Energy and Buildings 158 (2018) 32–36

Contents lists available at ScienceDirect

Energy and Buildings

journal homepage: www.elsevier.com/locate/enbuild

Short communication

Appliance classification using VI trajectories and convolutional neural networks

Leen De Baets*, Joeri Ruyssinck, Chris Develder, Tom Dhaene, Dirk Deschrijver

Department of Information Technology, Ghent University – imec, Technologiepark-Zwijnaarde 15, 9052 Ghent, Belgium

ARTICLE INFO

Article history:
Received 10 April 2017
Received in revised form
11 September 2017
Accepted 27 September 2017
Available online 5 October 2017

Keywords:
Non-intrusive load monitoring
Appliance recognition
VI trajectory
Convolutional neural network

ABSTRACT

Non-intrusive load monitoring methods aim to disaggregate the total power consumption of a household into individual appliances by analysing changes in the voltage and current measured at the grid connection point of the household. The goal is to identify the active appliances, based on their unique fingerprint. An informative characteristic to attain this goal is the voltage–current trajectory. In this paper, a weighted pixelated image of the voltage–current trajectory is used as input data for a deep learning method: a convolutional neural network that will automatically extract key features for appliance classification. The macro-average *F*-measure is 77.60% for the PLAID dataset and 75.46% for the WHITED dataset.

1. Introduction

A basic but crucial step towards increased energy efficiency and savings in residential settings, is to have an accurate view of energy consumption. To monitor residential energy consumption cost-effectively, i.e., without relying on per-device monitoring equipment, non-intrusive load monitoring (NILM) provides an elegant solution. It identifies the per-appliance energy consumption by first measuring the aggregated energy trace at a single, centralized point in the home and then disaggregating this power consumption for individual devices using machine learning techniques.

Classifying active appliances is mostly done by extracting features from the monitored data and training a machine learning classifier. These features are often extracted once it is detected that a device is switched on/off [1]. The type of features depends strongly on the sampling rate of the measurements. When using low frequency data ($\leqslant 1$ Hz), the most common features are the power levels and the on/off durations [2]. A drawback of this approach is that only energy-intensive appliances can be detected. This can be alleviated by performing fine-grained measurements at the cost of an increased data storage rate and more complex data analytics. It is then possible to calculate features like the harmonics [3] and the frequency components [4] from the steady-state and transient behavior of the current and voltage signal. More recently, the possibility to consider voltage–current (VI) trajectories has also been considered [5–7]. Once the features are extracted, they can be fed into different classification methods, like support vector machines (SVM) [13], decision trees [14], or nearest neighbors [15]. Some methods use the complete aggregated power signal as feature. In [16], this is used as input for different convolutional neural networks (one per appliance) that each determine the total power consumption of the corresponding appliance. The total power consumption can also be disaggregated in power traces per appliance. This could be done with the discriminative sparse coding [17], deep neural networks like long short-term memory networks (LSTM), and denoising autoencoders [18,20], or recurrent LSTMs [19,20].

In order to distinguish appliances based on their VI trajectories, powerful measuring devices must be used that are able to sample high frequency data.

In this letter, it is proposed to interpret the VI trajectories as weighted pixelated VI images that can be used as inputs for a CNN. Such networks are often used for classification tasks in computer vision, due to their excellent discriminative power to classify images [8]. In this paper, it is shown that a CNN approach can also be valuable in a NILM context to discriminate active appliances based on the weighted pixelated VI image. The results of this novel approach are benchmarked on the PLAID [10] and WHITED [11] datasets.

2. Weighted pixelated VI image

The VI trajectory of an appliance is obtained by plotting the voltage against the current for a defined time period when the

* Corresponding author.
E-mail address: leen.debaets@ugent.be (L. De Baets).

https://doi.org/10.1016/j.enbuild.2017.09.087
0378-7788/© 2017 Elsevier B.V. All rights reserved.

