A Tree-Structured Neural Network Model for Household Energy Breakdown

Yiling Jia
University of Virginia
Charlottesville, VA
yj9xs@virginia.edu

Nipun Batra
IIIT Gandhinagar
Gandhinagar, India
nipun.batra@iitgn.ac.in

Hongning Wang
University of Virginia
Charlottesville, VA
hw5x@virginia.edu

Kamin Whitehouse
University of Virginia
Charlottesville, VA
whitehouse@virginia.edu

Abstract

Residential buildings constitute roughly one-fourth of the total energy use across the globe. Numerous studies have shown that providing an *energy breakdown* 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 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 TreeCNN model 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, which can be well captured by filters learned in CNNs; ii) tree structure isolates the pattern learning of each appliance that helps avoid magnitude variance problem, while preserves 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 outperformed seven existing baselines on a public benchmark dataset with lower estimation error and higher accuracy on detecting the active states of appliances.

CCS Concepts

• **Human-centered computing → Ubiquitous computing**; *Empirical studies in ubiquitous and mobile computing*.

Keywords

energy breakdown; convolutional neural networks

ACM Reference Format:

Yiling Jia, Nipun Batra, Hongning Wang, and Kamin Whitehouse. 2019. A Tree-Structured Neural Network Model for Household Energy Breakdown. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313405

1 Introduction

Residential buildings constitute roughly one-fourth of the total energy usage across the global [20]. Studies have shown that providing an energy breakdown can motivate behavioral changes, potentially reducing energy consumption by 15% [2].

WWW'19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313405

Various energy breakdown approaches have been proposed since the pioneering work on non-intrusive load monitoring (NILM) [9]. NILM algorithms are designed for high-frequency data (sampling frequencies > 1/60 Hz), and do not apply when dealing with low sampling rates. However, high-frequency sensors are expensive; and smart meter specifications [21] across the world suggest that the largest proportion of smart meters sample at an hourly rate. This urges the need to develop 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 energy consumption [4, 6, 7]. The key idea is that common design patterns create a shared structure in residential buildings and give 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 structure. Nevertheless, such techniques cannot be directly applied to higher sampling rates, which rapidly increase the dimension of observations and model complexity.

Our extensive data analysis on a large public U.S. residential energy dataset suggests that sparsity and temporal regularity also exist in hourly appliance energy usage, such as the time of a day, day of a week. This motivates us to view such time series data as a high dimensional compound, rather than just a one-dimension sequence. For example, time of a day might differentiate the use pattern of microwaves from other appliances, while the day of a week might indicate usage pattern of dryers. 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 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, and assume each appliance can be uniquely characterized by a subset of them. The energy use of each appliance can be isolated from the aggregate readings by applying its corresponding set of bases. For example, at mealtime, the observed energy consumption should more likely come from a microwave than a dryer.

In this paper, we perform household energy breakdown at an hourly rate. We extract the temporal bases and predict the appliance energy consumption from aggregate energy readings via a set of convolutional neural networks (CNN) [15], which are organized in a tree structure. Thus, we name the solution TreeCNN. At each node, a CNN model is placed to reconstruct appliance energy. The root node of the tree takes aggregate energy reading 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 its 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. Thus, each appliance's usage pattern is modeled with "refined" aggregate energy consumption to avoid the overshadow magnitude problem. Further, with such a tree structure, the unknown consumption can be modeled as a special 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.

We used the public Dataport [17] dataset for evaluation. We compared TreeCNN against nine state-of-the-art baselines and found TreeCNN provides the most promising performance. Our 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 related work in energy breakdown can be broadly classified as: event-based and total-load based learning approaches.

Event-based methods [8] find step changes in the power signal and assign them to different appliances. Such methods are generally used when high sampling frequency is available, as the events cannot be recognized at low frequencies. Besides, they do not work well when appliances change states simultaneously, nor for appliances that have a highly variable power draw like electronics.

