

A Generic Optimisation-based Approach for Improving Non-intrusive Load Monitoring

¹Kanghang He, ²Dusan Jakovetic, ¹Bochao Zhao, ¹Vladimir Stankovic, ¹Lina Stankovic and ^{3,4}Samuel Cheng

¹Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK

²Department of Math. and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia

³Department of Computer Science and Technology, Tongji University, Shanghai, China

⁴School of Electrical and Computer Engineering, University of Oklahoma, Tulsa, USA

Abstract—The large-scale deployment of smart metering worldwide has ignited renewed interest in electrical load disaggregation, or non-intrusive load monitoring (NILM). Most NILM algorithms disaggregate one appliance at a time, remove the estimated appliance contribution from the total load, and then move on to disaggregate the next appliance. On one hand, this is efficient since multi-class classification is avoided and analytical models for each appliance can be developed independently of other appliances with the benefit of being transferred to unseen houses that have different sets of appliances. On the other hand, however, these methods can significantly under- or over- estimate the total consumption since they do not minimise the difference between the measured aggregate load and the sum of estimated individual loads. Motivated by minimising the latter difference without losing the benefits of existing NILM algorithms, we propose novel post-processing approaches for improving the accuracy of existing NILM. This is posed as an optimisation problem to refine the final NILM result using regularisation, based on the level of confidence in the original NILM output. First, we propose a heuristic method to solve this (combinatorial) boolean quadratic problem through relaxing zero-one constraint sets to compact zero-one intervals. Convex-based solutions, including norm-1, norm-2 and semi-definite programming-based relaxation, are proposed trading off accuracy and complexity. We demonstrate good performance of the proposed post-processing methods, applicable to any event-based NILM, compared with 4 state-of-the-art benchmarks, using public REFIT and REDD electrical load datasets.

Index Terms—energy disaggregation, smart metering, NILM

I. INTRODUCTION

The key motivator for ongoing large-scale smart metering deployments worldwide [1], [2] is to maximise benefits of the smart grid. Smart meter data has been shown to improve grid operation and maintenance of distribution networks [3], fault detection [4], non-technical loss detection [5], outage prediction [6], load forecasting [7], demand response [8] and improving customer satisfaction, including accurate billing and meaningful energy feedback via Non-Intrusive Load Monitoring (NILM), that is, disaggregating the total household consumption down to the load level [9], [10], [11]. Hence, smart meter data analytics are critical to the success of the smart grid [12].

While this paper focuses on NILM, the proposed methods are also applicable to the broader smart meter analytics community, many of which rely on optimisation methods, e.g., demand response management [8], electricity theft detection [13], and power flow optimisation in microgrids [14].

Estimates of individual load consumption can be obtained by submetering. However, this is intrusive and non-scalable, particularly for the residential sector, since this requires additional appliance-level meters, beyond a standard smart meter. An alternative approach, i.e., NILM, is to estimate consumption of individual loads directly from aggregate, smart meter readings through algorithms designed for this purpose. NILM can thus support energy efficiency measures [15], demand side management including load shifting [16], anomaly detection [17] and many other applications that require load-level energy consumption.

While research on NILM has primarily revolved on high frequency load measurements, the lower rates provided by smart metering initiatives [1], [2] are driving research into low-rate smart meter datasets [18] and low-rate NILM methods [10], [16]. Low-rate NILM is particularly challenging due to noise from unknown appliances, signal transients that act as noise, load fluctuations, and the fact that the average household owns over 40 appliances.

NILM methods can be classified as *state-based* or *event-based* methods. State-based NILM approaches represent the operation of an appliance as a finite state machine and perform disaggregation based on the state transition model learned during training. The most common representations are Hidden Markov Model (HMM)-based methods [19] and their variants, including Factorial HMM (FHMM) [20], Additive Factorial HMM with quadratic programming [21] or semi-definite programming relaxation (SDR) relaxation [22], Super-state HMM [9], and Latent Bayesian modelling [23]. Though HMM-based approaches in some cases provide state-of-the-art results, they require a large amount of relatively noiseless appliance training data to construct state transition models. Hence, they perform poorly in the presence of noise, caused by unknown appliances [16]. The complexity is exponential with the number of loads, and they struggle when appliance state durations vary significantly [24].

Event-based NILM approaches [25], [26], [27], on the other hand, are based on detecting the *event* of appliance being switched on or off, and then classifying the extracted features, where each class corresponds to one appliance. These methods usually consist of three steps: (1) event detection: detecting changes in time-series aggregate signal due to one or more appliances changing their state [28]; (2) feature extraction: the electrical features, such as active power, duration, min/max

value, etc., are isolated for each event; (3) classification and pattern matching, used to classify the events using extracted features into pre-defined categories, each corresponding to one appliance, that are learned during training. The multi-state appliances are usually treated as multiple single-state appliances, where each state is considered as a single appliance and afterwards the contribution of all the states are summed to obtain an estimate of the total appliance consumption. Various classification methods have been used, including support vector machines (SVM) [29], neural networks [30], non-negative tensor factorization [31], k-means [32], decision trees (DT) [33], and Graph Signal Processing (GSP) [10].

Event-based NILM methods are popular due to their low complexity, since a simple edge detection is needed followed by conventional classification. However, most event-based NILM methods disaggregate one appliance at a time, and do not check if the sum of the disaggregated loads is approaching the true measured result, e.g., [19], [20], [33], [34], [29], [35], [36], [37]. This is a preferred approach (vs. disaggregating all appliances at once) since it facilitates transfer learning [38] (i.e., applying the developed appliance models to ‘unseen’ houses) and avoids multi-class classification [9], [20], [39], [25], [40], which is often less robust to noise. However, disaggregating one appliance at a time, potentially results in significant load over/under estimation [31]. By monitoring how close the sum of the disaggregated loads is to the aggregate value after a first pass at NILM, that is, through *post-processing*, one can potentially improve the accuracy of current NILM methods without losing their advantages.

Prior work on post-processing NILM output has focused on manually checking if the disaggregated result is within acceptable limits, e.g., [16], [41], or is proposed as part of a specific NILM algorithm [10], [21], [42], [23]. In contrast, we propose a *generic* method to improve the accuracy after conventional NILM is applied, that does not require any manual intervention, by casting our post-processing problem as an optimisation problem that aims to minimise the distance between the sum of the disaggregated loads and the total measured consumption. A regularisation term is added to assign a weight based on confidence in the accuracy of the initial disaggregation result for each appliance.

The resulting optimisation problem is a boolean quadratic problem (combinatorial in nature) with zero-one type constraints, that belongs to the class of NP-hard problems [43], [44]. By adapting recent convex optimisation methods [43], [45], we provide three approaches to solve the posed optimisation problem which trade off complexity and accuracy. The first two methods are based on relaxation which changes the constraint of the optimisation problem to a soft real value between zero and one: the first, more complex approach, uses norm-2 minimisation, while the second, reduces the complexity, by casting the problem as a norm-1-type minimisation (see, e.g., [45]). After the optimal solution is found, the result is projected back to 0 or 1, entry-wise. The third approach is based on semi-definite programming relaxation (SDR) [43].