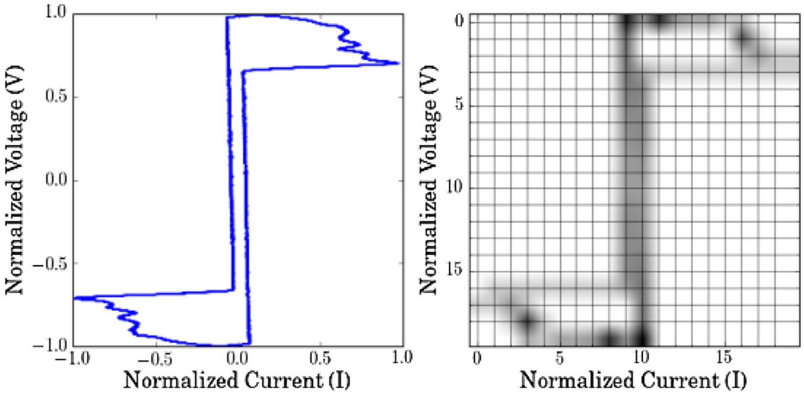

Fig. 1. Transformation from a continuous VI trajectory of a compact fluorescent lamp (left) into the weighted pixelated VI image (right) for $n = 50$.

appliance is turned on. It is shown in [5] that manually extracted features from the VI-trajectory like the enclosed area, slope of the middle segment, etc. can be used to classify the appliances.

Nevertheless, extracting features from the VI trajectory is not straightforward. As an alternative, the VI trajectory can be converted into a pixelated VI image ($n \times n$ matrix) by meshing the VI trajectory. In [6,7], each cell of the mesh is assigned a binary value that denotes whether or not it is traversed by the trajectory. Based on this pixelated VI image, several features can be extracted to classify different power loads [6]. Examples of features are the number of continuums of occupied cells, the binary value of the left horizontal cell and central cell. In [7], the pixelated VI image is re-arranged into an input vector that can be fed directly into a classifier, like random forests, to classify different appliances.

Previous approaches compress the information contained in the VI-trajectory into a limited amount of correlated summary statistics. To take full advantage of the information contained in the VI trajectory, this letter proposes to represent the VI trajectory as a weighted pixelated image. In contrast to [6,7] where the image has continuous values instead of binary values. The necessary processing steps are:

1 Taking the voltage V and the current I when the appliance is active over a certain amount of time (the steady-state behaviour),
2 Normalizing such that $[V, I] \in [-1, 1]^2$,
3 Creating the continuous VI trajectory,
4 Overlaying it with a $n \times n$ mesh,
5 Counting for each cell in the mesh the amount of trajectory points it contains,
6 Normalizing the values of the cells such that the maximum value of the cells is one.

Fig. 1 illustrates the transformation from the continuous VI trajectory to the weighted pixelated VI image.

3. Convolutional neural networks

Once the VI trajectory is transformed into the weighted pixelated image, a CNN is applied for the classification task. CNNs are a type of neural networks (NNs) that are often used in computer vision because they are highly suitable to classify images [8]. The (C)NN takes training samples as input and classifies them by automatically extracting informative features from the data. To this end, an architecture and training procedure is needed.

The architecture of a NN consists of different layers, see Fig. 2. The first layer is always the input layer containing as many nodes as the dimension of a sample (here, $n \times n$). This is followed by one (or more) fully connected layers which are hidden. Each of these layers contains a certain number of nodes that have learnable weights and biases and each of the nodes receives some inputs, performs a dot product and optionally applies a non-linearity. This non-linearity is often obtained by using a rectified linear unit that replaces all negative values by zero. At the end, the output of the last fully connected layer is fed into the output layer. Since the NN is used for classification, the output layer has K nodes with K equal to the number of classes. The values of the output nodes are chosen to lie between 0 and 1 and sum to 1, which is achieved by applying the softmax function. In other words, each node represents the probability that a VI image corresponds to certain class. The output node with the maximal value represents the predicted class.

To create a CNN from a NN, convolutional layers are added. These are placed between the input and output layers as desired and are consequently also hidden. The main difference between a convolutional and fully connected layer is that each node in a convolutional layer is connected to a small region of the input matrix exploiting local correlation, see Fig. 2. In each node, a convolution is performed by adding each element of the input image to its local neighbours, weighted by a matrix called a filter. After the convolutional layer, it is common to implement a pooling layer to downsample the convolved matrix. This reduces the spatial size of the representation, and the amount of parameters, and hence also manages overfitting. This downsampling is achieved by sliding a $d \times d$ window over the input (here, with $d = 2$) and each time outputting the largest element of the window.

The implemented CNN in this letter has the following structure: it takes as input the weighted pixelated VI image (a $n \times n$ matrix, with $n = 50$), and has the following hidden layers: a convolutional layer with f filters of size 5, a pooling layer, another convolutional layer with f filters of size 5, another pooling layer, a fully connected layer with n^2 nodes and an output layer with K nodes. The number of filters f is set to 50. The number of output nodes K is determined by the number of different appliances present in the dataset (i.e., the number of classes). An analysis of alternative parameter settings for n and f proved no significant changes in the results.