Total-load based methods model the aggregate consumption as a sum of constituent loads, while estimating these constituent loads at all sample points. Factorial Hidden Markov Model (FHMM) has been successfully applied to this problem [14], where each appliance is modelled as a Gaussian HMM. However, it only incorporates Markovian-type relationships in power draw and is not suited for capturing repeated patterns. There is a line of work for energy breakdown at a monthly level. The key insight of such approaches is that common design for buildings creates a sparse set of features contributing to energy variation across homes. Matrix factorization [7] and kernel density estimation [6] 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 [13] 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.

More recently, neural network based approaches for energy breakdown have been proposed: [11, 12] applied recurrent neural networks (RNN) to capture the time-series dependency of the energy signals sampled at a high frequency. However, a RNN model captures the one-dimension relationships in power draw, but is incompetent to capture other types of temporal dependencies. For example, in the hourly sampled 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. 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 of Appliance Usage Patterns

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

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

In this work, we use the public Dataport [17] dataset, which 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 used 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). We use the data of household total consumption and five major appliances: i) air conditioning system (HVAC); ii) fridge; iii) dryer; iv) dishwasher; v) microwave. These appliances contribute significantly to the total consumption. Besides, they also represent a diverse class of appliances: background (fridge) v.s. interactive (microwave), weather dependent (HVAC) v.s. time dependent (dryer, dishwasher).

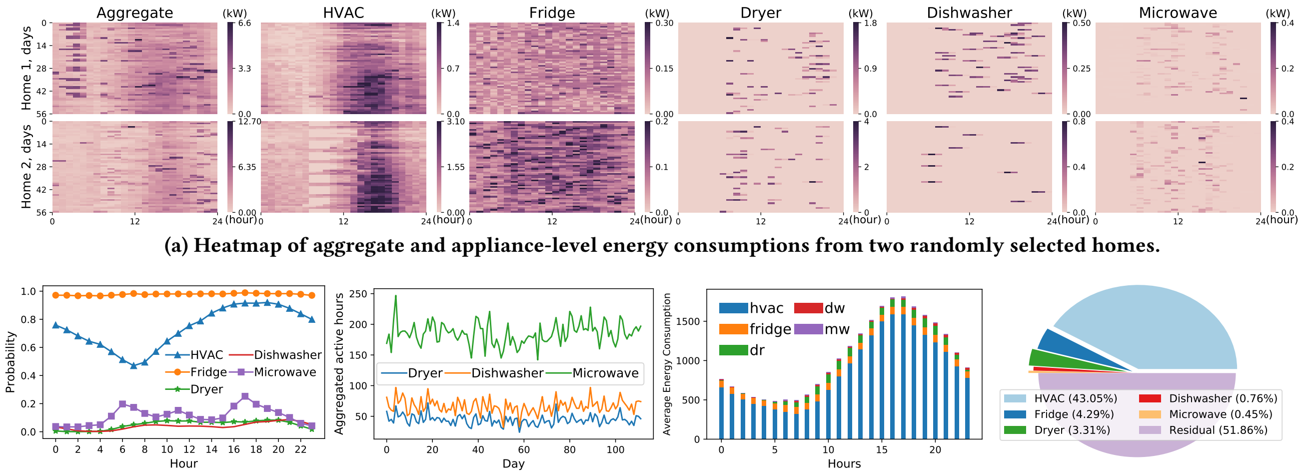
Our focused appliances can generally be classified into two categories [3]: i) appliances that are constantly ON, such as fridge; and ii) ON/OFF appliances, such as washing machine. When dealing with low sampling rates, ON/OFF appliances introduce additional challenges - many of these appliances would only be used partially within an hour, which is the main reason that existing NILM algorithms fail at a low sampling rate. To understand the significance of this phenomenon in our dataset, we studied the shortest active time interval (α) of the 5 appliances (detailed results are in Table 1).

For 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 as "actively used". On consultation with domain experts, we set the active threshold δ_a for each appliance a as:

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

where H represents the number of homes, $E_{h,d,a,t}$ is the energy consumed by appliance a on day d at hour t for home h, and α_a is the minimum active time for appliance a. As we know, for appliance occasionally used, it is easy to get 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 appliances and evaluate a model's performance in two classes, i.e., error in ON/OFF states.

