DEEP LEARNING APPROACH FOR ELECTRIC APPLIANCES RECOGNITION

Frans Juanda Simanjuntak

Abstract— Monitoring energy consumption of such appliances is really essential for the welfare of human being. The impact is not merely to inform the consumers how much energy they consume but also makes them aware of their electric consumption profiles. In this paper, we report the development of a system that is able to recognize appliances based on their electric consumption behavior. The goal of this study is to encourage consumers to apply rules to control the usage of their appliances in order to reduce cost and to save energy. In this study, we use smart meter device called SMAPPEE to measure the electric consumption at low frequency. We also apply deep learning approach called Recurrent Neural Network (RNN) to perform classification of appliances based on their consumption behavior. The result of the experiment shows that the deep learning approach used in this study leads to promising performance of appliance recognition.

Index Terms—Monitoring energy consumption, appliances, recognition, SMAPPEE, deep learning, RNN

1 INTRODUCTION

Monitoring energy consumption is essential for the consumers. It helps them to be aware of how much energy they consume that resulting in financial saving. For example, consumers would be able to determine the time of use of each appliance and set a rule to switch on all charging appliances during night time when the electricity is cheaper [11]. Moreover, monitoring energy consumption does not merely help consumers to reduce the cost but also it is seen as a solution to the problem of reducing their environmental impacts such as greenhouse gas emission.

Many approaches have been implemented to monitor energy consumption such as providing a continuous information feedback and fine-tuned automated management of the appliances. The outcome of these approaches allow the reduction of the energy bill of 15 % up to 30 % [4]. However, these are still hard to configure and very expensive.

Another approach to monitor the energy consumption is by using smart meter such as SMAPPEE. This device is able to capture how much energy have been consumed by our appliances. Not only does it provide a very basic consumption parameter as another smart meters does (normally only the voltage) to be used but the parameters such as reactive power, apparent power, active power, current, and cos-phi are also available. This information is way much better when it comes to monitoring appliances. Beside, monitoring energy consumption with SMAPPEE is considered to be less costly and easy to configure.

The benefits offered by smart meter like SMAPPEE motivates us to develop a system that is able to monitor the energy consumption of appliances. However, before developing a system that is able to provide the detail of the energy consumption, as the preliminary step, the system must be able to recognize appliances based on their consumption behavior. This goal can be achieved with such a popular approach so-called machine learning. Machine learning is a method of data analysis that automates analytical model building. Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results [2].

This study shows machine learning approaches can be used to perform appliance recognition based on energy consumption behavior. The implementation is based on deep learning model called Recurrent Neural Network (RNN). Basically, RNN is a neural network (NN) that processes sequential data and takes in as input from both the new input at the current time step and the output (or a hidden layer) of the net in the previous time step. RNN is very popular among other deep learning models because it allows end-to-end learning. The type of RRN used in this study called Long short-term memory (LSTM). The results shows that the deep learning approach leads to a promising performance when recognizing appliances.

The rest of the report is structured as follows. Section 2 presents the related works in relation to this study. Section 3 describes the system architecture and description, while the details about data acquisition is given in section 4. In section 5, the RNN algorithm used to perform the appliance recognition is described. Finally, the results and the conclusion are drawn in section 6 and 7 respectively.

2 RELATED WORK

Several works in relation to electricity energy consumption behavior monitoring have been found. Most of the studies monitor the consumption using non-intrusive method. The existing studies reviewed NILM method in the existing system such as residential building and offices, created a disaggregation framework, and applied machine learning techniques on NILM.

An existing work done by IEEE aims at reviewing the methodology of consumer system for Non-Intrusive Appliance Load Monitoring (NIALM) in residential building [10]. The system was reviewed based on several common principles of available NIALM methods. They started analyzing the features of selected appliances which were followed by reviewing hardware installation both sensor and data acquisition system. Lastly, they analyzed the mathematical algorithm used to detects the features in the overall signal. The results of this study concludes no complete NIALM solution suitable for all types of household appliances is available, no complete set of robust widely accepted appliance features has been identified, using more mutually-independent features improves accuracy however using several orthogonal disaggregation algorithm may improve accuracy, but optimal fusion needs to be implemented, and ROC curves was highly encouraged to use rather than the ambiguous accuracy metrics.

