International Journal of Artificial Intelegence Research
Vol. 8, No. 2, December 2024
ISSN 2579-7298

Enhancing Electricity Consumption Prediction with Deep Learning Through Advanced-Data Splitting Techniques

Adinda Putri Pratiwi [a,1] , R.V Hari Ginardi [b,2,*], Ahmad Saikhu [a,3]

[a] Informatics Department, Institut Teknologi Sepuluh Nopember, Indonesia, 60111 [b]Information Technology Department, Institut Teknologi Sepuluh Nopember, Indonesia, 60111[1]
6025211030@student.its.ac.id; [2] hari@its.ac.id*; [3] saikhu@if.its.ac.id

ARTICLE INFO

Article history
Received
Revised
Accepted

Keywords
Energy Prediction
Deep Learning
Splitting Data
Multistep
Time-series

ABSTRACT

Energy consumption is increasing due to population growth and industrial activity, making electricity essential in human life. With limited natural resources, effective management of electrical resources is crucial to reduce energy usage amidst rising demand. The current trends on using deep learning as prediction can enhance the performances. To have good performance it needs correct preprocessing data, so it will produce a model with less overfitting. This research proposes a model using time-series cross-validation as the splitting data and correlation to choose the best features set for the prediction of electricity consumption. Experiments will compare time-series cross- validation and holdout methods to see the performances of splitting data and enhancing the multi-horizon data. The experiment used 8 sets of feature lists, which are paired in combination based on correlation to ensure the best features that are related. The result is splitting data using time-series cross-validation can maintain good perfomances on mode and holdout can maintain a good evaluation performance across the horizon. Feature sets that include temporal features have excellent results, especially when combined with features that have the strongest correlation relationship with electricity consumption, leading to an enhanced R^2. Among all the models tested, CNN-GRU had the best model for multistep prediction across various every horizons and featuresets.

1. Introduction

The increasing energy consumption all over the world has given special attention to governments and researchers. The increasing energy consumption happens because the population growth and industrial activity, but the availability of the energy is limited because of its source. Most of the energy that people use now is from natural resources, because of the reason we need to manage the energy to make it efficient, especially in electricity consumption.

The approach to managing the efficiency of electricity has been done by many researchers, mostly for building electricity management [1]. To reach the efficiency of managing electricity buildings can be achieved by early diagnosis. The early diagnosis becomes possible using the concept of forecasting. Forecasting is a technique to predict future information by using historical data as its input [2]. Management of electricity consumption in buildings can be challenging cause it has uncertain variables who had influence it. So, need to create a model forecasting that can capture the pattern of the historical data and good accuracy to have valid forecasting to ensure its effectiveness [3]. In forecasting there are two types, short-term and long-term, the difference between the types is the time window[4], short-term is predicting the next hours until the next weekand long-term is predicting the 6 months until a year or more.

Technological advances make forecasting easier as time goes by using artificial intelligence. Artificial intelligence can do more complex tasks and bigger datasets when compared to conventional ones

[5]. To reach better accuracy the artificial intelligence model needed better pre- processing, one of them is splitting data. Splitting data in a time-series dataset is very important, cause the right amount and method of splitting data will increase the accuracy of the model, if not it will make the model overfitting [6]. Splitting data for a time-series dataset had been discussed that a good method for splitting data will increase the accuracy and performance of the model [7], this was shown in an experiment using linear regression that using time-series cross-validation as splitting data made the accuracy of model increasing than using other method. Another thing about forecasting is the right amount of features and the feature itself, choosing the right feature will lead the model to have better performance [8].

Choosing the right model according to the dataset for prediction will produce better performance [9]. The traditional method is used in the research [10] using ARIMA to predict the next month's peak demand and KMeans clustering for clustering the time when the peak demand is in a day. Another research using clustering based on heatmap [11]. According to the study using deep learning will have a better understanding cause the algorithm will learn to analyze the data better than others [12], because it will extract the raw data in autonomous and hierarichal [13]. Another research to clustering the days by Kmeans and using the combination of neural network and genetic algorithm [14], ANN, K* algorithm, and ensemble bagging [15]. Another research was predicted by combining RNN and CNN [16], where combining one had better results than processing just one model. Another research [13] used just one model to predict, in this research state using LSTM to predict short time had poorer results than long-term forecasting.

Based on the several research that have been discussed, there are limited time windows in predicting, mostly using one-time prediction or direct prediction. These approaches had struggled to capture the dynamic time window. Need robustness for the model to reach the limitation of a fixed time window by using multistep prediction [17]. The multistep prediction method had the ability to predict over multiple time steps [18]. This approach can enhance the understanding of temporal patterns and trends from the data. Using effective multistep prediction can provide flexibility for future events [19]. The approaches using a multistep done by several researchs, in this research using multi-step for predicting the next 24 steps [20] using LSTM [21], Sarimax [22], XGBoost [23], R-CNN with adding ML-LSTM [24]. Another research [25] using LSTM-MIMO and comparing it with SVM and tree decisions to predict the next hour and next day, using deep learning had better results than machine learning.

Based on several research that had been discussed there are gaps, specifically in splitting data and the time horizon of prediction, stated the multi-horizon made the perfomances of model become poor. The research by [16] using combination of CNN and RNN had a good performances, but do not explore multi-horizon or various combinations of features. The challenge in multi-horizon prediction is the model prone to have poor performance and overfitting, because of the limited in input sequence.

To order the gaps in multi-horizon prediction, this study offered model forecasting to prevent overfitting with dynamic time prediction, employing a variety of pre-proceesing and feature selections approaches. A key focus is to analyze electricity consumption with the best performance using different time horizons and lookbacks, as well as the feature sets who influence the trend of electricity consumption.The rest of the paper will be organized at section 2 for the method, section 3 for the result and discussion, and section 4 is the conclusion of this research.

2. Method

Improving the forecasting so the model does not experience overfitting, it needs to use the right preprocessing data. Fig. 1 will briefly explain the steps in this research. The first step is collecting the dataset for the electricity consumption, the outdoor weather, the national holiday, and the duration of peak demand. The second step is to do preprocessing by checking the missing value, filling in the missing value, resampling data into the desired interval, and merging all datasets into one. The third step is feature selection, correlation to the target feature to see which features had strong and weak correlation.