Our motivation for using SDR comes from its demonstrated high accuracy in many similar problems, such as electricity theft detection with smart meters [13], power flow optimisation

in microgrids [14], and finding approximate solution with FHMM-based NILM [22], where the quadratic programming relaxation of [21] is replaced by SDR to boost performance.

Optimisation methods have been used to address the NILM problem by minimising the difference between power consumption of the detected events and all possible combinations of loads [46], [24], [47]. Though various optimisation methods have been proposed, e.g., integer programming [48] and genetic algorithms [49], noise, unknown loads, and similar appliance signatures render these approaches ineffective (see the last paragraph of Section IIIA). Our proposed methods differ from the aforementioned approaches, since optimisation is used to refine or *post-process* the NILM result by regularising the cost function with our *confidence* in the initial NILM estimate.

To the best of our knowledge, this is the first attempt to develop a generic approach that can be used to improve the output of NILM, yielding the following contributions:

- a novel modelling method of how one can incorporate systematically a first-pass NILM and post-processing into a common framework leading to a clear mathematical formulation;
- novel adaptive parameter selection based on the level of confidence in the first-pass NILM output for each appliance;
- three proposed approximate solutions to the formulated optimization problem based on convex relaxation and convex optimization tools [43], [45];
- demonstrating improvements of the proposed generic post-processing methods on two state-of-the-art first-pass NILM methods using two public electrical load measurement datasets;
- demonstrating improvements of the proposed post-processing methods with four state-of-the-art post-processing methods [10], [21], [42];
- a detailed analysis, in terms of accuracy and complexity, of the suitability of the proposed and existing post-processing NILM methods for different appliance combinations contributing to the aggregate load.

II. BACKGROUND

In this section we review state-of-the-art post-processing approaches in order to identify the research gap and how our contributions addresses this.

Despite significant progress made in recent years to crack the low-rate NILM problem suited for widespread national smart metering programmes [9], [10], [32], [16], [50], [51], [52], state-of-the-art low-rate NILM methods still do not demonstrate acceptable levels of accuracy, scalability and complexity necessary for widespread deployment.

Therefore most approaches have revolved around improving low-rate NILM accuracy and complexity. Event-based NILM, as discussed in the previous section, has gained significant traction for low-rate NILM due to its relatively lower complexity and robustness to noise compared to state-based methods.

Only recently has post-processing of NILM, that is, improving the accuracy of NILM output, been gaining attention

as a way of leveraging on NILM advantages and improving accuracy in a targeted manner by observing the results of the first-pass of NILM.

While [41] and [16] manually check the appliance power level and operation time after disaggregation and retain only the estimates that are within expected limits, they do not apply any post-processing method to improve the estimated result. In [10], Simulated Annealing (SA) is used to minimise the difference between the sum of estimated power of appliances and actual measurements. SA is a probabilistic technique for approximating the global optimum, which randomly searches around the starting points obtained from NILM and updates the result when improvements are observed in the objective function. Though SA can find the optimal result, it usually requires many iterations to converge to the global optimum. Moreover, often SA results in a local minimum, away from the globally optimal result.

Additive Factorial Approximate MAP (AFAMAP) in Additive Factorial HMM [21], [23] compares the observation (aggregate measurement) with the sum of disaggregated loads. Additionally, in [21], Branch and bound (BNB) is used to refine the results of AFAMAP where the difference between the sum of estimated and aggregate power is minimised. BNB is a well-investigated method for discrete and combinatorial optimisation problems, that partitions the solution into two branches and recalculates the objective function; depending on the obtained value, one branch is chosen to continue the partition until the optimal result is found. BNB finds the global optimum within an accuracy of ϵ , but is often slow. Indeed, the worst case complexity is comparable to the full search.

More recently, in [42], a GSP-based post-processing (GSP-P) method is proposed to refine the disaggregation results by matching the falling and rising edges obtained by edge detection. For each rising edge, a graph is generated and according to minimisation of graph signal total variation all possible candidate rising edges are picked. Due to high complexity, the method is used only for appliances that tend to be confused with other appliances with similar power levels.

However, the above post-processing methods are designed for specific NILM algorithms, whereas we propose generic post-processing methods for improving the result of *any* event-based NILM approach. We cast the NILM post-processing problem as an optimisation problem with a regularisation term that depends on the output of the original NILM, and propose solutions to optimally and automatically tune the regularisation parameters by adapting three state-of-the-art convex optimisation methods: two relaxation-based methods and a third on SDR. These convex-based relaxed optimisation solutions (polynomial) are drastically smaller than solving the original NP hard problem. Complexity constraints are addressed through the proposed norm-2 and norm-1 approaches. We analyse how the different relaxations compare in terms of both complexity and accuracy for our NILM post-processing problem.

III. NOTATION AND PROBLEM FORMULATION

A. Notation

The task of NILM is to estimate individual loads contributing to the aggregate meter data. Focusing on the most common case when the meter measures only active power, the aggregate reading of the meter at time sample j can be expressed as:

$$P_j = \sum_{m=1}^M P_j^m + n_j, \quad (1)$$

where $j = 1, \dots, N$ and $m = 1, \dots, M$ with N being the total number of samples and M the number of known appliances. In Eq. (1), P_j and P_j^m are the total aggregate power and power consumption of Appliance m at time sample j , respectively; n_j is the noise that includes measurement errors, base-load and all unknown appliances in the on state at time sample j . The NILM task is to estimate all P_j^m , given P_j .

Let $\Delta P_j = P_{j+1} - P_j$ be the change of the aggregate power signal. Let w be a threshold, such that if $|\Delta P_j| > w$ an event is detected, i.e., an appliance changed its state, e.g., switched on or off. Let E_i denote the i -th event, where $i = 1 \dots, N_E$ with N_E being the total number of events in a processing window. We set the value of E_i to j if Event i is detected at time instant j . Once events are identified, these events are classified and appliance consumption determined.

Let $\alpha_i^m = 1$ if, after NILM, it is predicted that Event i is caused by Appliance m changing its state, and $\alpha_i^m = 0$, otherwise. When $\alpha_i^m = 1$, if $\Delta P_{E_i} > 0$, the detected edge is a falling edge, otherwise, it is a rising edge. Based on this rule, we set $S_j^m = 1$, if Appliance m is running at time sample j , or $S_j^m = 0$ otherwise. Note that, given α_i^m (which are results of NILM), we predict the state S_j^m of Appliance m . For example, $\alpha_2^1 = 1$ and $\alpha_3^1 = 1$ indicate that Appliance 1 changed its state during Events 2 and 3. Suppose that $\Delta P_{E_2} > 0$ and $\Delta P_{E_3} < 0$, then the power change at Event 2 corresponds to a rising edge and Event 3 to a falling edge, indicating the time when the appliance was most likely switched on and off, leading to $S_j^1 = 1$, for time interval $j \in [E_2, E_3]$.

