A Tree-Structured Neural Network Model for Household Energy Breakdown

Anonymous Author(s)

ABSTRACT

Residential buildings constitute roughly one-third of the total energy usage across the globe. Numerous studies have shown that providing an *energy breakdown, i.e., per-appliance energy consumption*, increases residents' awareness of energy use and can help save up to 15% energy. A significant amount of prior work has looked into source-separation techniques collectively called non-intrusive load monitoring (NILM), and most prior NILM research has leveraged high-frequency household aggregate data (1 minute or less sampling interval) for energy breakdown. However, in practice most smart meters only sample hourly or once every 15 minutes, and existing NILM techniques show poor performance at such a low sampling rate.

In this paper, we propose a tree-structured convolutional neural network (CNN) for energy breakdown on low frequency data. There are three key insights behind the design of our model: i) households consume energy with regular temporal patterns (e.g. time of day, day of week, etc.), which can be well captured by filters learned in CNNs; ii) tree structure isolates the pattern learning of each appliance that helps avoid the magnitude variance problem, while preserves the relationship among appliances; iii) tree structure enables the separation of known appliance from unknown ones, which de-noises the input time series for better appliance-level reconstruction. Our TreeCNN model outperforms seven existing baselines on a publicly available dataset with lower estimation error and higher accuracy on detecting the active states of appliances.

KEYWORDS

Non-intrusive load monitoring, time series analysis, convolutional neural networks

ACM Reference format:
Anonymous Author(s). 2019. A Tree-Structured Neural Network Model for Household Energy Breakdown. In *Proceedings of ACM WWW conference, San Francisco, USA, May 2019 (WWW'19)*, 11 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

Residential buildings constitute roughly one-third of the total energy usage across the global [23]. Studies have shown that providing an *energy breakdown: per-appliance energy consumption statistics*, can motivate behavioral changes, potentially reducing energy consumption by 15% [2].

WWW'19, May 2019, San Francisco, USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

Various approaches have been proposed for providing an energy breakdown in the past. A straightforward solution involves installing a sensor on each appliance [9, 13]. But this approach scales poorly owing to the high number of sensors required per home, and the associated high cost of hardware and labor. In contrast, a set of techniques, collectively named as non-intrusive load monitoring (NILM) [11], require only a single sensor per home. NILM techniques are source separation techniques, which breakdown the household aggregate power time series (typically measured using a smart meter) into constituent appliances. NILM algorithms are designed for high-frequency data (sampling frequencies > 1/60 Hz), and do not apply when dealing with low sampling rates, like hourly samples, since at lower frequencies the time series do not provide such algorithms signals for accurate energy breakdown. However, high-frequency sensors are of high prices; and prior literature [25] and statistics about smart meter specifications [24] across the world suggest that the largest proportion of smart meters sample only at an hourly rate. This urges the need to develop new algorithms suited for time series data with lower sampling rates.

On the other end of the spectrum, there are approaches providing energy breakdown at a monthly level, e.g., using monthly bills as aggregate monthly energy consumption [4, 6, 7]. The key idea explored in such approaches is that common design patterns create a shared and repeating structure in residential buildings giving rise to a sparse set of features contributing to energy variations across homes. Matrix factorization [7] and kernel density estimation [6] techniques are introduced to exploit the sparsity. Nevertheless, such techniques cannot be directly applied to higher sampling rates, which rapidly increase the dimension of observations and model complexity. Moreover, such solutions only provide residents with limited resolution of energy breakdown, i.e., at a monthly level, which is less informative in helping them save energy.

In this paper, we study household energy breakdown at an hourly rate. Our extensive data analysis on a large public U.S. residential household energy consumption data repository suggests that sparsity and temporal regularity also exist in hourly appliance energy usage patterns, such as the time of a day, day of a week, which motivates us to view such time series data as a high dimensional compound, rather than just a one-dimensional sequence. For example, time of a day might differentiate the use pattern of microwave (typically used around meal times) from other appliances in the aggregate power readings, while the day of a week might indicate usage pattern of dryer and washing machine. Each of such temporal patterns creates a unique dimension to recognize a particular type of appliance's energy usage in the aggregate energy readings.

But it is clearly impossible to manually exhaust such temporal patterns for each type of appliance beforehand. We appeal to a learning-based solution to automatically extract such patterns from data. We view each temporal pattern as a latent basis of the high dimensional compound of aggregate energy time series, and assume each appliance can be uniquely characterized by a subset of them.

We isolate an appliance's energy use from the aggregate readings by applying its corresponding set of bases as filters over the aggregate. For example, at mealtime, the observed energy consumption should more likely come from microwave than dryer.

Learning the latent temporal bases from aggregate energy time series data is challenging - the model complexity increases exponentially with the number of sources that constitute the aggregate energy. First, the magnitude of energy consumption in different appliances varies significantly, which leads to a poor estimation of low-energy consuming appliances in traditional solutions [7, 11], as such appliances get overshadowed by high-energy consuming appliances. For example, microwave typically takes less than 5% energy consumed by dryer when they are on. Therefore, when both of them are on, the aggregated reading will be dominated by the dryer and it becomes very hard to recognize whether the microwave is on. Second, different homes consist of different appliances, and oftentimes they might include new appliances that are not previously modeled in an energy breakdown solution. For example, a small number of homes might have electric vehicles, which consume a large amount of energy for charging, while most of other homes in the same region do not have electric vehicles. Because an energy breakdown solution only takes aggregate energy as input, the existence of such unknown energy sources introduces a considerable source of error in the decomposed appliance energy, as the model tries to fit the aggregate energy with only those known appliances.

To address the aforementioned challenges, we extract the temporal bases and predict per-appliance energy consumption from aggregate energy readings via a set of convolutional neural networks (CNN) [18], which are organized in a tree structure. Thus, we name the solution TreeCNN. At each node of the tree, a CNN model is placed to reconstruct a particular type of appliance energy time series by the latent temporal bases. The root node of the tree takes aggregate energy readings as input and reconstructs its designated appliance's reading as output. The residual, i.e., the difference between its input and output, is passed to the child node as the subsequent input. The reconstruction is thus performed by recursively traversing the tree. Such an iterative procedure isolates the appliance model learning in each step while preserving all appliances as a whole, since the prediction residual is updated and passed through the tree. Thus, each appliance's usage pattern is modeled with a "purer" aggregate energy consumption to avoid the overshadow magnitude problem. Further, with such a tree structure, the unknown consumption can be modeled as a pseudo appliance to further de-noise the aggregate readings. It is known that finding the optimal tree structure is NP-complete, and thus we introduce a greedy approach to find the tree structure. It is worth to mention that our proposed solution is applicable to energy data with both higher or lower frequencies, such as minute and monthly data. The CNN nodes in the tree structure is expected to capture the temporal patterns across minutes (e.g., ON/OFF states) and across months and years (e.g., seasonal pattern). But in our evaluation, we focus on hourly data because: 1) the availability of homes with minute data is much less than those with hourly data [24, 25]; 2) the consumption of some appliances may not change much across minutes, which is somehow redundant. For monthly data, we can only capture seasonal patterns, whose granularity might not be as informative hourly breakdown results.