The fourth step is splitting data into training and testing sets. The data splitting technique is based on well established from previous research, however unlike the previous study exclusively

using one single-step prediction prediction, our approached specifically addressed the need for multi-horizon prediction. After splitting the data, training will perform the training model using LSTM, GRU, CNN-LSTM, and CNN-GRU. After training the model will use testing data to make a prediction. Lastly, an evaluation of which model had the best performance using R2, MAE, MSE,and RMSE.

This research aims to make a forecast model using LSTM but with preprocessing by using the right splitting data and will resample data based on the peak demand of the day. Adding more features who related to electricity consumption, like outdoor weather data, public holidays, and the duration of peak demand in a day. This research will do prediction by a using multistep approach to strengthen the model so it can handle multi-horizon.

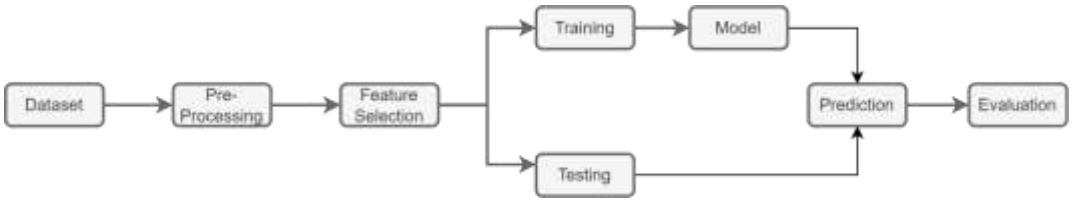


Fig. 1 The Workflow of Experiment

2.1. Dataset

The dataset used is a public dataset provided by Chulalongkorn University [26]. The dataset comes from the CU-BEMS building, which is intended for research in the energy sector. The CU- BEMS building has 7 floors and each floor is divided into several zones, for floors 1 and 2 have 4 zones, and 5 to 7 floors have 5 zones. The difference between the two zones is that floors 1 and 2 do not have environmental sensors. The data recorded in CUBEMS dataset are air conditioner, lamp, and plug loads, with a recording period of 18 months. The number of total data recorded was 790.560. The data has a time interval recorded per 1 minute. Every zone has an environmental sensor that records the indoor temperature, humidity, and ambient. However, the environmental sensor had much null data cause of the maintenance, so it couldn't be used.

The outdoor weather dataset is from NOAA [27] used data from 01-01-2018 until 31-12-2018 with a total data 730 data. The interval data used is 24 months with interval one data for one day. Features used in outdoor weather are the average temperature of the day, the maximal temperature of the day, the minimal temperature of the day, and precipitation.

The national holiday is from National Holiday Bangkok [28], which contains the type of holiday and the holiday name. The Dataset had labeled national holiday, weekend, weekday, observance, season, bank holiday, government holiday, and common local holiday, with a total data of 730. The national holiday is a list of days off in Thailand based on the government. The dataset had 7 labels, which were divided by national, observance, season, bank holiday, government holiday, common local holiday, and weekend. Besides the labels given, there's NaN data who considered as non- holiday day.

2.2. Pearson's Correlation Coefficient

Pearson's correlation coefficient is used to measure the strength of the relationship between two variables and the direction of the variable's linear relationship [29]. The strength of two variables can be measured with a value of -1 to +1, where a value of ±1 has a very good or interrelated relationship strength, the greater the value towards 0, the weaker the relationship, and the + sign indicates a positive relationship, and the - sign indicates a relationship negative.

2.3. LSTM

Long Short-Term Memory or what is commonly known as LSTM is a modification of RNN which is intended to overcome long dependencies in RNN. LSTM can maintain important information in previous data in one sequence to help with new data points. LSTM has a feedback connection, which allows LSTM to process the entire sequence of data [30]. LSTM has 3 gates, namely, the input gate, the forget gate, and the output gate. Forget gate functions to decide which information will be stored that is considered still relevant or not to the system. The input gate functions to receive information in the form of

hidden states which will be combined with the previous information. The gate output will determine the hidden state that will be sent to the next cell. Cell state is a vector that runs throughout the network and can be thought of as a conveyor belt between gates. Cell state has an important role in being used to store and carry information to pass between gates to the next cell.

2.4. GRU

Gated Recurrent Unit or GRU is a modification of RNN which is used to overcome dependencies on different time scales. GRU has two gates, namely reset and update. The reset gate is to determine how to combine old information and new input [31]. The update gate is to determine how much previous information will be kept. GRU combines the forget gate and input gate in LSTM into one, namely the update gate, and combines the hidden state and cell state. GRU is better used for models who had a small dataset.

2.5. CNN

Convolutional Neural Network or CNN is a type of neural network network that is usually used to detect or recognize objects in input data [32]. In CNN there are several layers, the first is a convolutional layer, a polling layer, and a fully connected layer. Convolutional layers are used to extract data features into feature maps that are used for training. Polling layers are used to create new filters based on rules so that they have dimensions that match the map features. A fully connected layer is a layer that contains features that have been extracted in the form of a multidimensional array, to perform flattening. The fully connected layer contains a hidden layer, anactivation function, a loss function, and an output layer.