After reviewing NIALM methods, they studied the basic concept of load signatures, structure and methodology of applying mathematical programming techniques, pattern recognition tools, and committee decision mechanism to perform load disaggregation [8] as a foundation to create a load disaggregation framework. The disaggregation framework used three disaggregation algorithm called committee decision making mechanisms (CDMs) which aims at performing load disaggregation at the meeting level and three random switching simulators to investigate the performance of different CDMs under a variety of scenarios [9]. In order to simulate the capability of disaggregation framework, the database from common household appliances were used and the output

• *Frans Simanjuntak is a master student specialization in Software Engineering and Distributed System at the University of Groningen. E-mail: f.simanjuntak@student.rug.nl.*

shows that all CDMs outperformed other single-feature and single-algoritm disaggregation methods in term of reliability and robustness.

Another study introduces the concept of an Office Plug Load Dataset (OPLD) which was brought by Balaji Kalluri et al [6]. Basically, OPLD was used to predict miscellaneous electrical plug loads (MELs). They attempted to characterize office desktop appliances across multiple states through a very large experimental dataset. The outcome of the study concludes that the OPLD has some lacunae especially when it was applied to office appliances. Currently, the OPLD dataset is not available for public yet since the study is still in progress.

Other several studies were focusing on recognizing the pattern of the energy consumption using deep learning. In the paper "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation", Jack Kelly and William Knottenbelt adapted three deep neural network architectures to energy disaggregation [7]. First, they used a form of neural network called long short-term memory (LSTM). Second, they framed the energy disaggregation as a denoising task. Third, they regressed the start time, end time, and average power demand of each appliance activation. In their experiment, they used seven metrics in order to test the performance of those three algorithm on real aggregate power data from five appliances. Tests were performed when the houses were seen during the training and not. The result of suggests LSTM works best for two state appliances but it does not perform well when applied on multi-state appliances such as the dish water and vending machine.

A somewhat similar architecture to the previous study was done Pedro Paulo Maques do Nascimento [5]. He applied different models of deep learning based on neural network called convolution neural network (CNN), recurrent convolution neural network (RCNN), long short-term memory (LSTM), gated recurrent unit (GRU), and residual neural network to infer the appliances being used and their individual consumption. The appliances used in this experiment are microwave, dishwasher, and refrigerator from five houses. Appliances in house 1 and house 3 were used for training and validation while the rest were for testing. In this experiment, for each appliance and house, 80% of the collected data was used for training the networks and the last 20% for validation.

After the networks trained using five different models, the results show that all the models were capable of obtaining good results. In detail, GRU was the best architecture overall performance and the LSTM was the one with the worst performance. On the other hand, the residual architecture did not perform very well probably due to the number of layers used.

3 SYSTEM DESCRIPTION

In this section, the description of the architecture of the system and the appliance recognition module are presented. The details of the appliance recognition algorithm are elaborated in section 5.

3.1 System Architecture

In order to record the consumption of electricity, the system relies on a sensor called SMAPPEE which is installed in the office room 564 Bernoulliborg building, University of Groningen. SMAPPEE is a sensor that detects all of the essential appliances in our home and tells us exactly how much they consume [1]. Not only is it able to monitor the energy consumption from the households but also it support the solar energy and gas and water. This sensor is quite easy to install because we just need to clip-on the SMAPPEE sensor to our fuse box or wire connected to the main socket. There are six parameters of energy consumption captured by SMAPPEE sensor: reactive power, apparent power, active power, cos-phi, current, and voltage.

Since SMAPPEE has several types, therefore we decided to use SMAPPE for household appliances (See Figure 1) in order to make the study in line with our goal to tracks the overall energy consumption in the office as well as the individual use of our essential appliances.

The architecture of the system is illustrated in Fig 2. It can be seen from the figure that the SMAPPEE sensor will capture the energy consumption of device 1 (CPU), device 2 (monitor iiyama), and device 3 (monitor philips). The details of the SMAPPEE architecture can be seen in Fig 3. From this figure, one can recognize that the source of the electric current is coming from socket 1 which is consumed by socket 2. SMAPPE sensor and socket 3 are plugged in socket 2 so the sensor can be turned on along with other devices connected to socket 3. Between socket 2 and socket 3, the wire is clipped with the SMAPPEE clipper so the sensor can measure how much energy is being consumed by the appliances.

