Energy Aggregation using Product of Experts[*]

Megha Gupta meghag@iiitd.ac.in
Haimonti Dutta haimonti@buffalo.edu
Amarjeet Singh amarjeet@iiitd.ac.in
Ullas Nambiar Ullas.Nambiar@emc.com

Abstract

1 Introduction

In todays scenario, wireless sensor networks (WSN) have found wide range of applications in areas like military, health, environmental sensing, etc. Though WSNs have been a significant improvement over traditional sensors but they have their own constraints like nodes with limited power, computational capacities and memory [Akyildiz *et al.*, 2002]. Therefore, a sensor node lifetime is strongly dependent on its battery lifetime. In the context of energy management, several challenges like finding data, collecting the right data and compiling all the data from different sources are to be faced. Efficient energy data aggregation is thus one of the leading problems in wireless sensor networks. Data aggregation also known as data fusion is defined as set of methods for combining the data that comes from many sensor nodes into a set of meaningful information [Heinzelman *et al.*, 2000].

In recent years, machine learning has been applied to the problem of energy consumption analysis[SHA *et al.*, 2010][Liao *et al.*, 2008][Liao *et al.*, 2011][Chen *et al.*, 2008][Yuea *et al.*, 2012][Lin *et al.*, 2012][Liu *et al.*, 2007]. Sensor data collected from smart homes are used to reveal activity patterns of the residents, which can then be correlated with the total energy consumption. This enables utility companies and their customers to associate activities with energy usage and costs, devise intelligent systems to control home environments improving energy efficiency and reducing costs. Typically, sequences of usage patterns that appear frequently at different time scales (daily, weekly, monthly, yearly) and across different homes are studied and outlier detection algorithms are designed to enable customers to be notified that they are consuming unusually large amount(s) of energy during some specific period. Related problems involve study of trends of electricity consumption (steadily increasing, decreasing, cyclic, seasonal) and sudden anomalous behavior (sudden peaks or drops on consumption) for individual homes and across the community. The role of the machine learning algorithm is to study the sensor data and provide alerts and warnings when anomalous behavior occurs or to inform (and remind) customers when certain activities were performed, which rooms they occupied, and what appliances they used most frequently during that period. This information can be transmitted to customers in timely fashion via phone, email or the Internet.

In this paper, we build machine learning models using products of HMM and apply them to the energy aggregation problem. Two different proof of concepts are presented – one on the REDD data set and another on real data collected at the faculty housing in India. There are many reasons why the product model constructed from many HMMs is appropriate. First, this model is ideal for data which is caused by multiple underlying influences. Second, HMMs alone are not efficient at capturing long range structure in time series (CITE) – in contrast to product of hidden markov models (PoHMM) [Brown, 2001] allow each model to remember a different piece of information about the past. [SAY SOMETHING MORE HERE]

Organization: This paper is organized as follows: Section 2 examines related work on energy aggregation; Section 3 provides a review of automata and their products and subsequently leads to an understanding of products of Hidden Markov Models (HMMs). Inference from Products of HMMs is dealt with in Section 4. The two proofs of concepts are introduced in Section 5 to illustrate the effectiveness of the use of product of HMMs in the energy aggregation problem. Finally, Section 7 concludes the work.

2 Related Work

2.1 Energy Aggregation

Devaine et al. (CITE) study ...

2.2 Prediction with expert advice

3 Review of Automata and their products

Distributed networks can be modeled using interacting automata. Benveniste [Fabre *et al.*, 2000] defines automaton as a quadruple, Á = (X,X_0,A,T) where X is a finite state of sets, X_0 is the subset of initial states, A is a finite set of messages, T is a set of transitions of the form t = $\{x_-,a,x\}$ where x_- is the previous state, a is the message label on which the state transitions to the next state x. The figure 1

[*] These match the formatting instructions of IJCAI-07. The support of IJCAI, Inc. is acknowledged.

