

# INDiC: Improved Non-Intrusive load monitoring using load Division and Calibration

Nipun Batra<sup>1</sup>, Haimonti Dutta<sup>2\*</sup>, Amarjeet Singh<sup>1</sup>

<sup>1</sup>Indraprastha Institute of Information Technology, Delhi, India  
{nipunb,amarjeet}@iiitd.ac.in

<sup>2</sup>The Center for Computational Learning Systems (CCLS)  
Columbia University, New York, NY  
haimonti@ccls.columbia.edu

**Abstract**—Residential buildings contribute significantly to the overall energy consumption across most parts of the world. While smart monitoring and control of appliances can reduce the overall energy consumption, management and cost associated with such systems act as a big hindrance. Prior work has established that detailed feedback in the form of appliance level consumption to building occupants improves their awareness and paves the way for reduction in electricity consumption. Non-Intrusive Load Monitoring (NILM), i.e. the process of disaggregating the overall home electricity usage measured at the meter level into constituent appliances, provides a simple and cost effective methodology to provide such feedback to the occupants. In this paper we present Improved Non-Intrusive load monitoring using load Division and Calibration (INDiC) that simplifies NILM by dividing the appliances across multiple instrumented points (meters/phases) and calibrating the measured power. Proposed approach is used together with the Combinatorial Optimization framework and evaluated on the popular REDD dataset. Empirical results demonstrate significant improvement in disaggregation accuracy, achieved by using INDiC.

## I. INTRODUCTION

Buildings account for significant proportion of overall energy use in both the developing (e.g. 47% of total energy in India [1]) and the developed (e.g. 41% and 45% in US and UK respectively) countries. Modest improvements in building energy use can result in significant aggregate impact at the national scale. While several automation and management systems have been proposed for improving the operational efficiency of building systems, such systems typically lack the ability to provide detailed consumption information (e.g. appliance level consumption). Prior work [2] has shown that better feedback systems, enabling appliance level consumption, that provide insights about occupant's energy usage information further encourages energy saving behavior resulting in 5-15% savings in electricity usage.

Measuring each appliance's consumption separately, for providing such a feedback, is both prohibitively expensive and difficult to manage. Alternatively, prior research has proposed Non Intrusive Load Monitoring (NILM) that involves disaggregating the aggregated electricity consumption obtained at the meter level into individual appliance consumption. Several modeling and inference approaches have been proposed (e.g. Factorial Hidden Markov Model [3], Combinatorial Optimization [4]) in the past to address NILM with varied level of accuracy. NILM work typically assumes that all the loads are assigned to the same meter. However, many practical scenarios (e.g. use of split-phase supply in many homes in USA and 3-phase supply for many homes in India) involve load division across different phases coming at the home level. Automated

assignment of different loads in a home to each phase followed by NILM application on each phase separately can reduce the overall modeling and inference complexity. We follow the convention introduced in REDD dataset [5] where these multiple instrumented points (meters/phases) are referred as **main(s)**.

Further, the measurements, both at the meter level and at the appliance level, are often taken with different equipment (Current Transformers, in-line measurements, ICs<sup>1</sup>) each with their own accuracy levels. Calibrating these diverse measurements will be beneficial for NILM modeling and inference. Grid conditions such as voltage fluctuations further motivate calibration. Motivated by these practical scenarios, we propose INDiC- Improved NILM using load **D**ivision and **C**alibration. Specific research contributions of our work are:

- INDiC involving two simple pre-processing steps - Load division (i.e. automated assignment of different loads in a home to each of the separate mains) and calibration (accounting for varied measurement accuracy across different equipment), that can be applied in a generic manner across several proposed NILM approaches to further improve their accuracy.
- Extensive empirical validation, using publicly available REDD [5] dataset, establishing the effectiveness of INDiC as a preprocessing step, specifically for Combinatorial Optimization based NILM.
- Release of open source implementation of the proposed work<sup>2</sup> for comparative analysis with other NILM approaches as an IPython notebook<sup>3</sup>.

We believe that this is the first extensive release of a generic NILM code base that can be used across many of the publicly available datasets and across several existing NILM modeling and inference approaches. Henceforth appliance(s) and load(s) are used interchangeably across the paper.

## II. RELATED WORK

Non-intrusive load monitoring was first studied by Hart [4] in the early 1990s by examining signatures in aggregated load to indicate activities of appliances. With proliferation of smart meters, the problem has gained traction in the recent times [6], [7], [8]. NILM systems can be broadly divided into two categories based on whether supervised or unsupervised disaggregation methods are used. Supervised

\*The author is also a visiting assistant professor at IIIT-Delhi.