Given the average working power \overline{P}^m of Appliance m , obtained by training on sub-metering data or using the appliance manual, we can estimate the power consumed by this appliance at each time sample j as $\hat{P}_j^m = \overline{P}^m S_j^m$. We can also express similarly estimated power signal change $\Delta \hat{P}_i^m = \overline{\Delta P}^m \alpha_i^m$ with $\overline{\Delta P}^m$ being the average of power change value $|\Delta P^m|$. We assume that only a single appliance triggers an event.

If NILM is successful, the sum of estimated power consumption of all appliances should be close to the aggregate power, that is,

$$\sum_{j=1}^N |P_j - P_j^0 - \sum_{m=1}^M S_j^m \overline{P}^m|^2. \quad (2)$$

should be almost zero, where P_j^0 is the estimated base-load at time sample j . Following terminology of [31], [10], we will refer to the above term as the *fidelity term*, which represents the difference between aggregate power without the base-load, i.e., $P_j - P_j^0$, and the sum of the loads estimated by NILM, i.e., $\sum_{m=1}^M S_j^m \overline{P}^m$.

The next logical step is to minimise Eq. (2) over all possible S_j^m . However, there are several reasons why minimising Eq. (2) is not a good idea. First, we cannot distinguish two appliances with similar working power $\overline{P^m}$ by minimising the fidelity term alone. Secondly, the fluctuations of power values around the mean $\overline{P^m}$ during the appliance operation is ignored. Thirdly, the sum of multiple appliance loads might be close to another load, leading to wrong minimisation. Finally, noise including measurement errors and unknown appliances is not taken into account.

B. Post-processing NILM: Proposed Problem Formulation

Instead of minimising Eq. (2) over all possible solutions, we assume that a NILM method has been applied to lead to a solution S_i^{m*} , for which, the fidelity term is fixed as:

$$\sum_{j=1}^N |P_j - P_j^0 - \sum_{m=1}^M S_j^{m*} \overline{P^m}|^2. \quad (3)$$

Starting from S_i^{m*} , in [10], SA is used to minimise the fidelity term by updating S_i^{m*} . Note that SA usually updates only several appliance states (appliances with unique high working power) to correct misclassification of the employed NILM.

To improve the reliability and accuracy of this *post-processing* step, we introduce the disaggregation result as a regularisation term. Then the optimisation problem becomes:

$$\min_{S_j^m \in \{0,1\}} \sum_{j=1}^N |P_j - P_j^0 - \sum_{m=1}^M S_j^m \overline{P^m}|^2 + \sum_{j=1}^N \sum_{m=1}^M \lambda_m |S_j^m - S_j^{m*}|^2 \quad (4)$$

where MN optimisation variables, S_j^m , take values from a discrete set (0 or 1), and $\lambda_m \geq 0$ is the weight of the regularisation term for Appliance m . (Here, again, S_j^{m*} , $j = 1, \dots, N$, is the estimate obtained by an initial NILM method used.) In this optimisation set-up, the fidelity term shows how far the result is from the observation, while the regularisation term weights our confidence in the original NILM output. Large λ_m means we have more confidence in the results of the original NILM for Appliance m . Small λ_m means that we have less confidence in the NILM result, and put more weight in minimising the fidelity term. Note that λ_m is appliance dependent, to reflect the case that a NILM method has different accuracy for different appliances.

To reduce the computational complexity and considering that the NILM algorithm will provide edge detection results α^* , we modify the objective function as:

$$\min_{\alpha_i^m \in \{0,1\}} \sum_{i=1}^{N_E} \left| |\Delta P_{E_i}| - \sum_{m=1}^M \alpha_i^m \overline{\Delta P^m} \right|^2 + \sum_{i=1}^{N_E} \sum_{m=1}^M \lambda_m |\alpha_i^m - \alpha_i^{m*}|^2 \quad (5)$$

to only optimise for sample i when the events are detected. The minimisation here is with respect to MN_E discrete variables α_i^m taking values 0 or 1. Since N is usually large, while number of events N_E is much smaller (i.e., appliances are rarely switched on/off), this significantly reduces complexity, and largely eliminates noise and fluctuations during appliance operation.

Besides testing the original SA, as proposed in [10], we also use, in Section V, SA and BNB to find the values of α_i^m that minimise Eq. (5) starting from α_i^{m*} .

IV. PROPOSED SOLUTIONS

In this section we provide three solutions to the optimisation problem Eq. (5) or a related problem (see ahead Eq. (6)).

A. Problem relaxation

Eq. (5) is a boolean (combinatorial) quadratic problem that is known to be hard to solve exactly [43]. To solve efficiently this optimisation problem, we introduce relaxation, that is, instead of being one or zero, α_i^m in Eq. (5) takes soft real-number values in the set $[0, 1]$. This way, we can convert the minimisation problem in (5) to a convex optimisation problem, which enables the use of known convex optimisation tools (a problem with convex quadratic cost and box constraints).

To solve Eq. (5), the infeasible path-following algorithm [53] is used based on two Newton steps per iteration, which always finds a non-negative solution, and is implemented in CVX, a package for specifying and solving convex programs [54], [55]. For large-scale problems, one can also implement other efficient methods such as [56].

After the above method is applied, and a solution $\alpha_i^m \in [0, 1]$ to the relaxed version of the problem in Eq. (5) is obtained, we replace the obtained α_i^m with $\alpha_{i,final}^m = 1$ if $\alpha_i^m > 0.5$, and with $\alpha_{i,final}^m = 0$, otherwise. In other words, we project the solution back to the discrete set $\{0, 1\}$.

We also consider an optimisation problem with a modified, ℓ_1 -norm type regularisation; see [45] for a similar regularisation in a different, compressed sensing, context. The main motivation for this is that both the output of the NILM and post-processing NILM are usually expected to be sparse, i.e., they have many zeros and a few non-zero entries. (Appliances generally rarely change state.) It is well known that ℓ_1 -type regularisation yields sparse solutions. This provides a motivation to attempt to improve the NILM post-processing by moving from ℓ_2 -norm-type regularisation in Eq. (5) to ℓ_1 -norm-type regularisation in Eq. (6). Note that the optimisation variable in Eq. (6) is the same as in Eq. (5), and the regularisation coefficients λ_m 's remain the same.

$$\min_{\alpha_i^m \in \{0,1\}} \sum_{i=1}^{N_E} \left| |\Delta P_{E_i}| - \sum_{m=1}^M \alpha_i^m \overline{\Delta P^m} \right|^2 + \sum_{i=1}^{N_E} \sum_{m=1}^M \lambda_m |\alpha_i^m - \alpha_i^{m*}|. \quad (6)$$

Note that each time the objective function in Eq. (6) is calculated, $N_E M$ fewer multiplication operations are needed compared to (5).