The basic statistics about this dataset with the active threshold are reported in Table 1. 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 different: fridge consumes roughly constant energy over time, while HVAC's consumption varies significantly (std = 800.2). For the ON/OFF appliances, such as dryer, it is seldom used, but once used, it consumes almost the highest energy. This macro-level analysis suggests the need of different temporal bases across appliances.



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

(b) Active probability. **(c) Aggregated active hours.** **(d) Average consumption.** **(e) Consumption proportion.**

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

Previous works [4, 7] show that the energy consumption pattern is sparse owing to the common design of residential buildings. Our data analysis suggests that due to the temporal human behavior patterns, such sparsity also exists at an hourly and daily level. Figure 1a shows the aggregate and five appliances' energy consumption from two randomly sampled homes over 24 hours across 56 days. We can recognize strong patterns within a day across these 56 days: i) both homes tend to consume more energy by HVAC in the afternoon and less in the morning; ii) fridge constantly runs with regular working peaks; and iii) dishwasher and microwave are more likely to be used at the mealtime. Figure 1b, which presents the probability of appliances being in active state during 24 hours, further indicates the hourly patterns. Besides, Home 2 consumes less HVAC energy in the morning and this pattern only appears on the weekdays. Further, people tend to use dryer periodically across days. Figure 1c shows the aggregated active hours across homes in each day for the ON/OFF appliances. It shows that, for dryer, the total number of active hours has a peak every week while dishwasher and microwave are used on an everyday basis.

Besides, energy consumption is highly imbalanced among appliances. "Minor" appliances are often a problem for many existing NILM algorithms owing to their small magnitude of consumption. A detailed comparison is shown in Figure 1d. We can notice that throughout a day, most energy is consumed by HVAC. For those ON/OFF appliances, such as microwave and dryer, when both are on, the one with smaller energy consumption is overshadowed by the larger one, which makes it even harder to differentiate their uses, as "minor" appliances can be confused with "noise". It should be noted that despite low contribution, simply predicting zero use is misleading, and detecting the energy use of these appliances has been shown useful for applications such as elderly monitoring [1].

In addition to the 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, the aggregate consumption of Home 1 shows 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, or electric cars. Figure 1e shows the energy consumption proportion of different energy sources, where the unknown consumption can take up 51.86% total energy in a home, and the failure to model them leads an algorithm to classify them to known appliances. To improve the accuracy of energy breakdown, such unknown consumption has to be carefully handled.

4 Methodology

We study the problem of disaggregating the aggregate energy in a single home to its constituent appliances at hourly intervals. Based on previous discussions, the hourly sampled energy 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 handle 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 energy use; ii) the aggregate energy is a composition of various and complicated appliance energy, 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 consumption. We now discuss each component of our proposed TreeCNN model.

• **Convolutional Neural Network (CNN).** Our analysis results show that different appliances have distinct temporal patterns. But, simply modeling the hourly time-series data as a one-dimension sequence cannot fully describe 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 data should be viewed as a high dimensional compound of various temporal patterns.

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 [15], we appeal to CNN models to extract energy usage basis. The key component of CNN model is the filters that

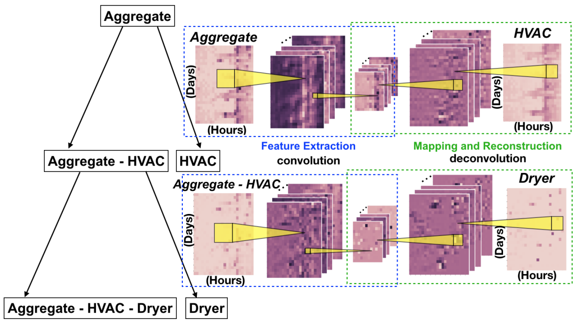


Figure 2: Our tree-structured iterative energy breakdown approach shown for two appliances (HVAC and Dryer).

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.