<sup>1</sup>Example IC for power measurement is Maxim 78M6612

<sup>2</sup>Add link to website

<sup>3</sup><http://www.ipython.org>

learning techniques include optimization-based methods such as integer programming [9] and genetic algorithms [10]. These approaches are compute intensive and appliances with similar or overlapping load signatures are typically difficult to discern. Other machine learning techniques (such as Artificial Neural Network (ANNs) [11] and Hidden Markov Models (HMM) [12]), have also been shown to work well for the task.

Variants of HMMs such as factorial hidden Markov models (FHMMs) [3], additive FHMMs [13], conditional factorial hidden semi-Markov models [14] and difference HMMs [15] have been studied in the context of NILM. In factorial HMMs several models evolve independently and in parallel with the observed output being a joint function of all the hidden states. Additive HMMs allow emission of a single real-valued (unobserved) output from each HMM and the output is the sum of these HMMs. In difference HMMs [15] each load is modeled using a graphical model followed by its disaggregation from the aggregate power – this process is repeated iteratively until all appliances (for which general models are available) are disaggregated. It is therefore possible to infer the probability that a change in aggregated power was generated by two consecutive states of an appliance.

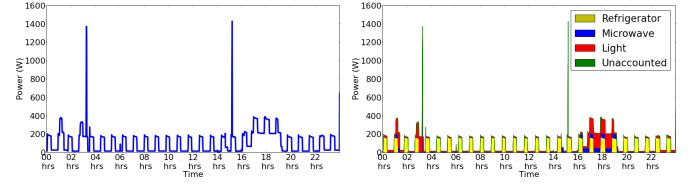
The applicability of these sophisticated techniques are generally hindered by the difficulty of inference from models with a large number of HMMs. Rahayu et. al. [16] propose a discriminative model for energy disaggregation that predicts the most likely appliance state configuration from aggregated load using nonparametric classification algorithms. They posit that “subset sum” type techniques are not very effective since a large portion of the home energy is not monitored directly.

In this paper, we put forward a simple idea – the aggregated load can be further split up by information on mains and knowing which appliance is assigned to which main. This would make the task of disaggregation inherently easier by solving a simpler optimization problem and most of the above sophisticated machine learning based modeling approaches can still be applied to the task<sup>4</sup>.

Several datasets have been released publicly for benchmarking energy disaggregation algorithms including REDD [5], Blued [17], Pecan [18] and Smart\* [19]. Our empirical analysis is on the REDD data primarily because this is the most popular data set for evaluating state-of-the-art NILM approaches.

### III. NILM

A typical NILM setup involves measuring the power mains with a smart meter and individual appliances with appliance meters for ground truth. The NILM problem in the supervised setting is formulated as predicting the power sequence for  $n^{th}$  appliance,  $y^n$ , given the measured power sequence for each appliance  $\theta^n$  (measured using appliance meters) and the total aggregate power sequence  $x$  (measured using the smart meter). Table I summarizes the terminologies and functions used in INDiC, many of which are adopted from prior literature ([5], [15], [4]). Figure 1 shows the process of disaggregation applied to a house mains, whereby the consumption patterns of 3 appliances: refrigerator, lighting and microwave can be seen. It must be highlighted that it is improbable to instrument all



(a) Aggregate home power measured by a smart meter (b) Meter power disaggregation for 3 appliances: refrigerator, lighting and microwave, together with some unaccounted power

Fig. 1: Disaggregating a home's electrical mains

the appliances in a home, thus, there will be some unaccounted power, which can also be seen in the figure. We now explain the Combinatorial Optimization framework initially proposed by Hart [4] for solving NILM.

