Household Energy Disaggregation Based on Pattern Consumption Similarities

Juan Chavat[(✉)] , Jorge Graneri, and Sergio Nesmachnow

Universidad de la República, Montevideo, Uruguay
{juan.pablo.chavat,jgraneri,sergion}@fing.edu.uy

Abstract. Non-intrusive load monitoring allows breaking down the aggregated household consumption into a detailed consumption per appliance, without installing extra hardware, apart of a smart meter. Breakdown information is very useful for both users and electric companies, to provide an accurate characterization of energy consumption, avoid peaks, and elaborate special tariffs to reduce the cost of the electricity bill. This article presents an approach for energy consumption disaggregation in residential households, based on detecting similar patterns of recorded consumption from labeled datasets. The proposed algorithm is evaluated using four different instances of the problem, which use synthetically generated data based on real energy consumption. Each generated dataset normalize the consumption values of the appliances to create complex scenarios. The `nilmtk` framework is used to process the results and to perform a comparison with two built-in algorithms provided by the framework, based on combinatorial optimization and factorial hidden Markov model. The proposed algorithm was able to achieve accurate results, despite the presence of ambiguity between the consumption of different appliances or the difference of consumption between training appliances and test appliances.

1 Introduction

Electricity utilization in homes has shown an uninterrupted increase worldwide, as detailed in the World Energy Outlook report, prepared by the International Energy Agency [6]. The electric power demanded in 2050 is expected to be twice as much as that demanded in 2010 [11]. Under this premise, many investigations have been carried out to achieve an efficient use of electricity in factories, buildings, and homes.

One of the approaches implemented to achieve a more efficient use of electric energy in homes is based on encouraging users to have a behavior change, favorable to saving. The incentives for behavioral changes are derived from the analysis of electricity utilization. For this analysis, Non Intrusive Load Monitoring (NILM) techniques are applied.

NILM allows determining the energy consumption of individual devices that are turned on and off, based on the detailed analysis of the current and voltage of the total load, measured at the interface with the source of the load.

This approach was developed to simplify the collection of energy consumption data by utilities, but it also has other applications. It is called non-intrusive to contrast it with techniques previously used to collect load data, which requires placing sensors on every appliance and, therefore, an intrusion on the user's energy consumption. In particular, NILM techniques are applied in residential households.

NILM uses only the aggregate signal to disaggregate the signal of each appliance, providing an easier way of generating detailed information about household energy consumption. The disaggregated information is useful to provide breakdown bill information to the consumer, schedule the activation of appliances, detect malfunctioning, and suggest actions that can lead a significant reduction in consumption (e.g., up to about 20% in some cases [12]), among other uses.

In this line of work, this article presents a first approach for solving the dissagregation problem by applying a simple algorithm for recognizing on/off appliances states using the aggregate consumption signal, and determine energy consumption patterns. The experimental evaluation of the proposed algorithm is performed over synthetic datasets, specifically built using real energy consumption data from the well-known UK-DALE repository [8]. Experiments are set to analyze the accuracy of the method varying the power consumption of appliances varies and generating complex scenarios including ambiguities between the power consumption of appliances. Experimental results are compared with two built-in methods of the `nilmtk` toolkit: Combinatorial Optimization (CO) and Factorial Hidden Markov Model (FHMM). Results shows that the proposed algorithm is able to achieve accurate results, accounting for an average of 0.95 on the F-score metric, in the most complex problem instances.

The proposal is developed within the project "Computational intelligence to characterize the use of electric energy in residential customers", funded by the National the Uruguayan government-owned power company (UTE) and Universidad de la República, Uruguay. The project proposes the application of computational intelligence techniques for processing household electricity consumption data to characterize energy consumption, determine the use of appliances that have more impact on total consumption, and identify consumption patterns in residential customers. The main contribution of this article is a simple approach to solve the problem of energy consumption dissagregation in residential households, conceived to be adapted to the main features of the Uruguayan system, and the experimental evaluation over a set of problem instances and the comparison with existing techniques.

The article is structured as follows. Section 2 presents the formulation of the problem addressed in the work. A review of the main related work is presented in Sect. 3. The proposed algorithms for solving the problem are described in Sect. 4. The experimental analysis is reported in Sect. 5. Finally, Sect. 6 presents the conclusions and the main lines of future work.

2   The Energy Consumption Dissagregation Problem

The problem consists of dissagregating the overall energy consumption of a house into the individual consumption of a number of appliances.