2.6. Perfomance Measures

In deep learning there is a type of evaluation metrics have been used. In forecasting usually to evaluate the model using R^2, MAE, MSE, and RMSE [33]. The first model will do training and the remaining data will be used as test data. The result of the evaluation of the model will be seen what best model in forecasting and a model that experiences overfitting. MAE is to calculate the average of value from the deviation between the actual value and predicted value, the formula is stated in Equation 1. MSE calculates the values of the mean squared error in prediction between the original value and the predicted value, the smaller it is, the better the model, the formula is stated in Equation 2. The root mean square error (RMSE) measures the average difference between a statistical model's predicted values and the actual values, the formula is stated in Equation 3. R^2 is a score to see how much the independent variable to the dependent variable, R^2 had value between 0 to 1, where 1 is the best and 0 is not, the formula is stated in Equation 4. Y_1 as the real value, $\hat{y}$ as the result of predicted data, n the numbers of data which been observed.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \quad (1)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \quad (2)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \quad (3)$$

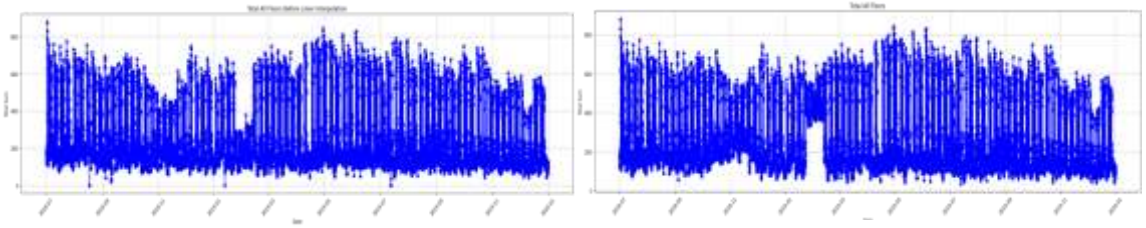
$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \quad (4)$$

3. Results and Discussion

3.1. Preprocessing

The data from the dataset are in 1-minute intervals, which need to be resampled to one hour, to reduce the length so it will be easier for the model to learn the pattern. The total data after resampled is 13.176 data. The electricity consumption from all floors will be summed into one and then filled with interpolation linear if had a missing value. Fig.2 (a) shows that there is a missing value at a certain time due to maintenance in the building, the missing value is filled and the result is shown next to before interpolation. From Fig. 2 (b) see that data series which has two see that data series y which has two coordinate points, with coordinates (x_1, y_1) and (x_2, y_3). x_1 and y_1 and are the initial and final time positions in the data series to be interpolated and x is the time position to indicate interpolation at the desired point in the time interval $[x_1, x_2]$. The formula for the standard scaler will be stated in Equation 5.

$$y = y_1 + \frac{(x - x_1)(y_2 - y_1)}{x_2 - x_1} \quad (5)$$



(a) (b)

Fig. 2 Imputing Data Using Linier Interpolation

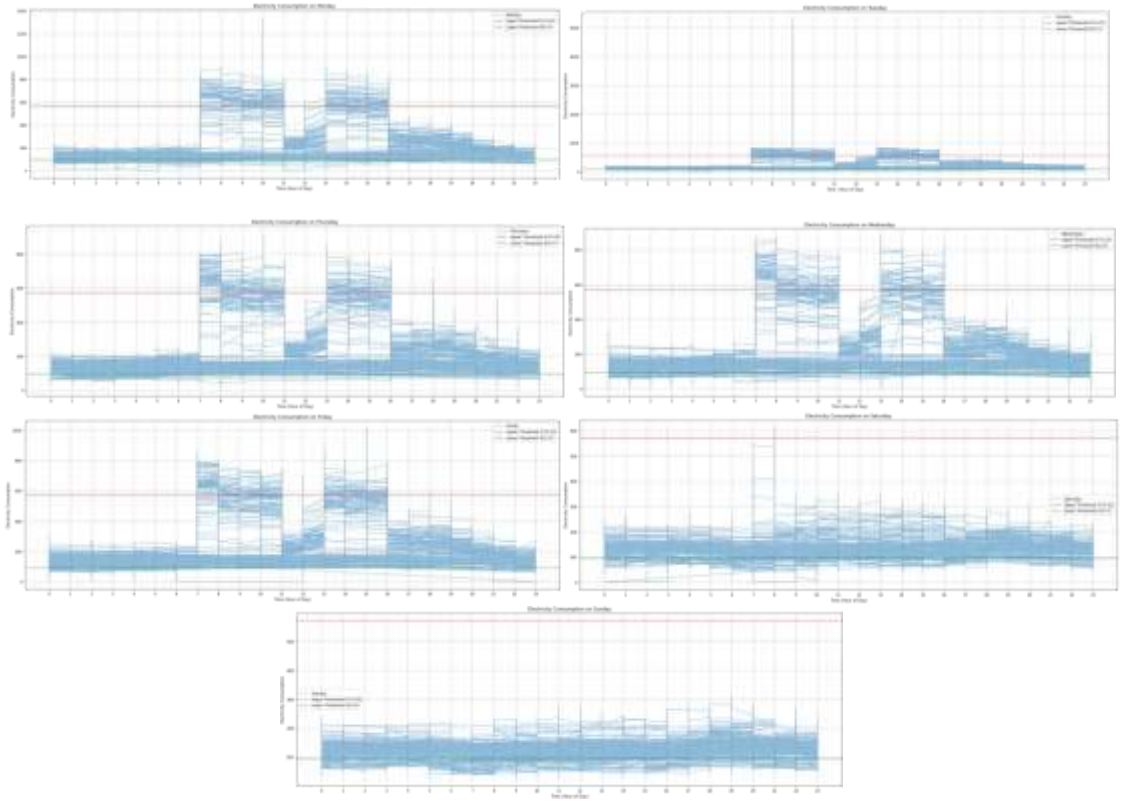


Fig. 3 Set Limit Threshold For Every day

A new feature was added to the dataset related to electricity consumption, the features were divided into a few categories, outdoor weather, national holidays, and the duration when peak demand load in a day. The outdoor weather contains the average temperature, minimal temperature, maximal temperature, and precipitation. The outdoor weather data was recorded in daily data intervals. The dataset had 7 labels type of holiday, and NaN data who considered as non-holiday. The NaN data will be filled as a weekday, after that the label will be encoded into 1 and 0. The weekday and the observance day will be labeled as 1 and the weekend and others as 0.

The duration of peak demand is the duration when the demand for electricity is above the threshold. The threshold is made to calculate the minimum and maximum of the day from total electricity data in a day with percentages of 10% and 90%. To differentiate days the weekdays and the weekend days by featuring weekdays. The threshold for the upper limit is 571, 03 and the lower limit is 94,27. The figure of the threshold every day is shown in Fig. 3. After that the upper limit will be set as True and False then encoded into 1 and 0. Dataset electricity consumption was resampled into 1-hour intervals, the total data after the interval is 13.176 data. The new feature will be merged into one electricity consumption dataset.

