Indonesian Journal of Electrical Engineering and Computer Science
Vol. 37, No. 2, February 2025, pp. 758~770
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v37.i2.pp758-770

Natural smart home automation system using LSTM based on household behaviour

Mochamad Susantok[1,2], Farhana Ahmad Po'ad[1], Ariffuddin Joret[1], Maulina Hilwa Salsabillah[2]
[1]Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, Malaysia
[2]Program Studi Teknik Elektronika Telekomunikasi, Politeknik Caltex Riau, Pekanbaru, Indonesia

Article Info

Article history:

Received Jul 16, 2024
Revised Sep 18, 2024
Accepted Sep 30, 2024

Keywords:

Electrical overload prevention
Household energy management
IoT smartplugs
LSTM
Real-time data collection
Smart home automation system

ABSTRACT

A smart home automation system (SHAS) utilizing data-driven learning is an advanced internet of things (IoT) application aimed to learn household behavior to prevent miniatur circuit breaker (MCB) trips due to overload. Unlike traditional deterministic methods, this study leverages a layered AI model, featuring real-time data collection, long short-term memory (LSTM) based learning, and an automatic control system. The LSTM classification model generates precise ON/OFF control signals sent to IoT smartplugs, optimizing appliance usage and reducing the risk of electrical overload. Data from smartplug sensors, including appliance status and environmental factors like power consumption, temperature, and humidity, were collected every minute over three months, yielding 80,818 data points. The system's performance was evaluated on three appliances: Air Conditioner, Television, and Water Pump Machine. Results showed high accuracy for Television at 98% and Water Pump Machine at 97.6%, with slightly lower accuracy for Air Conditioner at 81.9%. This demonstrates the system's effectiveness in real-world applications. The scalability and adaptability of the Natural SHAS model to different appliances and environments mark a significant advancement in smart home automation, offering a practical solution for preventing electrical overload and improving household energy management.

Corresponding Author:

Farhana Ahmad Po'ad
Faculty Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia
Parit Raja, Johor, 86400, Malaysia
Email: farhana@uthm.edu.my

1. INTRODUCTION

Households in Indonesia are categorized into several customer groups by Perusahaan Listrik Negara (PLN), the state owned electricity company, in terms of electricity consumption [1]. For example, the R-1/TR category limit electricity consumption to a maximum of 1,300 VA with a tariff of Rp. 1,444.70 per kWh. This means that the total electricity consumption from all appliances should not exceed 1,300 watts, assuming a power factor (cos phi) of 1. However, in reality, many households have various electrical appliances with a total power consumption exceeding this subscription limit. Consequently, when multiple appliances are used simultaneously, exceeding the total power limit, it cause the miniature circuit breaker (MCB) to trip. When the MCB trips, the electrical connection from the main PLN line to the house is cut off, causing all electrical appliances in the house to shut down. This condition leads to the occurrence of inrush current in electrical appliances when the MCB is switched back on. The inrush current phenomenon is disadvantageous as it increases electricity consumption at the beginning of operation due to the large momentary current and can also damage electrical equipment because of the large starting torque required [2].

Additionally, frequent overloading and subsequent tripping of the MCB pose safety risks to the household [3], as electrical equipment may be damaged [4], potentially leading to dangerous situations [5]. Furthermore, this frequent ON-OFF cycling of electricity can disrupt energy management in the home, causing inefficiencies and inconvenience [6].

To address this issue, automatic control systems based on load prioritization [3] and the elimination of high-consumption appliances [7] have been implemented. However, these approaches are often static and lack adaptability to natural environmental factors. In contrast, homeowners who are aware of their total power consumption typically develop habits of turning appliances ON and OFF as needed to avoid exceeding the maximum load limit. This behavior reflects a more dynamic and responsive approach to energy management within the home. This study focuses on the behavior of household residents as the object of research to provide solutions for preventing overloads. The method uses an electrical approach by leveraging internet of things (IoT) technology to capture data on residents' behavior and AI deep learning technology to automatically control the ON-OFF state of electrical appliances based on the residents' behavior.

The integration of AI and IoT known as AIoT has resulted in impressive technology where deep learning can derive new insights from continuously updated data from various sources. This advancement has brought progress to various fields, such as smart homes [8], smart cities [9], industry, healthcare [10], [11], and transportation [12]. In the context of smart homes, the application of this technology extends to enhancing the functionality and comfort of household residents. This includes the evolution from remote access and control to automatic system control based on learning data [13], thereby improving residents' comfort in using electrical appliances while achieving energy savings [14].

Referring to IoT technology, electrical appliances with smart sockets can transmit data on the status of an appliance, including power consumption, current, voltage, energy, power factor [7], [15], and ON/OFF status [16] to other devices and the cloud within an IoT network. Similarly, other smart home devices such as thermostats, PIR sensors for detecting human presence in a room, and water tank level sensors to trigger the water pump can serve as data sources for smart systems like the smart home automation system (SHAS) to understand household behavior patterns in operating electrical appliances. Furthermore, as to embedding AI technology in the IoT system, Deep learning models such as long short-term memory (LSTM) can be implemented as it suitable for learning datasets with time series data, like household behavior patterns that follow daily cycles [17]-[18]. Once patterns are identified through deep learning, or what is referred to as the model, SHAS will use this model to predict outputs in the form of automatic controls, such as turning an electrical appliance ON or OFF. Thus, automatic control happens naturally rather than deterministically [19], through conditioning or scheduling, such as the scheduled ON/OFF control of lights [20]. This adds value to SHAS with the application of AI in an IoT environment.

Based on an AIoT system, this study contributes to the design of a layered SHAS model using the LSTM algorithm with datasets from IoT devices in a house, including the ON-OFF status of electrical appliances and other supporting sensor data. The following sections of this paper will cover the methodology of the built system and its workflow, followed by a discussion of the results, which will present the dataset characteristics and the performance evaluation of LSTM in predicting the ON or OFF status of electrical appliances using the confusion matrix parameter.

2. METHOD

This study proposes a SHAS layered model, as depicted in Figure 1, which integrates AI within an IoT environment. The SHAS developed in this research utilizes LSTM as the AI feature to control electrical appliances based on household behavior data. In the physical layer, sensors generate data such as the ON/OFF status of smartplugs or smart sockets for electrical appliances, total electricity consumption from smart power meters, air quality data including temperature and humidity, water tank level data indicating the percentage of water height in the tank, and motion data detecting human presence in a room. These sensors wirelessly transmit their data within an IoT network to an edge device, functioning as a sensor hub that aggregates all sensor data into a streaming dataset named the automation home electrical appliance control system (AHEACS). The dataset duration is per minute, collected over three months from February to April 2024, totaling 80,818 data points.

In the management layer, LSTM processes the dataset to learn, evaluate the model's performance, and produce the most optimal predictive model to be used in the proposed SHAS model. The model's prediction results, in the form of ON/OFF commands, are sent to the sensor devices, specifically the smartplugs containing relays connected to the electrical appliance sockets. This communication can be carried out through a local IoT network such as wireless local area network (WLAN) and by using machine-to-machine communication protocols like message queuing telemetry transport (MQTT). In the case of communication between IoT devices, such as AI devices and smartplugs, they act as MQTT clients with an MQTT broker device that is also connected locally within the same WLAN. This AI process can be executed at the edge, ensuring that the automatic control process does not rely on internet connectivity [21]. Meanwhile, the application layer resides in the cloud, providing users with the option to monitor sensor data and manually remote-control electrical appliances via the internet.

