Current Peak based Device Classification for NILM on a Low-Cost Embedded Platform using Extra-Trees

Aashish Kumar Jain[α], Syed Shahbaaz Ahmed[β], Prahalathan Sundaramoorthy[γ],
Raghavendran Thiruvengadam[δ], Vineeth Vijayaraghavan[ϵ]
[αβγδ]Undergraduate Student, [ϵ]Director - Research & Outreach, Solarillion Foundation, Chennai, India
Velammal Engineering College[αγ]; Rajalakshmi Engineering College[β]; Sri Venkateswara Engineering College[δ]
(aashish_jain[α]; syed.shahbaaz[β]; prahalath[γ]; raghav.thiru[δ]; vineethv[ϵ])@ieee.org

***Abstract*—Non-Intrusive Load Monitoring (NILM) is a method for disaggregation of energy consumption of individual appliances in a household. This involves the classification of individual appliances, for which a number of electrical features in combination with machine learning algorithms have been used. Extraction of most of these features is a computationally demanding task, and use of complex machine learning algorithms further adds to this complexity. Although solutions to this problem exist, they tend to be expensive and unaffordable to consumers in developing countries. This necessitates a need for an inexpensive solution capable of running on low-cost embedded platforms. In this paper, the authors implement a machine learning approach on an embedded platform to address this problem using current-based features for device classification. The model was evaluated using the Building-Level fUlly-labeled dataset for Electricity Disaggregation (BLUED) which contains electrical measurements for a household in the US for one week. The classifier was trained on Raspberry Pi 3 in about 4 seconds and classification of an event was performed in under 400 ms, validating the feasibility of the classification model on such a platform.**

***Keywords*—Machine learning, Power demand, Electricity, Training**

I. Introduction

Rapid development in technology in the last decade has led to an increase in the number of appliances used in a household. This has raised energy consumption [1], resulting in a need for monitoring the energy usage of individual appliances. Non-Intrusive Load Monitoring (NILM), introduced by Hart [2], aims to solve this problem with minimal intervention in existing electrical systems. Since then, researchers in NILM have proposed several algorithms [3] to address this problem, but due to high computational complexity, they require expensive hardware for commercial deployment which is unaffordable by consumers in developing countries. In this paper, the authors present a feasible and computationally less intensive event-classifier model that can be used to build a low-cost NILM system.

II. Related Work

Since the origin of NILM in 1980s, many NILM algorithms have been developed with the help of conspicuous patterns, signatures and various parameters such as power, harmonics etc., of different appliances which aided in modelling them. These NILM algorithms used either macroscopic features (eg. real and reactive power), which can be extracted using a low sampling hardware, or, microscopic features (eg. transient power, harmonics etc), which need a relatively high sampling hardware [4].

Traditionally, NILM approaches are classified into event based and non-event based methods [5]. The event based method performs classification and disaggregation following the detection of an event as opposed to its non-event counterpart, which takes into account every sampled data point in the aggregate signal for classification. NILM algorithms have been developed and tested by researchers on custom datasets or publicly available datasets, or a combination of both for this purpose.

Duarte et al. used continuous wavelet transform for mapping the transient signal to time frequency plane and the features are extracted for training the Support Vector Machine [6]. Anderson et al. proposed an event detection algorithm in which the prediction is done based on a score function which is the combination of four different metrics employed [7].

Patri et al. have used time series shapelets in the context of NILM in which classification is performed using a decision tree classifier based on Euclidean distance and has given appreciable results [8]. However, these results were presented only for groups of appliances rather than individual appliances. Srinivasan et al. proposed a current peak based approach for NILM, which is less computationally intensive, thus bringing NILM to a simple embedded platform [9]. But as this was tested on a custom dataset with only 5 appliances and also due to limitations posed by the algorithm with increase in number of appliances, the generalisation of this concept on a real time system becomes uncertain.

In this paper, the authors use machine learning to leverage the concept of current peaks on an embedded platform and analyse its feasibility on the much larger and real time Building-Level fUlly-labeled dataset for Electricity Disaggregation (BLUED) [10].