Consider a set of appliances available in a house $A = \{a_i\}, i = 1, \ldots, m$, and let x_t be the aggregate power consumption of the house at a given time slice t. x_t can be expressed as the sum of the individual power consumption x_t^i of each appliance in use in that time slice. The status of each time slice is indicated by the binary variable y_t^i, that takes value 1 when appliance i is ON and 0 when it is OFF. The simplest (binary) variant of the problem assumes just two possible values for the power consumption of each appliance, i.e., $x_t^i = c_i \times y_t^i$, that is to say that the power consumption of appliance i is constant and does not depend on the activity being performed by the appliance.

The total power consumption is described as a function $f \colon \{0, 1\}^m \to R$ defined by the expression in Eq. 1.

$$x_t = f((y_t^1, y_t^2, \cdots, y_t^m)) = c_1 y_t^1 + c_2 y_t^2 + \cdots + c_m y_t^m \tag{1}$$

If function f is injective (one-to-one), the problem is trivial. Otherwise, the times series $\{x_t\}_{t \in T}$ must be studied in order to deduce from the variation of power consumption on time, the signatures of the individual appliances.

For instance, suppose the appliances are: fridge (power consumption 250 W), washing machine (2000 W), dish washer (2500 W), kettle (2500 W), and home theater (80 W). The aggregate power consumption is a non-injective function. There is ambiguity between the power consumption of dish washer and kettle, as defined by Eq. 2. The variation of the aggregate power consumption in time must be studied to deduce if the kettle or the dish washer is ON.

$$f((0, 0, 1, 0, 0)) = f((0, 0, 0, 1, 0)) = 2500 \tag{2}$$

Several attributes and patterns can be studied to solve ambiguities. In the previous example, additional information can be used to solve the ambiguity: e.g., the mean time of utilization of each appliance (it is a couple of minutes for kettle and longer than an hour for the dish washer). Other more sophisticated patterns can be detected to solve problem instances with more complex ambiguities.

3   Related Works

The analysis of the related literature allows identifying several proposals on the design and application of software-based methods for energy consumption dissagregation. The main related works are reviewed next.

Hart [5] presented the concept of Nonintrusive Appliance Load Monitoring (NALM). The author stated that the previously presented approaches on the subject had a strong hardware component, installing intrusively monitoring

points in each household appliance connected to a central information collector. Hart proposed an approach based on using a simple hardware and complex software for the analysis, thus eliminating permanent intrusion in homes.

The model proposed by Hart considers that electrical appliances are connected in parallel to the electrical network and that the power consumed is additive (Eq. 3), where $a_i(t)$ represents the ON/OFF state of an appliance at time t.

$$a_i(t) = \begin{cases} 1 \text{ if appliance } i \text{ is ON at time } t \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

Multiphase loads with p phases are modeled as vectors of dimension p where each component is the load in each phase. The total charge of the vector is the sum of the p components. P_i is defined as a vector representing the power consumed by device i when it is turned on (Eq. 4), where $P(t)$ is the corresponding to time t, and $e(t)$ represents the noise or the recorded error for time t.

$$P(t) = \sum_{i=1}^{n} a_i(t).P_i + e(t) \tag{4}$$

The proposed model involves solving a combinatorial optimization problem to determine vector $a(t)$ from P_i and $P(t)$, in order to minimize the error (Eq. 5).

$$\hat{a}(t) = \arg\min_a \left| P(t) - \sum_{i=1}^{n} a_i(t).P_i \right| \tag{5}$$

However, the resulting combinatorial optimization problem is NP-hard and therefore computationally intractable for large values of n. Heuristic algorithms allow computing solutions of acceptable quality, but their applicability is limited because in practice the set of vectors P_i is not fully known, the value n is not fixed, and unknown devices tends to be described as a combination of those already known. Furthermore, a small variation in the measurement of $P(t)$ can cause large changes in $a(t)$, mistakenly predicting simultaneous on and off events.

In recent works, NILM has been treated as a machine learning problem, applying supervised and unsupervised learning methods. Supervised learning approach is based on a data set of the consumption of each circuit device and the aggregate signal, and the objective is to generate models that learn to disaggregate the signal of the devices from the added signal. The techniques most commonly applied in this approach are Bayesian learning and neural networks. The unsupervised approach seeks to learn signatures of possible devices from the aggregate signal without knowing *a priori* what devices are inside the circuit. Bonfigli et al. [2] presented a survey of the test data sets available to researchers and the main techniques used for the unsupervised NILM approach. The most used unsupervised learning techniques are those based on Hidden Markov Models (HMM), which define a number of hidden states in which the model can be moved, representing the operating conditions of the device (e.g., on, off and possible intermediate states) and an observable result, which depends on the real state that represents the analyzed consumption data.

