

Feature engineering and long short-term memory for energy use of appliances prediction

I Wayan Aditya Suranata¹, I Nyoman Kusuma Wardana², Naser Jawas³,
I Komang Agus Ady Aryanto⁴

¹Department of Information Technology, Universitas Pendidikan Nasional, Bali, Indonesia

²Department of Electrical Engineering, Politeknik Negeri Bali, Bali, Indonesia

³Department of Information System, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia

⁴Department of Information Technology, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia

Article Info

Article history:

Received Jun 15, 2020

Revised Aug 23, 2020

Accepted Aug 31, 2020

Keywords:

Appliances

Feature engineering

Long short-term memory

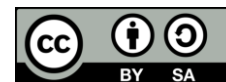
Principal component analysis

Prediction

ABSTRACT

Electric energy consumption in a residential household is one of the key factors that affect the overall national electricity demand. Household appliances are one of the most electricity consumers in a residential household. Therefore, it is crucial to make a proper prediction for the electricity consumption of these appliances. This research implemented feature engineering technique and long short-term memory (LSTM) as a model predictor. Principal component analysis (PCA) was implemented as a feature extractor by reducing the final 62 features to 25 principal components for the LSTM inputs. Based on the experiments, the two-layered LSTM model (composed by 25 and 20 neurons for the first and second later respectively) with lookback number of 3 found to give the best performance with the error rates of 62.013 and 26.982 for root mean squared error (RMSE) and mean average error (MAE), respectively.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

I Wayan Aditya Suranata

Faculty of Engineering and Informatics, Department of Information Technology

Universitas Pendidikan Nasional

Bedugul St. No. 39, Sidakarya, Kecamatan Denpasar Selatan, Kota Denpasar, Bali 80224, Indonesia

Email: aditya@undiknas.ac.id

1. INTRODUCTION

Based on the 2018 PLN Annual Report on connected power by customer segment in Indonesia, households occupy the highest proportion, with a power value of 63,577 megavolts-amperes (MVA) (48.8%) of the total connected power (130,281 MVA). The growth of connected power from 2016 to 2018 was recorded at 7.3%, exceeding the industrial and business segments with values of 5.6% and 5.9%, respectively [1]. These data conclude that energy use from households (or residential homes) is one of the critical factors affecting electricity consumption nationally.

Electrical appliances become one of the most significant sources of electricity use in a residential home. As an illustration, research conducted by Cetin *et al.* found that electrical equipment in a residential home in the United States can consume energy up to 30% of the total electricity demand [2]. Since the use of household appliances highly affect the total electrical energy consumption in a residential home, the prediction regarding the use of electrical energy for household appliances is an essential work [3].

There are various studies related to the prediction of the energy use by appliances, one of them conducted by Candanedo *et al.* [4]. Candanedo *et al.* implemented four different predictors to forecast the electricity consumption from a residential home, namely the linear regression model (LM), support vector

machine (SVM), random forest (RF) and gradient boosting machine (GBM). Our work refers to the research conducted by Candanedo *et al.* and we have done preliminary research as reported in [5]. However, compared to Candanedo *et al.*, we implemented different methods of predicting electricity use. We developed long short-term memory (LSTM) as a model predictor and applied principal component analysis (PCA) to perform the feature extraction process. We performed a feature engineering process in addition to the initial dataset. Candanedo *et al.* randomly divided the dataset as follows: 75% as training data and the rest 25% as test data. Instead of splitting dataset randomly, we maintain the sequence of each division (sequence-to-sequence prediction). In our work, we divided 60% of the dataset for training data, 20% for validation data, and 20% for test data.

LSTM [6] is a structural modification of the recurrent neural network (RNN) by adding memory cells in the hidden layer so that it can be applied to control the flow of information in a time-series data [7]. The data predicted in this study are classified as time-series data. Time-series data is a series of data that is observed based on a specific time interval. Time-series data can be implemented in various applications, such as regression, classification, and clustering [8]. LSTM has an excellent ability in predicting cases involving time series [9, 10]. Besides being implemented in the case of time-series, examples of other applications such as handwriting recognition [11], text classification [12], data intrusion in computer networks [13], and various other types of applications have been actively explored. LSTM can also be combined with other neural network models to improve performance [7, 14, 15].

Principal component analysis (PCA) technique can reduce the dimensions of the input data before these features are fed to the predictor model. Principal component analysis [16] is known as a technique of reducing dimensions, which transforms the initial data into the principal component space through a linear projection [17]. Due to its applicability and simplicity, PCA has become a popular method nowadays [18] and has an essential role in various applications such as pattern recognition, artificial intelligence, and data mining [19].

The main contributions of our work are the implementation of feature engineering and principal component analysis to the initial dataset for predicting the electricity consumption in a residential home. The feature engineering data were derived from the initial dataset. We expanded the existing dataset almost threefold, from 24 attributes to 62 attributes by implementing frame features, lag features, and window features techniques. To effectively recognise the pattern, the PCA then reduced the input dimension from 62 features to 25 features. These 25 features were then fed to the LSTM predictor.

This paper is organised into four parts. The first part reviews the background of the study. The second part discusses the research method, which includes a description of the data used in the study, an explanation of predictor models, and methods for evaluating the proposed model. The third section explains the selection of the most optimal models as well as the evaluation of the models. Finally, the fourth section summarises the research outcomes.

