Mixed-Integer Nonlinear Programming for State-based Non-Intrusive Load Monitoring

Marco Balletti, Veronica Piccialli, and Antonio M. Sudoso

***Abstract*—Energy disaggregation, known in the literature as Non-Intrusive Load Monitoring (NILM), is the task of inferring the energy consumption of each appliance given the aggregate signal recorded by a single smart meter. In this paper, we propose a novel two-stage optimization-based approach for energy disaggregation. In the first phase, a small training set consisting of disaggregated power profiles is used to estimate the parameters and the power states by solving a mixed integer programming problem. Once the model parameters are estimated, the energy disaggregation problem is formulated as a constrained binary quadratic optimization problem. We incorporate penalty terms that exploit prior knowledge on how the disaggregated traces are generated, and appliance-specific constraints characterizing the signature of different types of appliances operating simultaneously. Our approach is compared with existing optimization-based algorithms both on a synthetic dataset and on three real-world datasets. The proposed formulation is computationally efficient, able to disambiguate loads with similar consumption patterns, and successfully reconstruct the signatures of known appliances despite the presence of unmetered devices, thus overcoming the main drawbacks of the optimization-based methods available in the literature.**

***Index Terms*—Automatic State Detection, Energy Disaggregation, Mathematical Programming, Non-Intrusive Load Monitoring.**

NOTATION

Throughout the paper, we denote vectors by boldface lowercase letters and matrices by boldface uppercase letters. We also denote by $\mathbf{1}_n \in \mathbb{R}^n$ the vector of all ones of size n.

Inputs:

N	Number of appliances.
S_i	Number of power states of the appliance i.
M	Number of training time periods.
V	Number of validation time periods.
T	Number of test time periods.
H	Number of time intervals in T.
$y_i(t)$	Active power of the appliance i at time t.
$\hat{y}_i(t)$	Estimated active power of the appliance i at time t.
$y(t)$	Total active power of the time t.
$\boldsymbol{z}_i(t)$	Historical observations of the appliance i up to time t.

Sets:

$\mathcal{N} = \{1, \dots N\}$	Set of all appliances.
$\mathcal{O} \subseteq \mathcal{N}$	Set of Type-IV appliances.
$\mathcal{F} \subseteq \mathcal{N}$	Set of Type-II appliances.
$\mathcal{S}_i = \{1, \dots S_i\}$	Set of states of the appliance i.
$\mathcal{M} = \{1, \dots M\}$	Set of training indices.
$\mathcal{V} = \{1, \dots V\}$	Set of validation indices.
$\mathcal{T} = \{1, \dots T\}$	Set of test indices.
$\mathcal{H} = \{1, \dots, H\}$	Set of interval indices.
$\mathcal{T}_h$	Set of indices $t \in \mathcal{T}$ related to the interval $h \in \mathcal{H}$.

Model Parameters:

$\boldsymbol{p}_i$	Power levels of the appliance i.
λ_1	Non-negative penalty parameter for temporal smoothness.
w_i	Non-negative weight for the piece-wise constant consumption profile of the appliance i.
λ_2	Non-negative penalty parameter for device sparsity.
l_i	Non-negative weights inversely proportional to the ON time of the appliance i.
$s_i(t)$	Probability that the appliance i is ON at time t.
a_{ij}, b_{ij}	Minimum and maximum active time for state j of the appliance i.
$\boldsymbol{U}_i, \boldsymbol{D}_i$	Indicator matrices defining the transition order between the states of the appliance i.
o_i	Maximum number of upward transitions for the appliance i.
m_{ih}	Maximum consumption allowed in the h-th time interval for the appliance i.

Variables:

$\boldsymbol{x}_i(t)$	State indicator variable of the appliance i.
$\boldsymbol{u}_i(t), \boldsymbol{d}_i(t)$	Indicator variables for upward and downward transitions of the appliance i at time t.
$\boldsymbol{\phi}_{ij}$	Autoregressive coefficients for state j of the appliance i.
$\delta_{ij}(t)$	Assignment of $\boldsymbol{z}_i(t)$ to the j-th submodel of the appliance i.

I. INTRODUCTION

Non-Intrusive Load Monitoring (NILM), or energy disaggregation, is the task of estimating the energy consumption of each appliance starting from the aggregate power signal recorded by a single meter [1]. Given the aggregate power consumption and a set of target appliances, a NILM system aims to identify at each time step the active appliances and their relative contribution to the power consumption. NILM is recognized as an important Smart Grid technology that provides energy breakdown information without the need of installing multiple monitoring devices at the appliance level. Energy saving is arguably the most popular application of the NILM service [2], [3]. It can be useful for increasing energy savings for both providers and residential users. Regarding the providers, detailed appliance usage information allow them to predict the energy demand, apply personalized management policies and service recommendations, as well as to promote future economic and environmental objectives. Residential users, on the other hand, can obtain more awareness of how much energy being spent on their appliances and can exploit this information to take the proper actions for reducing their bills.

Since Hart's seminal paper [1], numerous energy disaggregation algorithms have been proposed in the literature to improve his results. NILM approaches can be divided into event-based or state-based methods. Depending on whether the ground truth is available or not, NILM algorithms can be supervised or unsupervised [4]. Event-based approaches use edge detection techniques to capture statistically significant variations of the signal. Most common event-based approaches involve an unsupervised event detection of the aggregate signal and a supervised classification to assign the known appliances to the detected events. The classification tools that can be found in the literature are based on Support Vector Machines (SVM) [5], Decision Trees [6] and a combination of various classification methods [7]. In addition, clustering techniques based on Dynamic Time Warping (DTW) are used to identify windows with similar consumption patterns and to extract representative load signatures [8], [9]. Recent research on event-based NILM involves Graph Signal Processing (GSP) algorithms [4], [10], both supervised and unsupervised, where the spatio-temporal correlation of the data is found by embedding the signals onto a graph. Recently, multi-label classification algorithms employing time-domain and wavelet-domain features have been suggested (see [11] and reference therein).

State-based approaches, instead, consider each appliance as a finite-state machine and disaggregate the total power signal based on the appliance load distribution models. In order to explore all the possible combinations among the different appliances' state sequences, Combinatorial Optimization (CO) and Hidden Markov Models (HMM) [12], [13], [14] are commonly used among the state-based approaches. However, as the number of appliances increases, the number of combinations of states increases exponentially, increasing the problem complexity. Another limitation of these approaches is that they tend to produce unsatisfactory results in presence of unknown loads. In [15] differential HMMs are used with the expectation-maximization algorithm to generate state transition models in an unsupervised manner. In [16] sub-metering measurements are used to build super-state HMMs and the inference is performed through a sparse Viterbi algorithm resulting in an efficient estimation of the energy consumption. In [17] a non-negative matrix factorization technique is used to decompose the aggregated signal into appliance signatures.

Among the supervised methods, in the last few years, state-of-the-art performance has been achieved by using end-to-end Deep Neural Networks (DNN) [18], [19], [20], [21]. The main drawback of DNNs, it that they strongly rely on extended training samples of both aggregated and disaggregated data. To obtain good performance it is recommended to use measurements at the granularity of seconds which may be not available since data points at the resolution of minutes are more practical to be recorded by commercial smart meters [22]. Furthermore, they focus on training one model per appliance (one-to-one mapping), resulting that a complete NILM system should integrate as many models as the number of devices the target environment contains. Thus, these types of solutions are not directly applicable in real-time situations. On the other hand, by training one model for each target appliance, DNNs are able to identify the activations and estimate the consumption of the associated appliances regardless of the number of appliances or the proportion of power consumption from non-target appliances [20], [21].

A. Related Work on Optimization

NILM can be treated as a blind source separation problem, which tries to estimate the composition of the input from the output (one-to-many mapping). In this light, optimization-based approaches are state-based, so the studied appliances are assumed to have discrete states in their signatures [23]. Unlike machine learning research on NILM which mostly focuses on training one model per appliance, state-based optimization approaches can disaggregate multiple loads at the same time. In [1], Hart formulates the energy disaggregation problem as a combinatorial optimization (CO) problem. In his formulation, the disaggregated energy consumption is obtained by combining all the possible appliances' states so that the difference between the aggregated power and the sum of the appliance-level power is minimized. In addition to the high computational cost, the drawbacks of Hart's formulation are the possible confusion derived from loads with similar power levels, and the inability to correctly disaggregate the target appliances in presence of unknown appliances. In fact, if the measurement noise or the contribution of the unknown appliances is significant, the CO problem generates a poor solution where all the appliances are set to an active state most of the time.

Recently, various integer programming approaches have been proposed in the literature, but very few of them have attempted to deal with the challenges of Hart's formulation. Suzuki et al. [24] formulated the NILM problem as an Integer Linear Programming (ILP) problem where they used current waveforms from multiple loads. Egarter and Elmenreich [25] investigated the CO approach from a theoretical point of view and discussed the equivalency with the knapsack problem. They concluded that it is hard to disaggregate loads with similar consumption patterns and proposed as future work a multi-objective optimization approach. Bhotto et al. [26] proposed several enhancements to the ILP model in [24], such as always-on appliances and median filtering. Most of the enhancements in [26] were included as a pre-processing step rather than constraints. They also proposed a linear programming refinement to deal with the minimum and maximum transient spans of the power states. In [27], Piga et al. proposed instead a convex formulation with continuous variables and enforced sparsity by means of the Lasso penalty. In [28], Wittmann et al. proposed a mixed-integer programming model that exploits both active and reactive power. They added basic state machine constraints but their method is window-based and does not work on every instance of data. More recently, Zeinal-Kheiri et. al [29] enriched the integer programming model by enforcing the power profiles of the appliances to be piece-wise constant over time. The state transitions of each appliance are modeled with non-linear constraints. Clearly, their proposed model is an integer non-linear programming problem that is difficult to solve to global optimality as the number of appliances or the scheduling horizon increase. In addition to deterministic optimization approaches, derivative-free search strategies such as ant colony optimization [30], differential evolution [31] and genetic algorithms [32] have also been employed.