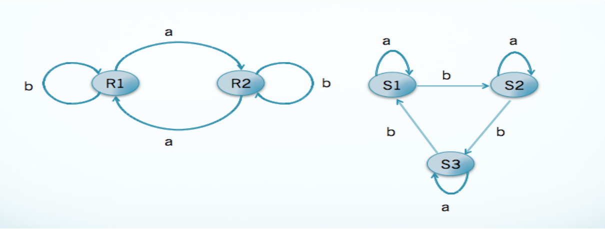


Figure 1: Automata R and S

below explains the automata with an example.

For automaton R, X$_R$ = {2; R1,R2}, X$_{0R}$ = {R1}, A$_R$ = {a,b}, T$_R$ = {R1,a,R1; R1,b,R2; R2,a,R2; R2,b,R1}
For automaton S, X$_S$ = {3; S1,S2,S3}, X$_{0S}$ = {S1}, A$_S$ = {a,b}, T$_S$ = {S1,a,S1; S1,b,S2; S2,a,S2; S2,b,S3; S3,a,S3; S3,b,S1}
The product of two automata Á = R x S is defined as follows:
X = X$_R$ x X$_S$
X_0 = X$_{0R}$ x X$_{0S}$
A = A$_R$ ∪ A$_S$
Benveniste uses a notion of stuttering transition which helps to distinguish between local and global time by inserting dummy transitions between two transitions of a local automaton attached to a node. This stuttering transition waits for others to progress.

A	R1	R2
R1	0.6	0.4
R2	0.3	0.7

Table 1: Transition probability, A

B	a	b
R1	0.2	0.8
R2	0.5	0.5

Table 2: Observed probability, B

	R1	R2
π	0.4	0.6

Table 3: Initial state probability, π

Talking in terms of HMM, requires us to equip products of automata with probabilities. Benveniste defines HMM as a triple (Á, μ, π) where Á = (X,X_0,A,T) is an automaton, μ is the initial state probability, π is factored as state transition probability π_x and message transition probability π_A. He uses a random arbiter α, with values first, second, third to choose automaton to initiate transition. If α = first then first automaton chooses any transition having a private message whereas second automaton performs a stuttering transition, and vice versa for α = second. If α = both, then both automata agree on some shared message and move accordingly.

Using the traditional HMM notation of the parameters λ = {A, B, π } where A is the transition probability, B is the observed probability, π is the initial state probability. For automata R, we have the values of A, B, π as shown in table 1, 2, 3 respectively.

3.1 Product of HMM

PoHMM is a combination of directed and undirected graphical models. The hidden states are connected with directed links where as the connection with visible states is through undirected links. This causes different conditional independence relationships among the variables in graphical model. Product of HMM is a way of combining HMM's to form distributed state time series model. It is defined by multiplying together the densities of its, k experts and renormalizing them. The figure 2 is a product of two HMMs shown in 1. For P = R x S, the quadruple becomes
X = {6; R1S1, R1S2, R1S3, R2S1, R2S2, R2S3}
X_0 = {R1S1}
A = {a,b}
The rules for synchronised product construction are :
1. $< p, q >$ –a–> $< p', q >$ if a ∈ A$_R$ ∩ A$_S$ and p –a–> p' and q –a–> q'
2. $< p, q >$ –a–> $< p', q >$ if a ∈ A$_R$, a ∉ A$_S$ and p –a–> p'
3. $< p, q >$ –a–> $< p, q' >$ if a ∉ A$_R$, a ∈ A$_S$ and q –a–> q'

4 Methodology

4.1 Inference in PoHMM

The main feature of PoE is its undirected graphical modelling with no direct connection among the latent variables as they only interact indirectly via observed variables. The hidden variables all the experts are rendered independent when conditioned on visible variables. So, if the inference in each of the constituent model is tractable then the inference in the product is also tractable. To generate a data point in this model, all the experts in PoE generate an observation and if they all generated the same point then it is accepted else they again generate an observation until all the experts agree to it. Therefore all the experts have some influence over the generated data. So, the inference determines the the probability that all the experts would have taken in order to generate the given observation.

