

End-to-end NILM System Using High Frequency Data and Neural Networks

Marchesoni-Acland*
Instituto de Ingeniera Elctrica
Universidad de la Repblica
Montevideo, Uruguay
marchesoni@fing.edu.uy

Mariño*
Instituto de Ingeniera Elctrica
Universidad de la Repblica
Montevideo, Uruguay
cmarino@fing.edu.uy

Masquil*
Instituto de Ingeniera Elctrica
Universidad de la Repblica
Montevideo, Uruguay
eliasmasquil@gmail.com

Masaferro
Instituto de Ingeniera Elctrica
Universidad de la Repblica
Montevideo, Uruguay
pmasaferro@fing.edu.uy

Fernndez
Instituto de Ingeniera Elctrica
Universidad de la Repblica
Montevideo, Uruguay
alicia@fing.edu.uy

Abstract—Improving energy efficiency is a necessity in the fight against climate change. Non Intrusive Load Monitoring (NILM) systems give important information about the household consumption that can be used by the electric utility or the end users. In this work the implementation of an end-to-end NILM system is presented, which comprises a custom high frequency meter and neural-network based algorithms. The present article presents a novel way to include high frequency information as input of neural network models by means of multivariate time series that include carefully selected features. Furthermore, it provides a detailed assessment of the generalization error and shows that this class of models generalize well to new instances of seen-in-training appliances. An evaluation database formed of measurements in two Uruguayan homes is collected and discussion on general unsupervised approaches is provided.

Index Terms—NILM, ANN, energy disaggregation

I. INTRODUCTION

Climate change is a consequence of greenhouse gas emissions and causes more extreme climate conditions that imply severe negative effects. According to the Intergovernmental Panel on Climate Change (IPCC), electricity and heat production accounts for a quarter of total global emissions. To limit temperature growth to 1.5 K ambitious efforts have to be carried out, including improving energy efficiency. Energy efficiency implies reducing the consumption for a given comfort or production level. Information availability is crucial to improve in this line. Non intrusive load monitoring (NILM) systems were introduced by Hart [2] and their objective is to get valuable information of consumption in an electric installation from measurements taken at only one point.

This work presents an end-to-end NILM system. This implies building or getting a meter, and using some disaggregation algorithm. The meter is usually connected to the main electrical panel of a house. The meter's goal is to measure, directly or indirectly, the power consumption of the house. A common solution is to use the electric utility's smart meters,

with the drawback of a low sample rate, 1.1 mHz in Uruguay. Another direct solution is to buy some commercial meter. The most common drawbacks are the low sample rate and lack of flexibility in its use. These meters are often equipped with a communication system that allows data transmission to other points, via Ethernet, WiFi, or IoT oriented protocols as MQTT or ZigBee. The maximum sample rate of commercial meters is about 1 Hz [3], [9]. The other options are designing a custom meter, as was done in this work, or purchasing a commercial acquisition board. For more details see [10].

The other axis of the solution refers to disaggregation methods. Classical algorithms are Combinatorial Optimization and Factorial Hidden Markov Chains [4], and an important reference for this work that introduces Artificial Neural Networks (ANNs) for the problem is [6]. In that work ANNs show promise, as it is reported that their generalization capabilities are good. In fact, performance seems to be better for unknown appliances than for appliances seen in training. This work replicates and amplifies that of Kelly, using the UK-DALE dataset [5] and proposing modifications to the architectures there defined. These modifications allow the introduction of high frequency features and the adaptability of the dimension of the autoencoder's latent space according to the appliance. An introduction to ANNs can be found at [7].

This work involves developments that are independent, although related to each other. On the one hand, the custom meter and the data collection system make up the low level component of the project. On the other hand, the software or signal processing part involves appliance identification over the PLAID database [1], neural network models for disaggregation and the theoretical high level discussion of unsupervised approaches.

The contributions of this work are listed as follows:

- The collection of the first NILM oriented dataset in Uruguay (Subsection II-C).

*These authors equally contributed.

- The study of high frequency features that yield a competitive appliance identification performance when used through a Random Forest classifier (Section IV).
- The validation of the usefulness of the algorithms presented in [6].
- The proposal and testing of a method for including high frequency features as input of ANN models along with the deeper autoencoder variant (Section V).
- The assesment of the generalization error of ANN based methods for NILM (Section VI).
- A macro and general description of unsupervised and scalable NILM methods (Section VII).

This paper is organized as follows. In Subsection II-A the custom meter is introduced along with its characteristics. This meter is integrated into the data collection system described in Subsection II-B. In Section III the data preparation is explained, including synthetic data and training, validation and testing splits. Then, a detailed study on high frequency features is described in Section IV, that is used to create the multivariate time series to be used by the models. A total of seven models are described in Section V, making emphasis on the training (Subsection V-A) and the model selection (Subsection V-B) procedures. Next, the evaluation scheme and the results are presented in Section VI. Furthermore, a discussion on the formulation of unsupervised approaches is provided in Section VII.

II. DATA COLLECTION

A. Custom meter

The first step towards a NILM system is building a device that can measure at a high enough frequency. The frequency requirements vary. For instance, public databases range from 1 Hz to 44 kHz, the private company Sense [12] says its meter’s sample rate is 1MHz, and there exists applications of NILM using frequencies up to 100 MHz [8]. Some experiments concerning the current clamp at use were carried out and revealed that it does not filter signal components under $7kHz$. The spectral analysis of different appliances’ current signals and the will to make comparison possible caused us to choose as Analog Digital Converter (ADC) a configurable soundcard named “audio injector” with a sample rate of up to 96kHz, although the final sample rate was set to 14kHz due to storage considerations. The meter was completed by printing a custom signal adaptation circuit, as shown in Figure 1 and using a Raspberry Pi 3B+ that provides the required software flexibility and allows fast prototyping.

