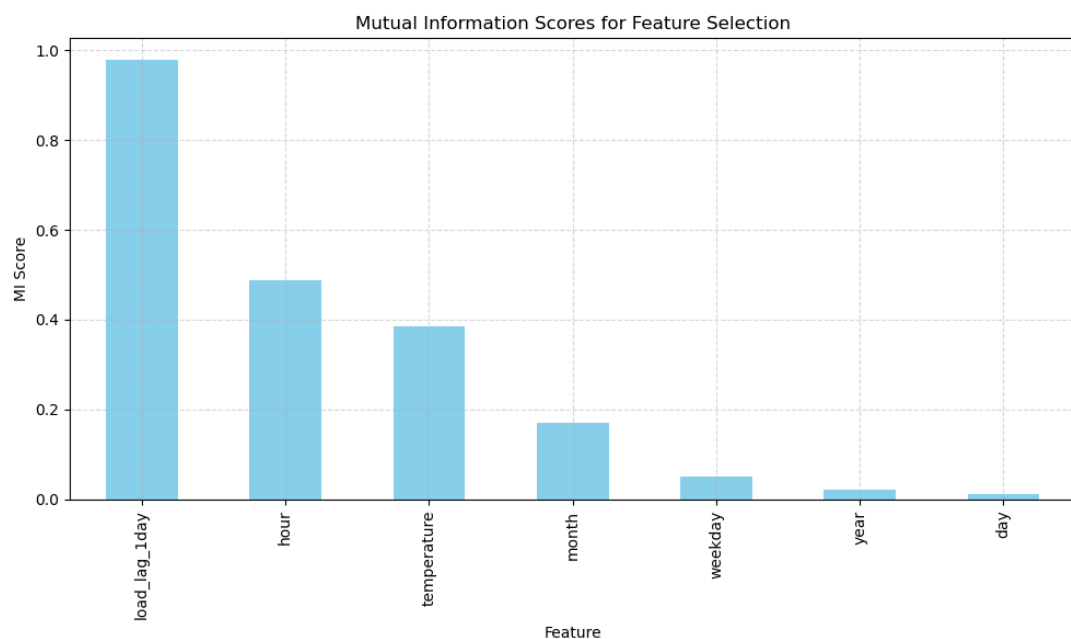


Figure 4.11 Mutual Information Scores of ISO-NE Data

4.11.2 Daily Load Forecasting Performance

The electricity load forecasts compiled and published by ISO New England (ISO-NE) one day in advance can show the performance of each of the six models through three indicators. These are compiled as statistical metrics indicating the lowest values for each model from the tested data. Key forecast dates, including

those showing the lowest values, are also included in the test dataset, as shown in Table 4.7. The forecast results show that the CNN-LSTM method yielded the lowest values for all three indicators from the tested dataset, with MAE at 89.08 kW, RMSE at 118.94 kW, and MAPE at 0.67% on October 9, 2014, making it the most accurate model.

Table 4.7 The best result for each model in daily forecasting

Models	Important dates	Metric Statistic		
	Date of Minimum	MAE (kW)	RMSE (kW)	MAPE (%)
Moving Average	2014-09-25	98.40278	142.3815	0.71
Attention-LSTM	2014-06-10	108.78	137.65	0.74
DNN	2014-07-19	238.51	281.87	1.75
LSTM	2014-10-21	90.73	126.32	0.69
Bi-LSTM	2014-06-05	96.84	123.52	0.70
CNN-LSTM	2014-10-02	89.08	118.94	0.67
CNN-LSTM (Top 3 features SHAP and MI)	2014-10-09	148.25	203.14	1.13

Fig. 4.12 shows a flat graph comparing actual load and other models to illustrate forecast trends for that day. The figure shows three consecutive 1-day load forecasts. The LSTM model yielded the lowest values: MAE 90.73 kW, RMSE 126.32 kW, and MAPE 0.69% on October 21, 2014. The Bi-LSTM model yielded the lowest values: MAE 96.84 kW, RMSE 123.52 kW, and MAPE 0.70% on June 5, 2014. The Moving Average model yielded the lowest values: MAE 98.40 kW, RMSE 142.38 kW, and MAPE 0.71% on date missing. On September 25, 2014, the forecasting method using the Attention-LSTM model yielded the following lowest metric values: MAE at 108.78 kW, RMSE at 137.65 kW, and MAPE at 0.74% on June 10, 2014. The forecasting method using the

DNN model yielded the following lowest metric values: MAE at 238.51 kW, RMSE at 281.87 kW, and MAPE at 1.75% on July 19, 2014. The models display the metric results from lowest to highest values, respectively.