TABLE I: Terminologies and Functions

Symbol	Meaning
$t \in 1, \dots, T$	Time slice
$n \in 1, \dots, N$	Appliance number
$\theta^n = \{\theta_1^n, \dots, \theta_T^n\}$	Measured power sequence for $n^{th}$ appliance
$\theta^{M1} = \{\theta_1^{M1}, \dots, \theta_T^{M1}\}$	Measured power sequence for main 1
$\theta^{M2} = \{\theta_1^{M2}, \dots, \theta_T^{M2}\}$	Measured power sequence for main 2
$x = \{x_1, \dots, x_T\} = \theta^{M1} + \theta^{M2}$	Measured aggregate power sequence
$e = \{e_1, \dots, e_T\}$	Aggregate noise power sequence
$p$	Number of electrical mains in a home
$N_i$ where $i \in \{1, \dots, p\}$	Number of loads in $i^{th}$ main
$y^n = \{y_1^n, \dots, y_T^n\}$	Predicted power sequence for $n^{th}$ appliance
$k \in 1, \dots, K$	Appliance power state eg. Stove has 2 states: On and Off
$z^n = \{z_1^n, \dots, z_T^n\}$	Appliance state sequence for $n^{th}$ appliance, $z_i^n \in [1, \dots, K]$
$z_{t,k}^n \in 0, 1$	Whether $n^{th}$ appliance is in $k^{th}$ state at time $t$
$\mu^n = \{\mu_1^n, \dots, \mu_K^n\}$	Power draw by $n^{th}$ appliance, where $\mu_k^n$ is the power draw by $n^{th}$ appliance in $k^{th}$ state
$\theta_{k^1, k^2}^n$	Measured power sequence when $n^{th}$ appliance transitions from $k^1$ to $k^2$ state
$Mapping[n] \in M_i$ where $i \in 1, 2$	Mapping of $n^{th}$ appliance to $i^{th}$ main
$s_t$	Value of a timeseries $s$ at time $t$
$Event(s, threshold)$	An event in timeseries $s$ occurs when $ s_t - s_{t-1}  > threshold$ An event has an associated time $t$ and magnitude $ s_t - s_{t-1} $
$Downsample(s, filter, resolution)$	Function to downsample a timeseries $s$ to a $resolution$ according to specified $filter$
$Timeseries\_sync([s^1, \dots, s^n], method)$	Function to ensure that $n$ timeseries start and end at same times and handling missing data using specified $method$
$Sort([s^1, \dots, s^n], parameter, order)$	Function to sort $n$ timeseries according to $parameter$ in specified $order$
$Event\_Detection(s, threshold)$	Function returning magnitude and times of $Events(s, threshold)$ in timeseries $s$
$Cluster(s, K, clustering\_algorithm)$	Function to divide timeseries $s$ into $K$ clusters based on $clustering\_algorithm$

**Combinatorial Optimization (CO):** At a given time an appliance can only be in a single state, expressed mathematically

as:  $\sum_{k=1}^{K} z_{t,k}^n = 1$ . The power consumption by  $n^{th}$  appliance

in  $k^{th}$  state at time  $t$  is given by:  $\hat{\theta}_{t,k}^n = \sum_{k=1}^K z_{t,k}^n \mu_k^n$ . The

overall power consumption of all appliances at a given time  $t$  is given by:  $\hat{x}_t = \sum_{n=1}^N \sum_{k=1}^K z_{t,k}^n \mu_k^n$ . The error in power signal (unaccounted power) after the load assignment is given by

<sup>4</sup>The use of combinatorial optimization is only for the purpose of illustration and there is clearly no requirement to adhere to this modeling technique alone.

$e_t = |x_t - \sum_{n=1}^N \sum_{k=1}^K z_{t,k}^n \mu_k^n|$ . Combinatorial optimization seeks to find the optimal combination of appliances in different states which will minimize this error term, using the following state assignment scheme:  $z_t = \operatorname{argmin}_{z_t} |x_t - \sum_{n=1}^N \sum_{k=1}^K z_{t,k}^n \mu_k^n|$

The corresponding predicted power draw by  $n^{th}$  appliance is given by  $y^n = \{\mu_{z_1^n}^n, \dots, \mu_{z_T^n}^n\}$ . This optimization problem resembles subset sum problem [20] and is NP-complete. The state space size of this optimization function is  $K^N$ , which means it is exponential in the number of appliances.

**Load division:** Across many countries, electrical distributions are planned such that different loads are connected to different mains (e.g. split-phase in the USA and 3-phase supply in India). Since NILM deployment typically involves monitoring different electrical mains separately, we leverage the load division to perform efficient disaggregation. Considering  $p$  mains in a home, we first perform automated load assignment (for a total of  $N$  loads in the home) to individual main (resulting in  $N_i$  loads being assigned to the  $i^{th}$  main). Such a load division across different mains results in exponential reduction in state space for disaggregation each main separately (the state space size for  $i^{th}$  main is given by  $K^{N_i}$ ). CO formulation for the  $i^{th}$  main after load division is given by the following optimization function:  $z_t = \operatorname{argmin}_{z_t} |\theta^{M_i} - \sum_{n=1}^{N_i} \sum_{k=1}^K z_{t,k}^n \mu_k^n| \quad \forall \quad i \in \{1, \dots, p\}$

