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:

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

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 mealtimes, 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 modeled 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>
<thead>
<tr>
<th>HVAC</th>
<th>Fridge</th>
<th>Dryer</th>
<th>Dishwasher</th>
<th>Microwave</th>
</tr>
</thead>
<tbody>
<tr>
<td>α(min)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>a_δ</td>
<td>230</td>
<td>20</td>
<td>250</td>
<td>55</td>
</tr>
<tr>
<td>Active</td>
<td>73.9%</td>
<td>97.8%</td>
<td>4.9%</td>
<td>4.1%</td>
</tr>
<tr>
<td>Max</td>
<td>5099.7</td>
<td>428.6</td>
<td>4364.1</td>
<td>1021.7</td>
</tr>
<tr>
<td>Mean</td>
<td>1162.7</td>
<td>88.6</td>
<td>1303.6</td>
<td>369.5</td>
</tr>
<tr>
<td>Std</td>
<td>800.2</td>
<td>40.2</td>
<td>756.2</td>
<td>206.5</td>
</tr>
</tbody>
</table>

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:

\[
\delta_a = \frac{a_{\alpha}}{60} \times \frac{1}{H} \sum_{h \in H} \max_{d,t \in (H)} E_{h,a,d,t}
\]

where H represents the number of homes, \(E_{h,a,d,t}\) is the energy consumed by appliance a on day d at hour t for home h, and \(a_{\alpha}\) 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.
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 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 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 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 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.
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 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 get the error of the entire tree before it is constructed, which makes the sequential decisions of tree order possible. This is essential to our model. The rationale behind \( \text{EEBE}^{GR} \) 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 \( \frac{e x}{N} \) 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.

5.1 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 \( \hat{E}(h, a, d, t) \) and \( \hat{E}(h, a, d, t) \), for appliance a, MAE is computed as:

\[
\text{MAE}(a) = \frac{\sum_{h} \sum_{d} \sum_{t} |E(h, a, d, t) - \hat{E}(h, a, d, t)|}{H \times D \times T}
\]  
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).

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.

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 \( 7 \times 7 \), 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 \( 7 \times 7 \) and \( 2 \times 2 \). 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.

---

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

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

Figure 4: Effect of filter size tuning on CNN models.
Table 4: TreeCNN performance under different tree orders. (UC: Unknown Consumption, agg: aggregate)

<table>
<thead>
<tr>
<th></th>
<th>Worst</th>
<th>Average</th>
<th>Greedy</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial agg</td>
<td>84.42</td>
<td>86.72</td>
<td>54.21</td>
<td>51.64</td>
</tr>
<tr>
<td>True agg w.o. UC</td>
<td>105.60</td>
<td>96.54</td>
<td>88.38</td>
<td>86.94</td>
</tr>
<tr>
<td>True agg w. UC</td>
<td>110.75</td>
<td>98.64</td>
<td>87.21</td>
<td>84.55</td>
</tr>
</tbody>
</table>

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