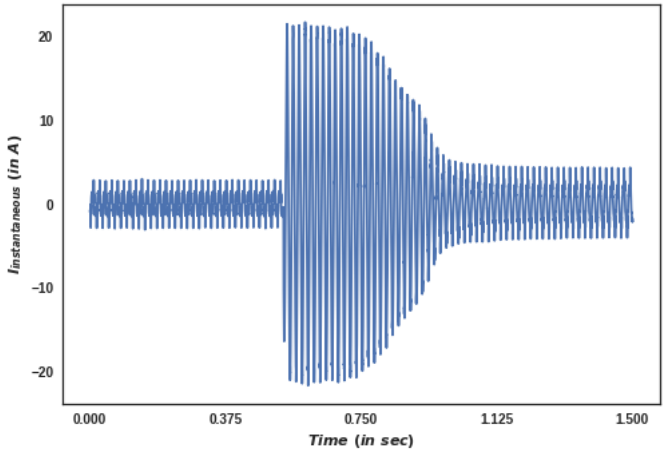

Fig. 1. Aggregate current data extracted using the 1.5 second window for an event of refrigerator (device label: 111)

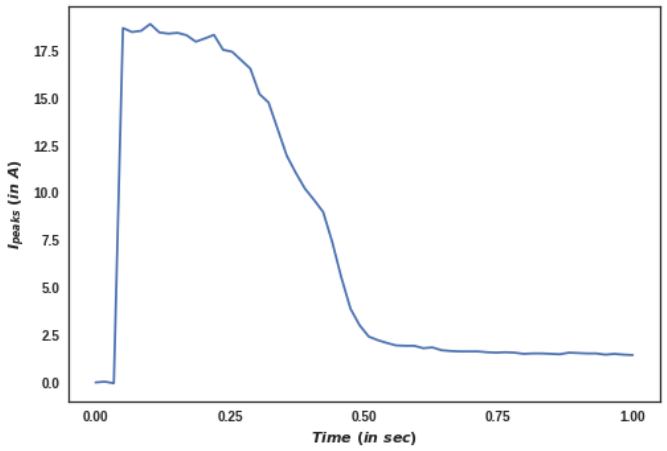

Fig. 2. Device Current Signature (DCS) obtained from the aggregate currrent data shown in Fig.1

III. Problem Statement

The increase in energy consumption and cost has spurred a need for efficient and low-cost load monitoring systems that can be made affordable to all consumers. Such a system should also be able to handle all the devices present in a typical modern household. Therefore, the authors propose an approach which is applicable to a low-cost embedded platform.

IV. Dataset

The BLUED dataset contains the aggregate voltage and current measurements sampled at a frequency (f_s) of 12 kHz for Phases A and B of a residential household in the US for a week. Around fifty appliances were monitored and each state transition in any appliance in the household is termed as an event. The dataset contains 904 events in Phase A, and 1,578 events in Phase B. Each event in both phases A and B is labeled with a timestamp and corresponding appliance number. However, the sources of certain events were unknown, and have been labeled as such. As this paper focuses on classification of devices, these unknown devices were removed. Simultaneously occurring events were also removed for the reasons mentioned in section V-B. After the removal of all these events, there were 820 events for 8 devices in Phase A, and 1436 events for 23 devices in Phase B. The timestamps and labels for these events provide the necessary ground truth for the evaluation of the proposed classification model.

TABLE I
F1-scores for varying DCS lengths (All events considered)

DCS Length	F1-score	
	Phase A	Phase B
5	0.856	0.703
4	0.857	0.694
3	0.861	0.690
2	0.866	0.692
1	0.874	0.704

TABLE II
Confusion Matrix for classification of Phase A device events; Yellow: Actual Device, Gray: Classfied Device, Green: Correct Classifications, Red: False Positives.

Device Labels	108	111	132	147	148	156	158	207
108	2	0	0	0	0	0	1	0
111	0	164	0	0	0	1	0	0
132	0	0	2	0	0	0	0	0
147	0	0	0	0	0	1	0	0
148	0	2	0	0	0	0	0	0
156	0	4	0	0	0	21	0	0
158	0	0	0	0	0	1	4	0
207	0	0	0	1	0	1	0	0

V. Feature Extraction

A. Current Peaks

The features are extracted from the aggregate current data for events considered in the dataset. The 60 Hz signal sampled at 12 kHz results in 200 data points in each current and voltage cycle. Let C be set of all the 200 data points in any given current cycle. The current peak, p, for the cycle is extracted using equation (1).

$$p = x : x \geq y \ \forall \ y \in C \tag{1}$$

B. Device Current Signature (DCS)

An event-extraction window is used, which extracts 0.5 seconds of data before the event and 1 second after. This window interval was found to be optimal, as discussed in section V-C. For the sampling frequency used, 0.5 and 1 second intervals - represented by t_a and t_b - consist of 6000 and 12000 data points respectively. Fig. 1 represents the plotted data extracted using the 1.5 second event-extraction window. Let n_s and n_t be the number of cycles in 0.5 and 1 second interval respectively. We define A as the set of all current peak values from the 0.5 second interval, as described in equation (2).

TABLE III
CONFUSION MATRIX FOR CLASSIFICATION OF PHASE B DEVICE EVENTS; YELLOW: ACTUAL DEVICE, GRAY: CLASSFIED DEVICE, GREEN: CORRECT CLASSIFICATIONS, RED: FALSE POSITIVES.