Kelly and Knottenbelt [7] analyzed three deep neural networks for disaggregation in the NILM problem. The proposed neural networks had between one and 150 million trainable parameters, so large amounts of training data was needed. The data set used was UK-DALE. The approach consisted of training a neural network for each household appliance, taking as input a sequence of aggregate total consumption and returning as a result the prediction of the power demanded by the associated appliance. Three architectures of neural networks were studied: (i) long short-term memory (LSTM) recurrent neural network, suitable for working with data sequences because of its ability to associate the entire history of the inputs to an output vector; (ii) a self-coding for noise elimination (denoising autoencoder, dAE) that cleans the aggregate consumption signal to obtain only that corresponding to the target appliance; and (iii) a rectangle network to detect the start and end of the use of the target appliance, and its average power demanded at that time. The networks were trained using 50% of real data and 50% of synthetic data, generated with the signatures of the UK-DALE appliances using the nilmtk tool. Results were compared with CO and FHMM. The dAE and the rectangle networks outperformed the results of both CO and FHMM in F1 score, precision, proportion of total energy correctly assigned, and mean absolute error; while LSTM outperformed CO and FHMM in on/off appliances but was behind in multi-state appliances.

Several related works have used the `nilmtk` tool [1], a framework for NILM analysis implemented in Python that facilitates using multiple data sets by converting them to a standard data model. `nilmtk` implements algorithms for data preprocessing, statistics to describe the data sets, two disaggregation algorithms (CO and FHMM), and metrics for evaluation. Within the preprocessing algorithms are *downsample*, to normalize the frequency of consumption signals; and *voltage normalization*, to solve the problem of the variation of voltage between different countries [5], which implements a method to normalize the data and is able to combine different sets of household data from different countries.

Kolter and Johnson [10] introduced the REDD dataset and studied the performance of a FHMM algorithm for dissaggregation using the available data. FHMM was evaluated using two weeks of data from five households, subsampled in ten-second intervals. Results showed that FHMM was able to disaggregate the total consumption, observing a clear degradation of the results when going from the prediction in the training set to the prediction in the evaluation set. The FHMM for the training set correctly classified 64.5% of the consumption, while for the evaluation set the correct classification was reduced to 47.7%. The authors posed the challenge of finding a way to combine REDD with the massive amount of untagged data generated daily by public energy service companies.

4   The Proposed Algorithm

This section describes the proposed algorithm to solve the problem of energy consumption disaggregation based on similar consumption patterns.

4.1   Algorithm Description

Function $f : \{0,1\}^m \rightarrow R$ gives the aggregate power consumption of a house for a set of appliances. A function $g : R^{2d+1} \rightarrow R^m$ is considered, where the positive number d determines a time neighbourhood for the predictions (Eq. 6).

$$(\hat{y}_t^1, \hat{y}_t^2, \cdots, \hat{y}_t^m) := g_{W,Z}(x_{t-d}^i, \cdots, x_t^i, \cdots, x_{t+d}^i) \tag{6}$$

In Eq. 6, $(\hat{y}_t^1, \hat{y}_t^2, \cdots, \hat{y}_t^m)$ is the estimated configuration of the set of house appliances. Function $g_{W,Z}$ has random elements; it is defined using the information of a training database $\{W, Z\} = \{w_t, z_t\}$ such that for $t = 1, \cdots, n$, $w_t \in \{0,1\}^m$, $z_t \in R$ and Eq. 7 holds.

$$z_t = f((w_t^1, w_t^2, \cdots, w_t^m)) \tag{7}$$

The parameters of function $g_{W,Z}$ are chosen empirically to maximize the sum given in Eq. 8, where A is the set of ambiguous configurations $A = \{y \in \{0,1\}^m / \exists y' \in \{0,1\}^m, y' \neq y, f(y') = f(y)\}$. This is equivalent to maximize the number of time slices $t \in T$ for which every appliance status is correctly detected.

$$\sum_{y_t \in A} \prod_{i=1}^m 1_{\{\hat{y}_t^i = y_t^i\}} \tag{8}$$

The proposed algorithm, named *Pattern Similarities* (PS), consists of two parts, training and testing (prediction), which are described next.

The output of the algorithm is y, the vector of disaggregated power consumption, computed using the following input:

– The vector x containing the aggregate power consumption of one house measured over a period of time with a certain time frequency.
– A training set z containing the aggregate power consumption of one or several houses measured over a period of time with the same time frequency as x.
– A training set w containing the disaggregated power consumption of the house (houses) described in z over the same period of time and with the same frequency as x is measured.
– The parameter δ that defines a power consumption neighbourhood.
– The parameter d that defines a time interval neighbourhood.
– The parameter H that separates high from low power consumption.

