An Open Problem: Energy Data Super-Resolution

Rithwik Kukunuri
kukunuri.sai@alumni.iitgn.ac.in
IIT Gandhinagar, India

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

Hongning Wang
hw5x@virginia.edu
University of Virginia, USA

ABSTRACT

In this notes paper, we present an open problem to the Buildsys community: energy data super-resolution, referring to the task of estimating the power consumption of a home at a higher resolution given the low-resolution power consumption. Super-resolution is especially useful when the smart meters collect data at a very low-sampling rate owing to a plethora of issues such as bandwidth, pricing, old hardware, among others. The problem is motivated by the success of image super resolution in the computer vision community. In this paper, we formally introduce the problem and present baseline methods and the algorithms we used to "solve" this problem. We evaluate the performance of the algorithms on a real-world dataset and discuss the results. We also discuss what makes this problem hard and why a trivial baseline is hard to beat.

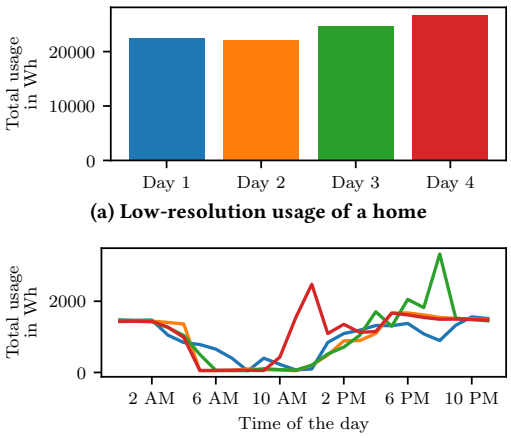
CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**.

KEYWORDS

super-resolution; smart meters; energy analytics

ACM Reference Format:
Rithwik Kukunuri, Nipun Batra, and Hongning Wang. 2020. An Open Problem: Energy Data Super-Resolution. In *The 5th International Workshop on Non-Intrusive Load Monitoring (NILM'20), November 18, 2020, Virtual Event, Japan.* , 4 pages. https://doi.org/10.1145/3427771.3429995



(a) Low-resolution usage of a home

(b) High-resolution usage of a home

Figure 1: Illustration of energy super resolution

1 INTRODUCTION

Energy data super-resolution refers to the task of estimating high-resolution power consumption of a home given the low-resolution power consumption of a home. Our inspiration comes from the success of the super-resolution of images [2]. Super-resolution is especially useful when the smart meters collect data at a very low-sampling rate owing to a plethora of issues such as bandwidth, pricing, old hardware, among others. In this paper, we specifically focus on the task of 24x super-resolution: We try to estimate the hourly power consumption of a home, given the daily power consumption of a home. Figure 1 shows the concept of energy super-resolution.

We believe that energy super-resolution has several similarities to the image super-resolution tasks. We expect nearby pixels in an image to have similar values. Similarly, we expect adjacent hours and days to have similar energy consumption. Further, we expect periodicity in the energy data: weekdays will likely have similar energy consumption.

Against this background, this paper introduces the problem of energy data super-resolution and discusses its various aspects. Our key contributions to this paper are i) introducing this problem to the community, ii) strong benchmarks, iii) exploring and understanding the reasons about the hard nature of this problem.

Our benchmark algorithms are based on the intuition that homes that are "similar" in features derived from low-resolution energy usage will help us identify "similar" homes in high-resolution energy usage. The notion of "similarity" is defined differently in various proposed algorithms.

Our evaluation on 68 homes from a publicly available data set suggests that the super-resolution performance is not better than a simple mean model (prediction is based on mean usage of train homes). We believe there are several factors explaining this bottleneck: i) high intra-home energy variance; ii) the intuition of "similar" homes fails due to the absence of high-fidelity features explaining deviation from a "routine" energy consumption pattern.

Through this note, we plan to engage with the Buildsys community to take the problem ahead.

2 METHODOLOGY

In this section, we first describe the mathematical formulation and then define the baseline and our proposed methods.