We use the public Dataport [20] dataset for evaluation. We compare our proposed TreeCNN solution against nine state-of-the-art baselines for energy breakdown and find TreeCNN provides the most promising performance in decomposing the hourly energy data. We also verify that the energy breakdown performance can be further improved by modeling the unknown energy source, which is only properly handled in our solution. Our empirical evaluation shows that the tree structure suggested by our greedy approach performs only 4% worse compared to the optimal order found via an exhaustive search.

2 RELATED WORK

The field of energy breakdown study was invented by George Hart in the early 80s [11]. Since then, a large number of learning-based solutions have been proposed, which can be broadly classified as: event-based and total-load models.

Event-based methods find step changes in the power signal and assign them to different appliances. Appliances turning "ON" would produce a positive step change in power and appliances turning "OFF" would produce a negative step change accordingly. Event-based methods [8] are generally used only when high sampling frequency (1 Hz or higher) is available, as the events cannot be recognized at low frequencies. They are thus ill-suited for hourly or even lower sampling rate. Event-based methods also have other shortcomings - they do not work well when appliances change state simultaneously, nor are they suited to appliances which have a highly variable power draw like electronics. Furthermore, errors (missed events and wrongly detected events) propagate forward in event-based methods, leading to generally poor energy breakdown performance. These factors generally limit such solutions' application in practice.

In contrast, total-load based methods model the aggregate consumption as a sum of constituent loads, while estimating these constituent loads at all sample points (unlike event-based methods). Our solution fails into this category. In particular, Factorial Hidden Markov Model (FHMM) has been successfully applied to this problem [17], where each appliance is modelled as a Gaussian HMM. While such total-load approaches are better suited for lower sampling rates compared to event-based approaches, one shortcoming of such approaches is that they only incorporate Markovian-type relationships in power draw (i.e., between states at adjacent time stamps) and are not suited for capturing patterns like repeated or similar energy consumption based on the hour-of-day or day-of-week. Another shortcoming is that they assume the aggregate consumption is the sum of the considered appliances, rather than all appliances in a home. This is limited by the availability of training data, as in practice it is very expensive to instrument all appliances. Often, when is high consumption resulted from unknown energy sources, such solutions give very poor performance. Our solution addresses this issue by modeling the unknown energy source as a pseudo appliance in the tree structure so as to model its temporal pattern for energy breakdown as well.

There is a line of work for energy breakdown at a monthly level. The key insight of such approaches is that common design in residential buildings creates a sparse set of features contributing to energy variation across homes. Matrix factorization [7] and kernel density estimation [6] methods have been used to exploit such sparsity. But such solutions cannot be directly applied to higher sampling rate, as their model complexity increases exponentially with the sampling frequency. Sparse coding based approaches [16] have been proposed to address these techniques' limitations on hourly data. But all such solutions assume the aggregate equals to the sum of the appliances and thus suffer under practical settings when the unknown energy consumption is high.

With the fast development in deep learning, there also exists some recent solutions that explore the capability of neural network applied in NILM. [14, 15] applied recursive neural networks (RNN) to capture the time-series dependency of the energy signals sampled at a high frequency. However, a RNN-type model mostly captures the one-dimensional Markovian relationships in power draw, but is incompetent to capture other types of temporal dependencies. For example, as we mentioned before in the hourly sampled energy data appliances like microwave can be well recognized by the time-of-day pattern, while others like dryer is easier to be modelled by day-of-week pattern. A RNN model will possibly fail to learn such periodical patterns with multiple dimension. Our solution considers time-series energy data as a high dimension compound of various temporal bases, and learns the bases from data to recognize different types of appliances from the aggregate readings.

3 DATA ANALYSIS

The goal of this section is to explore the temporal energy consumption patterns residential buildings towards the development of our proposed energy breakdown methods.

3.1 Dataset

In this work, we use the public Dataport [20] dataset. Dataport is the largest public residential home energy dataset. It contains power readings logged at minute intervals from hundreds of homes in the U.S. We use 112 days worth data from 68 homes from mid-June on-wards for the year 2015, as this period has the least amount of data issues (missing or incorrectly collected data). Besides household total consumption, we use data of five major appliances: i) air conditioning system (HVAC); ii) fridge; iii) dryer; iv) dishwasher; v) microwave. These appliances contribute significantly to the total energy consumption. Besides, they also represent a diverse class of appliances: fridge and HVAC being constantly ON appliances; HVAC being a high energy consuming appliance, dependent on external weather and thermostat settings; dishwasher and microwave being kitchen appliances; dryer and dishwasher being high power appliances typically used for a short duration. In this dataset, all appliances are monitored at a 1/60 Hz sampling rate (e.g., every minute).

3.2 Appliance usage patterns

Appliances can generally be classified into two categories [3]: i) appliances that are constantly ON - which usually run without active human intervention, such as fridge and HVAC; and ii) ON/OFF appliances - which require active human intervention and are typically run for short durations, such as washing machine and oven. When dealing with low sampling rates (like hourly), ON/OFF appliances introduce additional challenge - many of these appliances would only be used partially within an hour. This is also the main reason that existing NILM algorithms fail at this low sampling rate. To understand how significant this phenomenon is in our dataset, we studied the shortest active time interval of those five appliances in minutes across different homes (denoted as α). We found that all appliances have much shorter active time intervals than one hour: except microwave has minimum 2 minutes active interval, the others are all 5 minutes (detailed results can be found in Table 1).

Table 1: Energy statistics from Dataport dataset.

	HVAC	Fridge	Dryer	Dishwasher	Microwave
α (min)	5	5	5	5	2
δ_a	230	20	250	55	10
Active	73.9%	97.8%	4.9%	4.1%	11.3%
Max	5099.7	428.6	4364.1	1021.7	980.6
Mean	1162.7	88.6	1303.6	369.5	59.5
Std	800.2	40.2	756.2	206.5	53.1

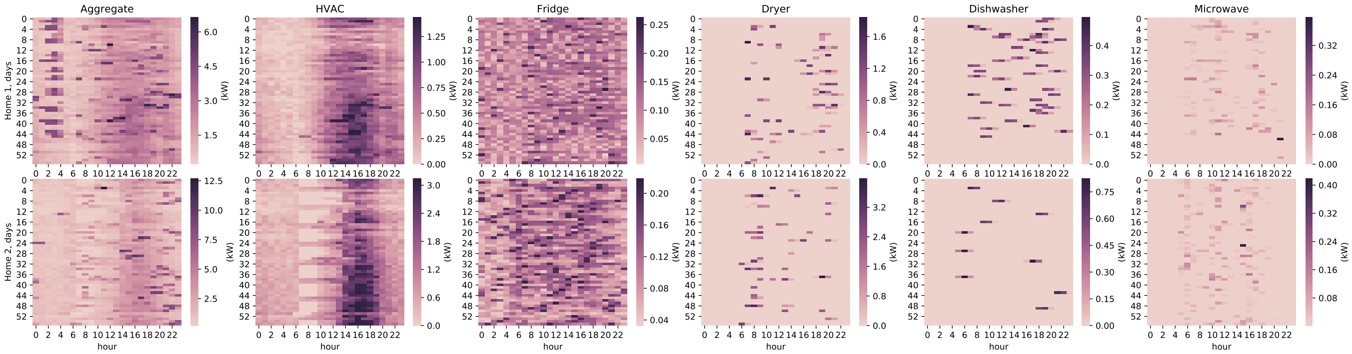
As we are performing hourly energy breakdown, the existence of short active intervals begs the question - how much energy should an appliance consume within an hour to be considered "actively used". On consultation with domain experts, we set the active threshold δ_a for each appliance a as:

$$\delta_a = \frac{\alpha_a}{60} \times \frac{1}{H} \sum_{h}^{H} \max_{d,t} E_{h,a,d,t} \quad (1)$$