B. Our Contribution

According to [3], household appliances can be categorized on the basis of their operational states: Type-I (ON/OFF), Type-II (Finite State Machines), Type-III (Continuously Variable), or Type-IV (Always-On). In this paper, we are concerned about expanding the CO problem for energy disaggregation in order to disambiguate the signatures of different types of appliances and handle the major challenges that arise due to the presence of unknown loads. Results presented in most of the optimization-based literature are obtained on an artificial signal, created by summing up the plug level power measurements of the target appliances and by subtracting the contribution of unknown appliances. This makes results not suitable for real-world applications unless nearly all the devices connected to the household main meter are known. Therefore, in this paper, special emphasis is placed on load disaggregation in an environment where the unmetered part is typically plenty and highly variable.

We formulate the energy disaggregation problem as a least-square error minimization problem with regularization terms that promote sparsity and exploit prior knowledge on the disaggregated traces. In order to further reduce the computational complexity, we narrow down the feasible region by adding appliance-specific and signal aggregate constraints that characterize the signature of different types of appliances operating simultaneously.

The contributions of our paper are the following:

1) A novel penalty-based binary quadratic programming formulation with appliance-specific constraints is proposed to better disambiguate the load of similar appliances in presence of unknown loads.
2) A novel optimization-based automatic state detection algorithm is developed to estimate the power levels of the appliances and their respective transient behavior.
3) An automatic procedure is proposed for inferring the parameters of the optimization problem by using a small training set of disaggregated data.
4) The superiority of the approach with respect to previous works in the field of optimization-based methods for NILM is validated by extensive testing on different publicly available datasets.
5) The deployment of the proposed algorithm in a real-world environment is discussed and evaluated experimentally.

Our optimization-based load disaggregation algorithm consists of three steps. First, we learn the parameters of the optimization problem by using a very small training set of power consumption signatures from different devices. More in detail, the power states and their dynamics are obtained by efficiently finding a good locally optimal solution of a non-convex mixed-integer optimization problem that jointly clusters and estimates an autoregressive model for each power state. Then, taking as input the aggregated signal, we solve the binary quadratic disaggregation problem which incorporates different types of constraints and prior knowledge on the estimated consumptions. Finally, we post-process the output of the optimization problem to keep into account the transient spans of each appliance. Our method is computationally efficient for low-frequency data, i.e. 1 minute to 10 minutes granularity, which is a realistic solution for many commercially available smart meters. The remainder of this paper is organized as follows. In Section II, NILM is formulated as a binary quadratic optimization problem with linear constraints. In Section III, a mixed-integer programming formulation is described to ensure reliable automatic state detection. In Section IV, the proposed method is experimentally evaluated against the previous optimization-based methods on publicly available datasets. Finally, Section V concludes the paper.

II. BQP Disaggregation Algorithm

In this section, we formulate the energy disaggregation problem as an error minimization problem. Specifically, we start from Hart's formulation in [1] and we propose several enhancements aimed at improving the disaggregation accuracy. For all $t \in \mathcal{T}$, the energy disaggregation problem can be formulated as the following integer problem:

$$\begin{aligned} \min \quad & \left| y(t) - \sum_{i=1}^{N} \boldsymbol{p}_i^\top \boldsymbol{x}_i(t) \right| \\ \text{s.t.} \quad & \boldsymbol{x}_i(t) \in \{0,1\}^{S_i} \quad \forall i \in \mathcal{N}, \end{aligned} \tag{1}$$

where $\boldsymbol{p}_i \in \mathbb{R}^{S_i}$ is the vector of the non-zero Watt consumption levels of the appliance i (i.e. the power states) and $\boldsymbol{x}_i(t) \in \{0,1\}^{S_i}$ is the state variable of the appliance i. The j-th component of $\boldsymbol{x}_i(t)$ is set to 1 if the state j of the appliance i is active, and 0 otherwise.

The solution of the Problem (1) is generally not unique since there are many combinations of power levels resulting in the same aggregate signal. Thus, we add regularization and several constraints to better distinguish the contribution of all the appliances.

A. Objective Function

The objective function we minimize is given by the sum of three terms. The main term is the fitting error calculated as the sum of squared differences between the aggregated power consumption and the sum of the disaggregated consumption of each appliance:

$$f(\boldsymbol{x}(t)) = \sum_{t=1}^{T} \left(y(t) - \sum_{i=1}^{N} \boldsymbol{p}_i^\top \boldsymbol{x}_i(t) \right)^2. \tag{2}$$

We stress that the fitting error takes into account the presence of unknown electrical loads since $y(t)$ is the aggregate power measured from the main meter. We add to the fitting error (2) a term to exploit the knowledge that the underlying appliance signals are piece-wise constant over time. More in detail, we enforce sparseness and temporal smoothness on the state indicator vectors $\boldsymbol{x}_i(t)$ by penalizing every change of consumption level experienced by each appliance during the optimization horizon. Penalizing the norm of the difference between two consecutive parameters $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_i(t-1)$ is commonly referred in the literature as fused Lasso [33]. For this purpose, we use the ℓ_2 norm to directly penalizes large changes with respect to the temporal structure:

$$g_1(\boldsymbol{x}(t)) = \lambda_1 \sum_{t=2}^{T} \sum_{i=1}^{N} w_i \|\boldsymbol{x}_i(t) - \boldsymbol{x}_i(t-1)\|_2^2, \qquad (3)$$

where λ_1 is a penalty parameter. The appliance-specific weights w_i are non-negative parameters used to modulate the piece-wise constant behavior of the consumption profile. These weights are chosen depending on the appliance type and should be inversely proportional to the number of state transitions. Thus, for appliances that change state very rarely (i.e. clothes dryer and dishwasher) the weight is higher than the one of the appliances that frequently switch between states. Finally, we add to the fitting error (2) a second penalty term that promotes robustness to noise and device sparsity by preferring configurations that use a small number of operating appliances. If the structured noise given by the contribution of the unknown appliances to the aggregated energy consumption is significant, the minimization of the quadratic term in (2) would lead appliances to be set to an active state most of the time, producing unsatisfactory results. To avoid this drawback, we impose a penalty on the number of active appliances at a given time:

$$g_2(\boldsymbol{x}(t)) = \lambda_2 \sum_{t=1}^{T} \sum_{i=1}^{N} l_i \big(1 - s_i(t)\big) \|\boldsymbol{x}_i(t)\|_2^2, \qquad (4)$$

where λ_2 is a penalty parameter. Since $\boldsymbol{x}_i(t)$ is a binary vector with at most one component equal to 1 (see constraint (5)), the sum of squared ℓ_2 norms counts the number of devices that contribute to the aggregate power consumption. The parameter $s_i(t)$ is the probability, learned during a short intrusive period, that the appliance i is active at time t of the time horizon; for instance, during the night hours is unusual that sparse appliances like the dishwasher are active, so an activation during this period should be penalized much more than an activation in the usual times. On the other hand, the non-negative parameters l_i are inversely proportional to the ON time of the appliance; in this way, the activation of an appliance that is often ON is penalized less than the one of an appliance that is OFF most of the time. The hyper-parameters λ_1 and λ_2 are tuned by the user through cross-validation for balancing the trade-off between minimizing the fitting error and maximizing the sparsity.

B. Constraints

We constrain the state variables to avoid multiple active states from the same appliance. Specifically, for each appliance we allow at most one active state at a given time:

$$\mathbf{1}_{S_i}^\top \boldsymbol{x}_i(t) \leq 1 \quad \forall t \in \mathcal{T},\ \forall i \in \mathcal{N} \setminus \mathcal{O}. \qquad (5)$$

There are some appliances that operate the whole time, i.e. are always-on. For these devices the above constraint must be true at the equality:

$$\mathbf{1}_{S_i}^\top \boldsymbol{x}_i(t) = 1 \quad \forall t \in \mathcal{T},\ \forall i \in \mathcal{O}, \qquad (6)$$

where $\mathcal{O} \subseteq \mathcal{N}$ is the set that contains the indices of the Type-IV appliances.

Many appliances operate as finite state machines and their possible state transitions can be described by a state transition diagram. In order to explicitly model the behavior of an appliance as a finite state machine, we introduce the variables $\boldsymbol{u}_i(t) \in \{0,1\}^{S_i}$ and $\boldsymbol{d}_i(t) \in \{0,1\}^{S_i}$ and we add the following linear constraints:

$$\boldsymbol{x}_i(t) - \boldsymbol{x}_i(t-1) = \boldsymbol{u}_i(t) - \boldsymbol{d}_i(t) \quad \forall t \in \mathcal{T}, \forall i \in \mathcal{N}, \qquad (7)$$

$$\boldsymbol{u}_i(t) + \boldsymbol{d}_i(t) \leq \mathbf{1}_{S_i} \quad \forall t \in \mathcal{T}, \forall i \in \mathcal{N}, \qquad (8)$$

These additional variables model, respectively, an upward transition (i.e. from the OFF state to an ON state) and the downward transition (i.e. from an ON state to the OFF state). In the equation (7), if the appliance i changes operating mode at time t, the j-th component of the vector of the difference $\boldsymbol{x}_i(t) - \boldsymbol{x}_i(t-1)$ can be 1 or -1. In the first case, the constraint sets the j-th component of $\boldsymbol{u}_i(t)$ to 1 representing an upward transition, and in the latter the j-th component of $\boldsymbol{d}_i(t)$ to 1 representing a downward transition. Constraint (8) prevents the j-th component of both $\boldsymbol{u}_i(t)$ and $\boldsymbol{d}_i(t)$ to be simultaneously 1.