2. RESEARCH METHOD

2.1. Dataset description

In this study, we used the dataset provided by Candanedo *et al.* [4], which can be downloaded from the University of California, Irvine (UCI) machine learning repository page. Indoor and outdoor data compose the components of the dataset. Indoor data (room temperature and humidity) were collected using a wireless sensor network technique. The consumption of electrical energy from various types of equipment and lighting in a residential home was also included in the dataset. Besides, the dataset is also equipped with outdoor data in the form of weather parameters (pressure, humidity, wind speed, visibility, and dew point) collected from the nearest airport station. Each row of data in the dataset was recorded with intervals of 10 minutes.

For indoor data, several sensors to measure room temperature and humidity transmitted data approximately every 3.3 minutes using the ZigBee protocol, while energy meters for measuring electrical energy consumption collected data every 10 minutes. The temperature and humidity data were then averaged to get 10 minutes intervals. In addition to the main energy meter, there were also sub-energy meters that specifically measure the energy consumption of lighting devices. Data from lighting devices are intended as predictors of room occupancy when combined with relative air humidity. For outdoor data, various weather parameters were collected from the weather station at the nearest airport. Since the measurement of this weather parameter was conducted every hour, a linear interpolation was performed to obtain 10 minutes of data intervals. The dataset consists of 19,735 rows (stating the amount of data) and 28 columns (stating the number of attributes/features). Table 1 shows the initial features downloaded from the UCI machine learning repository page. The more detailed explanations of the dataset used in this experiment can be referred to [4].

Table 1. The dataset features [4]

No	Attribute	Units	Description
1	Date	dd:mm:yyhh:mm:ss	Date and time
2	Appliances	Wh	Energy use
3	lights	Wh	Energy use of light fixtures in the house
4	T1	°C	Temperature in kitchen area
5	RH_1	%	Humidity in kitchen area
6	T2	°C	Temperature in living room area
7	RH_2	%	Humidity in living room area
8	T3	°C	Temperature in laundry room area
9	RH_3	%	Humidity in laundry room area
10	T4	°C	Temperature in office room
11	RH_4	%	Humidity in office room
12	T5	°C	Temperature in bathroom
13	RH_5	%	Humidity in bathroom
14	T6	°C	Temperature outside the building (north side)
15	RH_6	%	Humidity outside the building (north side)
16	T7	°C	Temperature in ironing room
17	RH_7	%	Humidity in ironing room
18	T8	°C	Temperature in teenager room
19	RH_8	%	Humidity in teenager room
20	T9	°C	Temperature in parents room
21	RH_9	%	Humidity in parents room
22	T_out	°C	Temperature outside
23	Press_mm_hg	mmHg	Pressure outside
24	Windspeed	m/s	Wind speed
25	Visibility	km	Visibility
26	Tdewpoint	°C	Dewpoint
27	rv1	-	Random variable 1
28	rv2	-	Random variable 2

As shown in Table 1, the targeted attribute in this work is electrical energy (appliances). The use of electrical energy varies over time. For example, energy usage may vary over different hours in the days, or it may also vary over days in the week. Visualisation of data can provide preliminary information about fluctuations of these features, before moving to quantitative analysis. Based on the time attributes provided in the dataset, it is obtained information that the logging process for the dataset started from January 11, 2016, at 17:00 until May 27, 2016, at 18:00.

From the original 28 features shown in Table 1, Candanedo *et al.* inserted three more features based on the date attribute, namely the number of seconds calculated from midnight for each day (NSM), day status (workweek or weekend) and the names of the corresponding days (monday to sunday). Extracted from date, we added one more feature, namely hour. This attribute helps to maintain information about the sequence of the retrieved data. In this study, the values of *ev1* and *rv2* in the original dataset were excluded from the further process. In the next following section, we will explain that there is an initial screening process by removing the attributes with a small correlation coefficient in association with Appliances. We will also justify that from the original 28 attributes, only 24 attributes are required, and 38 other features resulted from feature engineering technique are added. Therefore, there are 62 attributes involved in the further process.

Figure 1 (a) depicts variations in the use of electrical energy for the whole period, whereas a detailed review of the electrical energy used during the first week can be observed in Figure 1 (b). In addition, Figure 2 provides visual statistics of the dataset in the form of histogram frequency and boxplots. Based on the frequency histogram in Figure 2 (a), we conclude that the majority of electrical energy usage is at a value of less than 200 Wh. The highest amount of electrical energy usage is 1080 Wh, whereas the lowest is 10 Wh. The use of electrical energy is also known to vary over time of day, as shown in Figure 2 (b). The pattern of energy use starts to rise from 08.00 to 21.00, then decreases from 22.00 to 07.00. The highest consumption is at 17:00 and 18:00. Based on Figure 2 (c), the amount of electricity consumption on weekends (Saturday and Sunday) are higher than the working day. Electricity consumption is relatively stable every month, as shown in Figure 2 (d). In this study, from a total of 19,735 rows in the dataset, we divided the data as follows: 60% (11,841 rows) as training data, 20% (3,947 rows) as validation data and the remaining 20% (3,947 rows) as test data. We kept the order of this data sequence without randomising process. Thus, the characteristics of the time-series data are maintained.