Fig. 1. SMAPPEE sensor for household appliances [1].

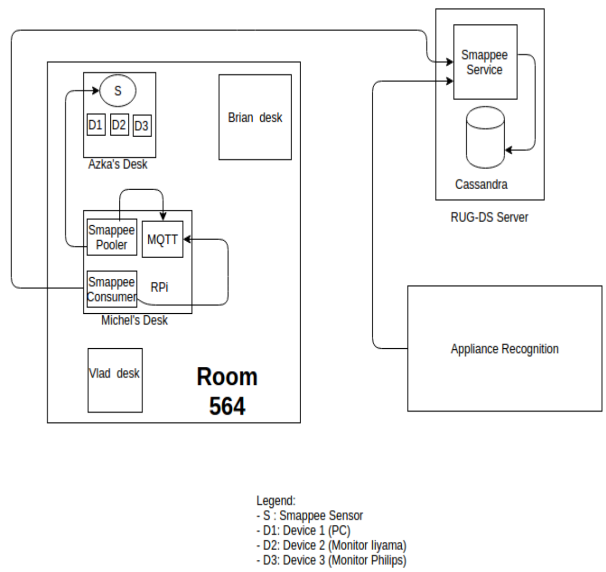

Fig. 2. The architecture of the system.

After collecting the raw data, then the Smappee Pooler which is running on Raspberry Pi3 will then connect to SMAPPEE sensor to collect them. Moreover, the pooler will put the raw data into MQTT queue to be processed by Smappee Consumer.

On the other hand, the Smappee Consumer is waiting for any message coming to the queue. Once it receives the message, the consumer will transform the message into JSON format and send it to smappee service (API) which is deployed on RUG-DS server. Then, the API will store the sensor data into Cassandra database so the appliance recognition module can consume it for further processing.

Furthermore, the appliance recognition module will connect directly to smappee service to obtain the raw data. This application was developed in order to recognize the type of the appliances based on the data taken from the SMAPPEE sensor.

3.2 Appliance Recognition Module

The appliance recognition module is aimed at recognizing the type of appliances, in this case CPU or monitors. This module is based on supervised machine learning technique and it relies on the data of

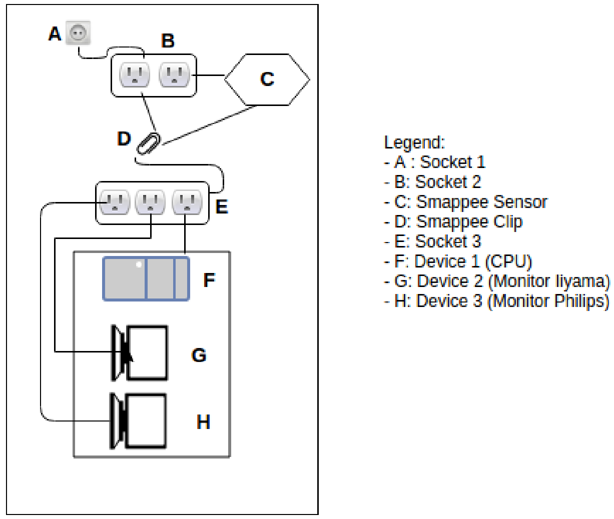

Fig. 3. The deployment of SMAPPEE sensor.

energy consumption of those appliances to create such models. The overall set of data used to create models is called training data and once the training is performed, the model can predict and classify the most probable appliance category from unseen data. This module consists of two sub modules: preprocessing module and classifier module. Preprocessing module performs the cleaning and transformation of raw data into a proper form while classifier module aims at performing training, validation, and testing. The details of this module is given in section 5.

4 DATA ACQUISITION

The sampling and parameters as well as the dataset were described in the following subsections.

4.1 Sampling and Parameters

Collecting data from SMAPPEE sensor is such an essential part because the data will be fed into appliances recognition algorithm for training, validation, and testing. In this study, we applied medium sampling rate in which data from each appliance, the pairwise combination of appliances, the combination of all appliances, and also when all appliances go off are sampled every five seconds for an hour. So in total the number of data points per hour of each class is 720.