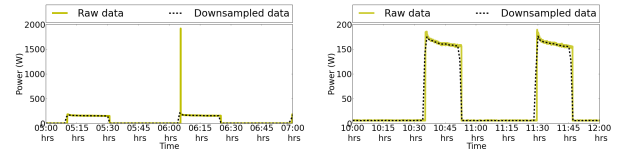
The corresponding predicted appliance power sequence for  $n^{th}$  appliance is given by:  $y^n = \{\mu_{z_1^n}^n, \dots, \mu_{z_T^n}^n\}$ . CO with load division is subject to the following constraints:

- 1) The sum of number of loads assigned to different mains must be equal to the total number of loads. This is given by:  $\sum_{i=1}^p N_i = N$
- 2) At any given time, an appliance can only be in a single state which is given by:  $\sum_{k=1}^K z_{t,k}^n = 1$
- 3) An appliance can belong to one and only one main.
- 4) The sum of power consumption of all appliances assigned to  $i^{th}$  main is always lesser than or equal to the total power of the main (i.e.  $e_t$  term for  $i^{th}$  main will be non negative).

#### IV. INDiC NILM

We now describe our proposed algorithm - Improved NILM using load Division and Calibration (INDiC). INDiC provides preprocessing procedures that can simplify NILM computation and improve the overall disaggregation accuracy. These procedures can broadly be classified as data cleaning (time series synchronization, downsampling and calibration) and problem division into subproblems (assigning loads to mains). INDiC can be used with any NILM approach described in Section II. Here, we present INDiC-CO (INDiC using Combinatorial optimization for NILM). Various steps of INDiC-CO NILM, shown in Algorithm 1, are described next.

**Time series synchronization:** Mains power and appliance power are typically measured using different hardware. As an example, in REDD [5] TED meters<sup>5</sup> are used to measure mains



(a) Filtering the high starting current from compressor of refrigerator (b) Filtering the voltage fluctuations and oscillations  
Fig. 2: Effect of downsampling appliance data

and Power House Dynamics<sup>6</sup> are used to measure appliance circuits. It is highly likely that some hardware malfunctions during the data collection process. In this step we ensure that the mains power and appliance power time series start and end at the same time. Further missing data is handled using the techniques such as forward filling (padding)<sup>7</sup>.

**Downsampling:** While performing CO, it is desired that transients and fluctuations in the power signal are filtered [4]. The transients occur due to the high starting current of the appliance, whereas the fluctuations are a consequence of minor voltage fluctuations and oscillatory nature of appliances. Figure 2a and Figure 2b show how starting current and voltage fluctuations can be filtered by down sampling. Filters such as mean/median can be used to down sample a time series to a time window, whereby the value assigned to the filtered series for a time window is the mean/median of original series occurring during that time window.

**Assigning Loads to mains<sup>8</sup>:** This step aims to identify the mapping ( $Mapping[n]$ ) between appliances and mains. Since an appliance can belong to a single main,  $Mapping[n]$  is a one-to-one function. Since the patterns corresponding to appliances having higher peak power are generally easier to extract from the main signal, we first sort the appliance in decreasing order of peak power. Starting from the appliance having the highest peak load, we take one appliance at a time and compare its power at all times with the power of each of the mains. If at any time the power of the appliance is greater than the power of any of the mains, we can assign the appliance to the other main. If we are not able to assign an appliance to mains using this approach, we find the times when *Events* occur in the appliance power series. This should be a subset of times when *Events* occur in one of the main to which this appliance is assigned. The threshold used to find these events should be suitably chosen to ensure that minor voltage fluctuations are not counted as events. Once an appliance has been assigned to a mains, using either of these two filter, its power sequence is subtracted from the corresponding mains to simplify mains assignment for the remaining appliances. Figure 3a shows refrigerator assignment to main 2 since during this time interval refrigerator power is more than that of main 1. One can also verify that the events in main 2 and refrigerator power series occur at the same time.