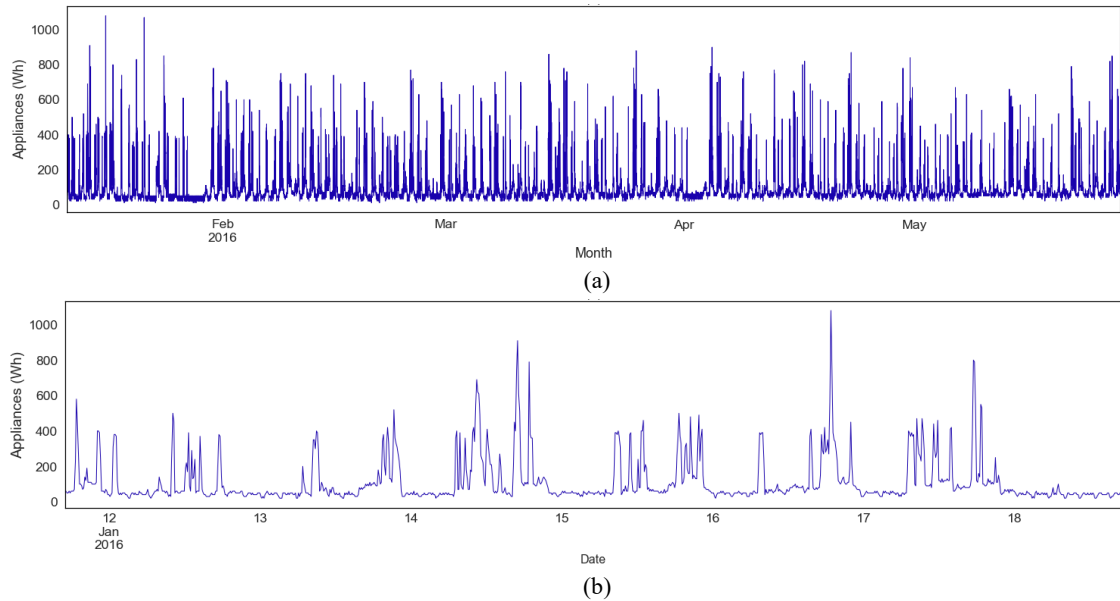


Figure 1. (a) The pattern of electricity consumption for the whole period, and (b) A more detailed review of the pattern of energy consumption during the first week

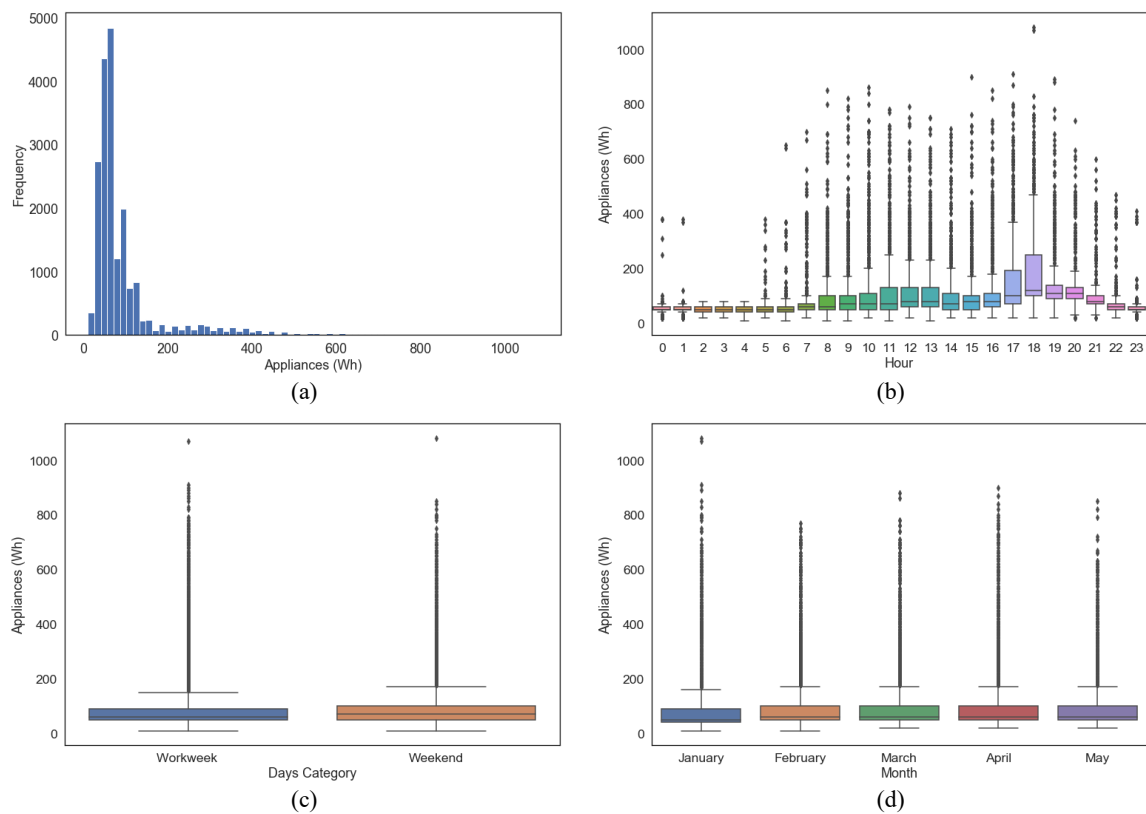


Figure 2. (a) Histogram of energy consumption frequencies, (b) Variations in energy consumption with hours, (c) Variations in energy consumption with days, and (d) Variations in energy consumption with months

2.2. Correlation analysis

The next step is the process to investigate the interrelationships between features, one of which is by conducting correlation analysis. Correlation analysis can provide information about the correlation of two

time-series data. If a time series data is vectored as $X = (x_1, x_2, \dots, x_n)$ and there is another vector $Y = (y_1, y_2, \dots, y_n)$, then the correlation coefficient r of the two vectors is calculated using the following equation [20]:

$$r = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sqrt{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \sqrt{n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2}} \quad (1)$$

The value of r in (1) is also known as the Pearson's correlation coefficient. When $0 < r < 1$, it is said that both features have a positive correlation, and when $-1 < r < 0$ it is said to be a negative correlation. A value of 0 indicates that there is no correlation between features. When the absolute value of r approaches 1, then both features have stronger correlations. It means that value r of 1 indicates that two series of data are identical. Table 2 shows the correlations coefficient of some features in the dataset.