This work presents the performance of LSTM as the AI component in the IoT SHAS application on the edge side of an IoT network, as shown in Figure 1. The methodology consists of three parts: the study of proposed dataset characteristics, the design of the LSTM model for each smartplug connected to an electrical appliance and also performance analysis of the LSTM algorithm in Google Colab environment using python.

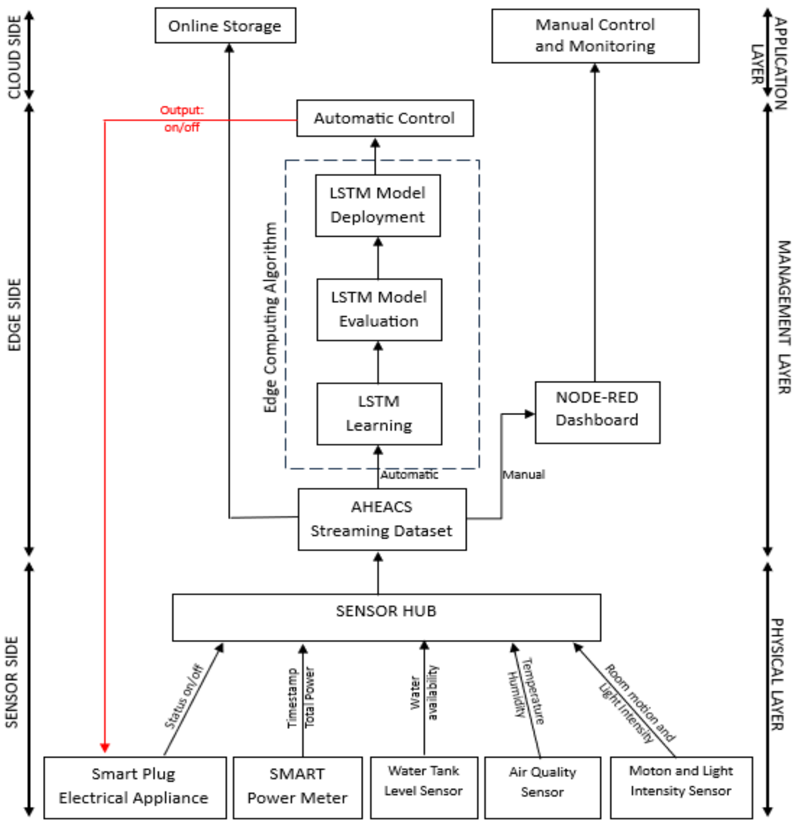

Figure 1. SHAS layered model

2.1. SHAS dataset streaming

The role of the sensor hub in the SHAS layered model design of this research showed in Figure 1 is to form a streaming AHEACS dataset from various sensors in a smart home application. This study uses a smart power meter to generate total power (TP) data, indicating total electricity consumption. Five smartplugs provide ON/OFF status data for each electrical appliance, such as an air conditioner (S-AC), Refrigerator (S-K), Water Pump (S-WPM), Washing Machine (S-MC), and Television (S-TV). Additionally, environmental sensors that related to the home's condition are used as factors influencing the residents' decision to turn electrical appliances ON or OFF. These include sensors for temperature (T), humidity (H), motion in the bedroom (Pb), motion in the living room (Pg) where the television is located, and water tank level (W), indicating the water level in the water tank of the chosen house.

Table 1 shows the design of the streaming dataset columns, where the timestamp (TS) represents the data collection time, set per minute. S-X represents smartplug status which contain S-AC, S-K, S-WPM, S-MC, and S-TV, with the possibility of adding other electrical appliances in the future. In this study, TS is detailed to reflect the influence of weekdays (Monday-Friday) and weekends (Saturday-Sunday), as well as operational hours (morning-afternoon-evening). Thus, TS is extracted into year, month, day, hour, minute, and second values, which are converted into numeric form. Year and month remain numeric, while the day is represented as (D), where Monday=1, Tuesday=2, and so on until Sunday=7. Hours, minutes, and seconds are converted into the number of seconds since midnight (NSM), a value that repeats daily. To distinguish between weekdays and weekends, the variable (WS) is used: 0 for weekdays and 1 for weekends.

Table 1. Dataset labeling

Variable	Description	Type of data
TS	Timestamp	Date and Time
NSM	Number of Seconds since Midnight, convert all time in a day to become number of second	Numerical
WS	Weekday status	Boolean (weekday=0, weekend=1)
D	The Day from Monday to Sunday	Numerical (Monday=1 until Sunday=7)
HR	Date	Date
WK	Time	Time
TP	Total Power (Watt)	Numerical
S-X	The state of electrical appliance operation	Boolean (on=1, off=0)
T	Temperature (°C)	Numerical
H	Humidity (%)	Numerical
Pb	Present people in bedroom	Boolean (present=1, empty=0)
Pg	Present people in television room	Boolean (present=1, empty=0)
W	Water tank level	Numerical

2.2. LSTM based for SHAS

The role of IoT technology is highly beneficial in collecting real-time usage data of electrical appliances in homes, which can be compiled into a dataset. This data is then processed and analyzed to understand user usage habits, leading to actions based on new insights obtained, as demonstrated in the SHAS application. This concept is known as the internet of behavior (IoB) [22].

In the context of SHAS, which focuses on the time-dependent usage behavior of household electrical appliances, a recurrent neural network (RNN), an extension of a Neural Network model specifically for learning sequential data over time, can be applied [23]. The expectation from an RNN is to capture long-term dependencies so that all past inputs can influence the output. However, RNNs face challenges with inputs that are too distant in the past. Fortunately, these problems can be addressed by applying an evolution of the RNN known as the LSTM model [24]. The LSTM model includes specialized units called memory cells, which can store information for an extended period, acting like a conveyor connecting LSTM blocks. These units involve three types of gates: input, forget, and output, which control the flow of information. These gates are crucial as they determine whether to allow new input, erase the current cell status, or let the status influence the output at a specific time step.

This paper does not explain the general workings of LSTM in detail, as LSTM itself is not new in Deep Learning algorithms. For more details, refer to previous literature on LSTM related to smart home applications [18], [25]. Here, the focus is on the detailed LSTM model for the streaming dataset case as previously explained. The LSTM architecture in this study, as shown in Figure 2, consists of one input layer, two hidden layers, and one output layer with fully connected layer.