Since the number of classes which are required for training is eight, therefore we need to acquire samples for all classes (see Table 1). The total number of 5760 data points that is obtained from the SMAPPEE sensor are divided into two parts. 6000 data points of each appliance are used for training and the remaining 1200 data points are used for validation. For testing, we collected unseen data separately by performing unsupervised monitoring for the total duration of six hours. That said, 4320 data points were obtained.

Table 1. Class of appliances.

Appliance Id	Label/Class	Description
0	0	No appliances
1	3	CPU
2	4	Monitor Iiyama
3	5	Monitor Philips
4	34	CPU and Monitor Iiyama
5	35	CPU and Monitor Philips
6	45	Monitor Iiyama and Monitor Philips
7	345	CPU, Monitor Iiyama, and Monitor Philips

Based on our analysis, applying medium sampling rate for collecting electricity energy consumption data is such a good approach because the sampling rate itself is in the middle of low and high. We thought that the low sampling rate is not a better approach in this study because it has drawbacks, for instance: one sample per second or less, is the consumption parameters would be indistinguishable. Conversely, if we apply high sampling rate for example above ten minutes, this approach is also inefficient for this study because the number of data points that we can obtain in one hour is roughly between 60 and 100. In fact, less data points will lead to overfitting.

There are six available energy consumption parameters provided by SMAPPEE: active power, reactive power, apparent power, current, cos-phi, and voltage. Active power is the power that is used to do work on the load, reactive power is the power not used to do work on the load, apparent power is the power supplied to the circuit, current is the rate at which charge is flowing, cos-phi is a measure of the systems electrical efficiency in an alternating current circuit, and voltage is the difference in charge between two points. Figure 4 shows an example of the electricity consumption measure for an appliance of class ”CPU” (lime: reactive power , red: apparent power, green: active power, magenta: cos-phi, black: current, blue: voltage). We stored all these parameters in our database for further processing.

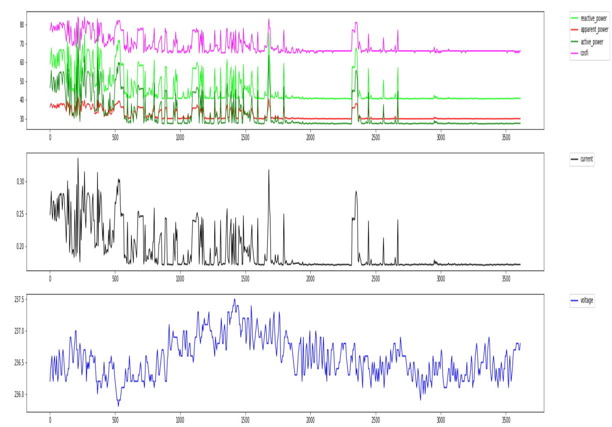

Fig. 4. Electricity consumption of appliance class CPU.

4.2 Dataset

A JSON data structure has been designed for storing the raw information and also its ground truth. The format consists of the following parts: the typeId describes the sensor or ground truth type, instanceId describes the sensor id or ground truth id, and the timestamp stores the date and time when the sensor acquired the data or the date time of ground truth. The values part stores values and attributes of each instance. For sensor data, this part contains the raw information such as reactive power, apparent power, active power, current, cos-phi, and voltage and also the value. However for ground truth data the value part only store the id of device and its status, on or off.

5 APPLIANCE RECOGNITION ALGORITHM

The algorithm used for the appliance classification were elaborated and described in the following subsections.

5.1 Preprocessing

We realize that the dataset acquired by SMAPPEE sensor does not contain the appliance id because the sensor did not know which appliances were being measured. To cope with this missing value, one solution that is possible to apply is to combine the raw data from sensor with its ground truth. As mentioned earlier, another android based application was developed in order to set the ground truth of the appliances. When the appliance(s) is on, we must set the ground truth manually via application and the same goes when the appliance(s) is off. This application is able to set the status of both appliances single and multiple appliances. Since we have 8 classes therefore the id of the appliances is ranging from 0 to 7 as listed in Table 1.