B. Semi-definite programming-based relaxation (SDR)

SDR is a powerful, computationally efficient approximation technique for a class of combinatorial optimisation problems that finds a wide range of applications [43], [13], [14], [22]. To make our optimisation suitable for SDR, we adapt Eq. (5) as follows. First we introduce the new optimisation variable $z_i^m = 2\alpha_i^m - 1$. Then the optimisation problem becomes:

$$\min_{z_i^m \in \{-1,+1\}} \sum_{i=1}^{N_E} \left| g_i - \sum_{m=1}^M z_i^m r_i^m \right|^2 + \sum_{i=1}^{N_E} \sum_{m=1}^M \lambda'_m |z_i^m - z_i^{m*}|^2, \quad (7)$$

where $g_i = |\Delta P_{E_i}| - \frac{1}{2} \sum_{m=1}^M |\overline{\Delta P^m}|$, $r_i^m = \frac{\overline{\Delta P^m}}{2}$, $z_i^{m*} = 2\alpha_i^{m*} - 1$ and $\lambda'_m = \lambda_m/4$. Eq. (7) has the optimisation

variable z_i^m , where z_i^m takes either value -1 or +1. Note that Problem Eq. (7) is equivalent to Eq. (5). For example, once the optimal value of z_i^m is obtained, α_i^m can be recovered as: $\alpha_i^m = 0.5(z_i^m + 1)$.

We now express Eq. (7) in the standard format of a boolean quadratic program, so that the SDR method [43] can be applied. To this end, we construct a new $N_E M$ length vector with all elements from z_i^m in a fixed order $\mathbf{z} = [z_1^1, z_2^1, \dots, z_{N_E}^1, z_1^2, z_2^2, \dots, z_{N_E}^2, \dots, z_1^M, z_2^M, \dots, z_{N_E}^M]^\top$. Then (7) is equivalently expressed as:

$$\min_{\mathbf{z} \in \{-1, +1\}^{N_E \times M}} \|\mathbf{B}\mathbf{z} - \mathbf{d}\|_2^2, \quad (8)$$

where $\mathbf{d} = [\mathbf{g}, \sqrt{\lambda_1'} \mathbf{z}^1, \sqrt{\lambda_2'} \mathbf{z}^2, \dots, \sqrt{\lambda_M'} \mathbf{z}^M]^\top$ with $\mathbf{g} = [g_1, g_2, \dots, g_{N_E}]$ and $\mathbf{z}^m = [z_1^m, z_2^m, \dots, z_{N_E}^m]$, and

$$\mathbf{B} = \begin{bmatrix} \mathbf{r}^1 \circ \mathbf{I} & \mathbf{r}^2 \circ \mathbf{I} & \dots & \mathbf{r}^M \circ \mathbf{I} \\ \sqrt{\lambda_1'} \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \sqrt{\lambda_2'} \mathbf{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \sqrt{\lambda_M'} \mathbf{I} \end{bmatrix} \quad (9)$$

where \mathbf{r}^m is an $N_E \times N_E$ matrix with all elements in the i th row equal to r_i^m , \mathbf{I} is an $N_E \times N_E$ identity matrix, and $\mathbf{0}$ is an $N_E \times N_E$ zero matrix. Here, \circ is the Hadamard product operation of two matrices.

Following [43] (see [43] for more details on the derivation of the involved algorithmic steps), we relax the problem of Eq. (8) into the following problem:

$$\begin{aligned} \min_{\mathbf{y} \in \mathbb{R}^{N_E M + 1}} \{\mathbf{y}^\top \mathbf{G} \mathbf{y}\} \\ \text{s.t.} \quad y_j^2 = 1, j = 1, \dots, N_E M + 1 \end{aligned} \quad (10)$$

where matrix \mathbf{G} is given by:

$$\mathbf{G} = \begin{bmatrix} \mathbf{B}^\top \mathbf{B} & -\mathbf{B}^\top \mathbf{d} \\ -\mathbf{d}^\top \mathbf{B} & \|\mathbf{d}\|_2^2 \end{bmatrix}.$$

Then we solve the semi-definite relaxation of problem (10):

$$\begin{aligned} \min \quad \text{tr}(\mathbf{G}\mathbf{Y}) \\ \text{s.t.} \quad \mathbf{Y} \geq 0, \quad \mathbf{Y} = \mathbf{Y}^\top \\ \mathbf{Y}_{i,i} = 1, \forall i = 1, \dots, N_E M + 1. \end{aligned} \quad (11)$$

where $\mathbf{Y} = \mathbf{y}\mathbf{y}^\top$. Here, notation $\mathbf{Y} \geq 0$ means that matrix \mathbf{Y} is positive semi-definite (PSD). (11) is a semi-definite program, and hence it can be efficiently solved. We solve it here numerically using CVX. Compared to the previous convex relaxation solving Eq. (5) or (6) with respect to $\alpha_i^m \in [0, 1]$, the optimisation variable size in Eq. (11) is squared. Hence, from a computational point of view, Eq. (5) or (6) is preferred for higher dimensions of $N_E \times M$. On the other hand, SDR may exhibit higher accuracy.

Once \mathbf{Y}^* is obtained, we need to recover the discrete $-1/+1$ variables z_i^m . This is achieved via the eigenvalue decomposition:

$$\mathbf{Y}^* = \sum_{i=1}^{N_E M + 1} \mu_i \mathbf{u}_i \mathbf{u}_i^\top, \quad (12)$$

where, μ_i is the i -th largest eigenvalue of \mathbf{Y} , and \mathbf{u}_i is the unit-norm eigenvector of \mathbf{Y}^* that corresponds to the eigenvalue μ_i .

Next, an $N_E M + 1$ length vector \mathbf{y}^* is set to $\sqrt{\mu_1} \mathbf{u}_1$, and we calculate an intermediate solution variable $\hat{\mathbf{y}}$ as:

$$\hat{y}_j = \begin{cases} +1, & \text{if } y_j^* > 0 \\ -1, & \text{otherwise} \end{cases} \quad \text{for } j = 1, \dots, N_E M + 1. \quad (13)$$

Since only the leading eigenvalue-eigenvector pair (u_1, μ_1) is needed, it is not necessary to perform the full eigenvalue decomposition, thus significantly reducing the computational cost of this algorithmic step. Then, we calculate $\hat{\mathbf{z}}$:

$$\hat{z}_j = \begin{cases} \hat{y}_j, & \text{if } \hat{y}_{N_E M + 1} = 1 \\ -\hat{y}_j, & \text{if } \hat{y}_{N_E M + 1} = -1 \end{cases} \quad \text{for } j = 1, \dots, N_E M. \quad (14)$$

and change $\hat{\mathbf{z}}$ back to the event classifier $\hat{\alpha}$:

$$\hat{\alpha} = \frac{\hat{\mathbf{z}} + 1}{2}. \quad (15)$$

such that the final NILM output is:

$$\alpha_i^m = \hat{\alpha}_{(m-1) \cdot N_E + i}. \quad (16)$$

C. Adaptive calculation of the regularisation term weight

We recommend the following heuristic choice for tuning parameters λ_m :

$$\lambda_m = \frac{\theta^2}{\Delta P_m^2} \frac{\beta}{\min_{n \in [1, M], n \neq m} \left(|\Delta P^n| - |\Delta P^m| \right)}. \quad (17)$$

Two parameters θ and β are used to balance the weight of the regularisation term. We set $\theta = \overline{|\Delta P_{app}|}$, where operand $\overline{[x]}$ rounds a positive number x to the nearest power of 10 and $\overline{\Delta P_{app}} = \frac{\sum_{m=1}^M |\Delta P^m|}{M}$. Similarly, we set $\beta = \overline{|\Delta P_{all}|}$, where $\overline{\Delta P_{all}}$ is the average value of $|\Delta P_i|$. Note that the accuracy of the final result is not very sensitive to the choice of λ_m , hence θ is rounded to the nearest power of 10.

