

Multi-Horizon Machine Learning Model for Home Energy Consumption Prediction Considering Appliances Usage

Irrine Budi Sulistiawati*, Aryunto
Soetedjo and Irmalia Suryani
Faradisa

Department of Electrical
Engineering, National Institute
of Technology (ITN), Malang,
Indonesia

*Email: irrine@lecturer.itn.ac.id

Received for publication
December 06 2025.

Abstract

Home appliance energy consumption increases as technology penetration in home appliances increases. Predicting home energy consumption is a critical task in home energy management, including load scheduling, improving energy efficiency, and reducing carbon emissions. This paper proposes a multi-horizon prediction of home appliance power consumption using machine learning techniques. The proposed approach exposes both individual appliance and total power consumptions, and the impact on improving the prediction accuracy. Using appliance-level data collected from an IoT-based SCADA monitoring system, the tree-based models (Random Forest, Light Gradient Boosting Machine, and eXtreme Gradient Boosting) and deep learning models (Recurrent Neural Network, Long Short-Term Memory, and Gated Recurrent Unit) are evaluated across four prediction horizons: 1 min, 15 min, 1 hr, and 1 day. The evaluation results show that no single algorithm achieves the best performance across appliance types and time horizons. Thus, the appliance-specific, horizon-dependent model selection strategy is essential for optimizing prediction accuracy in home energy management systems. The proposed approach provides a valuable finding: incorporating all individual appliance power measurements, along with total power, as model inputs improves total power prediction performance.

Keywords

multi-horizon, machine learning, energy consumption, prediction, home appliance

1. Introduction

Recently, the electricity consumption in residential buildings has become more complex due to the penetration of renewable energy, such as solar panels, and the increased use of electrical appliances resulting in a significant increase in household power consumption. [1]. In such systems, predicting home energy consumption plays a critical role in home energy management, where it can be used for load scheduling, improving energy efficiency, and

reducing carbon emissions. An accurate prediction model allows effective control of home appliances and energy resources. It plays a key role as an input to the home energy management system (HEMS) for energy optimization [2, 3].

Basically, the load prediction can be divided into four categories, with respect to the time horizons [1, 4]:

1. Very-short-term load forecasting (VSTLF): Power is predicted over a range of 1 min to 1 hr, making it suitable for fast response and real-time operation.

2. Short-term load forecasting (STLF): Power is predicted over a range of 1 hr to 1 week, making it suitable for short-term planning and daily operation.
3. Medium-term load forecasting (MTLF): Power is predicted over a range of 1 week to 1 year, making it suitable for medium-term planning and seasonal load adjustment.
4. Long-term load forecasting (LTLF): Power is predicted beyond 1 year, for long-term and strategic planning.

The short-term home appliance energy consumption prediction was proposed in [4], where time intervals of 10 min and 15 min were used for 1-day-ahead prediction. The work in [5] used a 15-min time interval. In [6], 1-hr data was used to predict the next 1-hr energy consumption. The hourly and daily energy consumptions of the residential buildings were proposed in [7]. The residential load predictions for 6-hr, 12-hr, and 24-hr time horizons were proposed in [8]. In [9, 10], the performances of household energy consumption predictions under different time horizons: VSTLF (every hour), STLF (every day and week), and MTLF (every month and quarter) were studied.

Recently, most studies have employed machine learning techniques to predict household energy consumption. Prediction techniques can be divided into four major categories: statistical models, traditional machine learning-based methods (including tree-based methods), deep learning-based methods, and hybrid or ensemble approaches. The artificial neural network (ANN) improved with the behavior model was proposed to predict the next day's hourly energy consumption [11]. The combination of weather and power data was used to train the ANN to predict the power consumption [12]. The traditional and deep learning-based machine learning techniques were employed to predict the appliances and total power consumption by considering the weather data [13].

The long short-term memory (LSTM), a powerful deep learning method for handling time-series data, is widely used in energy consumption prediction. In [14], the LSTM was employed to predict the energy consumption of the common home appliances in the HEMS. The multivariate LSTM that integrates the weather features was proposed in [15]. The load prediction was improved by integrating the LSTM with human behavior recognition [16]. The LSTM was used to predict appliance energy based on temperature, humidity, wind speed, dew point, and weight factors [17]. A cloud-assisted deep learning model

using multi-attribute convolutional long short-term memory was proposed to predict short-term energy consumption drawn by home appliances [18].

The LSTM, recurrent neural network (RNN), and gated recurrent unit (GRU) were employed to predict the energy consumption and environment parameters [19]. The combination of convolutional neural network (CNN) and LSTM was used to improve home consumption prediction accuracy [20], where the CNN was used to extract spatial features, while the LSTM was used to extract temporal features. The integration of CNN with a BiLSTM and an attention mechanism was proposed in [21] to improve the energy consumption model's handling of spatial and temporal features.

In [22], the energy consumption models ensemble the three statistical predictions, namely auto regressive moving average, auto regressive integrated moving average (ARIMA), and seasonal ARIMA (SARIMA) with ANN to improve the prediction accuracy. The fusion of statistical prediction, seasonal autoregressive integrated moving average with exogenous variables (SARIMAX) and machine learning techniques (random forest [RF], LSTM, extreme gradient boosting [XGB]) were employed to predict the energy load accurately [23] by utilizing the SARIMAX for dealing with time-series trends, RF and XGB for capturing non-linearity, and LSTM for capturing temporal dependencies.

Although many studies have examined residential energy consumption extensively, only a limited number explicitly discusses the detailed appliance usage information, despite its dominant influence on residential load variability. Furthermore, no research has yet analyzed the effect of individual load power consumption on the accuracy of predicting total power consumption. This research addresses these gaps by proposing a prediction framework that exposes both individual appliance and total power consumptions, and the impact on improving the prediction accuracy. To provide a comprehensive study, the comparisons of multi-horizon predictions and different machine learning methods are investigated.

The main contributions of the proposed approach are:

1. It considers both individual appliance power consumption prediction and total power prediction.
2. It examines the impact of individual appliance power consumption on the prediction performance of total power consumption, where individual appliances are divided into groups based on operational time flexibility for better analysis.

3. It compares the performance of predictions across the different time horizons: 1 min, 15 min, 1 hr, and 1 day.
4. It evaluates the performance of tree-based machine learning prediction techniques (RF, light gradient boosting machine [LGBM], XGB), which are suitable for non-sequential data, and deep learning-based prediction techniques (RNN, LSTM, GRU), which are suitable for sequential and time-series data.

The rest of the paper is organized as follows. Section II presents the research method. Section III presents the results and discussion. Section IV covers the conclusion.

II. Research Method

a. Proposed prediction models

The architectures of the proposed models are illustrated in Figures 1–5. It consists of five schemas with respect to the types of power prediction and input feature groups, as follows:

1. Scheme-1: The individual appliance power prediction.
2. Scheme-2: Total power prediction with Feature Group-A (water heater [WH] power, washing machine [WM] power, rice cooker [RC] power, microwave oven [MO] power, total power).

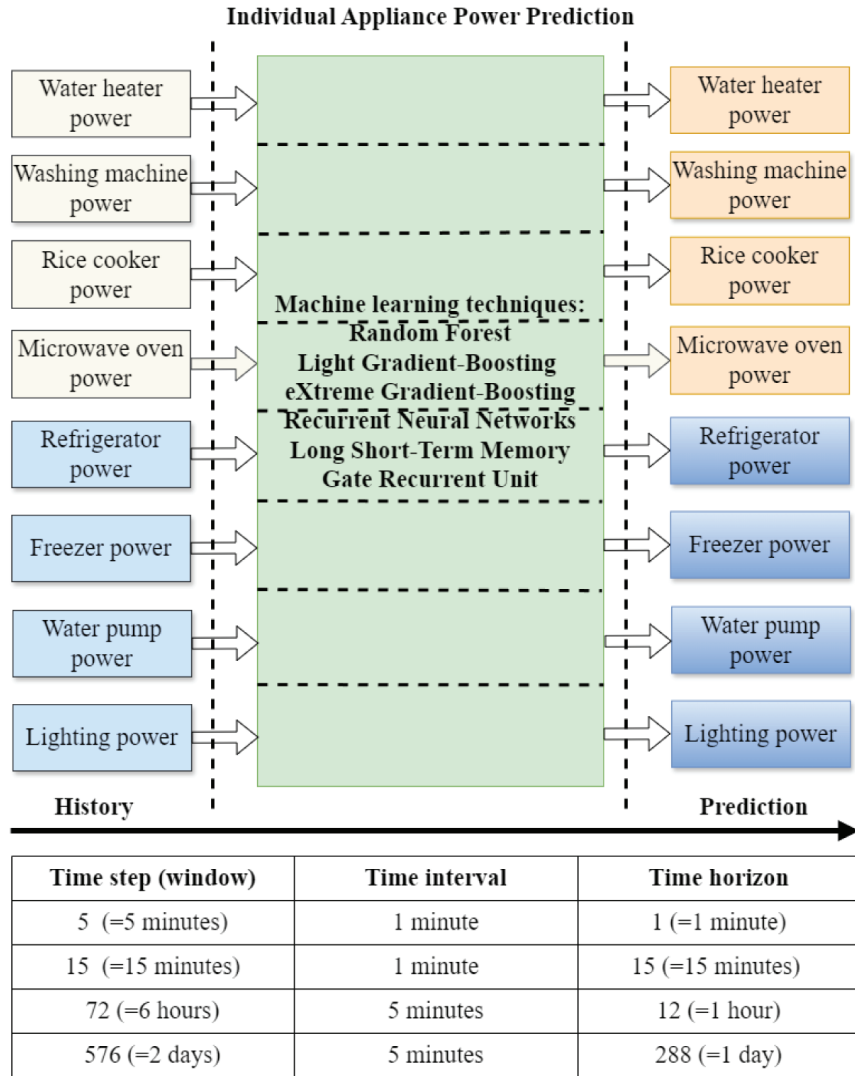


Figure 1: Architecture of Scheme-1. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

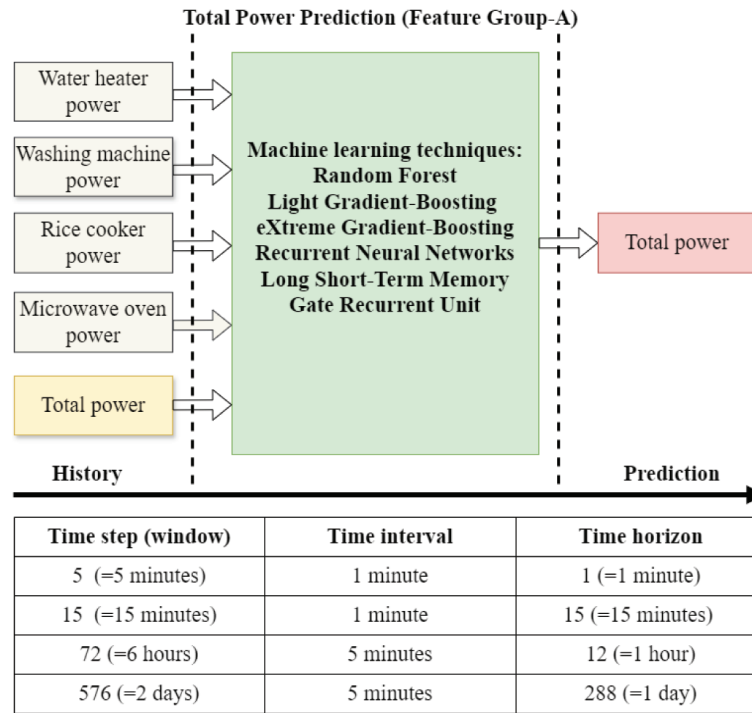


Figure 2: Architecture of Scheme-2. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