2.1 Mathematical Notation

We define the Proportion tensor (P) as the proportion of electricity consumed in the t^{th} hour on d^{th} day for h^{th} home.

NILM'20, November 18, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8191-8/20/11…$15.00
https://doi.org/10.1145/3427771.3429995

Table 1: Notation used throughout the paper

Term	Definition
H	No.of homes
D	No. of days
$X \in R^{H \times D}$	Low resolution matrix (Daily usage)
$Y \in R^{H \times D \times 24}$	High resolution tensor (Hourly usage)
$P \in R^{H \times D \times 24}$	Proportion tensor
$\hat{Y} \in R^{H \times D \times 24}$	Predicted super-resolution usage
$\hat{P} \in R^{H \times D \times 24}$	Predicted proportion tensor

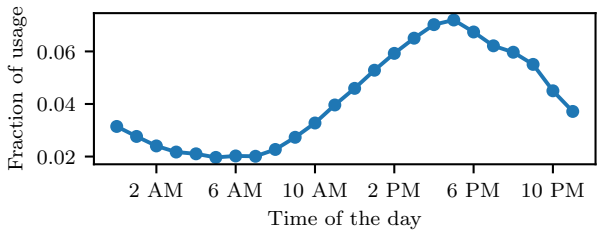


Figure 2: Dominant power consumption pattern in the training dataset

$$P[h,d,t] = \frac{\text{Usage at } t^{th} \text{ hour for home h on day d}}{\text{Total usage on day d for home h}} = \frac{Y[h,d,t]}{X[h,d]}$$

Table 1 denotes the notation used in the paper.

2.2 Baseline: Mean proportion algorithm

The key assumption in this method is that every home follows the same hourly energy consumption pattern and the homes differ only in the scaling factor. In this algorithm, we first find the most dominant power consumption pattern in the training data set. This dominant pattern ($W \in R^{24}$) can be found out by computing the mean of tensor P across homes and days as:

$$W[t] = \sum_{h=1}^{H} \sum_{d=1}^{D} \frac{Y[h,d,t]}{24 \times D \times H}$$

Figure 2 shows the most common power consumption pattern in the dataset. The power consumption peaks around 5 PM and is the minimum in the early morning hours. The hourly power consumption for a testing home can be estimated using: $\hat{Y}[h,d,t] = X[h,d] * W[t]$

2.3 Proposed Methods

In this section, we discuss our proposed approaches. Many of these approaches have been inspired by recent literature in image super resolution. The **key idea** in all these approaches is to find or curate features from low-resolution input that can accurately estimate the high-resolution output. Our methods are based on the premise that "similarity" in the input features is indicative of "similarity" in the output space. These methods will differ mostly in the design of the features and the definition of similarity.

In most of these methods, we will be dealing with the proportion of usage across the 24 hours to avoid scale issues (i.e. two homes with similar patterns in hourly and aggregate usage should ideally be considered similar though on a different scale).

2.3.1 K-Nearest Neighbors Regressor. In this method, given a testing home, we find the most similar homes in the training space and use them to estimate super-resolution usage. The similarity is decided trivially based on the feature vector of daily energy usage across D days.

First, for each training home (h), we compute a 24-dimensional weights vector (W_h) to describe the home. We can compute W_h for a home (h) by taking the mean across the days for the proportion usage vector. This vector W_h denotes the dominant pattern followed by the home h.

$$W_h = \frac{1}{D} \sum_{d=1}^{D} P[h,d] \tag{1}$$

We train a K-Nearest Neighbors regressor for each (x_h, W_h) in the training dataset. Given a testing home, we find the K-Nearest Neighbors and use the average of their weights to estimate the usage of the testing home. The same set of weights is used across all the days. Let the weights vector after averaging be $\hat{W}_h$. We can then element-wise multiply the aggregate daily energy of a home h for the d^{th} day with $\hat{W}_h$ to obtain the predicted hourly usage as $\hat{y}[h,d,t] = X[h,d] \times \hat{W}_h[t]$

2.3.2 Non-Negative Matrix Factorization (NNMF). The key intuition of this method is that the energy consumption of a home is largely spread across a small number of factors, such as: number of occupants, square footage, among others. This method tries to "learn" such important home energy consumption features and then use K-nearest neighbours in this latent space like earlier to estimate the hourly usage of a test home.