λ_m is inversely proportional to the appliance mean power, which implies that for high loads, we put more weight on the fidelity term, since these loads contribute to the total aggregate the most. For high loads m , the first term of Eq. (17) is smaller or very close to 1. Thus, the fidelity term is given more weight. On the other hand, if an appliance has very small $|\Delta P_m|$, a larger value of λ_m indicates that the optimisation weight is on the original NILM approach, i.e., we trust the NILM output.

If there is another appliance that has average power fluctuation similar to Appliance m , the denominator of the second term of Eq. (17) is small, which suggests larger λ_m . Then, we put more weight on the original NILM result, since optimising the fidelity term would not be able to separate these two appliances. Note that β is set to be close to the average value of aggregate power change for all events detected, including noise and unknown high load appliances, which are usually larger than θ . The second term of Eq. (17) will be much larger than 1, and hence ensures that the second term of Eq. (5) is of comparable size to the fidelity term.

TABLE I: Summary of the proposed post-processing algorithms.

Proposed NILM Post-processing	Objective Function	Computational Complexity	Average Num. Iterations
Norm2	Eq. (5)	$O(N_E M)$	25
Norm1	Eq. (6)	$O(N_E M)$	15
SDR	Eq. (11)	$O((N_E M + 1)^3)$	8

D. Summary of the proposed algorithms

All three proposed solutions consist of two steps: 1) solving a convex optimisation problem; and 2) ‘‘projecting’’ the solution back to the corresponding 0-1 set. For Step 1, the algorithms guarantee to converge and to a global solution of the convex problem [43]. Step 2, a simple projection step, can always be performed. Hence, all our methods are guaranteed to be *stable* and always *converge* to the global solution of the corresponding convex relaxed problem and to a sub-optimal solution of the overall NP-Hard combinatorial problem [43].

Table I summarises the three proposed methods and compares their computational complexity per iteration [57]. The average number of iterations per window is obtained by averaging the number of iterations needed to get optimal solutions for all testing windows. SDR has the largest computational complexity, so in practice, we expect higher computational cost and longer execution time.

V. RESULTS AND DISCUSSION

We apply the proposed methods to the output of two state-of-the-art event-based NILM approaches, GSP [10] and DT [33]. We use REFIT (8sec sampling rate) [18] and REDD (1sec sampling rate) [58] datasets. For the REDD dataset, we use the first week for training and the rest for testing. For REFIT, we pick one month of data (April 2014) to test the performance and use the previous month’s measurements for training. Two houses with typical appliances, and different ‘noise level’ [9], are chosen from each dataset. The total number of considered appliances M depends on the house and is between 6 and 10.

F_M is used as evaluation metric to assess the appliance classification accuracy, as in [16], [42], [10], [20]. It is defined as $F_M = 2(PR * RE) / (PR + RE)$, where $PR = TP / (TP + FP)$, $RE = TP / (TP + FN)$, and true positive (TP) stands for the number of edges detected correctly, false positive (FP) the number of edges detected that do not actually exist, and false negative (FN) indicates the number of edges of an appliance state change that are not detected. To assess energy disaggregation accuracy, we use Accuracy ($Acc.$) [58] defined as:

$$Acc. = 1 - \frac{\sum_{j=1}^N |\hat{P}_j^m - P_j^m|}{2 \sum_{j=1}^N P_j^m}. \quad (18)$$

To reduce complexity, the datasets are split into windows, which are independently processed. For NILM norm2 and norm1 methods, we use window size $N_E = 1000$, and for NILM SDR $N_E = 100$ (due to the larger size of the optimisation variable, which is a 101×101 matrix in this case).

For benchmarking, BNB method [21] applied to Eq. (5), GSP method [42] (denoted by GSP-P) and two different SA methods are used: SA1 denotes the method of [10], i.e., using

SA to optimise the fidelity term only, and SA2 refers to the method where SA is applied to Eq. (5).

To compare the relative complexity of different post-processing approaches, we show the execution time of the proposed methods, SA1, SA2, BNB and GSP-P, in Table II. For SA [10], Eq. (5) is optimised sample by sample to improve calculation efficiency, and each sample needs more than 300 iterations to converge to the minimum. It is clear that the proposed NILM norm1 converges faster than other methods. All three proposed post-processing approaches are faster than SA, BNB and GSP-P (except SDR for REDD houses). The average execution time of BNB is over 0.5sec per sample, which implies that roughly half an hour is needed to process one month of data from a single REFIT house (with 3000 events per month). The fastest proposed method, Norm1, only needs few seconds and the slowest of the proposed methods, SDR, needs no more than 3min to complete the same task.

TABLE II: Average execution time per sample in [sec] for two REDD and two REFIT houses for seven post-processing methods.

	REDD House1	REDD House2	REFIT House2	REFIT House6
SA1	0.19	0.18	0.20	0.17
SA2	0.21	0.22	0.23	0.18
Norm2	0.009	0.008	0.008	0.008
Norm1	0.001	0.001	0.002	0.003
SDR	0.065	0.053	0.072	0.063
BNB	0.55	0.61	0.49	0.54
GSP-P	0.037	0.043	0.082	0.075

Next, we compare the accuracy of the methods. We use F to label fridge, BG bathroom GFI, K kettle, T toaster, DW dishwasher, MW microwave, WM washing machine, WD washer dryer, KO kitchen outlet, S stove, and AVG for the average accuracy across all these appliances.

Fig. 1 shows per-appliance and average results for House 2 from the REFIT dataset. Fig. 2 show results for House 6, averaged over all considered appliances. It can be seen from the figures that all post-processing methods improve the disaggregation result for all listed appliances with respect to the original NILM. The three proposed methods are more accurate than SA1 and SA2. NILM SDR has the highest accuracy for most appliances compared to the other methods. Since BNB finds the globally optimal solution of the optimisation problem it solves (to within an accuracy ϵ), it is not surprising that it has similar, or occasionally slightly better, performance than the proposed methods. Note that the optimal solution to the problem of Eq. (5) does not necessarily correspond to the most accurate result for the NILM problem. The optimisation problem Eq. (5) itself is part of the modeling approach and makes certain inherent assumptions. For example, we assume each appliance is operating at its average power with no fluctuation, which may not ideally match actual submetered load measurements. This is an explanation why NILM BNB does not always provide the best result in terms of $Acc.$ However, the main problem of the BNB approach is its high execution time. GSP-P [42] provides certain improvements for some appliances, but its average is always worse than any of the three proposed methods.