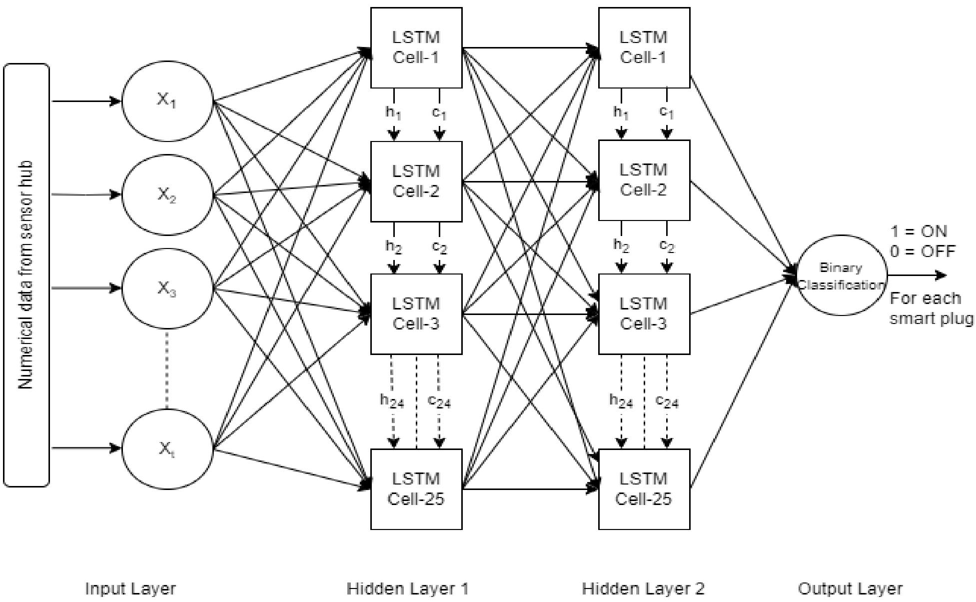

Figure 2. LSTM architecture for SHAS

In the input layer of the LSTM model, the number of input features is determined by various factors that influence the ON/OFF operation of each electrical appliance. For example, in the case of controlling an Air Conditioner (AC), relevant input features might include Total Power (TP), Temperature (T), Humidity (H), Bedroom Motion (Pb), and Water Pump Status (S-WPM). Thus, for the smartplug connected to the AC, the LSTM model receives 5 input features. However, within the LSTM neural network, the input layer is structured to take in one timestep or input dimension at a time. This is because all input features are measured simultaneously at a specific time or timestep. The input data, represented as $x_1, x_2, x_3, \ldots\ldots, x_t$ in the diagram, corresponds to these features for each electrical appliance.

In this study, the input features for different appliances vary based on their operational context. The notation S-X in the Table 2 o the specific smartplugs connected to different appliances, such as Air Conditioner (S-AC), Water Pump (S-WPM), Refrigerator (S-K), Washing Machine (S-MC), and S-TV (Television). The diagram illustrates how these inputs feed into the LSTM cells sequentially, with each cell learning from the input data and passing the learned information forward through the network, ultimately leading to the binary classification output that controls the appliance's ON/OFF state. This process is captured in the LSTM architecture shown in Figure 2, where the input layer feeds the data into a series of LSTM cells, each maintaining and updating the cell state $C_{(t)}$ and hidden state $h_{(t)}$. The final hidden state after processing all timesteps is then used for binary classification, which determines the operational control status of the appliance.

The determination of input features is based on the threshold value of the correlation coefficient with the (1):

$$r_{xy} = \frac{\sum(x-\bar{x})(y-\bar{y})}{N.S_x S_y}) \quad (1)$$

The Pearson correlation coefficient r_{xy} ranges from -1 to 1. If r_{xy} is positive, the input feature variables tend to move in the same direction, indicating a positive correlation. Conversely, if r_{xy} is negative, the correlation is negative, meaning the variables move in opposite directions. If r_{xy} approaches 0, there is no linier relationship between the two input feature variables. Meanwhile N represent the total number of data samples and $S_x S_y$ represent the standard deviation from two variable.

The correlation level between input features is detailed in the exploratory data analysis (EDA) section of this paper. From the input layer, a fully connected network is created to the first hidden layer, which consists of 25 LSTM cells. In this layer, the general LSTM process occurs, where memory cells with three gates are designed to read, store, and update past information. Each LSTM cell connects to the next LSTM cell through the cell state $C_{(t)}$ and hidden state $h_{(t)}$ with the (2) and (3):

$$C_{(t)} = f_{(t)}.C_{(t-1)} + i_{(t)}.\tilde{C}_{(t)} \quad (2)$$

$$h_{(t)} = o_{(t)}.\tanh\left(C_{(t)}\right) \quad (3)$$

Where is:
$f_{(t)}$= forget gate
$i_{(t)}$= input gate
$o_{(t)}$= output gate
$\tilde{C}_{(t)}$= candidate gate

The selection of 25 LSTM cells, as shown in Figure 2, is based on the complexity of the input feature variables and the dimensions of the data [26]. As previously mentioned, this study employs a single input dimension for time-series data, aiming to keep the model simple and suitable for implementation on the edge side of an IoT network. The LSTM network then connects to the second hidden layer, which mirrors the first layer in terms of parameters, such as the use of a dropout layer and the return sequence setting. The dropout layer is applied to prevent overfitting during training, with a dropout rate of 0.1 or (10%) of the total 25 neurons per hidden layer. The return sequence parameter in the first hidden layer is set to "true" to ensure that the output sequence at each timestep aligns with the input conditions. For the second hidden layer, this parameter is set to "false" so that the output dimension becomes singular, matching the desired final output. In the output layer, the LSTM model performs binary classification, predicting whether each smartplug connected to an electrical appliance should be turned ON (1) or OFF (0).

2.3. SHAS environment

This study uses Google Colab to run the LSTM algorithm with Python programming to measure its performance in SHAS. The TensorFlow module serves as the framework for training the deep learning model, with Keras as the integrated neural network API. Additionally, the Pandas library is used for the EDA process to determine the number of features in the input layer for each smartplug electrical appliance of the developed SHAS in this study. Data is imported from the sensor hub in .csv format and collected weekly, with dimensions of (80,818; 12), representing 80,818 rows of data and 12 columns or input features. The TS column is then extracted into NSM, WS, D, HR, and WK, as shown in Table 1, changing the dimensions to (80,818; 16). In this system, which uses an AI technology, the LSTM training data is taken weekly, assuming the household residents' habits repeat each week on normal days and not during long holidays. The dataset is then normalized to speed up computation and avoid large value ranges between input features [13] using the scikit-learn library in Google Colab. This normalization process involves removing data with NaN or null values originating from the sensor data sources. The occurrence of such values depends on the quality of the sensor readings and the data network during transmission to the sensor hub. After this process, the data is considered clean, with an equal number of entries for each column or feature variable. Min-Max scaling is performed during the preprocessing phase in EDA, with a range of -1 to 1 for all input feature values. The purpose of the Min-Max function is to ensure data consistency and accelerate the performance of algorithms during the learning process. This normalization technique transforms data from all features to a range of -1 to 1 [27]. The equation used in this research is as follows, where X_{max} is the highest value and X_{min} is the lowest value in the sample data.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \times (1 - (-1)) + (-1) \tag{4}$$

After normalizing all data within the range of -1 to 1, the next step is to separate the input and output data for each smartplug electrical appliance based on the correlation coefficient r_{xy} (1). In this study, the LSTM training process is set with the following parameters: 25 cell or neurons in each hidden layer (see Figure 2) with 10% of overfitting, 75% of the data is used for training, 25% for testing, 25 epochs, a learning rate of 0.0001, and the optimization algorithm used is Adam.

3. RESULTS AND DISCUSSION

This research investigates the impact of household behavior on the SHAS to prevent electrical overload. While previous studies also aimed to prevent overload, their approaches relied on deterministic factors without considering environmental conditions that influence household behavior. This section presents a detailed analysis of the research findings, starting with exploratory data analysis (EDA), which preprocesses raw data into a usable dataset. It then identifies key input variables for the LSTM model for each smart plug. The performance of SHAS is evaluated by comparing results from different devices, such as S-AC, S-TV, and S-WPM, using real-world data.

3.1. SHAS data preprocessing

In this study, data preprocessing begins with extracting the AHEACS dataset in CSV format from the dataset repository. The data is then transformed and normalized to ensure it is clean, with no null or NaN values, and consistent in size, meaning each feature contains the same number of data points. Afterward, correlation analysis is performed to identify the input features for the desired smartplug output using a heatmap visualization based on (1). Figure 3 shows the heatmap results of all dataset features or variables. In this heatmap, a value close to 1 (dark-red) indicates a strong positive correlation, while a value close to -1 (dark blue) indicates a strong negative correlation.

