

INTRODUCTION

Abstract

Energy disaggregation is the procedure that infers the energy consumption in the basis of appliances in a household given the total energy consumption from a single meter of that household. In recent years, this field has become increasingly popular as smart meters have begun to deploy and are installed in many households across the world. However, the field's popularity is mostly attributed to the extremely powerful techniques which are able to take advantage of the continuously improving computational resources and perform this disaggregation in a non-intrusive manner. The term non-intrusive implies that the appliance-based energy consumption is not determined by installing individual meters on each appliance and thus interfering with the occupant's privacy, but rather by estimating it using both deterministic and stochastic techniques.

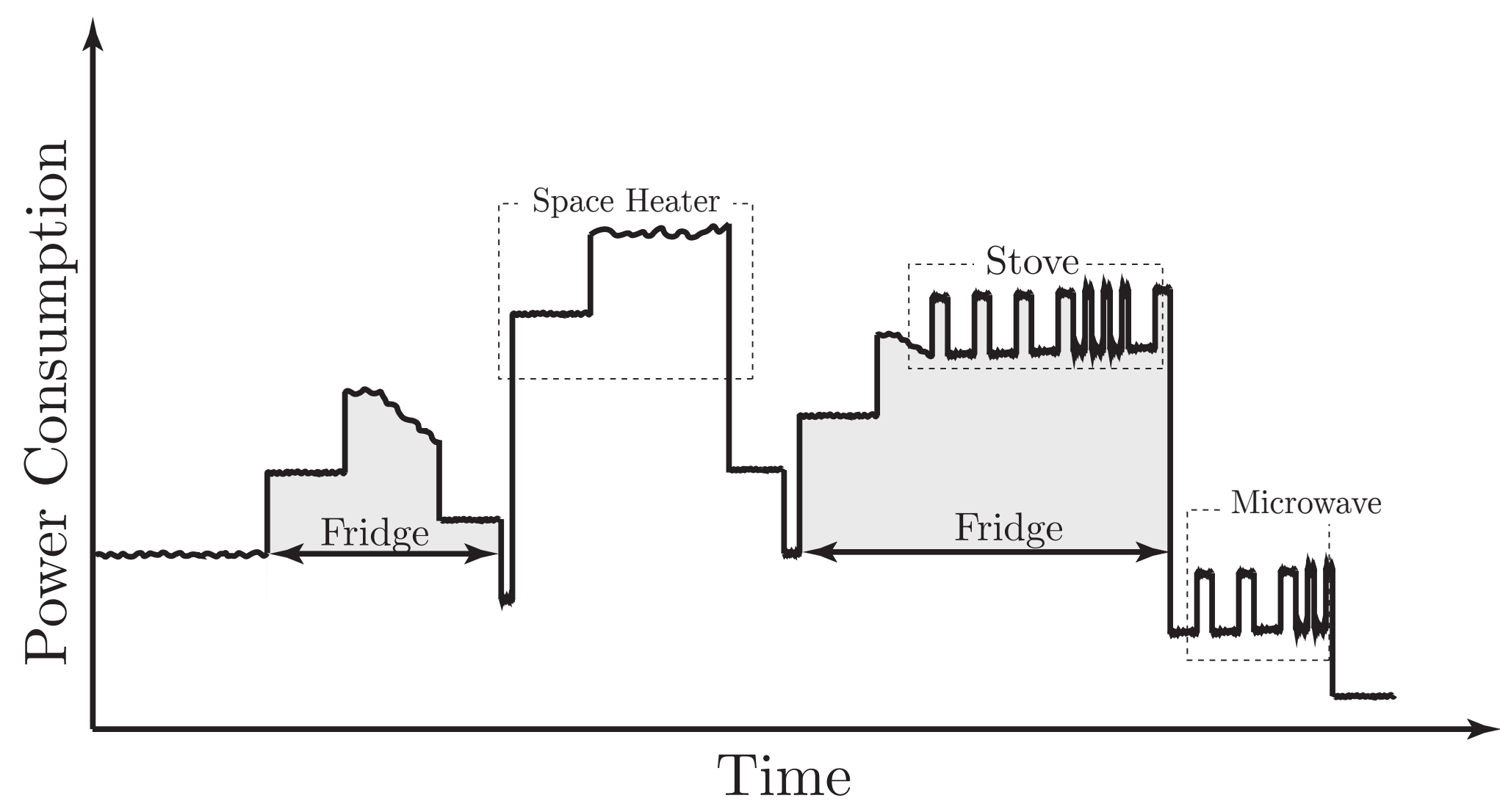


Figure: Typical Power Consumption in a household within a time frame that many appliances activate and deactivate

