RERIC

www.rericjournal.ait.ac.th

Data-Flow Programming based Non-Intrusive Load Monitoring for Electricity in Remote Area

Syafrudi*, Fransisco Danang Wijaya*,[1], and Sarjiya*

ARTICLE INFO

Article history:
Received 14 September 2021
Received in revised form
15 November 2021
Accepted 29 November 2021

Keywords:
Data flow programming
Electricity in remote areas
LabVIEW
Load disaggregation
NILM.

ABSTRACT

The use of electrical energy in remote areas must be carried out efficiently. This study proposed a load monitoring method in remote areas. This study developed a Non-Intrusive Load Monitoring (NILM) based on Data-Flow Programming (DFP) by applying a bagging decision tree algorithm to conduct load disaggregation. This study built the DFP and the Graphical User Interface (GUI) in LabVIEW connected to power sensor ADE9153A on Arduino UNO via serial communication. This experiment was conducted on the LabVIEW 2020 running on an Intel i5 2400-3.1 GHz CPU, 16 GB RAM, and 64-bit operating system computer. This study produced a good performance of NILM with 0.9617 of accuracy and 0.9728 of f1-score. The proposed method of the NILM process was suitable for electricity in remote areas because the DFP used in the algorithm is easy to understand, easy to operate, and inexpensive to build. Finally, the NILM technique can improve the efficiency of used electrical energy in remote areas. By applying NILM, the operator can determine the priority of which devices should be ON or OFF at a particular time as needed. In addition, the NILM can contribute to the balancing of small and weak microgrids in scenarios of high renewable energy penetration.

1. INTRODUCTION

As the largest archipelago country globally, Indonesia has more than 17,000 islands, of which about 6000 are inhabited [1]. Indonesia has various potential natural resources and natural beauty that can support the tourism sector [2]. However, with the spread of Indonesia's territory into many islands, some facilities, one of which is electricity, cannot be spread evenly. Whereas now almost all household appliances are operated using electrical energy. So that, electricity is a significantly needed in everyday activities recently.

Electricity in remote areas must be carried out efficiently because generating electricity in remote areas is difficult. The electricity generated in remote areas is obtained by utilizing the potential energy available in the location. Therefore, electrical energy management in the generation, distribution, and utilization must be carried out correctly.

By carrying out the monitoring process on used equipment, electrical energy efficiency can be increased by up to 12% [3]. This process needs to be carried out in electricity in remote areas, considering the importance of efficient use of electrical energy in these areas. In general, the process of monitoring electrical equipment can be carried out in two ways, specifically Intrusive Load Monitoring (ILM) and Non-Intrusive Load Monitoring (NILM). These two methods differ in that the ILM sensor is installed on every piece of electrical equipment being monitored. Whereas the sensor, in NILM, is only installed in the established electrical circuits' initial path. Compared to ILM, NILM has many advantages, such as it is easy to implement, affordable, and easy to develop [4]. In so doing, NILM has become a rapidly growing research topic [5].

Data-Flow Programming (DFP) is a programming method that internally represents an application as a directed graph, similar to a data flow diagram [6]. DFP aims to lower syntactic and cognitive barriers, such as programming language in general (*e.g.*, Python, C, C++, and so on) [7].

The objective of this study is to implement a load monitoring process to realize Demand Side Management (DSM) in a remote area's electricity. In addition, how to implement the process is also described. The authors implemented DFP in the NILM process by employing a bagging decision tree algorithm. This study uses LabVIEW 2020 to design the algorithm and build the Graphical User Interface (GUI). A power feature which is consists of active power, reactive power, and apparent power is utilized to conduct load disaggregation. The ADE9153A power sensor programmed with microcontroller Arduino UNO is used and connected to the computer via serial communication. This study uses a sample of household appliances: a led lamp, a rice cooker, a fan, a fridge, and a television. Employing the decision tree algorithm using DFP in the NILM process, the proposed method can improve electrical energy efficiency

* Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Yogyakarta, 55281, Indonesia.

1 Corresponding author;
Tel: +62 (274) 552305, Fax: +62 (274) 552305.
E-mail: danangwijaya@ugm.ac.id

in remote areas. By applying NILM, the operator can determine the priority of which appliances should be ON or OFF at a particular time as needed. In addition, the NILM can contribute to the balancing of small and weak microgrids in scenarios of high renewable energy penetration.

2. RELATED WORK

NILM has been widely applied in previous studies. For example, researchers in [8] applied NILM to dairy farms by adopting a deep learning development algorithm. They proposed the method for dairy farms in four selected appliances of targeted disaggregation.

Researchers in [9], through sensitive data collection, implemented artificial intelligence methods based on NILM load decomposition of each electrical equipment ON/OFF status information. They also checked student dormitory electrical devices online, designed and implemented network cloud configuration software and WeChat applet display equipment. A prototype of a network-based demonstration system was designed and built according to the student dormitory background. As a result, the real-time monitoring of the behavior of electrical appliances is more convenient. It significantly improves the safety of the domestic microgrid.

Researchers in [10] proposed an algorithm that can run on an economical and energy-efficient embedded system. They designed an intelligent meter solution to coordinate the embedded system in an offshore oil platform power system with the meter. All information was handled in the embedded system. It can recognize the behavior of electrical appliances, giving an information establishment for the demand-side response. From the review referenced above, it very well may be inferred that NILM has been widely applied to certain conditions and positively impacts electrical efficiency.

Usually, NILM incorporates the accompanying five stages: data collection, data preprocessing, event detection, feature extraction, and load identification [11]. There are various methods of NILM which have been proposed and implemented. To produce NILM with good performance, one of the critical steps is feature extraction. Determining the features becomes a crucial decision for the researchers at this step.

Steady-state features are the features extracted during the consistent state activity of electrical loads. The features are adapted for load identification, such as variation in the power, time, and frequency domain features, voltage-current trajectories, and voltage noise. Power-related features incorporate the root mean square (RMS) of voltage and current, real power, reactive power, apparent power, power factor, and shape factor from voltage and current signs [12].

In general, a steady-state feature approach is adopted when employing low sampling rate data. However, insufficient information about load parameters may exist during the event detection. Moreover, similar power consumption in steady-state operation may exist in different appliances. And so forth, the objection in NILM with low sampling data is to disaggregate appliances that have comparative power utilization.

Unsupervised NILM algorithms by applying Graph Laplacian Regularization (GLR) accomplished state-of-the-art [13]. Moreover, a Graph Signal Processing (GSP) algorithm was proposed on event-based NILM in [13]-[15]. The proposed method based on GSP can be carried out to better the outcome of diverse event-based NILM approaches. Unfortunately, the implementation of the GSP algorithm using the DFP approach is not easy to do.

Lower odd-numbered harmonics were employed for the NILM feature in [16]. The study implemented a bagging decision tree algorithm to disaggregate loads. The method can improve recognizing different loads. This method is under the concept of data flow. Therefore, this study implemented this method. However, handling a high sampling rate data makes computer performance faster and lose more electrical energy. This study considered the NILM for use in remote areas and needs attention to electrical energy efficiency.