For the smartplug connected to the AC (S-AC), 15 candidate input features were identified, with an absolute correlation value r_{xy} of |0.03|. This value was chosen to account for the time variable, influencing the S-AC output. It is not set smaller to avoid involving too many variables and overly complex model. Based on the heatmap in Figure 3, the input features are [TP, S-TV, S-WPM, S-MC, T, H, W, Pb, Pg, WS] where WS is the time variable. In addition to S-AC, this study also evaluates the S-TV and S-WPM smartplugs. For these, an r_{xy} of |0.1| was choosen. The input features for S-TV are [TP, S-AC, T, H, W, Pg, NSM, WK], and for S-WPM, the input features are [TP, S-AC, S-MC, T, W].

We found that time variables (WK, HR, D, WS, and NSM) have an influence on control variables such as S-AC, S-TV, and S-WPM. Although the correlations are small, ranging from 0.01 to 0.17, time variables extracted from the time series data are particularly significant due to their impact on household appliance usage, which is influenced by time-based patterns. The residents' specific routines for using appliances like air conditioners or washing machines are crucial for accurate energy consumption predictions. Although the correlations may appear low, capturing these temporal patterns enhances analysis depth compared to static deterministic control methods [19], [20]. The flexibility in determining the r_{xy} value allows for adjustments according to the desired model behavior and design.

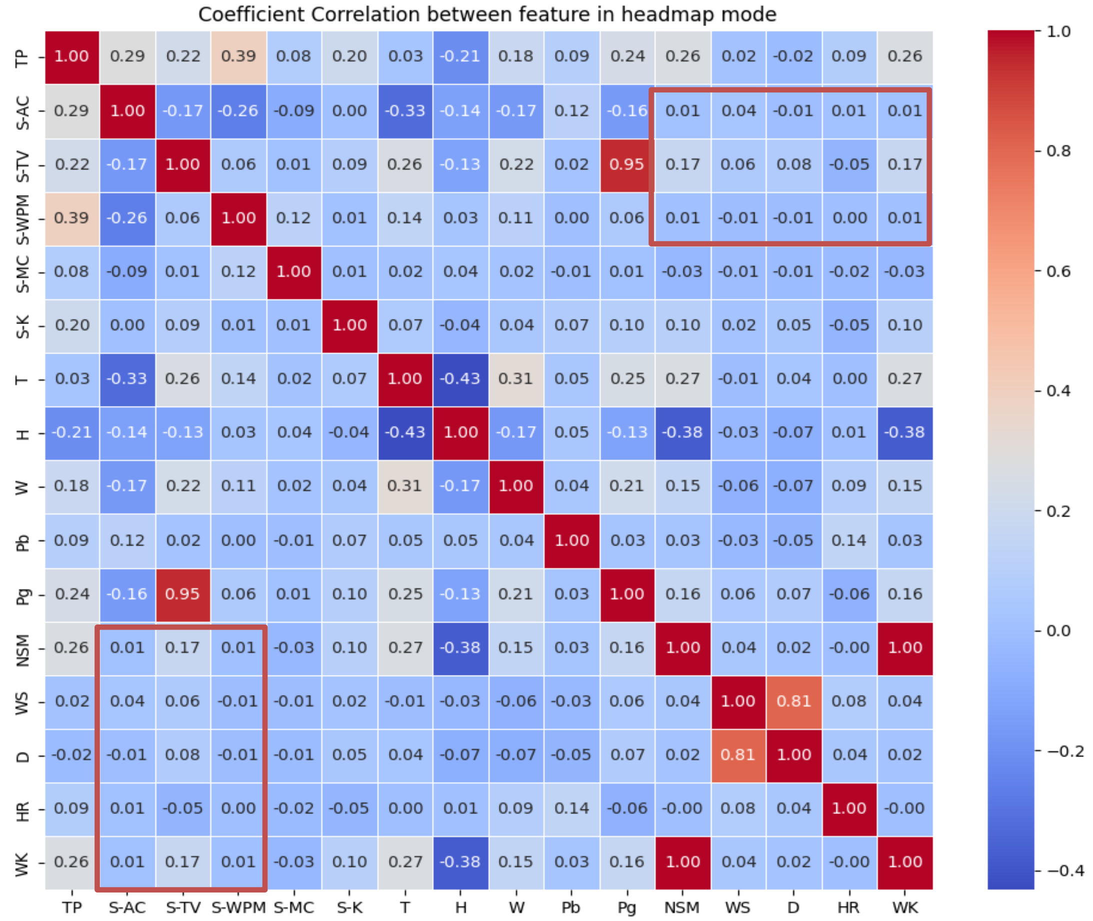

Figure 3. Coefficient correlation between feature in AHEACS dataset

After determining the variables to be used as input features for each smartplug (S-AC, S-TV, and S-WPM), the LSTM model learns from the dataset to build a classification model with outputs of either 'ON' or 'OFF' for each smartplug. Table 2 shows the results of the learning process conducted in the Google Colab environment. The correlation between Figure 2, which depicts the LSTM design, and Table 2, which presents the results of the implementation using Python, shows that this model has three layers: LSTM, Dropout, and Dense (Output). For example, the implementation for S-AC is explained as follows:

a) lstm_1: This is the first LSTM layer in the model, with an output shape of (None, 10, 25). This means that the layer has 25 units (or memory cells), and it produces a sequence of 10 output vectors, where each vector has 25 elements. These output vectors are equivalent to the number of input features: S-AC has 10 input features, S-TV has 8 input features, while S-WPM has 5 input features. The LSTM layer receives input from the input layer and processes it by updating its internal state, which consists of a cell state and a hidden state. The LSTM layer can learn to keep or discard information from the previous time steps, depending on its relevance for predicting the output.
b) dropout_1: This is a dropout layer, which is used to prevent overfitting by randomly setting a fraction of the input units to zero during training. The dropout rate is denoted by 10% or 0.1, as mentioned before in the previous section. The output shape of the dropout layer is the same as the input shape, which is (None, 10, 25) in this case.
c) lstm_2: This is the second LSTM layer in the model, with an output shape of (None, 25). This means that the layer has 25 units, and it produces a single output vector for each input sequence. This LSTM layer receives input from the previous layer and processes it by updating its internal state, which consists of a cell state and a hidden state.
d) dropout_2: This is another dropout layer, which is used to prevent overfitting by randomly setting a fraction of the input units to zero during training. The dropout rate is denoted by 10% and the output shape of this dropout layer is the same as the input shape, which is (None, 25) in this case.

e) dense: This is the output layer of the model, which is a fully connected layer with 1 unit (or neuron). The output shape of the dense layer is (None, 1), which means that it produces a single scalar value for each input sequence. The activation function of the dense layer is tanh, which is used for binary classification tasks.