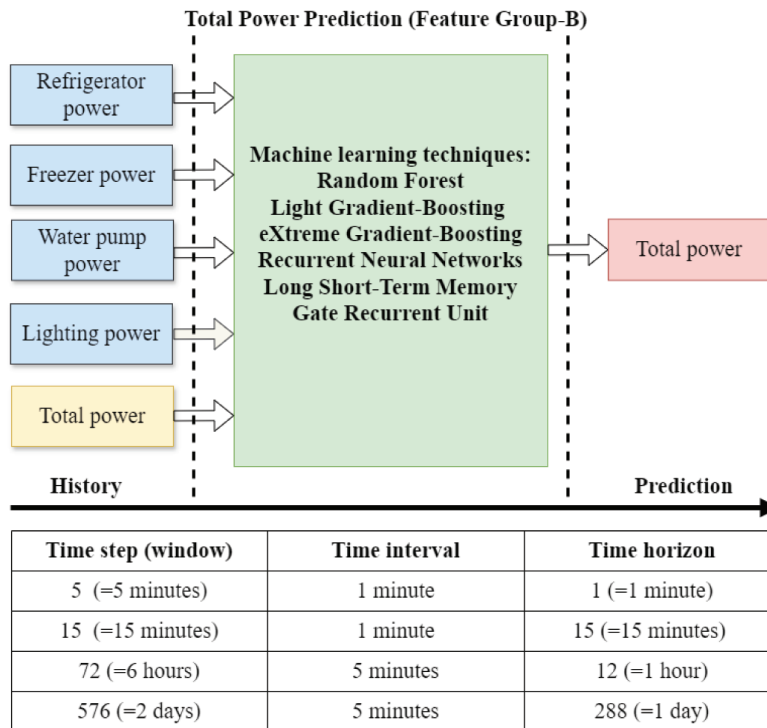


Figure 3: Architecture of Scheme-3. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

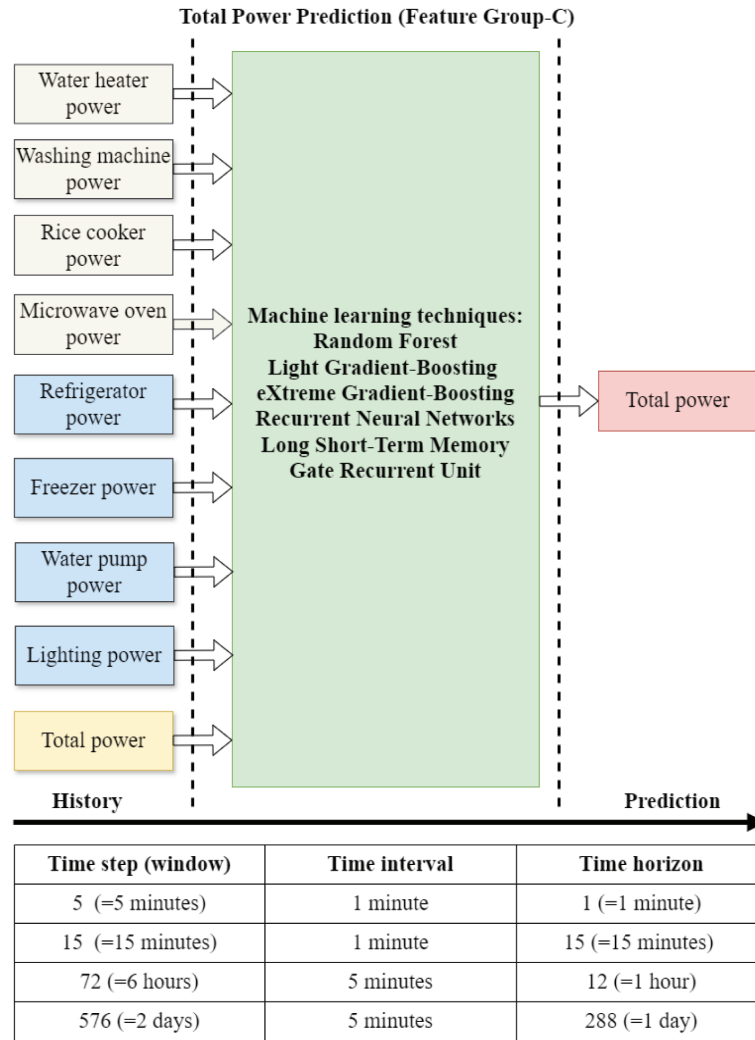


Figure 4: Architecture of Scheme-4. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

3. Scheme-3: Total power prediction with Feature Group-B (refrigerator [RFG] power, freezer [FZ] power, water pump [WP] power, lighting [LG] power, total power).
4. Scheme-4: Total power prediction with Feature Group-C (water heater power, washing machine power, RC power, MO power, RFZ power, FZ power, WP power, lighting power, total power).
5. Scheme-5: Total power prediction with Feature Group-D (total power).

Scheme-1 is used to predict the individual appliance power at a future time based on its previous time power consumption. In this work, eight home appliances are considered, namely water heater, washing machine, rice cooker, MO,

refrigerator, freezer, WP, and lighting. It is worth noting that with the recent development of IoT technology, individual appliance power consumption monitoring can be implemented easily, as developed in [24, 25].

Scheme-2 to Scheme-5 is used to predict the total power at a future time based on the different input features at previous times. Scheme-2 uses Feature Group-A, consisting of the reschedulable loads WH power, WM power, RC power, MO power and the total power, as the input features. The reschedulable load is the load that the user can control its operating time. Thus, Scheme-2 can be used to examine the impact of occupant behavior when using appliances on the prediction of total power consumption.

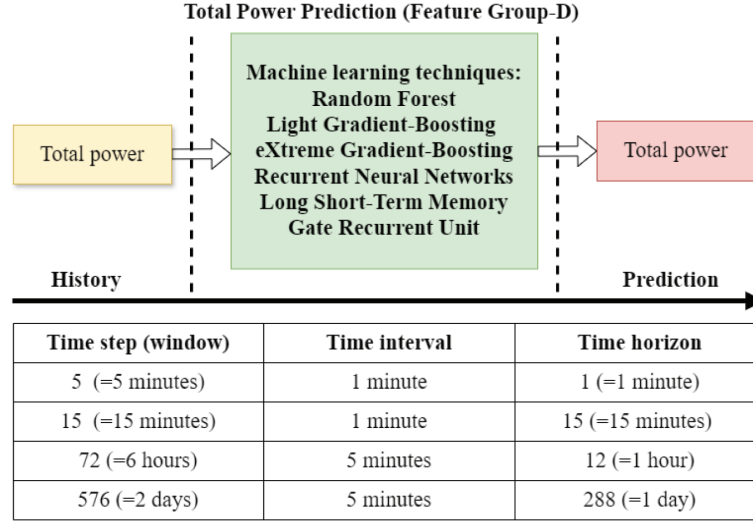


Figure 5: Architecture of Scheme-5. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

Scheme-3 uses Feature Group-B, consisting of the non-reschedulable loads RFG power, FZ power, WP power, LG power and the total power, as the input features. The non-reschedulable load is the load that the user cannot control its operating time. Thus, Scheme-3 can be used to examine the impact of the non-reschedulable appliances on the prediction of total power consumption.

Scheme-4 uses Feature Group-C, consisting of the reschedulable loads, non-reschedulable loads, and the total power, as the input features. Thus, Scheme-4 can be used to examine the impact of all individual appliances on the prediction of total power consumption.

Scheme-5 uses Feature Group-D, consisting solely of total power, as the input feature. Thus, Scheme-5 can be used to examine the performance of total power consumption prediction when using a single feature.

In each scheme, four different time horizons are examined, that is, (a) a 1-min time horizon with a 1-min time interval and a 5-min time step; (b) a 15-min time horizon with a 1-min time interval and a 15-min time step; (c) a 1-hr time horizon with a 5-min time interval and a 6-hr time step; (d) a 1-day time horizon with a 5-min time interval and a 2-day time step. These time-horizon variations fall under VSTLF and STLF. This selection is suitable for analyzing the real-time and hourly or daily operation of home appliances.

As shown in Figures 1–5, six prediction methods are evaluated in each scheme. The prediction techniques are divided into two categories: tree-based machine learning algorithms (RF, LGBM, and XGB) and deep learning-based algorithms (RNN, LSTM, and GRU). The algorithms are described briefly in the following section. The RF, LGBM, and XGB algorithms are usually used for predicting tabular, non-sequential data. Meanwhile, RNNs, LSTMs, and GRUs are commonly used for predicting sequential and time-series data. By evaluating both approaches, the suitability of the algorithm for predicting the short-term power consumption of individual appliances and the total power can be comprehensively analyzed.

b. Tree-based machine learning algorithms

The tree-based machine learning algorithms used in this work are RF, LGBM, and XGB. They are popular and influential tree-based algorithms used for prediction. RF is a tree-based ensemble machine learning technique developed by [26]. It uses trees as the base learners [27] denoted by $h_j(X, \Theta_j)$, where X represents the real-value input of predictor, Θ_j is a collection of random variables. The algorithm is as follows:

1. Prepare training set $TS = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where $x_i = (x_{i,1}, \dots, x_{i,p})^T$ is the p predictors, and y_i is the response;

2. For $j = 1$ to J (J is the number of regressors), repeat Step-3 to Step-4;
3. Pick a bootstrap sample TS_j of size N from the training set TS ;
4. Fit a tree using the bootstrapped sample TS :

- (a) Start with all observations on a single node
- (b) Randomly select the m predictors from the available p predictors
- (c) Find the best split from the m predictors
- (d) Split the node into two child nodes using the best split
- (e) Repeat (b) – (d) until the stopping criterion is met:
The prediction of the new point x is given by

$$\hat{f}(x) = \frac{1}{J} \sum_{j=1}^J \hat{h}_j(x). \quad (1)$$

where $\hat{h}_j(x)$ is the prediction at x using j^{th} tree.

The LGBM is a novel gradient boosting machine (GBM) algorithm [28] developed by [29]. It proposes the gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) to reduce the computation time. The GOSS reduces the number of samples by focusing on samples with larger gradients and performing random sampling on those with smaller gradients. At first, it sorts the data samples according to the absolute value of the gradients, and selects the highest $a \times 100\%$ samples. The algorithm then randomly selects $b \times 100\%$ samples from the rest of data. Finally, the information gain is calculated by amplifying the sampled data with small gradients

by a factor $\frac{1-a}{b}$. Meanwhile, the EFB is used to bundle exclusive features into a single feature, thus reducing the histogram complexity.

XGB [30] is the most popular implementation of the GBM [31, 32]. The algorithm is briefly described in the following. Suppose the dataset DS with m features and n number of samples is expressed as $DS = \{(x_i, y_i) : i = 1, \dots, n, x_i \in R^m, y_i \in R\}$. The predicted output $\hat{y}_i$ of an ensemble tree model is given by

$$\hat{y}_i = \varphi(x_i) = \sum_{k=1}^K f_k(x_i), f_k \in F, \quad (2)$$

where K is the number of trees, f_k denotes the k -th tree, and F is the space of regression trees. The above equation is solved by finding the best set of functions that minimize the following objective function:

$$L(\varphi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (3)$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|\omega\|^2 \quad (4)$$

where l is the loss function, Ω represents the complexity of the model, T is the number of leaf node, ω is the weight of the leaf node, γ and λ are the controlling factors. To optimize the above model Eqs (3) and (4), the model is trained by adding a new function f to minimize the following objective function

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (5)$$

The second-order Taylor polynomial can be used to replace the function f_t . Therefore, the objective function becomes

$$L^{(t)} \approx \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (6)$$

where

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \quad (7)$$

$$h_i = \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)}) \quad (8)$$

The constant terms in Eq. (6) can be removed to simplify the objective function as follows:

$$\tilde{L}^{(t)} = \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (9)$$

By expanding Ω , the objection function in Eq. (9) can be rewritten as

$$\tilde{L}^{(t)} = \sum_{i=1}^n \left[\left(\sum_{i \in I_j} g_i \right) \omega_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) \omega_j^2 \right] + \gamma T \quad (10)$$

where I_j is the instance set of leaf j . The optimal weight of leaf j is expressed by

$$\omega_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}, \quad (11)$$

and the corresponding optimal values is

$$\tilde{L}^{(t)} = - \frac{1}{2} \sum_{j=1}^T \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T. \quad (12)$$

c. Deep learning-based prediction techniques

The deep learning-based prediction techniques used in this work include RNNs, LSTMs, and GRUs. They are popular prediction algorithms suitable for time-series data. A RNN [33] is a type of ANN in which the output of a neuron is looped back to the same neuron in the next time step. Thus, it can memorize the previous input data. A RNN cell is illustrated in Figure 6, where the hidden layer state at time t (h_t) is expressed as

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_t) \quad (13)$$

where W_{hh} is the weight matrix for the hidden-to-hidden connection, h_{t-1} is the hidden layer state at time

step $t-1$, W_{xh} is the weight matrix for the input to hidden connection, x_t is the vector input at time step t , and b_t is the bias for the hidden layer.