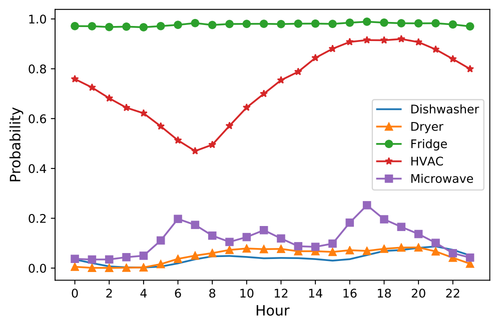
where H represents the number of homes, $E_{h,d,a,t}$ is the energy consumption reading of appliance a on day d at hour t for home h, and α_a is the minimum active time interval for appliance a. As we know, for appliance occasionally used, it's easy to get a good overall performance by giving all zero-predictions (e.g., microwave is OFF over 88% of time). However, such false negative prediction violates the original intention of energy breakdown, i.e., provide the opportunity of energy saving by informing users of how much energy each appliance consumes. With such an active threshold, we can recognize different states of each appliance and evaluate a model's prediction in two classes, i.e., error in ON/OFF states.

Basic statistics about this dataset with the active threshold are reported in Table 1. As we can notice, the constantly ON appliances, i.e., HVAC and fridge, are almost always on (active percentage: 73.9% and 97.8%); but their energy consumption patterns are quite different: fridge nearly consumes constant energy over time, while HVAC's consumption varies significantly over time (std = 800.2). For the ON/OFF appliances, such as dryer, while their use is seldom; but once used, it consumes almost the highest energy. This macro-level analysis suggests the need of different temporal bases for energy breakdown prediction across appliances.

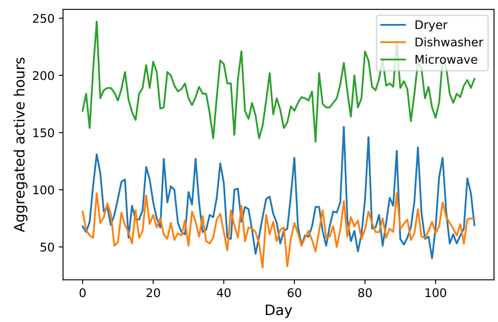
Now we move onto the detailed micro-level analysis about different appliances' temporal energy consumption. Previous works [4, 7], which studied monthly aggregated data, show that the energy consumption pattern across homes and appliances is sparse owing to the common design of residential buildings. Our data analysis suggests that such sparsity pattern also exists at an hourly and daily level, largely due to temporal human behavior patterns, such as using the microwave during the mealtime.



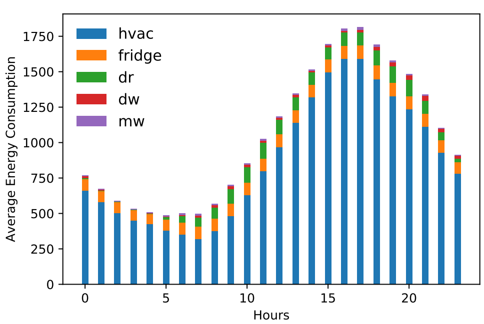
(a) Heatmap of aggregate and appliance-level energy consumptions from two randomly selected homes.



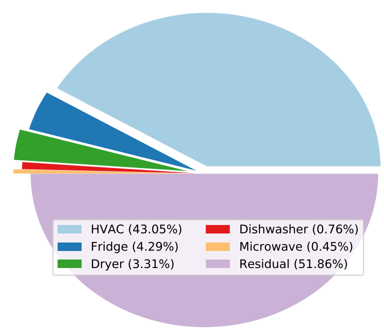
(b) Active probability.



(c) Aggregated active hours.



(d) Average energy consumption over 24 hours.



(e) Appliance consumption proportion.

Figure 1: Data analysis for residential home temporal energy consumption patterns.

Figure 1a shows the heatmap of aggregate and five appliances' energy consumption from two randomly sampled homes over 24 hours across 56 days. We can easily recognize strong hourly energy patterns within a day across these 56 days. For example, i) both homes tend to consume more energy by HVAC in the afternoon and less in the early morning; ii) fridge constantly runs with regular working peaks; and iii) dishwasher and microwave are more likely to be working at the mealtime. Figure 1b, which presents the probability of being in active state over five appliances during 24 hours across all homes, further indicates the hourly patterns, especially that sparsity also exists at a daily basis. In Figure 1a, Home 2 consumes much less HVAC energy in the morning and this pattern only appears on the weekdays. Further, different from dishwasher and microwave, people tend to use dryer periodically across days. Figure 1c shows the aggregated active hours among 68 homes in each day for the three ON/OFF appliances. It clearly shows that the total number of active hours of dryer has a peak every week while dishwasher and microwave are used on an everyday basis. These identified sparsity and temporal regularity motive us to view such time series data as a high dimensional compound instead of a one-dimension sequence.

Besides, energy consumption is also highly unbalanced among appliances (as indicated in Table 1). "Minor" appliances such as microwave are often a problem for many existing NILM algorithms owing to their small magnitude of energy consumption. A more detailed comparison is shown in Figure 1d. As we can notice that throughout a day, most energy is consumed by HVAC, which peaks in the late afternoons. For those ON/OFF appliances, such as microwave and dryer, when both them are on, the one with smaller energy consumption (e.g., microwave) is overshadowed by the larger one (e.g., dryer). And therefore it becomes even harder to differentiate their uses. Comparing to the major appliances such as HVAC, "minor" appliances like microwave are easily to be treated as noise. It should be noted that despite low contribution, simply predicting zero use is still misleading. In addition, it has been shown that detecting the energy use of these appliances, such as microwave, has been shown to help detect activities of daily living, especially useful for applications such as elderly monitoring [1].

In addition to the patterns of known appliances, it is important to note that the aggregate consumption is often not equal to the sum of the considered individual appliances' consumption. For example, in the aggregate consumption of Home 1 in Figure 1a, there is some regular high consumption in the early morning, which is not observed in the known appliances. Such unknown consumption comes from various sources, such as the living room usage, appliances like furnace, or special equipment like electric cars. Figure 1e shows the energy consumption proportion of each known appliance and unknown consumption. It is easy to recognize that in average the unknown consumption oftentimes can take up as much as 51.86% total energy in a home. However, this consumption is impossible to exhaust beforehand; and the failure to model them leads an energy breakdown algorithm to classify them to known appliances, such as misclassifying furnace's usage as HVAC's. To improve the accuracy of energy breakdown, such unknown consumption has to be carefully handled.