B. Labeled data collection

In order to properly test any NILM system in terms of its disaggregation capabilities some labeled data set must be used. A data collector system should coordinate the non intrusive meter and some intrusive meters that measure individual appliances. The system coordination was implemented on the non intrusive meter, more specifically, on the Raspberry Pi. The computer’s role was to save and compress the high frequency measurements in `.flac` files while serving as a

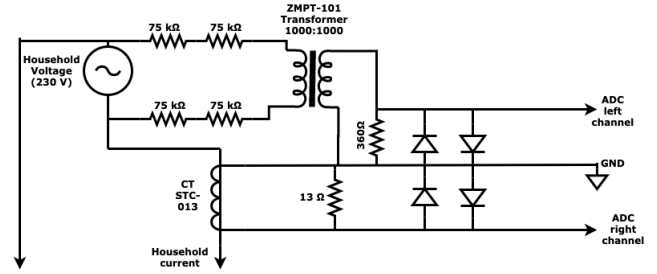


Fig. 1. Circuit diagram used in the custom meter.

MQTT server to the intrusive meters. The intrusive meters shown in Figure 2 were provided by the Uruguayan national electricity utility (UTE) and report active power at a 1 minute sample period. The system was programmed to start running as soon it was energized, and it is able to send alerts if some component is malfunctioning. The configuration files admit flexibility in terms of sample rate (up to 96kHz), bit depth (32 bits), compression period (1 hour `.flac`) and allow to write the data to external usb storage devices, save it in another computer in the same LAN or send it trough the TCP/IP to an external FTP server. A complete visual description of the system is provided in Figure 3.



Fig. 2. Intrusive meter provided by UTE.

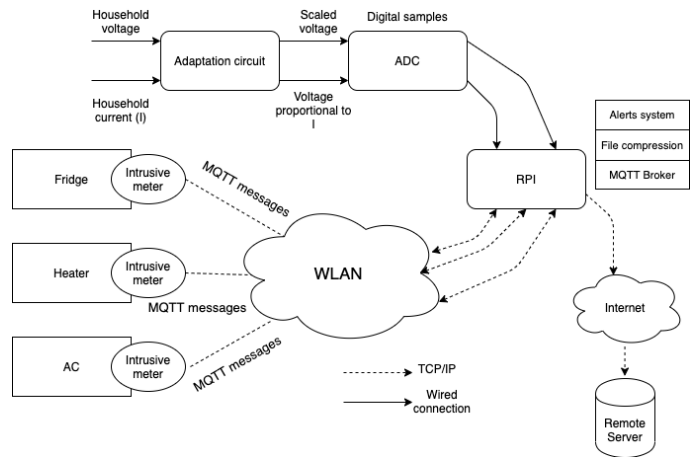


Fig. 3. Complete diagram for the data collection system.

C. Uruguayan data

Data were taken from two households in Montevideo, Uruguay, with a nominal frequency of 50Hz and a nominal

voltage of $230V_{\text{rms}}$. The total amount of recorded time sums up to 3 months of data or 0.5TB. In the first house 7 intrusive meters were installed in the fridge, electric water heater, microwave, washing machine (from now on just “washing”), air conditioner and bedroom plugs, whereas in the second house 8 intrusive meters took the measurements from an electric oven, an electric water heater, two air conditioners, a fridge, a washing machine, a dishwasher and a kettle. These appliances were the responsible for the majority of the households power consumption. The next section presents the processing of these data that serve as input for the neural network based models.

III. DATA

A. Inputs and outputs

The neural network algorithms to be presented in Section V belong to the class of supervised machine learning algorithms. This means that the algorithm is obtained or trained from a set of values (\mathbf{X}, \mathbf{Y}) where \mathbf{X} is a vector containing all input examples and \mathbf{Y} is a vector containing their corresponding labels or target values. On the one hand, this work uses as input $\mathbf{x} \in \mathbf{X}$ a univariate power time series or a multivariate time series that includes the former in one of its dimensions. These time series have a 6 seconds period and their length is given by the size of a window that varies according to the appliance as shown in Table I.

TABLE I
WINDOW SIZE USED.

	Window size (minutes)
Kettle	13
Fridge	60
Washing m.	180
Microwave	10
Dishwasher	150

On the other hand, an output $\mathbf{y} \in \mathbf{Y}$ takes two possible forms, corresponding to the two ANN architectures to be presented in Section V. The first of them involves three values: the beginning of the appliance activation, the end of the appliance activation, and the mean power consumed between these instants. An appliance’s activation is extracted by a function that takes as parameters the minimum and maximum switched on time of the appliance, the on-power threshold and the border or padding for the window. The second possibility for \mathbf{y} is an univariate power time series of the appliance, which has the same length and period as the input.

B. Data preparation

It should be noted that for Uruguayan data a first order hold is used to upsample the 1 minute period measurements. The Uruguayan data was used only for evaluation and not for training. The training was based on the UK-DALE dataset, whose detailed description is found on [5]. Succinctly, this database is formed up by measurements of five houses, three of which also include high frequency measurements. The set \mathbf{X} is built by activations extracted from this database together with non activations in equal proportion. A non activation could

be any window that does not fully include the functioning period of the appliance as defined by the activation-extracting function.