We expect an appliance to stay in a state for at least a few instants:

$$\sum_{\tau=t}^{t+a_{ij}-1} x_{ij}(\tau) \geq a_{ij} u_{ij}(t) \quad \forall t \in \mathcal{T}, \forall i \in \mathcal{N}, \forall j \in \mathcal{S}_i, \qquad (9)$$

where $a_{ij} \in \mathbb{R}$ is a parameter representing the minimum active time for each state j of the appliance i. Constraint (9) enforces the variables $x_{ij}(\tau)$ to be 1 for at least a_{ij} epochs when the appliance i goes into state j at time t, i.e, $u_{ij}(t) = 1$. On the other hand many appliances do not stay in the same state for a long time so we add the following constraint to force the appliances to change state before the maximum active time for that state:

$$\sum_{\tau=t}^{t+b_{ij}} x_{ij}(\tau) \leq b_{ij} \quad \forall t \in \mathcal{T}, \forall i \in \mathcal{N}, \forall j \in \mathcal{S}_i, \qquad (10)$$

where $b_{ij} \in \mathbb{R}$ is a parameter representing the maximum active time for state j of the appliance i. By constraining the appliance usage with the minimum and the maximum duration we can better disambiguate appliances with similar consumption profiles. For Type-II appliances, we can easily model a given state being active only if another state of the same appliance has finished:

$$\boldsymbol{U}_i \boldsymbol{u}_i(t) = \boldsymbol{D}_i \boldsymbol{d}_i(t) \qquad \forall t \in \mathcal{T}, i \in \mathcal{F}, \qquad (11)$$

where $\mathcal{F} \subseteq \mathcal{N}$ is the set that contains the indices of the Type-II appliances, $\boldsymbol{U}_i \in \{0,1\}^{S_i \times S_i}$ and $\boldsymbol{D}_i \in \{0,1\}^{S_i \times S_i}$ are indicator matrices defining the transition order between the states of the appliance i. If there is a transition from state j_1

to j_2, then the j_2-th component of $\boldsymbol{u}_i(t)$ is set to 1 only if the j_1-th component of $\boldsymbol{d}_i(t)$ is set to 1, meaning that the previous state of the appliance i is not active anymore.

Type-III appliances are the most challenging loads to disaggregate, as they arbitrarily change their power consumption [34]. Since we can not fully characterize Type-III appliances through explicit constraints, we treat them as ON/OFF multi-state appliances.

In order to limit the presence of spurious activations, we impose an upper bound o_i on the number of upward transitions for each appliance:

$$\sum_{t=2}^{T} \mathbf{1}_{S_i}^{\top} \boldsymbol{u}_i(t) \leq o_i \quad \forall i \in \mathcal{N}. \tag{12}$$

We notice that constraint (12) could be enforced by either using the upward transitions or the downward transitions. In fact, from the constraint (7) follows that the number of upward and downward transitions in the scheduled horizon differ at most by 1 in absolute value.

The following signal-aggregate constraint imposes that the sum of the disaggregated consumption profiles does not exceed the total power measured by the main meter:

$$\sum_{i=1}^{N} \boldsymbol{p}_i^{\top} \boldsymbol{x}_i(t) \leq y(t) \qquad \forall t \in \mathcal{T}. \tag{13}$$

To better characterize the different consumption patterns in a specific period (e.g., typically used appliances in the daytime and are unlikely to be used in the nighttime and vice-versa), we impose appliance-specific constraints on the maximum power consumption allowed in the scheduled horizon. Rather than making only one constraint for each appliance in the interval $[1, T]$, a series of tighter constraints could be enforced by partitioning the scheduling horizon into a fixed number of time intervals. As a result, the following constraint imposes that the energy consumption in a time period is less or equal than the maximum consumption m_{ih} allowed in that period:

$$\sum_{t \in \mathcal{T}_h} \boldsymbol{p}_i^{\top} \boldsymbol{x}_i(t) \leq m_{ih} \qquad \forall h \in \mathcal{H}, \forall i \in \mathcal{N}, \tag{14}$$

where $\mathcal{H} = \{1, \ldots, H\}$ is the set of H time intervals, and $\mathcal{T}_h$ is the set of indices $t \in \mathcal{T}$ related to the h-th time interval. Increasing H leads to tighter constraints but also implies a large number of upper bounds to estimate. Furthermore, when using limited historical observations, large H may lead to m_{ih} estimates varying too much in time so that the resulting accuracy may be compromised due to overfitting on the training set. In the following, we simply consider two time slots: from 1 AM to 5 AM for the nighttime and from 6 AM to 12 PM for the daytime.

In order to reduce the computational cost of the algorithm, we directly set to zero some variables by analyzing the aggregate consumption and the power levels. More in detail, let $\mathcal{Z} = \{t \in \mathcal{T}, i \in \mathcal{N}, j \in \mathcal{S}_i \mid y(t) \ll p_{ij}\}$ be the set of indices of the appliances that are more likely to be turned off, then we simply set $x_{ij}(t) = 0$, for all $(t, i, j) \in \mathcal{Z}$.

The overall optimization-based algorithm can be expressed as:

$$\begin{aligned} \min \; & \sum_{t=1}^{T} \left(y(t) - \sum_{i=1}^{N} \boldsymbol{p}_i^{\top} \boldsymbol{x}_i(t) \right)^2 \\ & + \lambda_1 \sum_{t=2}^{T} \sum_{i=1}^{N} w_i \|\boldsymbol{x}_i(t) - \boldsymbol{x}_i(t-1)\|_2^2 \\ & + \lambda_2 \sum_{t=1}^{T} \sum_{i=1}^{N} l_i \big(1 - s_i(t)\big) \|\boldsymbol{x}_i(t)\|_2^2 \end{aligned} \tag{15}$$

$$\begin{aligned} \text{s.t. } & \text{Constraints (5) - (14)} \\ & \boldsymbol{x}_i(t),\ \boldsymbol{u}_i(t),\ \boldsymbol{d}_i(t) \in \{0,1\}^{S_i} \quad \forall t \in \mathcal{T},\ \forall i \in \mathcal{N}. \end{aligned}$$

Problem (15) is an optimization problem with binary variables, quadratic objective function and linear constraints. The binary quadratic program with linear constraints (BQP) is a general class of optimization problems that are known to be very difficult to solve due to the non-convexity and the integrality of the variables. The number of variables and constraints is $O(T \sum_{i=1}^{N} S_i)$. Due to its combinatorial nature, the problem becomes expensive to solve as the scheduling horizon T and the number of appliances N increase. However, this formulation is computationally efficient for low-frequency data where the resolution ranges from 1 minute to 10 minutes. Smart meter data with this granularity are more common in practical applications due to the relatively low hardware cost. In this case, Problem (15) can be globally solved in order of seconds by state-of-the-art integer programming solvers without specialized hardware.

III. State detection and parameter estimation

The optimization model in (15) relies on several input parameters that need to be estimated. The most important ones are the power states $\boldsymbol{p}_i$, representing the steady-state ratings. Steady-state ratings are usually extracted from data sheets or by means of a clustering algorithm on the ground truth data [28], [29], and used to characterize the typical consumption signatures by a constant value representing the average power. In practice, the power states of household appliances fluctuate within a range, so that approximating them with a straight line may affect the quality of the disaggregation. As observed in [26], being able to capture appliances' dynamics in addition to the steady-state ratings could be effective to increase the disaggregation accuracy. However, incorporating these dynamics as additional states in the optimization problem would dramatically increase the computational cost. For this reason, we develop a novel clustering-based approach for steady-state ratings and transient spans extraction using a very small training set of disaggregated data. This is done by solving a mixed-integer optimization problem that jointly classifies the power levels into S_i clusters and estimates an autoregressive (AR) submodel of order q_i for each cluster. The autoregressive coefficient are incorporated in the post-processing phase to refine the flat estimate of each appliance. Formally, for all $i \in \mathcal{N}$, we solve:

$$\begin{aligned} \min \quad & \sum_{j=1}^{S_i} \sum_{t=q_i+1}^{M} \delta_{ij}(t)\big(y_i(t) - \boldsymbol{\phi}_{ij}^\top \boldsymbol{z}_i(t)\big)^2 \\ \text{s.t.} \quad & \mathbf{1}_{S_i}^\top \boldsymbol{\delta}_i(t) = 1 \qquad \forall t \in \mathcal{M}, \\ & \delta_{ij}(t) \in \{0,1\} \quad \forall j \in \mathcal{S}_i, \forall t \in \mathcal{M}, \\ & \phi_{ij}^h \in \mathbb{R} \qquad\quad \forall j \in \mathcal{S}_i, \forall h \in \{0,\dots,q_i\}, \end{aligned} \tag{16}$$

where $\boldsymbol{z}_i(t) = [1, y_i(t-1), \dots, y_i(t-q_i)]^\top$ is the input vector of the historical observations and $\boldsymbol{\phi}_{ij} = [\phi_{ij}^0, \phi_{ij}^1, \dots, \phi_{ij}^{q_i}]^\top$ is the vector of the autoregressive coefficients. Each binary variable $\delta_{ij}(t)$ decides whether the data point $\boldsymbol{z}_i(t)$ is assigned to the j-th submodel of the appliance i, under the constraint that each data point must be assigned to only one submodel. Problem (16) is a mixed-integer non-linear program that can be shown to be equivalent to its continuous relaxation, where the binary constraint on δ_{ij} is replaced by $\delta_{ij} \geq 0$. This equivalence derives from the optimal solution of the relaxation being integer. The resulting continuous problem is still non-convex. In order to efficiently produce a good local optimum, we use an iterative two-block Gauss-Seidel decomposition method [35]. In fact, the structure of the objective function and the constraints imply that the computation of the global minimum with respect to each block can be done in a computationally efficient way. More in detail, by fixing the variables $\delta_{ij}(t)$, Problem (16) becomes the least-squares estimation of S_i autoregressive models of order q_i:

$$\boldsymbol{\phi}_{ij}^* = \underset{\boldsymbol{\phi}_{ij}}{\arg\min} \left[\sum_{j=1}^{S_i} \sum_{t=q_i+1}^{M} \overline{\delta}_{ij}(t)\big(y_i(t) - \boldsymbol{\phi}_{ij}^\top \boldsymbol{z}_i(t)\big)^2 \right]. \tag{17}$$