As a side note, this data analysis is to show the regularity and sparsity do exist in the hourly data of household energy consumption. We only perform the analysis on energy data collected in U.S., but we believe even consumption behavior varies among different cultures and geo-regions, the energy data generated should also has the sparse property, such as periodical patterns, as it is largely affected by human behavior pattern and diurnal/seasonal patterns.

4 METHODOLOGY

We study the problem of disaggregating the aggregate energy from a single home to its constituent appliances at hourly intervals. Based on previous discussions, the hourly sampled energy time series data has several important properties, i) the existence of sparsity and regularity in multiple temporal dimensions, ii) energy consumption magnitude varies significantly across appliances, and iii) the existence of unknown consumption sources. We will discuss our solutions to each of them in the following sections.

4.1 TreeCNN Model

The key intuition of TreeCNN lies in two aspects: i) the distinct and multi-dimensional temporal patterns of appliance energy constitute the sparsity and regularity in appliance-level energy use; ii) the aggregate energy is a composition of various and complicated appliance energy consumption, such that the decomposition should be performed in a joint and recursive manner to avoid the errors introduced by the magnitude problem and potential unknown

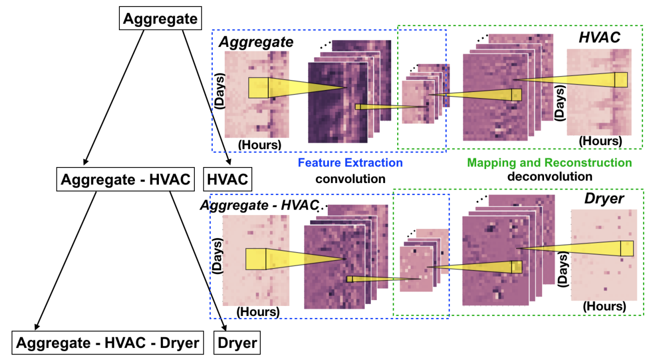


Figure 2: Our Tree-structured iterative energy breakdown approach shown for two appliances (HVAC and Dryer).

consumption. We now discuss each component of our proposed TreeCNN model in detail.

• **Convolutional Neural Network (CNN).** In Section 3, our analysis results clearly show that different appliances have distinct temporal usage patterns. Such patterns differentiate the appliance energy readings from each other and thus make energy disaggregation possible. But, simply modeling the hourly time-series energy usage data as a one-dimension sequence cannot fully describe the appliances. For example, microwave is more frequently used during the meal time (a hourly pattern), while the dryer is easier to model across days for its periodical usage (a daily pattern). Thus, this time-series of appliance energy usage should be viewed as a high dimensional compound of various temporal patterns, i.e., a set of latent bases that construct the unique characteristics.

Besides the patterns observed in the appliance usage data, there might also exist other higher order temporal patterns that cannot be simply exhausted manually. Thus, we turn to learning-based solutions to automatically extract the latent bases from data. Inspired by the successful applications of convolutional neural networks (CNN) in image analysis [18], which can be considered as a high-order matrix analysis, we appeal to CNN models to extract the temporal bases for energy time series data. The key component of CNN model is the filters that are learned in its convolutional layers to capture the spatial features of an image. In an analogy, the hourly energy readings can also be viewed as a 2-D matrix (Figure 1a) and thus can be well described by the spatial filters learned from CNNs. It's worth mentioning that Recurrent Neural Nework (RNN) is also capable to capture the patterns across time. However, it only models the one-dimensional time-series data (e.g., a Markovian relation among observations in a sequence), and easily fails to capture the periodical patterns across days in the hourly energy data.

With a CNN model, the filters learned with the energy data represent the temporal bases for energy breakdown. Thus, distinguishable filters can be learned for appliances with distinct temporal patterns. For example, filters learned on microwave may emphasize more on the hour dimension, filters for dryer will emphasize on the day dimension, and the filters for HVAC may be a compound of patterns on both dimensions. With such patterns represented as filters, the aggregate readings can be projected into its corresponding appliance usage. In Figure 2, we give an example of a CNN model which learns the mapping from aggregate readings to

HVAC consumption. The CNN model consists of two major parts, convolution and deconvolution, which correspond to feature extraction and reconstruction phrases. In the convolution phrase, CNN model takes the aggregate as input and tries to reduce it with a series of encoders into a much denser representation. In this phrase, due to the sparsity and granularity, the periodical patterns will be extracted and the input will be represented as a denser matrix with a smaller dimensionality. And then in the deconvolution phrase, the decoder performs the opposite operations that reverse the action of encoders: map the learned representations to the target and reconstruct the appliance readings, e.g., HVAC in this example.

• **Tree Structure.** Another challenge in energy breakdown is the model complexity increases exponentially with the number of sources constituting the aggregate energy. As discussed before, the aggregate reading is constituted by various energy sources, such as the constantly ON appliance (e.g., fridge), ON/OFF appliances with significant temporal patterns (e.g., microwave), and the unknowns. One consequence of the various constituent components is the magnitude issue, where some appliances can get overshadowed by others. As shown in Figure 1e, HVAC consumption is almost 100 times the consumption of microwave, which makes the microwave signal easily being overshadowed as noise.

Different from conventional energy breakdown techniques, which either estimate energy usage for each appliance independently or disaggregate the energy altogether at once, we propose a tree-structured model to extract the appliance patterns in a stage-wise manner. with respect to a recursively learned tree structure. With the tree structured model, our approach revolves around the idea of performing an iterative energy breakdown: at each iteration, we subtract out a source from the aggregate and use it as input to recognize the designated appliance. Figure 2 depicts an example of our tree-structured iterative energy breakdown approach with two appliances, HVAC and dryer. The root node of the tree takes the aggregate energy readings as input and reconstructs the HVAC consumption as its output. The difference between the input and the output will be passed to its child node as refined aggregate reading input for the next appliance, e.g., dryer in the figure. With such "purer" aggregate consumption, the magnitude problem is highly eased for the minor appliances, such as microwave, if we place them at the lower end of the tree. In the contrast, if we jointly decompose the aggregate readings into appliances' readings, the minor appliances will be easily overshadowed as they are competing with the major appliances at the same time, and given zero predictions, which violates the original intention of energy breakdown.

In our TreeCNN model, as we are reducing one appliance at each iteration, we are effectively simplifying the energy breakdown problem step by step. In each node, the CNN model performs an end-to-end learning process for the target appliance, which isolates pattern learning across appliances to avoid the overshadow problem while preserving all appliances as a whole.

• **Modeling Unknown Consumption.** In addition to the magnitude problem caused by the various constitutes of aggregate, the unknown consumption also introduces tremendous errors in energy breakdown. From the previous analysis, the unknown energy consumption comes from various sources and therefore is hard to specify beforehand. To the best of our knowledge, prior work has not yet looked at modeling the unknown energy consumption. In our tree-structured model, the unknown consumption can be simply viewed as a special appliance which consists of multi-dimension temporal patterns and therefore can be treated as a dummy appliance in all homes. Modeling the unknown consumption makes it possible to remove such dominant energy from the true aggregate consumption, which leads to a more accurate estimation of the observed appliances.