Table 2. Summary LSTM classification model

Layer	S-AC		S-TV		S-WPM	
	Output Shape	Param #	Output Shape	Param #	Output Shape	Param #
lstm_1	(None, 10, 25)	2700	(None, 8, 25)	2700	(None, 5, 25)	2700
dropout_1	(None, 10, 25)	0	(None, 8, 25)	0	(None, 5, 25)	0
lstm_2	(None, 25)	5100	(None, 25)	5100	(None, 25)	5100
dropout_2	(None, 25)	0	(None, 25)	0	(None, 25)	0
dense	(None, 1)	26	(None, 1)	26	(None, 1)	26

Total params: 7826
Trainable params: 7800
Non-trainable params: 26

The total number of parameters in the model is 7,826, which includes 7,800 trainable parameters and 26 non-trainable parameters. The trainable parameters are the weights and biases of the LSTM and dense layers, which are learned during training. The non-trainable parameters are the dropout masks, which are randomly generated and fixed during training. These results demonstrate the reliability of LSTM as one of the best-performing deep learning algorithms, particularly in handling large-scale data such as in this study which uses a model with 5 layers (1 input layer, 3 hidden layers, and 1 output layer) [28]. However, the LSTM design in this study, as previously explained, maintains simplicity to ensure its applicability on edge devices within IoT networks. Further refine LSTM architectures to reduce computational complexity while maintaining or even improving accuracy. Research could focus on pruning techniques, quantization, and efficient memory management to make LSTM models even more suitable for edge devices with limited resources.

3.2. SHAS performance analysis

This study uses LSTM in a SHAS application to produce output in the form of ON or OFF classifications. These ON or OFF messages are then forwarded to the smartplug via the MQTT protocol, where they are translated by the relay in the smartplug into an ON or OFF switch to control the power flow to the electrical appliance.

To evaluate the performance of the LSTM model in this SHAS, the confusion matrix method is used, which is a standard approach in classification modeling. The confusion matrix $M = \begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}$ provides insight into the model's accuracy [29]. Here, True Positive (TP) represents cases where the smartplug is correctly predicted to be ON, False Negative (FN) represents cases where the plug is predicted to be OFF when it is actually ON, True Negative (TN) represents correct predictions that the plug is OFF, and False Positive (FP) represents incorrect predictions where the plug is predicted to be ON when it is OFF. All the predicted data from the LSTM model, ordered by time, will be classified into these four variables. Below are the results of the confusion matrix from the LSTM model with a total of 80,818 data points on the dataset for each S-AC, S-TV, and S-WPM, which were run in the Google Colab environment using Python:

$$M_{S-AC} = \begin{pmatrix} 40594 & 4339 \\ 10418 & 25467 \end{pmatrix} M_{S-TV} = \begin{pmatrix} 22273 & 450 \\ 1155 & 56940 \end{pmatrix} M_{S-WPM} = \begin{pmatrix} 5728 & 1044 \\ 952 & 73094 \end{pmatrix}$$

Next, the confusion matrix parameters are calculated using the (5)-(8) [30]:

$$accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{5}$$

$$precision = \frac{TP}{TP+FP} \tag{6}$$

$$recall = \frac{TP}{TP+FN} \tag{7}$$

$$F1-score = 2 * \frac{precision*recall}{precision+recall} \tag{8}$$

The performance evaluation of the SHAS using (5)-(8) is summarized in Table 3 and visually represented in Figure 4. The evaluation indicates that the LSTM model's learning of the household behavior is quite successful, achieving high accuracy for the smartplug TV (S-TV) at 98% and the smartplug Water Pump Machine (S-WPM) at 97.6%, but slightly lower for the smartplug AC (S-AC) at 81.9%. For S-TV, precision and recall are both very high at 95% and 98%, respectively. This indicates that the LSTM model accurately predicts the ON/OFF status of the TV, with minimal false positives (FP) and false negatives (FN). S-WPM also demonstrates good performance, with a precision of 85.7% and recall of 85%, meaning that most instances are correctly detected, though a few are missed. In contrast, S-AC shows a lower precision of 79.5% compared to a recall of 90.3%, suggesting that while the model correctly predicts most S-AC instances, there are more false positives. The performance gap between S-TV and S-AC could be due to various factors affecting appliance usage patterns. The accurate prediction of smartplug operations, particularly for S-TV and S-WPM, directly contributes to preventing electrical overloads. This prevention is crucial for maintaining household safety by reducing the risk of MCB trips, which can lead to equipment damage or safety hazards. Additionally, the system ensures energy management is optimized by only activating appliances when necessary, based on learned user habits.

Future research could explore contextual and environmental factors, such as weather conditions, time of day, and user behavior patterns, that impact appliance usage. Additionally, the accuracy of sensors, like those detecting room occupancy, could be another key factor contributing to this gap. Improving sensor accuracy or using sensor fusion techniques could enhance model predictions, especially for appliances like AC that are sensitive to room occupancy.

Table 3. The SHAS performance

Performance Metrics	Smartplug		
	S-AC	S-TV	S-WPM
Accuracy (%)	81.9	98	97.6
Precission (%)	79.5	95	85.7
Recall (%)	90.3	98	85
F1 Score (%)	85	97	85

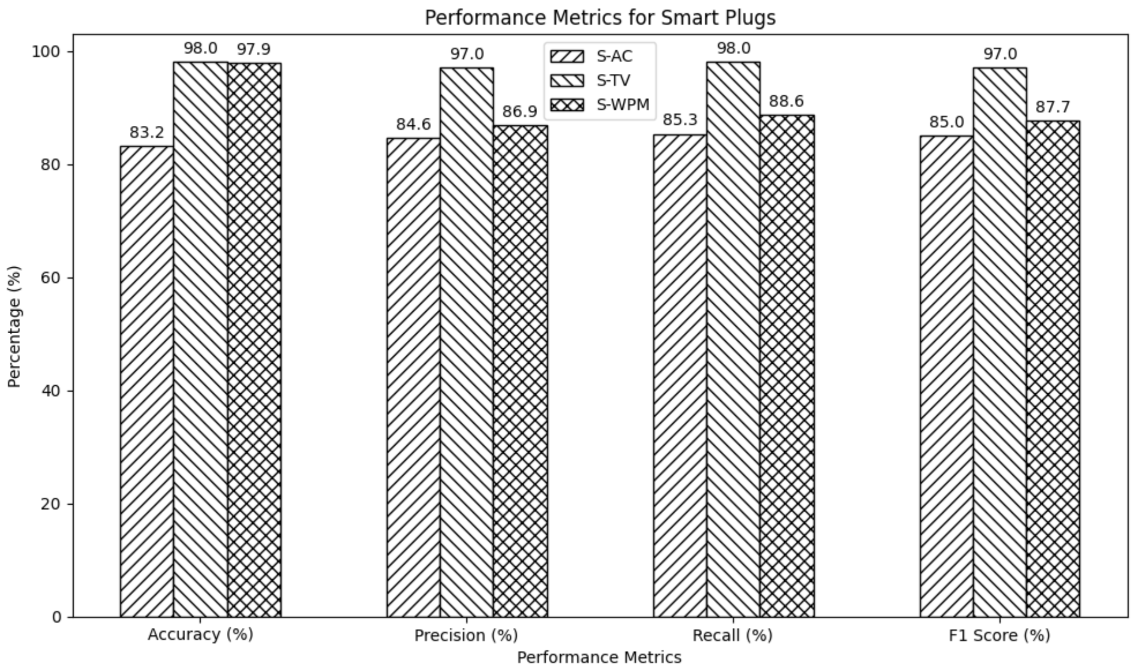

Figure 4. SHAS Confusion matrix performance with comparison of three smartplugs

3.3. SHAS performance evaluation with real-world data

The performance of the above model was further tested with out-of-sample data or external validation data, specifically data outside the range February to April 2024 as described in the methodology section. This study used external validation data from the period of May 14-21, 2024, with a total of 5,520 data points. After pre-processing, 5,109 data points remained, as shown in Figure 5. Figure 5(a) shows the first 6 rows of the dataset, indicating some NaN values. Following pre-processing, including transformation of the TS column and normalization, the data became consistent across all columns as describe in Figure 5(b) dan 5(c).