Problem (17) is an unconstrained convex optimization problem that can be solved in closed form by setting the gradient equal to zero. On the other hand, given $\boldsymbol{\phi}_{ij}$, the problem with respect to $\delta_{ij}(t)$ is linear programming problem that is separable into $M - (q_i + 1)$ subproblems. In particular, the t-th subproblem has the form:

$$\begin{aligned} \min \quad & \sum_{j=1}^{S_i} \delta_{ij}(t)\big(y_i(t) - \overline{\boldsymbol{\phi}}_{ij}^\top \boldsymbol{z}_i(t)\big)^2 \\ \text{s.t.} \quad & \mathbf{1}_{S_i}^\top \boldsymbol{\delta}_i(t) = 1, \\ & \delta_{ij}(t) \geq 0 \qquad \forall j \in \mathcal{S}_i. \end{aligned} \tag{18}$$

The optimal solution of (18) can be constructed by observing that if $y_i(t)$ is close to the power level j^*, then $\delta_{ij^*}(t)$ is set to 1, whereas the variables with $j \neq j^*$ are set to 0. The algorithm starts with an initial guess of the power levels $\boldsymbol{p}_i^0$ and minimizes over one block of variables with the other fixed, and vice versa until convergence. The convergence of this alternating minimization procedure can be proved by adapting the k-means convergence proof that can be found in [36]. The main idea is that the variables $\delta_{ij}(t)$ can only assume a finite number of values (i.e., the number of possible assignments is finite). Therefore, since the algorithm finds the global minimum with respect to $\boldsymbol{\phi}_{ij}$ for a given assignment, there will exist a subsequence of iterations where power levels and assignment variables do not change, satisfying the stopping criterion $\|\overline{\boldsymbol{p}}_i^k - \overline{\boldsymbol{p}}_i^{k-1}\|_2 \leq \epsilon$.

An initial estimate of the power levels can be automatically obtained by a clustering algorithm, such as k-means [37], [38]. Since the number of power levels is unknown, k-means is run with an increasing number of clusters. The number of power levels is then validated with the elbow method and the cluster centers, representing the average power consumption, are selected as the initial guess $\boldsymbol{p}_i^0$ of the operating modes. The overall decomposition method for automatic state detection is illustrated in Algorithm 1.

In the initialization phase, the parameters $\boldsymbol{\phi}_{ij}$ of the autoregressive subprocess are unknown, so the binary variables $\delta_{ij}(t)$ are chosen so that the squared distance between the power demanded by the appliance i at time t and the power levels $\boldsymbol{p}_i^0$ is minimized. In the main loop, the AR coefficients are estimated and the variables $\delta_{ij}(t)$ are adjusted on the basis of the current parameters. Finally, for each appliance i we refine the estimate of the power level j by simply taking the conditional expectation of the j-th autoregressive subprocess. The algorithm terminates when the difference between the power levels of two consecutive iterations is less or equal than a small value ε and returns the estimated power levels $\boldsymbol{p}_i$ and the AR coefficients $\boldsymbol{\phi}_{ij}$.

Algorithm 1: Automatic State Detection

```
Data: The training set $y_i(t)$, and the tolerance $\varepsilon = 10^{-5}$.
Result: The estimated power $\overline{\boldsymbol{p}}_i$ and the AR coefficients $\boldsymbol{\phi}_{ij}$.
$\overline{\boldsymbol{p}}_i^0 \leftarrow$ run the k-means algorithm on $y_i(t)$
for $t = q_i + 1 \dots M$ do
    $j^* \leftarrow \arg\min_{j=1\dots S_i} \big(y_i(t) - \overline{p}_{ij}^0\big)^2$
    $\delta_{i,j}^0(t) = \begin{cases} 1 & \text{if } j = j^* \\ 0 & \text{otherwise} \end{cases}$
end
$k \leftarrow 1$
do
    $\boldsymbol{\phi}_{ij}^k \leftarrow \arg\min \sum_{j=1}^{S_i} \sum_{t=q_i+1}^{M} \delta_{i,j}^{k-1}(t)\big(y_i(t) - \boldsymbol{\phi}_{ij}^\top \boldsymbol{z}_i(t)\big)^2$
    for $t = q_i + 1 \dots M$ do
        $j^* \leftarrow \arg\min_{j=1\dots S_i} \big(y_i(t) - \boldsymbol{\phi}_{ij}^{k\top} \boldsymbol{z}_i(t)\big)^2$
        $\delta_{ij}^k(t) = \begin{cases} 1 & \text{if } j = j^* \\ 0 & \text{otherwise} \end{cases}$
    end
    for $j = 1 \dots S_i$ do
        $\overline{p}_{ij}^k \leftarrow \frac{\phi_{ij}^{k,0}}{1 - \sum_{h=1}^{q_i} \phi_{ij}^{k,h}}$
    end
    $k \leftarrow k + 1$
while $\|\overline{\boldsymbol{p}}_i^k - \overline{\boldsymbol{p}}_i^{k-1}\|_2 > \epsilon$
return $\overline{\boldsymbol{p}}_i$ and $\boldsymbol{\phi}_{ij}$
```

Given the optimal solution of (15) $\boldsymbol{x}_i^*(t)$, the disaggregated power consumption of the appliance i at time t is given by $\hat{y}_i(t) = \boldsymbol{p}_i^\top \boldsymbol{x}_i^*(t)$. This estimate relies only on the static power levels. In order to refine the estimate we need to post-process the output by including the transient spans through the AR coefficients. This is done by applying a moving horizon forecasting procedure for $t = q + 1, \dots, T$:

$$\hat{y}_i(t) = \max\left\{0, \sum_{j=1}^{S_i} x_{ij}^*(t)\big(\phi_{ij}^0 + \sum_{k=1}^{q_i} \phi_{ij}^k \hat{y}_i(t-k)\big)\right\}. \tag{19}$$

Similarly to the power states, the remaining parameters in (15) can be estimated from time series analysis and limited historical information. The extraction of some of them may be done by visual inspection. However, automatic extraction becomes necessary when applying the model to a real-world scenario. Furthermore, for practical NILM applications, the training phase should be as short and simple as possible [39]. To this end, for $o_i, m_{ih}, a_{ij}, b_{ij}$ we compute the frequency distribution over the training set and we obtain the estimate by selecting a suitable percentile. In particular, the parameters o_i and m_{ih} are computed for each interval of length T over the training set and estimated by taking the 95th percentile. Likewise, the minimum and maximum time in state a_{ij} and b_{ij} are estimated by looking at the left and the right tails of the histogram and taking, respectively, the 5th and 95th percentile of the time spent in state j in the interval T. The parameters w_i are inversely proportional to the frequency of state transitions that characterize the device, therefore they are calculated as follows:

$$w_i = \frac{M}{r_i} \qquad \forall i \in \mathcal{N}, \tag{20}$$

where r_i is the number of state transitions in the training set for the appliance i. The parameters $s_i(t)$ represent the probability that the device i is active (with respect to a threshold of 10 Watt) at time t of the scheduling horizon. These probabilities can be estimated as follows:

$$s_i(t) = \frac{v_i(t)}{M/T} \qquad \forall t \in \mathcal{T}, i \in \mathcal{N}, \tag{21}$$

where $v_i(t)$ is the number of times that the appliance i is ON at time t, whereas the denominator represents the number of time horizons in the training set. Finally, the parameters l_i, which are inversely proportional to how much the device is typically used, are calculated as:

$$l_i = \frac{M}{\sum_{t=1}^{M} v_i(t)} \qquad \forall i \in \mathcal{N}. \tag{22}$$

The validation set is used only for selecting and tuning the regularization parameters $\lambda_1 \in \Lambda$ and $\lambda_2 \in \Lambda$ in (15) by means of a grid-search procedure over the grid $\Lambda \times \Lambda$. Also in the validation phase, the sub-metered power consumptions are assumed to be available.

The overall NILM algorithm can be summarized by the diagram reported in Figure 1.

IV. Results

In this section we present the experiments carried out to test the performance of the overall NILM algorithm and we address several research questions:

1) How does the proposed algorithm compare to state-based approaches available in the literature?
2) Is the approach robust to the presence of unmetered appliances?
3) How reliable is the parameter estimation procedure and how often does one need to update the parameters to reflect changes in the system?

Overall, by answering these research questions we aim to provide insights on how the proposed algorithm can be used in practice. We compare our NILM-BQP with NILM algorithms proposed for low sampling rates and active power measurements. Specifically, we compare with Hart's formulation (CO) [1], Aided Linear Programming (ALIP) [26], Sparse Optimization (SO) [27], State Transition Integer Programming (STIP) [29] and Super State HMM (SHMM) [16]. We first evaluate the robustness to noise on a synthetic dataset to assess how the presence of unmetered appliances affects the quality of the disaggregation, and then we evaluate our algorithm on 5 houses coming from 3 real-world datasets. Finally, we evaluate the reliability of the parameter estimation procedure and how the freshness and staleness of the parameters impact the accuracy of the disaggregation.