Figs. 3 and 4 show the results for REDD Houses 1 and 2, respectively. Similarly to REFIT results, NILM SDR and BNB

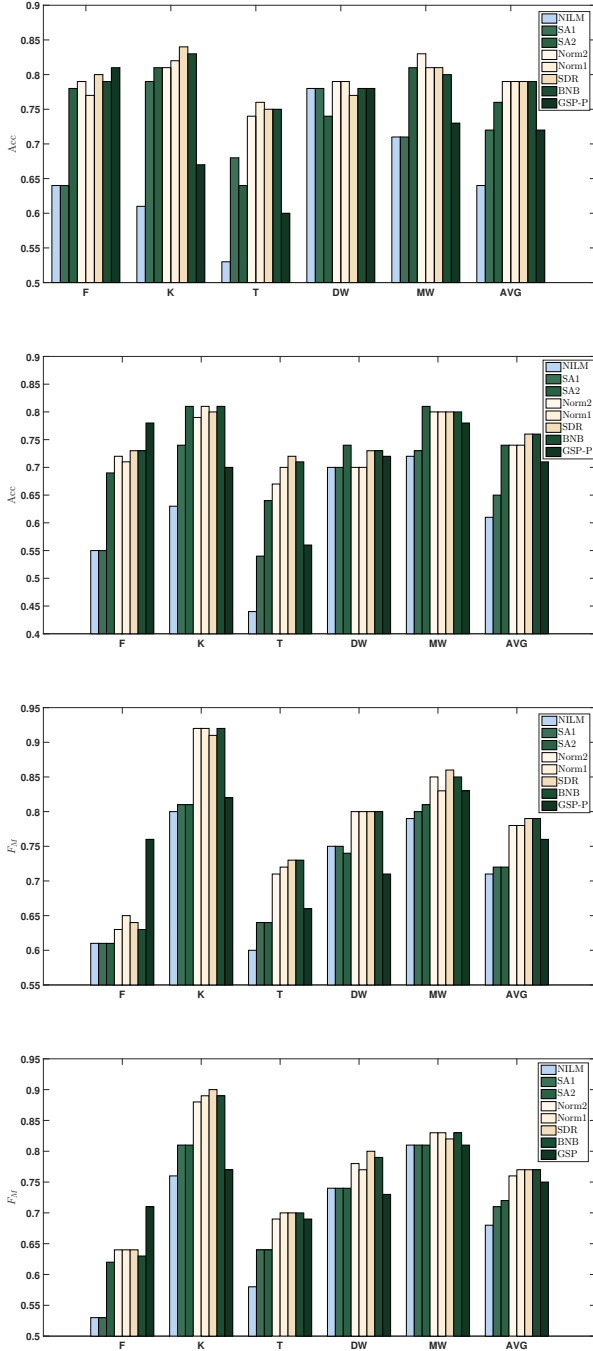


Fig. 1: Results for REFIT House2. NILM denotes the result without any post-processing, i.e., after GSP-based or DT-based NILM. From top to bottom, the figures show: Acc after GSP-based NILM; Acc after DT-based NILM; F_M after GSP-based NILM; F_M after DT-based NILM.

provide the best post-processing results for most appliances. For some appliances, such as KO in House 2, WD and F in House 1, NILM approaches without post-processing already have fairly accurate results and thus post-processing cannot improve much. For appliances such as BG and MW, which have unique and relatively high loads, we can observe significant improvement.

Based on the above results, the following conclusions can be

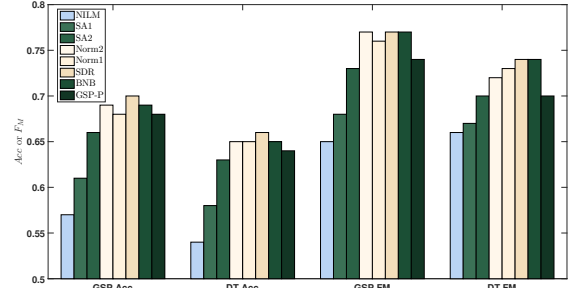


Fig. 2: REFIT House 6 results: Average performance across all appliances.

made: 1. The gain with post-processing is larger for the noisier, REFIT dataset, since for the relatively clean, REDD dataset, NILM approaches without post-processing already provide good results. 2. If an appliance is disaggregated accurately with original NILM, then post-processing does not help much (e.g., F and WD in REDD House 1, K and MW in REFIT House 2). 3. The post-processing gains are similar for the two NILM algorithms (GSP and DT). 4. All the three proposed methods (as in Table I) outperform SA and GSP-P methods, and exhibit (at least) comparable accuracy with respect to BNB, while significantly reducing the computational time.

VI. CONCLUSION

In this paper, three post-processing methods based on convex optimisation tools are introduced to improve accuracy of NILM algorithms. The proposed methodology involves, as an intermediate step, a heuristic approach to solve a (combinatorial) boolean quadratic problem through relaxing zero-one constraint sets to compact zero-one intervals. SDR is applied to solve boolean quadratic problems with zero-one constraint sets. The three proposed approaches provide different trade-offs between performance and computational efficiency. The performance is compared with several post-processing NILM methods [10], [21], [42]. The experiments show that the proposed methods have better or similar performance to the benchmarks, but at much lower complexity.

Regarding the overall combinatorial (NP hard) problem, we do not have an estimate of ‘quality’ of the sub-optimal solution, i.e., we do not have a guarantee on how far we are from the global solution of the combinatorial problem, e.g., in terms of approximation accuracy. Note that, approximation accuracy guarantees have been established for some related problems [43]. The proposed methods, as add-ons to the existing NILM, slightly increase the overall complexity of the disaggregation module. If the employed NILM has very poor result, it is unlikely that the proposed methods will lead to improvements. An interesting area of research is assessing the quality of the NILM output without relying on ground-truth to decide whether post-processing should be applied or not.

VII. ACKNOWLEDGEMENT

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734331.