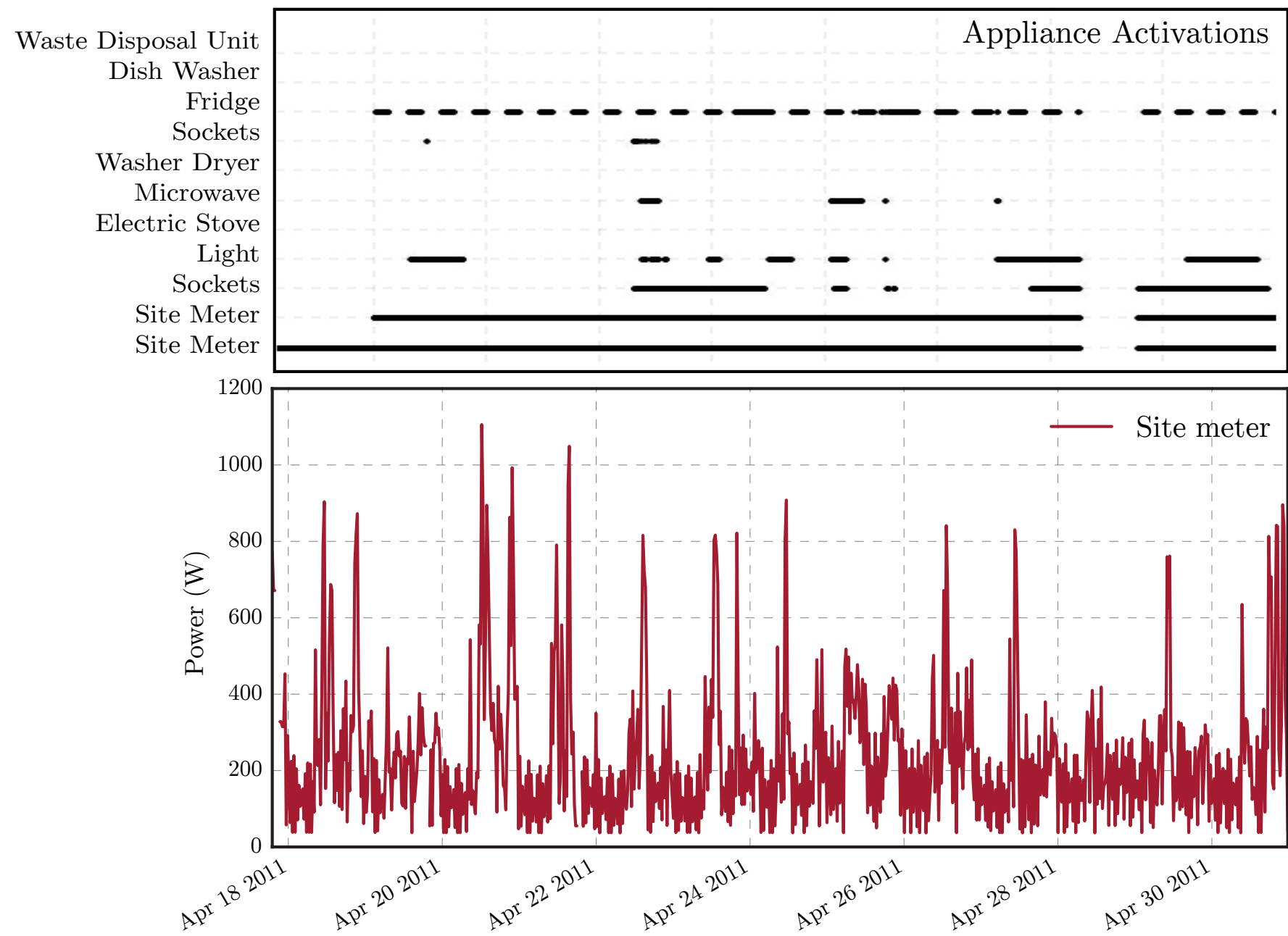


Figure: The energy consumption of a household (Building 2 in the REDD dataset) during a period of 1 month

THE REDD DATA SET

In the context of this project we make use of the REDD data set [4], a public data set for energy disaggregation research. The REDD dataset is not readily accessible since access can only be granted by the authors of [4] at MIT. While REDD, is what made this project possible, it cannot be directly fed into the