We first decompose the low-resolution matrix into home feature matrix ($U \in R^{H \times r}$) and time feature matrix ($V \in R^{r \times D}$) and r denotes the chosen rank, using $X \approx UV, s.t. U, V >= 0$. We next train the K-nearest neighbors model on (U_h, W_h), where W_h denotes the weights vector that denotes the pattern followed by home h.

Given a testing home h, we compute the coefficients vector U_h, and then we find K-Nearest Neighbors using U_h and take the average of the weights ($\hat{W}_h$). The super-resolution usage can be estimated in similar way as the earlier KNN method.

2.3.3 Tensor Decomposition. Prior literature [1] has used tensor decomposition for the task of energy disaggregation. We extend this algorithm for the task of super-resolution. Unlike our NNMF approach, where we first decomposed low-resolution data and then fed the features into KNN, in this approach we directly "complete" the tensor to obtain estimated hourly usage.

We create a three-dimensional tensor $T \in R^{H \times D \times 25}$. This tensor contains both the training homes and the testing homes. In the temporal (last dimension of size 25) of tensor T, the first channel contains the low-resolution usage for both the training homes and the testing homes. The super-resolution usage of the training homes for each hour is in the remaining 24 channels. For the testing homes, we do not have access to their super-resolution usage, and we wish to estimate it. Hence, we the last 24 channels are considered missing in the temporal dimension. We now estimate the tensor T by computing the Low-Rank Parafac Decomposition into r components using $\hat{T} \approx (H \times r) \otimes (D \times r) \otimes (25 \times r)$

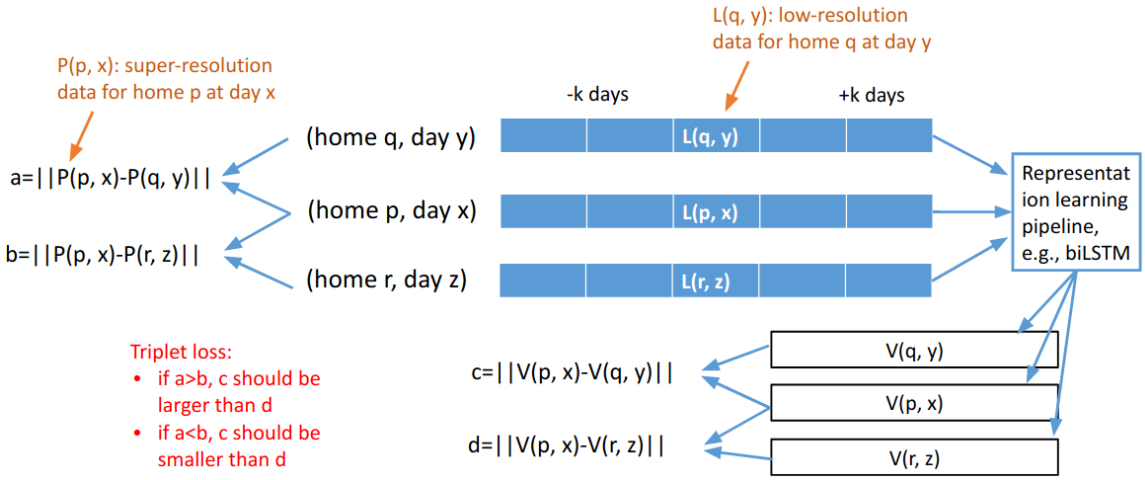


Figure 3: An overview of triplet loss for energy super-resolution

After the loss is converged, we now have $\hat{T}$, which now contains the estimated super-resolution usage of the testing homes.