3.2. Feature Selection

The selection feature used Pearson's correlation coefficient to see the best correlation among the features set in the dataset. The highest value in correlation is the duration of the peak demand because it'll have the most influence on electricity consumption with a value of 0.75. The least influence on electricity consumption is the precipitation of the day with a value of 0.034. The correlation matrix is shown in Fig. 4. From Fig. 4 the duration of peak demand had the strongest linear relationship to electricity consumption with a positive correlation, caused when the electricity demand increases, the duration feature will capture as a peak and will calculate how longthe peak takes until a threshold is determined.

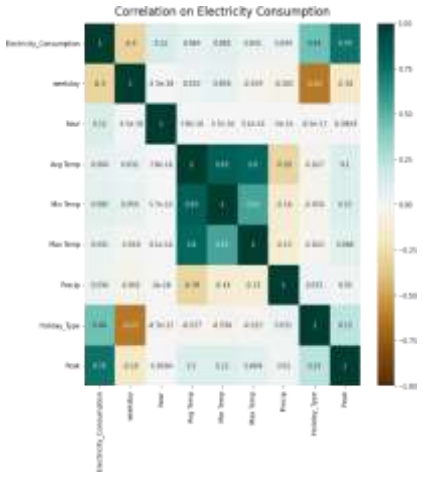


Fig. 4 Correlation Features to Electricity Consumption

The holiday type feature had a value of 0.44 and is considered as a moderate correlation to electricity consumption with a positive value, because of the people's behavior and usage pattern when the holiday came the usage of electricity tended to decrease than the regular days. The outdoor weather feature had a very weak positive correlation, which means there was a slight tendency between the two variables, but the relationship is not strong. The value is 0.082 for the minimal temperature, 0.063 for the average temperature, 0.041 for the maximal temperature, and 0.034 for the precipitation. The temperature feature had a slight increase due to a slight increase in electricity consumption, but not enough conclusion that the temperature influences the electricity consumption. The weekday and hour feature for showing the present time, the hour feature had a positive but weak correlation cause the value is around 0.11, it did not have enough influence on the electricity consumption feature. The weekday feature had a value of -0.030 which means it has a negative and weak correlation to electricity consumption. The negative correlation means that electricity consumption tends to be lower on weekend than the weekdays, cause the work days.

3.3. Model Evaluation

The experiment uses splitting data with the size of the training set at 90% and the test set at 10%. The experiment compares splitting data using time-series cross-validation and holdout. The time-series cross-validation function will split data based on test and training size and divide it into n_folds. The function's method of data splitting is shown in Fig. 5. Every fold will do a test in the dataset based much n_data for every fold, so the data will gradually changing to much as much as the fold increases. The benefit of using time-series cross-validation splitting data is could maintain the data temporal order of data and ensure the validation set will not come up before the training set. With time-series cross-validation model will learn the pattern better. Every fold in training will learn the pattern based on the last fold. In Fig. 4b the second fold could not predict the pattern cause in the first fold there is no pattern with small data, but in the third fold in Fig. 4c until Fig. 4e the model gradually learned the pattern better than the holdout. Time-series cross-validation made robustness for the model to perform in all folds.