4. Model training

Once the architecture is specified, a training procedure is initiated so that the CNN learns to classify the different classes. To this end, multiple training examples are needed. These are images $\mathbf{X} = (X_1, X_2, \ldots, X_N)$ labelled with their corresponding class $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_N)$ where $\mathbf{t}_i$ is a 1-of-K coding of the classes. The aim of the training is to find weights and biases such that a cost function is minimized. Since the class labels are categorical, the cost function is defined as the cross-entropy function [9]:

$$L = -\sum_{i=1}^{N}\sum_{k=1}^{K} t_{i,k} \log(y_{i,k}) \quad (1)$$

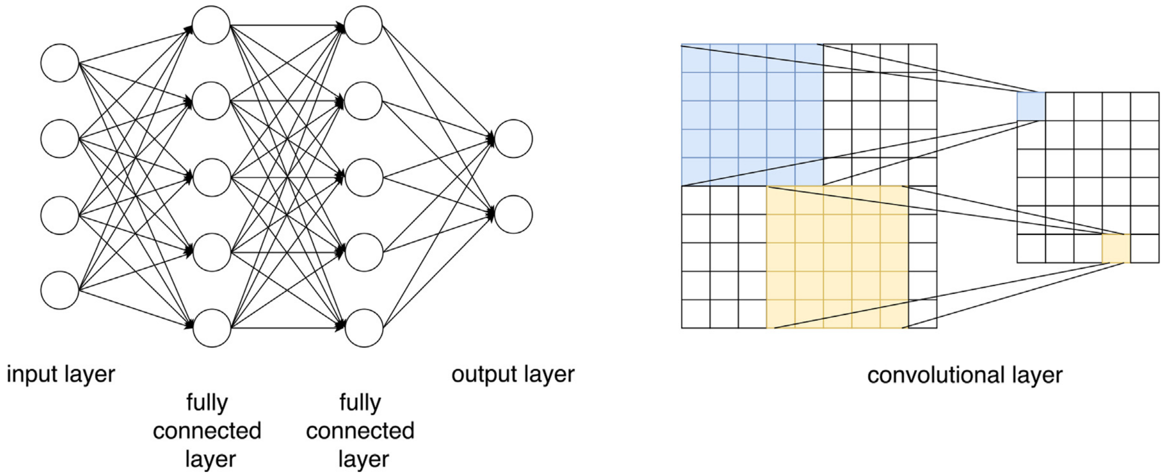

Fig. 2. (left) a NN consisting of an input layer, two fully connected layers and an output layer, (right) a convolutional layer.

where the predicted outputs y_i depend on all the weights and biases in the CNN. The cost function L is minimized using gradient descent, and decreases as the predicted output y_i approximates the real output t_i for all N training samples. As such, the whole CNN learns its weights and biases in such a way that the filters are able to represent spatial connections and features in the data. The reader is referred to the online book 'Neural Networks and Deep Learning' [9] to learn more about (C)NNs and their training.

5. Results

Ideally, to test the proposed method, a dataset having high frequency aggregated and high frequency sub-metered V and I signals would be used. However, to the best of our knowledge, no public dataset includes both. For this reason the PLAID [10] and WHITED [11] datasets are considered. PLAID [10] is a public dataset including sub-metered V and I measurements sampled at 30 kHz for 11 different appliance types. More than 200 individual appliances are available, captured in 55 households. For each appliance, at least 5 start-up events are measured, resulting in a total of 1074 measurements. WHITED [10] is a public dataset including sub-metered V and I measurements sampled at 44 kHz for 46 different appliance types. For each appliance type, 1 to 9 different appliances are available. For each appliance, 10 start-up events are measured, resulting in a total of 1100 measurements.

The presented research on appliance classification is a first step towards a more realistic NILM setting starting from the aggregated power measurements. Still, this is a very meaningful one, as typically appliances are turned on/off one at a time, and the single appliance current (and thus VI trajectory) can be extracted from the aggregated measurements by considering the difference in current before and after the event. Future work needs to confirm the practical feasibility of this idea.

After explaining the evaluation criteria in Section 5.1, the results for the PLAID and WHITED dataset are respectively shown in Sections 5.2 and 5.3.