With CNN model, 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 filters, the aggregate readings can be projected into its corresponding appliance usage. Figure 2 shows an example of a CNN model which learns the mapping from aggregate readings to HVAC consumption. In the convolution phase, CNN model takes the aggregate as input and tries to reduce it to a much denser representation. Due to the sparsity and granularity, the temporal patterns can be extracted and the input will be represented as a denser matrix with a lower dimensionality. In the deconvolution phase, the decoder performs the opposite operations that reverse the action of encoders.

• **Tree Structure.** The model complexity in energy breakdown increases exponentially with the number of sources constituting the aggregate. Further, the usage of some appliances can get overshadowed by others (Figure 1e) creating a "magnitude" problem.

Different from conventional techniques, which either estimate appliance usage independently, or disaggregate the energy altogether at once, we propose a tree-structured model to extract appliance patterns in a "stage-wise manner". With the tree structure, our approach performs 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 model with HVAC and dryer. The root node takes the aggregate readings as input and reconstructs the HVAC consumption as its output. The difference between them will be passed to the child node as refined input for the next appliance, e.g., dryer in the figure. The magnitude problem is thus eased for the minor appliances, such as microwave, if we place them at the lower end of the tree. In contrast, if we jointly decompose the aggregate readings into appliances' readings, the minor appliances will be overshadowed by major appliances, and mostly given zero predictions, which defeats the original intention of energy breakdown.

In our TreeCNN model, we effectively simplify the energy breakdown iteratively. In each node, the CNN model performs an end-to-end learning 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 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, no existing work models the unknown consumption. In our tree-structured model, the unknown consumption can be viewed as a special appliance which consists of multi-dimension temporal patterns. Modeling the unknown consumption makes it possible to remove such energy from the true aggregate consumption, which leads to a more accurate estimation of the observed appliances.

4.2 Tree Order

Given N appliances, we would have $N!$ possible tree structures. For a residential home at the U.S., one can usually expect 7-10 major appliances in monitoring. For any larger values of N, exhaustively finding the "optimal" tree order can be computationally expensive, where "optimality" is defined as per given energy breakdown metric M. Since the error of one decomposition will be propagated through the tree structure, the tree order is essential to our model.

We propose a greedy algorithm to find a suitable tree order to mitigate error and reduce the search space. The key operation is to estimate the overall performance with partial information via the inverse propensity weighting scheme [16, 22]. Assume we have N appliances and metric $M(E(a_i), \hat{E}(a_i))$ is used to compute the error for appliance a_i where $\hat{E}(a_i)$ and $E(a_i)$ denote the estimation and the ground-truth of appliance a_i. In the first iteration, we will create N candidate splits. Among these N models, we select k models with the smallest *estimated energy breakdown error* ($EEBE^{GR}$) on the validation set. Next, $(N-1)$ sub-trees are created for each selected k parent tree. The selection repeats until we have constructed the whole tree. In particular, local metric $EEBE^{GR}$ is used to estimate the overall energy breakdown error, before the whole tree has been constructed. This metric calculates the ratio between a chosen metric 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 to aggregate, then the aggregate would have an expected error of $\frac{e}{x}$ from this appliance. Thus, we can estimate the final error by estimating the prediction error of each appliance during the tree construction. 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 Evaluation

In this section, we evaluate the our model on the hourly data collected from 68 homes over 112 days in the Dataport dataset.

5.1 Experimental settings

5.1.1 Baselines. We first describe the baselines

• **Mean Energy:** This baseline computes the predicted energy of an appliance as its mean energy in the training set.

• **Factorial Hidden Markov Model (FHMM):** FHMMs [14] 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. They proposed a modified CP (MCP) to mitigate the scaling problem.
• **Sparse Coding:** Sparse coding [13] model approximates the bases and activations for each appliance with sparsity constraints. The authors also proposed a structured prediction based method called discriminative sparse coding (DSC).
• **Recurrent Neural Networks (RNN):** We performed the decomposition with individual RNN model and TreeRNN model, which captures the time-series dependency of the energy signals.
• **Convolutional Neural Networks (CNN):** We use individual CNNs and JointCNN. Individual CNNs estimates appliances' energy separately, and JointCNN estimates them all together at once.