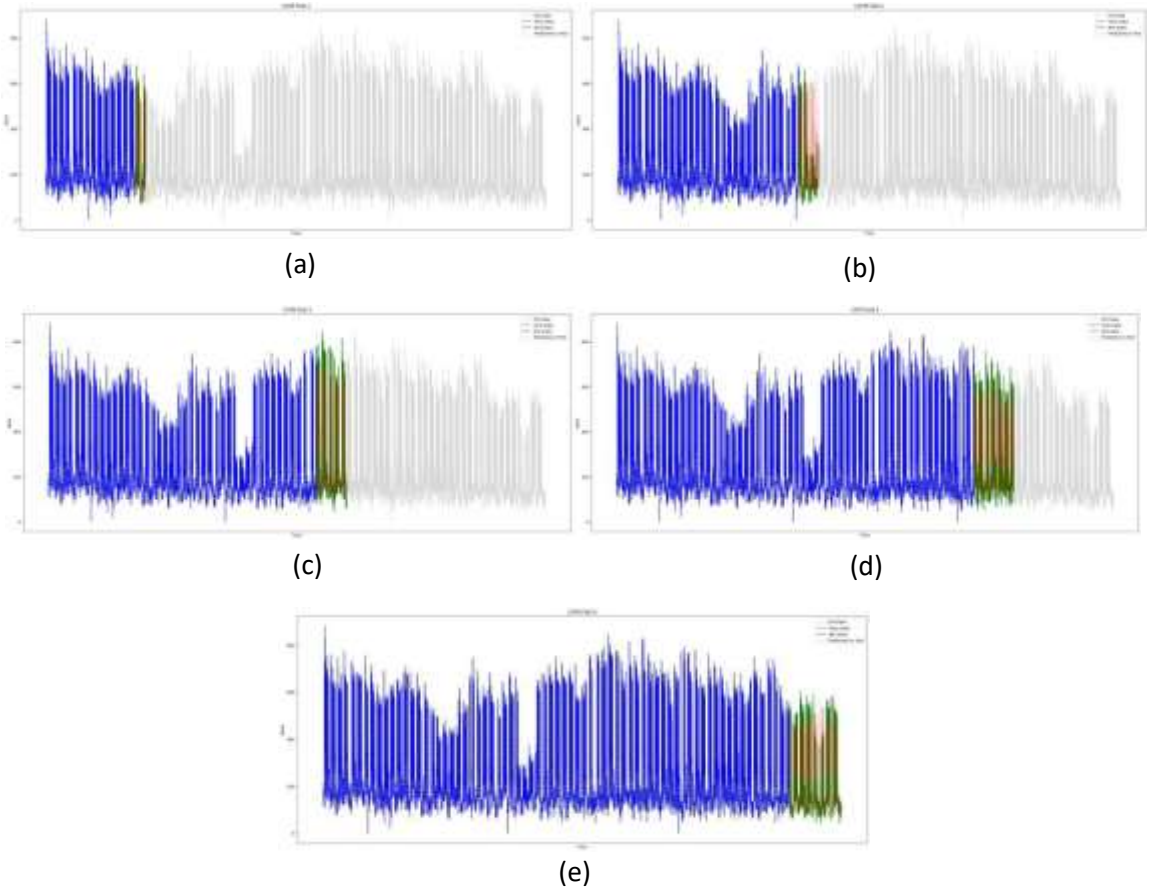


Fig. 5 Split Data Using Timeseries Cross Validation

The experiment is done with several scenarios of set feature combinations and methods of splitting data. The feature set combination had 8 features based on the strongest correlation and combining all features. The first scenario is using 1 feature which contains electricity consumption. Second, adding the present time which is the day and the hour, made it have 3 features. The third is adding outdoor weather. Fourth is adding a holiday national type. Five is adding the last feature, duration when it is peak demand. The sixth scenario is using the electricity consumption with the strongest correlation positive, duration when it is peak demand. The seventh scenario uses electricity consumption with, duration when it is peak demand and holiday type. The last scenario is to using the present time, duration when it is peak demand, and holiday type. The last scenario to see the relationship between the temporal features with the best correlation features. The list of features set will seen in Table 1.

Evaluation in this experiment will choose the best model based on R^2 for predicting electricity consumption by using lookback as a key variable. The aim is to see how different amounts of historical data used affect prediction results for different horizons. The set features in the experiment will add the reason as the indicator to the seasonal trends and external variables' relationships to electricity consumption. By evaluating the model across multiple horizons and different sets of features. we ensure that the selected models are robust an effective for varying forecast length. Horizon used in this experiment is 1, 10, 24, 72, and 168. Each horizon will represent the best model in the experiment, in the experiment, the model used is LSTM, GRU, CNN-LSTM, and CNN-GRU.

Table 1. List of Features

Features Set	*Variable*
El_Consumption	A
El_Consumption, Weekday, Hour	B
El_Consumption, Weekday, Hour, Avg_Temp, Min_Temp, Max_Temp, Precip	C
El_Consumption, Weekday, Hour, Avg_Temp, Min_Temp, Max_Temp, Precip, Holiday_Type	D
El_Consumption, Weekday, Hour, Avg_Temp, Min_Temp, Max_Temp, Precip, Holiday_Type, Peak	E
El_Consumption, Peak	F
El_Consumption, Peak, Holiday_Type	G
El_Consumption, Weekday, Hour, Peak, Holiday_Type	H

Table 2 shows the result of the evaluation from lookback value 2. The table shows the best score value for each horizon on different splitting data. From the table can be seen that the combination features electricity consumption, weekday, and hour had the best results from all horizons. CNN-GRU with R^2 value 0.971. CNN-GRU had the best result from all models that had been tried the prove robust performance from all different feature sets and horizons. From the experiment can be seen that the R^2 in splitting data holdout gradually decreased as the horizon increased. The result of the best R^2 using LSTM model with value 0.939. The R^2 on splitting data time-series cross-validation is decreased due to the long sequence. For the error value from all experiments, the best result is in the smallest horizon, because it has more data to learn than others. Which can be seen that horizon value 1 had the smallest error than horizon value 168. Splitting data using holdout had less error than the time-series cross-validation. The holdout made an insight clear into performance across different scenario features set and the time-series cross-validation indicating the model's potential stability.

Table 2. Evaluation on Lookback 2

Splitting Data	*Feature Set*	*Horizon*	*Model*	*R^2*	*MAE*	*MSE*	*RMSE*
Holdout	B	1	CNN_GRU	0.971	15.139	618.184	24.863
	H	10	GRU	0.918	29.167	1944.406	44.095
	H	24	CNN_GRU	0.841	36.556	3639.408	60.328
	H	72	GRU	0.780	43.247	5165.030	71.868
	B	168	CNN_GRU	0.770	44.011	5191.006	72.049
Time-series Cross Validation	B	1	CNN_LSTM	0.939	20.489	1291.577	33.454
	E	10	CNN_GRU	0.861	38.117	3573.080	58.428
	H	24	CNN_LSTM	0.817	41.444	4724.071	67.304
	B	72	CNN_LSTM	0.752	50.767	6607.401	80.090
	H	168	CNN_LSTM	0.769	51.693	6468.947	78.948

As the lookback increases into value 4 shown in Table 3, features set electricity consumption, weekday, and hour, still had the best result among all features set. CNN_GRU model in horizon value 1 had the best result with R^2 of 0.973. The performance of the model generally declined as the horizon value was longer in the holdout method, can be seen from the R^2 the result consistently decreased and the error became bigger.

In time-series cross-validation LSTM model with horizon value 1 had the best result with the R^2 0.942, but the error was a little bit high proving that maybe had overfitting through the model due to the long epoch, but it caught the pattern well. From all results in lookback value 4 CNN-GRU had the best results. From the experiment in splitting data time-series cross-validation had an error value increasing rapidly.

Table 3 Evaluation on Lookback Value 4

Splitting Data	*Features set*	*Horizon*	*Model*	*R2*	*MAE*	*MSE*	*RMSE*
Holdout	H	1	CNN_GRU	0.973	13.867	562.258	23.712
	H	10	LSTM	0.923	27.933	1830.946	42.790
	H	24	GRU	0.852	34.292	3340.895	57.800
	H	72	CNN_LSTM	0.792	39.717	4674.638	68.371
	H	168	CNN_GRU	0.777	43.256	5114.279	71.514
Time-series Cross Validation	B	1	LSTM	0.942	20.065	1275.366	33.742
	E	10	CNN_GRU	0.872	36.200	3209.142	55.432
	H	24	CNN_LSTM	0.827	40.754	4504.376	65.930
	H	72	CNN_GRU	0.771	48.578	6198.598	77.324
	H	168	CNN_GRU	0.774	51.051	6371.962	78.158

In another experiment using the lookback value 8 shown in Table 4, the performance splitting data holdout in horizon value 1 had the best result. LSTM had the best R^2 with a value of 0.978 among all horizons on splitting data holdout. Splitting data holdout results is the increasing value of horizon made an error became bigger and the performance of models became decreased. From the results can be seen LSTM in horizon value 1 had the best R^2, with a value of 0.963, and had the least error among all horizons. Features set electricity consumption, weekday, hour, holiday_type, and peak still had the best combination result. LSTM model had the best result among all, especially in small horizon.