Otherwise, very low sampling rate data was used in [17]. The study used active power (P) as a feature. Data retrieval was carried out in 15-30 minutes. Using very low sampling rate data makes the disaggregation performance of concurrent appliances lower.

This study chose to develop NILM using DSP because DFP is a method that is easier to understand and modify than other programming methods. Considering the issues mentioned earlier, generally, remote areas lack human resources equipped with programming skills.

Previously DFP was employed in [7] as a visual programming language that allows end-users to create customized programs in their smart home. Although visual programming aims to lower syntactic and cognitive barriers, existing programming languages lack simple methods for accessing innovative space services around the home. Therefore, it can limit the innovative capabilities of users. By applying the DFP, modifying the NILM system can be done more quickly if desired changes to the system.

In this study, several features to improve NILM's performance was applied. Low sampling rate data was employed to minimize the computational process and gain the sensor's information to be read in more detail.

3. METHODOLOGY

3.1 Bagging Decision Tree Model

Quinlan first introduced the decision tree in 1986, which was called ID3 and later developed into C4.5. The decision tree is perhaps the most popular way to classify data. A decision tree is made by separating the attribute values into branches for every probability. Then, at this point, it works via looking for the root to the branch until the class of an item is found.

Assuming the sample set, T is divided into $T_1, T_2, ..., T_S$ subsets based on *s* various values of the discrete attribute A. Then, the data gain rate is characterized as a ratio $(A,T) = gain(A,T)/split(A,T)$ collected by partitioning the sample set T by discrete attribute A. The gain can be realized as follows:

$$gain(A,T) = \inf(T) - \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \inf(T_i) \quad (1)$$

in which *n* different bootstrapped training datasets.

The split method is an essential factor in building a decision tree algorithm. In this study, the split value is obtained by the following equation:

$$split(A,T) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \log_2 \left(\frac{|T_i|}{|T|} \right) \quad (2)$$

The decision tree was widely applied in the previous NILM. Primarily the extension of the decision tree is bagging, random forest, and AdaBoost. This study implemented the bagging technique on the decision tree. Bagging stands for bootstrap aggregating. Bagging is a technique in a decision tree algorithm to decrease the difference in a decision tree prediction.

The bagging technique in the decision tree comprises three stages. The first stage is separating the columns and rows of the original information and dividing it into several information subsets. Afterward, create a classifier for the subset of information respectively. The similarity and dissimilarity segregations are made for each information subset. At long last, the more significant part vote is taken on to select the best out of all segregations. As can be seen, Figure 1 represents the bagging decision tree model. Meanwhile, its equation is illustrated as follows:

$$P(c \mid f) = \arg\max_n (P_1, P_n) \quad (3)$$

Where n represents the different bootstrap training datasets, each tree is trained with bootstrap to obtain the $P_i(c \mid f)$ value. Then the prediction results are averaged.

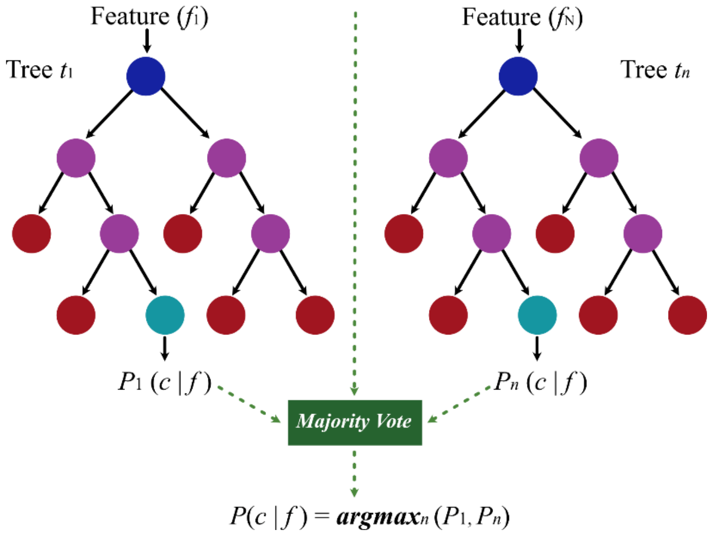

Fig. 1. Bagging decision tree.

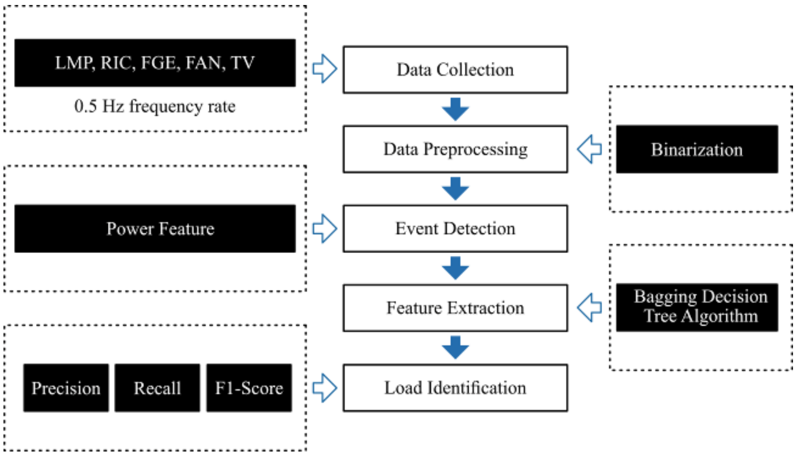

Fig. 2. Flow chart diagram.

3.2 Flow Chart Diagram

This study built the NILM system in 5 stages: data collection, data preprocessing, event detection, feature extraction, and load identification. Each step is described further in the next section. It can be seen in Figure 2, which represents the flow chart diagram of this study to make it easier to understand.

3.3 Implementation of Practical Situation

In general, the proposed NILM method referred to conditions in remote areas applying off-grid-based electrification systems. This study assumed three main parts to the electrification system in remote areas: generation, distribution network, and consumers. Between the generation and distribution network is a monitoring station. The monitoring station monitors the generator's performance and the quality of the generated voltage. It also controls the operation of the electrical system. The proposed NILM method was located at the monitoring station, as shown in Figure 3.

Sensor selection is crucial because it is related to the quality of data sensing. This study employed the ADE9153A-shield auto-calibration sensor, which can be connected directly to the Arduino Uno microcontroller. In the practical situation of usage on a grid with a sizeable current rating, this sensor cannot be applied due to the low current rate of the sensor.

Several things need to be considered in implementing the sensor in practical conditions. First, the sensor needs to meet the specifications required for electricity in the remote area. Furthermore, because this study uses power as the feature, the sensor must be capable of reading active power, reactive power, and apparent power. Then, some types of sensors have limitations in data transfer to the computer. Some can do this work, but some just can read the parameter. To use the sensor as part of the NILM process, we need to pay attention to the sensor that can transfer the data to the computer.

Figure 4 describes the proposed NILM technique. In carrying out data communication between the sensor and the computer, serial communication using USB A to B connector was implemented. However, another way of communication can be applied considering certain factors in practical situations.

Data processing was done by employing LabVIEW 2020, which is DFP-based programming software. The software ran on an Intel i5 2400-3.1 GHz CPU, 16 GB RAM, and 64-bit operating system computer. In practical usage, the different specifications of the computer may affect the performance of the software.