4.2 Tree Order

In this section, we discuss the order to process appliances in our tree structured model.

Given N appliances to disaggregate in each residential home, we would have $N!$ possible tree structures. For a typical residential home at the U.S., one can usually expect 7-10 major appliances of interest in monitoring, such as fridge, air conditioning, and washing machine. For any larger values of N, exhaustively finding the "optimal" order can be computationally expensive, where "optimality" is defined as per given energy breakdown metric M. And the error of one decomposition will be propagated through the tree structure. Thus the order of processing appliance disaggregation is essential to our model. We thus propose a greedy algorithm to find a suitable tree order to mitigate the error, and reduce the search space. The key operation in such a greedy approach is to estimate the overall performance of a tree order while having only partial/local information, i.e., at each node in the tree, we wish to estimate the final energy breakdown performance over all appliances via the performance of a single appliance at a given level of the tree. And the key idea lies in the inverse propensity weighting scheme [19, 26], which is extensively explored in importance sampling techniques [10].

Let us assume that we have N appliances in total and we use a metric $M(E(a_i), \hat{E}(a_i))$ to compute the error for appliance a_i where $\hat{E}(a_i)$ and $E(a_i)$ denote the estimated energy usage and the ground truth in appliance a_i. In the first iteration, we will create N candidate splits, e.g., HVAC v.s., the rest. Among these N models, we select k with the smallest *estimated energy breakdown error* ($EEBE^{GR}$) on the validation set. In the next iteration, we will create $(N-1)$ subtrees for each of the selected k parent trees. The selection repeats until we have constructed the whole tree.

Our local metric $EEBE^{GR}$ is used to estimate the overall energy breakdown error, before the whole tree has been constructed. This metric is calculated as the ratio between a chosen energy breakdown metric calculated for an appliance and the proportion of energy consumed by this appliance:

$$EEBE^{GR}(a_i) = \frac{M(E(a_i), \hat{E}(a_i))}{\frac{\sum_h \sum_d \sum_t E(h, a_i, d, t)}{\sum_h \sum_a \sum_d \sum_t E(h, a, d, t)}} \tag{2}$$

The rationale behind $EEBE^{GR}$ is that it assumes if an appliance has an error e and contributes x proportion of aggregate, then the aggregate would have an expected error of $\frac{e}{x}$ from this appliance. Thus, we are able to estimate the error over the complete tree structure by such an estimation over the prediction error of each individual appliance during the construction of the tree. In such way, we get the error of the entire tree before it is constructed, which makes the sequential decisions of tree order possible.

5 EMPIRICAL EVALUATIONS

In this section, we evaluate the proposed TreeCNN model on the hourly data collected from 68 homes over 112 days in the Dataport dataset. We perform a series of NILM evaluations to validate the performance of our TreeCNN model against several state-of-the-art models in energy breakdown in energy breakdown estimation accuracy.

5.1 Baselines

We first describe the baselines included for comparisons.

• **Mean Energy:** Previous research [16] indicates that a simple baseline like mean energy is effective. This baseline computes the predicted energy of an appliance as the mean energy of that appliance in the train set.

• **Factorial Hidden Markov Model (FHMM):** FHMMs [17] model each appliance as a Gaussian hidden Markov model and couple the individual appliance HMM in a factorial structure.

• **Tensor Factorization:** Canonical polyadic (CP) decomposition [4] is used to factorize the energy tensor into latent matrices. The authors mentioned that CP decomposition tends to concentrate on high energy appliances, while ignoring learning components for low energy appliances. And they proposed a modification to CP called Modified CP (MCP) to mitigate the shortcoming.

• **Sparse Coding:** Sparse coding [16] model approximates the bases and activations for each appliance with sparse constraint. And then estimate the activations with aggregate signal and the concatenated bases. The authors also proposed a structured prediction based method called discriminative sparse coding (DSC) to optimizing the energy breakdown performance.

• **Recurrent Neural Networks (RNN):** Several neural based approaches [14] have been proposed for energy breakdown problem, which captures the time-series dependency of the energy signals. For comparison, we performed the decomposition with individual RNN model and TreeRNN model to verify the effectiveness of CNN model and tree structure in our TreeCNN model.

• **Convolutional Neural Networks (CNN):** We use individual CNNs and Joint CNN model, which decompose the aggregate readings into appliances' readings altogether at once, as baseline methods, given that our approach uses a tree-structure over the CNN nodes. Individual CNNs create one CNN model to estimate each individual appliance energy consumption separately, while Joint CNN estimates them all together at once.

5.2 Experimental settings

5.2.1 Dataset related settings. We use a 5-fold cross-validation strategy in the experiments. The final 20% of the train set in each run is set for validation purpose.

5.2.2 Approach settings. For each baseline and our method, the optimal parameters are learned via an exhaustive grid search. The optimal parameters that give the best performance on the validation set are used for testing. For FHMM model, we vary the number of states per appliance from 2 to 5 according to [5, 28]. CP and MCP are optimized with Adagrad [4], and we vary the rank of latent factors from 1 to 12. For sparse coding models, we vary the number of latent factors from 1 to 50.

We implemented all neural network models with PyTorch [22]. For RNN-based models, we have the following parameters: cell type: {GRU, LSTM, RNN} number of hidden units: {20, 50, 100}; number of layers: {1, 2, 3}; bidirectional: {True, False}, number of iterations: {1000, 2000, 3000}. For CNN-based models, as we only have limited training data, complex network design will easily cause overfitting. In this model, we have the encoders consist of two convolutional layers and two deconvolutional layers with normalization [12] to accelerate the training process, and ReLU to accelerate the training process and introduce non-linearity. For individual CNN model, we choose the learning rate from {1e-2, 1e-1, 1} and the number of iterations from {1000, 2000, 3000}. For TreeRNN and TreeCNN model, one important aspect is the appliance order. We perform both exhausted search and greedy search described before. Since we have five appliances, we use top-k = 3 results at each stage of greedy search.

For all baselines barring the neural network model, we clamp the estimated appliance energy to a maximum of the observed aggregate energy. Such a post-processing step reduces the error for high energy consuming appliance such as HVAC. For neural network based methods, we perform such a clamping within the model estimation stage.

5.2.3 Metric. Based on prior literature, we evaluate the performance with mean absolute error (MAE) [5]. If the ground-truth and estimated energy for home h, appliance a, day d and hour t are given by $E(h, a, d, t)$ and $\hat{E}(h, a, d, t)$, for appliance a, the MAE is given as:

$$MAE(a) = \frac{\sum_h \sum_d \sum_t |E(h, a, d, t) - \hat{E}(h, a, d, t)|}{H \times D \times T} \tag{3}$$

where H, D, T indicate the total number of homes and days, and hours in a day. We use the average MAE across appliances to measure accuracy. For the model comparison, we use the mean MAE to represent the accuracy.

$$meanMAE = \frac{\sum_a MAE(a)}{A} \tag{4}$$

where A represent the number of appliances. Lower mean MAE indicates better energy breakdown performance. As shown before, in some ON/OFF appliances, such as dryer, the active time is generally low (< 5%). The MAE alone cannot fully reflect the performance, as zero predictions can also give a good MAE for them. Thus, we also separate MAE into two parts, corresponding to the active and inactive states based on the ground-truth, and the threshold is reported in Table 1.