As shown in Table 2, there is a positive correlation between the consumption of electrical energy by various appliances (appliances) and the use of lighting devices (lights). Similarly, T1 and RH_1 have a positive correlation to Appliances, although the correlation is low. The same correlation is also seen between the outside air temperature (T_out) and wind speed (WindSpeed). On the contrary, RH_9 and WeekStatus have a negative correlation. The negative correlation is reasonable as the use of electrical equipment increases when all occupants are staying at home during the holidays. More detailed explanations of the relationship between features can be found in [4]. In this study, features with a correlation value of less than 0.005 with reference to Appliances will be removed. In this case, the Visibility attribute is excluded in the next process because it only has a value of $r = 0.00023$.

Table 2. The selected correlation coefficients for some features

	Appliances	lights	T1	RH_1	T9	RH_9	T_out	Windspeed	Visibility	WeekStatus
Appliances	1.00	0.20	0.06	0.09	0.01	-0.05	0.10	0.09	0.00	-0.02
lights	0.20	1.00	-0.02	0.11	-0.16	0.00	-0.07	0.06	0.02	0.05
T1	0.06	-0.02	1.00	0.16	0.84	0.07	0.68	-0.09	-0.08	-0.01
RH_1	0.09	0.11	0.16	1.00	0.12	0.76	0.34	0.20	-0.02	0.02
T9	0.01	-0.16	0.84	0.12	1.00	0.00	0.67	-0.18	-0.10	0.01
RH_9	-0.05	0.00	0.07	0.76	0.00	1.00	0.22	0.24	0.00	-0.03
T_out	0.10	-0.07	0.68	0.34	0.67	0.22	1.00	0.19	-0.08	-0.04
Windspeed	0.09	0.06	-0.09	0.20	-0.18	0.24	0.19	1.00	0.00	-0.09
Visibility	0.00	0.02	-0.08	-0.02	-0.10	0.00	-0.08	0.00	1.00	0.06
WeekStatus	-0.02	0.05	-0.01	0.02	0.01	-0.03	-0.04	-0.09	0.06	1.00

2.3. Feature engineering

In this study, the input dimension will be raised higher than the dimension available in the original dataset through a process known as feature engineering. The feature engineering technique is processed by synthesising new features from existing dataset to improve the performance of the predictor model [21, 22]. Feature engineering used in this study can be categorised into three categories, namely data frame features, lag features and window features. Frame feature data were extracted from the date attribute. From this attribute, sampling time can be determined. For example, the description of hours, number of minutes and number of seconds of each data can be extracted from the date. Another example of frame feature data is that the status of the day (workweek or weekend) can be easily decided. We also included lag feature attributes, e.g. to predict the value of appliances at $t+1$, then the value of $t-1, t-2, \dots, t-n$ can be included in the modelling process. Window features are related to the information taken from past data, e.g. the average of appliances for the last 30 minutes, or the maximum and minimum values of appliances in the last 2 hours, and so on. Table 3 summaries these auxiliary features in addition to the original dataset.

The total attributes involved in the modelling are 62 features, of which 24 were taken from the original dataset (by excluding date, visibility, rv1, and rv2 in the calculation), and 38 features yielded from the feature engineering process. A total of 62 of these features will be processed using PCA before entering the predictor model, which is the LSTM model. The LSTM input with 62 features is considered as a high dimensionality input. Therefore, we need to reduce this input dimension to a lower dimension.

2.4. PCA

PCA reduces the number of predictor variables and transforms them into new variables, known as principal components (PCs) [23]. The purpose of PCA is to find data summaries only by using a limited number of PCs. To find the proper dimension, the process to evaluate the cumulative variance of principal components is needed. The first PC value is selected to minimise the total distance between data and their projection to

the PC. By minimising this distance, the variance will also be maximised. The rest of the PCs are also chosen with the same concept, but with the condition that there is no correlation between the current PC and the previous PCs [24]. Technically, the number of variants maintained by each PC is measured using eigenvalue. If it is assumed that the initial matrix has the dimension d with n observations, and it is desirable to reduce the dimension to k , then the transformation is written as [25]:

$$y = E_{d \times k}^T \cdot X_{d \times n} \quad (2)$$

where $E_{d \times k}$ is the projection matrix with k eigenvectors and $X_{d \times n}$ is the mean-centred data matrix.