The multivariate time series for both Uruguayan and UK-DALE data had to be obtained. For the latter, web scrapping was used to download the 7.6TB of data, from which the multivariate series values were computed for each required datetime. Comparison of power values extracted from the high frequency time series against UK-DALE’s low frequency power data was made in order to check the correctness of the procedure. The code was reused on uruguayan data. The two high frequency features computed, namely form factor and phase shift of fundamental components of current and voltage, were selected for inclusion on the inputs after the analysis presented in the next section.

C. Synthetic data

Data augmentation was also effectuated by superposing to an appliance activation other appliances activations with some probability, defined as $p = 0.4$ of adding a “distractor” appliance to the individual activation. The sum of these individual power values then composes the aggregate series to be used as input. For samples not containing activations of the target appliances, only “distractor” activations were included with probability p . It should be noted that synthetic data for multivariate series can not be constructed, as there are no individual measurements of the high frequency features.

D. Dataset division

The usual preparation of data in supervised learning algorithms involves dividing the dataset into three sets: training set, validation set, and testing or test set. We follow this use, but there will be four test sets to be used. For each appliance, measurements of one of the houses of the UK-DALE dataset are set aside for the test set I. For the measurements in the other houses the last two weeks of data are also set aside for the test set II. The rest of the data is used to form the training and validation sets as will be shortly explained. The last two test sets correspond to the measurements taken from the two Uruguayan households.

From the time series that form the training and validation sets activations are extracted via the activation-extracting function. Also, non-activations are selected from the time period between two activations. The resulting activations dataset is approximately balanced, and it is split randomly into the training set (80%) and the validation set (20%).

E. Preprocessing

It is a well known practice the standardization and normalization of the signals to be used at training. The way we do that is extracting the mean standard deviation of the input training samples σ_{input} and the maximum power output value \max_{target} of the training targets. The preprocessing implies, for each input sample, independently of which set it belongs to, its own mean is subtracted before dividing by σ_{input} . For training, the targets are divided by \max_{target} , and for prediction, the output is scaled for the same value.

TABLE II
PERFORMANCE OVER SUBSETS OF FEATURES.

Instances Features / Classifier	1074		1793	
	KNN	RF	KNN	RF
Transient	61.70	88.68±0.17	59.35	87.06±0.06
Steady state	75.88	87.24±0.28	66.76	84.23±0.25
Steady state + Transient	-	91.47±0.09	-	88.33±0.25
Steady state + VI	75.97	86.67±0.49	66.82	84.14±0.43
All features	-	92.79 ± 0.13	-	89.08±0.38

VI corresponds to pixels of the VI image. The tolerance is the standard deviation between the three runs.

IV. HIGH FREQUENCY FEATURES

In order to select the high frequency features to be included in the form of a multivariate time series together with the active power, a simpler problem was confronted. This sub-problem is the identification of an appliance’s name from its current and voltage signature. The PLAID database [1] is made up of 1000+ measurements of isolated appliances from 55 different households at a 30kHz sample rate. More than 30 features were computed for each voltage and current waveform: power values as defined in [13], VI trajectory image, statistical moments, audio features, among others. As each instance contains the switching-on of the appliance, the transient was included in most of the instances.

Extracting this transient allowed to compute features over both transient and regime states. The features’ importance was evaluated via Random Forest (RF) classifier and Mutual Information (MI) measure. Figure 4 shows the normalized importance assigned to the transient features by the RF classifier and the MI measure. The criteria used to select the most important features was based on the assumption that any feature that is useful for the aggregate problem should be useful in this sub-problem too. The selected features comply with arbitrarily defined requirements: the selected features should be both transient and regime features (for instance, transient duration is not considered as a candidate) and should belong to the top 10 features for transient and regime states under the importance criteria given by both the RF classifier and the MI measure. The two features that satisfy the requirements above are the form factor of the current and the phase shift between the fundamental component of the current and voltage signals.

The value of the original set of considered features is denoted by the high disaggregation performance obtained, showed in Table II, where 1-Nearest-Neighbor was used as a proxy classifier. It should be noted that the result that corresponds to all the possible features is only surpassed by the best result in [11] by a 0.5% difference in accuracy, being superior to all other results found in the literature. This proves that the set of all features is powerful when used to feed a RF classifier.

V. MODELS

The trained models are the originally used in [6] although some additional modifications were proposed. These models’ names are “autoencoder” and “rectangles” as named in the

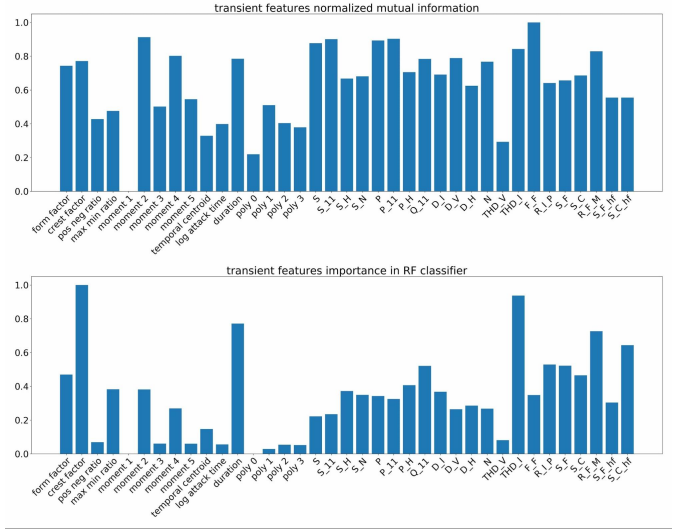


Fig. 4. Transient features importance via random forest (top) and mutual information (bottom).

previously cited work. The visual description is provided in Figure 5 and the Tensorflow model is exposed in the Section A. The proposed modifications correspond to:

- Changing the first convolutional layer so it is able to get multivariate time series as input.
- Making the autoencoder deeper and the code length dependent on the input window size.