5.2 Feature Generation and Extraction

Feature generation and extraction are the steps that need to be performed after preprocessing. This process involves the action of acquiring raw data and defining features (i.e. variables) for potential use in our analysis. In this step, we define some important features to be included in training phase.

in the beginning, we included all features gathered from SMAPPEE sensor as input parameters such as active power, reactive power, apparent power, current, cos-phi, voltage and device id as target. However, after analyzing the value of those features, the feature so-called ”voltage” was not able to distinguish unique values on each appliance because they are almost similar for all appliances. If we include this as training features in our algorithm, the performance of the model will be decreasing that is why we decided to take this feature out. As a final dataset for training, we convert all considered features into CSV format.

5.3 Splitting

After the final dataset is converted into CSV, then we split them into two disjoint sets. From the total of 5760 data points (720 * 8 appliances), 83 percent (4800 data points) was used for training and 17 percent (960 data points) for validation. The data points used for training and validation were chosen in sequent mode for each appliance. The index starts from 1 to 600 were used for training and the rest for validation.

The reasons why we set the percentage of training and validation to 83:17 because each appliance has to meet the dimension of 60x5 (row: 60 (data point), column: 5 (input parameter)) for each iteration. The same pattern goes for validation, therefore, the total number of batch of one appliance to train non-overlapped window is 10 (Total of 600 data point divided by 60 data points per batch) and to train overlapped window is 20 (Total batch of non-overlapped + overlapped window, see section 5.4). Moreover,the total number of batch to validate the model is 2 (Total number of 120 data points divided by 60 data points per batch).

5.4 Training and Classification

Training and classification are the most essential part of the whole study because it shows us whether it is possible to detect appliances based on consumption behavior. In the experiment, we use one of deep learning method called RNN based on LSTM approach. The details of the algorithm are described in the following sections.

5.4.1 RNN and LSTM

Recurrent Neural Network is a kind of neural network that the output at some time instant depends on the network’s past state, hence some connections form a directed cycle (different from the feedforward neural networks). With this kind of configuration, the network can exhibit a temporal behavior. The network also creates an internal memory gaining the ability to process sequences of inputs [5].

On the other hand, long short-term memory (LSTM) is a type of recurrent neural networks. This architecture was designed to solve the vanishing gradient problem, that is common in vanilla RNNs. It uses gates to have a better control of the gradient flow [5].

In this study, we wrote an RNN algorithm based on LSTM approach. The program is written in python and it uses a library called Tensor Flow from Google [3]. The parameters used for this algorithm as follows:

- Matrix 60x5 as input.
- Matrix 1x8 as output (The appliance id was converted into binary of 8 digits).
- The total number of neuron is 8.
- The learning rate is 0.001.
- The optimizer used in RNN is Adam Optimizer.

Based on the above parameters, we train our model using overlapped and non-overlapped window intervals approaches. The details of both approaches are described in the following subsections.

5.4.2 Overlapped Window Length Selection

The window length selection is the input size that fed into neural networks and it is defined by the number of samples, which is in fact were sampled each five seconds. The overlapped window length selection depicts that the starting point of the window is initiated at minute 0 and then it moves until it reaches minute 5. After that, the next starting point moves 2 minutes from the previous point with the total length of 5 minutes. The window is iterated over and over again until it reaches the last point. The total number of batch to train the entire overlapped window is 167 as shown in Fig 5.

The challenge when using this approach to train such model is the fact that how to determine if one batch contains multiple appliance id. In order to overcome this issue, we decided to select the most frequent appliance id occurs in the batch as target label. For instance, if in one batch, 45 data points are labeled as appliance id 1 and the rest 215 data points as appliance id 2, based on overlapped window length selection approach, this batch will be labeled as appliance id 1 because 75 % of the data points are appliance id 1.