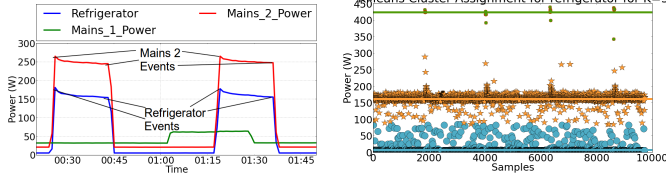
**Clustering:** Prior knowledge and appliance circuitry [21] are used to identify the number of states associated with an appliance. For instance, a refrigerator is a compressor based appliance and exhibits three states in increasing order of power demand (compressor Off, compressor On, defrost mode). Cor-

<sup>6</sup><http://goo.gl/9VQba>

<sup>7</sup><http://goo.gl/FfR6o>

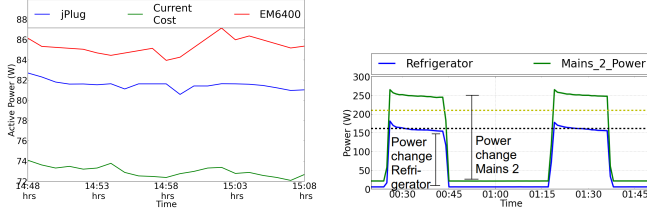
<sup>8</sup>While our approach is fine tuned for two mains, it can be easily extended to support further load division

<sup>5</sup><http://goo.gl/CEu2y>



(a) Assignment of refrigerator to main 2 (Refrigerator power > main 1 power) (b) Clustering refrigerator power consumption - compressor off (blue), compressor on (yellow) and defrosting (green)

Fig. 3: Load Assignment and Clustering



(a) Different power measurements for the same load (b) Calibrating refrigerator power with mains power

Fig. 4: Need for and utility of Calibration

responding cluster assignment is shown in Figure 3b.

**Appliance Power Calibration:** Power measured by appliance level meters may need calibration due to several reasons, including:

- Difference in the measurement devices can result in different measurements for the same appliance [22]. To illustrate this difference, we measured our refrigerator<sup>9</sup> power with 3 different devices: i) jPlug<sup>10</sup>; ii) Current Cost CT<sup>11</sup>; and iii) EM6400 smart meter<sup>12</sup>. Figure 4a illustrates the power measurement by each of these devices. There is a difference in approx. 10 W in measurements reported by jPlug and Current Cost. jPlug gives comparable results to EM6400.
- Voltage fluctuations from the grid resulting in power measurement fluctuations [4]
- Missing meta data - labeling the appliance level power consumption as real or apparent power

In comparison to appliance data, mains data is usually measured with better precision devices. Thus, we keep mains data as a reference and calibrate appliance data against it. In the clustering step, value of appliance power at each time is associated with corresponding cluster state ( $k \in \{1, \dots, K\}$ ). Since in Off state ( $k=1$ ) appliance power consumption is almost zero, it does not require any calibration. We find out *Event* times when appliance transitions from a lower state( $k$ ) to a higher state( $k+1$ ). During these times, we find the ratio of the magnitude of power change occurring in the assigned mains and the appliance. This ratio serves as a corrective multiplicative factor for a particular state of an appliance. Cluster centroids obtained in the previous step are multiplied by this factor to obtain calibrated cluster centroids. This process is shown in Figure 4b where it is observed that before and after calibration (with mains 2) refrigerator power in state 2 was 162 W and 214 W respectively, with the calibration factor of 1.34.

<sup>9</sup><http://goo.gl/lzdDk>

<sup>10</sup>A variant of nPlug [23]

<sup>11</sup><http://www.currentcost.net/transmitterspec.html>

<sup>12</sup><http://goo.gl/Oi98q>

**Combinatorial optimization:** Combinatorial optimization is now performed separately for both mains as per the description in Section III.

#### Algorithm 1: INDiC-CO

---

**Input:**  $x, \theta^n, \theta^{M_1}, \theta^{M_2}$   
**Output:**  $y^n, \mu_k^n$

**Time series synchronization**

- 1  $\theta^1, \dots, \theta^n, \theta^{M_1}, \theta^{M_2} \leftarrow$   
*Timeseries\_sync*( $[\theta^1, \dots, \theta^n, \theta^{M_1}, \theta^{M_2}]$ , forward fill)

**Downsampling**

- 2 **for**  $n \in 1, N$  **do**  
     $\theta^n \leftarrow \text{Downsample}(\theta^n, \text{filter}, \text{resolution})$
- 3  $\theta^{M_1} \leftarrow \text{Downsample}(\theta^{M_1}, \text{filter}, \text{resolution})$
- 4  $\theta^{M_2} \leftarrow \text{Downsample}(\theta^{M_2}, \text{filter}, \text{resolution})$
- 5  $j \leftarrow \text{Sort}([\theta^1, \dots, \theta^n], \text{peak power})$