In this study, the authors used some samples of household appliances. In practical situations, the designer can make the decision for which appliances to be monitored. This selection is based on the conditions and situations in the remote area.

4. EXPERIMENTS

4.1 Household Electrical Appliances

Household electrical appliances are electrical loads that consist of inductive (L), resistive (R), or capacitive (C) components, and a combination of them. Each electrical device has different characteristics from one another. Furthermore, the type of load is classified into several types based on certain factors. According to the operating conditions, electrical appliances are divided into several types of loads. There are ON/OFF loads, Finite State Machines (FSM) loads, and Continuously Variable Device (CVD) loads [18]. An ON/OFF load is a load with two different states where each has a constant level of energy consumption (such as lights and toasters). On the other hand, an FSM load has a limited number of switching states (such as washing machine and fruit mixer). Furthermore, A CVD load is a load without a precise number of switching states (such as computers and refrigerators).

Other load classifications have been presented in [17]. They divided electrical appliances into different types by considering the operating cycles. Type 1 is a device that has a long operating cycle. In type 1, the device has a flash time of more than 30 minutes, has a specific energy consumption, and considers various operating modes. Type 2 is appliances with heating and cooling components. Heating, ventilation, and air conditioning (HVAC) equipment are affected by outdoor temperatures and infrequent equipment. Type 3 is a small device with low power and operated for a long time. Appliances in type 3 are operated at a constant power over a specific time, for example, lamps and televisions. Type 4 is a small device with low power and operates in a short time. Appliances in type 4 are not operated more than 30 minutes a day, for example, vacuum cleaners and kettles.

In this study, five different electrical devices, as shown in Table 1, are described. The selection of those electrical devices took into account the tendency to use these devices in daily activities. As well as the selection of the load must represent all categories of the load mentioned earlier. A lamp (LMP), a rice cooker (RIC), a refrigerator (FGE), a fan (FAN), and a television (TV) were employed as electrical appliances. LMP belongs to ON/OFF load; RIC and FAN belong to FSM load; TV and FGE belong to CVD load.

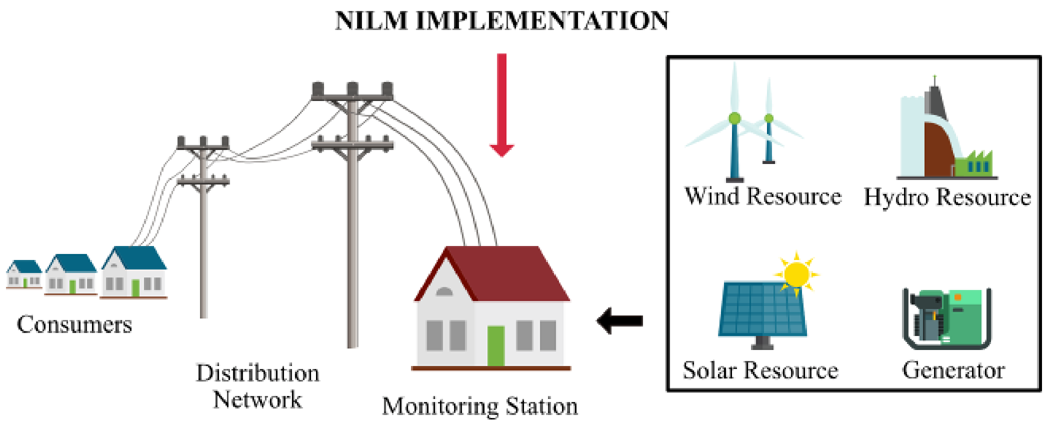

Fig. 3. NILM implementation on remote area electrification.

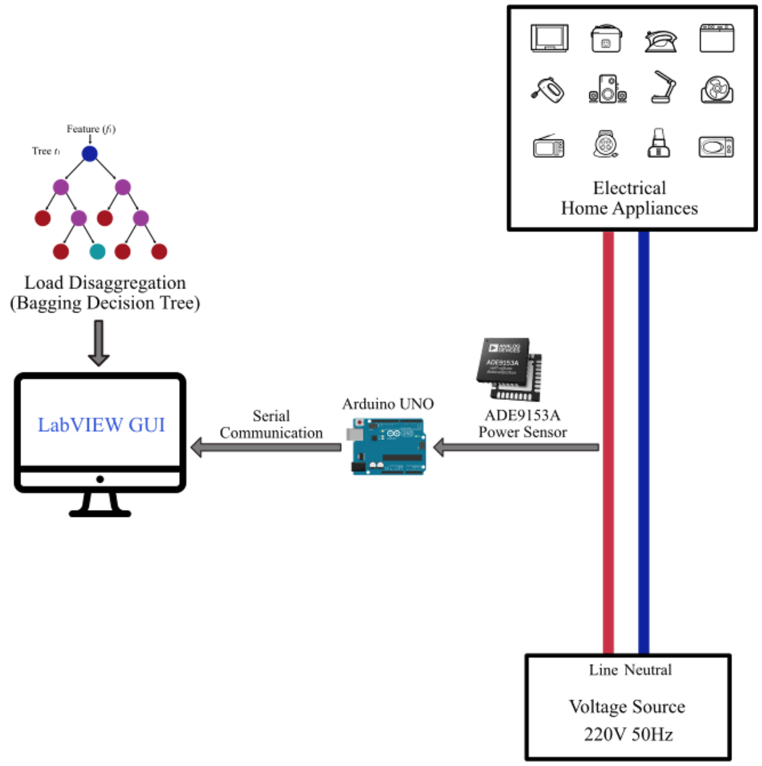

Fig. 4. Proposed NILM process.

Table 1. Electrical appliances used.

Appliances	Specifications	Load Type
LED Lamp (LMP)	Philips LED 220V 6W	ON/OFF
Rice Cooker (RIC)	Miyako MCM-606A 220V 350W	FSM
Fan (FAN)	Cosmos Pedestal Fan 220V 50Hz 45W	FSM
Fridge (FGE)	Panasonic NR-A198G 220V 50Hz 80W 0.68A	CVD
Television (TV)	Polytron PLD 40B150 220V 50Hz	CVD

Table 2. Hardware specification.

Hardware	Specifications
Computer	Intel i5 2400-3.1 GHz CPU, 16 GB RAM
Power Sensor	Arduino-viable, single-phase energy measurement shield with the ADE9153A Nominal Current: 5 A Maximum Current: 10 A Maximum Voltage: 240 V L-N
Serial Communication	USB A to B 2.0
Microcontroller	Arduino UNO Atmega328P

4.2 Hardware and Software

Hardware is utilized as equipment to measure and record the information. This study employed some hardware: a computer, a power sensor, and a data communicator. In this study, the computer was employed for the data calculations in the NILM system process. Furthermore, the computer displayed GUI on the monitor screen. A power sensor was applied to retrieve data from the power grid. The USB A to B cable connected the sensor to the computer as a serial communicator.

This study employed the Arduino UNO microcontroller. It is because the ADE9153A sensor can only work when processed through the microcontroller. The separate microcontroller may not be needed in some power sensors because it is usually produced as a set in the sensor. Some additional hardware may be added to support the system to work correctly. The following Table 2 shows the hardware and its specifications used to build this system.