Table 3. Auxiliary features resulted from the feature engineering process

No	Attribute	Description	Category
1	hour	Hour of day (0 to 23)	Data frame feature
2	NSM	Number of seconds counted from midnight	Data frame feature
3	WSt	Week status (workday or weekend)	Data frame feature
4	DoW	Days of the week (Monday to Sunday)	Data frame feature
5	lagApp10	The value of Appliances (Wh) for the past 10 minutes	Lag feature
6	lagApp20	The value of Appliances (Wh) for the past 20 minutes	Lag feature
7	lagLight10	The value of Lights (Wh) for the past 10 minutes	Lag feature
8	lagLight20	The value of Light (Wh) for the past 20 minutes	Lag feature
9	meanApp30	The mean value of Appliances (Wh) for the past 30 minutes	Window feature
10	meanApp60	The mean value of Appliances (Wh) for the past 1 hour	Window feature
11	minApp30	The minimum value of Appliances (Wh) for the past 30 minutes	Window feature
12	minApp60	The minimum value of Appliances (Wh) for the past 1 hour	Window feature
13	maxApp30	The maximum value of Appliances (Wh) for the past 30 minutes	Window feature
14	maxApp60	The maximum value of Appliances (Wh) for the past 1 hour	Window feature
15	meanLight30	The mean value of light (Wh) for the past 30 minutes	Window feature
16	meanLight60	The mean value of light (Wh) for the past 1 hour	Window feature
17	minLight30	The minimum value of light (Wh) for the past 30 minutes	Window feature
18	minLight60	The minimum value of light (Wh) for the past 1 hour	Window feature
19	maxLight30	The maximum value of light (Wh) for the past 30 minutes	Window feature
20	maxLight60	The maximum value of light (Wh) for the past 1 hour	Window feature
21	meanT1_30	The mean value of T1 (°C) for the past 30 minutes	Window feature
22	meanT2_30	The mean value of T2 (°C) for the past 30 minutes	Window feature
23	meanT3_30	The mean value of T3 (°C) for the past 30 minutes	Window feature
24	meanT4_30	The mean value of T4 (°C) for the past 30 minutes	Window feature
25	meanT5_30	The mean value of T5 (°C) for the past 30 minutes	Window feature
26	meanT6_30	The mean value of T6 (°C) for the past 30 minutes	Window feature
27	meanT7_30	The mean value of T7 (°C) for the past 30 minutes	Window feature
28	meanT8_30	The mean value of T8 (°C) for the past 30 minutes	Window feature
29	meanT9_30	The mean value of T9 (°C) for the past 30 minutes	Window feature
30	meanRH1_30	The mean value of RH_1 (%) for the past 30 minutes	Window feature
31	meanRH2_30	The mean value of RH_2 (%) for the past 30 minutes	Window feature
32	meanRH3_30	The mean value of RH_3 (%) for the past 30 minutes	Window feature
33	meanRH4_30	The mean value of RH_4 (%) for the past 30 minutes	Window feature
34	meanRH5_30	The mean value of RH_5 (%) for the past 30 minutes	Window feature
35	meanRH6_30	The mean value of RH_6 (%) for the past 30 minutes	Window feature
36	meanRH7_30	The mean value of RH_7 (%) for the past 30 minutes	Window feature
37	meanRH8_30	The mean value of RH_8 (%) for the past 30 minutes	Window feature
38	meanRH9_30	The mean value of RH_9 (%) for the past 30 minutes	Window feature

2.5. LTSM

The input features obtained from the dimension reduction process will be trained using the LSTM model. The structure of the LSTM is shown in Figure 3. The network input and output on the LSTM structure is described as follows [7]:

$$F_t = \sigma(W_f \cdot [H_{t-1}, X_t] + b_f) \quad (3)$$

$$I_t = \sigma(W_i \cdot [H_{t-1}, X_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(W_c \cdot [H_{t-1}, X_t] + b_c) \quad (5)$$

$$C_t = F_t * C_{t-1} + I_t * \tilde{C}_t \quad (6)$$

$$O_t = \sigma(W_o \cdot [H_{t-1}, X_t] + b_o) \quad (7)$$

$$H_t = O_t * \tanh(C_t) \tag{8}$$

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{9}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{10}$$

with W_f , W_i , W_c and W_o are input weights, b_f , b_i , b_c , and b_o are biases, t is the current time, $t-1$ represents a previous state, X is the input, H is the output, and C is the status of cell. The notation σ is a sigmoid function, which produces an input between 0 and 1. A value of 0 means not allowing any value to pass to the next stage, while a value of 1 means to let the output fully enter the next stage. The hyperbolic tangent function (\tanh) is used to overcome the loss of gradients during the training process, which generally occurs in the RNN structure.

The modelling and testing processes were done using Python programming language. This study uses a Keras framework with Tensorflow as a back-end. Some other Python libraries that were used, namely Scikit-learn, Pandas, Matplotlib, Numpy, and Seaborn. The model was trained with the backpropagation method, using Adam's optimisation algorithm.

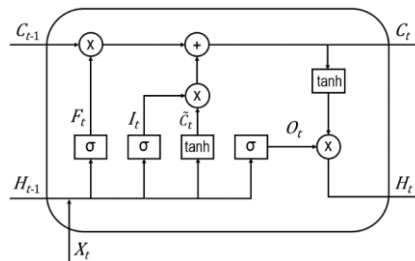


Figure 3. LSTM Structures

2.6. Process workflow

Figure 4 depicts the main workflow of this work. There are 62 attributes gained from both original dataset and feature engineering process. After performing principal component analysis process, these 62 attributes were then reduced to 25 features (principal components). The number of principal components was evaluated based on the experimental process. Based on the PCA outcomes, the LSTM model will predict the value of appliances one-step-ahead (1 hour in the future). The main activity for the model is determining the best model architecture for the LSTM. Both layer and number of neurons will be evaluated based on the model performance.

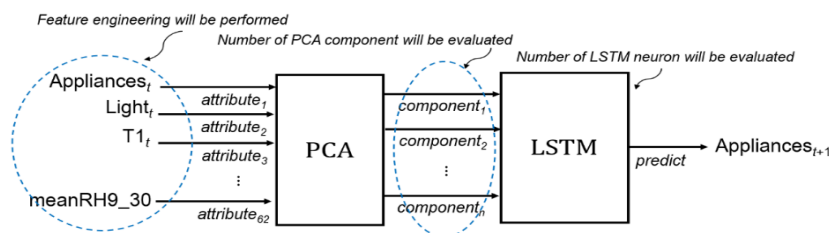


Figure 4. Process workflow

2.7. Predictors performance evaluation

In this work, we implemented root mean squared error (RMSE) and mean average error (MAE), as evaluation parameters. RMSE and MAE can each be calculated using (11) and (12).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n}} \tag{11}$$

$$MAE = \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i|}{n} \quad (12)$$

Where n is the total number of the data sample, Y_i is the measured value, and \hat{Y}_i is the predicted value.