Algorithm 1 describes the processing on the training stage. The goal is to build an array (M_Z) with information relating each consumption record with its neighbour records. The information act as a feature of each appliance signature, for each sample. The main loop (lines 2–10) iterates over each sample in the training set. In each iteration step, the algorithm checks if the neighbour samples has similar consumption values to the currently analyzed sample (lines 4–8); if they have, then a counter is incremented. In the end, the array with the processed values is generated for each testing sample. That array is used in the testing stage to find samples whose consumption is similar to the sample being processed.

Algorithm 1. PS algorithm: training stage

1: $M_Z \leftarrow$ array of lenght Z
2: **for all** $z_i \in Z$ **do**
3:    counter $\leftarrow 0$
4:    **for all** $\{z_j \in Z : |j - i| < d\}$ **do**
5:       **if** $z_j > z_i - \varphi$ **then**
6:          counter $\leftarrow$ counter $+ 1$
7:       **end if**
8:    **end for**
9:    $M_Z[i] \leftarrow$ counter
10: **end for**

Algorithm 2. PS algorithm: testing stage

1: $M_X \leftarrow$ array of lenght X
2: **for all** $x_i \in X$ **do**
3:    counter $\leftarrow 0$
4:    **for all** $\{x_j \in X : |j - i| < d\}$ **do**
5:       **if** $x_j > x_i - \varphi$ **then**
6:          counter $\leftarrow$ counter $+ 1$
7:       **end if**
8:    **end for**
9:    $M_X[i] \leftarrow$ counter
10: **end for**
11: **for all** $x_i \in X$ **do**
12:    $I \leftarrow \emptyset$
13:    **for all** $z_j \in Z$ **do**
14:       **if** $|z_j - x_i| \leq \delta$ **and** $x_i > H$ **then**
15:          $I \leftarrow I \cup \{j\}$
16:       **end if**
17:       **if** $|I| \geq 1$ **then**
18:          $J \leftarrow \text{argmin}\{|M_Z(I(\cdot)) - M_X(i)|\}$
19:       **else**
20:          $J \leftarrow \text{argmin}\{|z(\cdot) - x(i)|\}$
21:       **end if**
22:       $k \leftarrow rand\{1, \dots, length(J)\}$
23:       $y(i, \cdot) \leftarrow w(I(J(k)), \cdot)$
24:    **end for**
25: **end for**

Algorithm 2 presents the testing stage. The first loop (lines 1–10) is similar to the main loop in the training stage, but applied to the testing dataset. This loop builds an array (M_X) with the processed value of signature feature for each testing sample. It is used to compare with the array built into the training stage. The second loop (lines 11–26) iterates over each testing sample to find similarities with the samples of the training dataset. In line 13, each training sample is compared to the consumption of the sample being processed, if the difference between both is lower than a threshold (δ) and the testing sample

have a consumption value greater than a minimum (H), it is added to set I, to be considered for next comparisons. If the set I is not empty, i.e., at least one training sample was found similar to the processing sample, the samples that minimize the difference between signature features (the difference between M_Z and M_X) are selected, and one of them is chosen randomly (line 18 and 22). If set I is empty, i.e., no training samples were found similar to the processing sample, the algorithm select the training samples that minimize the difference of consumption with the sample that is being processed, and one of them is chosen randomly (lines 20 and 22). Once the algorithm have found a similar training sample, it maps the consumption per appliance at the time of the training sample to the prediction results (line 23).

4.2   Implementation

A first version of the proposed algorithm was developed on Matlab, version 8.3.0.532 (R2014a), as a proof of concept. After that, it was re-implemented on python version 3, using `pandas` and `numpy`, which allows the implementation to be included as part of a pipe of execution in `nilmtk`. For this stage, several modifications were included in the metrics and utils files of the framework.

Two scripts were implemented for generating the synthetic datasets. The first script reads the UK-DALE dataset (HDF5 file), normalizes the values for the indicated houses and appliances, and builds a directory structure that contains metadata and the normalized data in CSV files. The normalization replaces all records over a given threshold by an indicated value, and set all other values to zero. The second script reads the directory structure and its content to generate a new HDF5 file with the synthetic dataset. In the resulting dataset, data have the same sample rate than in the original dataset, with the particularity that it does not present gaps, i.e., if original sample rate is six seconds, the generated dataset will have a record each six seconds. The gaps presented in the original dataset are filled by zeros. The algorithm implementation, the scripts for generating the datasets and the modified nilmtk files are available on a public repository (https://gitlab.com/jpchavat/nilm-scripts).

5   Experimental Analysis

This section presents the experimental analysis of the proposed algorithm. In the experiments, the algorithm was executed in a `nilmtk` pipeline of execution, using a synthetic dataset based on UK-DALE dataset as input. Results were compared with CO and FHMM algorithms executed in same settings.