2.3.4 Triplet learning. Triplet learning became popular after its successful application in FaceNet [4]. The key idea is to project input (images in FaceNet) into an embedding space, where similar input have similar embeddings, and dissimilar input have dissimilar embeddings. These embeddings are later used for classification and clustering. The neural network used for the projection into the embedding space is learned using triplet learning. Let M be a neural network, which maps the input to an embedding space. Let (x_i, y_i) and (x_j, y_j) be the (input features, output) for two examples i and j. In triplet learning, we can learn a neural network M such that: if y_i and y_j are similar, then $M(x_i)$ and $M(x_j)$ are similar; and if y_i and y_j are not similar, then $M(x_i)$ and $M(x_j)$ are not similar.

We now discuss the algorithm for using triplet learning for energy super-resolution. Let $L(h, d)$ denote the feature vector for home h on day d. The feature vector consists of the usage of the past k days, the usage on d^{th}, and the usage of the next k days. It is a feature vector of length 2k+1, given as: $L(h, d) = [x_{h,d-k}, \ldots, x_{h,d-1}, x_{h,d}, x_{h,d+1}, \ldots, x_{h,d+k}] = x_{h,d-k:d+k}$

If any of the future reading or the past reading is not available, then it is replaced with zero. This way, we can create the feature vector for the samples that do not have the readings for the future days or past days.

The embedding vector of a sample home h and day d, can be computed by doing a forward pass of the neural network on the feature vector $L(h, d)$. Let $V(h, d)$ denote the embedding vector generated by $L(h, d)$. Let us now assume we have three examples (p,x), (q,y) and (r,z), where p,q,r denote the homes and x,y,z denote the days. As per our definition of triplet learning, we would like, if example 1 and 2 are closer compared to example 1 and 3 in the output space, then, the same order should be preserved in the embedding space. For doing these calculations, we compute four quantities (a, b, c, d) and apply loss function as shown in Table 2.

$$a = \left\|P(p,x) - P(q,y)\right\|^2; b = \left\|P(p,x) - P(r,z)\right\|^2$$

$$c = \left\|V(p,x) - V(q,y)\right\|^2; d = \left\|V(p,x) - V(q,y)\right\|^2$$

Table 2: Loss functions for triplet learning

S. No	Condition 1	Condition 2	Loss
i	a > b	c > d + margin	0
ii	a > b	c < d + margin	$(d + margin - c)^2$
iii	a < b	c + margin < d	0
iv	a < b	c + margin > d	$(c + margin - d)^2$

We use a margin in the loss function, to ensure that the separation in the embedding spaces is significant. This way, dissimilar examples stay farther from each other. We sample multiple triplets spanning across multiple homes and multiple days. These samples are used to train the neural network, which maps the features to an embedding space. Finally, given a testing sample, we project the sample into the embedding space and find similar neighbors in the embedding space and use the average of their proportions to compute the super-resolution usage. Figure 3 summarizes our triplet learning algorithm. Figure 4 shows the architecture used for the embedding network. This network is inspired from the state-of-the-art NILM technique called Seq2Point [5].

3 EVALUATION AND DISCUSSION

We evaluate the performance of the algorithms on data from 68 homes over 112 days from the Dataport dataset [3]. We used 3-Fold cross-validation and reported the mean error observed across all the folds. For the neighbourhood based methods, we varied the K from 1 to the number of samples in the training dataset. For the rank based models we varied the rank from 1 to 80. In the triplet learning method, we chose the margin to be 10, so that there is significant separation between different samples. The hyperparameters for all the methods were optimized using nested cross-validation. The code can be found here. [1].

From Table 3, we can observe that all the models have a similar performance. We can say that the Mean Proportions model is the "best" model since it has low model complexity and yet is comparable to the best performing method. The optimal hyperparameters for the number of neighbors for the K-Nearest Neighbors,

[1] https://github.com/Rithwikksvr/Buildsys-2020-SR

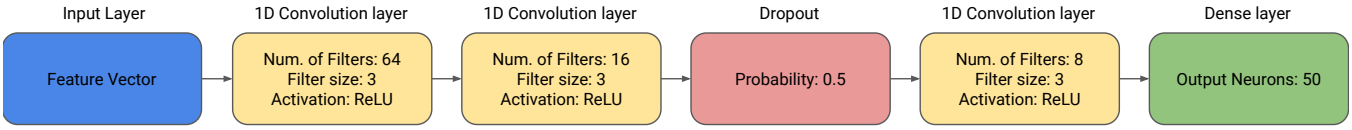


Figure 4: Embedding network architecture

Table 3: Energy super-resolution performance of the trivial baseline (mean proportion) is comparable or better than proposed sophisticated methods