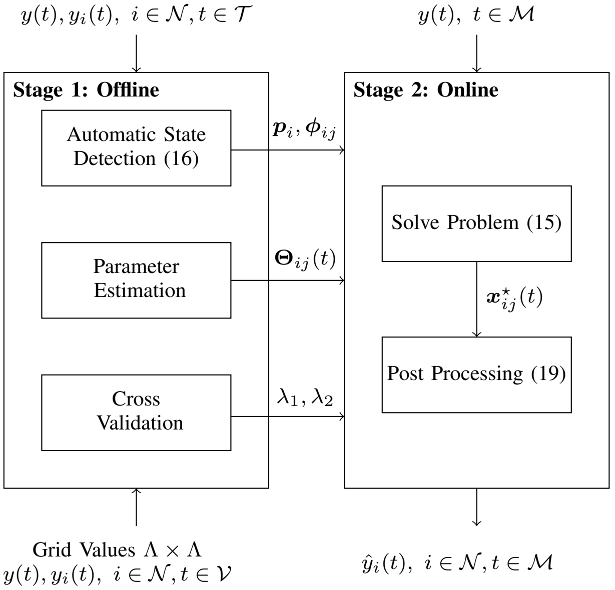
$y(t), y_i(t),\ i \in \mathcal{N}, t \in \mathcal{T}$
$y(t),\ t \in \mathcal{M}$
Stage 1: Offline
Stage 2: Online
Automatic State Detection (16)
$\boldsymbol{p}_i, \boldsymbol{\phi}_{ij}$
Parameter Estimation
$\boldsymbol{\Theta}_{ij}(t)$
Cross Validation
λ_1, λ_2
Solve Problem (15)
$\boldsymbol{x}^\star_{ij}(t)$
Post Processing (19)
Grid Values $\Lambda \times \Lambda$
$y(t), y_i(t),\ i \in \mathcal{N}, t \in \mathcal{V}$
$\hat{y}_i(t),\ i \in \mathcal{N}, t \in \mathcal{M}$

Figure 1. Overall disaggregation process: Stage 1 (Offline) and Stage 2 (Online). For all i, j, t we denote by $\boldsymbol{\Theta}_{ij}(t) = [w_i, l_i, s_i(t), a_{ij}, b_{ij}, o_i, m_{ih}]^\top$ the vector of the parameters.

A. Performance Metrics

In order to evaluate the performance of our method, we use regression and classification metrics. Regression metrics measure how closely the energy consumption of an appliance matches the energy consumption predicted for that appliance, whereas the classification metrics measure how accurately NILM algorithms can predict what appliance is running in each state. Following [40], we use the Estimation Accuracy (EA) for the appliance $i \in \mathcal{N}$, and the Overall Estimation Accuracy (OEA) for N appliances:

$$EA_i = 1 - \frac{\sum_{t=1}^{T} |y_i(t) - \hat{y}_i(t)|}{2\sum_{t=1}^{T} y_i(t)}, \tag{23}$$

$$OEA = 1 - \frac{\sum_{i=1}^{N}\sum_{t=1}^{T} |y_i(t) - \hat{y}_i(t)|}{2\sum_{i=1}^{N}\sum_{t=1}^{T} y_i(t)}. \tag{24}$$

In order to calculate the accuracies of non-binary classifications, we use the Finite State F-score (FS_i) that is the harmonic mean of the adjusted precision (P_i) and recall (R_i) for the appliance i:

$$P_i = \frac{TP_i - A_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i - A_i}{TP_i + FN_i}, \tag{25}$$

$$A_i = \sum_{t=1}^{T} \frac{|\hat{s}_i(t) - s_i(t)|}{S_i}, \quad FS_i = 2\frac{P_i \cdot R_i}{P_i + R_i}, \tag{26}$$

where TP_i, FP_i, and FN_i stand for true positives, false positives, and false negatives for the appliance i, respectively. In the NILM context, TP_i counts how many times the algorithm correctly predicts that the appliance is ON, FP_i how many times the appliance is OFF but predicted ON, and FN_i how many times the appliance is ON but predicted OFF. The quantity A_i is the inaccuracy portion of true positives which converts them into a discrete measure from a binary one, $\hat{s}_i(t)$ is the estimated state of the appliance i at time t, $s_i(t)$ is the ground truth state, and S_i is the number of states for the appliance i. The Overall FS-Fscore (OFS) is obtained by summation over all the appliances for each TP_i, FP_i, FN_i, and A_i.

To measure the contribution of unmetered loads, we report the percentage of noise in each test. The percent-noisy measure [40] $(\% - NM)$ can be calculated on the ground truth data as follows:

$$\% - NM = \frac{\sum_{t=1}^{T} |y(t) - \sum_{i=1}^{N} y_i(t)|}{\sum_{t=1}^{T} y(t)}. \tag{27}$$

B. Experimental Setup

The proposed algorithm assumes low-sampling rate measurements. We use observations at 60 seconds resolution, and we downsample the data when needed to satisfy this assumption. We use the same autoregressive order $q_i = 3$ for each appliance, even though the estimation of the transient spans could be further improved by tuning this parameter. We adopt the same experimental design for all the methods: we use only two weeks for the estimation of the parameters, a week to cross-validate the choice of the hyperparameters λ_1 and λ_2, and a week as test set. The scheduling horizon is set to 1 day ($T = 1400$) which is suitable for decision making using low-frequency data [41]. Furthermore, in order to minimize any bias effect, we run all the NILM algorithms one day at a time to obtain the disaggregated traces for the test week (i.e., we solve 7 optimization problems and we average the results). The values of λ_1 and λ_2 are determined offline through a grid-search procedure by maximizing the sum of OEA and OFS on the validation set. The grid values used in the experiments are in $\Lambda = \{200, 300, \ldots, 2200\}$ and their optimal values for each dataset are reported in Table I. BQP is implemented in AMPL [42] and solved to global optimality with Gurobi optimizer [43]. All the experiments have been performed on a laptop with Intel Core i7-8565U CPU and 8 GB of RAM. The source code and the parameters of all the considered case studies are available at https://github.com/antoniosudoso/nilm-bqp.

Table I
OPTIMAL HYPER-PARAMETERS λ_1 AND λ_2

Dataset	λ_1	λ_2
AMPDS	1000	2000
UKDALE 1	500	1300
UKDALE 2	800	1500
REFIT 3	800	1600
REFIT 9	300	900

C. Robustness to Noise

It is well known that the presence of unknown appliances negatively affects the disaggregation accuracy of known appliances. Therefore a critical aspect for the deployment of the NILM service in a real-world scenario is the robustness to noise. In order to test the effectiveness of our formulation when the noise increases, we use the artificial dataset SYND [44]. It contains 180 days of data at the granularity of 200 milliseconds of a single household with 21 appliances. This dataset has been generated using appliance-specific signatures of real-world NILM datasets. We generate artificial aggregates with an increasing number of appliances. Specifically, we disaggregate the 4 top-consuming appliances that are dishwasher, electric stove, washing machine, and iron. Starting from the ideal case where no noise is present, i.e., the aggregate signal is given by the sum of the 4 top-consuming appliances, we add the traces of the remaining devices one at a time in decreasing order of energy consumption, and we treat them as unknown appliances. In this way, we generate 18 different datasets, corresponding to artificial aggregates of increasing noise scenarios, where the percentage of noise ranges from 0% to around 30%. The noisiest dataset contains all the remaining 17 appliances treated as noise, resulting in the original artificial aggregate signal.

In Figures 2 and 3, we report the average OEA and the average OFS obtained by all the methods when the number of appliances added to the aggregated signal increases. This picture shows the well-known difficulty encountered by most of the state-based approaches when the noise increases. Our method, on the other hand, does not suffer from the increase in noise, showing very high accuracy in all scenarios. This behavior stays essentially the same when looking at the single appliance. These results confirm the robustness to noise of our formulation, which allows overcoming the drawback of the other optimization-based approaches. This behavior is also confirmed by the experiments on real-world datasets described in the next section.

D. Real-world Datasets

We test our formulation on three real-world datasets: AMPDS [45], UKDALE [46] and REFIT [47], containing both the aggregate and the appliance-wise active power. From each dataset, we select at most two houses where the percentage of noise is up to 40% and we use as target appliances the top-consuming ones. From AMPDS we select 6 appliances: clothes dryer (CD), dishwasher (DW), fan and thermostat (FT), entertainment (EN), fridge (FR), and heat pump (HP). We model FT and EN as always-on, whereas we model CD and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSG.2022.3152147, IEEE Transactions on Smart Grid

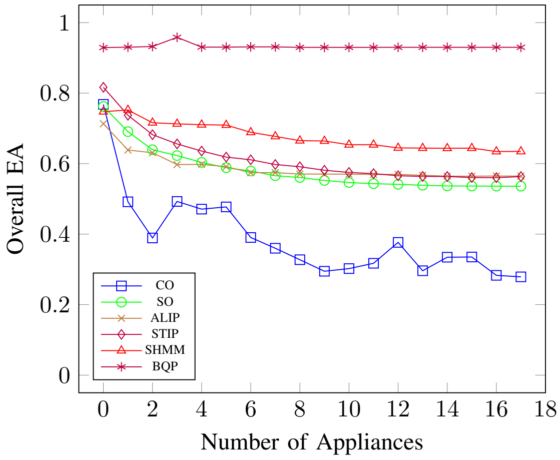

Figure 2. Overall Estimation Accuracy as the number of unmetered appliances varies.

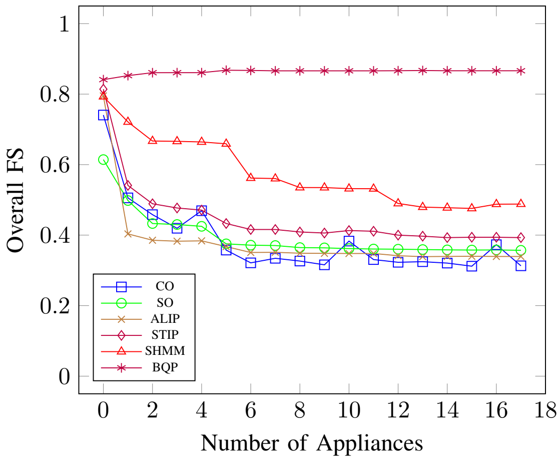

Figure 3. Overall FS as the number of unmetered appliances varies.

HP as state machines. From UKDALE we select house 1 and 2. For house 1 we disaggregate 5 appliances: boiler (BO), washing machine (WM), dishwasher (DW), home theater (HT), and fridge (FR). For this house, WM and DW are treated as state machines, whereas BO and HT are treated as always-on. From House 2 we select 5 appliances: server (SE), kettle (KE), dishwasher (DW), fridge (FR), and modem (MO). Here, SE and MO are always-on, whereas DW is a state machine. Finally, for the REFIT dataset we choose house 3 and 9. For house 3 we use 4 appliances: fridge (FR), tumble dryer (TD), dishwasher (DW), and television (TV). We model TD and DW are state machines. For house 9 we select 4 appliances: fridge (FR), washing machine (WM), dishwasher (DW), and kettle (KT). We treat WM and DW are state machine appliances. In Table II we report the training, validation and test days and the percent-noisy measure for each house, computed according to (27).