5.1   Problem Instances and Datasets

The synthetic datasets used for the experiments are based on house #1 of the UK-DALE dataset, considering the following appliances: fridge, washing

machine, kettle, dishwasher, and home theatre. These appliances are representative of devices that contribute the most to household energy consumption [14].

Four different instances were generated for the experimental analysis. All datasets were generated by downsampling the UK-DALE dataset period to 5 min. A *datetime* range limit was established for training and testing data. For training data, the limits were set from 2013-01-01 at 00:00:00 to 2013-07-01 at 00:00:00, while for the testing data the limits were set from 2013-07-01 at 00:00:00 to 2013-12-31 at 23:59:59. A threshold of minimum consumption was applied in the normalization, which was set to 5.0 W. This value allows discarding standby power consumption records. Instances were generated to analyze the efficacy of the proposed algorithm to solve different cases of energy consumption ambiguity. A description of each problem instance and the motivation of using it is provided next.

Instance #1. The generated dataset normalizes the consumption of each appliance using the median of maximum consumption per activation (i.e., periods of time in which an appliance remains in state ON). Outliers were filtered by lower and upper limits defined by the standard deviation. The generated dataset is used for training and testing. This instance aims at working with values close to the real ones but keeping constant consumption values over time.

Instance #2. The generated dataset normalizes the consumption values to generate ambiguity between the consumption of kettle and dish washer. The same dataset is used for training and testing the algorithms. This instance aims at testing how the algorithms solves the most basic case of ambiguity.

Instance #3. The dataset normalizes consumption values like instance #2, but including ambiguities between the sum of consumption of fridge, home theater, and washing machine with the consumption of the dish washer. The same dataset is used for training and testing the algorithms. This instance aims at studying how the algorithms solves a more sophisticated case of ambiguity.

Instance #4. The training dataset is the same than in instance #2; but a new dataset was generated for the testing step, introducing small variations in the consumption of every appliance, but the washing machine. For example, the consumption of the fridge was normalized to 260 instead of 250. This instance

Table 1. Normalized values per appliance for each instance

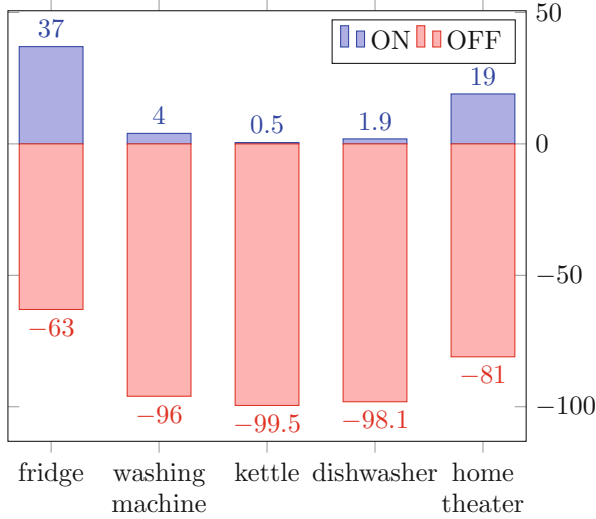| Instance	Appliance				
	Fridge	Washing machine	Kettle	Dishwasher	Home theater
#1 (testing, training)	117	3325	2390	2741	93
#2 (testing, training)	250	2000	2500	2500	80
#3 (testing, training)	300	1800	2200	2300	200
#4 (testing)	250	2000	2500	2500	80
#4 (training)	260	2000	2400	2600	70

Fig. 1. Percentage of operating time of each appliance

aims at testing the algorithm in an scenario where testing appliances are similar but not equal to the appliances used for the training.

Table 1 reports the normalized value of the datasets used for training and testing for each instance, and Fig. 1 shows the percentage of records when each appliance is in state ON/OFF, which is the same for all the generated datasets.

5.2   Software and Hardware Platform

The `nilmtk` framework was used to implement the pipeline of execution for the experiments, as described in Fig. 2.
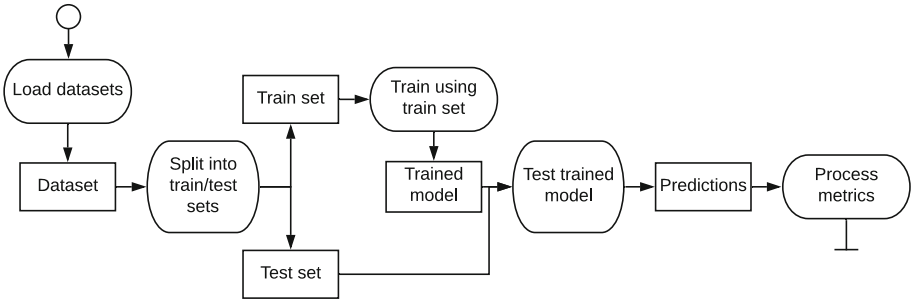

Fig. 2. Execution pipeline implemented in nilmtk

The first stage of the pipeline loads the dataset while the second splits the dataset into a training set and a testing set. The training set is used to train

each algorithm and after that, the testing set is used to obtain the results of dissagregation. Finally, results are compared with the ground truth data (i.e. the test set) to compute a set of metrics.