	TS	TP	S-AC	S-TV	S-WPM	S-MC	S-K	T	H	W	Pb	Pg
0	5/14/2024 12:13	220.2	0.0	0.0	0.0	0.0	1.0	30.2	70.92	26	1.0	0
1	5/14/2024 12:14	231.1	0.0	0.0	0.0	0.0	1.0	30.2	70.92	25	1.0	0
2	5/14/2024 12:15	210.0	0.0	0.0	0.0	0.0	1.0	30.2	70.92	26	1.0	0
3	5/14/2024 12:16	212.5	0.0	0.0	0.0	0.0	1.0	30.2	70.92	25	1.0	0
4	5/14/2024 12:17	211.4	0.0	0.0	0.0	0.0	1.0	30.2	NaN	25	NaN	0
5	5/14/2024 12:18	194.7	0.0	0.0	0.0	0.0	1.0	30.2	70.92	25	1.0	0
6	5/14/2024 12:19	198.8	0.0	0.0	0.0	0.0	1.0	30.2	70.92	25	1.0	0

(a)

	TP	S-AC	S-TV	S-WPM	S-MC	S-K	T	H	W	Pb	Pg	NSM	WS	D	HR	WK
0	203.1	1.0	0.0	0.0	0.0	1.0	30.9	50.0	21.0	1.0	0.0	0	0	4	0.201852	0.000000
1	202.2	1.0	0.0	0.0	0.0	1.0	31.2	50.0	25.0	1.0	0.0	60	0	4	0.201852	0.000694
2	200.2	1.0	0.0	0.0	0.0	1.0	31.3	51.0	21.0	1.0	0.0	120	0	4	0.201852	0.001389
3	207.4	1.0	0.0	0.0	0.0	1.0	31.3	51.0	24.0	1.0	0.0	180	0	4	0.201852	0.002083
4	206.4	1.0	0.0	0.0	0.0	1.0	31.3	51.0	21.0	1.0	0.0	240	0	4	0.201852	0.002778
5	204.6	1.0	0.0	0.0	0.0	1.0	31.3	52.0	13.0	1.0	0.0	300	0	4	0.201852	0.003472
6	200.3	1.0	0.0	0.0	0.0	1.0	31.3	52.0	21.0	1.0	0.0	360	0	4	0.201852	0.004167

(b)

	TP	S-AC	S-TV	S-WPM	S-MC	S-K	T	H	W	Pb	Pg	NSM	WS	D	HR	WK
count	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000	5109.000000
mean	0.449521	0.554707	0.391270	0.169309	0.011744	0.687806	0.697463	0.767144	49.212958	0.888236	0.406929	0.499852	0.283030	4.271873	0.203673	0.499852
std	0.240856	0.497047	0.488082	0.375061	0.107742	0.463434	0.017225	0.125759	25.977056	0.315106	0.491310	0.295109	0.450515	1.574772	0.000158	0.295109
min	0.074583	0.000000	0.000000	0.000000	0.000000	0.000000	0.666000	0.467667	-12.000000	0.000000	0.000000	0.000000	0.000000	2.000000	0.203446	0.000000
25%	0.266667	0.000000	0.000000	0.000000	0.000000	0.000000	0.683429	0.652000	26.000000	1.000000	0.000000	0.236111	0.000000	3.000000	0.203545	0.236111
50%	0.412167	1.000000	0.000000	0.000000	0.000000	1.000000	0.697000	0.811833	45.000000	1.000000	0.000000	0.531250	0.000000	4.000000	0.203645	0.531250
75%	0.615250	1.000000	1.000000	0.000000	0.000000	1.000000	0.710429	0.873167	68.000000	1.000000	1.000000	0.759028	1.000000	6.000000	0.203844	0.759028
max	1.139917	1.000000	1.000000	1.000000	1.000000	1.000000	0.742714	0.897833	106.000000	1.000000	1.000000	0.999306	1.000000	7.000000	0.204143	0.999306

(c)

Figure 5. Pre-processing external validation data: (a) initial of external validation data, (b) external validation data with extracted TS, and (c) external validation data after normlization and cleaning

The LSTM model was then tested on this external validation data for each smartplug (S-AC, S-TV, and S-WPM), with accuracy results shown in Table 4 and Figure 6. The test was divided into three categories: weekdays (Monday to Friday), weekends (Saturday and Sunday), and all days combined. Accuracy for S-AC was lower compared to S-TV and S-WPM, particularly during weekdays, where S-AC achieved 51%, S-TV 66.1%, and S-WPM 83.3%. Interestingly, the accuracy of smartplug S-AC increases during weekends compared to weekdays, reaching 67.3%, while for S-WPM, it remains relatively stable at around 82.2%. Another interesting observation is that S-TV achieving its lowest accuracy of only 47.5% on weekends but fair in weekdays at 66.1%. When considering all days, S-WPM consistently shows higher accuracy compared to S-AC and S-TV at an average score 82.8%. This suggests that the model may be better at predicting the behavior of the water pump machine compared to the air conditioner and television. Additionally, the modeling of S-WPM is not influenced by time factors, as evidenced by the low correlation coefficient values r_{xy} for S-WPM with input feature variables in the time category such as [WK, HR, D, WS, NSM], with a value of |0.01| (refer back to Figure 3). These variations in accuracy, particularly for S-AC and S-TV, highlight the importance of considering time-based user habits in preventing overloads. During weekends, when usage patterns are less predictable, the system's ability to adjust to these changes becomes critical for both safety and energy efficiency. By dynamically adapting to different scenarios, the system ensures that appliances are not left running unnecessarily, thereby reducing the risk of overloads and optimizing power consumption.

Table 4. SHAS performance using external validation data

Performance metrics	Type of data	Smartplug		
		S-AC	S-TV	S-WPM
Accuracy (%)	Weekdays	51	66.1	83.3
	Weekend	67.3	47,5	82.2
	All Days	55	60.8	83
	Average of accuracy (%)	57.8	58.1	82.8

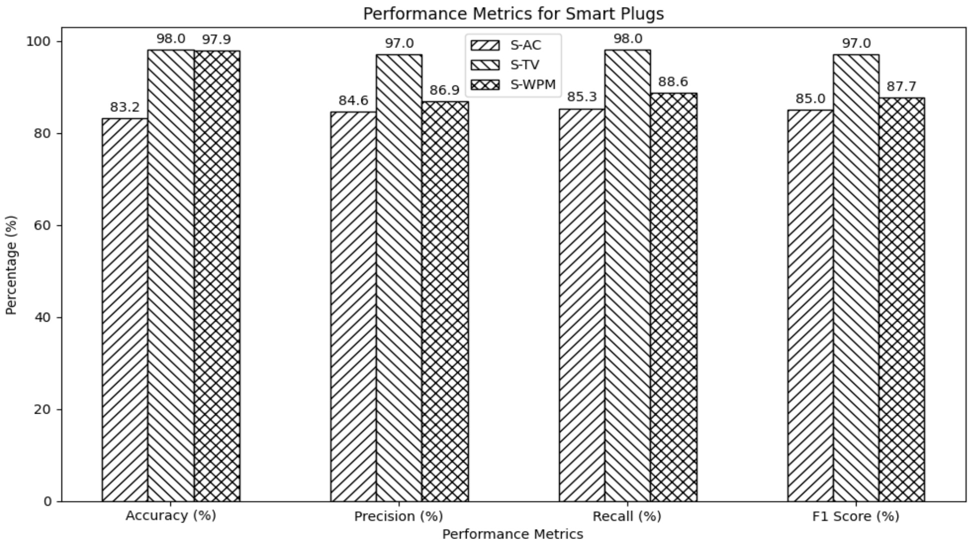

Figure 6. SHAS automation performance comparation using external validation data

To conclude, this study reveals variability in prediction accuracy among smart plugs, with air conditioners (S-AC) showing less consistency compared to water pumps (S-WPM) and televisions (S-TV). The stable accuracy of S-WPM indicates less sensitivity to temporal factors, while S-AC and S-TV are more influenced by user behavior and timing. These findings underscore the importance of considering appliance-specific characteristics and contextual factors such as user habits, environmental conditions, device interactions, usage priorities, and energy availability in predictive models for SHAS.