**Appliance to mains mapping**

- 6 **for** Appliance  $n \in j$  **do**  
    7 **if**  $\theta_t^n > \theta_t^{M_1}$  for any  $t \in 1, T$  **then**  
         $\text{Mapping}[n] = M_2$   
    8 **else if**  $\theta_t^n > \theta_t^{M_2}$  for any  $t \in 1, T$  **then**  
         $\text{Mapping}[n] = M_1$   
    9 **else**  
    10 **if** *Event\_Detection*( $\theta^n, \text{threshold}$ ).*Times*  $\subset$   
        *Event\_Detection*( $\theta^{M_1}, \text{threshold}$ ).*Times* **then**  
         $\text{Mapping}[n] = M_1$   
    11 **else**  
         $\text{Mapping}[n] = M_2$   
    12  $\theta^{\text{Mapping}[n]} \leftarrow \theta^{\text{Mapping}[n]} - \theta^n$

**Divide data into train and test set**

**Clustering on train set**

- 13 **for**  $n \in 1, N$  **do**  
    14  $\mu_k^n \leftarrow$   
        *Cluster*( $\theta^n, K, \text{clustering\_algorithm}$ ) for  $k \in 1, K$

**Calibration on train set**

- 15 **for**  $n \in 1, N$  **do**  
    16 **for**  $k \in 2, K$  **do**  
    17  $\mu_k^n \leftarrow$   
         $\frac{\mu_k^n * \text{Event\_Detection}(\theta_{k-1,k}^{\text{Mapping}[n]}, \text{threshold}).\text{Magnitude}}{\text{Event\_Detection}(\theta_{k-1,k}^n, \text{threshold}).\text{Magnitude}}$

**Combinatorial Optimization(CO) on test set**

- 18 Solve CO as described in Section III
- 19 **return**  $y^n, \mu_k^n$

---

## V. EVALUATION

We use Reference Energy Disaggregation Data Set (REDD) [5] for validating our algorithms using the metrics as discussed in Section V-B.



TABLE II: Mains assignment and appliance states power before and after calibration. Kitchen and stove are 2 state appliances

Appliance(n)	Main	States Power (W)					
		Pre calibration			Post calibration		
		$\mu_1^n$	$\mu_2^n$	$\mu_3^n$	$\mu_1^n$	$\mu_2^n$	$\mu_3^n$
Refrigerator	2	7	162	423	7	214	423
Microwave	2	9	822	1740	9	822	1740
Lighting	2	9	96	156	9	113	156
Dishwasher	1	0	260	1195	0	260	1195
Stove	1	0	373	-	0	373	-
Kitchen	1	5	727	-	5	727	-
Kitchen 2	1	1	204	1036	1	204	1036

### A. Dataset

REDD contains power data for mains (split-phases) as well as appliances from 6 homes in Boston area collected in the summer of 2011. The data is made available as raw, high frequency (sampled at 15 KHz) and low frequency (mains at 1 Hz, appliances at 0.3 Hz). Considering the practical implications of residential smart meter installation, we believe that low frequency data represents the most realistic scenario and thus we use this data for analysis.

### B. Evaluation Metric

We use the following metrics that have also been used in prior work [15], [5]:

**Mean Normalized Error (MNE-orig):** Normalized error in the energy assigned to an appliance  $n$  over time period  $T$ , given by:

$$MNE - orig(n) = \frac{|\sum_{t=1}^T \theta_t^n - \sum_{t=1}^T y_t^n|}{\sum_{t=1}^T \theta_t^n} \quad (1)$$

Note that this metric will give 0% MNE-orig for a two state appliance (with 0 watts and 10 watts as consumption in the two states) which is predicted completely inaccurately i.e.  $y^n = [0, 10, 0, 10]$  and  $\theta^n = [10, 0, 10, 0]$ . Since this can be misleading, we propose a modified Mean Normalized Error metric further referred as MNE as:

$$MNE(n) = \frac{\sum_{t=1}^T |\theta_t^n - y_t^n|}{\sum_{t=1}^T \theta_t^n} \quad (2)$$

Since  $|\sum a - \sum b| \leq \sum |a - b|$  where  $a$  and  $b$  are vectors containing positive floating point numbers, our results may appear to be worse than if MNE-orig was used.