LabVIEW 2020 was handled as the main software in this study. LabVIEW is a graphics-based programming environment software created by National Instrument. LabVIEW can be downloaded for free through its official website (https://www.ni.com). LabVIEW was able to develop a DFP-based NILM system. Another software is Arduino IDE which is played to program the sensor.

4.3 Data Collection

Data collection was done by measuring the electrical device directly at a specific time interval. The data collection method utilized a low sampling rate, *ie*. 0.5 Hz data sampling. It means the sensor sensed the data and collected it every two seconds. RMS current (I), RMS voltage (V), active power (P), reactive power (Q), apparent power (S), power factor (λ), frequency (F), and temperature (T) were obtained by employing the ADE9153A sensor.

Active power (P) was used in this study as an event. The data were collected from the sensor. Then the data were exported to the computer to get analytical results for the event. The sensor resulted in original active power data, as shown in Figures 5 to 9.

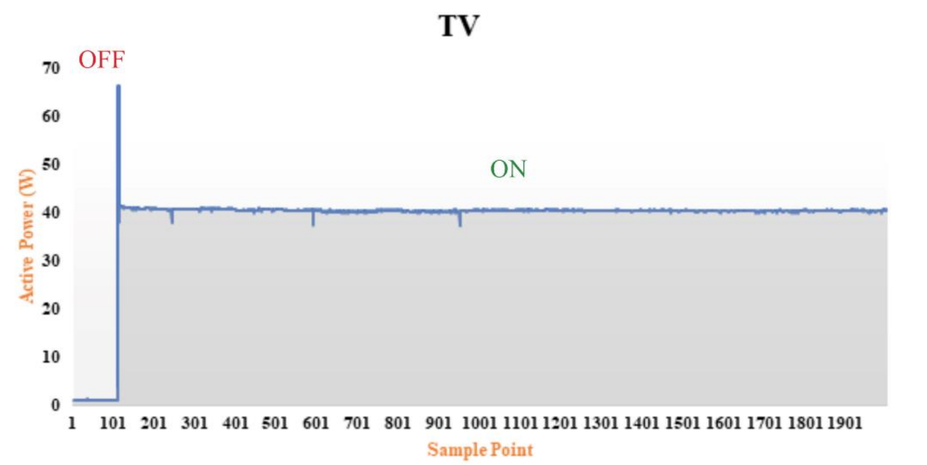

Fig. 5. Original active power information of TV.

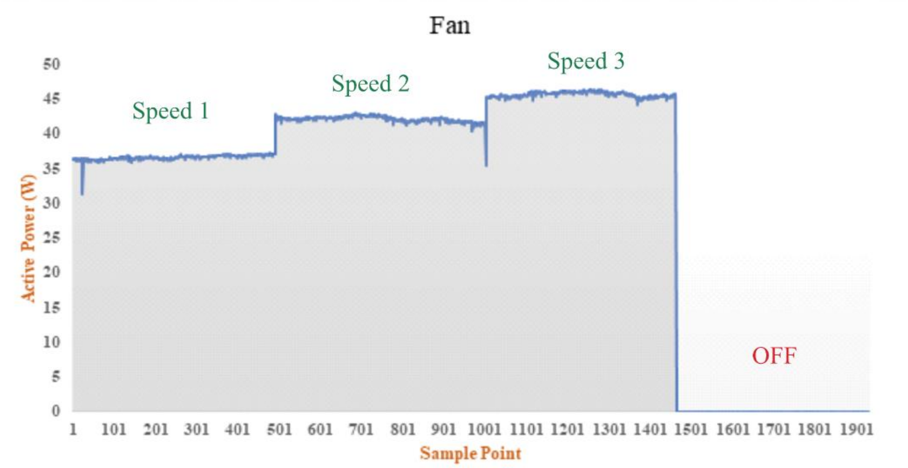

Fig. 6. Original active power information of FAN.

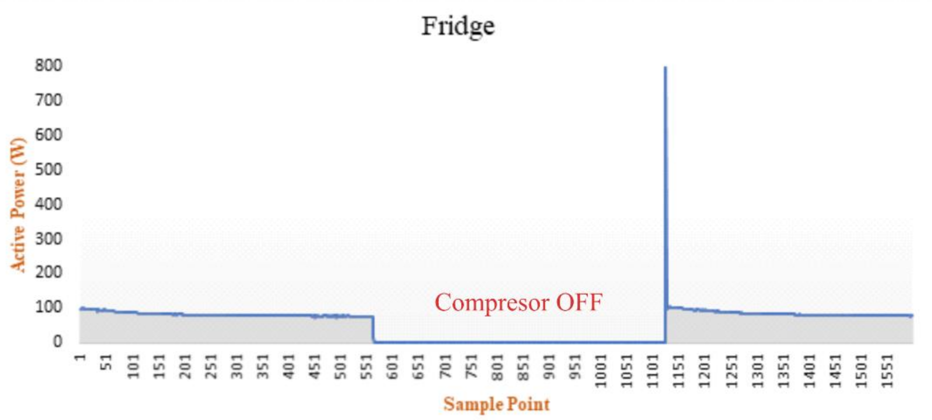

Fig. 7. Original active power information of FGE.

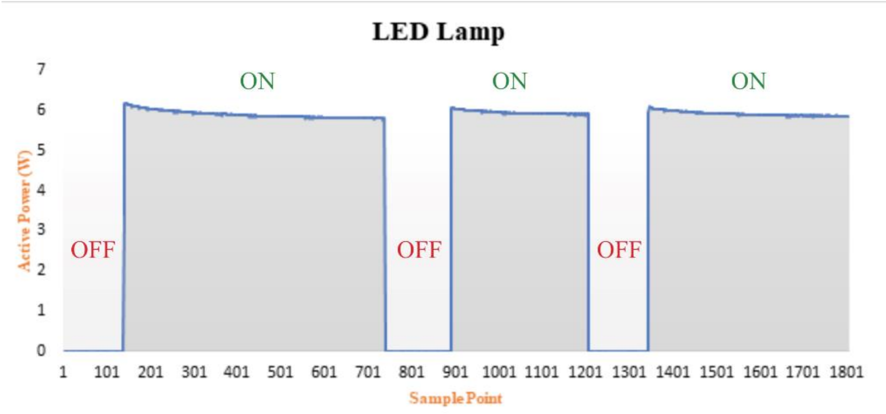

Fig. 8. Original active power information of LMP.

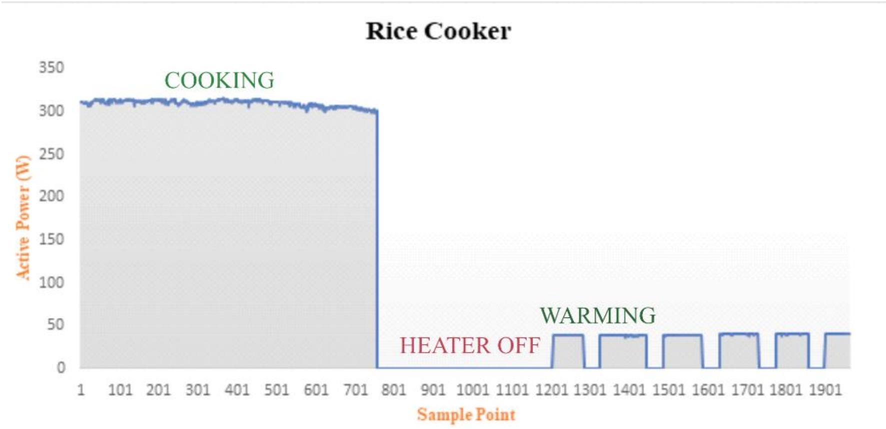

Fig. 9. Original active power information of RIC.