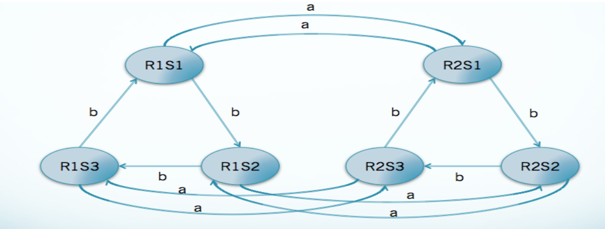


Figure 2: Product of HMMs, P = R x S

4.2 Training product of experts by minimising contrastive divergence

PoE is a method of combining densities of many latent variable models. It is defined by the following formula:
To fit the model to the data, we need to maximize the likelihood of the dataset or minimise the Kullback-Liebler divergence between the real data and the fantasy data. The contrastive divergence algorithm for training the PoHMM has the following steps:

1. Calculate each model's gradient on a data point using forward backward algorithm.
2. For each model take a sample from the posterior distribution of paths through state space.
3. At each time step, multiply together the distributions and renormalize to get the reconstruction distribution at each step.
4. Draw a sample from the reconstruction distribution at each time step to get a reconstructed sequence. Compute each model's gradient on the new sequence.
5. Update the parameters

5 Proof of Concepts

5.1 REDD House 2

5.2 Aim

To represent streams of energy consumption data from n^1 appliances by product of k HMMs.

5.3 Method

- **Data** The Reference Energy Disaggregation Data Set (REDD) is used in empirical analysis. The data contains power consumption from real homes, for the whole house as well as for each individual circuit in the house (labeled by the main type of appliance on that circuit). It is intended for use in developing disaggregation methods, which can predict, from only the whole-home signal, which devices are being used. The REDD data set contains two main types of home electricity data: high-frequency current/voltage waveform data of the two power mains (as well as the voltage signal for a single phase), and lower-frequency power data including the mains and individual, labeled circuits in the house. The main directory consists of several house directories, each of which contain all the power readings for a single house. Each house subdirectory consists of a labels and channels files. The labels file contains channel numbers and a text label indicating the general category of device on this channel. Each channel_i.dat file has two columns containing UTC timestamps (as integers) and power readings (recording the apparent power of the circuit) for the channel. Experiments reported here use the House 2 data from REDD. It has 11 channels where each channel corresponds to the following appliance:

  1. mains_1
  2. mains_2
  3. kitchen_1
  4. lighting
  5. stove
  6. microwave
  7. washer_dryer
  8. kitchen_2
  9. refrigerator
  10. dishwaser
  11. disposal

  The dataset has 318759 records and 2 columns. We randomly sample 300 records for our initial experiment. Time series data from two appliances are represented as product of k HMMs.

- **Time Series :** The time series data of the microwave, dryer, kitchen_2 and refrigerator are plotted below in Figures 3, 4, 5, 6.

- **Code** The implementation of the product of experts model is obtained from Iain Murray's website[2]. It implements the technique described in Geoff Hinton's paper [Hinton, 2000].

- **Additional details** Some additional details regarding experiments:

  1. The product of HMMs model (PoHMM) minimizes "contrastive divergence" as described in the paper [Hinton, 2000].
  2. The number of experts, k used here is 15. This is set somewhat arbitrarily and needs to be experimented on.
  3. Learning rate is $\epsilon = \frac{1}{300}$.