The experimental evaluation was performed on National Supercomputing Center (Cluster-UY) infrastructure that counts with Intel Xeon-Gold 6138 nodes (up to 1120 CPU cores), 3.5 TB RAM, and 28 GPU Nvidia Tesla P100, connected by a high-speed 10 Gbps Ethernet network (cluster.uy) [13].

5.3   Baseline Algorithms for Comparison

Two methods from the related literature were considered as baseline for the comparison of the results obtained by the proposed algorithm: CO and FHMM.

The CO method was first presented by Hart [5], and included in the `nilmtk` framework. The approach of CO is to find the optimal combination of appliance states that minimises the difference between the total sum of aggregated consumption and the sum of the consumption of the predicted state on of appliances. CO searches for a vector $\hat{a}$ that minimises the expression on Eq. 5 Given the complexity of the CO algorithm, which is exponential in the number of appliances, it is not useful to address scenarios with a large number of appliances. The complexity of the CO algorithm is exponential in the number of appliances. Thus, it is not useful to address scenarios with a large number of appliances.

FHMM was introduced by Gharamani and Jordan [4]. Different variations of the original method were developed by Kim et al. [9] to solve the disaggregation problem. HMM are mixture models that encode historical information of a temporal series in a unique multinomial variable, represented as a hidden state; FHMM extends HMM to allow modeling multiple independent hidden state sequences simultaneously. FHMM scales worst than CO in scenarios with a large number of appliances.

5.4   Metrics for Results Evaluation

Standard metrics were applied to evaluate the efficacy of the studied algorithms. Let $x_i^{(n)}$ be the actual status series for appliance n and $\hat{x}_i^{(n)}$ the status predicted by the algorithm, True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) ratios are defined by Eqs. 9–12.

$$TP = \sum_i AND(x_i^{(n)} = 1, \hat{x}_i^{(n)} = 1) \tag{9}$$

$$FP = \sum_i AND(x_i^{(n)} = 0, \hat{x}_i^{(n)} = 1) \tag{10}$$

$$TN = \sum_i AND(x_i^{(n)} = 0, \hat{x}_i^{(n)} = 0) \tag{11}$$

$$FN = \sum_i AND(x_i^{(n)} = 1, \hat{x}_i^{(n)} = 0) \tag{12}$$

Five metrics are considered in the analysis:

– *precision* of the prediction, defined as an estimator of the conditional probability of predicting ON given that the appliance is ON (Eq. 13).
– *recall*, defined as the conditional probability that the appliance is ON given that the prediction is ON (Eq. 14).
– *F–Score*, defined as the harmonic mean of precision and recall (Eq. 15).
– *Error in Total Energy Assigned* (TEE), defined as the error of the total assigned consumptions (Eq. 16).
– *Normalized Error in Assigned Power* (NEAP), defined as the mean normalized error in assigned consumptions (Eq. 17).

$$precision = \frac{TP}{TP + FN} \tag{13}$$

$$recall = \frac{TP}{TP + FP} \tag{14}$$

$$F\text{–}Score = \frac{2 \times precision \times recall}{precision + recall} \tag{15}$$

$$\text{TEE}^{(n)} = \left| \sum_t y_t^{(n)} - \sum_t \hat{y}_t^{(n)} \right| \tag{16}$$

$$\text{NEAP}^{(n)} = \frac{\sum_t \left| y_t^{(n)} - \hat{y}_t^{(n)} \right|}{\sum_t y_t^{(n)}} \tag{17}$$

5.5 Results

Tables 2, 3, 4 and 5 report the results of the proposed algorithm (PS) and the baseline algorithms (CO and FHMM), on instances #1 to #4. All results were obtained using the following parameter configuration, set by a rule-of-thumb and empirical evaluation: $\delta = 100$, $d = 10$, $H = 500$ and $\varphi = 250$.

Results in Table 2 indicate that *PS* was able to accurately solve problem instances without ambiguity between power consumption of appliances. F-score values between 0.92 and 1.0 were obtained. Both *CO* and *FHMM* got F-score values around 0.6 for fridge and washing machine, around 0.3 for dish washer and home theater, and 0.04 (i.e., almost null) for kettle. In all cases, F-score values were lower than the obtained with *PS*.