Table 4. Evaluation on Lookback Value 8

Splitting Data	*Features set*	*Horizon*	*Model*	*R^2*	*MAE*	*MSE*	*RMSE*
Holdout	H	1	LSTM	0.978	13.046	455.209	21.336
	H	10	LSTM	0.926	25.984	1686.362	41.065
	H	24	GRU	0.855	33.775	3159.914	56.213
	H	72	CNN_GRU	0.789	39.465	4469.934	66.858
	B	168	GRU	0.780	40.967	4694.473	68.516
Time-series Cross Validation	B	1	LSTM	0.963	17.818	880.778	29.174
	E	10	GRU	0.874	35.504	3123.575	54.656
	H	24	GRU	0.842	39.011	4179.094	63.533
	H	72	CNN_LSTM	0.781	47.881	5993.864	76.010
	H	168	GRU	0.782	50.085	6127.046	76.857

The experiment increased the lookback value to 10, the results are shown in Table 5. The result for splitting data holdout had the best R^2 of 0.979 using the LSTM model in horizon value 1. The best result seen was that a high R^2 led to minimal small errors. As from the last experiment in splitting data holdout if the horizon increases the performance model will decrease too. It is shown in horizon 168 that the R^2 value is just 0.780 and the error became bigger. splitting data time-series cross-validation best R^2 value 0.970 in horizon value 24 with model CNN-GRU. As the horizon value increased the model performance slightly became smaller than the small lookback and the error became a little bit bigger too.

The increasing value of lookback improved the smaller horizon as the historical data became larger but it became an issue in the larger horizon, cause may lead to bigger errors and overfitting to models. It's seen in the error value slowly bigger. Features set electricity consumption, weekday, hour, holiday_type, and peak still dominated the best combination result to performance for every horizon value.

Table 5 Evaluation on Lokkback Value 10

Splitting Data	*Features set*	*Horizon*	*Model*	*R^2*	*MAE*	*MSE*	*RMSE*
Holdout	H	1	LSTM	0.979	13.733	451.754	21.255
	H	10	LSTM	0.936	24.657	1436.759	37.905
	H	24	LSTM	0.858	33.254	3029.837	55.044
	H	72	CNN_LSTM	0.800	38.396	4410.716	66.413
	H	168	CNN_GRU	0.780	42.175	4842.332	69.587
Time-series Cross Validation	B	1	LSTM	0.970	17.446	755.712	27.155
	H	10	CNN_GRU	0.891	32.552	2686.958	49.865
	H	24	CNN_GRU	0.857	38.300	3851.577	61.111
	H	72	GRU	0.785	47.718	5919.686	75.481
	H	168	GRU	0.782	50.085	6140.724	76.842

The lookback value is increasing to use a day data with value 24, shown in Table 6. Splitting data holdout had the best R^2 value of 0.984 with the LSTM model and horizon value 1, indicating the lowest error in the experiment. Feature set B still dominates the splitting data holdout as the best feature combination. In splitting data time-series cross-validation the LSTM model with horizon value 1 had the best performance error with R^2 value of 0.975. Tmodel performance gradually decreased in both method of splitting data. The decreasing value R2 and error matrix became bigger because the experiment has a sequence longer. So, when the first fold for processing the train became smaller the learning model did not catch the pattern better.

Table 6. Evaluation on Lookback Value 24

Splitting Data	*Features set*	*Horizon*	*Model*	*R^2*	*MAE*	*MSE*	*RMSE*
Holdout	B	1	CNN_LSTM	0.984	12.422	340.594	18.455
	H	10	GRU	0.950	21.277	1046.164	32.344
	H	24	LSTM	0.873	30.899	2528.630	50.285
	B	72	GRU	0.803	36.717	3997.923	63.229
	B	168	LSTM	0.771	41.574	4534.072	67.336
Time-series Cross Validation	H	1	LSTM	0.975	16.394	646.713	25.083
	H	10	CNN_GRU	0.920	29.193	2020.321	44.017
	H	24	CNN_LSTM	0.864	38.039	3703.296	59.939
	H	72	GRU	0.788	49.007	5958.995	75.448
	B	168	GRU	0.782	49.953	6148.201	77.015

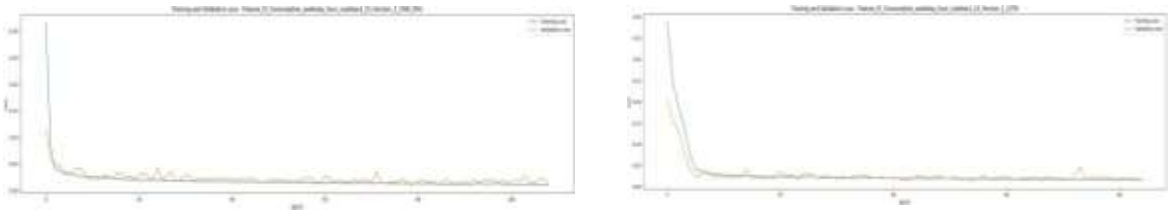
As the lookback value became bigger, can be said that the learning pattern became much clearer. The bigger value of lookback and the smaller horizon made the performances better. The splitting data holdout maintains a good performance due to the size of the training set being the same for all experiments. It is seen from Table 7 that the R^2 on splitting data holdout had silghtly better result as the lookback incereased, with the lowest error value. In the time-series cross- validation, the R^2 value gradually decrease but the error became a little bit higher than as the increased of horizon.