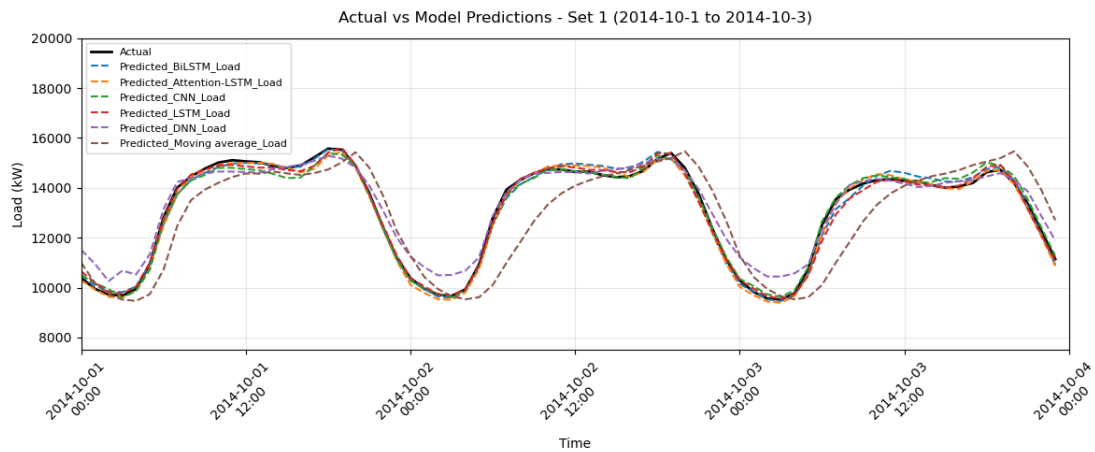


Figure 4.12 Comparison of the best forecasting results of the CNN-LSTM model

The results show that the CNN-LSTM hybrid model is the most accurate for 1-day forecasting. This is because the forecasted data contains recurring daily load changes over short forecast periods. The CNN model analyzes the similarity of these short-term patterns before passing them to the LSTM model to learn longer-term patterns, resulting in the lowest forecast metric. The LSTM model, which is less accurate, primarily learns from time-series memory and performs well on relatively simple datasets. Bi-LSTM models typically learn both past and future patterns, indicating that future pattern learning is not very helpful for this type of dataset. Moving average models, acting as baselines, only learn from historical electricity load data and therefore cannot understand more complex data. Attention-LSTM models can learn from time series datasets that span many steps ahead and are more complex than basic models. However, using them for 1-day forecasting on a fixed-pattern dataset would make the model too complex and unsuitable for the data. The creation of excess parameters leads to worse results compared to other models, and basic DNN

models show the least accurate results because they lack a temporal understanding of forecasting. At each time point, the model does not understand the temporal relationships between features, which is a crucial issue for electricity load forecasting. When there are many parameters used in forecasting, the model is prone to overfitting.

Furthermore, a re-simulation of the most accurate CNN-LSTM model was performed using the top three most influential features identified from SHAP and MI: Lag Load, Yearly Hours, and Ambient Temperature, respectively. Analysis of the data and model, along with statistical methods, indicated that historical load shows the most significant pattern in electricity consumption and has the greatest influence on usage. This is followed by temporal features reflecting important daily patterns, and weather factors tend to reflect electricity usage based on temperature. The lowest values were recorded on October 9, 2014: MAE at 148.25 kW, RMSE at 203.14 kW, and MAPE at 1.13%. The simulation results showed that while the three identified features are highly significant, using only these three results in less accurate forecasting compared to using all features. This is because the model learning process requires diverse datasets and contexts to effectively identify relationships and achieve optimal performance.