The main drawbacks of RNN are the vanishing and exploding gradient problems, in which the gradient becomes too small (vanishing) or too large (exploding). To overcome these problems, the LSTM was proposed by [34]. The LSTM cell is depicted in Figure 7. It consists of an input gate, a forget gate, and an output gate. The input gate is expressed using the following formula:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (14)$$

where σ is a sigmoid function, W_i is the weight matrix for the input gate, b_i is the bias for the input

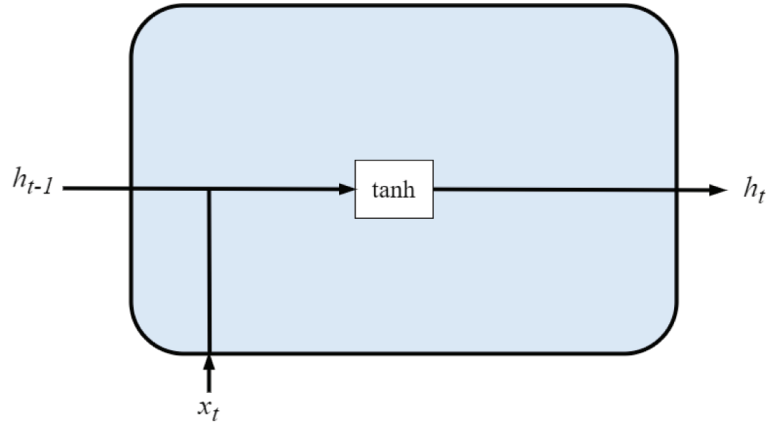


Figure 6: RNN cell. RNN, recurrent neural network.

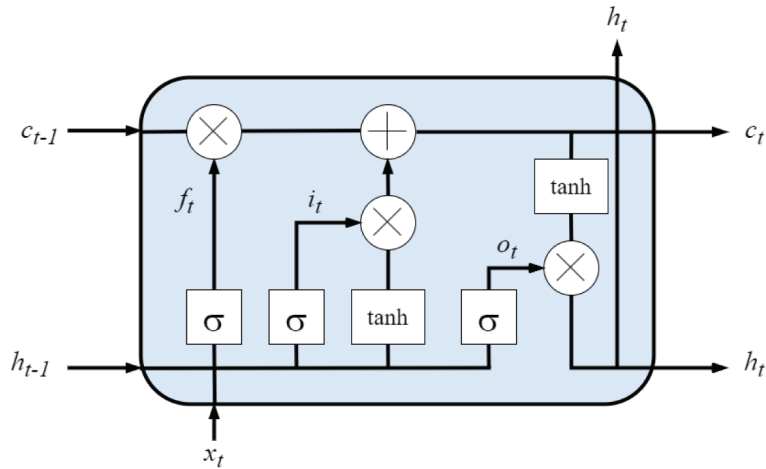


Figure 7: LSTM cell. LSTM, long short-term memory.

gate. The forget gate is expressed using the following formula:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (15)$$

where W_f is the weight matrix for the forget gate, b_f is the bias for the forget gate. The output gate is expressed using the following formula:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (16)$$

where W_o is the weight matrix for the output gate, b_o is the bias for the output gate. The cell state is expressed using the following formula:

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (17)$$

where $*$ indicates the element-wise multiplication, $\tanh$ is a hyperbolic tangent function, W_c is the weight matrix for the cell state, b_c is the bias for the cell state. The current hidden state is expressed using the following formula:

$$h_t = o_t * \tanh(c_t) \quad (18)$$

The GRU proposed by [35] is a simplified version of LSTM that improves the performance with less training time. The GRU cell is depicted in Figure 8. The GRU has only two gates, namely the update and reset gates. The update gate is expressed using the following formula:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (19)$$

where W_z is the weight matrix for the update gate. The update gate is expressed using the following formula:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (20)$$

where W_r is the weight matrix for the reset gate. The hidden state is expressed using the following function:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t]) \quad (21)$$

where W_h is the weight matrix for the hidden state.

d. Algorithm implementation and evaluation metric

The prediction algorithms are implemented in Python, with machine learning libraries and frameworks employed. The RF is implemented using the scikit-learn library [36]. The LGBM is implemented using the LGBM library [37]. The XGB is implemented using the XGB library [38]. The RNN, LSTM, and GRU are implemented using the Tensorflow framework [39].

The performance of the prediction algorithms are evaluated using two metrics: Normalized mean absolute error (NMAE) and normalized root mean square error (NRMSE). The NMAE and NRMSE are expressed using the following formulas:

$$NMAE = \frac{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|}{y_{\max} - y_{\min}} \times 100\% \quad (22)$$

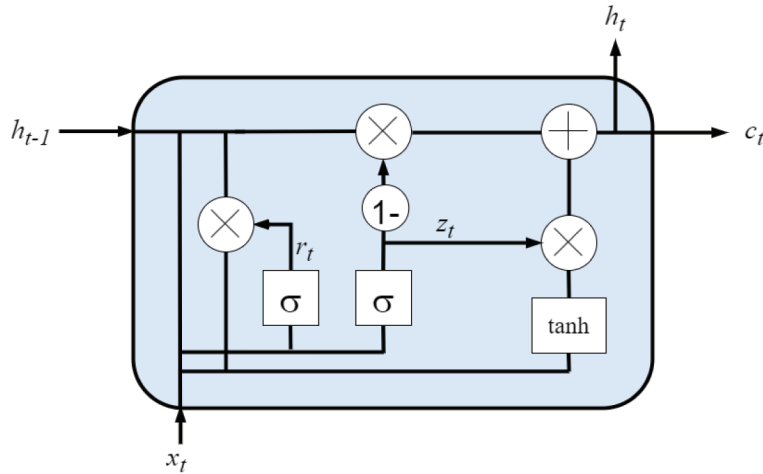


Figure 8: GRU cell. GRU, gated recurrent unit.

$$NRMSE = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}}{y_{\max} - y_{\min}} \times 100\% \quad (23)$$

where, $y, \hat{y}, n, y_{\max}, y_{\min}$ are the actual value, the predicted value, the number of samples, the maximum of the actual value, and the minimum of the actual value, respectively.

III. Results and Discussion

a. Dataset and machine learning parameters

The dataset was collected from a home using an IoT-based SCADA monitoring system, where individual appliances are measured by smart meters connected to the HMI SCADA wirelessly [25]. The HMI/SCADA system was equipped with local data storage and can be accessed remotely via the internet. It is worth noting that this configuration provided a simple, low-cost home appliance power-monitoring system.

The data were collected over a month, from August 1, 2025 to August 31, 2025, with a 1-min time interval. The dataset consisted of timestamps and power consumption for home appliances (e.g., WH, WM, RC, MO, RFG, FZ, WP, LG), as well as the total power consumption. The dataset is saved in the comma-separated value format. The dataset for 1 month contains 44,637 data points.

To ensure accurate and stable predictions, the dataset is cleaned and preprocessed, including handling missing values, outliers, and normalizing the data to a consistent scale. Then, the data are split into 0.75 for training and 0.25 for testing.

The machine learning parameters used in this work are listed in Table 1 for tree-based algorithms and in Table 2 for deep learning algorithms.

b. Evaluation of individual appliance power prediction

Figures 9–12 depict the NMAE of individual appliance power predictions for 1-min, 15-min, 1-hr, and 1-day time horizons. In each figure, the NMAEs for six methods are compared across eight appliances.

Table 1: Tree-based machine learning parameters

RF		LGBM		XGB	
Parameter	Value	Parameter	Value	Parameter	Value
No. of estimators	200	Boosting type	“gbdt”	Booster	“gbtree”
Criterion	“Squared error”	No. of leaves	31	Min. loss reduction (gamma)	0
Max. depth	None	Max. depth	−1	Max. depth	6
Min. samples split	2	Learning rate	0.05	Learning rate	0.05
Min. samples leaf	1	No. of estimators	500	No. of estimators	500
Max. features	1.0	Objective	“Regression”	Objective	“reg:squarederror”
Max. leaf nodes	None	Min. split gain	0	Min child weight	1
Bootstrap	True	Subsample ratio	0.8	Subsample ratio	0.8
		Subsample ratio of columns	0.8	Subsample ratio of columns	0.8
		Regularization alpha	0	Regularization alpha	0
		Regularization lambda	0	Regularization lambda	1

LGBM, light gradient boosting machine; RF, random forest; XGB, extreme gradient boosting.

Table 2: Deep learning-based machine learning parameters

Parameter	Value		
	RNN	LSTM	GRU
No. of units	64	64	64
Activation	"tanh"	"tanh"	"tanh"
Recurrent activation	-	"Sigmoid"	"Sigmoid"
Use bias	True	True	True
Kernel initializer	"glorot uniform"	"glorot uniform"	"glorot uniform"
Recurrent initializer	"Orthogonal"	"Orthogonal"	"Orthogonal"
Bias initializer	"Zeros"	"Zeros"	"Zeros"
Input shape	(WINDOW,1)	(WINDOW,1)	(WINDOW,1)
Dropout	0.2	0.2	0.2
Dense units	1	1	1

GRU, gated recurrent unit; LSTM, long short-term memory; RNN, recurrent neural network.

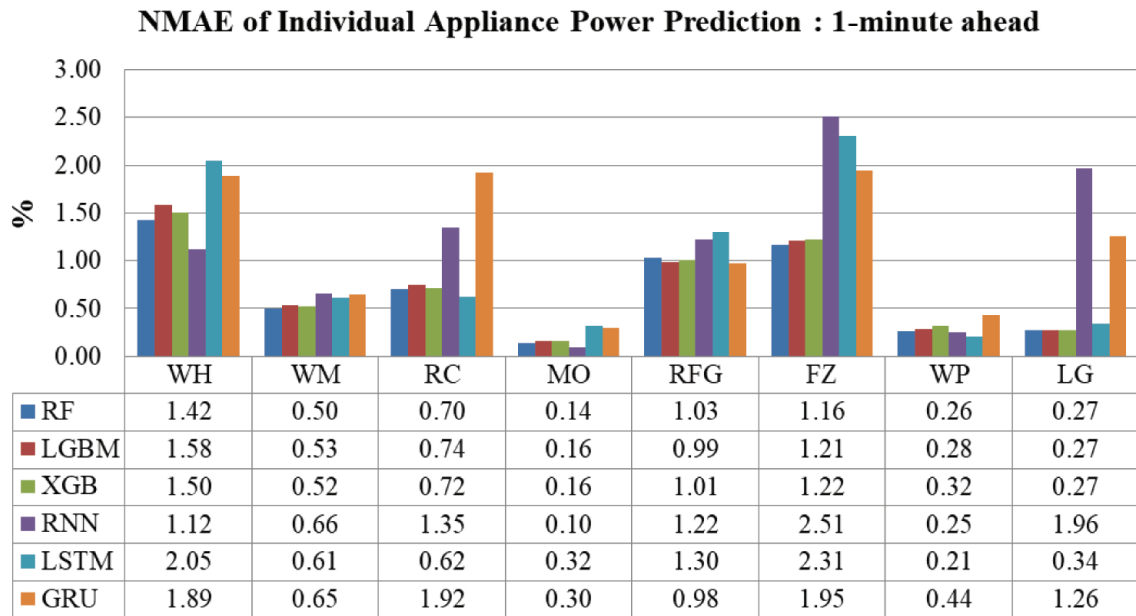


Figure 9: NMAE of 1-min ahead individual appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; WP, Water pump; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RC, rice cooker; RFG, refrigerator; WH, water heater; WM, washing machine.