In Tables III-V, we report for each method and for each house the Estimation Accuracy (EA) and the Finite State F-Score (FS) for each appliance averaged on the test week. Furthermore, we report the average Overall EA and the average Overall FS. Note that we report the performance of our disaggregation algorithm with (BQPP) and without post-processing (BPQ), to evaluate the impact of the post-processing phase.

Table II
TRAINING, VALIDATION AND TEST DAYS.

Dataset	Training	Validation	Test	Noise
AMPDS	10-23/04/12	24-30/04/12	01-07/05/12	38%
UKDALE1	09-22/06/14	23-29/06/14	30-06/06-07/14	39%
UKDALE2	01-14/06/13	15-21/06/13	22-29/06/13	36%
REFIT3	20-02/10-11/13	03-09/11/13	10-16/11/13	35%
REFIT9	18-01/11-12/14	02-08/12/14	09-15/12/14	40%

Finally, we report the average execution time in seconds for each method. All the compared approaches are based on a two-stage procedure: an offline phase for parameter estimation and an online phase for the actual disaggregation. In all the cases, the reported time represents the computational time of the online procedure only. Note that, anyway, the offline phase of our approach is extremely cheap in terms of computational time: the most expensive part is the grid search for λ_1 and λ_2 that requires few minutes. Looking at the execution times, the largest ones are required by CO, due to the larger feasible region, and by STIP, due to the nonlinear constraints induced by the state machine constraints. The solution of our BQP problem only requires a few seconds, despite the large size: indeed, we have thousands of binary variables and thousands of constraints. Therefore, our approach could be used in a real-time setting, at least for the considered data granularity, and the considered scheduling horizon. This makes the proposed method suitable for low-cost microcontrollers.

As for the performance metrics, it turns out that our method is always the best in both metrics on all the appliances. To get a better picture of the different methods' behavior, we extract from the tables the average Overall EA and the average Overall FS on all the different houses, and plot them in Figures 4 and 5. The superiority of our method in both metrics is evident, and in Figure 4 also emerges the advantage of the post-processing phase, which has an impact only on the EA since it does not influence the activations. Looking at the detailed performance on the single appliances in Tables III-V, it can be seen that most of the approaches are able to correctly disaggregate always-on appliances (see for example appliance FT of AMPDS), whereas the BQP is by far more robust on appliances that have a variable number of activations (see for example appliances DW of both house 1 and 2 of UKDALE). The high performance on these appliances derives from the ability of the proposed formulation to disambiguate the appliances' states, combined with the correct estimation of the power states. The beneficial effect of the post-processing is confirmed since the EA metric improves on all the appliances in all the datasets. To better show the performance of the proposed algorithm, in Figure 8 we show the 1-day disaggregation results produced by the top three methods BQPP, SHMM and STIP and we select appliances with different power signatures and behaviours. We can see that the fridge typically shows a peak at the beginning of its working cycle. Differently from STIP that produces a flat estimate, our approach manages to capture this behaviour thanks to the AR coefficients. On

Table III
AMPDS HOUSE 1

Method	Metric	Overall	CD	DW	FT	EN	FR	HP	Time
CO	EA	0.493	0.018	-1.548	0.611	0.398	0.189	0.867	1133.5
	FS	0.583	0.057	0.045	0.718	0.635	0.398	0.781	
SO	EA	0.569	0.815	0.500	**0.978**	0.195	0.490	-0.141	1.8
	FS	0.607	0.692	0.001	**1.000**	0.657	0.056	0.809	
ALIP	EA	0.587	0.764	0.375	**0.978**	0.361	0.195	0.629	3.40
	FS	0.675	0.447	0.316	**1.000**	0.862	0.324	0.307	
STIP	EA	0.681	0.942	0.194	**0.978**	0.432	0.249	0.842	1206.6
	FS	0.683	0.836	0.543	**1.000**	0.872	0.513	0.653	
SHMM	EA	0.754	0.496	0.364	0.851	0.835	0.686	0.834	49.2
	FS	0.786	0.310	0.561	0.841	0.943	0.662	0.757	
BQP	EA	0.889	0.955	0.863	**0.978**	0.849	0.721	0.907	52.5
	FS	**0.942**	**0.932**	**0.844**	**1.000**	**0.978**	**0.760**	**0.919**	
BQPP	EA	**0.902**	**0.976**	**0.888**	**0.978**	**0.862**	**0.736**	**0.923**	52.8
	FS	**0.942**	**0.932**	**0.844**	**1.000**	**0.978**	**0.760**	**0.919**	

Table IV
UKDALE HOUSES 1 AND 2

		House 1							House 2						
Method	Metric	Overall	BO	WM	DW	HT	FR	Time	Overall	SE	KE	DW	FR	MO	Time
CO	EA	-0.192	0.375	-0.501	-2.029	-1.421	0.411	1034.5	0.403	0.754	0.622	-0.030	0.584	0.742	755.7
	FS	0.421	0.587	0.155	0.074	0.598	0.422		0.662	0.710	0.535	0.101	0.786	0.755	
SO	EA	0.305	0.977	0.485	-3.221	0.559	0.574	1.0	0.640	0.990	0.507	0.589	0.620	**0.997**	0.9
	FS	0.677	0.997	0.480	0.105	0.829	0.677		0.406	**1.000**	0.009	0.003	0.012	**1.000**	
ALIP	EA	0.174	0.195	0.383	-1.468	-0.933	0.557	9.5	0.564	0.990	0.661	0.325	0.659	**0.997**	3.2
	FS	0.561	0.789	0.442	0.149	0.616	0.454		0.638	**1.000**	0.105	0.209	0.597	**1.000**	
STIP	EA	0.157	0.313	0.581	-0.506	-0.651	0.569	723.6	0.579	0.990	0.661	0.387	0.496	**0.997**	957.1
	FS	0.579	0.821	0.405	0.178	0.843	0.549		0.778	**1.000**	0.565	0.236	0.857	**1.000**	
SHMM	EA	0.629	0.884	0.478	0.346	0.184	0.731	14.7	0.723	0.966	0.519	0.571	0.802	0.981	4.8
	FS	0.782	0.913	0.306	0.779	0.818	0.712		0.856	0.982	0.780	0.382	0.876	0.982	
BQP	EA	0.869	0.958	0.711	0.903	0.699	0.795	25.4	0.909	0.990	0.778	0.907	0.921	**0.997**	5.9
	FS	**0.951**	**0.997**	**0.884**	**0.802**	**0.967**	**0.853**		**0.970**	**1.000**	**0.789**	**0.868**	**0.975**	**1.000**	
BQPP	EA	**0.894**	**0.979**	**0.746**	**0.932**	**0.702**	**0.809**	25.7	**0.926**	**0.991**	**0.790**	**0.933**	**0.938**	**0.997**	6.1
	FS	**0.951**	**0.997**	**0.884**	**0.802**	**0.967**	**0.853**		**0.970**	**1.000**	**0.789**	**0.868**	**0.975**	**1.000**	

Table V
REFIT HOUSES 3 AND 9

		House 3						House 9					
Method	Metric	Overall	FR	TD	DW	TV	Time	Overall	FR	WM	DW	KE	Time
CO	EA	0.264	0.482	0.344	-0.092	0.589	989.1	-0.262	0.358	-0.467	-0.017	-0.043	687.3
	FS	0.501	0.676	0.441	0.160	0.426		0.164	0.240	0.074	0.116	0.000	
SO	EA	0.375	0.640	0.281	-0.863	0.763	0.9	0.166	0.619	0.522	0.471	-1.007	1.1
	FS	0.630	0.578	0.409	0.218	0.805		0.603	0.702	0.106	0.282	0.332	
ALIP	EA	0.538	0.535	0.554	0.278	0.874	1.1	0.226	0.658	0.402	0.432	0.486	0.9
	FS	0.673	0.548	0.465	0.350	0.872		0.490	0.587	0.120	0.204	0.000	
STIP	EA	0.553	0.495	0.669	0.131	0.752	468.9	0.556	0.667	0.398	0.469	0.581	509.6
	FS	0.526	0.549	0.637	0.124	0.740		0.525	0.599	0.197	0.308	0.416	
SHMM	EA	0.676	0.798	0.494	0.500	0.889	3.1	0.695	0.762	0.485	0.745	0.500	4.9
	FS	0.744	0.802	0.103	0.358	0.782		0.637	0.789	0.272	0.611	0.005	
BQP	EA	0.828	0.807	0.841	0.819	0.881	3.6	0.806	0.771	0.737	0.835	0.762	7.8
	FS	**0.862**	**0.858**	**0.855**	**0.815**	**0.892**		**0.801**	**0.814**	**0.771**	**0.799**	**0.643**	
BQPP	EA	**0.847**	**0.813**	**0.878**	**0.844**	**0.894**	3.8	**0.823**	**0.788**	**0.744**	**0.846**	**0.772**	7.9
	FS	**0.862**	**0.858**	**0.855**	**0.815**	**0.892**		**0.801**	**0.814**	**0.771**	**0.799**	**0.643**	

the other hand, the entertainment is a Type-III appliance whose activations are successfully captured even though the consumption is approximated by the average power. Looking at the dishwasher, we can see that our method is successful at capturing with very good accuracy the activations and the kW-level power consumption. In general, our approach is able to correctly disambiguate appliances with similar consumption patterns and to correctly estimate the load of active appliances in a real-world scenario. The competitors tend to produce false positives for the appliance that are not active (see the disaggregation results for television and boiler in Figure 8). Furthermore, the combination of our post-processing phase and of our parameter estimation procedure allows to capture with great accuracy the scale of the active power states, increasing the EA metric.