5.3 Experiment Results

In almost all the publicly available datasets, it is seldom the case that the aggregate energy consumption equals the sum of the appliances that is collected. Moreover, there is always some extra energy expended in the internal household wiring. In the following sections, we first test the energy breakdown capabilities of different algorithms in an ideal case, where we set the aggregate energy as the sum of selected appliances, and then perform the same experiments on the true aggregate dataset where the unknown consumption is included. Further, we compare the baselines and our TreeCNN model with and without modeling the unknown consumption to

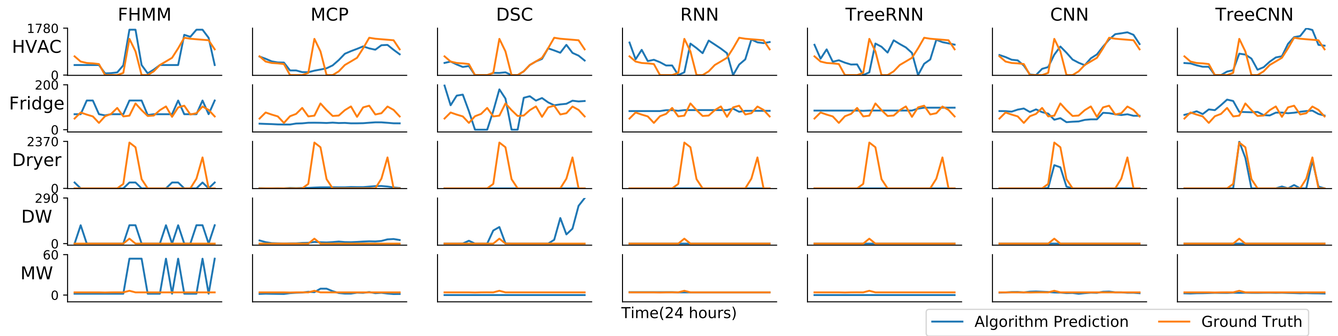


Figure 3: Comparison of a few baseline algorithms for a sample day, where DW stands for dishwasher and MW stands for microwave.

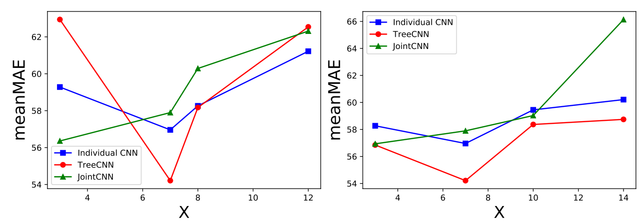
show the effectiveness of unknown consumption modeling. Last, we report the results of greedy algorithm on tree order estimation.

5.3.1 Filter size in CNN models. In CNN-based models, size of the filters plays an important role in capturing the temporal patterns. A large filter might overlook the detailed temporal feature and miss the essential details in the input data, while a small filter might provide too much redundant information and unnecessarily increase the number of parameters. Thus, determining the most suitable size of a filter is a must-do step in CNN models. We first explored the effect of different sizes of filters in the first convolutional layer of our CNN models. The results are shown in Figure 4. We can clearly observe that the performance of CNN models are quite sensitive with the size of filters. Besides, as discussed in Section 3, there exist some periodical temporal patterns in the hourly energy data, and different appliances can be differentiated by latent temporal bases due to such distinct temporal features. And the results here follow what we discussed. As shown in the figure, when the size of filters equals to 7×7, most models achieve the best performance (i.e., lowest error), as such filters can well capture the periodical patterns across hours and days, such as time of a day patterns for appliances like microwave, and day of a week patterns for appliances like dryer. Accordingly, when we have small sized filters, the model might miss some periodical patterns across the multiple dimensions; and when we have large sized filters, it might fail to capture the local features and also generate redundant information.

In the following experiments, the filters of each layer in CNN models are set to 7×7 and 2×2. And the decoder is a mirrored version of the encoders with 2 deconvolutional layers. And we use the L1-loss as the objective function. We fix CNN architecture and only tune the hyper-parameters.

5.3.2 TreeCNN v/s baselines on data with artificial aggregation - ideal case. First, we test the energy breakdown capabilities of different algorithms in the ideal case, when the sum of the appliances equals the aggregate. We simulate this setup by manually setting the artificial aggregate. Various previous studies have used such a setup for evaluation [16, 27, 28]. Our main results presented in Table 2 show that our TreeCNN algorithm has better prediction performance compared to all the baselines. The p-value is calculated between the predictions of TreeCNN and the second best model for each column. As an illustration, Figure 3 shows the energy breakdown estimation from a set of baselines for a randomly chosen day of one randomly selected home.

We now discuss why the baseline algorithms fail to provide an accurate energy breakdown even in the ideal setting. FHMM assumes that we can model appliances with a Gaussian HMM using discrete states, which considers the transition and emission probabilities between states on a time-series data. While it can well model appliances have time-series dependency, such as HVAC and fridge, it is not well suited to appliances such as dryer which is sparsely



(a) Filter size: $7 \times X$ **(b) Filter size: $X \times 7$**

Figure 4: Effect of filter size tuning on CNN models.

Table 2: Mean MAE on artificial aggregate dataset.

	MAE	MAE on Active	MAE on Inactive
FHMM	114.99	360.76	80.34
CP	106.19	390.39	96.11
MCP	103.46	390.90	96.93
SC	92.20	411.93	28.61
DSC	172.95	388.29	**15.03**
Mean	100.34	404.03	49.13
RNN	67.34	388.90	37.87
TreeRNN	64.26	383.84	36.79
CNN	56.96	310.51	41.53
Joint-CNN	57.90	332.45	42.07
TreeCNN	**51.64***	**261.30***	40.05

* p−value < 0.05

Table 3: Mean MAE on true aggregate dataset.

	MAE	MAE on Active	MAE on Inactive
FHMM	134.99	400.76	82.94
CP	123.90	414.01	103.10
MCP	125.30	412.33	107.81
SC	245.70	515.54	215.57
DSC	218.43	521.35	190.84
Mean	126.86	421.50	102.22
RNN	97.80	435.38	60.32
TreeRNN	94.52	433.01	54.85
CNN	89.15	417.85	68.28
Joint-CNN	91.35	429.87	69.26
TreeCNN	**86.94***	**391.50***	**61.69***

* p−value < 0.05

used. Further, we can observe that FHMM predicts the dishwasher and microwave to be active much more often than they actually are. MCP and DSC algorithms are both of a similar vein focusing on learning various "basis" or temporal patterns of energy consumption. Both these algorithms perform reasonably well in general. However, they are not well-tuned for the instances when energy consumption patterns differ from the average patterns. Moreover, they are poor at capturing the active states of the ON/OFF appliances. From Table 2, we can observe that for MAE on Inactive, SC and DSC have better performance. This is because the design of their two-stage decomposition adds the equality constraint between the sum of appliances and aggregate readings. The mean baseline though is very simple, performs reasonably well. However, it should be noted that it does so as it models the appliances to be mostly off and thus a low MAE for the inactive cases, but high MAE for the active cases. Previous research has also validated similar observations a different data set [16].