[1] n=2

[2] http://homepages.inf.ed.ac.uk/imurray2/code/

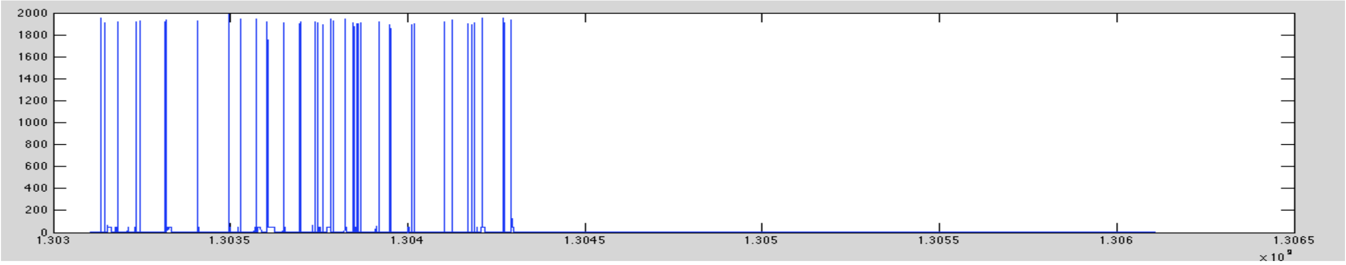


Figure 3: Microwave

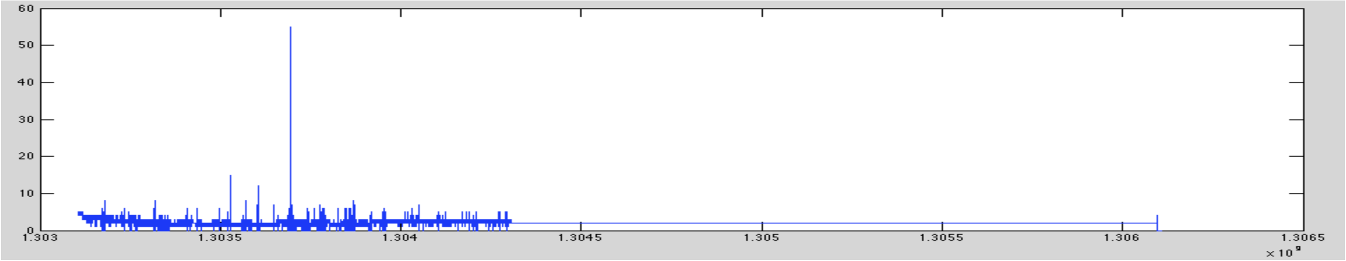


Figure 4: washer_dryer

5.4 Experimental Setup for REDD house 2

Experiments are performed on the REDD which contains 9 appliances each containing 318759 rows of energy consumption data. Experiments are done into 4 phases, in the first phase the number of data samples are varied corresponding to which the values of KL Divergence and convergence time are noted. In the second phase, the number of experts are varied keeping the best value of the sample from the first phase fixed. In the third phase, number of iterations are varied keeping the best values from above first two phases fixed. In the fourth part, the no. of appliances to be aggregated are varied.

Samples	$KLDiv$	$T(sec)$	$Iterations$
300	2.4864	186.212 ±9.087	18600
500	0.6761	106.564 ±10.046	10200
1000	1.1088	158.521 ±1.97	11200
1500	3.8829	92.896 ±8.075	5300
2000	1.8686	130.98 ±1.932	6900
2500	0.4733	215.563 ± 2.471	9900
3000	2.8204	258.213 ±1.918	11000
3500	1.2332	204.661 ±1.713	7900
4000	0.8959	292.666 ±0.619	10400
4500	1.1118	222.558 ±1.967	7200
8000	6.392	381.635 ±2.952	8100
10000	8.276	887.932 ±13.824	10500
15000	0.7201	1368.514 ±13.605	9400

Table 4: Effect of varying samples on KL div and time

5.5 Results

The evaluation of how well the learning has taken place is done by using a Kullback-Leibler divergence. KL divergence of P from Q, D_{KL}(P‖Q) is the measure of information lost when Q is used to approximate P. Here, P is the real data and Q is a fantasy data. The two probability distributions in the REDD example refer to the expert probabilities in real and fantasy data. The learned parameters from the training are fitted to the fantasy data to measure the information lost when fantasy data is used to approximate real data.

Experts	$KLDiv$	$T(sec)$	$Iterations$
5	0.774	72.968 ±1.177	5200
10	1.424	117.482 ±1.966	6700
15	0.473	210.249 ±1.258	9900
20	1.56	217.739 ±10.452	9000
25	7.469	347.019 ±8.23	12100
30	2.4968	413.802 ±7.304	12900
35	1.5012	348.906 ±14.651	11300

Table 5: Effect of varying experts on KL div and time

Threshold	$KLDiv$	$T(sec)$	$Iterations$
.1	0.473	210.6 ±1.493	9900
.05	0.443	240.607±2.436	10900
.01	0.454	431.536 ±14.509	18000
.005	0.509	1167.243 ±43.412	49800

Table 6: Effect of varying min threshold on KL div and time

6 Proof of concept on Faculty housing data

6.1 Aim

To represent streams of energy consumption data from all the floors of faculty housing as a product of k HMMs.