E. Parameters Stability

The proposed optimization model relies on several parameters that may need to be updated in order to reflect changes

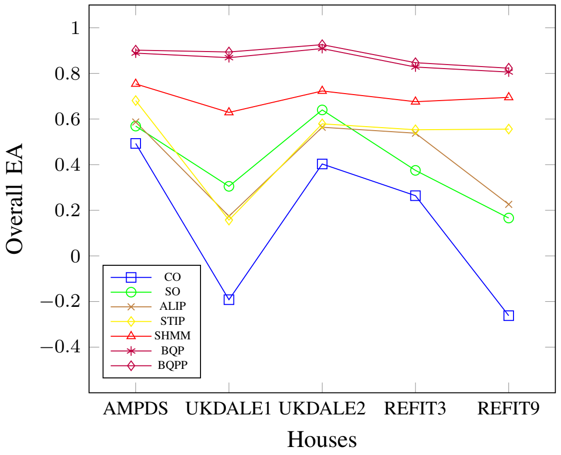

Figure 4. Overall Estimation Accuracy for each house and method.

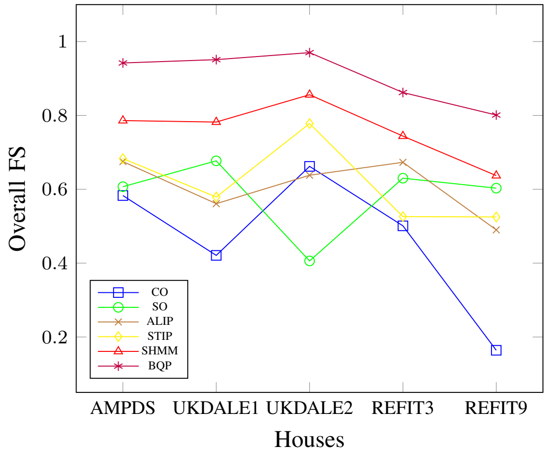

Figure 5. Overall FS-Fscore for each house and method.

in the system. As long as there is no significant change in the power profiles, the estimated parameters should remain stable over time. In the following, we conduct experiments to study whether the freshness and staleness of the parameters impact the accuracy of the disaggregation. We compare two different scenarios:

a) Scenario A (Fixed): the parameters are estimated just once, at the beginning of the experiment, using the training and validation weeks reported in Table II. Then, we simulate a periodic disaggregation once per month until six months after the beginning of the experiment. Therefore, we perform Stage 2 of our disaggregation procedure on six different test weeks, further and further away from the weeks used for parameter estimation. In this setting, we evaluate what happens when the parameters are not updated for a long time.

b) Scenario B (Updated): the overall disaggregation procedure is repeated at one month interval up to six months ahead on the same test weeks of Scenario A. However, in this case the parameters are re-estimated by using two weeks for training and one for validation immediately before the test week. This is an "ideal scenario" where Stage 1 is executed right before Stage 2. We note that this could still be viable in practice if historical observations were available.

To conduct the experiments, we use the AMPDS dataset since it has a larger number of appliances, and thus a larger number of parameters. Results are reported in Figures 6 and 7. In Figure 6 we evaluate in both scenarios the overall EA and FS scores obtained in the six test weeks placed at one month interval. Interestingly enough the overall performance do not differ too much in the two settings, with a slight decrease in Scenario A starting from the fourth month. To better understand the two performance, in Figure 7 the same metrics are reported in both scenarios on the single appliances. We can see that CD, DW, FT, and FR are not affected by the parameter staleness, since the accuracy in the two scenarios is almost identical. On the other hand, the only two appliances that suffer from staleness are the HP starting from the fifth month and EN starting form the third one. According to the data sheet in [45], the heat pump cools the house in summer and heats the house in winter. The behavioural change of HP is related to a seasonal change, and hence the need to use the appliance more often in the heating setting. Looking at the ground truth signature, this change implies the presence of an additional state that is successfully captured in Scenario B when the parameters are refreshed, but not in Scenario A. As for the EN, in [45] is treated as a composite appliance having multiple loads such as television set, personal video recorder and external amplifier. Looking at the active power signature it seems that from the fourth month onwards, it continuously changes its behaviour displaying a much bigger variability that is better captured in Scenario B and hence justifying the need for a new parameter estimation.

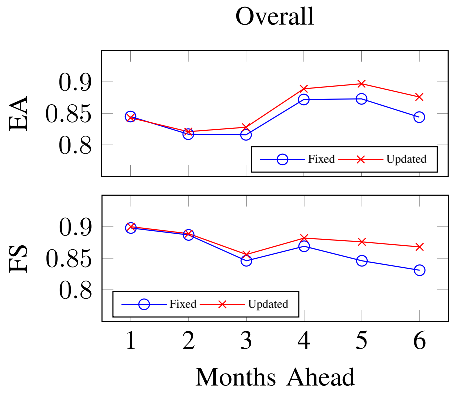

Figure 6. Overall accuracy of the proposed approach in Scenario A (Fixed Parameters) and Scenario B (Updated Parameters).

V. Conclusions

In this paper, we have proposed a novel optimization-based algorithm for NILM. Our approach can separate many appliances almost perfectly using only the total aggregate signal. It is computationally efficient for low-frequency data which are commonly installed in many smart homes. The additional penalty terms promoting sparsity and the appliance-specific constraints narrow down the feasible region, improve the algorithm performance and reduce the computational burden. We also have proposed a mixed-integer formulation for automatic state detection, and a post-processing technique to include state

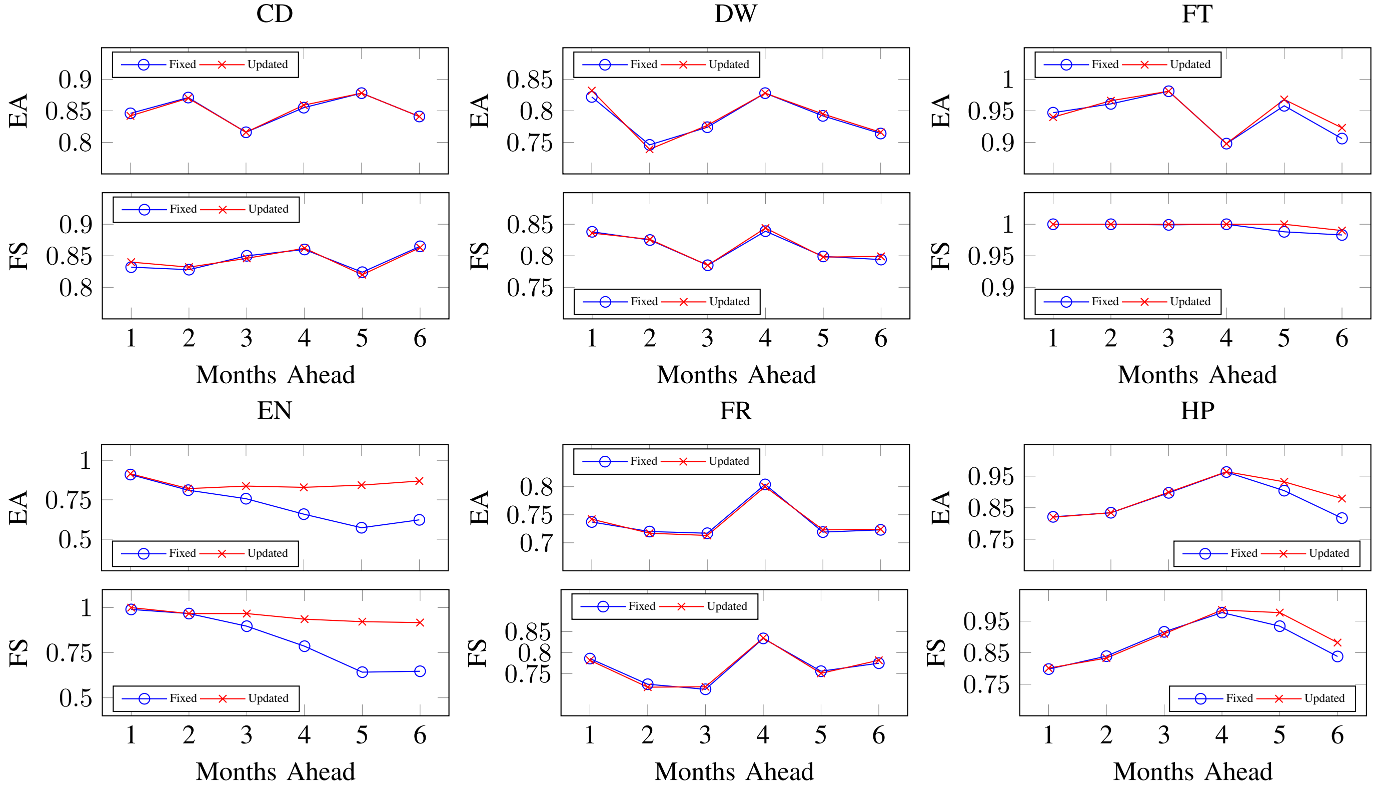

Figure 7. Appliance-wise accuracy of the proposed approach in Scenario A (Fixed Parameters) and Scenario B (Updated Parameters).

dynamics. Our algorithm effectively disambiguates appliances with similar operating modes, ensures reliable parameters estimation, and leads to accurate results, even when relatively small training data at a low sampling rate are considered. When considering very large samples with resolution in the order of few seconds, specialized solvers are needed. One possible lead of research may be to relax the binary constraints in the BQP formulation and solve the resulting convex relaxation by using fast algorithms such as ADMM [48]. However, we leave investigating convex relaxations of our NILM algorithm for future work.

Acknowledgments

We would like to thank the reviewers for their thoughtful comments that greatly helped to improve the manuscript.

References

[1] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[2] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.

[3] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

[4] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1739–1747, 2016.

[5] T. Hassan, F. Javed, and N. Arshad, "An empirical investigation of vi trajectory based load signatures for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 870–878, 2013.