3. RESULTS AND ANALYSIS

3.1. The number of principal components

The number of principal components (PCs) were selected based on the input variance. Typically, the explained variance to be between 95-99%. However, in this work, we selected the range between 85-99%, allows the model predictor trained a wider variety of input numbers. Based on this range, we determined the minimum and maximum required components. The covariance matrix of the normalised features was also calculated. The normalisation process will scale the features between 0 and 1. The general formula for a min-max scaler of [0,1] is given by (13).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (13)$$

Where x is the original value, and x' is the normalised value.

Based on the cumulative variance calculation, the number of components that produce cumulative variance between 85-99% fall between 8 to 26, as shown in Figure 5. PCA components in this range will be trained using LSTM, and the model performances (RMSE and MAE) for each component is summarised in Table 4. In this initial experiment, we determined the LSTM model by only one hidden layer, 15 neurons inside the hidden layer, and 3 lookback lengths (time steps).

As shown in Figure 6, the smallest error value is obtained by 25 principal components, with values of 62.165 and 28.096 for RMSE and MAE, respectively. Thus, the number of these components will be retained for the next process. These number of principal components indicate the number of features as the LSTM inputs. Therefore, 25 features will be fed to our LSTM model. After determining the number of LSTM inputs, we will then move to the next step, that is finding the number of neurons from LSTM. Tuning the number of the hidden layer as well as the number of neurons may significantly improve the model performance.

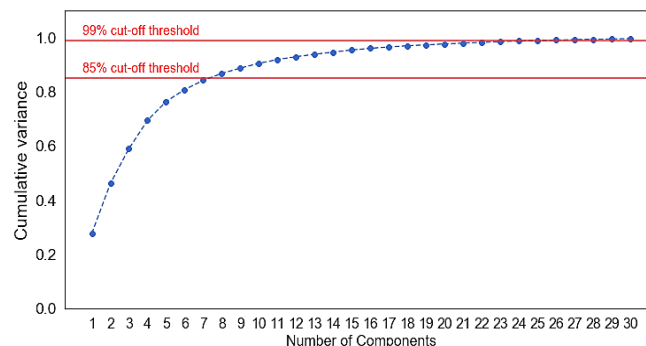


Figure 5. Selecting the number of PCA components

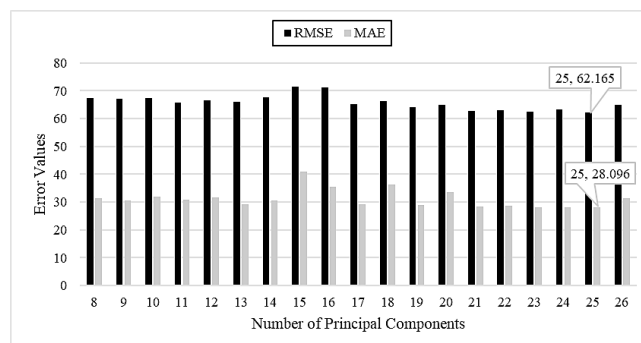


Figure 6. Selecting the number of principal components

3.2. Number of neurons selection

One of the most regulated hyperparameters in training the LSTM model is the number of neurons inside the hidden layer(s). In this work, we determined the number of neurons, either using one or two hidden layers. First, we selected the number of neurons only within one hidden layer, starting from 3 neurons to 150 neurons. The results of RMSE and MAE obtained by each number of neurons were recorded, and the best neuron producing the smallest error will be used to add the neurons in the second layer. The results in selecting the number of neurons and layers for LSTM is summarised in Table 4.

For the first step, we found that the one-layered LSTM with 25 neurons produced the best performance (lowest errors). Then, we add another layer using the previously obtained neuron. We found that the 25 and 20 neurons for the first and second layers produced the smallest errors, with values of 62,103 and 26,982 for RMSE and MAE, respectively. Thus, this 25-20 model architecture will be used in the later stage.

3.3. Number of lookback selection

In time-series modelling, the appropriate selection of the amount of current (or past) data to predict future data can improve the performance of the model. The amount of data that has passed is known as lookback. Lookback in this study is arranged from 1 to 10. This scenario states that the author makes a combination from 1 to 10 of the previous data (including the current data) to predict one data in the future. Because each data has a 10-minute interval, lookback of 1 indicates that the current 10-minute value is used to predict the value of the next 10 minutes. Lookback of 10 means that the author uses 10 data to predict one future data. Figure 7 illustrates this process. Table 5 shows the model performances based on the lookback variations. Lookback of 3 produced the smallest errors, with values of 62.013 and 26.982 for RMSE and MAE, respectively.