To facilitate analysis, the comparison results are summarized in Table 3. Table 3 shows that the lowest NMAE of 0.09% is achieved by the MO using an RNN at a 15-min horizon for reschedulable load, and the lowest NMAE of 0.21% is achieved by the WP

using LSTM at a 1-min horizon for non-reschedulable load.

Table 3 shows that for reschedulable load, the MO achieves the lowest NMAE for all algorithm types and time horizons. At very short-term prediction, deep

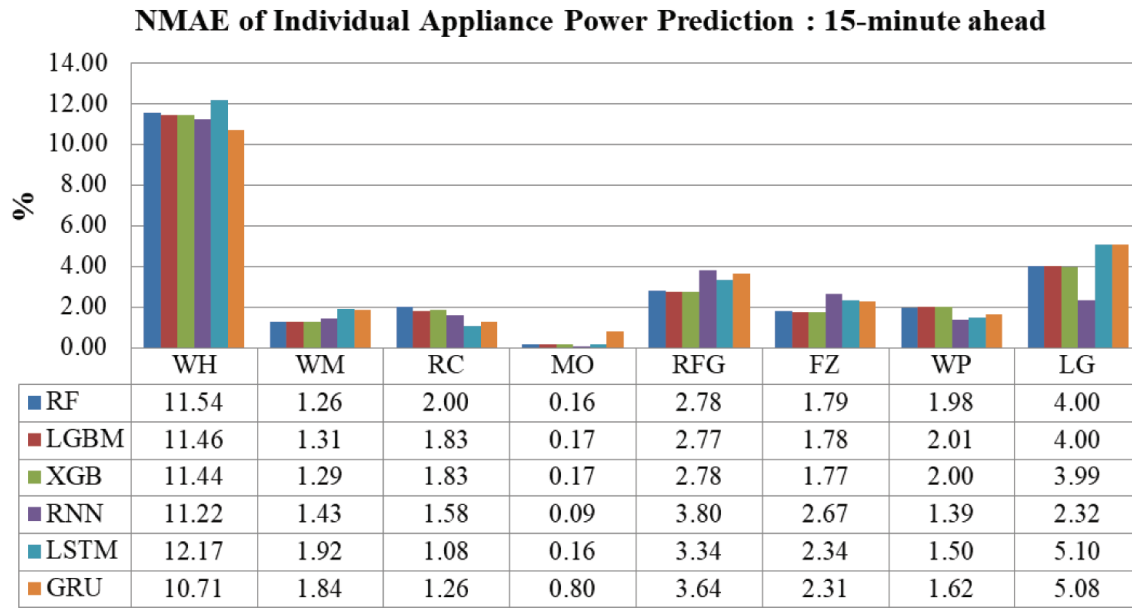


Figure 10: NMAE of 15-min ahead individual appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; WP, Water pump; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RC, rice cooker; RFG, refrigerator; WH, water heater; WM, washing machine.

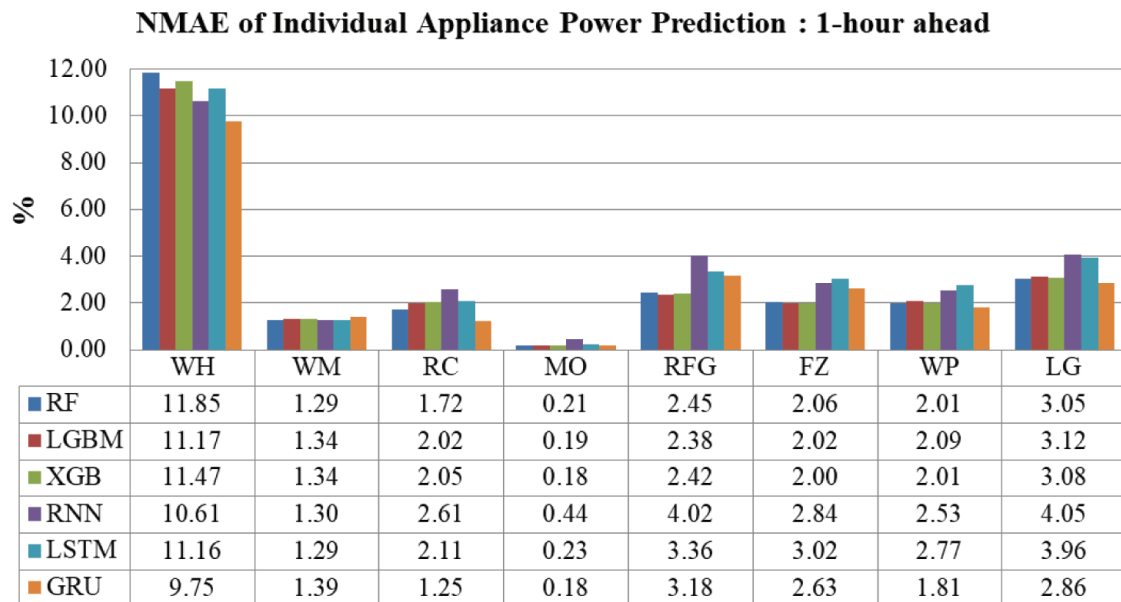


Figure 11: NMAE of 1-hr ahead individual appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; WP, Water pump; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RC, rice cooker; RFG, refrigerator; WH, water heater; WM, washing machine.

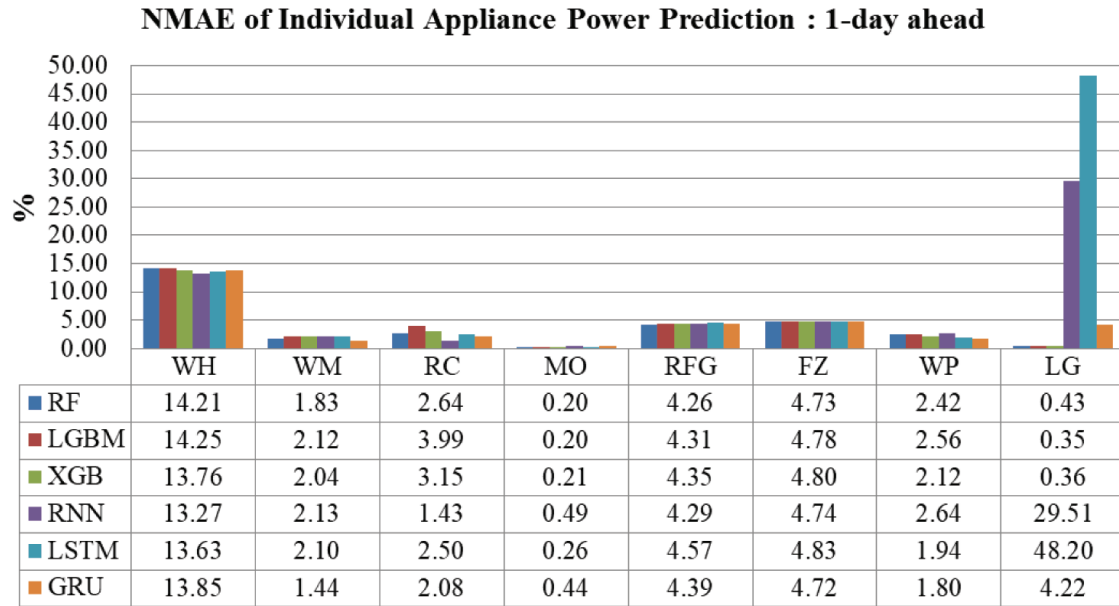


Figure 12: NMAE of 1-day ahead individual appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; WP, Water pump; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RC, rice cooker; RFG, refrigerator; WH, water heater; WM, washing machine.

Table 3: NMAE comparison of individual appliance power prediction

Time horizon	Lowest NMAE (%) (appliance, method)			
	Reschedulable load		Non-reschedulable load	
	Tree-based algorithm	Deep-learning algorithm	Tree-based algorithm	Deep-learning algorithm
1-min	0.14 (MO, RF)	0.10 (MO, RNN)	0.26 (WP, RF)	0.21 (WP, LSTM)
15-min	0.16 (MO, RF)	0.09 (MO, RNN)	1.77 (FZ, XGB)	1.39 (WP, RNN)
1-hr	0.18 (MO, XGB)	0.18 (MO, GRU)	2.00 (FZ, XGB)	1.81 (WP, GRU)
1-day	0.20 (MO, RF/LGBM)	0.26 (MO, LSTM)	0.35 (LG, LGBM)	1.80 (WP, GRU)

GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; WP, water pump.

learning models outperform tree-based models. It suggests that the MO power usage pattern at short-term time scales shows strong temporal dependence, so it is better captured by temporal-based models such as RNNs. However, at a longer horizon (1-day), the temporal dependency decreases, so that non-temporal models, such as RF or LGBM, outperform temporal-based models.

For the non-reschedulable load, no single appliance achieves the lowest NMAE across all scenarios. For a short-term time horizon, the deep-learning models perform better than tree-based. Interestingly, at a 1-day time horizon, the tree-based model (LGBM) shows the lowest NMAE for LG power prediction. For a stable daily power pattern, such as lighting, a temporal-based model might not exploit it.

Overall comparison in Table 3 indicates that no single prediction algorithm shows superiority across the different appliances and time horizons. Deep learning models perform better for short-term forecasting and for appliances with complex temporal dependencies. Tree-based models perform better at longer time horizons or for appliances with stable usage patterns.

Figures 13–16 depict the NRMSE of individual appliance power predictions for 1-min, 15-min, 1-hr, and 1-day time horizons. Table 4 compares the lowest NRMSEs for individual-appliance power predictions across different scenarios, where the lowest NRMSE of 2.45% is achieved by the MO using an RNN at a 1-min horizon for reschedulable load, and the lowest NRMSE of 3.14% is achieved by the RFG using GRU at a 1-min horizon for non-reschedulable load.

For reschedulable load, the MO achieves the lowest NRMSE for all scenarios. At very short-term time horizons (1-min and 15-min), the tree-based model shows slight superiority over the deep learning model. Oppositely, at 1-hr and 1-day time horizons, the deep learning model shows slight superiority over tree-based models.

For non-reschedulable load, no single appliance achieves the lowest NRMSE across all scenarios.

The tree-based models outperform deep learning at longer time horizons.

Overall comparison in Table 4 indicates that tree-based models perform better at predicting non-reschedulable loads, while deep learning models perform better for reschedulable loads.

Figures 17 and 18 show the NMAE and NRMSE of multi-horizon average appliance power predictions, respectively. Figure 17 shows that the RF achieves the lowest NMAE of 0.69% for a 1-min time horizon, the RNN achieves the lowest NMAE of 3.06% for a 15-min time horizon, the GRU achieves the lowest NMAE of 2.88% for a 1-hr time horizon, and the RF achieves the lowest NMAE of 3.84% for a 1-day time horizon.

Figure 18 shows that the LGBM achieves the lowest NRMSE of 4.22% for a 1-min time horizon, the LSTM achieves the lowest NRMSE of 9.09% for a 15-min time horizon, the GRU achieves the lowest NRMSE of 9.03% for a 1-hr time horizon, and the RF achieves the lowest NRMSE of 9.31% for a 1-day time horizon. The above results suggest that the tree-based models outperform the deep learning models for multi-horizon average appliance power prediction at the very short (1-min) and long (1-day) time horizons.