Table 7. Evaluation on Lookback Value 72

Splitting Data	*Features set*	*Horizon*	*Model*	*R^2*	*MAE*	*MSE*	*RMSE*
Holdout	H	1	CNN_GRU	0.987	12.263	295.496	17.190
	H	10	GRU	0.955	21.126	932.072	30.530
	H	24	CNN_LSTM	0.870	31.976	2741.579	52.360
	H	72	CNN_LSTM	0.804	38.033	4406.445	66.381
	B	168	GRU	0.775	40.142	4049.738	63.638
Time-series Cross Validation	H	1	CNN-LSTM	0.973	16.502	654.498	24.857
	H	10	GRU	0.926	29.362	1997.727	43.554
	B	24	CNN-GRU	0.850	41.377	4141.962	62.955
	B	72	CNN-LSTM	0.782	47.634	6155.165	76.578
	B	168	GRU	0.775	52.283	6355.911	78.059

The experiments aim to see the performance model by several values of lookbacks and horizons. It appears that using the splitting data holdout method the problem is the increasing horizon value made the performance model became poor. This happens due to the small dataset to learn and the patterns of the data become more complex. However, it can be fixed by adding more lookback value so the accuracy of model performance will increase. This matters because will provide more historical data to be learned by the model and will enhance the performance.

The result of the time-series cross-validation method is the opposite of splitting data holdout, for maintaining the model performance, despite the error matrix value becoming bigger as the horizon increases, the model stabilizes during the training and validation. It can be seen in Fig.6, using timeseries more stabilizing than the holdout. The time-series better in handle variability and complex data than the holdout. This happens because the number of folds will affect the total data in the first fold, this made the model's ability to learn the data patterns limited in the next folds. The larger the number of folds the larger data will split into smaller subsets. The limited dataset will lead model performance to become prone to overfitting and have worse accuracy. Overfitting occurs when the model learns will catch the noises and fluctuations in the training data rather than learn the patterns, so the results had poorer generalization in the new data in the next folds. It caused the errors to become progressive and add more errors to the next folds.



(a) (b)

Fig. 6 Plot Loss Holdout (a) Time-Series Cross-Validation (b)

The feature set B and H has the best results among other combinations, although feature set B it has the least positive correlation with electricity consumption. It shows that the feature that had capturing-based time, made a lot of improvement than using just the feature with a strong correlation. The features will provide an understanding of electricity consumption cycles, based on daily or weekly. The Pearson's correlation measures the linear relationships between the pairs of variables. A feature with a low correlation to the target does not mean not important feature. In this case, using the low correlation features combining with strong correlation features can capture the underlying patterns better than other features. Features set H proving that with combinations of the strong correlation and the temporal fetures will had best perfomancies on R^2, MAE, MSE, and RMSE.

The model used in the research has the result using combinations of CNN and RNN models makes it easier for the model to learn due to the smaller dimensions of the input shape. It can be seen that the combination using CNN-GRU has the best performance results among the experiments. Even though many of the features used by CNN-GRU are still superior to other models, the model can recognize complex features in electricity consumption data. LSTM is good for models with the least features because it makes the learning pattern not too complex but using features that are related to the target. The ability of both models in performances to predict multihorizon is highly effective, due to the robustness of avoiding noise.

The variation in the results of the experiment leads to the differing ability of the model to capture temporal dependencies and patterns in the data. These made that finding the importance of selecting the right method to understand the needs of the data will affect the accuracy of electricity consumption prediction. However, the prediction of electricity consumption is not solely based on the model itself, but with the right preprocessing, features, and time window will increase the performance of the model prediction.

4. Conclusion

Using method the time-series cross-validation as splitting data made better performance by learning patterns better than the holdout method. This can be seen in the plot that using time-series cross validation had better result on the model performances. The holdout method is a good splitting data if the horizon is fixed with a bigger lookback, cause it can catch the patterns better. The bigger horizon and smaller lookback will cause the model performances to become worse. So, it needs a fixed value in the lookback and the horizon. The combination of set features for training had proven the strong correlation value would not had a better result if the features did not catch the patterns, it better had a feature which had a good relation than a good correlation. The impact of combining the strong correlation might be not effective, giving less information to the model and making noises to the model. Combining features which had the strongest correlation with temporal features will had the best result among all combinations.

Multistep has emerged due to the advantages of predicting a few multi-steps ahead. The challenge in multisteps is to find the right sequence to prevent overfitting in the model. This experiment found that combining the RNN and CNN made the performance of the model improve rapidly, but the wrong sequence made the model error become bigger and led to the loss became NaN. It will cause the gradient to explode or vanish. Multistep forecasting can made the error propagate and accumulate into the last prediction so it will lead to the error becoming bigger. Proven by the experiment the MAE, MSE, and RMSE had bigger result values if the horizon is increasing. LSTM better at catching pattern on long sequence with small time-window for prediction.

From this experiment can be identified that the deep learning did not catch well long temporal dependencies cause the limited caches. Future research can be explored to use another method like attention or transfer learning, so it can catch better temporal dependencies. Advancing the data augmentation to enrich the training dataset, so it will make robustness to the model. Checking the multicollinearity in features so it's not redundant. Through this experiment, another research can be done with the same dataset so it will produce another insight with a betterapproach.

Declarations

Author contribution: APP Data Curation, Writing – Original Draft, Validation, and Investigation. RVHG Conceptualization, Methodology, Formal analysis, Writing – Review & Editing, and Supervision. AS Conceptualization, Methodology, Formal analysis, Writing – Review & Editing, and Supervision.
Funding statement: There was no external funding involved in this project. **Conflict of interest:** There is no conflict of interest was declared by author. **Additional information:** No additional information is available for this paper.

References

[1] C. Lu, S. Li, and Z. Lu, "Building energy prediction using artificial neural networks: A literature survey," *Energy Build.*, vol. 262, p. 111718, 2022.

[2] J. Cifuentes, G. Marulanda, A. Bello, and J. Reneses, "Air temperature forecasting using machine learning techniques: a review," *Energies*, vol. 13, no. 16, p. 4215, 2020.

[3] X. Liu, L. Yang, and Z. Zhang, "Short-term multi-step ahead wind power predictions based on a novel deep convolutional recurrent network method," *IEEE Trans. Sustain. Energy*, vol. 12, no. 3, pp. 1820–1833, 2021.