disaggregation algorithms developed/borrowed in this project. A data Pipeline is necessary to preprocess and feed the data for both training as well as testing purposes. While the training and disaggregation steps are significantly different in all of the methods that we use, the remaining steps are the same and are outlined below.

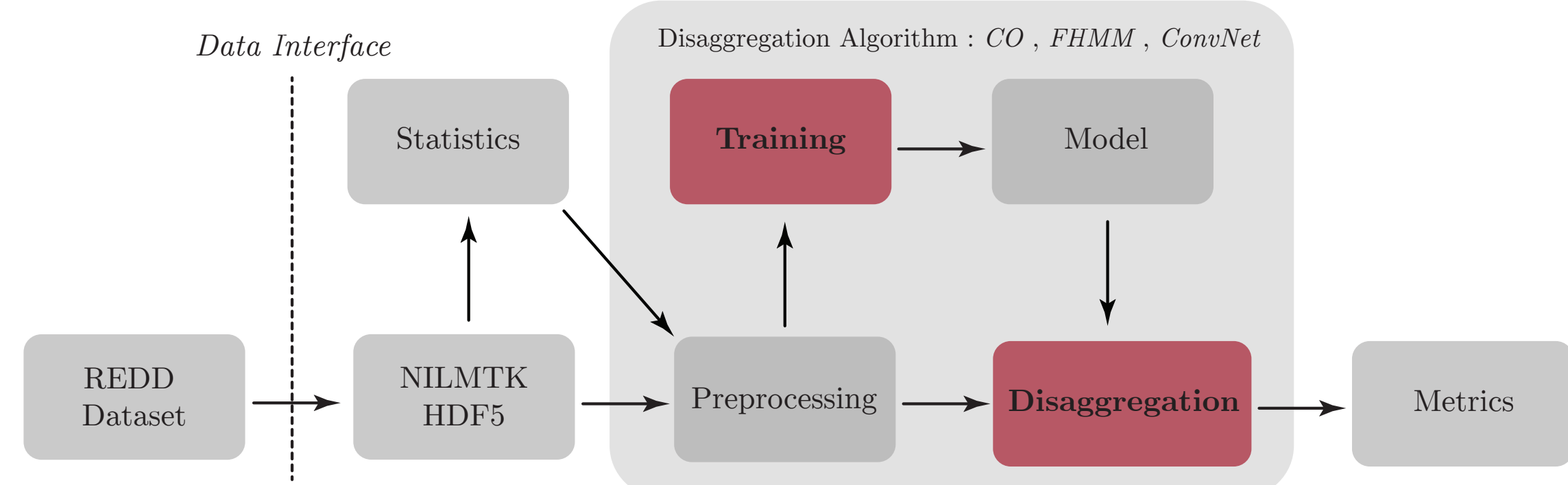


Figure: The NILM pipeline At each stage of the pipeline, results and data can be stored to or loaded from disk