Figures 17 and 18 demonstrate that the NMAE and NRMSE increase when the time horizon increases

NRMSE of Individual Appliance Power Prediction : 1-minute ahead

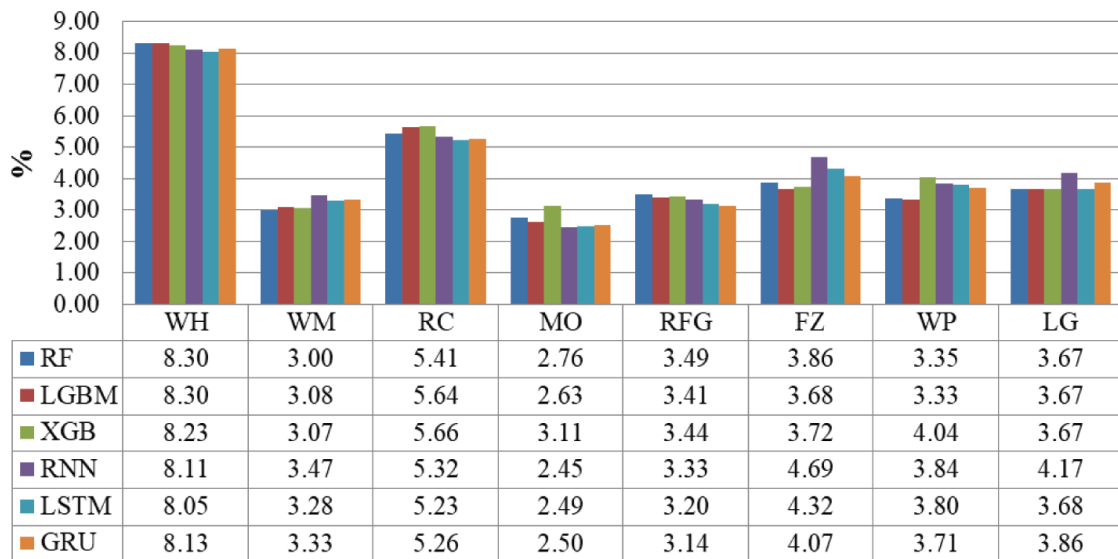


Figure 13: NRMSE of 1-min ahead individual appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; WP, Water pump; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RC, rice cooker; RFG, refrigerator; WH, water heater; WM, washing machine.

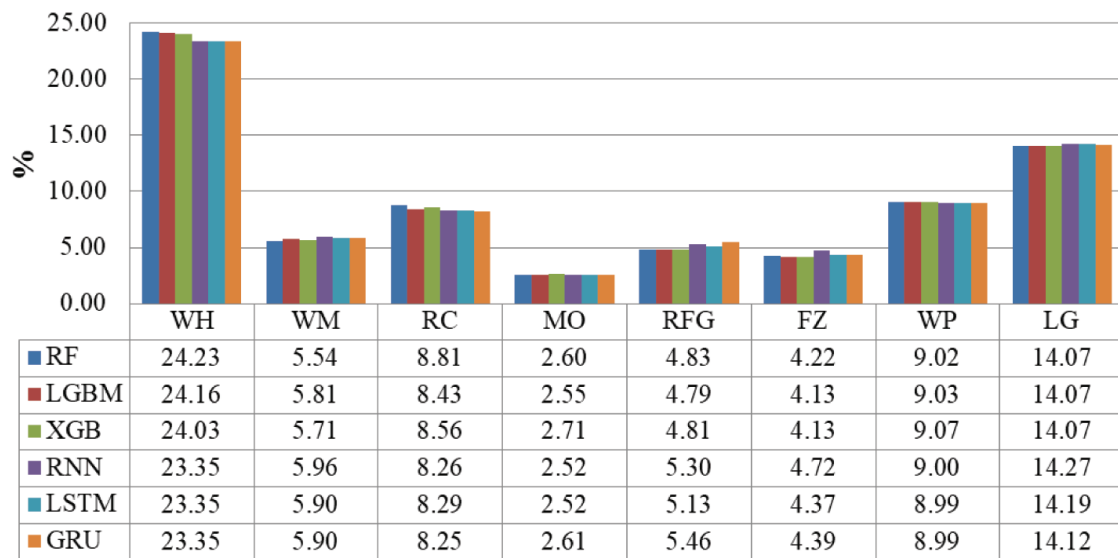
NRMSE of Individual Appliance Power Prediction : 15-minute ahead

Figure 14: NRMSE of 15-min ahead prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; WP, Water pump; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RC, rice cooker; RFG, refrigerator; WH, water heater; WM, washing machine.

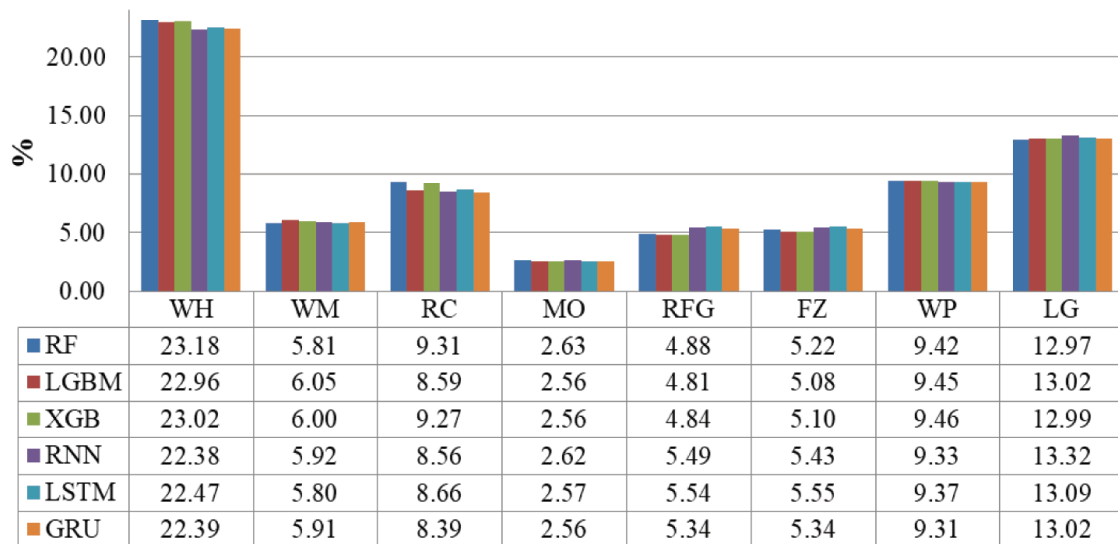
NRMSE of Individual Appliance Power Prediction : 1-hour ahead

Figure 15: NRMSE of 1-hr ahead individual appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; WP, Water pump; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RC, rice cooker; RFG, refrigerator; WH, water heater; WM, washing machine.

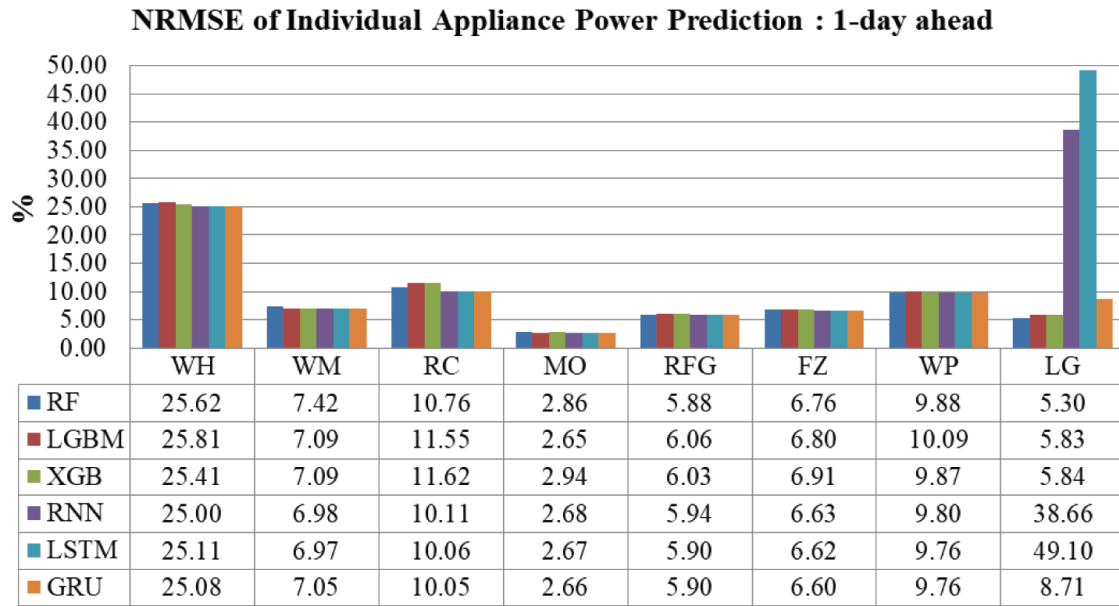


Figure 16: NRMSE of 1-day ahead individual appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; WP, Water pump; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RC, rice cooker; RFG, refrigerator; WH, water heater; WM, washing machine.

Table 4: NRMSE comparison of individual appliance power prediction

Time horizon	Lowest NRMSE (%) (appliance, method)			
	Reschedulable load		Non- reschedulable load	
	Tree-based algorithm	Deep-learning algorithm	Tree-based algorithm	Deep-learning algorithm
1-min	2.63 (MO, LGBM)	2.45 (MO, RNN)	3.33 (WP, LGBM)	3.14 (RFG, GRU)
15-min	2.55 (MO, LGBM)	2.52 (MO, RNN/ LSTM)	4.13 (FZ, LGBM/XGB)	4.37 (FZ, LSTM)
1-hr	2.56 (MO, LGBM/ XGB)	2.56 (MO, GRU)	4.81 (RFG, LGBM)	5.34 (RFG/FZ, GRU)
1-day	2.65 (MO, LGBM)	2.66 (MO, GRU)	5.30 (LG, RF)	5.90 (RFG, LSTM/ GRU)

GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; MO, microwave oven; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting; FZ, freezer; LG, lighting; RFG, refrigerator; WP, water pump.

for all prediction techniques. It suggests that prediction accuracy decreases as the forecasting interval increases. Both tree-based and deep-learning models face increasing difficulty in capturing the underlying appliance behavior.

c. Evaluation of total power prediction

Figures 19–22 depict the NMAE of total power predictions for 1-min, 15-min, 1-hr, and 1-day time horizons. In each figure, the NMAEs for six methods are compared across four different feature groups. The

NMAE of Average Appliance Power Prediction: Multi-Horizon

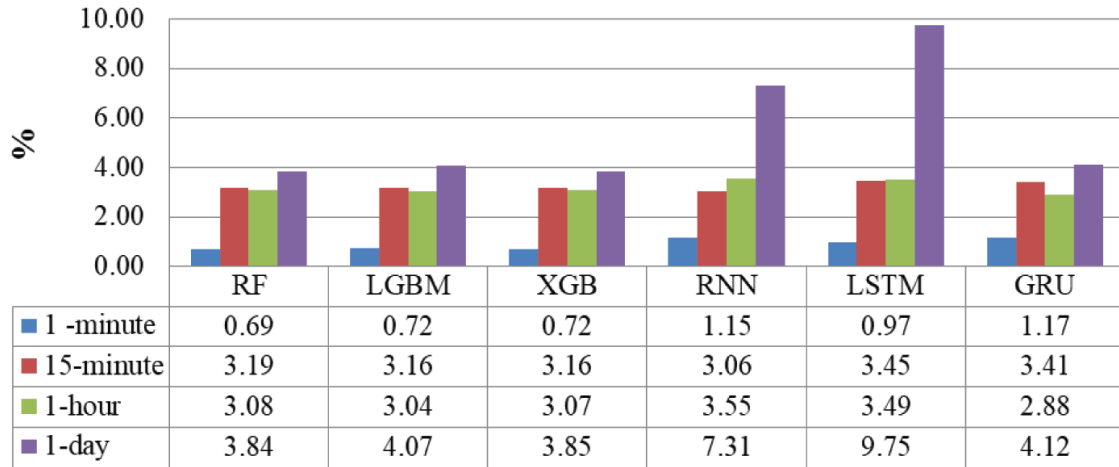


Figure 17: NMAE of multi-horizon average appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

NRMSE of Average Appliance Power Prediction: Multi-Horizon

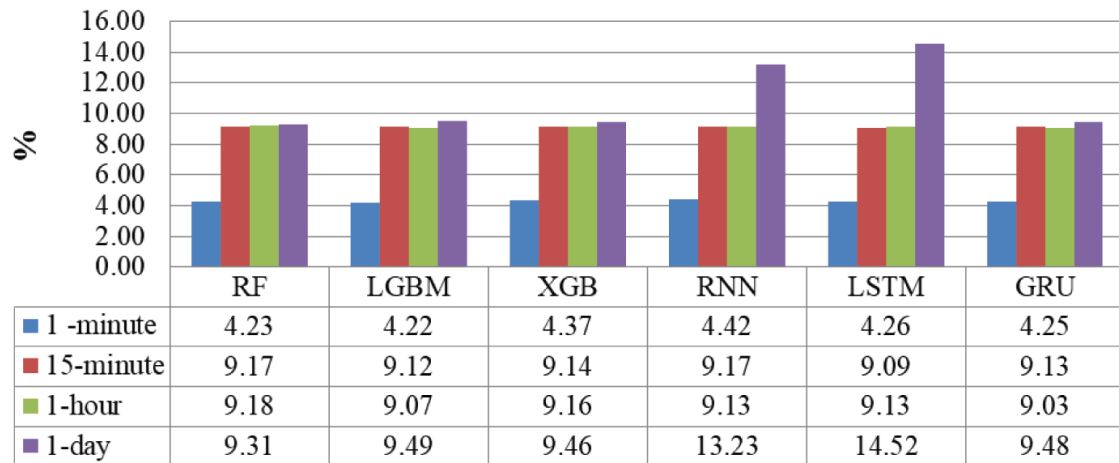


Figure 18: NRMSE of multi-horizon average appliance power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

comparison results are summarized in Table 5. Table 5 shows that the lowest NMAE of 2.03% is achieved by the Feature Group-C using an LGBM or XGB at a 1-min horizon for the tree-based algorithm, and the lowest NMAE of 2.53% is achieved by the Feature Group-C using GRU at a 1-min horizon for the deep learning algorithm.