4.4 Data Preprocessing

As shown in Figures 5 to 9, the *P* data as an event is unstable and tends to change over time. This issue can be considered a change event and can reduce NILM's performance. For this reason, the initial data processing was undertaken by binarization. Binarization is intended to obtain 0 (when the appliance is OFF) or 1 (when ON). Therefore, *P* data that has been binarized P_bin was obtained.

The means of the binarization calculation are as per the following.

Step 1 is determining the threshold of the original active power data. If *P* > threshold, enter Step 3. If not, go to Step 2. In this condition $P\_bin = 0$.

Step 2 is reading the following time *P* information. At that moment, go back to Step 1.

Step 3 is defining $P\_bin = 1$. Then return to Step 1.

The data was obtained by using binarization, as shown in Figures 10 to 14 below:

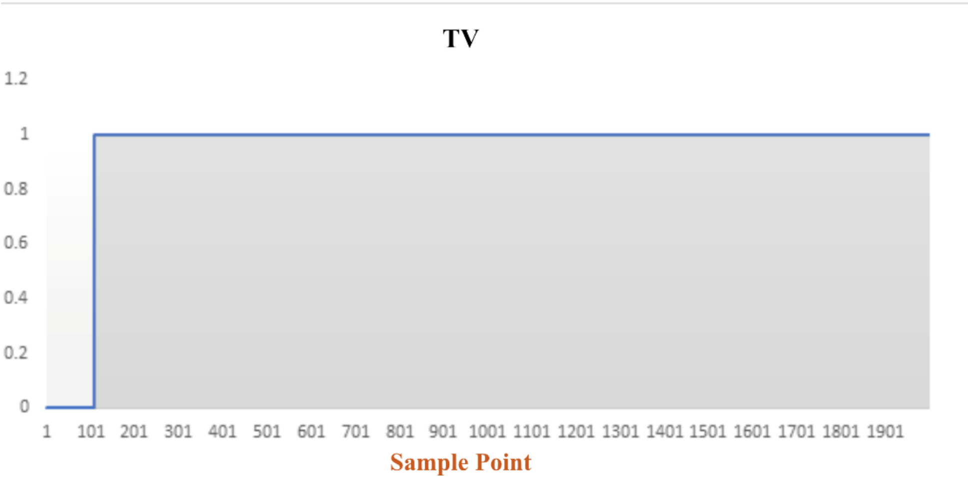

Fig. 10. Binarization of TV active power data.

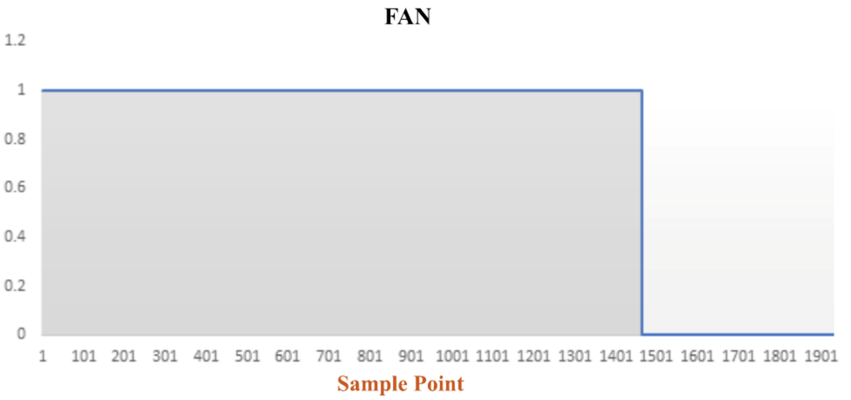

Fig. 11. Binarization of FAN active power data.

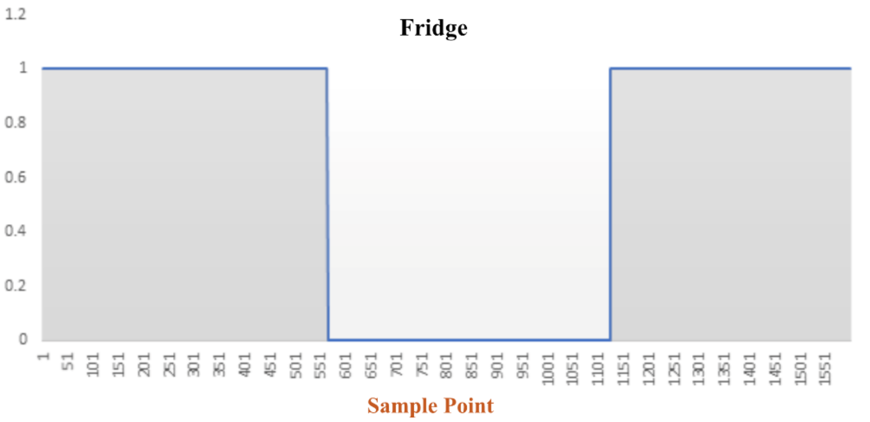

Fig. 12. Binarization of FGE active power data.

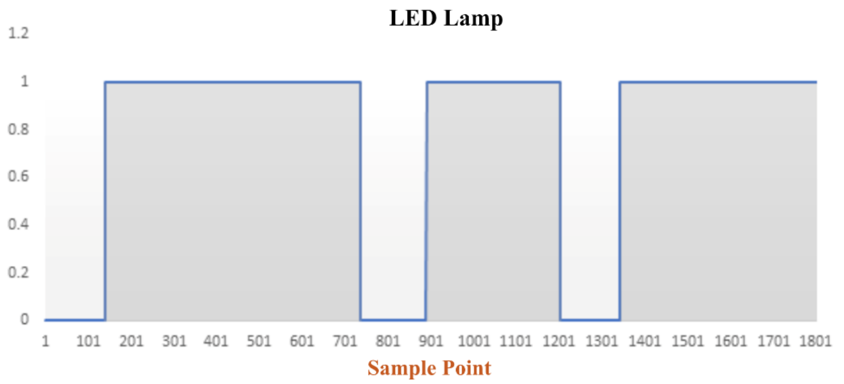

Fig. 13. Binarization of LED active power data.