[4] S. Athiyarath, M. Paul, and S. Krishnaswamy, "A comparative study and analysis of time series forecasting techniques," *SN Comput. Sci.*, vol. 1, no. 3, p. 175, 2020.

[5] R. V Klyuev *et al.*, "Methods of forecasting electric energy consumption: A literature review," *Energies*, vol. 15, no. 23, p. 8919, 2022.

[6] A. A. Pierre, S. A. Akim, A. K. Semenyo, and B. Babiga, "Peak electrical energy consumption prediction by ARIMA, LSTM, GRU, ARIMA-LSTM and ARIMA-GRU approaches," *Energies*, vol. 16, no. 12, p. 4739, 2023.

[7] M. Hasanov, M. Wolter, and E. Glende, "Time Series Data Splitting for Short-Term Load Forecasting," in *PESS+ PELSS 2022; Power and Energy Student Summit*, 2022, pp. 1–6.

[8] S. Karasu, A. Altan, S. Bekiros, and W. Ahmad, "A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series," *Energy*, vol. 212, p. 118750, 2020.

[9] S. J. Fong, G. Li, N. Dey, R. G. Crespo, and E. Herrera-Viedma, "Finding an accurate early forecasting model from small dataset: A case of 2019-ncov novel coronavirus outbreak," *arXiv Prepr. arXiv2003.10776*, 2020.

[10] B. Nepal, M. Yamaha, A. Yokoe, and T. Yamaji, "Electricity load forecasting using clustering and ARIMA model for energy management in buildings," *Japan Archit. Rev.*, vol. 3, no. 1, pp. 62–76, 2020.

[11] X. M. Zhang, K. Grolinger, M. A. M. Capretz, and L. Seewald, "Forecasting residential energy consumption: Single household perspective," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 110–117.

[12] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, 2021.

[13] A. Mishra, H. R. Lone, and A. Mishra, "DECODE: Data-driven energy consumption prediction leveraging historical data and environmental factors in buildings," *Energy Build.*, vol. 307, p. 113950, 2024.

[14] X. J. Luo, L. O. Oyedele, A. O. Ajayi, O. O. Akinade, H. A. Owolabi, and A. Ahmed, "Feature extraction and genetic algorithm enhanced adaptive deep neural network for energy consumption prediction in buildings," *Renew. Sustain. Energy Rev.*, vol. 131, p. 109980, 2020.

[15] D. A. Musleh and M. A. Al Metrik, "Machine Learning and Bagging to Predict Midterm Electricity Consumption in Saudi Arabia," *Appl. Syst. Innov.*, vol. 6, no. 4, p. 65, 2023.

[16] M. Jayashankara, P. Shah, A. Sharma, P. Chanak, and S. K. Singh, "A novel approach for short-term energy forecasting in smart buildings," *IEEE Sens. J.*, vol. 23, no. 5, pp. 5307–5314, 2023.

[17] M. Sangiorgio and F. Dercole, "Robustness of LSTM neural networks for multi-step forecasting of chaotic time series," *Chaos, Solitons & Fractals*, vol. 139, p. 110045, 2020.

[18] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of deep learning models for multi-step ahead time series prediction," *Ieee Access*, vol. 9, pp. 83105–83123, 2021.

[19] K. J. Miller and S. J. C. Venditto, "Multi-step planning in the brain," *Curr. Opin. Behav.*

Sci., vol. 38, pp. 29–39, 2021.

[20] Y. Gao, S. Miyata, and Y. Akashi, "Multi-step solar irradiation prediction based on weather forecast and generative deep learning model," *Renew. Energy*, vol. 188, pp. 637–650, 2022.

[21] K. D. Unlu, "A data-driven model to forecast multi-step ahead time series of Turkish daily electricity load," *Electronics*, vol. 11, no. 10, p. 1524, 2022.

[22] R. Basmadjian, A. Shaafieyoun, and S. Julka, "Day-ahead forecasting of the percentage of renewables based on time-series statistical methods," *Energies*, vol. 14, no. 21, p. 7443, 2021.

[23] S. Hadri, M. Najib, M. Bakhouya, Y. Fakhri, and M. El Arroussi, "Performance evaluation of forecasting strategies for electricity consumption in buildings," *Energies*, vol. 14, no. 18, p. 5831, 2021.

[24] M. F. Alsharekh, S. Habib, D. A. Dewi, W. Albattah, M. Islam, and S. Albahli, "Improving the efficiency of multistep short-term electricity load forecasting via R-CNN with ML-LSTM," *Sensors*, vol. 22, no. 18, p. 6913, 2022.

[25] R. Chalapathy, N. L. D. Khoa, and S. Sethuvenkatraman, "Comparing multi-step ahead building cooling load prediction using shallow machine learning and deep learning models," *Sustain. Energy, Grids Networks*, vol. 28, p. 100543, 2021.

[26] M. Pipattanasomporn *et al.*, "CU-BEMS, smart building electricity consumption and indoor environmental sensor datasets," *Sci. Data*, vol. 7, no. 1, p. 241, 2020.

[27] "Bangkok Weather," 2018. https://www.ncei.noaa.gov/cdo-web/ (accessed Apr. 27, 2024).

[28] "Bangkok National Holiday," 2018. https://www.timeanddate.com/holidays/thailand/ (accessed Apr. 27, 2024).

[29] J. Fan, L. Wu, X. Ma, H. Zhou, and F. Zhang, "Hybrid support vector machines with heuristic algorithms for prediction of daily diffuse solar radiation in air-polluted regions," *Renew. Energy*, vol. 145, no. 2, pp. 2034–2045, 2020, doi: 10.1016/j.renene.2019.07.104.

[30] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 2018, pp. 1394–1401.

[31] W. Shu, K. Cai, and N. N. Xiong, "A short-term traffic flow prediction model based on an improved gate recurrent unit neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16654–16665, 2021.

[32] W. K. H. W. K. Amir, A. B. M. Soom, A. M. Jasin, J. Ismail, and A. Asmat, "Sales Forecasting Using Convolution Neural Network," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 30, no. 3, pp. 290–301, 2023.

[33] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *Peerj Comput. Sci.*, vol. 7, p. e623, 2021.