Device Labels	103	112	118	120	123	128	129	131	134	135	140	149	150	151	152	153	155	157	159	204	209	210	211
103	1	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	2	0
112	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
118	0	0	7	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
123	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
128	0	0	0	1	0	11	0	0	0	0	2	0	0	1	0	0	2	0	2	0	0	0	0
129	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
131	0	0	0	0	0	0	0	35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
134	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
135	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
140	0	0	0	1	0	0	0	0	0	0	30	1	0	0	1	0	0	0	2	1	0	0	2
149	0	0	0	0	0	1	0	0	0	0	1	6	0	1	0	0	1	0	0	0	0	0	3
150	0	0	0	0	0	1	0	0	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0
151	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	1
152	0	0	0	0	0	0	0	0	0	0	1	0	0	0	14	0	1	0	0	0	0	0	0
153	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
155	0	0	0	0	0	0	0	0	0	0	5	1	0	0	0	0	7	1	0	0	0	0	0
157	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	4	1	0	0	0	0
159	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	0	8	0	0	0	0
204	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
209	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	4
210	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	19	0
211	0	0	0	0	0	0	1	0	0	0	2	1	1	0	1	0	0	1	0	0	0	0	90

TABLE IV
STRATIFIED K-FOLD CROSS-VALIDATION SCORES FOR THE FIVE ALGORITHMS CONSIDERED

Classifier	Phase A	Phase B
Decision Trees	0.875	0.701
SVM	0.826	0.533
k-NN	0.908	0.698
Random Forests	0.929	0.778
Extra-Trees	0.946	0.801

$$A = p \in C_i \ \forall \ i \in [1, n_s]$$
$$where, \ n_s = \frac{f_s * t_a}{n(C)} = 30 \tag{2}$$

Similarly, B is defined as the set of all current peak values from the 1 second interval, as described in equation (3).

$$B = p \in C_j \ \forall \ j \in (n_s, n_s + n_t]$$
$$where, \ n_t = \frac{f_s * t_b}{n(C)} = 60 \tag{3}$$

Thus, A will contain 30 current peaks and B will contain 60 current peaks. Since the absolute values of the current peaks for a given state transition are contingent on the previously occurred events, only the relative magnitudes of current peaks have been used to extract the final set of features. In order to obtain the relative magnitude of the current peaks in set B with respect to set A, the mean steady-state peak current, $\bar{p}$, of set A is obtained first using equation (4).

$$\bar{p} = \frac{1}{30} \sum_{i=1}^{30} \alpha_i \ \forall \ \alpha_i \in A \tag{4}$$

Then, we define the Device Current Signature (DCS) as the set Δ in equation (5).

$$\Delta = x - \bar{p} \ \forall \ x \in B \tag{5}$$

The DCS is the set of features used in this paper to train the classifier. Fig. 2 shows the DCS extracted from set B using (5). The DCS of simultaneously occurring events overlap, which violates the distinctive nature of DCS. Therefore, such events have not been considered.

C. Feature Size

The BLUED dataset defines an event to last atleast 5 seconds, which contains 60000 current data points, i.e., 300 current peaks. Initially a feature size of 300 was considered, but to find the optimal size of feature set, the DCS size was decreased gradually and the f1-score was found to be similar for time intervals greater than 1 second as shown in Table I. For time intervals less than 1 second, a part of the transient state was not captured for certain events and considering that the transients last in the order of milliseconds, a 1 second window was found to be sufficient to capture both the transient and steady state signatures. Hence the optimal time interval required for the extraction of features was found to be 1 second or equivalently 60 current peaks.

VI. EXPERIMENT

Five classifiers were initially evaluated based on cross validation scores using stratified k-fold cross validation technique. As can be observed from Table IV, Random Forests and Extremely Randomized Trees (Extra-Trees) perform significantly better than the others, with Extra-Trees performing the best. Also, Extra-Trees as proposed by Guerts et al. [11], has been proven to be computationally more efficient, and hence was selected over Random Forests. The model was then implemented on a Raspberry Pi 3, which gave a training time of 4.15 seconds and classification of each event was performed in 0.4 seconds on average, validating that the

TABLE V
PRECISION, RECALL AND F1-SCORES FOR THREE DEVICES IN PHASES A AND B

	Phase A			Phase B		
Device Labels	108	111	132	211	129	140
Precision	1.000	0.946	1.000	0.924	1.000	0.625
Recall	0.750	0.995	1.000	0.948	0.917	0.814
f1-score	0.857	0.969	1.000	0.936	0.956	0.707

classification model is capable of running on a low-cost single-board computer in reasonable time. The results obtained on Raspberry Pi 3 are given in section VII.