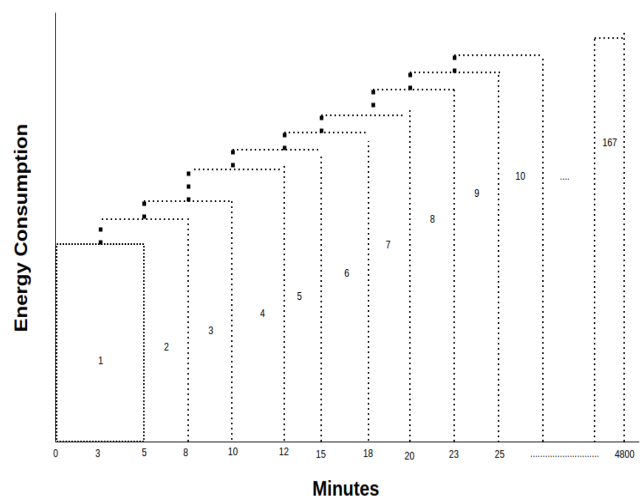

Fig. 5. Overlapped window selection.

5.4.3 Non-Overlapped Window Length Selection

Not only was the overlapped window trained but also we trained non-overlapped one. This approach is quite straight forward. First, the algorithm will divide the datasets into batches with interval time per 5 minutes. Since there is no overlapped appliance id in one batch, therefore one batch definitely contains one appliance id which is used as target label. After training one batch is finished, the starting point will move 5 minutes from the previous one. This process will iterate continuously until it reaches the maximum iteration (See Fig 6 for more details). The maximum iteration for the entire batch is 80 which is derived from the total number of the entire data points (4800) divided by the number of data points in one batch (60).

6 RESULTS

For training, 10 batches of each class (60 data points per batch) were used to train the model and the remaining instances were used for validation. The RNN algorithm was used to train the model for both

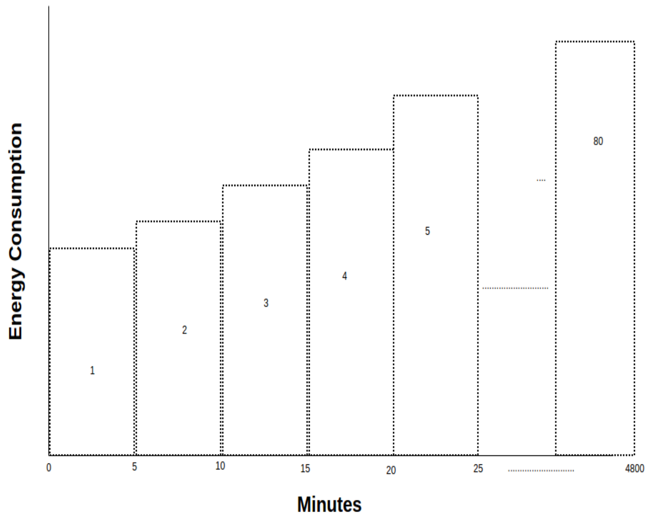

Fig. 6. Non-overlapped window selection.

overlapped and non-overlapped window selection. The validation results show that the RNN algorithm leads to a promising performance of 100 % correct recognition of the appliances on both methods (See Fig.7 and Fig.8).

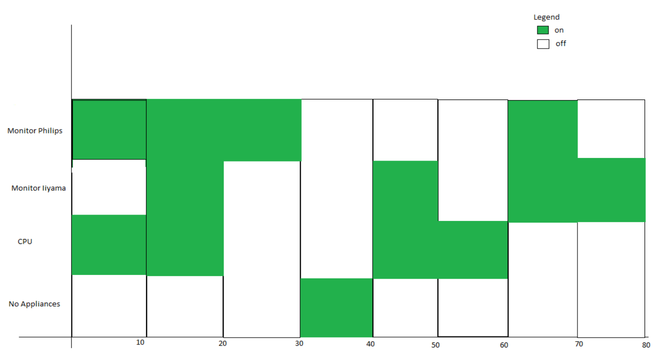

Fig. 7. Target classes of overlapped and non-overlapped validation.