The seven models considered are presented in Table III. The “baseline models” are the ones trained with the augmented low frequency training set, i.e. the ones trained with synthetic data.

TABLE III
USED MODELS.

	Low freq	Synthetic data	High freq	“Big”
Rectangles	Yes	Yes	Yes	No
Autoencoder	Yes	Yes	Yes	Yes

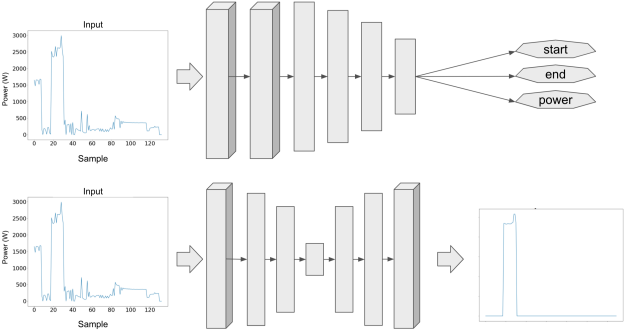


Fig. 5. Diagram of the implemented ANN architectures. Rectangles network (top) and Autoencoder (bottom). 3D blocks correspond to convolution or upsampling layers.

A. Training

It is common knowledge that ANNs must be trained. Any training procedure involves finding a set of weights that achieves a low loss over the training set while at the same time keeping the loss over the validation set controlled. We found that the training delicate, as it is easy for the optimizers to get stuck into local minima, being the most evident case any set of parameters that yields always the same output. To avoid local minima and get a good performance, multiple runs were made for each model. To find a good set of parameters for each model, two techniques were used. The first involved a grid search in the space of optimizers. After discarding Adadelta [15] and large learning rates, six points were tried. These points arise from the combination of the learning rates values of 0.002, 0.001, 0.0005 and the Adam and Adamax optimizers [7]. The training procedure for one of the models for the microwave is shown in Figure 6. The second technique involves tracking the error over the validation set and saving the model corresponding to the iteration with the lowest error. The number of iterations for every model ran is 200. This training procedure gives a total of 6 runs per model \times the 7 models evaluated \times 5 appliances = 210 runs. The 42000 iterations were computed on a NVIDIA TITAN Xp graphics card.

TABLE IV
AUC OF BEST EXPERIMENTS FOR THE MICROWAVE.

	Low Freq	Synthetic Data	High Freq	“Big”
Rectangles	0.933	0.937	0.927	-
Autoencoder	0.936	0.944	0.949	0.932

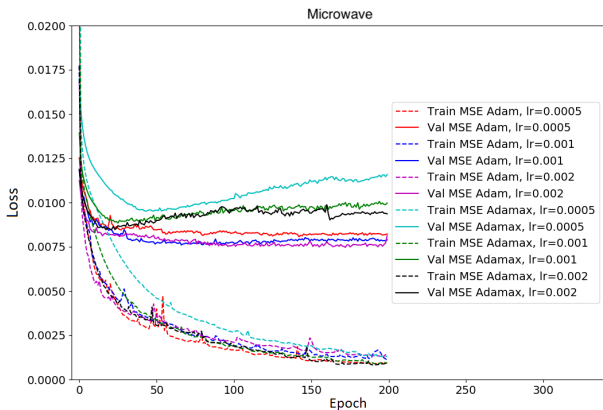


Fig. 6. Losses evolution during training. Training loss (solid line). Validation loss (dashed line). The loss function is the MSE.

B. Selection

The goal is to get the “best” model for each appliance, i.e. to select for each appliance only one model of the 42. As the loss function is the mean squared error between the output series and the target series it is not expected for this regression score to be unacceptably large for any model. After manually analyzing the output we found that the Area Under the Curve

(AUC) value of the Receiving Operating Characteristic (ROC) curve is strongly related to a visually good performing model. This metric will be used in the next steps to select the best models. For each of the 7 models per appliance the selected set of parameters comes from choosing the weights that achieve the largest AUC between the 6 sets of weights of the grid.

To calculate the ROC the problem has to be turned into a classification one. To make this possible two criteria have to be defined: (i) the definition of the ground truth label, using for this the activation-extracting function described in Section III, and (ii) the definition of the predicted class, that is done by defining a threshold for the maximum on the predicted power series. This first step gives tables similar to Table IV for each appliance. From these tables the best of the seven models is chosen by maximizing the AUC again. The results of which the best model is for every appliance are shown in Table V. Full results are included in the Appendix A.

TABLE V
SELECTED MODEL.

	Selected model
Kettle	High frequency autoencoder
Fridge	High frequency rectangles
Washing m.	High frequency rectangles
Microwave	High frequency autoencoder
Dish washer	Big autoencoder

VI. RESULTS

Once the models are chosen, there are many evaluations to be made. These arise from the combination of two evaluation procedures, “rolling window” and “activations”. These procedures will be described in Subsection VI-B. Besides the evaluation procedures there are the four test sets described in Section III. Combining the different test sets with the two evaluation procedures allows finding the answers to the following questions:

- 1) How do the models work for what they were trained for? - this corresponds to the evaluation over test set II using the “activations” procedure.
- 2) How do the models generalize for unseen appliances? - this corresponds to the evaluation over test set I using the “activations” procedure.
- 3) How would the models behave in a real case scenario? - this corresponds to the evaluation over test set I using the “rolling window” procedure.