5.1. Evaluation criteria

Once a CNN model is built, its generalization properties are validated using leave-one-house-out cross-validation, as recommended in [7]. For the PLAID dataset, this can be done straightforwardly as the data is divided per house (55 houses in total). In the WHITED dataset, the annotation of measurement locations is not available. Houses are created artificially by assigning each appliance of each appliance type randomly to one house. The total number of houses is set to 9, which corresponds to the maximum number of appliances per appliance type. As a consequence, appliance types having only one appliance are left out. The final dataset for the experiment contains 22 appliance types.

As proposed in [12], the F-measure is used to evaluate the classification performance and as both datasets are imbalanced, this is done for each appliance type separately.

$$F_a = 2 \cdot \frac{precision_a \cdot recall_a}{precision_a + recall_a} \tag{2}$$

$$precision_a = \frac{TP_a}{TP_a + FP_a} \tag{3}$$

$$recall_a = \frac{TP_a}{TP_a + FN_a} \tag{4}$$

The true positives (TP_a), false positives (FP_a) and false negatives (FN_a), per appliance type a, are summed for each test set. In the end, the average of all the F-measures are taken, leading to the so-called macro-average:

$$F_{\text{macro}} = \frac{1}{A}\sum_a F_a \tag{5}$$

where A is the total number of different appliance types.

5.2. Results on PLAID

In order to obtain the weighted pixelated VI images for the PLAID dataset, the voltage and current are measured over 20 cycles of the voltage signal, resulting in 10,000 samples. In PLAID, there are $K = 11$ appliance types. The F-measure per appliance for the PLAID dataset is shown in Fig. 3. The $F_{\text{macro}} = 77.60\%$. For all appliances except the washing machine, fan, fridge and air conditioner (AC), the F-measure is higher than F_{macro}. When investigating the confusion matrix in Fig. 4, it is clear that the lower F-measure obtained for the washing machine, fan, fridge and AC is caused by confusion among each other. Common electrical components can explain this phenomenon: the AC also has a fan, and both the washing machine and fridge have a motor.

In [21], some F-measure results on the PLAID dataset are given, however the train-test split approach is not done in a leave-one-house-out manner, making comparison pointless. Moreover, the results in [7] are expressed using the accuracy which is an evaluation metric incapable to deal with the imbalance of the dataset.