METHODS FOR DISAGGREGATION (1)

Combinatorial Optimization (CO)

CO is a deterministic method that finds the optimal combination of appliance states which minimizes the difference between the sum of the predicted appliance power and the observed aggregate power subject to a set of appliance models [1]. The optimal combination of appliance states $\hat{x}_t^{(n)}$ of appliance n at time instance t is given by [5]

$$\hat{x}_t^{(n)} = \underset{\hat{x}_t^{(n)}}{\operatorname{argmin}} \left| \bar{y}_t - \sum_{n=1}^N \hat{y}_t^{(n)} \right| \quad (1)$$

where \bar{y}_t is the aggregate power consumption at t and $\hat{y}_t^{(n)}$ is the power consumption of appliance n in the current state, which is determined by training the CO.

Pros	Cons
Easy to Implement	Computationally Inefficient
Mathematically Attractive	Limited disaggregation accuracy

The Computational Cost is $O(TK^N)$ where N is the number of appliances, K is the number of appliances states and T is the number of time slices in the dataset. CO is only suitable for small problems.

Factorial Hidden Markov Model (FHMM)

The power demand of every appliance can also be modelled after hidden Markov Model (HMM) [1] with the hidden states being the states of each appliance. To effectively disaggregate the energy it is necessary to simultaneously decode the power draw of n appliances and thus a Factorial Hidden Markov Model is necessary. Such a model requires 3 parameters:

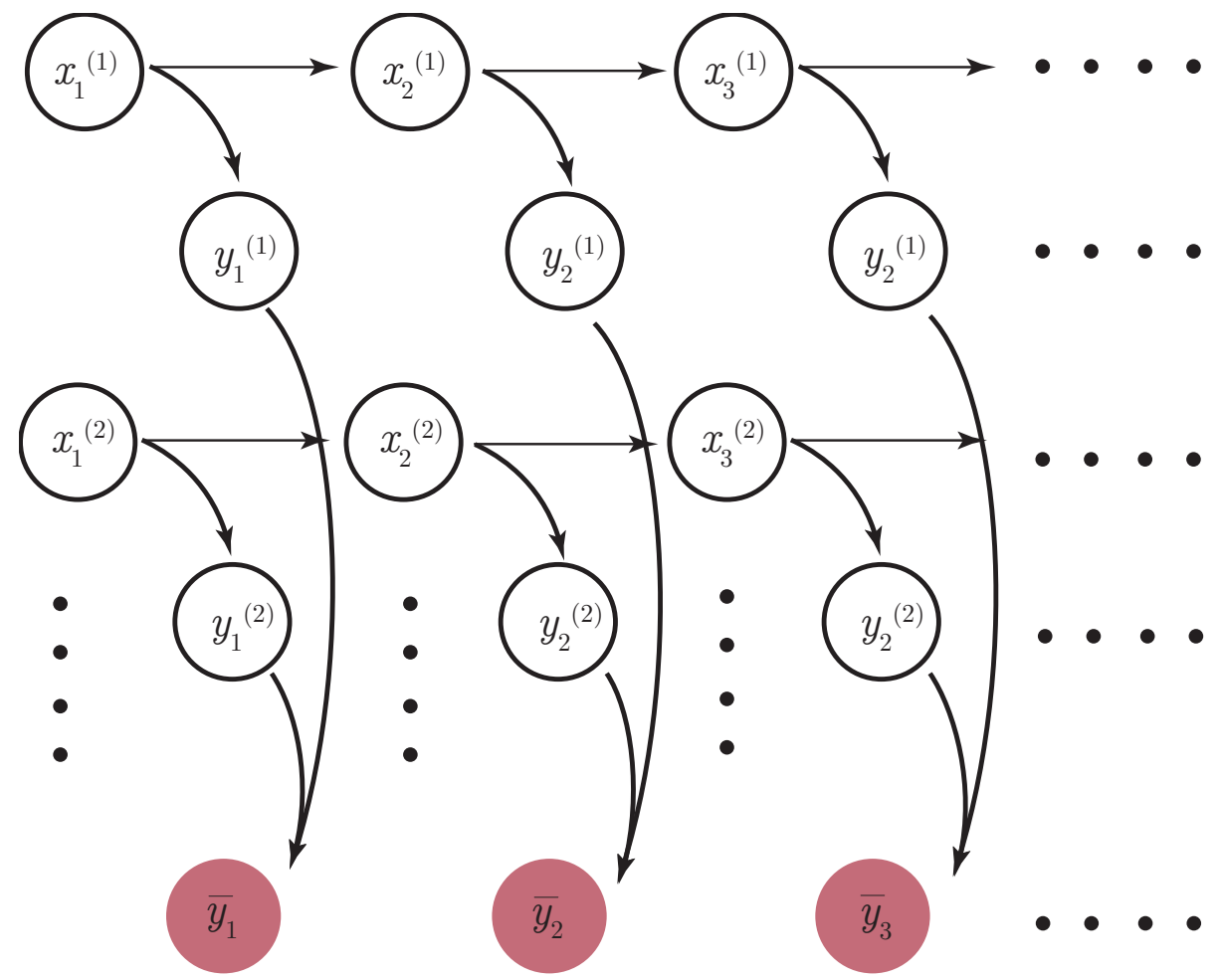


Figure: A typical diagram of the Factorial Hidden Markov Model considered here [4]

- A prior probability π_0 with K^N entries, typically drawn from a multinomial distribution.
- The transition probability matrix A with dimensions $K^N \times K^N$, typically drawn from a multinomial distribution.
- The emission probability matrix B with dimensions $K^N \times 2$, typically drawn from a Gaussian distribution since observations are continuous.

Expectation-Maximization is used to train the model parameters and the Viterbi algorithm to decode the most likely state sequence.

Pros	Cons
State Transitions are Naturally Modelled	Even more computationally Inefficient
Improved Disaggregation Accuracy	Harder Implementation

The Computational Cost is $O(TK^{2N})$ where N is the number of appliances, K is the number of appliances states and T is the number of time slices in the dataset. FHMM is more capable than CO but still extremely computationally inefficient. Disaggregation across many appliances with multiple states and for a large time window is almost impossible! Methods such as variational inference can produce approximations that are more computationally efficient.

Convolutional Neural Networks (ConvNet)

Convolutional Neural Networks are similar to ordinary Neural Networks (multi-layer perceptrons). Each neuron receive an input, perform a dot product with its weights and follow this with a non-linearity (here we only use ReLu). The whole network has a loss function that is here the RMS error. The network implements the 'rectangle method'. From the input sequence we invert for the start time, the end time and the average power of ONE appliance.

Pros	Cons
Quite Effective	Difficult to Implement
Fast Disaggregation	Training is time consuming

METHODS FOR DISAGGREGATION (2)

Convolutional Neural Networks (ConvNet)

Convolutional neural networks have revolutionized computer vision. From an image the convolutional layer learns through its weights low level features. In the case of an image the features detectors (filters) would be: horizontal lines, blobs etc. These filters are built using a small receptive field and share weights across the entire input, which makes them translation invariant. Similarly, in the case of time series, the filters extract low level feature in the time series. By experimenting we found that only using 16 of these filters gives a good predictive power to the ConvNet. This convolutional layer is then flatten and we use 2 hidden layers of 1024 and 512 neurons with ReLu activation function before the output layer of 3 neurons (start time, end time and average power).