Results in Table 3 indicate that F-score values of *PS* for appliances with ambiguities decreased up to 9%, while the rest of the F-score values remains similar to instance #1. Regarding the baseline algorithms, *CO* showed a decrease of 50% in the prediction of appliances with ambiguity, while results of *FHMM* remained similar to the ones computed for instance #1, with exception of the kettle (F-score decreased 66%).

Results in Table 4 indicates that the F-score values of *PS* decreased for washing machine (3%), dish washer (6%), and kettle (the worst value, 25% less than

Table 2. Results of CO, FHMM, and PS on instance #1

CO

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	292.18	2318.84	2767.39	6651.69	1118.64
NEAP	0.8663	0.7644	5.9284	2.6279	2.1975
Precision	0.8324	0.9863	0.7153	0.9758	0.8413
Recall	0.5584	0.4827	0.0228	0.2301	0.2814
F-score	0.6684	0.6481	0.0442	0.3724	0.4218

FHMM

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	306.46	3209.08	3399.42	5371.37	948.72
NEAP	0.8843	0.8367	6.8117	2.7134	2.3119
Precision	0.7576	0.9817	0.7810	0.9768	0.5799
Recall	0.5408	0.5078	0.0258	0.2377	0.2199
F-score	0.6311	0.6694	0.0500	0.3823	0.3188

PS

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	23.87	0.00	0.00	0.00	29.67
NEAP	0.0218	0.0000	0.0000	0.0000	0.1497
Precision	0.9839	1.0000	1.0000	1.0000	0.9409
Recall	0.9942	1.0000	1.0000	1.0000	0.9121
F-score	0.9891	1.0000	1.0000	1.0000	0.9263

Table 3. Results of CO, FHMM, and PS on instance #2

CO

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	2228.32	1701.36	5595.52	7206.13	685.29
NEAP	1.0053	1.5412	9.8478	3.0491	1.6285
Precision	0.6973	0.8271	0.6715	0.9807	0.7781
Recall	0.5123	0.2457	0.0111	0.1184	0.2907
F-score	0.5907	0.3789	0.0219	0.2113	0.4233

FHMM

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	1401.84	962.88	7904.24	5016.79	431.27
NEAP	0.9007	1.1175	13.2448	2.1841	1.7049
Precision	0.7687	0.9149	0.7007	0.9787	0.6649
Recall	0.5573	0.4790	0.0084	0.2379	0.2850
F-score	0.6461	0.6288	0.0166	0.3828	0.3990

PS

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	0.00	0.00	42.50	42.50	14.88
NEAP	0.0000	0.0000	0.1788	0.0473	0.1264
Precision	1.0000	1.0000	0.9416	0.9681	0.9460
Recall	1.0000	1.0000	0.8866	0.9843	0.9289
F-score	1.0000	1.0000	0.9133	0.9761	0.9374

Table 4. Results of CO, FHMM, and PS on instance #3

CO

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	1690.42	2194.13	6298.06	6720.05	949.90
NEAP	0.9386	1.6483	12.1919	3.0818	1.7343
Precision	0.8217	0.8678	0.5876	0.9826	0.8432
Recall	0.5754	0.2400	0.0073	0.1212	0.3250
F-score	0.6768	0.3760	0.0145	0.2157	0.4692

FHMM

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	2069.24	1655.52	6273.13	6895.43	1561.12
NEAP	1.1036	1.2927	12.1388	3.1483	2.0024
Precision	0.4318	0.9067	0.6387	0.9797	0.7645
Recall	0.4512	0.3677	0.0087	0.1380	0.2942
F-score	0.4413	0.5232	0.0171	0.2419	0.4249

PS

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	4.50	82.80	15.40	89.70	13.60
NEAP	0.0221	0.0668	0.5000	0.1092	0.0377
Precision	0.9893	0.9771	0.7372	0.9266	0.9845
Recall	0.9886	0.9570	0.7566	0.9629	0.9780
F-score	0.9889	0.9670	0.7468	0.9444	0.9812

Table 5. Results of CO, FHMM, and PS on instance #4

CO

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	2543.42	2239.86	5208.68	7414.75	637.84
NEAP	0.9819	1.7824	9.6185	3.0921	1.8408
Precision	0.6597	0.7653	0.6533	0.9826	0.7895
Recall	0.5202	0.1823	0.0121	0.1193	0.3205
F-score	0.5817	0.2944	0.0238	0.2128	0.4559

FHMM

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	1829.65	699.35	8567.56	5080.58	560.53
NEAP	0.9218	1.1591	14.6967	2.2034	1.9148
Precision	0.7209	0.8403	0.6971	0.9797	0.6961
Recall	0.5453	0.4634	0.0083	0.2383	0.2931
F-score	0.6210	0.5974	0.0163	0.3834	0.4125

PS

Metric	Fridge	Washing machine	Kettle	Dishwasher	Home theater
TEE (kW)	182.69	62.00	42.60	111.00	145.00
NEAP	0.0440	0.0142	0.3221	0.0821	0.2720
Precision	0.9985	1.0000	0.8066	0.9758	0.9666
Recall	0.9957	0.9860	0.8984	0.9787	0.9166
F-score	0.9971	0.9930	0.8500	0.9773	0.9409

for instance #1), increased for home theater (6%), and did not vary for fridge. *CO* decreased for washing machine (42%), kettle (67%), and dish washer (42%), compared with instance #1. F-score values for FHMM decreased for all the appliances (up to 66% for kettle), but the home theater (increased 33%).