Compared with the non-neural network baselines, the MAE is largely reduced by RNN and CNN based models. Between the individual neural network models, CNN outperforms RNN. The main reason is with the learned filters, CNN can capture the multi-dimensional usage patterns, e.g., hourly pattern and daily pattern in our dataset, while RNN treats the energy readings as one-dimensional time-series ignoring the periodical usage patterns. For example, in Figure 3 with CNN models, the active states of the ON/OFF appliances, such as dryer, is well detected, while RNN gives zero predictions. From Table 2 and Figure 3, the performance of neural network models are both improved with the tree structure, i.e., decreased MAE; by considering the relationship among appliances, TreeCNN makes more accurate energy estimations than all baselines. Though JointCNN also encodes the relationship among appliances, direct decomposition could not overcome the magnitude problem, which generates significant larger error in MAE on Active as the "Minor" appliances are easily overshadowed and given zero-prediction.

5.3.3 TreeCNN v/s baselines on data with true aggregation - real-world case. Now, we evaluate the models on the true aggregate dataset, which provides a more realistic energy breakdown evaluation of the state-of-the-art solutions.

Table 4: Mean MAE on true aggregate dataset.

	MAE	MAE on Active	MAE on Inactive
FHMM	134.99	400.76	82.94
DSC	218.43	521.35	190.84
RNN	97.80	435.38	60.32
CNN	89.15	417.85	68.28
Joint-CNN	91.35	429.87	69.26
TreeCNN	**86.94**	**391.50**	**61.69**

Table 5: MAE of best baseline and TreeCNN with/without unknown consumption modeling. (UC: Unknown Consumption, DW: Dishwasher, MW: Microwave)

	HVAC	Fridge	Dryer	DW	MW	Average
TreeRNN w.o. UC	351.83	29.88	66.93	15.45	8.53	94.52
TreeRNN w. UC	337.69	30.02	67.16	15.43	8.43	91.75
TreeCNN w.o. UC	306.38	33.74	70.09	15.49	9.00	86.94
TreeCNN w. UC	296.11	33.34	69.01	15.45	8.83	**84.55**

Table 4 shows the energy breakdown performance for different algorithms when using true aggregate. We should note that the mean baseline shows a different result than that in the artificial aggregate dataset since we applied the post-processing (clamping appliance energy ≤ aggregate) to ensure that appliance energy consumption does not exceed the aggregate. We can see that with true aggregate, TreeCNN model still outperforms the other baselines.

Comparing the results from these two settings, we can clearly notice that all algorithms show poorer performance with true aggregate. As discussed before, this can be explained by the high amount and variety of unknown consumption. Unknown consumption is prevalent in energy breakdown owing to limited technology and resources for instrumentation. If we can better model the unknowns, the performance of energy breakdown should improve, as the problem becomes closer to the ideal case.

In our TreeCNN, the unknown sources can be simply treated as a special appliance which is also consisted with multiple high dimensional temporal patterns. The model will automatically learn the complex latent bases for such unknown sources via filters in CNN as well. Table 5 shows the improvements when we consider the unknown energy sources in the model. The experiment is performed on the true aggregate dataset. A model with Unknown Consumption means we treat the unknowns as a special node in the structure and without Unknown Consumption means we simply ignore it. Though the MAE difference is small, we performed paired t-test on the predicted energy consumption of TreeCNN with and without unknown sources, and the test statistics show almost all the estimations are significantly improved except for the dryer. In this work, we only have one model for the unknown sources while the unknown energy might come from a combination of various sources. We leave this exploration to the future work.

5.3.4 TreeCNN with greedy v/s exhaustive tree orders. Table 6 compares the mean MAE performance of our greedy tree-order algorithm with the best, worst and average order found by exhaustive search on three settings: i) artificial aggregate data; ii) true

Table 6: TreeCNN performance under different tree orders. (UC: Unknown Consumption, agg: aggregate)

	Worst	Average	Greedy	Best
Artificial agg	84.42	68.72	54.21	51.64
True agg w.o. UC	105.60	96.54	88.38	86.94
True agg w. UC	110.75	98.64	87.21	84.55

Table 7: Comparison of the tree order learned via our greedy algorithm to the best and the worst order found via an exhaustive search on artificial aggregate dataset.

Fold	Best order	Worst order	Greedy order
1	(DW, MW, FR, HV, DR)	(HV, MW, DW, DR, FR)	(FR, MW, HV, DR, DW)
2	(FR, MW, HV, DW, DR)	(DW, HV, FR, MW, DR)	(FR, MW, HV, DW, DR)
3	(FR, HV, DW, DR, MW)	(HV, DR, MW, DW, FR)	(DW, FR, HV, DR, MW)
4	(MW, FR, HV, DW, DR)	(HV, DR, FR, MW, DW)	(DW, MW, FR, HV, DR)
5	(FR, DW, HV, MW, DR)	(DW, HV, DR, FR, MW)	(DR, MW, FR, DW, HV)

Table 8: MAE on energy data sampled every 15 minutes.

	HVAC	Fridge	Dryer	DW	MW	Mean
MTF	8.70	0.80	0.34	1.64	0.20	2.34
SC	9.73	1.26	0.81	1.48	0.84	2.82
FHMM	7.49	0.98	0.40	1.98	0.67	2.31
RNN	3.49	**0.64**	0.27	1.01	0.17	0.11
TreeRNN	3.28	0.65	0.24	0.95	0.19	1.06
CNN	3.32	0.71	0.23	1.01	0.13	1.08
TreeCNN	**2.98**	0.65	**0.23**	**0.91**	**0.13**	**0.98**

aggregate data without unknown sources modelling; iii) true aggregate data with unknown sources modelling. It can be seen that our greedy algorithm performs substantially better than the average and is only about 4% worse compared to the best order found via exhaustive enumeration. We now dive deeper into the different tree orders. In Table 7, we compare the best, worst and greedy order for the five folds. We can find that HVAC occurs in the initial two positions in all the five folds for the worst order. We believe that placing HVAC in the first few positions in the tree structure gives poor energy breakdown performance, as the HVAC energy is easily to be over-estimated to "eat" up the energy of other appliances. The greedy order is similar ones as the best order, for example, both tend to have fridge in one of the initial positions and the dryer in one of the latter.

5.3.5 Generaliability of TreeCNN.. As mentioned before, our proposed TreeCNN model is applicable to energy data with both higher or lower frequencies, as the CNN nodes in the tree structure will capture the temporal patterns across minutes or hours. We thus apply some of the best baselines and our model to energy data sampled with 15 minutes and 3 hours and report the results in Table 8 and 9. It is worth mentioning that the errors are in different scales in these two tables, as data sampled every 15 minutes is much smaller than that sampled every 3 hours. As shown in the results, comparing with the existing baselines, our TreeCNN model gives the best or comparable performance on appliance-wise and overall estimation quality with energy data sampled with a higher or lower rate. Even with different sampling rates, there still exist the multi-dimensional temporal patterns in the energy data, that is why TreeCNN is still applicable for energy breakdown. Because the magnitude problem is natural in energy breakdown, directly decomposing the aggregate readings still gives worse performance than iterative decomposition with tree structure.