Recent observations indicate that SHAS can naturally adapt to household behavior, enhancing safety by preventing electrical overloads. This ability not only safeguards appliances from potential damage caused by power surges but also classifies energy usage for better management. By minimizing unnecessary energy consumption, the system supports more sustainable and cost-effective power use in the long term. These dual benefits highlight the effectiveness of the system in technical performance and practical applications, emphasizing the need for future research to further refine these models to improve accuracy and power management.

4. CONCLUSION

The Natural SHAS has been successfully implemented in this study. The use of LSTM deep learning is highly relevant to sequential time-based data like the AHEACS dataset. This study highlights the importance of preventing overloads that can cause MCB trips by automatically controlling electrical appliances based on awareness of household habits. This approach addresses a crucial issue for enhancing household safety and optimizing energy management. The LSTM model classification yields excellent results in automating the Television (S-TV) and Water Pump Machine (S-WPM) smartplugs, achieving accuracies of 98% and 97.6%, respectively, though slightly lower for the Air Conditioner (S-AC) at 81.9%. The proposed layered model SHAS concept performs exceptionally well when tested with new or upcoming data, as discussed in the evaluation external validation data, spanning from May 14 to 21, 2024, with a total of 5,109 data points. This evaluation divides the data into three types: weekdays, weekends, and all data, to observe differences in accuracy across different behaviors. Unlike other intelligent SHAS systems that rely on static datasets, the natural SHAS concept dynamically profiles and controls each electrical appliance based on varying household behaviors. Interestingly, the system's accuracy fluctuates across different smartplugs depending on the day. For instance, the S-AC plug achieves the lowest accuracy of 51% during weekdays but improves to 67.3% on weekends. Similarly, the S-TV plug exhibits the lowest accuracy of 47.5% on weekends, with better performance during weekdays. Meanwhile, the S-WPM demonstrates consistent performance, with an average accuracy of 82.8% across all conditions. This stability in S-WPM suggests that the system effectively controls the water pump machine, regardless of time-based variations. These variations are attributed to changes in operational conditions, such as the system's response to less predictable usage patterns during holidays, rather than user behavior directly influencing the system. Future research should focus on enhancing the natural SHAS model by incorporating advanced AIoT techniques.

This could involve using more sophisticated deep learning for precise control and integrating IoT to account for additional factors such as appliance priorities and contextual conditions. Developing methods to manage complex household energy use and adapt to varying scenarios could further improve the system's accuracy and efficiency in preventing overloads.

ACKNOWLEDGEMENTS

Author would like to express gratitude to the Faculty of Electrical and Electronic Engineering (FKEE) at Universiti Tun Hussein Onn Malaysia and Politeknik Caltex Riau for their full support of this research.

REFERENCES

[1] "Tarif Adjustment." Accessed: Jan. 02, 2024. [Online]. Available: https://web.pln.co.id/pelanggan/tarif-tenaga-listrik/tariff-adjustment

[2] Su-Kang Park *et al.*, "A study on the reduction of in-rush current for energy saving of single phase induction motor," in *ICEMS'2001. Proceedings of the Fifth International Conference on Electrical Machines and Systems (IEEE Cat. No.01EX501)*, Int. Acad. Publishers, pp. 67–71, doi: 10.1109/ICEMS.2001.970611.

[3] S. A. Suhandini, A. Suhendi, L. Andiani, S. M. Oktaviani, and I. P. Sari, "Priority based automatic electrical load control for the prevention of electrical overload in household installations," in *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, IEEE, Dec. 2022, pp. 761–765, doi: 10.1109/ICITISEE57756.2022.10057803.

[4] Y. Lin, Z. Xiong, and J. Zhu, "Optimization of the electricity consumption behaviors of users under uncertain electricity prices and consumption patterns," *Energy Reports*, vol. 9, pp. 646–657, Sep. 2023, doi: 10.1016/j.egyr.2023.04.336.

[5] Q. Antoine and K. Vliegen, "Impact of MCB transient overvoltages on household appliances," in *27th International Conference on Electricity Distribution (CIRED 2023)*, Institution of Engineering and Technology, 2023, pp. 251–255. doi: 10.1049/icp.2023.0285.

[6] T. Alquthami and A. P. S. Meliopoulos, "Smart house management and control without customer inconvenience," *IEEE Trans Smart Grid*, vol. 9, no. 4, pp. 2553–2562, Jul. 2018, doi: 10.1109/TSG.2016.2614708.

[7] M. Susantok, "Smart power usage control system using IoT-based highest value elimination method," *Jurnal Elektro dan Mesin Terapan*, no. Vol. 8 No. 2 (2022), pp. 104–112, Nov. 2022, doi: 10.35143/elementer.v8i2.5552.

[8] P. Rodriguez-Garcia, Y. Li, D. Lopez-Lopez, and A. A. Juan, "Strategic decision making in smart home ecosystems: A review on the use of artificial intelligence and Internet of things," *Internet of Things*, vol. 22, p. 100772, Jul. 2023, doi: 10.1016/j.iot.2023.100772.

[9] O. E. Ghati, O. Alaoui-Fdili, O. Chahbouni, N. Alioua, and W. Bouarifi, "Artificial intelligence-powered visual internet of things in smart cities: a comprehensive review," *Sustainable Computing: Informatics and Systems*, p. 101004, May 2024, doi: 10.1016/j.suscom.2024.101004.

[10] A. A. Medhekar, M. Duggal, S. K. Singh, and P. Nayadkar, "Healthcare and internet of things: The future of artificial intelligence and IoT technology," *Oral Oncology Reports*, vol. 10, p. 100279, Jun. 2024, doi: 10.1016/j.oor.2024.100279.

[11] V. Sheth, A. Priyal, K. Mehta, N. Desai, and M. Shah, "Schematized study for tackling COVID-19 with machine learning (ML), artificial intelligence (AI), and internet of things (IoT)," *Intelligent Pharmacy*, Apr. 2024, doi: 10.1016/j.ipha.2024.04.003.

[12] A. Khallel Al-Ani, S. U. A. Laghari, H. Manoharan, S. Selvarajan, and M. Uddin, "Improved transportation model with internet of things using artificial intelligence algorithm," *Computers, Materials and Continua*, vol. 76, no. 2, pp. 2261–2279, 2023, doi: 10.32604/cmc.2023.038534.

[13] J. Jaihar, N. Lingayat, P. S. Vijaybhai, G. Venkatesh, and K. P. Upla, "Smart home automation using machine learning algorithms," in *2020 International Conference for Emerging Technology (INCET)*, IEEE, Jun. 2020, pp. 1–4, doi: 10.1109/INCET49848.2020.9154007.