4.11.3 Test Set Load Forecasting Performance

Table 4.8 presents the complete test dataset, simulating each of the six models and comparing cases simulating the top three most important features from the SHAP and MI methods with the CNN-LSTM model. The results show the indicators of the entire test set. The model with the lowest MAE is CNN-LSTM, with an MAE value of 333.99 kW. The model with the lowest RMSE is Attention-LSTM, with an RMSE value of 475.15 kW. And the model with the lowest MAPE is CNN-LSTM, with a MAPE value of 2.29%. These results indicate that the CNN-LSTM model is the most stable and accurate, predicting closest to the true values with low MAE (low mean error per data point) and low MAPE (low percentage error across the load range). The Attention-LSTM

model has the lowest RMSE, indicating its ability to control errors occurring during high load periods. This demonstrates the good predictive ability of the entire test dataset during peak load periods. The graphs showing the complete test suite for each model are presented in Appendix C.

Table 4.8 Overall Test Set Performance Metrics of ISO-NE Data

Models	Test Set		
	MAE (kW)	RMSE (kW)	MAPE (%)
Attention-LSTM	334.5389	475.1453	2.32
Bi-LSTM	368.1364	534.2110	2.50
CNN-LSTM	333.9936	497.3806	2.29
LSTM	358.5621	516.5337	2.45
Moving Average	1129.35	1601.82	7.67
DNN	611.4789	824.5546	4.22
CNN-LSTM (Top 3 features SHAP and MI)	625.95	848.3157	4.34

4.11.4 SHAP-Based Feature Importance Analysis

SHAP is used to explain the importance of features affecting the decision-making of Moving Average, DNN, LSTM, Bi-LSTM, CNN-LSTM, and Attention-LSTM models, respectively, as shown in Appendix D. The top three most important features in NN models are Lag Load, shown in Fig. 4.13, a key variable significantly impacting model decision-making due to its largest SHAP distribution (shown with red dots close to the right and left, indicating that higher yesterday's load will raise today's forecast, and lower yesterday's load will lower today's forecast); Yearly Hours, which is clustered with purple dots concentrated on the right, suggesting that mid-range values improve forecasts; and Ambient Temperature, where the purple dots are clustered on the left in SHAP distribution, meaning that mid-range values (neither too high nor too low) decrease forecasts. The weekday, month, day, and year features show a SHAP

distribution closer to zero. The impact of features on model decision-making decreases with increasing frequency. The moving average method is a basic method that uses statistical values for forecasting, employing Lag 1, Lag 2, and Lag 3 features as historical electrical load data for the past 1, 2, and 3 days, respectively. The time-related data, the Hour feature, represents hourly data for all three days at the same time each day. The Hour feature is the most widely distributed and has the greatest impact on model decision-making.