For the tree-based algorithm, Feature Group-C achieves the lowest NMAE across all time horizons, where LGBM is superior at 1-min and 15-min time horizons, XGB is superior at 1-min and 1-hr, and RF is superior at 1-day. It suggests that the Feature Group-C, which combines all appliance power consumption and total power as the input features

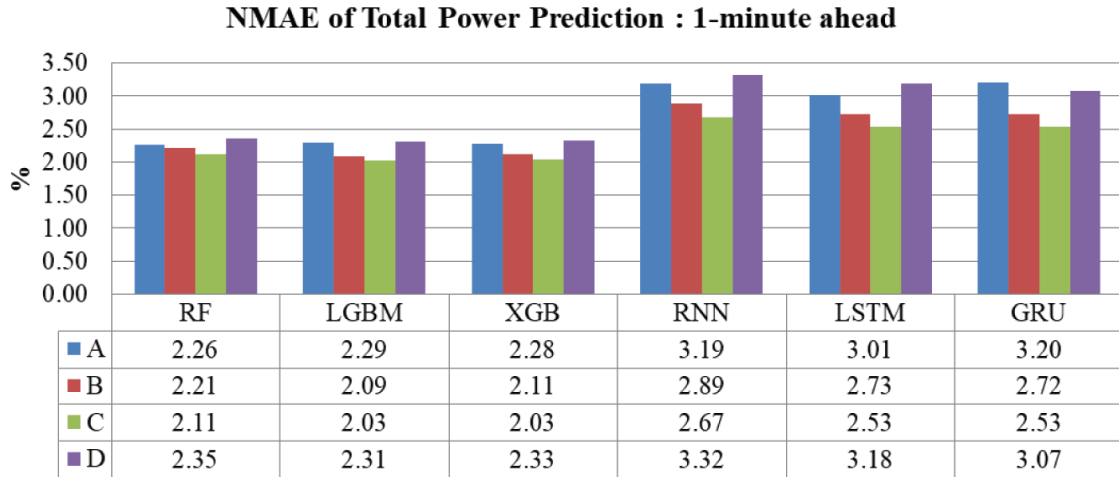


Figure 19: NMAE of 1-min ahead total power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

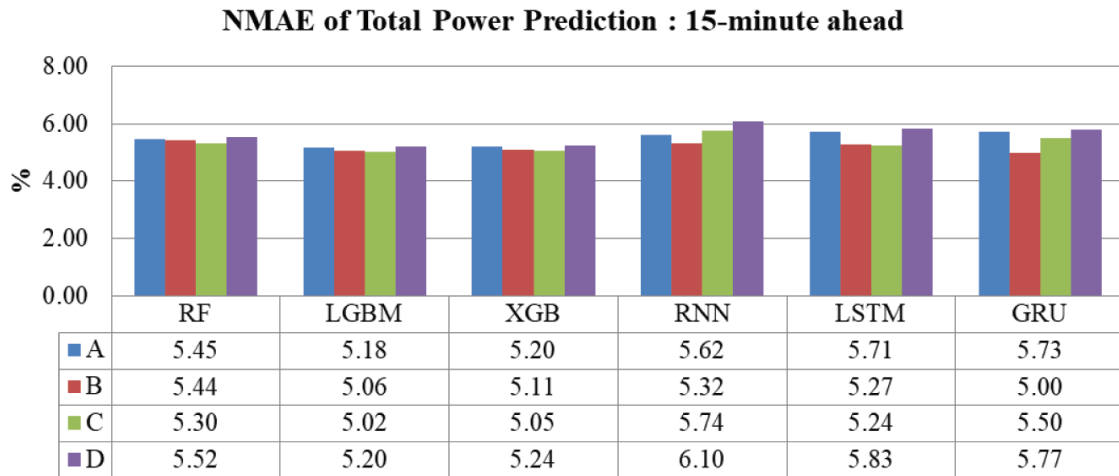


Figure 20: NMAE of 15-min ahead total power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

of the prediction models, effectively reduces the NMAE.

For the deep learning algorithm, Feature Group-C achieves the lowest NMAE at 1-min and 1-hr time horizons, where GRU is superior, and LSTM is superior at 1-hr and 1-day time horizons. Feature Group-A achieves the lowest NMAE at 1-day time horizon using the LSTM. Again, the Feature Group-C effectively reduces the NMAE, especially for very short-term prediction.

Figures 23–26 depict the NRMSE of total power predictions for 1-min, 15-min, 1-hr, and 1-day time horizons. In each figure, the NRMSEs for six methods are compared across four different feature groups. The comparison results are summarized in Table 6, where the lowest NRMSE of 4.61% is achieved by the Feature Group-C using an LGBM at a 1-min horizon for the tree-based algorithm, and the lowest NRMSE of 4.91% is achieved by the Feature Group-C using GRU at a 1-min horizon for the deep learning algorithm.

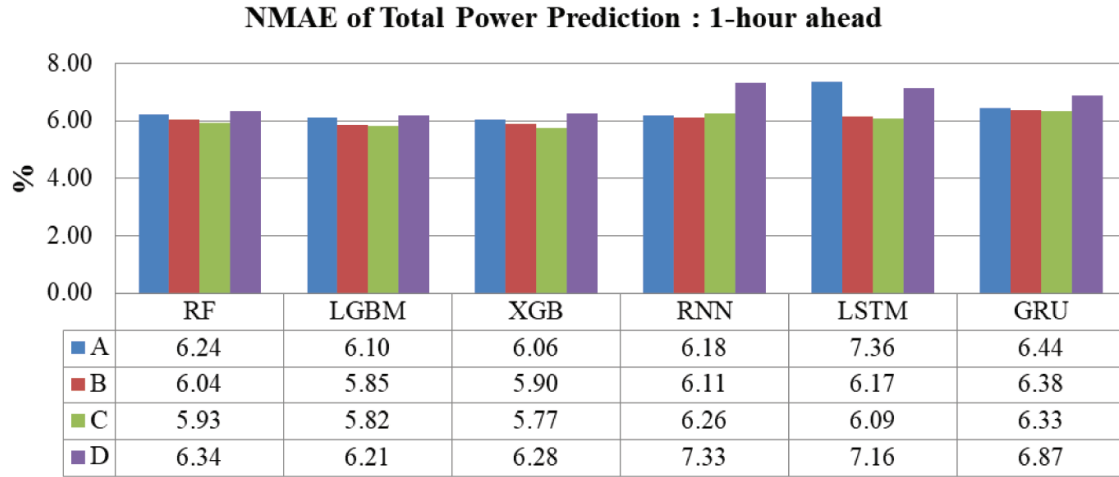


Figure 21: NMAE of 1-hr ahead total power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

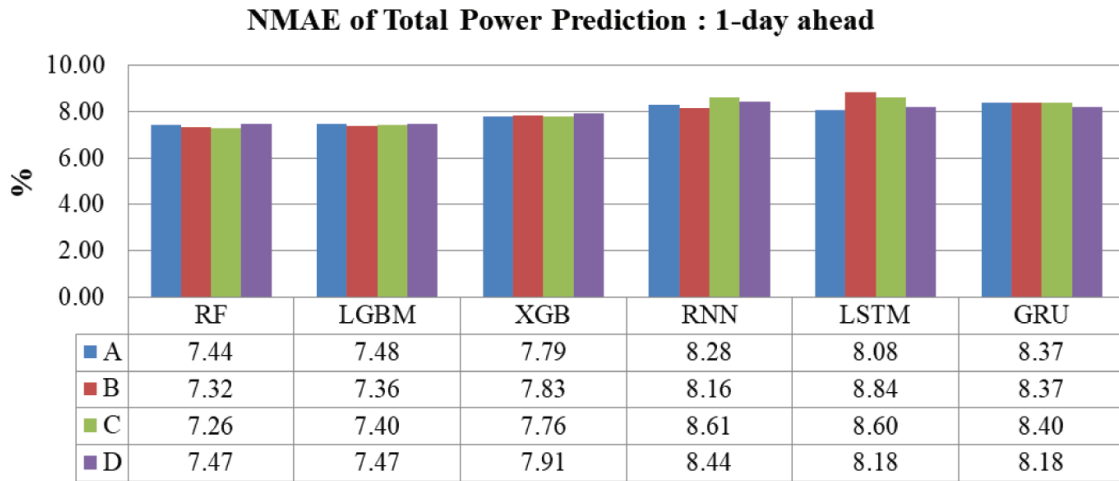


Figure 22: NMAE of 1-day ahead total power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

For the tree-based algorithm, Feature Group-C achieves the lowest NRMSE at very short-term time horizons, where LGBM is superior at 1-min and 15-min time horizons, XGB is superior at 1-hr. Feature Group B and D achieve the lowest NRMSE at 1-day using the RF. It suggests that the Feature Group-C effectively reduces the NRMSE, especially for very short-term prediction.

For the deep learning algorithm, Feature Group-C achieves the lowest NRMSE at very short-term time horizons, where GRU is superior at a 1-min time horizon, LSTM at a 15-min time horizon. Feature Group-B

achieves the lowest NRMSE at 1-hr time horizon using the RNN. Feature Group-A achieves the lowest NRMSE at 1-day using the RNN. Similarly, before that, the Feature Group-C effectively reduces the NRMSE, especially for very short-term prediction.

Figures 27 and 28 show the NMAE and NRMSE of multi-horizon total power predictions, respectively. Figure 27 shows that the Feature Group-C achieves the lowest NMAE of 2.32% for a 1-min time horizon, the Feature Group-B achieves the lowest NMAE of 5.20% for a 15-min time horizon, the Feature Group-C achieves the lowest NMAE of 6.03% for a

Table 5: NMAE comparison of total power prediction

Time horizon	Lowest NMAE (%) (feature, method)	
	Tree-based algorithm	Deep learning algorithm
1-min	2.03 (C, LGBM/XGB)	2.53 (C, LSTM/GRU)
15-min	5.02 (C, LGBM)	5.00 (B, GRU)
1-hr	5.77 (C, XGB)	6.09 (C, LSTM)
1-day	7.26 (C, RF)	8.08 (A, LSTM)

GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NMAE, normalized mean absolute error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

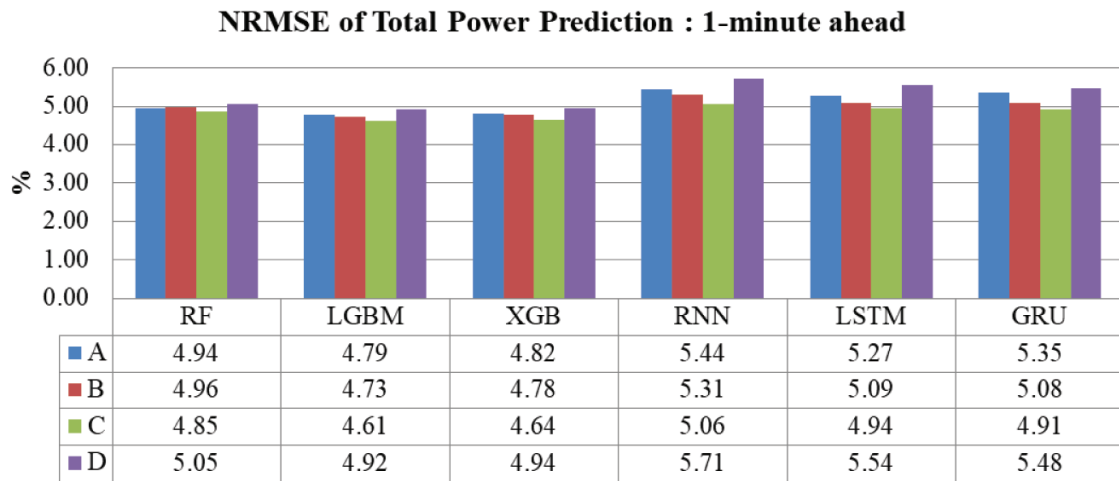


Figure 23: NRMSE of 1-min ahead total power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

1-hr time horizon, and the Feature Group-A achieves the lowest NMAE of 7.91% for a 1-day time horizon.