Table 9: MAE on energy data sampled every 3 hours.

	HVAC	Fridge	Dryer	DW	MW	Mean
MTF	991.82	104.33	319.68	67.94	32.56	303.26
SC	1077.92	90.24	287.09	89.27	38.66	316.64
FHMM	1134.86	96.53	475.06	107.33	42.26	371.21
RNN	949.36	75.76	202.99	37.01	33.06	259.64
TreeRNN	905.00	71.64	208.81	33.52	**20.22**	249.04
CNN	886.03	76.78	203.28	46.83	24.01	247.39
TreeCNN	**836.97**	**70.10**	**161.32**	**33.07**	20.72	**224.44**

Like in image processing, we can view the data sampled with different frequencies as images with different resolutions. Energy readings sampled every 15 minutes should contain more detailed information such as the states changes of appliances, especially for ON/OFF appliances such as dryers; and energy data sampled every 3 hours are more likely to show the major patterns. Comparing with the results in previous sections where dryer has higher error than fridge on the hourly data, with data sampled every 15 minutes, all the model gets better predictions on dryer than fridge, which indicates that for appliances like dryer, besides the periodical patterns across days, the detailed information of state changes is another key feature that distinguishes dryer from other appliances.

6 CONCLUSIONS

In this paper, we presented a new approach for hourly energy breakdown. Our data analysis revealed that hourly energy data has notable high-dimensional sparsity and temporal regularity, which can be exploited for energy breakdown by learning their temporal bases. We introduced a tree-structured CNN model to estimate such temporal patterns and handle some of the shortcomings of existing methods, particularly when the unknown consumption is high. Empirical evaluation on a real-world household energy data set confirmed the effectiveness of our solution. With the vast amount of hourly smart meter data, we believe our approach has the scope to be scaled to millions of homes.

There are a few future extensions that we would like to explore. First, TreeCNN currently treats the residual as one dummy appliance. However, residual could be a compound of various sources of energy consumption. We can introduce several residual models, or using prior models [21] to first extract latent appliances and generate a combined residual estimation. Second, our current approach does not fully incorporate the dependencies that might exist between different appliances (e.g., correlation between dryer and washing machine). We can incorporate such dependencies by creating additional links between different appliances, giving us a more general graph. Third, TreeCNN only takes energy data as an input. However, energy consumption patterns are dependent on a variety of factors such as external temperature, household area, etc. In the future, we would like to incorporate such factors to guide our learning process.

REFERENCES

[1] José Alcalá, Oliver Parson, and Alex Rogers. 2015. Detecting anomalies in activities of daily living of elderly residents via energy disaggregation and cox processes. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*.

[2] K Carrie Armel, Abhay Gupta, Gireesh Shrimali, and Adrian Albert. 2013. Is disaggregation the holy grail of energy efficiency? The case of electricity. *Energy Policy* 52 (2013), 213–234. https://doi.org/10.1016/j.enpol.2012.08.062

[3] Sean Barker, Sandeep Kalra, David Irwin, and Prashant Shenoy. 2013. Empirical characterization and modeling of electrical loads in smart homes. In *IEEE IGCC*. Arlington, VA, USA. https://doi.org/10.1109/IGCC.2013.6604512

[4] Nipun Batra, Yiling Jia, Hongning Wang, Kamin Whitehouse, et al. 2018. Transferring Decomposed Tensors for Scalable Energy Breakdown across Regions. *AAAI* (2018).

[5] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring. In *Fifth International Conference on Future Energy Systems*. Cambridge, UK. https://doi.org/10.1145/2602044.2602051 arXiv:1404.3878

[6] Nipun Batra, Amarjeet Singh, and Kamin Whitehouse. 2016. Gemello: Creating a Detailed Energy Breakdown from Just the Monthly Electricity Bill. In *SIGKDD 2016*. https://doi.org/10.1145/2939672.2939735

[7] Nipun Batra, Hongning Wang, Amarjeet Singh, and Kamin Whitehouse. 2017. Matrix Factorisation for Scalable Energy Breakdown.. In *AAAI*. 4467–4473.

[8] Mario E Berges, Ethan Goldman, H Scott Matthews, and Lucio Soibelman. 2010. Enhancing electricity audits in residential buildings with nonintrusive load monitoring. *Journal of industrial ecology* 14, 5 (2010), 844–858.

[9] Samuel DeBruin, Branden Ghena, Ye-Sheng Kuo, and Prabal Dutta. 2015. Powerblade: A low-profile, true-power, plug-through energy meter. In *Sensys 2015*.

[10] Arnaud Doucet, Nando De Freitas, and Neil Gordon. 2001. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*. Springer, 3–14.

[11] George William Hart. 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12 (1992), 1870–1891. https://doi.org/10.1109/5.192069

[12] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*.

[13] Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler. 2009. Design and implementation of a high-fidelity ac metering network. In *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*. IEEE, 253–264.

[14] Jack Kelly and William Knottenbelt. 2015. 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*. ACM, 55–64.

[15] Jihyun Kim, Thi-Thu-Huong Le, and Howon Kim. 2017. Nonintrusive load monitoring based on advanced deep learning and novel signature. *Computational intelligence and neuroscience* 2017 (2017).

[16] J. Z. Kolter, S. Batra, and A. Y. Ng. 2010. Energy Disaggregation via Discriminative Sparse Coding. In *NIPS 2010*. Vancouver, BC, Canada.

[17] J. Z. Kolter and T. Jaakkola. 2012. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. La Palma, Canary Islands.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[19] Roderick JA Little. 1988. Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics* 6, 3 (1988), 287–296.

[20] Oliver Parson, Grant Fisher, April Hersey, Nipun Batra, Jack Kelly, Amarjeet Singh, William Knottenbelt, and Alex Rogers. 2015. Dataport and NILMTK: A building data set designed for non-intrusive load monitoring. In *GlobalSIP 2015*. IEEE.

[21] Oliver Parson, Siddhartha Ghosh, Mark J Weal, and Alex Rogers. 2012. Non-Intrusive Load Monitoring Using Prior Models of General Appliance Types.. In *AAAi*.

[22] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. 2017. Pytorch. (2017).

[23] Luis Pérez-Lombard, José Ortiz, and Christine Pout. 2008. A review on buildings energy consumption information. *Energy and buildings* 40, 3 (2008), 394–398.

[24] Staff Report. 2017. 2017 Assessment of Demand Response and Advanced Metering. (2017).

[25] Kamalanath Bandara Samarakoon. 2012. *Use of smart meters for frequency and voltage control*. Ph.D. Dissertation. Cardiff University.

[26] Jeffrey M Wooldridge. 2007. Inverse probability weighted estimation for general missing data problems. *Journal of Econometrics* 141, 2 (2007), 1281–1301.

[27] Mingjun Zhong, Nigel Goddard, and Charles Sutton. 2014. Signal aggregate constraints in additive factorial HMMs, with application to energy disaggregation. In *NIPS 2014*.

[28] Mingjun Zhong, Nigel Goddard, and Charles Sutton. 2015. Latent Bayesian melding for integrating individual and population models. In *NIPS 2015*.