[6] D. Chowdhury and M. M. Hasan, "Non-intrusive load monitoring using ensemble empirical mode decomposition and random forest classifier," in *Proceedings of the International Conference on Digital Image and Signal Processing (DISP), Oxford, UK*, 2019, pp. 29–30.

[7] Y.-H. Lin and M.-S. Tsai, "Non-intrusive load monitoring by novel neuro-fuzzy classification considering uncertainties," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2376–2384, 2014.

[8] G. Elafoudi, L. Stankovic, and V. Stankovic, "Power disaggregation of domestic smart meter readings using dynamic time warping," in *2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*. IEEE, 2014, pp. 36–39.

[9] B. Liu, W. Luan, and Y. Yu, "Dynamic time warping based non-intrusive load transient identification," *Applied energy*, vol. 195, pp. 634–645, 2017.

[10] B. Zhao, K. He, L. Stankovic, and V. Stankovic, "Improving event-based non-intrusive load monitoring using graph signal processing," *IEEE Access*, vol. 6, pp. 53 944–53 959, 2018.

[11] S. M. Tabatabaei, S. Dick, and W. Xu, "Toward non-intrusive load monitoring via multi-label classification," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 26–40, 2016.

[12] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Artificial intelligence and statistics*. PMLR, 2012, pp. 1472–1482.

[13] W. Kong, Z. Y. Dong, D. J. Hill, J. Ma, J. Zhao, and F. Luo, "A hierarchical hidden markov model framework for home appliance modeling," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3079–3090, 2016.

[14] R. Bonfigli, E. Principi, M. Fagiani, M. Severini, S. Squartini, and F. Piazza, "Non-intrusive load monitoring by using active and reactive power in additive factorial hidden markov models," *Applied Energy*, vol. 208, pp. 1590–1607, 2017.

[15] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, 2012.

[16] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2575–2585, 2015.

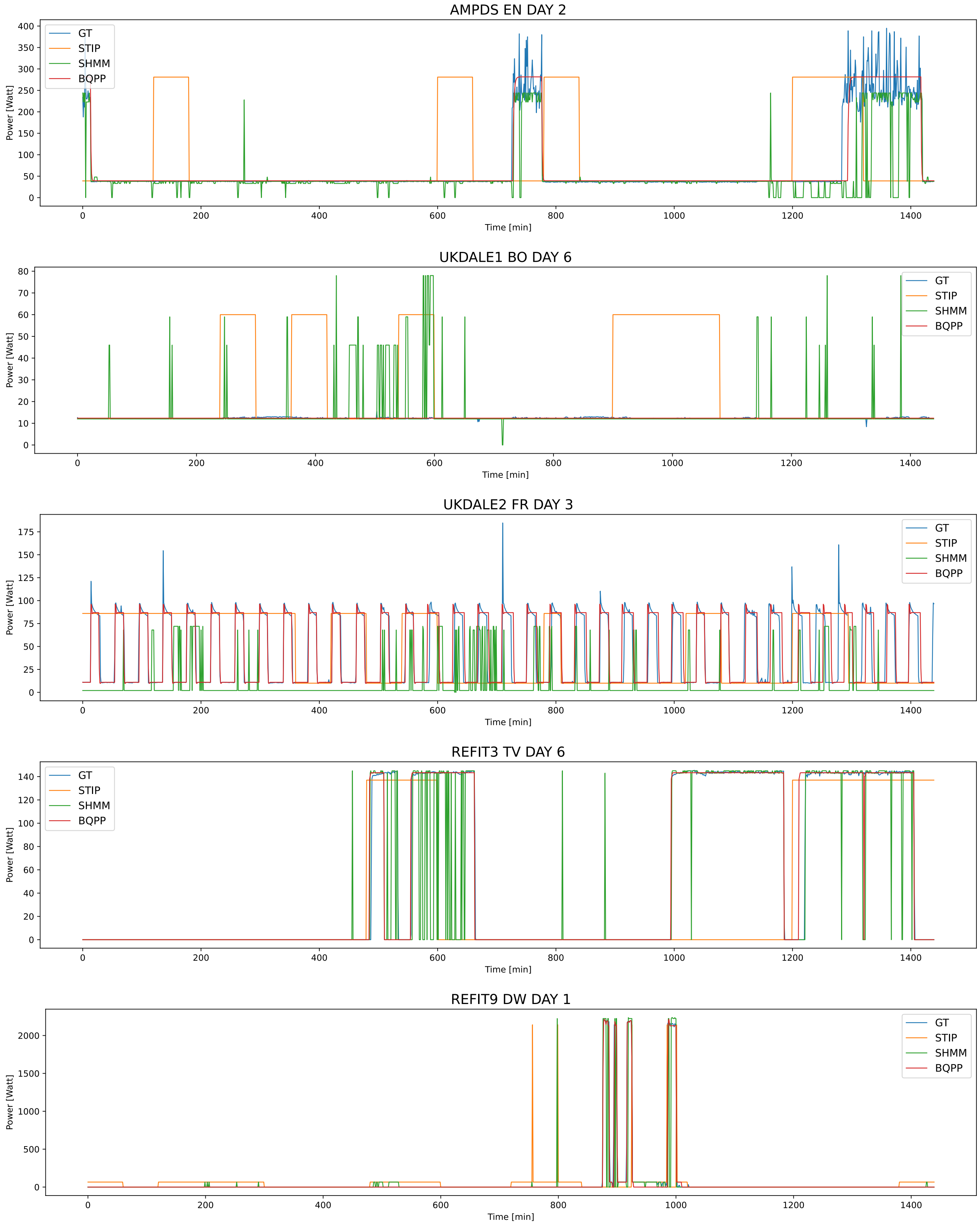

Figure 8. Ground truth (GT) and estimated power consumption of different appliances.

[17] A. Rahimpour, H. Qi, D. Fugate, and T. Kuruganti, "Non-intrusive energy disaggregation using non-negative matrix factorization with sum-to-k constraint," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4430–4441, 2017.

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

[19] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware energy disaggregation using adaptive bidirectional lstm models," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3054–3067, 2020.

[21] V. Piccialli and A. M. Sudoso, "Improving non-intrusive load disaggregation through an attention-based deep neural network," *Energies*, vol. 14, no. 4, p. 847, 2021.

[22] C. Shin, S. Rho, H. Lee, and W. Rhee, "Data requirements for applying machine learning to energy disaggregation," *Energies*, vol. 12, no. 9, p. 1696, 2019.

[23] L. Mauch and B. Yang, "A new approach for supervised power disaggregation by using a deep recurrent lstm network," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2015, pp. 63–67.

[24] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito, "Nonintrusive appliance load monitoring based on integer programming," in *2008 SICE Annual Conference*. IEEE, 2008, pp. 2742–2747.

[25] D. Egarter and W. Elmenreich, "Load disaggregation with metaheuristic optimization," in *Energieinformatik*, 2015, pp. 1–12.

[26] M. Z. A. Bhotto, S. Makonin, and I. V. Bajić, "Load disaggregation based on aided linear integer programming," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 7, pp. 792–796, 2016.

[27] D. Piga, A. Cominola, M. Giuliani, A. Castelletti, and A. E. Rizzoli, "Sparse optimization for automated energy end use disaggregation," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 1044–1051, 2015.

[28] F. M. Wittmann, J. C. López, and M. J. Rider, "Nonintrusive load monitoring algorithm using mixed-integer linear programming," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 180–187, 2018.

[29] S. Zeinal-Kheiri, A. M. Shotorbani, and B. Mohammadi-Ivatloo, "Residential load disaggregation considering state transitions," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 743–753, 2019.

[30] Y.-H. Lin and M.-S. Tsai, "Development of an improved time–frequency analysis-based nonintrusive load monitor for load demand identification," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 6, pp. 1470–1483, 2013.

[31] C. Lu, L. Ma, T. Xu, G. Ding, C. Wu, and X. Jiang, "Non-intrusive load monitoring method based on improved differential evolution algorithm," in *2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*. IEEE, 2019, pp. 279–283.

[32] D. Egarter, A. Sobe, and W. Elmenreich, "Evolving non-intrusive load monitoring," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2013, pp. 182–191.

[33] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[34] W. Wichakool, Z. Remscrim, U. A. Orji, and S. B. Leeb, "Smart metering of variable power loads," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 189–198, 2014.

[35] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear gauss–seidel method under convex constraints," *Operations research letters*, vol. 26, no. 3, pp. 127–136, 2000.

[36] S. Z. Selim and M. A. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Transactions on pattern analysis and machine intelligence*, no. 1, pp. 81–87, 1984.

[37] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[38] V. Piccialli, A. M. Sudoso, and A. Wiegele, "Sos-sdp: an exact solver for minimum sum-of-squares clustering," *arXiv preprint arXiv:2104.11542*, 2021.

[39] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proceedings of the 2011 SIAM international conference on data mining*. SIAM, 2011, pp. 747–758.

[40] S. Makonin and F. Popowich, "Nonintrusive load monitoring (nilm) performance evaluation," *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, 2015.

[41] C. Shin, S. Rho, H. Lee, and W. Rhee, "Data requirements for applying machine learning to energy disaggregation," *Energies*, vol. 12, p. 1696, 05 2019.

[42] R. Fourer, D. M. Gay, and B. W. Kernighan, "A modeling language for mathematical programming," *Management Science*, vol. 36, no. 5, pp. 519–554, 1990.

[43] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: http://www.gurobi.com

[44] C. Klemenjak, C. Kovatsch, M. Herold, and W. Elmenreich, "A synthetic energy dataset for non-intrusive load monitoring in households," *Scientific Data*, vol. 7, no. 1, pp. 1–17, 2020.

[45] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajić, "Ampds: A public dataset for load disaggregation and eco-feedback research," in *2013 IEEE Electrical Power & Energy Conference*. IEEE, 2013, pp. 1–6.

[46] J. Kelly and W. Knottenbelt, "The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes," *Scientific data*, vol. 2, p. 150007, 2015.

[47] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Scientific data*, vol. 4, no. 1, pp. 1–12, 2017.

[48] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.