Figure 28 shows that Feature Group-C achieves the lowest NRMSE of 4.83% for a 1-min time horizon, the lowest NRMSE of 8.31% for a 15-min time horizon, the lowest NRMSE of 9.64% for a 1-hr time horizon, and Feature Group-D achieves the lowest NRMSE of 11.31% for a 1-day time horizon. The results suggest that the Feature Group-C is an effective feature in multi-horizon average total power prediction, especially for the very short-term forecasting.

Figures 27 and 28 demonstrate that the NMAE and NRMSE increase when the time horizon increases, regardless of feature group. A longer horizon makes

it harder for the algorithm to capture the model, even with several features adopted.

d. Evaluation summary

From previous discussions, several findings can be summarized as follows:

1. Individual appliance power prediction:
 - a. The NMAE comparison shows that deep learning models are advantageous for short-term forecasting and appliances with complex temporal behavior. In contrast, tree-based models are more effective for longer horizons or appliances with stable usage patterns.

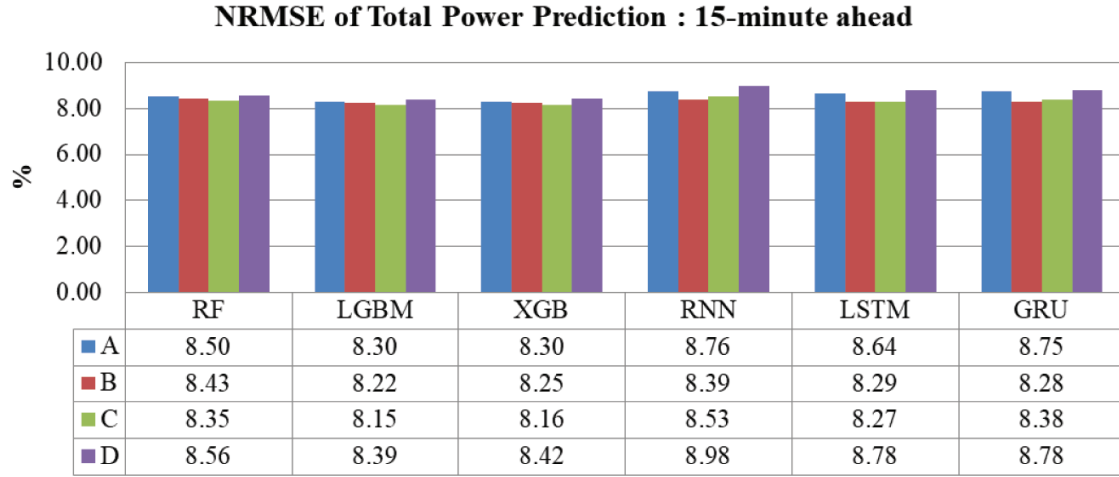


Figure 24: NRMSE of 15-min ahead total power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

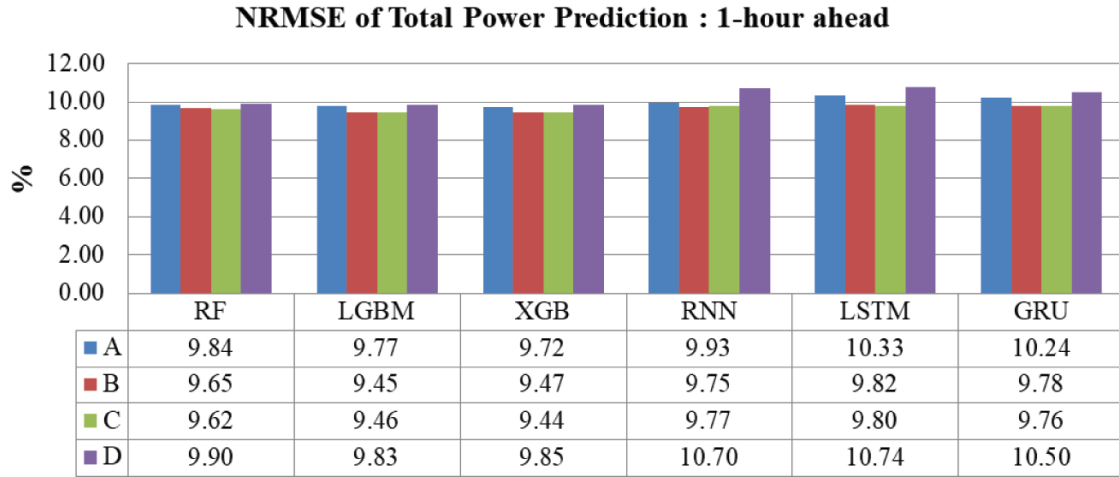


Figure 25: NRMSE of 1-hr ahead total power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

- b. The NRMSE comparisons show that tree-based models are more effective for non-reschedulable loads, while deep-learning models provide advantages for forecasting of reschedulable loads.
- c. The NMAE and NRMSE increase with the time horizon, regardless of the prediction algorithm. It indicates that the prediction algorithm becomes increasingly challenging to capture the underlying appliance behavior over a longer horizon.
2. Total power prediction:
 - a. Feature Group-C, which combines all individual appliance power consumption with total power, achieves the lowest NMAE and NRMSE across most horizons for both tree-based and deep-learning models, particularly for very short-term forecasts.

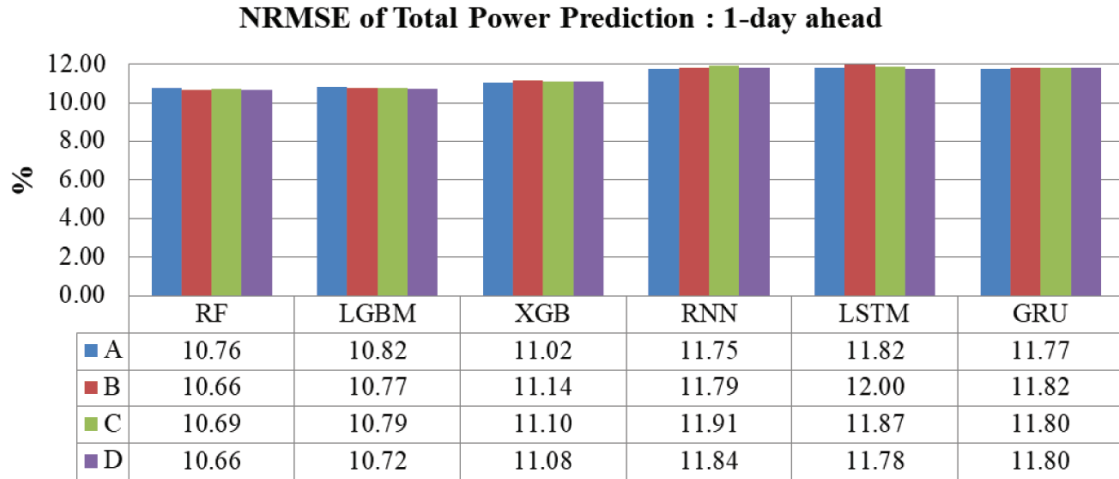


Figure 26: NRMSE of 1-day ahead total power prediction. GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

Table 6: NRMSE comparison of total power prediction

Time horizon	Lowest NRMSE (%) (feature, method)	
	Tree-based algorithm	Deep-learning algorithm
1-min	4.61 (C, LGBM)	4.91 (C, GRU)
15-min	8.15 (C, LGBM)	8.27 (C, LSTM)
1-hr	9.44 (C, XGB)	9.75 (B, RNN)
1-day	10.66 (B/D, RF)	11.75 (A, RNN)

GRU, gated recurrent unit; LGBM, light gradient boosting machine; LSTM, long short-term memory; NRMSE, normalized root mean square error; RF, random forest; RNN, recurrent neural network; XGB, extreme gradient boosting.

- b. Among tree-based methods, no single model performs across multiple horizons. In deep learning, GRUs and LSTMs dominate short-term horizons.
- c. The NMAE and NRMSE increase when the time horizon increases, regardless of feature group. It indicates that the algorithm is more complicated to capture the model, even with several features adopted.
3. The appliance-specific, horizon-dependent model selection strategy is essential for optimizing prediction accuracy in HEMS.
4. Incorporating all individual appliance power measurements, along with total power, as model inputs enhances total power prediction accuracy.

IV. Conclusion

This paper presented a comprehensive multi-horizon prediction framework for home energy consumption that simultaneously considers individual appliance loads and total household power. An IoT-based SCADA monitoring system was used to collect the home appliance power consumption. The approach was evaluated using six machine learning techniques, comprising tree-based and deep learning models, across four forecasting horizons. The evaluation results for NMAE and NRMSE show that deep learning techniques are more effective for short-term predicting the individual appliance power. Meanwhile, tree-based algorithms are superior for longer prediction horizons and for appliances with stable usage patterns. The most valuable finding is that

NMAE of Total Power Prediction: Multi-Horizon

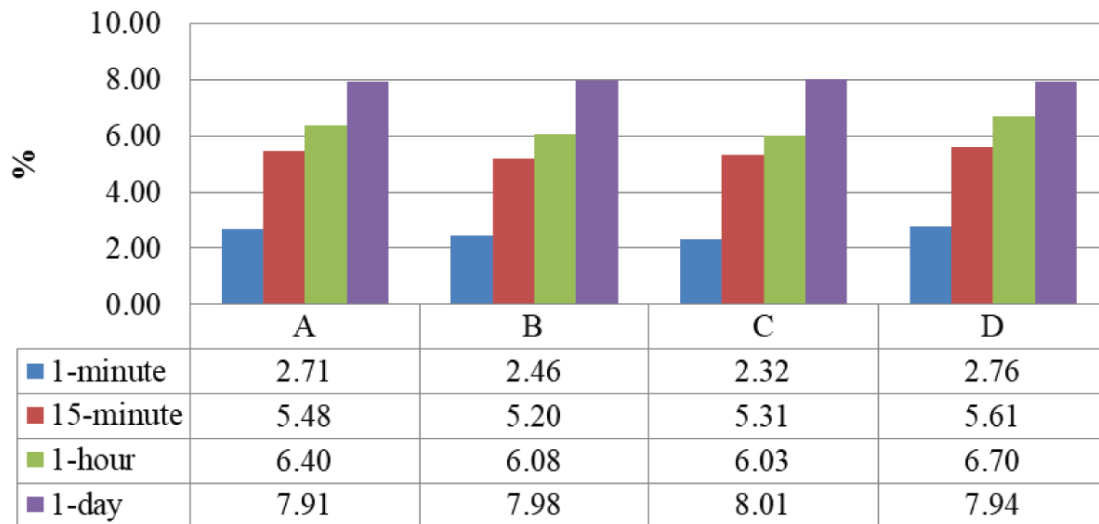


Figure 27: NMAE of multi-horizon total power prediction. NMAE, normalized mean absolute error.