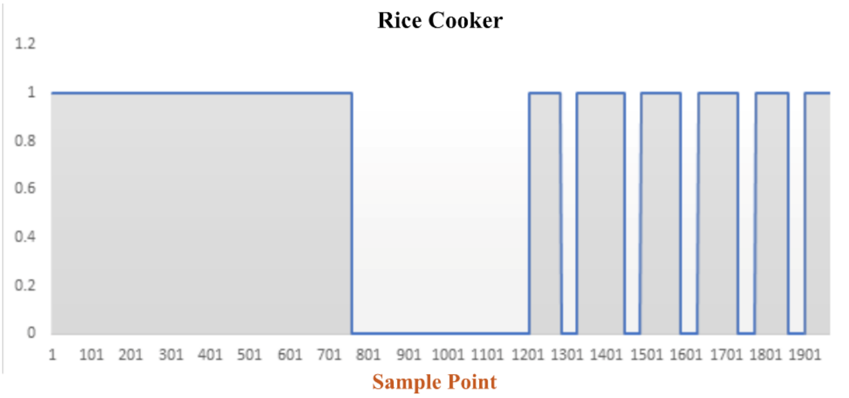

Fig. 14. Binarization of RIC active power data.

4.5 Event Detection

Event detection is a step to detect a change in the value determined as a reference for further NILM processing. This study adopted the value of active power that had been processed by binarization P_bin as an event. Thus, the change in the event was defined as a change in the value of P_bin.

$$\Delta P\_bin_t = P\_bin_t - P\_bin_{t-1} \tag{4}$$

The means of the event detection algorithm are as per the following.

Step 1. Ascertain the distinction P_bin between the ongoing time of active power and the aftermost time. Enter Step 3 whenever $P\_bin_t > P\_bin_{t-1}$. Else if $P\_bin_t < P\_bin_{t-1}$, enter Step 4. Then, enter Step 2.

Step 2. Read the following time information and return to Step 1.

Step 3. The ON event is detected. Then the algorithm process the ON event to disaggregate the ON load. Go back to Step 1.

Step 4. The OFF event is detected. Then the algorithm process the OFF event to disaggregate the OFF load. Return to Step 1.

4.6 Feature Extraction

Features were obtained from the ADE9153A sensor. As mentioned earlier, the ADE9153A sensor can detect many parameters of the electrical circuit. There are RMS of current and voltage; active, reactive, and apparent power; power factor, frequency, and temperature. This study employed the power feature to be processed in the bagging decision tree algorithm. The additional feature such as phase angle is also used.

4.7 Load Identification

The final result of the NILM process is knowing whether or not an electrical device is ON or OFF. The bagging decision tree algorithm was used to identify which device is ON or OFF. This study built the algorithm by implementing DFP programming in the LabVIEW 2020 software. The results were displayed through a GUI designed on the same software.

The means of the bagging decision tree algorithm for load disaggregation are as per the following.

Step 1. The event detection method decides if the load change occasion happens. If not, enter Step 2. In any case, enter Step 3.

Step 2. Read the following time information and go back to Step 1.

Step 3. Decide if the equipment wherein the event happened is purely resistive. In case it is pure resistive electrical loads, Step 4 ought to be followed. In any case, go to Step 5.

Step 4. Correlated with the power database of purely resistive loads. Only the active power needs to be correlated.

Step 5. Yield the loads with the most related active power as the identification result.

Step 6. Compared with non-purely resistive electrical loads utilizing the other feature.

Step 7. Determine the output of several features.

Step 8. Yield the loads with the most equipment result in Step 6 as the disaggregation result.

5. RESULTS

5.1 GUI

This study resulted in a NILM technique that can be implemented at monitoring stations in the electricity of remote areas. Therefore, in building the GUI, this study focused on developing NILM and monitoring electrical parameters. By employing LabVIEW software, GUI can be created.

GUI contained three basic windows. The first window is the information about monitoring status, as shown in Figure 15. RMS of current and voltage; active, reactive, and apparent power; power factor, frequency, and temperature were displayed.

The second window is the information about PQ-map, as displayed in Figure 16. By displaying the PQ-map, the different loads are at different points on the graph. Thus, load disaggregation can be interpreted manually using PQ-map.

The third window is the information about load identification. In this window, we can focus on which appliance is ON or OFF as the output of the bagging decision tree algorithm. An ON appliance is represented by a flashing light under the corresponding appliances icon, as illustrated in Figure 17.

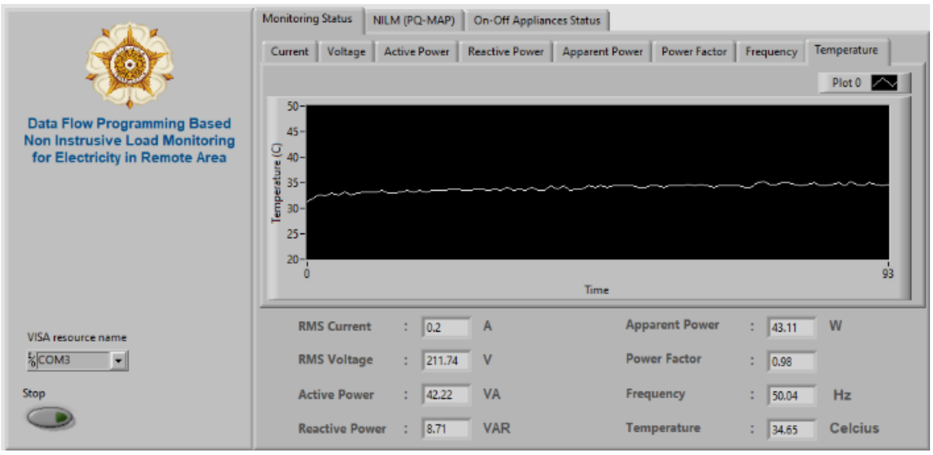

Fig. 15. GUI (monitoring status).

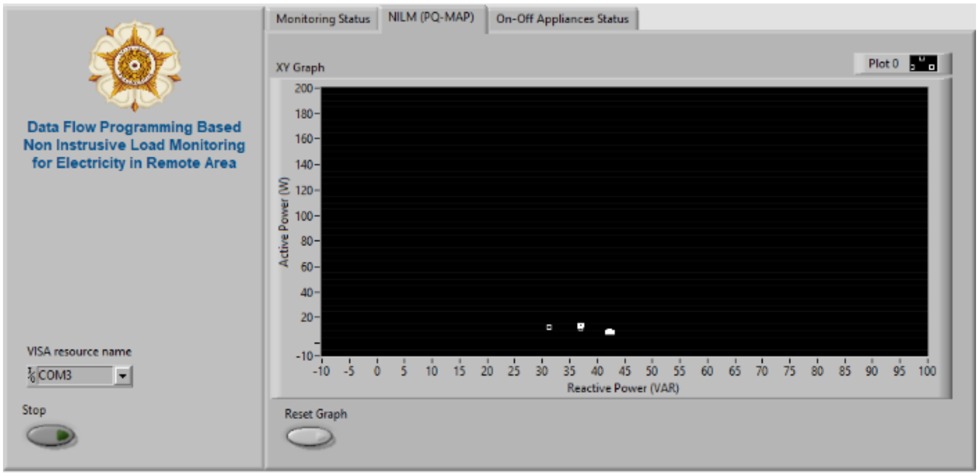

Fig. 16. GUI (PQ-map).