Models are further compared with the performance of the best models found at [6] as a reference. These are the ones that include synthetic data into the training set. Finally, when answering the last question the value of the metrics to be introduced in the next subsection is reported.

A. Metrics

To report the results a few common metrics were selected, in line with [6]. The metrics for the regression problem quantify how well the outputs approximate the targets, and are affected when time shifts occur. The classification metrics are agnostic

to in-window time shifts, and only consider if the appliance is detected as energized or not. The regression metrics to be presented are the Mean Absolute Error (MAE) and Relative Error In Total Energy (REITE):

$$\text{REITE} = \frac{|\hat{E} - E|}{\max(\hat{E}, E)} \quad (1)$$

$$\text{MAE} = \frac{1}{N} \sum_t |\hat{y}_t - y_t| \quad (2)$$

where N is the total number of power values considered. The predictions can be Positive (P) or Negative (N) and result True (T) if they are equal to the ground truth label or False (F) otherwise. Traditional classification metrics are defined as:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4)$$

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (5)$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

B. Evaluation procedures

Two evaluation procedures are used in this work. The ‘‘activations’’ procedure involves extracting activations and non-activations from a test set, making estimates for each window and comparing that with the ground truth. It is the simplest evaluation procedure and it has two important features: (i) the resulting windows are approximately balanced and (ii) the method is the same as the one used over the validation set.

Alternatively, the ‘‘rolling window’’ procedure is as unbalanced as the real use of the appliances. Furthermore, it is applied over the whole time series, resembling a real use case. This procedure starts estimating the output for each one of the inputs determined by a rolling window with stride= 1. This is, estimate one output, shift the window 6 seconds, estimate another output, and so on. At the end, for every datetime, there are $w = \text{window size}$ estimates, that are averaged and multiplied by a factor of $\frac{w}{w-2a}$, where a is the average activation length for the appliance calculated over the training and validation sets, in order to account for the fact that our ANNs only recognize *complete* activations. After the estimation stage, the same time series is divided in non overlapping windows over which the resulting estimates are compared with the ground truth.

C. Results

Numerical results will be summarized in this sub-section. First, it can be seen in Table VII that the best models yields the best AUC for three of the five appliances for the case depicted in question 1. This means that the models perform very well when evaluated over new instances of previously seen appliances and that the training and selection

procedures were correctly designed. Second, Table VII shows that the performance of the best models declines more than the performance of the reference models when testing over unseen appliances. The conclusion is that the generalization capability of the model to probability distributions other than the one generating the training and validation data is limited. Finally, Tables VI-C and VI-C summarizes the performance obtained on the real case scenario over the unseen appliances of the UK-DALE database. A MAE under 60W for almost every appliance and good classification metrics for three of five appliances are indicative of acceptable results. Note here that the threshold used to get the classification scores was defined for each appliance by maximizing the F_1 score over the validation set.

TABLE VI
AUCS SCORES VIA ACTIVATIONS METHODOLOGY.

		Test set II	Validation set
Kettle	Best model	0.999	0.984
	Autoencoder	0.959	0.977
	Rectangles	0.941	<u>0.982</u>
Fridge	Best model	0.941	0.901
	Autoencoder	0.500	0.500
	Rectangles	<u>0.871</u>	0.811
Washing m.	Best model	0.850	0.951
	Autoencoder	0.884	0.875
	Rectangles	0.812	<u>0.900</u>
Microwave	Best model	0.976	0.949
	Autoencoder	0.932	0.944
	Rectangles	<u>0.976</u>	0.937
Dishwasher	Best model	0.947	0.997
	Autoencoder	0.986	<u>0.989</u>
	Rectangles	0.954	0.981

Evaluation via ‘‘activations’’ methodology. Bold font indicates the best score achieved by the three models. Underlines indicate best score achieved between datasets for each model.

TABLE VII
AUC SCORES VIA ACTIVATIONS METHODOLOGY.

		Test set II	Test set I
Kettle	Best model	0.999	0.983
	Autoencoder	0.959	0.993
	Rectangles	0.941	<u>0.994</u>
Fridge	Best model	0.941	0.703
	Autoencoder	0.500	0.500
	Rectangles	0.871	<u>0.887</u>
Washing m.	Best model	0.850	0.795
	Autoencoder	0.884	0.908
	Rectangles	0.812	<u>0.884</u>
Microwave	Best model	0.976	0.879
	Autoencoder	0.932	0.891
	Rectangles	0.976	0.962
Dishwasher	Best model	0.947	0.962
	Autoencoder	0.986	0.984
	Rectangles	0.954	0.973

Evaluation via ‘‘activations’’ methodology. Bold font indicates the best score achieved by the three models. Underlines indicate best score achieved between datasets for each model.

The results obtained for the Uruguayan households are presented in Table X. For the household that only contains three appliances the performance is bad. No model generalizes well

TABLE VIII
AUCS SCORES VIA ROLLING WINDOW METHODOLOGY.

	Test set II	Test set I
Kettle	1.000	0.998
Fridge	0.854	0.751
Washing m.	0.763	0.864
Microwave	0.973	0.956
Dishwasher	0.898	0.962

TABLE IX
RESULTS OVER TEST SET I OF THE BEST MODELS VIA ROLLING WINDOW METHODOLOGY.