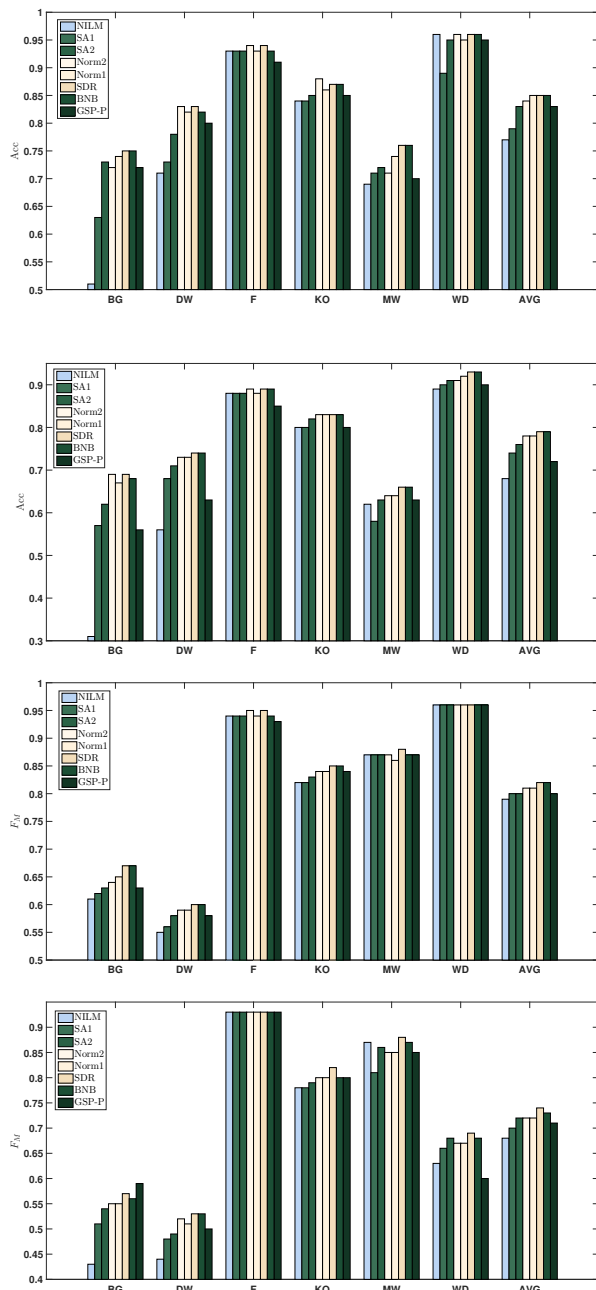


Fig. 3: Results for REDD House1. From top to bottom, the figures show: Acc after GSP-based NILM; Acc after DT-based NILM; F_M after GSP-based NILM; F_M after DT-based NILM.

Research of D. Jakovetic is also supported by the Serbian Ministry of Education, Science, and Technological Development, Grant no. 174030

REFERENCES

- [1] *My Energy Data.*, Smart Grids Task Force. Ad hoc group of the Expert Group 1 Standards and Interoperability., Nov 2016.
- [2] *Smart Metering Equipment Technical Specifications Version 2*, Department of Energy and Climate Change, UK, 2013.
- [3] Satya Jayadev Pappu, Nirav Bhatt, Ramkrishna Pasumarthy, and Aravind Rajeswaran, "Identifying topology of low voltage distribution networks based on smart meter data," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5113–5122, 2018.

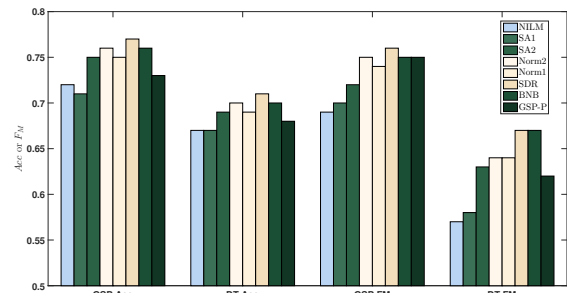


Fig. 4: REDD House 2 results: Average performance across all appliances.

- [4] Soham Chakraborty and Sarasij Das, "Application of smart meters in high impedance fault detection on distribution systems," *IEEE Transactions on Smart Grid*, 2018.
- [5] Madalina-Mihaela Buzau, Javier Tejedor-Aguilera, Pedro Cruz-Romero, and Antonio Gómez-Expósito, "Detection of non-technical losses using smart meter data and supervised learning," *IEEE Transactions on Smart Grid*, 2018.
- [6] Ramin Moghaddass and Jianhui Wang, "A hierarchical framework for smart grid anomaly detection using large-scale smart meter data," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5820–5830, 2018.
- [7] Barry Patrick Hayes and Milan Prodanovic, "State forecasting and operational planning for distribution network energy management systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1002–1011, 2016.
- [8] Yantai Huang, Lei Wang, Weian Guo, Qi Kang, and Qidi Wu, "Chance constrained optimization in a home energy management system," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 252–260, 2018.
- [9] S. Makonin, F. Popowich, I. V. Bajji, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 7, pp. 2575–2585, Nov 2016.
- [10] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Trans. Smart Grid*, vol. 9, pp. 1739–1747, May 2018.
- [11] Ding Li and Scott Dick, "Residential household non-intrusive load monitoring via graph-based multi-label semi-supervised learning," *IEEE Transactions on Smart Grid*, 2018.
- [12] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, 2018.
- [13] C.-L. Su, W.-H. Lee, and C.-K. Wen, "Electricity theft detection in low voltage networks with smart meters using state estimation," in *IEEE Int. Conf. Industrial Technology (ICIT)*, 2016, pp. 493–498.
- [14] T. Erseghe and S. Tomasin, "Power flow optimization for smart microgrids by sdp relaxation on linear networks," *IEEE Trans. Smart Grid*, vol. 4, pp. 751–762, 2013.
- [15] Carrie Armel, Abhay Gupta, Gireesh Shrimali, and Adrian Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213234, 01 2013.
- [16] B. Zhao, L. Stankovic, and V. Stankovic, "On a training-less solution for non-intrusive appliance load monitoring using graph signal processing," *IEEE Access*, vol. 4, pp. 1784–1799, 2016.
- [17] Haroon Rashid, Pushpendra Singh, Vladimir Stankovic, and Lina Stankovic, "Can non-intrusive load monitoring be used for identifying an appliance's anomalous behaviour?," *Elsevier Applied Energy*, pp. 796–805, 3 2019.
- [18] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Nature Scientific Data*, vol. 4, pp. 160122, 01 2017.
- [19] O. Parson, S. Ghosh, M. J. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types,," 2012.
- [20] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. the 2011 SIAM Int. Conf. Data Mining*, SIAM, 2011, pp. 747–758.
- [21] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," *J. Machine Learning Research - Workshop and Conf. Proc.*, vol. 22, pp. 1472–1482, 2012.
- [22] K. Shaloudegi, A. György, C. Szepesvari, and W. Xu, "Sdp relaxation with randomized rounding for energy disaggregation," in *Advances in*