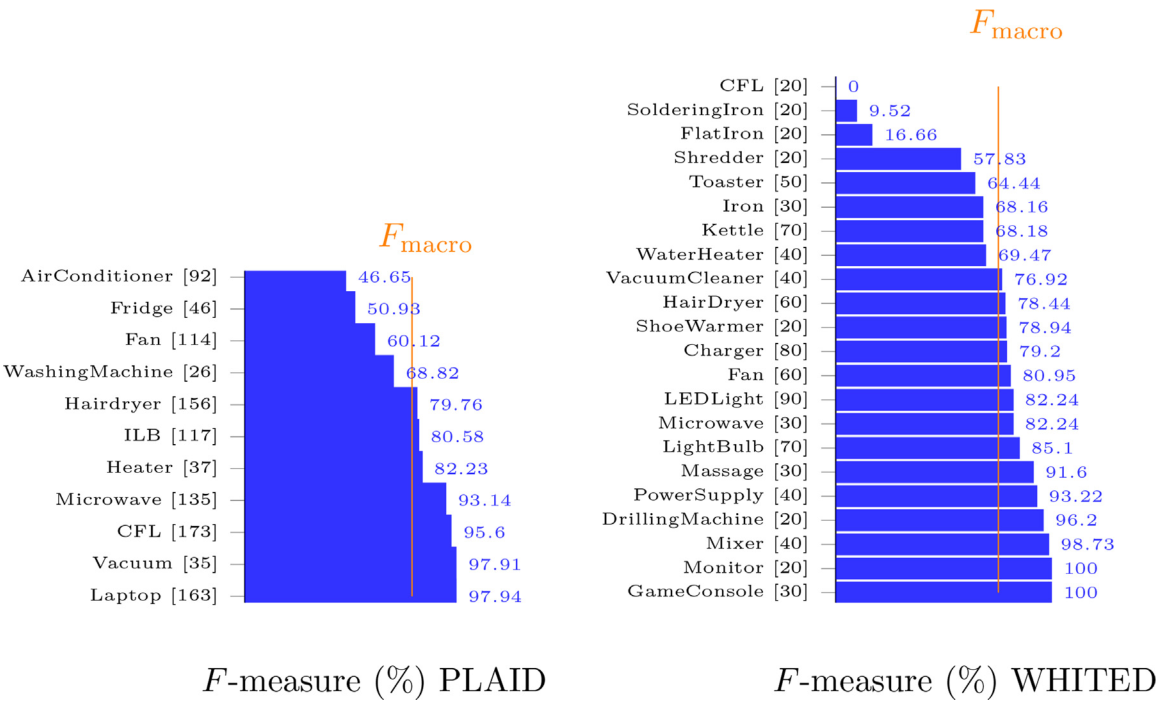

Fig. 3. The F-measure per appliance for the PLAID (left) and WHITED (right) dataset of the CNN for $n = 50$ and $f = 50$. The number of appliances per type are mentioned between the brackets. (CFL = compact fluorescent lamp, ILB = incandescent light bulb).

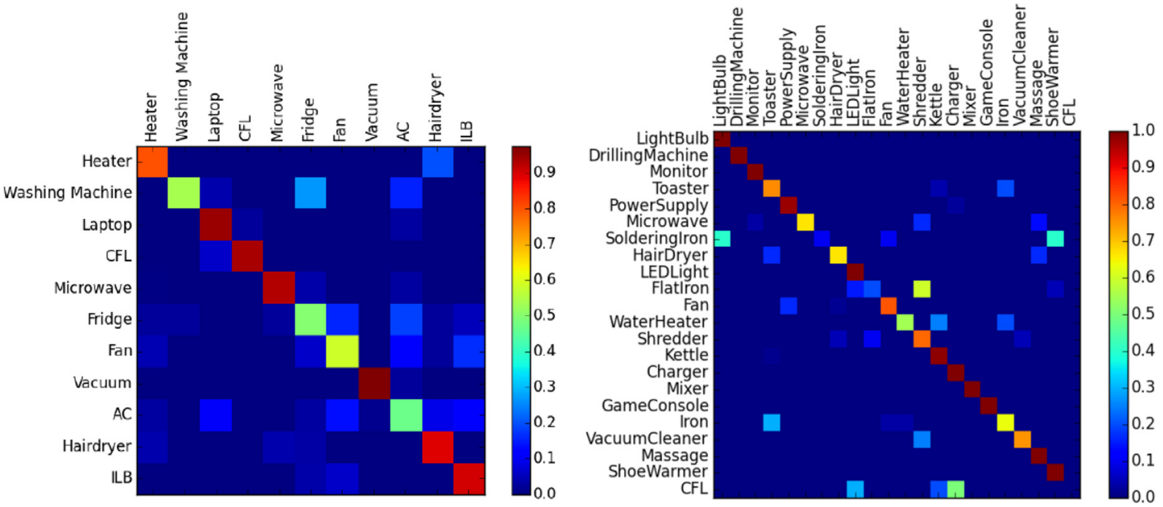

Fig. 4. The confusion matrix of the PLAID (left) and WHITED (right) dataset when classified using $n = 50$ and $f = 50$. The rows are the real labels of the appliance and the columns the predicted labels.

5.3. Results on WHITED

In order to obtain the weighted pixelated VI images for the WHITED dataset, the voltage and current are measured over a time interval of 20 cycles of the voltage signal, resulting in 17,600 samples. In WHITED, there are $K = 22$ appliance types. The F-measure per appliance for the WHITED dataset is shown in Fig. 3. The $F_{\text{macro}} = 75.46\%$. For the soldering iron, flat iron and compact fluorescent lamp (CFL), the method performs poorly. When investigating the confusion matrix in Fig. 4, it is clear that the soldering iron is confused with the light bulb and the shoe warmers (all having resistive heating and more measurements of the light bulbs are present), the flat iron with the shredder (no common electrical components but both only have 20 examples) and the CFL mostly with the charger (both having an element limiting the voltage and more measurements of the chargers are present). This confusion caused by the FNs explains the low F-measure. The FPs for these classes are quite low.

In [21], some F-measure results on the WHITED dataset are given, however the train-test split approach is not done in a leave-one-house-out manner, making comparison pointless.