5.1.2 Approach settings. Among all methods, we used 5-fold cross-validation in the experiments. The final 20% of the train set is set for validation purpose. For each algorithm, the optimal parameters are learned via 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 [5, 23]. 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 rank of latent factors from 1 to 50.

We implemented all neural network models with PyTorch [19]. For RNN models, we have the following parameters: cell type: {GRU, LSTM, RNN}; number of hidden units: {20, 50}; number of layers: {1, 2, 3}; number of iterations: {1000, 2000, 3000}. For CNN models, complex network will easily cause overfitting due to the limited training data. Thus, we have the encoders consist of two convolutional layers and two deconvolutional layers with normalization [10] to accelerate the training process, and ReLU activation function to introduce non-linearity. We choose the learning rate from {0.01, 0.1, 1} and the number of iterations from {1000, 2000, 3000}. For tree-structured models, we perform both exhaustive and greedy search. We use top-$k = 3$ results at each stage of greedy search. We use the L1-loss as the objective function. For neural network based methods, we clamp the estimated consumption to a maximum of the observed aggregate energy. Our entire codebase, baselines, analysis and experiments can be found on Github [1].

5.1.3 Metric. Based on prior literature [4, 7], we evaluate the performance with mean absolute error (MAE). Denote the ground-truth and estimation for home h, appliance a, day d and hour t as $E(h, a, d, t)$ and $\hat{E}(h, a, d, t)$, for appliance a, MAE is computed as:

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

where H, D, T indicate the number of homes and days, and hours in a day. We use the average MAE across appliances to measure the model accuracy. Lower mean MAE indicates better performance.

As shown before, in some ON/OFF appliances, the active time is generally low. The MAE alone cannot fully reflect the performance, as zero predictions can also give a good MAE. Thus, we separate MAE into two parts, corresponding to the active and inactive states based on the ground-truth (threshold is reported in Table 1).

[1] https://github.com/yilingjia/TreeCNN-for-Energy-Breakdown.git

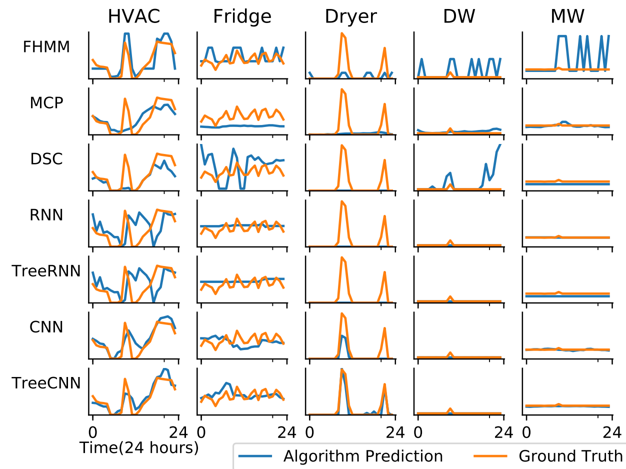
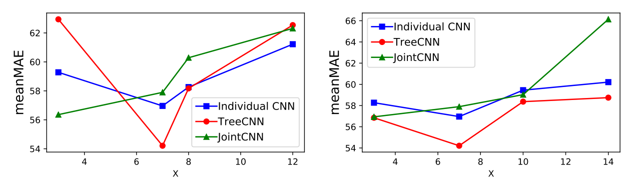


Figure 3: Comparison of baseline algorithms for a sample day. (DW: Dishwahser, MW: Microwave)

5.2 Experiment Results

In the following sections, we first test the energy breakdown capabilities in an ideal case, where the aggregate energy equals 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 study the effectiveness of unknown consumption modeling. Last, we report the results of greedy algorithm on tree order estimation.