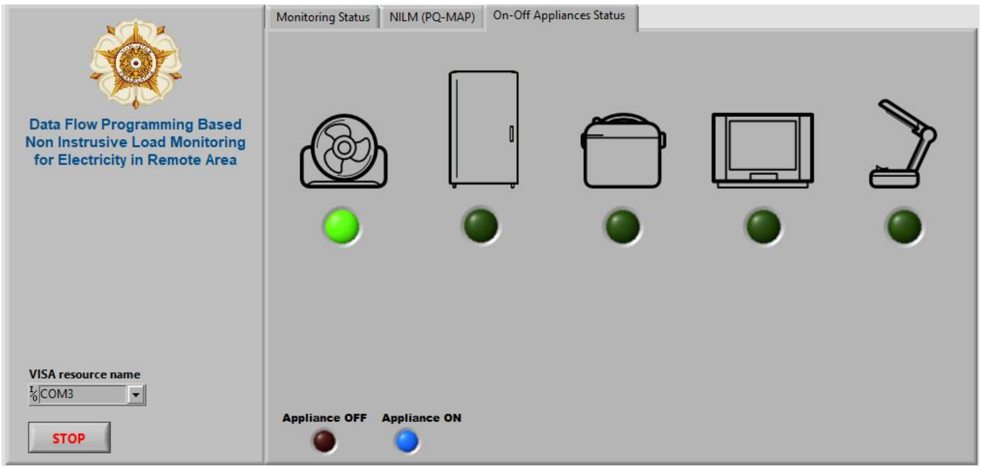

Fig. 17. GUI (ON-OFF appliances).

5.2 NILM Performance

A confusion matrix (CM) based on similar literature in [12] was applied to assess our proposed NILM performance. CM is a table to describe classification performance. There are two types of CM, namely binary-class and multi-class, as shown in Figure 18 and Figure 19. Multi-class CM was performed in this study to determine NILM performance by calculating accuracy, precision, recall, and F1-score.

Accuracy is the level of the number of tests that predict the outcome effectively. The calculation implements the following equation.

Binary class		Predicted class: Appliance	Predicted class: Non-appliance	
Actual class	Appliance	True Positives (TP)	False Negatives (FN)	**Recall**/Sensitivity Rate $\frac{TP}{TP+FN}$
Actual class	Non-appliance	False Positives (FP)	True Negatives (TN)	Specificity Rate $\frac{TN}{TN+FP}$
		Precision/Positive Predictive Value $\frac{TP}{TP+FP}$	Negative Predictive Value $\frac{TN}{TN+FN}$	**Accuracy** $\frac{TP+TN}{TP+TN+FP+FN}$

Fig. 18. The binary class CM of load identification

Multi-class		Predicted class: Appliance 1	Appliance 2	. . .	Appliance N	Marginal sum of predictions
Actual class	Appliance 1	f_{11} **TP**	f_{12}	. . .	f_{1N} **FN**	$\sum_{j=1}^{N} f_{1j}$
	Appliance 2	f_{21}	f_{22}	. . .	f_{2N}	$\sum_{j=1}^{N} f_{2j}$
	. . .	. . .	. . .	. . .	. . .	. . .
	Appliance N	f_{N1} **FP**	f_{N2}	. . .	f_{NN} **TN**	$\sum_{j=1}^{N} f_{Nj}$
	Marginal sum of predictions	$\sum_{i=1}^{N} f_{i1}$	$\sum_{i=1}^{N} f_{i2}$	. . .	$\sum_{i=1}^{N} f_{iN}$	$T = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij}$

Fig. 19. The multi-class CM of load identification.

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

The probability that all sample tests predicted to be positive have a genuinely positive value is called accuracy. The following equation calculates precision.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

The probability that all positive samples are predicted to be positive samples is called recall. The following equation calculates recall.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

Precision and recall calculations are generally conflicting. In case one is higher, the other is mostly lower. The most well-known strategy for consolidating the two indices is to utilize the F1-Score. F1-Score is a weighted harmonic average of precision and recall. The following equation calculates F1-Score.

$$F1 - Score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{8}$$

The authors tested the method in the lab by turning ON and OFF on 30 trials for TV and 50 trials for FAN, RIC, and LMP. For FGD, the authors connected the sensor for 10 hours and got ten events. Figure 20 and 21 shows the experiment. The result is the CM, as shown in Figure 22. By taking Equations 5 to 8, the accuracy, precision, recall, and F1-score can be determined.

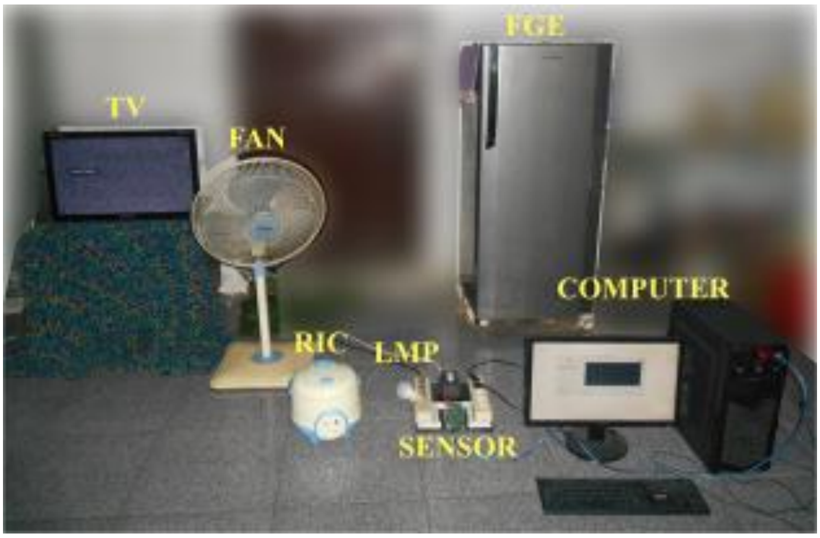

Fig. 20. NILM performance test.

Fig. 21. ADE9153A sensor.

	TV	FAN	FGE	LMP	RIC
TV	30				
FAN		49			1
FGE			10		
LMP				43	
RIC		6			44

Fig. 22. CM of NILM performances.

The way to determine the accuracy of the method based on Figure 22 following equation 5 is as follow:

$$\text{Accuracy} = \frac{30+49+10+43+44}{30+49+10+43+44+6+1} = 0.961749$$

To determine the precisions of this method, the authors need to determine the precision of each load as follow:

$$\text{Precision of TV} = \frac{30}{30} = 1$$

$$\text{Precision of FAN} = \frac{49}{55} = 0.8909$$

$$\text{Precision of FGE} = \frac{10}{10} = 1$$

$$\text{Precision of LMP} = \frac{43}{43} = 1$$

$$\text{Precision of RIC} = \frac{44}{45} = 0.9778$$

$$\text{Precision} = \frac{1.0+0.8909+1.0+1.0+0.9778}{5} = 0.9737$$

The authors need to determine the recall of each load to determine the recall of this method as follow:

$$\text{Recall of TV} = \frac{30}{30} = 1$$

$$\text{Recall of FAN} = \frac{49}{50} = 0.98$$

$$\text{Recall of FGE} = \frac{10}{10} = 1$$

$$\text{Recall of LMP} = \frac{43}{43} = 1$$

$$\text{Recall of RIC} = \frac{44}{50} = 0.88$$

$$\text{Recall} = \frac{1.0+0.98+1.0+1.0+0.88}{5} = 0.972$$

Finally, to determine the F1-score, the authors used equation 8 and resulted as follow:

$$\text{F1-score} = \frac{2\times 0.9737\times 0.972}{0.9737+0.972} = 0.9728$$

6. DISCUSSION

The proposed NILM serves as a handy tool to disaggregate the types of electrical loads. As a DMS/Demand Response tool, NILM can be employed in managing non-manageable renewable energy in some scenarios. In electricity in remote areas, certain types of electrical loads may not be permitted to turn ON at a particular time due to the capacity shortage of the grid. For example, a light that is ON during the day or an air conditioner (AC) ON in winter. Here, the NILM plays a significant role in helping the electricity operator know which appliances are ON or OFF. Then, the operator can follow up the case by employing many ways to achieve electrical efficiency in the grid. Therefore, NILM can balance small and weak microgrids in high renewable energy penetration scenarios.