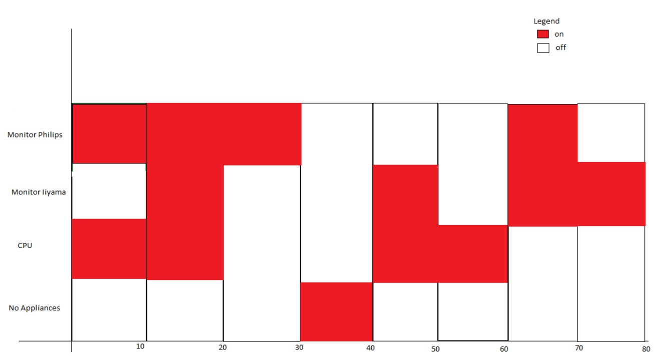

Fig. 8. Predicted classes of overlapped and non-overlapped validation.

However, the results from testing show different outcomes. The non-overlapped model is able to predict 97.22 % of the label target correctly while the overlapped model just 84.72 % as described in Fig.9, Fig.10 and Fig.11 respectively.

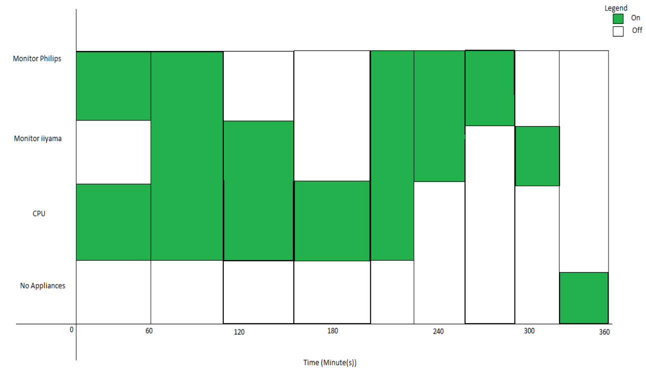

Fig. 9. Target classes of overlapped and non-overlapped testing.

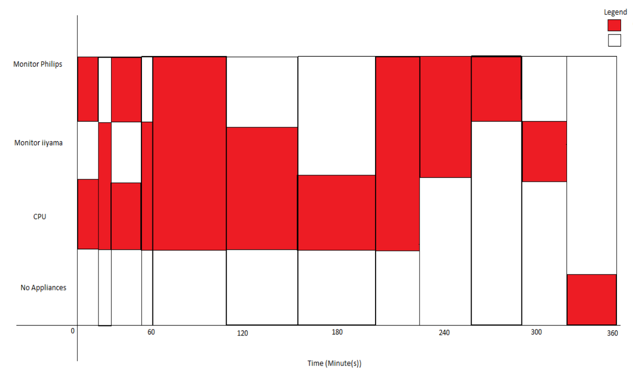

Fig. 10. Testing result of non-overlapped window selection.

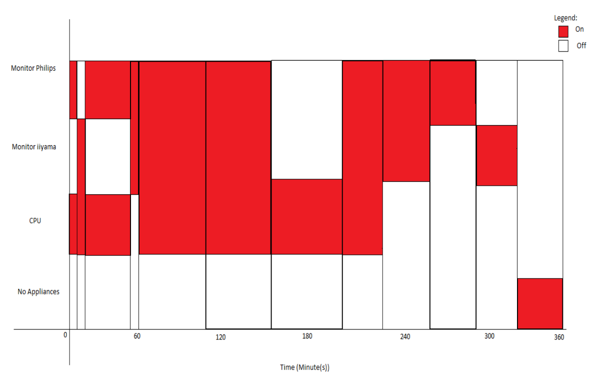

Fig. 11. Testing result of overlapped window selection.

Based on our analysis, the main success factor of the outstanding results that we obtained from non-overlapped model is the limited number of appliances we use for this study. We just rely on three appliances that make it is easier for RNN to train the models. Another success factor is the length of window interval when feeding the inputs of neural network. This interval defines the number of data points fed into network. Based on our assumption, five minutes window interval is considerably great strategy which is in fact leads to great performance when training the models. This is considered as a good approach because the mean values of all parameters of each device in the window interval will be used as inputs that helps the neural network to train the model easily. Moreover, it is also powered by the unique values of each appliance that make each appliance as unique instance.

However, the results of overlapped window selection model shows less promising performance compared with non-overlapped one. The combination of values from two different appliances that are stored in one batch has significant effect on RNN when adjusting the network. It can be seen from the results that the overlapped model miss classified appliance id 4(the combination of CPU and Monitor Iiyama) as appliance id 7 (the combination of CPU, Monitor Iiyama, and Monitor Philips).