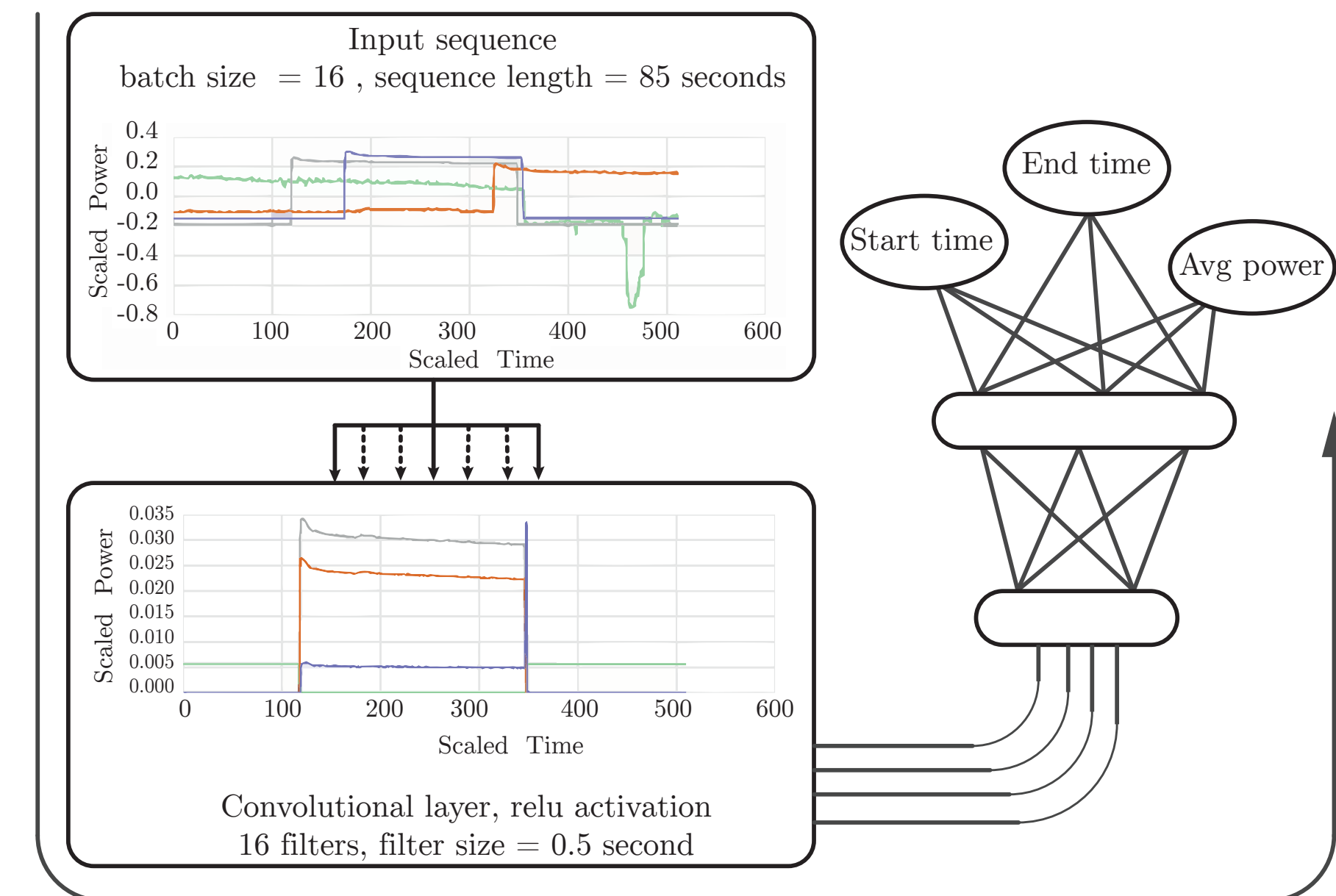


Figure: A schematic representation of the architecture of the convolutional neural network

ACCURACY METRICS

It is necessary to introduce a number of metrics in order to access the accuracy of energy disaggregation for each method in order to facilitate the comparison between them but also to outline the flaws and limitations of each one. In the context of this project, the following metrics were used, implemented and computed for all of the methods.