	Acc.	Prec.	Recall	F1	MAE	REITE
Kettle	0.987	0.686	0.967	0.802	22.48	0.609
Fridge	0.585	0.545	0.959	0.695	42.04	0.305
Washing m..	0.612	0.107	0.904	0.191	237.98	0.962
Microwave	0.835	0.019	0.964	0.038	58.48	0.173
Dish washer	0.961	0.679	0.743	0.710	45.00	0.639

to this household. For the other one, a better performance was obtained. The models could be useful to get some information for three of the five appliances, namely the ones in which F_1 score is high. However, these models' performance is far from excellent and they do not seem suitable for standalone use.

We present visual examples of some of the experiments under the "rolling window" evaluation scheme. The softness of the curve is due to the averaging needed on the estimation stage.

VII. ABOUT UNSUPERVISED APPROACHES

ANNs are being used widely in the supervised setting, and they present a good performance when carefully trained with data that resembles the real case scenario. But the best possible performance of supervised methods is bounded by the characteristics of the training datasets. These have to be extensive and include many kind of appliances and their combinations in order to make possible an adequate generalization capability. Synthesizing data is a good approach to solve this issue. Another approach, that we will present here as a theoretical macro design, is to shift from the supervised approach to an unsupervised learning approach. Notwithstanding, there are other methods that involve learning and could be useful, such as active learning approaches.

The unsupervised approach is presented in a generic way. This takes into account the requirements of a NILM algorithm and focuses on its scaling ability. Any learning algorithm should have a loss function or error signal. The most intuitive error definition for the unsupervised case is the reconstruction error of the aggregate power time series. The minimization of this error implies a good estimation of the aggregate power signal. Furthermore, the estimation of the aggregate power should be done from useful data, for instance from the appliances' power consumption estimates. The name we assign to the block that creates an estimated aggregate signal from useful information is "simulator". What this framing is implying is that minimizing the error between the aggregate power signal and an estimation of it from some useful esti-

mates is forcing the useful estimates to be accurate. The last block needed to complete this generic formulation is the black box that computes the useful estimates, which represents the disaggregation algorithm that adjusts its parameters according to the error signal.

Every component of the proposed general algorithm is presented in Figure 7 and is a subject of research. We consider that most of the unsupervised approaches can be framed this way, although the presented building blocks are not necessarily simple nor independent.

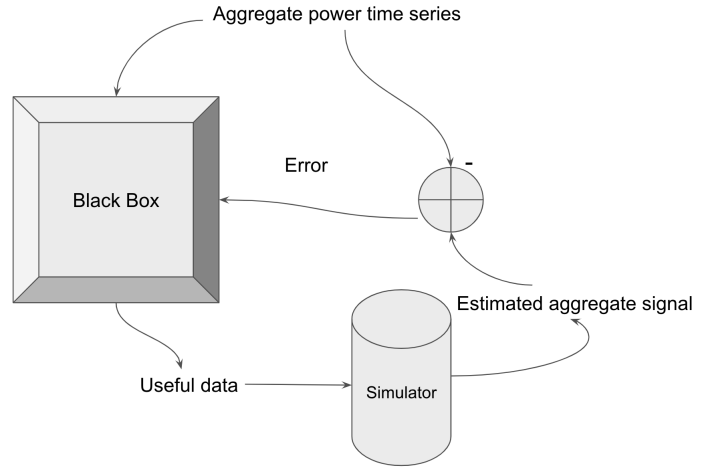


Fig. 7. Diagram of the general unsupervised NILM algorithm.

VIII. CONCLUSIONS

To conclude, let us note that a NILM system can be built and evaluated as was done in this work. The implemented data collector system is robust, as it was capable of collect measurements without intervention for three months.

The first conclusion refers to the high frequency features. These were studied and using the MI measure and the RF classifier two of them were selected: phase shift and form factor. The proposed methodology to integrate high frequency information in the algorithm, that involved obtaining a low frequency multivariate time series of descriptors, was correctly implemented. Furthermore, the model selection procedure selected the models that included high frequency information as the best model for almost all studied appliances. The good performance obtained over the test set II, the one containing known appliances, proves the value added by these high frequency features.

Secondly, neural network based models can achieve very good performance metrics for appliances that were seen during training. This is the expected behavior of correctly trained supervised approaches.

Notwithstanding, the generalization power to appliances unseen during training is limited, although it can not be ignored that the number of houses used for training is less than 5. The results are not as good over the test set I than over the test set II. This performance decline is more notorious for test

TABLE X
RESULTS OVER URUGUAYAN HOUSEHOLDS VIA ROLLING WINDOW METHODOLOGY.

Appliance	Household 1						Household 2					
	Accuracy	Precision	Recall	F1	MAE (W)	REITE	Accuracy	Precision	Recall	F1	MAE (W)	REITE
Kettle	-	-	-	-	-	-	0.953	0.286	0.545	0.375	75	0.909
Fridge	0.759	0.781	0.959	0.861	143	0.507	0.918	0.990	0.926	0.957	96	0.071
Washing m.	0.071	0.057	1.000	0.107	793	0.995	0.323	0.300	1.000	0.462	691	0.971
Microwave	0.506	0.018	0.650	0.036	71	0.920	0.533	0.066	0.818	0.122	90	0.880
Dish washer	-	-	-	-	-	-	0.859	0.720	0.720	0.720	150	0.478

sets corresponding to the Uruguayan households. However, for one of these households, the AUC values surpass 0.8, denoting a respectable performance.