[14] H. Yue, K. Yan, J. Zhao, Y. Ren, X. Yan, and H. Zhao, "Estimating demand response flexibility of smart home appliances via NILM algorithm," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, Jun. 2020, pp. 394–398. doi: 10.1109/ITNEC48623.2020.9084961.

[15] O. Munoz, A. Ruelas, P. Rosales, A. Acuña, A. Suastegui, and F. Lara, "Design and development of an IoT smart meter with load control for home energy management systems," *Sensors*, vol. 22, no. 19, p. 7536, Oct. 2022, doi: 10.3390/s22197536.

[16] A. Lentzas and D. Vrakas, "Machine learning approaches for non-intrusive home absence detection based on appliance electrical use," *Expert Systems with Applications*, vol. 210, p. 118454, Dec. 2022, doi: 10.1016/j.eswa.2022.118454.

[17] L. Cascone, S. Sadiq, S. Ullah, S. Mirjalili, H. U. R. Siddiqui, and M. Umer, "Predicting household electric power consumption using multi-step time series with convolutional LSTM," *Big Data Research*, vol. 31, p. 100360, Feb. 2023, doi: 10.1016/j.bdr.2022.100360.

[18] R. E. Alden, H. Gong, C. Ababei, and D. M. Ionel, "LSTM forecasts for smart home electricity usage," in *2020 9th International Conference on Renewable Energy Research and Application (ICRERA)*, IEEE, Sep. 2020, pp. 434–438, doi: 10.1109/ICRERA49962.2020.9242804.

[19] H. Yar, A. S. Imran, Z. A. Khan, M. Sajjad, and Z. Kastrati, "Towards smart home automation using IoT-enabled edge-computing paradigm," *Sensors 2021*, vol. 21, no. 14, p. 4932, Jul. 2021, doi: 10.3390/S21144932.

[20] R. P. Pratama, "ESP8266-based home light controller using MQTT protocol," *TESLA: Jurnal Teknik Elektro*, vol. 22, no. 1, p. 56, Mar. 2020, doi: 10.24912/tesla.v22i1.7862.

[21] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, *Edge AI: Convergence of Edge Computing and Artificial Intelligence*, Springer Singapore, 2020, doi: 10.1007/978-981-15-6186-3/COVER.

[22] H. Elayan, M. Aloqaily, F. Karray, and M. Guizani, "Internet of behavior (IoB) and explainable AI systems for influencing IoT behavior," *IEEE Network*, vol. 37, no. 1, pp. 62–68, Sep. 2021, [Online]. Available: http://arxiv.org/abs/2109.07239

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016. Accessed: Nov. 19, 2023. [Online]. Available: https://www.deeplearningbook.org/

[24] M. Abumohsen, A. Y. Owda, and M. Owda, "Electrical load forecasting using LSTM, GRU, and RNN algorithms," *Energies (Basel)*, vol. 16, no. 5, p. 2283, Feb. 2023, doi: 10.3390/en16052283.

[25] Y.-H. Lin, H.-S. Tang, T.-Y. Shen, and C.-H. Hsia, "A smart home energy management system utilizing neurocomputing-based time-series load modeling and forecasting facilitated by energy decomposition for smart home automation," *IEEE Access*, vol. 10, pp. 116747–116765, 2022, doi: 10.1109/ACCESS.2022.3219068.

[26] L. Palagi, A. Pesyridis, E. Sciubba, and L. Tocci, "Machine learning for the prediction of the dynamic behavior of a small scale ORC system," *Energy*, vol. 166, pp. 72–82, Jan. 2019, doi: 10.1016/j.energy.2018.10.059.

[27] N. Mahesh, J. J. Babu, K. Nithya, and S. A. Arunmozhi, "Water quality prediction using LSTM with combined normalizer for efficient water management," *Desalination Water Treat*, vol. 317, p. 100183, Jan. 2024, doi: 10.1016/j.dwt.2024.100183.

[28] R. Olu-Ajayi *et al.*, "Building energy performance prediction: A reliability analysis and evaluation of feature selection methods," *Expert Systems with Applications*, vol. 225, p. 120109, Sep. 2023, doi: 10.1016/j.eswa.2023.120109.

[29] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020, doi: 10.1186/s12864-019-6413-7.

[30] S. Manna, "Small sample estimation of classification metrics," in *2022 Interdisciplinary Research in Technology and Management (IRTM)*, IEEE, Feb. 2022, pp. 1–3, doi: 10.1109/IRTM54583.2022.9791645.

BIOGRAPHIES OF AUTHORS

Mochamad Susantok is currently pursuing a Ph.D. in Faculty Electrical and Electronic Engineering (FKEE) at University Tun Hussein Onn Malaysia. His research interests include Aritifical Intelligent of Things (AIoT), Computer Network Automation, and Edge/Cloud Computing. He holds a bachelor's and master's degree in electrical engineering from Institut Teknologi Sepuluh Nopember Indonesia. Prior to pursuing his Ph.D., he gained valuable experience working as a lecturer at Politeknik Caltex Riau. He has published several articles in reputable journals and presented his research findings at various international conferences. He can be contacted at email: ge230025@student.uthm.edu.my or santok@pcr.ac.id.

Farhana Ahmad Po'ad received her first degree in Electrical and Electronics Engineering from Kolej Universiti Tun Hussein Onn in 2003. She then continued her studies and received a Master of Electrical Engineering from Universiti Tun Hussein Onn Malaysia (UTHM) in 2007. She obtained her Ph.D. in 2016 in the research field of Wireless Sensor Networks (WSN) and the application of RFID and IoT from Universiti Sains Malaysia (USM). She is currently a senior lecturer in the Faculty of Electrical and Electronics Engineering at UTHM. She has been entrusted with managing research grants across 13 research contracts. She has also supervised and co-supervised more than 3 master's students and 2 Ph.D. students in the last few years. She has authored or co-authored more than 24 publications: 8 proceedings, 9 journals, 1 book chapter, 2 books, and 4 general articles, with an H-index of 5 (Scopus) and more than 92 citations (Scopus). Her research interests include WSN, the application of RFID and IoT. She can be contacted at email: farhana@uthm.edu.my.

Ariffuddin Joret is a senior lecturer in Faculty of Electrical and Electronics Engineering, Universiti Tun Hussein Onn Malaysia (UTHM). He received his first degree in electrical and electronic engineering from Universiti Teknologi MARA (UiTM) in 2000. In 2006, he recived his Master of Science (Electrical and Electronics Engineering) in Neural Network from Universiti Sains Malaysia (USM). He completed his Ph.D. in 2018 in signal processing field from Universiti Tun Hussein Onn Malaysia (UTHM). His current research interests including ground penetrating radar, radar and IoT system development including application. He can be contacted at email: ariff@uthm.edu.my.

Maulina Hilwa Salsabillah was born on May 27, 2002, in Pekanbaru, Indonesia. She completed her undergraduate studies with cum laude honors in Electronic and Telecommunication Engineering at Politeknik Caltex Riau. With a strong interest in exploring new fields, Maulina expanded her skills into data-related areas without leaving behind her background in Electronic and Telecommunication Engineering. In addition to her academic pursuits, she has been active in student organizations and committees, aiming to build connections and engage in various new activities. In 2022, she was awarded a research grant by BPDPKS, a program supported by Indonesia's Ministry of Finance. As the eldest of three siblings, Maulina is motivated to consistently achieve her best, setting an example for her younger siblings. She can be contacted at email: maulinahs27@gmail.com.