TP-FP-FN-TN

The acronyms stand for True Positives, False Positives, False Negatives and True Negatives respectively. The definitions using the problem's notation are:

$$TP^{(n)} = \sum_t \text{AND} \left(x_t^{(n)} = on, \hat{x}_t^{(n)} = on \right) \quad (2)$$

$$FP^{(n)} = \sum_t \text{AND} \left(x_t^{(n)} = off, \hat{x}_t^{(n)} = on \right) \quad (3)$$

$$FN^{(n)} = \sum_t \text{AND} \left(x_t^{(n)} = on, \hat{x}_t^{(n)} = off \right) \quad (4)$$

$$TN^{(n)} = \sum_t \text{AND} \left(x_t^{(n)} = off, \hat{x}_t^{(n)} = off \right) \quad (5)$$

where $x_t^{(n)}, \hat{x}_t^{(n)}$ are the predicted and ground truth states of appliance n at t respectively.

Precision / Recall

Precision is the fraction of time for which an appliance was correctly predicted to be ON that it was actually OFF. Recall is the fraction of time for which an appliance was correctly predicted to be ON that it was actually ON. The definitions are:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

Accuracy / F1

Accuracy is defined as the fraction of time where the appliance was correctly classified as either on or off whereas the F score is a harmonic mean of the Precision and Recall.

$$\text{Accuracy} = \frac{TP + TN}{P + N}, \quad F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

where P and N are the positives and negatives in ground truth.