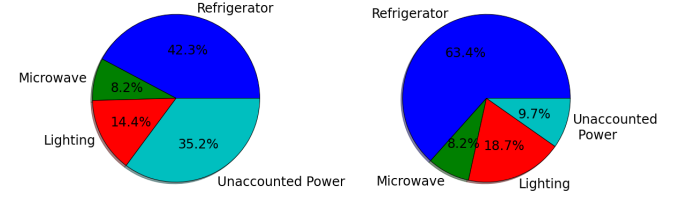
**RMS Error (RE Watts):** RMS error in power assignment to an appliance  $n$  per time slice  $t$  is given as:

$$RE(n) = \sqrt{\frac{1}{T} \sum_{t=1}^T (\theta_t^n - y_t^n)^2} \quad (3)$$

Since both MNE and RE represent error, the lower their value, better is the disaggregation accuracy.

### C. Empirical Analysis

We performed empirical analysis on REDD dataset Home 2, which consists of 11 channels (including 2 mains and 9 appliances)<sup>13</sup>. We believe that the same analysis can be easily repeated across multiple homes. Timeseries synchronization



(a) Before Calibration - more than one-thirds (b) After Calibration - Unaccounted power reduces to less than 10% of total power

Fig. 5: Main 2 break down by load

was applied since the appliance level data collection begun about 6 hours after mains data collection. Moreover, there were small intervals of missing data, which we filled using forward filling. Two appliances - washer dryer and disposal were ignored from further analysis since washer dryer had a peak power consumption of 8 W (implying that it was Off throughout) and the contribution of disposal to overall power was less than 0.1 %. We downsampled this time synchronized data to one minute resolution using mean filter. 1 minute is a sufficient resolution to get rid of startup transients and voltage fluctuations.

Event detection, as explained in section IV, is an important part of INDiC. A threshold of 30 W was chosen for event detection to ensure that there are no false events (occurring due to minor power fluctuations). As per the Appliance to mains step in INDiC-CO algorithm described earlier, we assigned loads to the two different mains. Table II shows the resultant load assignment to different mains. Table II further shows the learned power states of these appliance via KMeans++<sup>14</sup> [24] clustering. Refrigerator and lighting showed significant difference in power states post calibration. Based on the prior experience and appliance circuitry [21], we believe that since only these two appliances needed calibration, it may be a case that the appliance level monitor measured real power instead of apparent power and this metadata was missing from the released dataset. These two loads constitute a major portion of main 2 power. Figure 5 shows the reduction in unassigned power due to calibrating these two appliances.

To show the significance of load division and calibration as a preprocessing step to NILM algorithm, we considered 4 possible cases: i) no calibration, no load division; ii) no calibration, load division; iii) calibration, no load division; iv) calibration, load division (INDiC). Corresponding results with CO are presented in Table III. All appliances (barring MNE for microwave) show reduction in MNE and RE after applying INDiC-CO. However, there is significant improvement in correctly predicting refrigerator and lighting, which contribute the most to the overall electricity consumption. Figure 6 shows the confusion matrix for refrigerator prediction using CO (without and with INDiC). It can be seen that after applying INDiC out of the 4810 instances of refrigerator in state 2, 4434 are correctly identified. Before applying INDiC only 2860 instances were correctly identified.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we present INDiC, which consists of novel preprocessing steps - Load Division and Calibration, to reduce the

<sup>13</sup>Highest accuracy has been reported for this Home in previous work [5].

<sup>14</sup>We also used DBScan, SOM, EM, Mini Batch KMeans and Hierarchical clustering algorithms and found KMeans++ to be the most scalable.

St- ate	# of usage events	1	2	3
1	5070	4740	288	41
2	4810	1775	2860	176
3	200	112	63	25

(a) After applying CO based NILM  
- State 2 is predicted often as State 1

St- ate	# of usage events	1	2	3
1	5070	4541	430	98
2	4810	221	4434	156
3	200	5	44	151

(b) After applying INDiC-CO based  
NILM - Significant improvement in  
State 2 and 3 prediction

Fig. 6: Confusion Matrices for refrigerator disaggregation. [m,n] in the matrix represents appliance's  $m^{th}$  state to be predicted as  $n^{th}$  state. Grey cells along the diagonal show true positive.