Table 4. Model performances obtained from the different number of neurons

Number of Neurons	RMSE	MAE	Number of Neurons	RMSE	MAE
3	67.251	29.817	25-10	62.339	29.760
6	63.714	28.299	25-12	63.639	30.222
9	68.830	41.016	25-14	63.791	27.762
15	62.165	28.096	25-16	62.482	27.149
25	62.099	28.094	25-18	63.221	29.410
50	65.489	29.701	25-20	62.013	26.982
100	67.021	30.317	25-22	67.131	44.545
150	65.415	29.760	25-25	64.520	30.851

Table 5. Results of selection of neurons

Lookback	RMSE	MAE
1	66.589	28.808
2	64.221	27.272
3	62.013	26.982
4	63.859	27.997
5	63.985	30.258
6	62.709	28.979
7	62.561	29.092
8	63.837	37.676
9	64.031	29.491
10	64.077	29.673

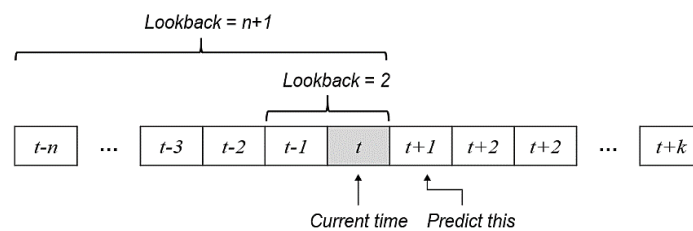


Figure 7. Selecting the number of PCA Components

As an illustration in the selection of this lookback values, the autocorrelation function of a time-series data can be applied. If current conditions y_t are simplified as A , and future conditions y_{t+k} as B , where k is the time delay, then the autocorrelation function is calculated using the following equation:

$$\text{cor}(A, B) = \frac{\text{cov}(A, B)}{\text{std}(A) \text{std}(B)} \tag{14}$$

where $\text{cov}(A, B)$ is the covariance between A and B , while $\text{std}(A)$ and $\text{std}(B)$ are the standard deviations from A and from B , respectively. Figure 8 shows the autocorrelation coefficient of the Appliances vs time lag. In the figure, the delay time of more than 10 does not have a significant correlation. In this study, the lookback value of 3 produces the most optimal output.

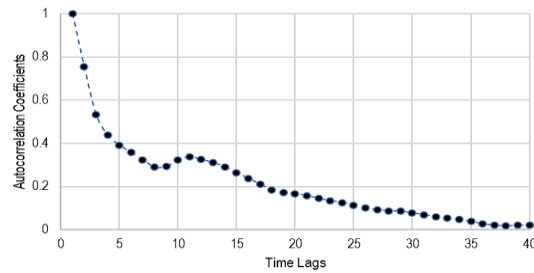


Figure 8. The value of the autocorrelation function with time delay

3.4. Overview of actual values with predicted values

As discussed in section 2.1, the dataset has 19,735 rows of data. The first 60% of data is used as training data (11,841 rows), the next 20% is validation data (3,947 rows), and the last 20% is test data (3,947 rows). If referring to the time of data collection in the dataset, the training data starts from January 11, 2016 at 17:00 until April 02, 2016 at 22:20. Validation data starts on April 2, 2016 at 22:30 until April 30, 2016 at 8:10. Finally, the test data begins on April 30, 2016 at 08:20 to May 27, 2016 at 18:00. It should be noted that we did a feature engineering process in this research, one of which used the window rolling method. For example, this study uses the meanApp60 attribute (see Table 3), which means that the average 60 minutes of data that has passed (including current data) is used as input to predict one data ahead. As a result, five earliest pieces of data are missing to produce one current input data.

The graph between the prediction results and the actual values in the test data and the first 500 test data plots are shown as in Figure 9. The continuous line shows the actual values, while the dotted line shows the predicted results. Based on these figures, it appears that in general, the prediction results have followed the actual patterns. Fluctuations for low Wh values can be well followed. However, the model has not perfectly captured the high surge of Wh values.

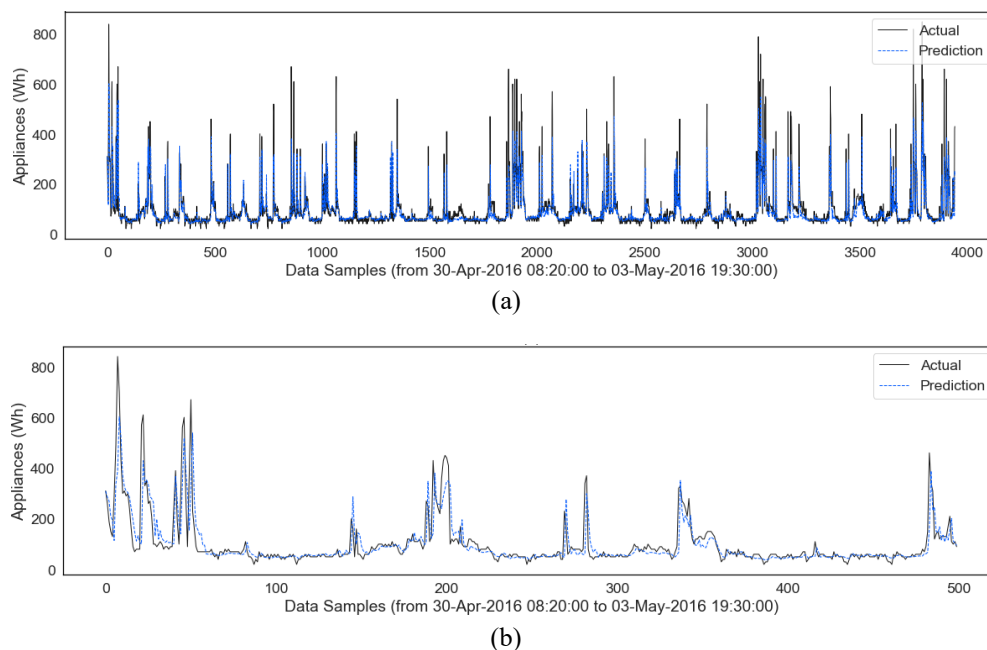


Figure 9. (a) Comparison of actual and predicted values for the full period, and (b) comparison for the first 500 test data