RESULTS AND COMPARISON

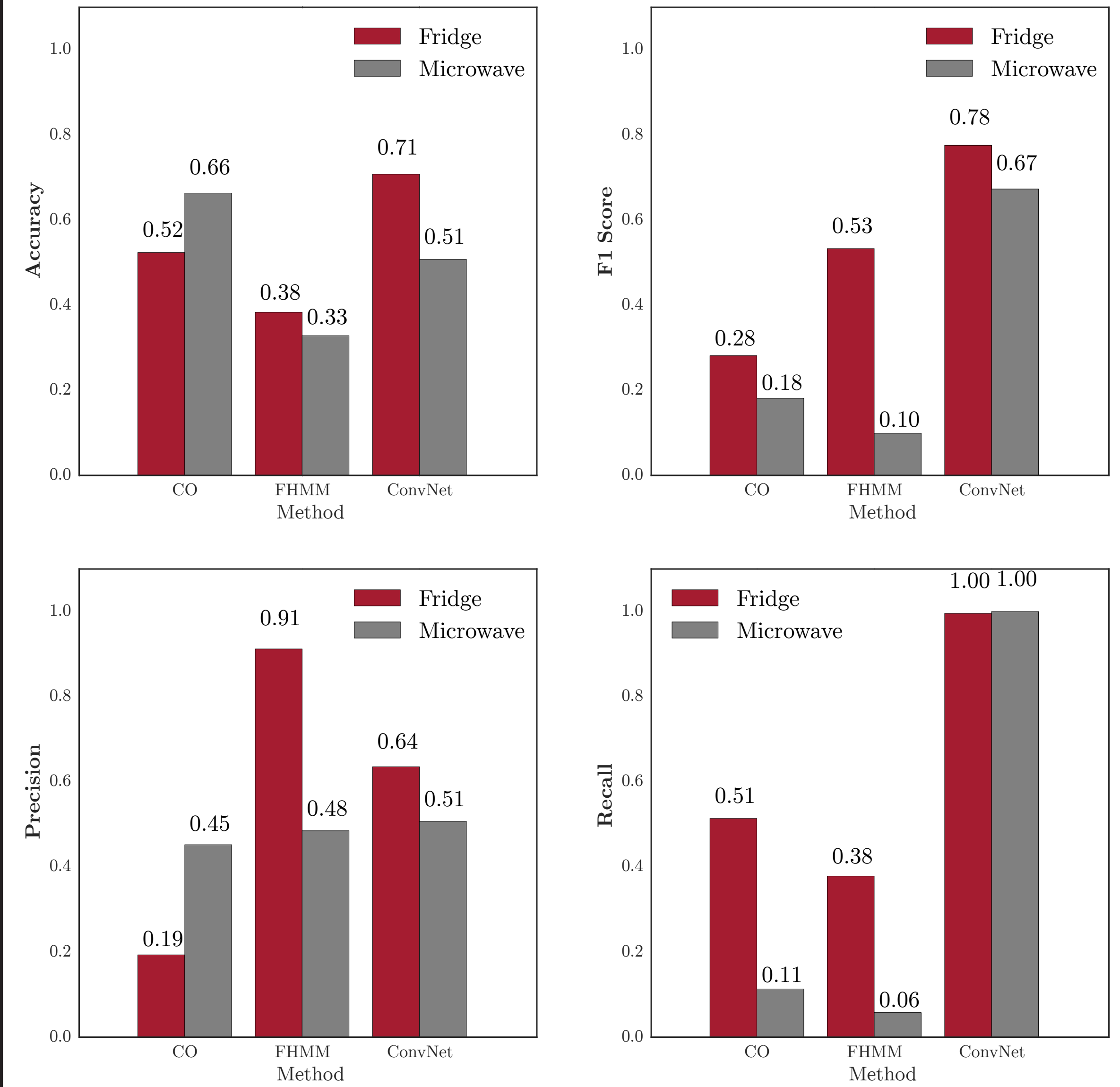


Figure: The Accuracy, F1, Recall and Precision Scores for each method used for Disaggregation and per appliance disaggregated

DISCUSSION

Appliance based energy disaggregation is a complicated and very difficult problem if it is to be performed on data from actual households over long periods of time and for a wide class of appliances. The disaggregation efficiency of deterministic methods commonly employed in these types of problems such as Combinatorial Optimization is extremely poor and the computational cost significant. Stochastic methods are certainly much more capable in these types of problems and the metrics corresponding to the Factorial Hidden Markov Model underline this point. The computational cost for FHMM is even more significant than that of CO which suggests that even FHMM is not suitable for disaggregation over large data sets. In contrast, the efficiency achieved by the Convolutional Neural Network is remarkable! It outperforms all other methods in every metric and is ideal for these types of problems. Analysis performed by members of our team indicates that the neural network is capable of achieving high accuracy disaggregation even with a few layers. The downside is that it requires considerable expertise to implement and a large data set to train. Training usually takes a very long time for Neural Nets but the network, once trained, is able to disaggregate fairly quickly and certainly much quicker than CO and FHMM.

REFERENCES

- [1] Batra, N. Kelly, J. Parson, O. Dutta, H. Knottenbelt, W. Rogers, A. Singh, A. Srivastava, M., (2014), NILMTK: An Open Source Toolkit for Non-intrusive Load Monitoring, *Int. Conference on Future Energy Systems*, Cambridge, UK
- [2] Kelly, J. Knottenbelt, W. (2015), Neural NILM: Deep Neural Networks Applied to Energy Disaggregation, *ACM BuildSys'15*, Seoul
- [3] Hart G. W. (1992), Nonintrusive Appliance Load Monitoring, *Proceedings of the IEEE*, (80)12
- [4] Kolter, J.Z. Johnson, M.J. (2011), REDD: A Public Data Set for Energy Disaggregation Research, *SustKDD 2011*, San Diego, CA, USA
- [5] Hart G. W. (1992), Nonintrusive Appliance Load Monitoring, *Proceedings of the IEEE*, (80)12