- Neural Inform. Proc. Sys.* 29, pp. 4978–4986. Curran Associates, Inc., 2016.
- [23] M. Zhong, N. Goddard, and C. Sutton, “Latent bayesian melding for integrating individual and population models,” in *Advances in Neural Inform. Processing Sys.*, 2015, pp. 3618–3626.
- [24] A. Zoha, A. Gluhak, I. Muhammad, and S. Rajasegarar, “Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey,” *Sensors*, vol. 12, pp. 16838–16866, 2012.
- [25] H. Goncalves, A. Ocneanu, M. Berges, and RH Fan, “Unsupervised disaggregation of appliances using aggregated consumption data,” in *The 1st KDD Workshop Data Mining App. in Sust. (SustKDD)*, 2011.
- [26] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, “Enhancing electricity audits in residential buildings with nonintrusive load monitoring,” *J. Industrial Ecology*, vol. 14, no. 5, pp. 844–858, 2010.
- [27] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, “Learning systems for electric consumption of buildings,” in *Computing in Civil Engineering (2009)*, pp. 1–10. 2009.
- [28] K. D. Anderson, M. E. Berges, A. Ocneanu, D. Benitez, and J. M. F. Moura, “Event detection for non intrusive load monitoring,” in *IECON 38th Annual Conf. IEEE IES*, Oct 2012, pp. 3312–3317.
- [29] G. Lin, S. Lee, J. Y. Hsu, and W. Jih, “Applying power meters for appliance recognition on the electric panel,” in *2010 5th IEEE Conf. Industrial Electronics and App.*, June 2010, pp. 2254–2259.
- [30] D. Srinivasan, W. S. Ng, and A. C. Liew, “Neural-network-based signature recognition for harmonic source identification,” *IEEE Trans. Power Delivery*, vol. 21, pp. 398–405, Jan 2006.
- [31] M. Figueiredo, B. Ribeiro, and A. D. Almeida, “Electrical signal source separation via nonnegative tensor factorization using on site measurements in a smart home,” *IEEE Trans. Instrum. and Measurement*, vol. 63, pp. 364–373, Feb 2014.
- [32] H. Altrabalsi, V. Stankovic, J. Liao, and L. Stankovic, “Low-complexity energy disaggregation using appliance load modelling,” *AIMS Energy*, vol. 4, pp. 884–905, 2016.
- [33] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, “Non-intrusive appliance load monitoring using low-resolution smart meter data,” in *2014 IEEE Int. Conf. Smart Grid Comm.*, Nov 2014, pp. 535–540.
- [34] S. Giri and M. Berges, “An energy estimation framework for event-based methods in non-intrusive load monitoring,” *Energy Conversion and Management*, vol. 90, pp. 488 – 498, 2015.
- [35] J. Kelly and W. Knottenbelt, “Neural nilm: Deep neural networks applied to energy disaggregation,” in *Proc. the 2Nd ACM Int. Conf. Embedded Systems for Energy-Efficient Built Env.*, New York, NY, USA, 2015, pp. 55–64, ACM.
- [36] N. Henao, K. Agbossou, S. Kelouwani, Y. Dub, and M. Fournier, “Approach in nonintrusive type i load monitoring using subtractive clustering,” *IEEE Trans. Smart Grid*, vol. 8, pp. 812–821, March 2017.
- [37] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O’Hare, “Real-time recognition and profiling of appliances through a single electricity sensor,” in *2010 7th Annual IEEE Commun. Society Conf. Sensor, Mesh and Ad Hoc Commun. and Networks (SECON)*, June 2010, pp. 1–9.
- [38] D. Murray, L. Stankovic, V. Stankovic, S. Lulic, and S. Sladojevic, “Transferability of neural network approaches for low-rate energy disaggregation,” in *ICASSP-2019 IEEE Int. Conf. Acoustic, Speech, and Sig. Proc.*, May 2019.
- [39] D. Li and S. Dick, “Residential household non-intrusive load monitoring via graph-based multi-label semi-supervised learning,” *IEEE Trans. Smart Grid*, pp. 1–1, 2018.
- [40] W. Kong, Z. Y. Dong, J. Ma, D. J. Hill, J. Zhao, and F. Luo, “An extensible approach for non-intrusive load disaggregation with smart meter data,” *IEEE Trans. Smart Grid*, vol. 9, pp. 3362–3372, July 2018.
- [41] K. Kumar and M. G. Chandra, “An intuitive explanation of graph signal processing-based electrical load disaggregation,” in *2017 IEEE 13th Int. Coll. Sig. Proc. its App. (CSPA)*, March 2017, pp. 100–105.
- [42] B. Zhao, K. He, L. Stankovic, and V. Stankovic, “Improving event-based non-intrusive load monitoring using graph signal processing,” *IEEE Access*, pp. 1–1, 2018.
- [43] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Sig. Proc. Magazine*, vol. 27, pp. 20–34, May 2010.
- [44] D. E. Knuth, “Postscript about np-hard problems,” *SIGACT News*, vol. 6, no. 2, pp. 15–16, Apr. 1974.
- [45] J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, “Compressed sensing with prior information: Optimal strategies, geometry, and bounds,” *IEEE Trans. Inform. Theory*, vol. 63, July 2017.
- [46] G. W. Hart, “Nonintrusive appliance load monitoring,” *Proc. the IEEE*, vol. 80, pp. 1870–1891, Dec 1992.
- [47] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, “Load signature study part i: Basic concept, structure, and methodology,” *IEEE Trans. Power Delivery*, vol. 25, pp. 551–560, April 2010.
- [48] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito, “Nonintrusive appliance load monitoring based on integer programming,” in *SICE Annual Conf.*, 2008. IEEE, 2008, pp. 2742–2747.
- [49] M. Baranski and J. Voss, “Genetic algorithm for pattern detection in nialm systems,” in *2004 IEEE Int. Conf. Sys., Man and Cybernetics*, Oct 2004, vol. 4, pp. 3462–3468 vol.4.
- [50] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, “Sequence-to-point learning with neural networks for nonintrusive load monitoring,” *arXiv:1612.09106*, Dec. 2016.
- [51] L. Mauch and B. Yang, “A novel dnn-hmm-based approach for extracting single loads from aggregate power signals,” in *2016 IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP)*, March 2016.
- [52] M. Pchacker, D. Egarter, and W. Elmenreich, “Proficiency of power values for load disaggregation,” *IEEE Trans. Instrum. and Measurement*, vol. 65, pp. 46–55, Jan 2016.
- [53] P. Tseng, “An infeasible path-following method for monotone complementarity problems,” *SIAM J. Optimization*, vol. 7, pp. 386–402, 1997.
- [54] M. Grant, S. Boyd, and Y. Ye, “Cvx: Matlab software for disciplined convex programming,” 2008.
- [55] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” *Recent advances in learning and control*, pp. 95–110, 2008.
- [56] Y. H. Dai and R. Fletcher, “Projected barzilai-borwein methods for large-scale box-constrained quadratic programming,” *Numerische Mathematik*, vol. 100, pp. 21–47, 2005.
- [57] J. Wimp, “Pi and the agm: A study in analytic number theory and computational complexity (jonathan m. borwein and peter b. borwein),” *SIAM Review*, vol. 30, pp. 530–533, 1988.
- [58] J. Z. Kolter and M. J. Johnson, “Redd: A public data set for energy disaggregation research,” in *Proc. Workshop Data Mining App. Sust. (SIGKDD)*, vol. 25, pp. 59–62, Jan 2011.