Moreover, the AUC was used as the main metric for model selection, using the ROC as was recommended in [14]. Finally, unsupervised approaches were decomposed in a few building blocks, and we hope this helps conceptualize this kinds of approaches in the future.

IX. ACKNOWLEDGEMENTS

The authors thankfully acknowledge the financial support provided by Uruguays National Research and Innovation Agency (ANII), the Julio Ricaldoni Foundation and the National Administration of Power Plants and Electrical Transmissions.

REFERENCES

- [1] Jingkun Gao, Suman Giri, Emre Can Kara, and Mario Bergés. Plaid: a public dataset of high-resolution electrical appliance measurements for load identification research: demo abstract. In *proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, pages 198–199. ACM, 2014.
- [2] George William Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.
- [3] Ming Jin, Ruoxi Jia, and Costas J Spanos. Virtual occupancy sensing: Using smart meters to indicate your presence. *IEEE Transactions on Mobile Computing*, 16(11):3264–3277, 2017.
- [4] Jack Kelly, Nipun Batra, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. Nilmtk v0.2: A non-intrusive load monitoring toolkit for large scale data sets. In *The first ACM Workshop On Embedded Systems For Energy-Efficient Buildings at BuildSys 2014*, Memphis, USA, 2014.
- [5] Jack Kelly and William Knottenbelt. The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. *Scientific data*, 2:150007, 2015.
- [6] Jack Kelly and William J. Knottenbelt. Neural NILM: deep neural networks applied to energy disaggregation. *CoRR*, abs/1507.06594, 2015.
- [7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [8] Shwetak N Patel, Thomas Robertson, Julie A Kientz, Matthew S Reynolds, and Gregory D Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line (nominated for the best paper award). In *International Conference on Ubiquitous Computing*, pages 271–288. Springer, 2007.
- [9] Hamed Nabizadeh Rafsanjani, Changbum R Ahn, and Jiayu Chen. Linking building energy consumption with occupants energy-consuming behaviors in commercial buildings: Non-intrusive occupant load monitoring (niolm). *Energy and Buildings*, 172:317–327, 2018.
- [10] Antonio Ruano, Alvaro Hernandez, Jesus Ureña, Maria Ruano, and Juan Garcia. Nilmt techniques for intelligent home energy management and ambient assisted living: A review. *Energies*, 12(11):2203, 2019.
- [11] Nasrin Sadeghianpourhamami, Joeri Ruyssinck, Dirk Deschrijver, Tom Dhaene, and Chris Develder. Comprehensive feature selection for appliance classification in nilm. *Energy and Buildings*, 151:98–106, 2017.
- [12] Sense. Sense. <https://sense.com/>.
- [13] K. Yumak and O. Usta. A controversial issue: Power components in nonsinusoidal single-phase systems. In *2011 7th International Conference on Electrical and Electronics Engineering (ELECO)*, pages 1–157–161, Dec 2011.
- [14] Michael Zeifman and Kurt Roth. Nonintrusive appliance load monitoring: Review and outlook. *IEEE transactions on Consumer Electronics*, 57(1):76–84, 2011.
- [15] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

APPENDIX

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 127, 8)	40
flatten_1 (Flatten)	(None, 1016)	0
dense_3 (Dense)	(None, 1016)	1033272
dense_4 (Dense)	(None, 128)	130176
dense_5 (Dense)	(None, 1016)	131064
reshape_1 (Reshape)	(None, 127, 8)	0
zero_padding1d_1	(None, 130, 8)	0
conv1d_3 (Conv1D)	(None, 130, 1)	33
Total params: 1,294,585		
Trainable params: 1,294,585		
Non-trainable params: 0		

Fig. 8. Autoencoder for kettle (window length 130).

Layer (type)	Output Shape	Param #
conv1d_4 (Conv1D)	(None, 127, 16)	80
conv1d_5 (Conv1D)	(None, 124, 16)	1040
flatten_2 (Flatten)	(None, 1984)	0
dense_6 (Dense)	(None, 4096)	8130560
dense_7 (Dense)	(None, 3072)	12585984
dense_8 (Dense)	(None, 2048)	6293504
dense_9 (Dense)	(None, 512)	1049088
dense_10 (Dense)	(None, 3)	1539
Total params: 28,061,795		
Trainable params: 28,061,795		
Non-trainable params: 0		

Fig. 9. Rectangles network for kettle.

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 127, 16)	208
conv1d_3 (Conv1D)	(None, 124, 16)	1040
flatten_1 (Flatten)	(None, 1984)	0
dense_3 (Dense)	(None, 4096)	8130560
dense_4 (Dense)	(None, 3072)	12585984
dense_5 (Dense)	(None, 2048)	6293504
dense_6 (Dense)	(None, 512)	1049088
dense_7 (Dense)	(None, 3)	1539
Total params: 28,061,923		
Trainable params: 28,061,923		
Non-trainable params: 0		

Fig. 10. High frequency rectangles network for kettle.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 127, 8)	104
flatten (Flatten)	(None, 1016)	0
dense (Dense)	(None, 1016)	1033272
dense_1 (Dense)	(None, 128)	130176
dense_2 (Dense)	(None, 1016)	131064
reshape (Reshape)	(None, 127, 8)	0
zero_padding1d	(None, 130, 8)	0
conv1d_1 (Conv1D)	(None, 130, 1)	33
Total params: 1,294,649		
Trainable params: 1,294,649		
Non-trainable params: 0		

Fig. 11. High frequency autoencoder for kettle.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 127, 8)	40
conv1d_1 (Conv1D)	(None, 124, 8)	264
flatten (Flatten)	(None, 992)	0
dense (Dense)	(None, 1016)	1008888
dense_1 (Dense)	(None, 254)	258318
dense_2 (Dense)	(None, 13)	3315
dense_3 (Dense)	(None, 254)	3556
dense_4 (Dense)	(None, 1016)	259080
reshape (Reshape)	(None, 127, 8)	0
zero_padding1d	(None, 130, 8)	0
conv1d_2 (Conv1D)	(None, 130, 1)	33

=====
 Total params: 1,533,494
 Trainable params: 1,533,494
 Non-trainable params: 0

Fig. 12. "Big" autoencoder for kettle.