6. Conclusion

This letter introduces the novel concept of representing VI trajectories as weighted pixelated images and using them as input for a deep learning method: a CNN that can automatically extract relevant spatial features from the VI trajectories. The method is applied on the PLAID and WHITED dataset resulting in a macro-average F-measure of respectively 77.60% and 75.46%. The F-measure per appliance shows that the method gives good results for a large number of appliances. (Admittedly, for a minor selection of them performance is quite low, possibly due to the quality/size of the particular data set used – an issue we leave for future work.) The confusion between appliances can be explained by common electrical components or the small number of appliance instances.

References

[1] L. De Baets, J. Ruyssinck, C. Develder, T. Dhaene, D. Deschrijver, On the Bayesian optimization and robustness of event detection methods in NILM, Energy Build. (2017).

[2] N. Henao, K. Agbossou, S. Kelouwani, Y. Dube, M. Fournier, Approach in nonintrusive type I load monitoring using subtractive clustering, IEEE Trans. Smart Grid 99 (2015) 1–9.

[3] W. Wichakool, Z. Remscrim, U.A. Orji, S.B. Leeb, Smart metering of variable power loads, IEEE Trans. Smart Grid 6 (1) (2015) 189–198.

[4] H.H. Chang, K.L. Chen, Y.P. Tsai, W.J. Lee, A new measurement method for power signatures of nonintrusive demand monitoring and load identification, IEEE Trans. Ind. Appl. 48 (2) (2012) 764–771.

[5] T. Hassan, F. Javed, N. Arshad, An empirical investigation of VI trajectory based load signatures for non-intrusive load monitoring, IEEE Trans. Smart Grid 5 (2) (2014) 870–878.

[6] L. Du, D. He, R.G. Harley, T.G. Habetler, Electric load classification by binary voltage–current trajectory mapping, IEEE Trans. Smart Grid 7 (1) (2016) 358–365.

[7] J. Gao, E.C. Kara, S. Giri, M. Berges, A feasibility study of automated plug-load identification from high-frequency measurements, in: IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2015, pp. 220–224.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 15 (3) (2015) 211–252.

[9] M. Nielsen, Neural Networks and Deep Learning, 2017 (accessed 17.06.17) http://neuralnetworksanddeeplearning.com/.

[10] J. Gao, S. Giri, E.C. Kara, M. Berges, PLAID: a public dataset of high-resoultion electrical appliance measurements for load identification research: demo abstract, in: Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, 2014, pp. 198–199.

[11] M. Kahl, A.U. Haq, T. Kriechbaumer, H.A. Jacobsen, WHITED – a worldwide household and industry transient energy data set, in: Workshop on Non-Intrusive Load Monitoring (NILM), 2016.

[12] S. Makonin, F. Popowich, Nonintrusive load monitoring (NILM) performance evaluation, Energy Effic. 8 (4) (2015) 809–814.

[13] H. Altrabalsi, V. Stankovic, J. Liao, L. Stankovic, Low-complexity energy disaggregation using appliance load modelling, AIMS Energy 4 (1) (2016) 884–905.

[14] M. Nguyen, S. Alshareef, A. Gilani, G.M. Walid, A novel feature extraction and classification algorithm based on power components using single-point monitoring for NILM, in: IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), vol. 3, 2015, pp. 7–40.

[15] K. Basu, A. Hably, V. Debusschere, S. Bacha, G.J. Driven, A. Ovalle, A comparative study of low sampling non intrusive load dis-aggregation, in: 42nd Annual Conference of the IEEE on Industrial Electronics Society, vol. 513, 2016, pp. 7–5142.

[16] C. Zhang, M. Zhong, Z. Wang, N. Goddard, C. Sutton, Sequence-to-point learning with neural networks for nonintrusive load monitoring, 2016 arXiv:1612.09106.

[17] Z. Kolter, S. Batra, A. Ng, Energy disaggregation via discriminative sparse coding, Adv. Neural Inf. Process. Syst. (2010) 1153–1161.

[18] J. Kelly, 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, vol. 5, 2015, pp. 5–64.

[19] L. Mauch, B. Yang, A new approach for supervised power disaggregation by using a deep recurrent LSTM network, in: IEEE Global Conference on Signal and Information Processing (GlobalSIP), vol. 6, 2015, pp. 3–67.

[20] P. Nascimento, Applications of Deep Learning Techniques on NILM, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, 2016 (MSc).

[21] M. Kahl, A. Ul Haq, T. Kriechbaumer, H.A. Jacobsen, A comprehensive feature study for appliance recognition on high frequency energy data, in: Proceedings of the Eighth International Conference on Future Energy Systems, vol. 12, 2017, pp. 1–131.