VII. RESULTS

To analyze the performance of the proposed model, the classification metric of f1-score was used. The model was evaluated on BLUED using 75% of data for training and 25% for testing.

A. Classification of devices

Phase A: Table II shows the confusion matrix for Phase A. It contains 8 devices, of which refrigerator (device label: 111) has the most number of event occurrences, which is significantly larger than the others. This non-uniformity in the number of events for different devices can be attributed to the real-time nature of the dataset. Therefore, it can be observed that the devices with very few events have low classification rates, but since the total number of devices connected to Phase A is comparatively less than Phase B, the overall f1-score of the Phase A is higher.

Phase B: As can be seen from Table III, there are a total of 23 devices in Phase B, and the distribution of number of events for each of the devices is more uniform than Phase A. It was observed that this increase in the number of devices made the occurrences of events more concentrated at some particular intervals of time than others, which could prove to be detrimental to the overall performance. This was overcome due to relatively small size of the window used for extraction of features.

Patri et al have also used BLUED, but grouped devices into classes before performing device classification, with accuracies of 83.75% for classes in Phase A and 77.92% for classes in Phase B. The model proposed in this paper performs classification for individual appliances, with accuracies of 94% and 80% respectively.

Table V shows 3 devices in both Phases A and B with their f1-scores. The overall f1-scores for Phases A and B are given in Table VI. As classification is performed for individual devices, different types of lights - for example, office lights (device label: 149), hallway lights (device label: 152) and basement lights (device label: 159) - which belong to the same class can be satisfactorily classified at most times. This leads to the inference that current peaks can also be used in distinguishing similar appliances in a household.

TABLE VI
OVERALL F1-SCORES FOR PHASE A AND PHASE B

Phase	Precision	Recall	F1-score
A	0.9228	0.9415	0.9315
B	0.8023	0.7939	0.7889

VIII. CONCLUSION

The authors have implemented a machine learning approach for classification of devices in a modern household. The classification model gave overall f1-scores of 0.93 for Phase A and 0.79 for Phase B. The use of current peaks as features for classification of devices was validated and was also found to be feasible for a real-time scenario on an embedded platform. While the Raspberry Pi 3 (35 USD) may not be ideal for a low-cost system, the authors believe that it is the first step in making NILM affordable to consumers in developing countries, and with minor modifications, the classification model can be run on a Raspberry Pi Zero (5 USD), further reducing the cost and making it a more feasible option.

IX. ACKNOWLEDGEMENT

The authors would like to acknowledge Solarillion Foundation for its support and funding of the research work carried out.

REFERENCES

[1] International Energy Outlook 2016 - Energy Information Administration, Eia.gov, 2016. [Online]. Available:http://www.eia.gov/forecasts/ieo/.
[2] Hart, G.W., "Nonintrusive Appliance Load Monitoring", Proceedings of the IEEE, December 1992, pp. 1870-1891.
[3] Zoha, Ahmed, et al. "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey" Sensors 12.12 (2012): 16838-16866.
[4] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," in IEEE Transactions on Consumer Electronics, vol. 57, no. 1, pp. 76-84, February 2011.
[5] Y. F. Wong, Y. Ahmet ekerciolu, T. Drummond and V. S. Wong, "Recent approaches to non-intrusive load monitoring techniques in residential settings," 2013 IEEE Computational Intelligence Applications in Smart Grid (CIASG), Singapore, 2013, pp. 73-79.
[6] C. Duarte, P. Delmar, K. W. Goossen, K. Barner and E. Gomez-Luna, "Non-intrusive load monitoring based on switching voltage transients and wavelet transforms," 2012 Future of Instrumentation International Workshop (FIIW) Proceedings, Gatlinburg, TN, 2012, pp. 1-4.
[7] K. D. Anderson, M. E. Bergs, A. Ocneanu, D. Benitez and J. M. F. Moura, "Event detection for Non Intrusive load monitoring," IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, 2012, pp. 3312-3317.
[8] O. P. Patri, A. V. Panangadan, C. Chelmis and V. K. Prasanna, "Extracting discriminative features for event-based electricity disaggregation," 2014 IEEE Conference on Technologies for Sustainability (SusTech), Portland, OR, 2014, pp. 232-238.
[9] G. Srinivasan, C. Anandan, S. A. K. Jain, S. S. Ahmed and V. Vijayaraghavan, "Low-cost non-intrusive device identification system," 2016 IEEE Dallas Circuits and Systems Conference (DCAS), Arligton, TX, 2016, pp. 1-4.
[10] Filip, Adrian. "BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research." 2nd Workshop on Data Mining Applications in Sustainability (SustKDD). 2011.
[11] Geurts, P., Ernst, D. and Wehenkel, L. Mach Learn (2006) 63: 3. doi:10.1007/s10994-006-6226-1.