4. CONCLUSION

Electrical appliances become one of the most significant sources of electricity use in a residential home. This study applied feature engineering and long short-term memory (LSTM) to predict the amount of electricity used in a residential home. The feature engineering technique was conducted by synthesising new features from existing dataset to improve the performance of the predictor model. Feature engineering used in

this study can be categorized into three, namely data frame features, lag features and window features. Principal component analysis reduced the input dimensions yielded by feature engineering process from 62 features to 25 features. The LSTM model with the architecture of 25-20 (25 neurons in the first layer and 20 neurons in the second layer) with lookback of 3 produced the best performance, with the error magnitude of 62.013 and 26.982 for RMSE and MAE, respectively.

REFERENCES

- [1] PLN, "Annual Report PT Perusahaan Listrik Negara (Persero) 2018." PT.PLN (Persero), 2018, [Online]. Available: <https://web.pln.co.id/stakeholder/laporan-tahunan>.
- [2] K. S. Cetin, P. C. Tabares-Velasco, and A. Novoselac, "Appliance daily energy use in new residential buildings: Use profiles and variation in time-of-use," *Energy and Buildings*, vol. 84, pp. 716-726, 2014.
- [3] N. Arghira, L. Hawarah, S. Ploix, and M. Jacomino, "Prediction of appliances energy use in smart homes," *Energy*, vol. 48, no. 1, pp. 128-134, 2012.
- [4] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*, vol. 140, pp. 81-97, 2017.
- [5] I. N. K. Wardana, N. Javas, and I. K. A. A. Aryanto, "Prediction of Electrical Energy Usage in Residential Homes Using Long Short-Term Memory (in Indonesia: Prediksi Penggunaan Energi Listrik pada Rumah Hunian Menggunakan Long Short-Term Memory)," *TIERS Information Technology Journal*, vol. 1, no. 1, Art. no. 1, Jun. 2020, Accessed: Jul. 01, 2020. [Online]. Available: <http://journal.undiknas.ac.id/index.php/tiers/article/view/2475>.
- [6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
- [7] T. Li, M. Hua, and X. Wu, "A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM_{2.5})," *IEEE Access*, vol. 8, pp. 26933-26940, 2020.
- [8] Á. Carmona-Poyato, N. L. Fernández-García, F. J. Madrid-Cuevas, and A. M. Durán-Rosal, "A new approach for optimal time-series segmentation," *Pattern Recognition Letters*, vol. 135, pp. 153-159, 2020.
- [9] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Computers & Industrial Engineering*, vol. 143, 2020.
- [10] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297-305, 2018.
- [11] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, 2017.
- [12] X. Bai, "Text classification based on LSTM and attention," in *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, Sep. 2018, pp. 29-32.
- [13] A. H. Mirza and S. Cosan, "Computer network intrusion detection using sequential LSTM Neural Networks autoencoders," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 2018, pp. 1-4.
- [14] C.-J. Huang and P.-H. Kuo, "A Deep CNN-LSTM Model for Particulate Matter (PM_{2.5}) Forecasting in Smart Cities," *Sensors*, vol. 18, no. 7, 2018.
- [15] Q. Zhang, J. C. Lam, V. O. Li, and Y. Han, "Deep-AIR: A Hybrid CNN-LSTM Framework for Fine-Grained Air Pollution Forecast," *arXiv:2001.11957 [eess]*, Jan. 2020, Accessed: May 28, 2020. [Online]. Available: <http://arxiv.org/abs/2001.11957>.
- [16] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 37-52, Aug. 1987.
- [17] X. Peng, L. Zhang, F. Tian, and D. Zhang, "A novel sensor feature extraction based on kernel entropy component analysis for discrimination of indoor air contaminants," *Sensors and Actuators A: Physical*, vol. 234, pp. 143-149, 2015.
- [18] F. Bencheikh, M. F. Harkat, A. Kouadri, and A. Bensmail, "New reduced kernel PCA for fault detection and diagnosis in cement rotary kiln," *Chemometrics and Intelligent Laboratory Systems*, vol. 204, 2020.
- [19] J. Ma and Y. Yuan, "Dimension reduction of image deep feature using PCA," *Journal of Visual Communication and Image Representation*, vol. 63, 2019.
- [20] Q. Tao, F. Liu, Y. Li, and D. Sidorov, "Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU," *IEEE Access*, vol. 7, pp. 76690-76698, 2019.
- [21] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," in *SoutheastCon 2016*, Mar. 2016, pp. 1-6.
- [22] R. Punmiya and S. Choe, "Energy Theft Detection Using Gradient Boosting Theft Detector with Feature Engineering-Based Preprocessing," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2326-2329, Mar. 2019.
- [23] L. Bai, J. Wang, X. Ma, and H. Lu, "Air Pollution Forecasts: An Overview," *Int J Environ Res Public Health*, vol. 15, no. 4, 2018.
- [24] J. Lever, M. Krzywinski, and N. Altman, "Principal component analysis," *Nature Methods*, vol. 14, no. 7, 2017.
- [25] F. Taşpınar, "Improving artificial neural network model predictions of daily average PM₁₀ concentrations by applying principle component analysis and implementing seasonal models," *Journal of the Air & Waste Management Association*, vol. 65, no. 7, pp. 800-809, 2015.