Finally, results in Table 5 demonstrate that *PS* has a robust behavior when using different normalized datasets for training and testing steps. The F-score for *PS* was over 0.99 for fridge and washing machine, over 0.97 for dish washer, and over 0.94 for home theater. The lowest F-score value was obtained for kettle (0.85) With respect to instance #1, the F-score of the kettle decreased 15%. The rest of the appliances experienced a decrease/increase lower than 2%. For *CO*, F-score values decreased for all appliances but the home theater For *FHMM*, F-score values of fridge and dish washer varied less than 1.6% with respect to instance #1, and decreased for washing machine and kettle (up to 67%).

Overall, the proposed *PS* algorithm achieved satisfactory results for all the studied instances. Improvements on F-score were 60% over *CO* and 57% over *FHMM* in average, and up to 64% over *CO* in problem instance #4 and up to 60% over *FHMM* in problem instance #3. Furthermore, *PS* systematically obtained the lowest values of both TEE and NEAP metrics for all instances. Degraded results obtained for kettle in problem instances with ambiguity suggest that the lower percentage of operating time (0.5% for kettle) affects the results negatively and the more complex the dataset is, the more consumption samples are needed in the testing dataset.

6   Conclusions and Future Work

This article presented an approach to address the problem of household energy disaggregation. An algorithm based on pattern similarities was proposed. The experimental evaluation performed over realistic problem instance showed that, overall, the proposed algorithm is effective for addressing the problem of energy consumption disaggregation. Results can be applied to household energy planning by using intelligent recommendation systems [3].

The main lines for future work are related to study instances with different sample rates and noise in the power consumption, and extend the parameter analysis of the proposed algorithm. In addition, more sophisticated computational intelligent methods can be evaluated to solve the problem.

References

1. Batra, N., et al.: NILMTK: an open source toolkit for non-intrusive load monitoring. In: 5th International Conference on Future Energy Systems, pp. 265–276 (2014)
2. Bonfigli, R., Squartini, S., Fagiani, M., Piazza, F.: Unsupervised algorithms for non-intrusive load monitoring: an up-to-date overview. In: 15th International Conference on Environment and Electrical Engineering (2015)

3. Colacurcio, G., Nesmachnow, S., Toutouh, J., Luna, F., Rossit, D.: Multiobjective household energy planning using evolutionary algorithms. In: Iberoamerican Congress on Smart Cities (2019)
4. Ghahramani, Z., Jordan, M.: Factorial hidden Markov models. In: Advances in Neural Information Processing Systems, pp. 472–478 (1996)
5. Hart, G.: Nonintrusive appliance load monitoring. Proc. IEEE **80**(12), 1870–1891 (1992)
6. International Energy Agency: World Energy Outlook 2015. White paper (2015)
7. Kelly, J., Knottenbelt, W.: Neural NILM: deep neural networks applied to energy disaggregation. In: 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, pp. 55–64 (2015)
8. Kelly, J., Knottenbelt, W.: The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. Sci. Data **2**, 150007 (2015)
9. Kim, H., Marwah, M., Arlitt, M., Lyon, G., Han, J.: Unsupervised disaggregation of low frequency power measurements. In: SIAM International Conference on Data Mining, pp. 747–758. SIAM (2011)
10. Kolter, J., Johnson, M.: REDD: a public data set for energy disaggregation research. In: Workshop on Data Mining Applications in Sustainability, pp. 59–62 (2011)
11. Larcher, D., Tarascon, J.: Towards greener and more sustainable batteries for electrical energy storage. Nat. Chem. **7**(1), 19–29 (2015)
12. Neenan, B., Robinson, J., Boisvert, R.: Residential electricity use feedback: a research synthesis and economic framework. Electric Power Research Institute (2009)
13. Nesmachnow, S., Iturriaga, S.: Cluster-UY: high performance scientific computing in Uruguay. In: International Supercomputing Conference in Mexico (2019)
14. Orsi, E., Nesmachnow, S.: Smart home energy planning using IoT and the cloud. In: IEEE URUCON (2017)