NRMSE of Total Power Prediction: Multi-Horizon

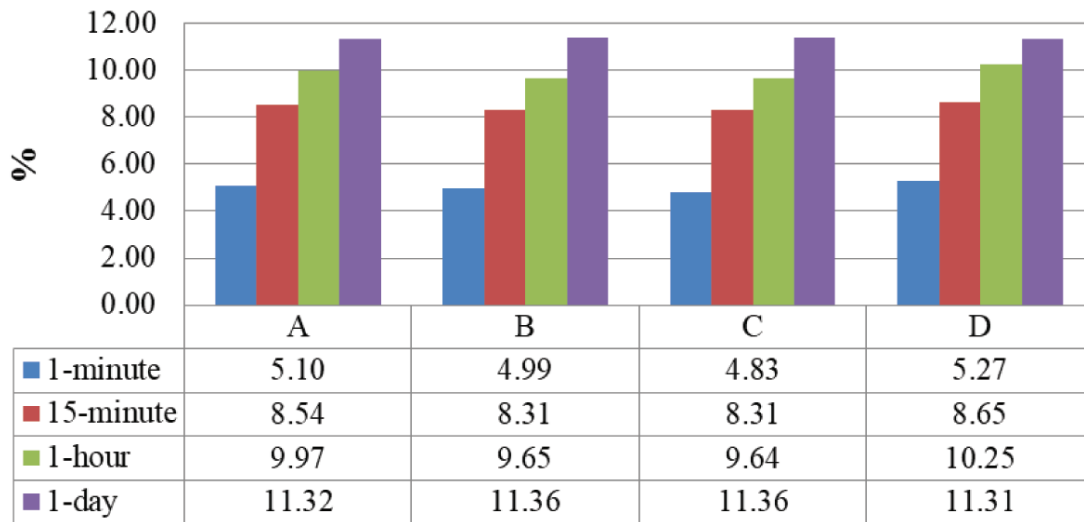


Figure 28: NRMSE of multi-horizon total power prediction. NRMSE, normalized root mean square error.

combining appliance-level power measurements significantly improves the accuracy of total power prediction.

Future work may extend this research by integrating weather data, user behavior patterns, and occupancy information, and by exploring advanced algorithms to enhance predictive capability further.

Acknowledgment

The authors thank the Institute for Research and Community Service ITN Malang for funding this research. (Research funding No.: ITN.07.037.011/I. LPPM/2025).

References

- [1] Y. Zhao, J. Li, C. Chen, and Q. Guan, "A Diffusion-Attention-Enhanced Temporal (DATE-TM) Model: A Multi-Feature-Driven Model for Very-Short-Term Household Load Forecasting," *Energies*, vol. 18, no. 3, pp. 1–21, 2025, doi: 10.3390/en18030486.
- [2] S. R. Vangipuram and A. V. Giridhar, "Short Term Residential Load Forecasting Using Temporal Weather Based Embedding Stacked LSTMs," *IEEE Latin America Transactions*, vol. 23, no. 6, pp. 497–507, 2025, doi: 10.1109/TLA.2025.11007195.
- [3] A. Bhandary, V. Dobariya, G. Yenduri, R. H. Jhaveri, S. Gochhait, and F. Benedetto, "Enhancing Household Energy Consumption Predictions Through Explainable AI Frameworks," *IEEE Access*, vol. 12, no. March, pp. 36764–36777, 2024, doi: 10.1109/ACCESS.2024.3373552.
- [4] M. Razghandi and D. Turgut, "Residential appliance-level load forecasting with deep learning," in *Proceedings - IEEE Global Communications Conference, GLOBECOM*, Taipei, 2020, pp. 1–6. doi: 10.1109/GLOBECOM42002.2020.9348197.
- [5] Y. Zhou, A. S. Nair, D. Ganger, A. Tripathi, C. Baone, and H. Zhu, "Appliance Level Short-term Load Forecasting via Recurrent Neural Network," in *IEEE Power and Energy Society General Meeting*, Denver, 2022, pp. 1–5. doi: 10.1109/PESGM48719.2022.9917171.
- [6] Z. Severiche-Maury, C. Uc-Ríos, J. E. Sierra, and A. Guerrero, "Predicting Energy Consumption and Time of Use of Home Appliances in an HEMS Using LSTM Networks and Smart Meters: A Case Study in Sincelejo, Colombia," *Sustainability (Switzerland)*, vol. 17, no. 11, 2025, doi: 10.3390/su17114749.
- [7] F. Wahid, R. Ghazali, L. H. Ismail, and A. M. A. Aseere, "An Optimal Neural Network for Hourly and Daily Energy Consumption Prediction in Buildings," *International Journal of Swarm Intelligence Research*, vol. 14, no. 1, pp. 1–13, 2023, doi: 10.4018/IJSIR.316649.
- [8] H. Tantyoko, S. R. Purnama, and E. Vianita, "Multi-Horizon Short-Term Residential Load Forecasting Using Decomposition-Based Linear Neural Network," *Advance Sustainable Science, Engineering and Technology*, vol. 7, no. 3, pp. 1–8, 2025, doi: 10.26877/asset.v7i3.2033.
- [9] L. Ismail, H. Materwala, and F. K. Dankar, "Machine Learning Data-Driven Residential Load Multi-Level Forecasting with Univariate and Multivariate Time Series Models Toward Sustainable Smart Homes," *IEEE Access*, vol. 12, no. April, pp. 55632–55668, 2024, doi: 10.1109/ACCESS.2024.3383958.
- [10] B. Devanathan, J. J. Varshitha, P. P. Kumar, S. A. Lakshmanan, and K. K. Prakash, "Explainable AI Framework Using XGBoost With SHAP and LIME for Multi-Scale Household Energy Forecasting," *IEEE Access*, vol. 13, no. August, pp. 149750–149764, 2025, doi: 10.1109/ACCESS.2025.3602673.
- [11] F. Lazzari et al., "User behaviour models to forecast electricity consumption of residential customers based on smart metering data," *Energy Reports*, vol. 8, pp. 3680–3691, 2022, doi: 10.1016/j.egy.2022.02.260.
- [12] K. Liang, F. Liu, and Y. Zhang, "Household Power Consumption Prediction Method Based on Selective Ensemble Learning," *IEEE Access*, vol. 8, pp. 95657–95666, 2020, doi: 10.1109/ACCESS.2020.2996260.
- [13] S. Iram et al., "An Innovative Machine Learning Technique for the Prediction of Weather Based Smart Home Energy Consumption," *IEEE Access*, vol. 11, no. July, pp. 76300–76320, 2023, doi: 10.1109/ACCESS.2023.3287145.
- [14] Z. Severiche-Maury et al., "LSTM Networks for Home Energy Efficiency," *Designs*, vol. 8, no. 4, pp. 1–19, 2024, doi: 10.3390/designs8040078.
- [15] A. Pai H, K. K. Mishra, M. T. R, J. V. M. L. Jeyan, and A. Sayal, "Enhanced household energy consumption forecasting using multivariate long short-term memory (LSTM) networks with weather data integration," *Results in Engineering*, vol. 27, no. April, p. 106512, 2025, doi: 10.1016/j.rineng.2025.106512.
- [16] L. Fan, J. Li, and X. P. Zhang, "Load prediction methods using machine learning for home energy management systems based on human behavior patterns recognition," *CSEE Journal of Power and Energy Systems*, vol. 6, no. 3, pp. 563–571, 2020, doi: 10.17775/CSEEJPES.2018.01130.
- [17] A. Mehmood, K. T. Lee, and D. H. Kim, "Energy Prediction and Optimization for Smart Homes with Weather Metric-Weight Coefficients," *Sensors*, vol. 23, no. 7, pp. 1–22, 2023, doi: 10.3390/s23073640.
- [18] R. Li et al., "MACLSTM: A Weather Attributes Enabled Recurrent Approach to Appliance-Level Energy Consumption Forecasting," *Computers, Materials and Continua*, vol. 82, no. 2, pp. 2969–2984, 2025, doi: 10.32604/cmc.2025.060230.
- [19] Q. W. Khan, R. Ahmad, A. Rizwan, A. N. Khan, K. T. Lee, and D. H. Kim, "Optimizing energy efficiency and comfort in smart homes through predictive optimization: A case study with indoor environmental parameter consideration," *Energy Reports*, vol. 11, no. October 2023, pp. 5619–5637, 2024, doi: 10.1016/j.egy.2024.05.038.

- [20] T. Singh, A. Solanki, S. K. Sharma, N. Z. Jhanjhi, and R. M. Ghoniem, "Grey Wolf Optimization-Based CNN-LSTM Network for the Prediction of Energy Consumption in Smart Home Environment," *IEEE Access*, vol. 11, no. October, pp. 114917–114935, 2023, doi: 10.1109/ACCESS.2023.3311751.
- [21] A. Binbusayyis and M. Sha, "Energy consumption prediction using modified deep CNN-Bi LSTM with attention mechanism," *Heliyon*, vol. 11, no. 1, p. e41507, 2025, doi: 10.1016/j.heliyon.2024.e41507.
- [22] K. H. Kumar Reddy, R. K. Behera, M. H. Gururaj, and R. K. Bailayar Singh, "An Ensemble Learning based Energy Forecasting Model: A Sustainable Home Energy Management System for Smart City," *Procedia Computer Science*, vol. 258, pp. 1316–1325, 2025, doi: 10.1016/j.procs.2025.04.365.
- [23] K. H. Baesmat, E. E. Regentova, and Y. Baghzouz, "A Hybrid machine learning–statistical based method for short-term energy consumption prediction in residential buildings," *Energy and AI*, vol. 21, no. July, p. 100552, 2025, doi: 10.1016/j.egyai.2025.100552.
- [24] S. Priyanto, A. Soetedjo, and I. B. Sulistiawati, "Monitoring the Power Consumption of Home Appliances Using an IoT-Based SCADA System," in *2024 8th International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2024*, Yogyakarta, Indonesia: IEEE, 2024, pp. 441–446. doi: 10.1109/ICITISEE63424.2024.10730065.
- [25] S. Priyanto, A. Soetedjo, and I. B. Sulistiawati, "Integration of SCADA and embedded fuzzy-based load scheduling in Home Energy Management System with grid-connected PV," *INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS*, vol. 18, no. 1, pp. 1–24, 2025, doi: 10.2478/ijssis-2025-0020.
- [26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [27] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random Forests," in *Ensemble Machine Learning*, C. Zhang and Y. Ma, Eds., Boston, MA: Springer, 2012, pp. 157–175. doi: 10.1007/978-1-4419-9326-7.
- [28] J. H. Friedman, "GREEDY FUNCTION APPROXIMATION: A GRADIENT BOOSTING MACHINE," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [29] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proceedings of the Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 3146–3154.
- [30] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [31] A. Ibrahim Ahmed Osman, A. Najah Ahmed, M. F. Chow, Y. Feng Huang, and A. El-Shafie, "Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia," *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 1545–1556, 2021, doi: 10.1016/j.asej.2020.11.011.
- [32] W. Dong, Y. Huang, B. Lehan, and G. Ma, "XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring," *Automation in Construction*, vol. 114, p. 103155, Jun. 2020, doi: 10.1016/j.autcon.2020.103155.
- [33] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990, doi: 10.1016/0364-0213(90)90002-E.
- [34] S. Hochreiter and J. Schmidhuber, "Lstm," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1007/978-3-030-63416-2_300359.
- [35] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014, [Online]. Available: <http://arxiv.org/abs/1412.3555>
- [36] "scikit-learn: Machine Learning in Python." Accessed: Oct. 01, 2024. [Online]. Available: <https://scikit-learn.org/stable/>
- [37] "Light Gradient Boosting Machine." Accessed: Jul. 01, 2025. [Online]. Available: <https://github.com/microsoft/LightGBM>
- [38] "eXtreme Gradient Boosting." Accessed: Jul. 01, 2025. [Online]. Available: <https://github.com/dmlc/xgboost?tab=readme-ov-file>
- [39] "TensorFlow: An end-to-end platform for machine learning." Accessed: Jul. 01, 2025. [Online]. Available: <https://www.tensorflow.org/>