Nevertheless, we realize that this study is just a very basic step of recognizing electric appliance using deep learning approach. It still requires further detailed experiment because it just relies on three appliances due to a short time left for the research. This study needs more appliances to be trained and more data to be collected that are going to be used for training, validation, and testing the models. Also, the window interval and input parameters can be adjusted as needed in order to create a robust model.

Based on the findings from both validation and testing results, we can draw a conclusion that the non-overlapped window selection shows promising outcomes than the non-overlapped one when recognizing appliances based on their electricity consumption behavior.

7 CONCLUSION

In this paper, it is already explained the development of a system which is able to recognize the appliances based on their electric consumption profiles measured with SMAPPEE sensor. One of deep learning approach called Recurrent Neural Network was used as algorithm to recognize the appliances. This technique works in sequential ways which is started from the pre-processing, feature selection, training model, validating model, and testing model. When training the models, we employed two different approaches namely overlapped and non-overlapped window selection. The overlapped approach trains the model based on two overlapped appliances data in one batch. It considers the appliance id that appears very often in a batch as a target class. Conversely, the non-overlapped one doesn't combine the values of different appliance id in one batch. However, it trains the model by processing similar appliance ids on each iteration of the batch.

The result of the study shows that the non-overlapped model works perfectly by giving 97.2 % of accuracy from the results of both validation and testing. However, the overlapped model shows less performance compared with the non-overlapped one. It is just able to classify almost 85 % of the appliances correctly. This model does not seem promising to be used for electricity appliances recognition since it miss-classified the whole dataset of one appliance. Training values from two different appliances in one batch as we did when training overlapped model will change the weight of neural network. Therefore we consider the overlapped window selection approach is not good for RNN.

In this study we have seen that the deep learning technique such as RNN was able to recognize the appliances based on their electric consumption profiles. This study is just the first step of the energy consumption monitoring process. Since the algorithm is able to recognize the appliances automatically, the next step so-called gathering the energy consumption of appliances can be performed. After the appliance recognition and acquiring the energy consumption done, in the future we can set some rules regarding our energy consumption which resulting in reducing cost.

As future work, we plan to add more appliances and collect more data to make our modeling more robust and accurate.

REFERENCES

[1] Energy monitor. `http://www.smappee.com/be_en/energy-monitor`. Accessed: 2017-06-21.

[2] Machine learning. `https://www.sas.com/en_us/insights/analytics/machine-learning.html`. Accessed: 2017-07-14.

[3] An open-source software library for machine intelligence. `https://www.tensorflow.org`. Accessed: 2017-06-23.

[4] S. Darby et al. The effectiveness of feedback on energy consumption. *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, 486(2006), 2006.

[5] P. P. M. do Nascimento. *APPLICATIONS OF DEEP LEARNING TECHNIQUES ON NILM*. PhD thesis, Universidade Federal do Rio de Janeiro, 2016.

[6] B. Kalluri, S. Kondepudi, K. H. Wei, T. K. Wai, and A. Kamilaris. Opld: Towards improved non-intrusive office plug load disaggregation. In *Building Efficiency and Sustainable Technologies, 2015 IEEE International Conference on*, pp. 56–61. IEEE, 2015.

[7] J. Kelly and W. Knottenbelt. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, pp. 55–64. ACM, 2015.

[8] J. Liang, S. K. Ng, G. Kendall, and J. W. Cheng. Load signature studypart i: Basic concept, structure, and methodology. *IEEE transactions on power Delivery*, 25(2):551–560, 2010.

[9] J. Liang, S. K. Ng, G. Kendall, and J. W. Cheng. Load signature studypart ii: Disaggregation framework, simulation, and applications. *IEEE Transactions on Power Delivery*, 25(2):561–569, 2010.

[10] M. Zeifman and K. Roth. Nonintrusive appliance load monitoring: Review and outlook. *IEEE transactions on Consumer Electronics*, 57(1), 2011.

[11] D. Zufferey, C. Gisler, O. A. Khaled, and J. Hennebert. Machine learning approaches for electric appliance classification. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pp. 740–745. IEEE, 2012.