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

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

5.2.1 Filter size in CNN models. In CNN-based models, filter size plays an important role in capturing the temporal patterns. We explored the effect of different filter sizes in the first layer of CNN models. From Figure 4, we can observe that the performance of CNN models is quite sensitive with the filter sizes. When the size equals to 7×7, most models achieve the best performance, as such filters can well capture the patterns across hours and days. With small sized filters, the model might miss some periodical patterns across multiple dimensions; and with large sized filters, it will fail to capture the local features and generate redundant information. In the following experiments, the filters of each layer are set to 7×7 and 2×2. And the decoder is a mirrored version of the encoders with two deconvolutional layers.

5.2.2 Ideal case. Like previous studies [13, 23], we simulate the ideal case by manually setting the artificial aggregate. Shown in Table 2, TreeCNN algorithm outperforms all the baselines (p-value is calculated between the predictions of TreeCNN and the second best model). Figure 3 shows the energy estimation from a set of baselines of a randomly chosen day of one randomly selected home.

Table 2: MAE on artificial aggregate and true aggregate dataset.

		FHMM	CP	MCP	SC	DSC	Mean	RNN	TreeRNN	CNN	Joint-CNN	TreeCNN
Artificial Aggregate	MAE	114.99	106.19	103.46	92.20	172.95	100.34	67.34	64.26	56.96	57.90	**51.64***
	MAE on Active	360.76	390.39	390.90	411.93	388.29	404.03	388.90	383.84	310.51	332.45	**261.30***
	MAE on Inactive	80.34	96.11	96.93	28.61	**15.03**	49.13	37.87	36.79	41.53	42.07	40.05
True Aggregate	MAE	134.99	123.90	125.30	245.70	218.43	126.86	97.8	94.52	89.15	91.35	**86.94***
	MAE on Active	400.76	414.01	412.33	515.54	521.35	421.50	43538	433.01	417.85	429.87	**391.50***
	MAE on Inactive	82.94	103.10	107.81	215.57	190.84	102.22	60.32	**54.85**	68.28	69.26	61.69

$* \, p\text{-value} < 0.05$

Table 3: Effect of UC modelling. (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**

FHMM assumes the appliances can be modeled with a Gaussian HMM using discrete states, which considers the transition and emission probabilities between states. But it is not well suited to sparsely used appliances. MCP and DSC algorithms are both of a similar vein focusing on learning various "basis" of energy consumption. Both algorithms performed reasonably well in general. However, they are not well-tuned for the instances when energy consumption patterns differ from the average patterns. And 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 simple, performs reasonably well. However, 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.

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 a one-dimension time-series ignoring the periodical 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. 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.

5.2.3 Real-world case. Now, we evaluate the models on the true aggregate dataset. From Table 2, we can see that with true aggregate, TreeCNN model still outperforms the other baselines. Comparing the results from these two settings, we can 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. In our TreeCNN, the complex latent bases of unknown consumption can be captured via filters. Table 3 shows the improvement when we consider the unknown consumption in the model. Paired t-test shows almost all the estimations significantly improve except for the dryer. In this work, we only set one model for the unknown consumption, while it might come from a combination of various sources, which we defer to future work.

Table 4: 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

5.2.4 TreeCNN with greedy v/s exhaustive tree orders. Table 4 compares the mean MAE performance of TreeCNN model with the best and worst tree orders found by exhaustive search and the order found with greedy search. We also report average MAE over all tree orders explored in the exhaustive search. It shows that our greedy algorithm performs substantially better than the average and is only about 4% worse than the best order found via exhaustive enumeration. The learned tree order also discloses interesting property of this problem: placing HVAC in the first few levels in the tree leads to poorer performance, as the HVAC energy is easily to be over-estimated to "eat" up the energy of 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. 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.

We would like to explore a few future extensions. 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 [18] 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.

7 Acknowledgements

This work is based upon work supported by the National Science Foundation under grant CNS-1646501, IIS-1553568, IIS-1718216, and NVIDIA GPU grant.

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

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

[11] 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.

[12] 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).

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

[14] 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.

[15] 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.

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

[17] 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.

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

[19] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. 2017. Pytorch.

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

[21] Staff Report. 2017. 2017 Assessment of Demand Response and Advanced Metering.

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

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