Model	MAE (Lower is better)
Mean proportions model	441
K-Nearest Neighbors	441
NNMF + K-Nearest Neighbors	450
Tensor Decomposition	439
Triplet learning	443

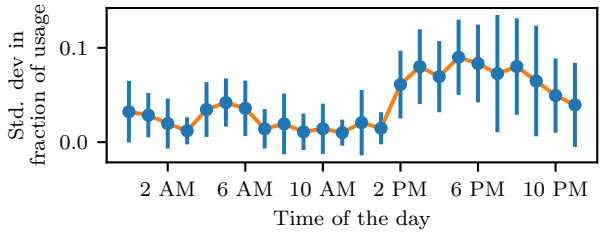


Figure 5: Mean ± Std. dev across a single type of day for a single home showing high intra-home energy variation making energy super-resolution hard.

NNMF + K-Nearest Neighbors, and Triplet learning methods is high. These models were not able to accurately estimate the weights for each home; hence in order to minimize the loss on the validation set, they are choosing high number of neighbors, which was the same as Mean Proportions model. We now discuss two factors to explain average super-resolution estimation. **High-Intra home variance:** We observed that there is significant difference in the hourly energy consumption of a home across days. Figure 5 shows the Mean ± Standard deviation for the hourly proportion of usage for a single home across a single day of the week (Mondays across different weeks). These differences may be attributed to occupant behaviour among other factors, none of which are available in our dataset. Without these features (which may be highly occupancy or behavior driven) the super-resolution performance is bottle-necked to the mean proportion usage, since the intra-homes variations are unlikely to be present across "neighbours". Thus, solving the energy super-resolution problem is a challenging problem given the lack of "distinguishing signals".

Existence of Multiple Solutions: We observed that that two homes with similar daily usage vectors might have completely different hourly usage vectors and vice-versa, rendering our "neighborhood-based" methods ineffective. Figure 6 shows the dominant usage pattern of two homes which have the most similar low-resolution vectors. Due to this, similar low-resolution vectors predict the same

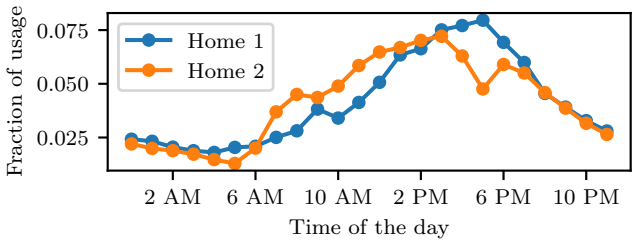


Figure 6: Super-resolution usage of two homes with very similar low-resolution usage can be different making energy super-resolution hard.

super-resolution usage, which is not the actual case. We need more features for accurately determining the type of each home.

4 CONCLUSIONS

In this work we introduced the problem of energy super resolution. We wish to engage the community in a dialogue on the various facets of this problem: i) what applications can we envision?; ii) as an adversary or privacy conscious person, what should one do to prevent accurate super-resolution?; iii) how do we solve the information bottle-neck realistically?; iv) are there any information-theoretic guidelines to approach this problem?; v) how can data compression algorithms be leveraged for this problem especially when the data bandwidth remains a concern; vi) Recent computer vision literature has also studied the problem of colorization. An analogue to the same in our domain could be to perform energy disaggregation. Finally, we could combine super-resolution with colorization, aka, disaggregate to high-resolution appliance components using low-frequency energy components.

REFERENCES

[1] Nipun Batra, Yiling Jia, Hongning Wang, and Kamin Whitehouse. 2018. Transferring decomposed tensors for scalable energy breakdown across regions. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*. Springer, 184–199.
[3] 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 *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 210–214.
[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.
[5] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. 2018. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Thirty-second AAAI conference on artificial intelligence*.