TABLE III: MNE and RE for CO based NILM with and without INDiC. Results for INDiC-CO are highlighted in grey

Appliance	Without calibration				With calibration			
	Without load division		With load division		Without load division		With load division	
	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %
Refrigerator	91	52	74	31	111	95	67	25
Microwave	96	96	96	113	98	95	96	113
Lighting	64	176	63	195	53	89	43	63
Dishwasher	131	662	52	73	156	1537	52	73
Stove	85	1428	35	271	74	1091	35	271
Kitchen	70	198	58	165	77	219	58	165
Kitchen 2	93	246	83	100	92	218	83	100

complexity of NILM modeling and improve the disaggregation accuracy. While we take the case of Load Division across two mains (specific to the REDD dataset used for evaluation), this can be easily extended for disaggregation across larger number of circuits. While our calibration only involved calibrating the appliance power with corresponding main, this can be further extended for calibration when the grid voltage fluctuates. This is particularly useful in the context of developing countries, e.g. India where we have personally observed high voltage fluctuations from 180 Volts to 250 Volts. Application of INDiC, together with Combinatorial Optimization, on the data from a real home from REDD dataset showed significant improvement in disaggregation accuracy as compared to when no pre-processing step is used. We further release our code as open source implementation and believe that this is the first extensive release of a generic NILM code base that can be used across many of the publicly available datasets and across several existing NILM modeling and inference approaches.

In the future we intend to apply INDiC as a preprocessing step to other classes of NILM approaches to establish its wide applicability. We have an ongoing deployment across two homes in India where we are collecting both the appliance level and meter level data. We intend to apply INDiC on this dataset, whereby the loads are significantly different from the ones used in the developed countries to understand the wide applicability of our proposed approach for diverse datasets.

## REFERENCES

- [1] M. Evans, B. Shui, and S. Somasundaram, "Country report on building energy codes in india," *PNNL*, vol. 177925, 2009.
- [2] S. Darby, "The effectiveness of feedback on energy consumption," *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, 2006.
- [3] Z. Ghahramani, M. I. Jordan, and P. Smyth, "Factorial hidden markov models," in *Machine Learning*. MIT Press, 1997.
- [4] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

- [5] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *proceedings of the SustKDD workshop on Data Mining Applications in Sustainability*, 2011, pp. 1–6.
- [6] K. Carrie Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, 2012.
- [7] 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.
- [8] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *Consumer Electronics, IEEE Transactions on*, vol. 57, no. 1, pp. 76–84, 2011.
- [9] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito, "Nonintrusive appliance load monitoring based on integer programming," in *SICE Annual Conference, 2008*, 2008, pp. 2742–2747.
- [10] M. Baranski and J. Voss, "Genetic algorithm for pattern detection in nialm systems," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 4, 2004, pp. 3462–3468 vol.4.
- [11] A. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor," in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, 2010, pp. 1–9.
- [12] T. Zia, D. Bruckner, and A. Zaidi, "A hidden markov model based procedure for identifying household electric loads," in *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, 2011, pp. 3218–3223.
- [13] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," *Journal of Machine Learning Research - Proceedings Track*, vol. 22, pp. 1472–1482, 2012.
- [14] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *SDM*, 2011, pp. 747–758.
- [15] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *26th AAAI Conference on Artificial Intelligence*, 2012.
- [16] D. Rahayu, B. Narayanaswamy, S. Krishnaswamy, C. Labbe, and D. Seetharam, "Learning to be energy-wise: Discriminative methods for load disaggregation," in *Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on*, 2012, pp. 1–4.
- [17] A. Filip, "Blued: A fully labeled public dataset for event-based non-intrusive load monitoring research."
- [18] C. Holcomb, "Pecan street inc.: A test-bed for nilm," in *International Workshop on Non-Intrusive Load Monitoring*, 2007, pp. 271–288.
- [19] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart\*: An open data set and tools for enabling research in sustainable homes," in *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011.
- [20] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.
- [21] K. Ting, M. Lucente, G. S. Fung, W. Lee, and S. Hui, "A taxonomy of load signatures for single-phase electric appliances," in *IEEE PESC (Power Electronics Specialist Conference)*, 2005, pp. 12–18.
- [22] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Training load monitoring algorithms on highly sub-metered home electricity consumption data," *Tsinghua Science & Technology*, vol. 13, pp. 406–411, 2008.
- [23] T. Ganu, D. P. Seetharam, V. Arya, R. Kunnath, J. Hazra, S. A. Husain, L. C. De Silva, and S. Kalyanaraman, "nplug: a smart plug for alleviating peak loads," in *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*. ACM, 2012, p. 30.
- [24] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.