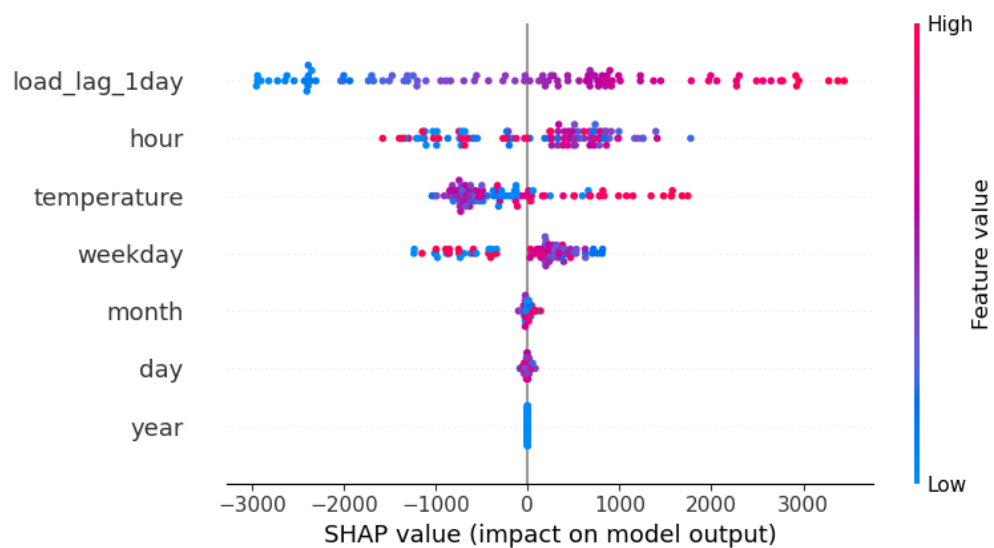


Figure 4.13 SHAP Value of Each Model of ISO-NE Data: CNN-LSTM

4.11.5 Comparative Analysis of Feature Importance Models and Methods

Table 4.9 shows the priority of features from highest to lowest, with increasing priority as the number increases, down to feature number 9, which has the lowest importance to the model. CNN-LSTM models show the highest forecasting accuracy. Based on SHAP analysis, the priority of features from highest to lowest is as follows: Lag Load, Yearly Hours, Ambient Temperature, Weekday, Month, Day, and Year. A comparison of the six models is shown. LSTM and Attention-LSTM models have similar priorities. Bi-LSTM models have a similar priority, with features 2 (temperature) and 3 (hourly) swapped. Because the models learn in two directions within the same

sequence, there is temporal continuity, and the temperature feature exhibits this continuity, giving it a higher priority in Bi-LSTM. The hourly feature, however, does not translate according to time but simply indicates the position of a 24-hour cycle. DNN models have a similar priority, with features 5 (Day) and 6 (Month) swapped, as temporal continuity is a factor. The model does not recognize the time sequence and directly affects the load. However, the Day feature clearly impacts electricity user behavior, making it easy for the model to grasp the relationship, thus ranking higher in DNN models.

Table 4.9 Feature Importance Ranking of ISO-NE Data

Feature	Feature importance ranking of NN model by SHAP and MI					
	LSTM	Bi-LSTM	CNN-LSTM	Attention-LSTM	DNN	MI
Lag Load	1	1	1	1	1	1
Yearly Hours	2	3	2	2	2	2
Ambient Temperature	3	2	3	3	3	3
Weekday	4	4	4	4	4	5
Month	5	5	5	5	6	4
Day	6	6	6	6	5	7
Year	7	7	7	7	7	6

Based on the MI analysis, the feature priority is similar to that of the CNN-LSTM model by SHAP, but with some positional changes. Feature position 4 represents the Month feature, and feature position 5 represents the Weekday feature. Feature positions 6 represent the Year feature, and 7 represents the Day feature. The MI method views the overall statistical relationship of the data, prioritizing features that result in the lowest forecasting error. The Month feature highlights different seasons, and since the MI method considers the overall year and therefore has lower

error margins, it is more important than the Weekday feature, which recurs over short periods. Similarly, the Year and Day features are given higher priority by the MI method because they can better separate load groups over longer periods.

4.12 Summary of Results

This research proposes the development of forecasting techniques using deep learning. The accuracy of the models was compared to underlying neural network (NN) models. Bi-LSTM and CNN-LSTM models showed the highest accuracy and performance for forecasting daily electricity consumption. Data collected, after cleaning and feature generation, was transformed into features ready for input into forecasting models. Electricity consumption forecasts were most consistent with actual values. Overall, the Bi-LSTM model with feature selection using the SHAP method, Attention-LSTM, and CNN-LSTM provided the most accurate forecasts and also yielded the lowest average metrics among all tested datasets, indicating the models' potential for forecasting in various contexts.

Furthermore, the model reliability investigation indicates that transparent feature analysis and model description using MI can explain the relationships between raw data and electricity loads and filter static data for inspection before inputting the model. This reduces the number of features required for model input and reduces training time. SHAP provides in-depth model descriptions, demonstrating the actual use of the model's features, understanding complex features, and increasing model reliability because all features have underlying relationships that can be demonstrated. Using this combined approach, comprehensive coverage is achieved. It provides accurate forecasting results and can also provide causal explanations.