TABLE XI
VALIDATION AUC FOR ALL TRAINED MODELS.

Elec	Model	Freq	Run	Adam .0005	Adam 0.001	Adam 0.002	AMax 0.0005	AMax 0.001	AMax 0.002	Best
dish	rectangulos	data	If_0	0.9863	0.9914	0.9692	0.9875	0.9881	0.977	0.9914
dish	rectangulos	data	If_syn_0	0.9806	0.9794	0.9775	0.9777	0.9753	0.9723	0.9806
dish	rectangulos	data_hf	hf_0	0.9778	0.9739	0.9708	0.9893	0.9825	0.9491	0.9893
dish	autoencoder	data	ae>If_run	0.9763	0.5	0.5	0.9846	0.9793	0.5	0.9846
dish	autoencoder	data	ae_big>If_run	0.9752	0.5	0.5	0.9811	0.5	0.9968	0.9968
dish	autoencoder	data	ae>If_syn_run	0.9752	0.9799	0.5	0.9886	0.9737	0.5	0.9886
dish	autoencoder	data_hf	ae_hf_run	0.977	0.9753	0.5	0.9412	0.9661	0.5	0.977
Fridge	rectangulos	data	If_0	0.8001	0.5	0.5	0.8102	0.7953	0.8139	0.8139
fridge	rectangulos	data	If_syn_0	0.811	0.5	0.5	0.8037	0.809	0.8028	0.811
fridge	rectangulos	data_hf	hf_0	0.8689	0.6374	0.5	0.9012	0.868	0.8797	0.9012
fridge	autoencoder	data	ae>If_run	0.5	0.5	0.5	0.5	0.5	0.5	0.5
fridge	autoencoder	data	ae_big>If_run	0.5	0.5	0.5	0.5	0.5	0.5	0.5
fridge	autoencoder	data	ae>If_syn_run	0.5	0.5	0.5	0.5	0.5	0.5001	0.5001
fridge	autoencoder	data_hf	ae_hf_run	0.5	0.5	0.5	0.8312	0.4998	0.5	0.8312
kettle	rectangulos	data	If_0	0.9748	0.9767	0.5003	0.9707	0.9741	0.9686	0.9767
kettle	rectangulos	data	If_syn_0	0.9695	0.9757	0.5	0.9783	0.9759	0.9815	0.9815
kettle	rectangulos	data_hf	hf_0	0.9652	0.9714	0.9529	0.9642	0.9673	0.9555	0.9714
kettle	autoencoder	data	ae>If_run	0.9778	0.9769	0.9792	0.9791	0.9772	0.9781	0.9792
kettle	autoencoder	data	ae_big>If_run	0.9754	0.9726	0.9646	0.9779	0.9795	0.971	0.9795
kettle	autoencoder	data	ae>If_syn_run	0.9772	0.9768	0.9745	0.9768	0.9765	0.9755	0.9772
kettle	autoencoder	data_hf	ae_hf_run	0.9763	0.984	0.979	0.9762	0.9759	0.9794	0.984
microwave	rectangulos	data	If_0	0.8999	0.9205	0.5	0.9277	0.9331	0.9231	0.9331
microwave	rectangulos	data	If_syn_0	0.9125	0.9329	0.5221	0.9357	0.9368	0.912	0.9368
microwave	rectangulos	data_hf	hf_0	0.9203	0.919	0.5007	0.9267	0.9247	0.9122	0.9267
microwave	autoencoder	data	ae>If_run	0.934	0.9315	0.9256	0.9358	0.9339	0.9263	0.9358
microwave	autoencoder	data	ae_big>If_run	0.9126	0.9226	0.5	0.9321	0.932	0.9233	0.9321
microwave	autoencoder	data	ae>If_syn_run	0.9316	0.9337	0.934	0.9439	0.9318	0.9314	0.9439
microwave	autoencoder	data_hf	ae_hf_run	0.942	0.9492	0.9374	0.5	0.9442	0.9407	0.9492
washing	rectangulos	data	If_0	0.883	0.9002	0.8817	0.8893	0.8582	0.9037	0.9037
washing	rectangulos	data	If_syn_0	0.8823	0.8658	0.5	0.9004	0.8867	0.8919	0.9004
washing	rectangulos	data_hf	hf_0	0.9252	0.9176	0.8893	0.9509	0.9419	0.9362	0.9509
washing	autoencoder	data	ae>If_run	0.8774	0.8709	0.5	0.8774	0.5	0.5	0.8774
washing	autoencoder	data	ae_big>If_run	0.5	0.5	0.5	0.8496	0.5	0.5	0.8496
washing	autoencoder	data	ae>If_syn_run	0.8633	0.8746	0.5	0.8682	0.4983	0.5	0.8746
washing	autoencoder	data_hf	ae_hf_run	0.8761	0.5	0.5	0.49	0.494	0.5	0.8761