This study brought the DFP as the programming method in building the algorithm. Based on the references, DFP is easier to understand programming methods and then easier to modify than other programming methods. The proper programming method used in this study is essential in remote areas' electricity considering that remote areas lack human resources equipped with programming skills.

The proposed method yielded a good NILM performance. However, there are limitations to this method. This method was tested at a steady-state voltage at a voltage source of 220 V 50 Hz. So that in electrical applications in remote areas, it is possible for instability that is not taken into account in this study. The implementation of energy storage components (such as batteries) is not considered in this study which may also affect the stability of the electric power system in the remote area.

In the calculation of NILM performances, the results of this study cannot be compared to the other research due to the different appliances employed in this study. This study does not apply any dataset to the proposed method. On the other hand, the authors use five different loads considering the more monitored loads, the lower NILM performance. It is because of the greater probability of the similarity of the load parameter.

Then, the CM does not consider the appliance which NILM should predict, but it does not. This issue can be shown by the value of samples on LMP. It should have been 50, but only 43 were read. It may be due to the LMP being a load that has low power. Finally, the error occurs in FAN and RIC because these appliances have the same P parameter. However, these appliances have different parameters of Q and S, which make these appliances still disaggregated.

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

The use of the NILM technique in remote area electricity was proposed in this study. The bagging technique is applied in the classification step by implementing the decision tree algorithm. By utilizing data flow programming in the LabVIEW, the algorithm and GUI can be built. The authors conducted this experiment by employing LabVIEW 2020 on an Intel i5 2400-3.1 GHz CPU, 16 GB RAM, and 64-bit operating system computer. This study resulted in the NILM process, which has a good performance accuracy of 0.9617, precision 0.9737, recall 0.9720, and F1-score 0.9728.

Not all electricity has a monitoring station, so the proposed NILM cannot apply to all remote areas. Furthermore, using devices with similar parameters, devices with low power, and simultaneously active devices is still a challenge in NILM. In addition, network stability should also be considered because it can affect the NILM system. Finally, the NILM technique can improve the efficiency of electrical energy in remote areas. By knowing which device is ON or OFF, the operator can determine the priority of the load or electrical device that should be ON or OFF at a particular time as needed.

7.2 Future Work

In future work, the authors consider developing this study in more depth and adjusting the parameters of electricity generation in remote areas precisely. The study will continue to use the NILM for balancing energy production and consumption in electrical systems.

ACKNOWLEDGEMENT

This study has received funding from Indonesia Endowment Fund for Education (LPDP). The authors thank LPDP for the support.

REFERENCES

[1] Embassy of The Republic of Indonesia. 2017. *Facts & Figure of Indonesia*. An online article from the World Wide Web: https://www.embassyofindonesia.org/index.php/basic-facts/.

[2] Samudra K., Mulyono S.B., Wisudo S.H., and Iskandar B.H. 2017. Marine tourism potential in kapoposang island, pangkep regency. *Journal of Marine Fisheries Technology and Management* 1(2): 87-89.

[3] Liu H., Wu H., and Yu C., 2019. A hybrid model for appliance classification based on time series features. *Energy and Building* 196: 112–123.

[4] Aladesanmi E.J. and K.A. Folly. 2015. Overview of non-intrusive load monitoring and identification techniques. *IFAC-PapersOnLine* 48-30(2015):415-420.

[5] Liu H., 2019; Non-intrusive load monitoring: Theory, technologies and applications, first edition, chapter 1, page 3-7; Science Press, Beijing, China.

[6] Sousa T.B., 2012. Data-flow programming: concept, languages and applications. In *7th Doctoral Symposium on Informatics Engineering* 7: 1-13.

[7] Altendeitering M. and S. Schimmler. 2020. Data-flow programming for smart homes and other smart spaces. In *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing.* 2020-August: 16–19.

[8] Yadav A., Sinha A., Saidi A., Trinkl C., and Zörner W. 2020. NILM based energy disaggregation algorithm for dairy farms. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring.* The virtual event, Japan, 18 November. Association for Computing Machinery.

[9] Yu Q., Liu Y., Jiang Z., Xu C., Yang R., and Zhang R. 2020. Research of load behavior based on NILM in domestic microgrid. In *International Conference on Signal Processing Proceedings, ICSP*. 2020-December: 337–340.

[10] Yu Q., Jiang Z., Yang D., Liu Y., Long G., and Zhang T. 2020. Research of NILM in offshore oil platform power system. In *Proceedings of 2020 IEEE International Conference on Power, Intelligent Computing and Systems, ICPICS 2020*. 28-10 July. Shenyang: IEEE.

[11] Hosseini S.S., Agbossou K., Kelouwani S., and Cardenas A. 2017. Non-intrusive load monitoring through home energy management systems: A comprehensive review. *Journal of Renewable and Sustainable Energy Reviews* 79: 1266-1274.

[12] Gopinath R., Kumar M., Joshua C.P.C., and Srinivas K., 2020. Energy management using non-intrusive load monitoring techniques – State-of-the-art and future research directions. *Journal of Sustainable Cities and Society* 62: 102411.

[13] He K., Stankovic V., and Stankovic L. 2020. Building a graph signal processing model using dynamic time warping for load disaggregation. *Journal of Sensors (Switzerland)* 20(22): 1–16.

[14] Zhao B., He K., Stankovic L., and Stankovic V., 2018. Improving event-based non-intrusive load monitoring using graph signal processing. *IEEE Access* 6: 53944-53959.

[15] He K., Stankovic L., Liao J., and Stankovic V., 2018. Non-intrusive load disaggregation using graph signal processing. *IEEE Transactions on Smart Grid* 9(3): 1739-1747.

[16] Le T.T.H., Kang H., and Kim H., 2020. Household appliance classification using lower odd-numbered harmonics and the bagging decision tree. *IEEE Access* 8: 55937–55952.

[17] Eskander M.M. and C.A. Silva. 2021. A complementary unsupervised load disaggregation method for residential loads at very low sampling rate data. *Journal of Sustainable Energy Technologies and Assessments* 43 (2021): 100921.

[18] Ahajjam M.A., Licea D.B., Essayeh C., Ghogho M., and Kobbane A., 2020. MORED: A Moroccan buildings' electricity consumption dataset. *Energies* 13 (24): 6737.