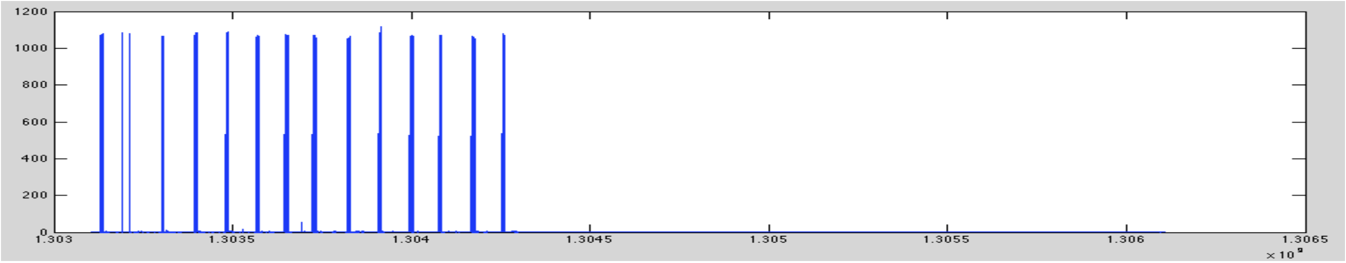


Figure 5: Kitchen_2

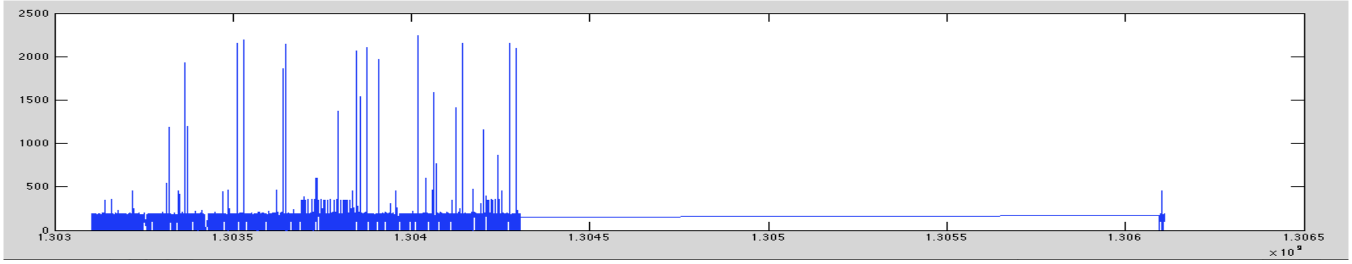


Figure 6: refrigerator

Appliances	$KLDiv$	$T(sec)$	$Iterations$
3	5.559	233.664 ±0.579	10700
4	0.188	465.634 ±5.275	19900
5	.432	338.416 ±3.988	13400
6	8.736	606.062 ±7.534	28100
7	5.054	411.457 ±10.051	17300
8	0.436	260.544 ±cc27.862	10700
9	0.15	474.579 ±14.619	20600

Table 7: Effect of varying appliances on KL div and time

6.2 Method

- **Data** This data represents the energy consumed by the IIIT Delhi faculty housing building. As a part of research, a team from IIIT Delhi has installed various temperature, light and motion sensors to perform real world studies and to analyse user preferences for energy conservation. For our analysis, we selected one month's historical data ranging from 01-01-2014, 00:01 hours to 31-01-2014, 23:59 hours. The two smart meters installed captures the data from all the floors. The first meter gives out readings from floors 0 to 5 and the second meter gives out readings from floors 6 to 11. The dataset includes timestamp and power consumed in watts and 84133 records. Time series data from two streams are modelled as a product of k HMMs. We also have the total power consumed by the faculty housing building which would serve as the ground truth to compare product of k HMMs with. The data is obtained from the website whose screenshot is shown in Figure 7

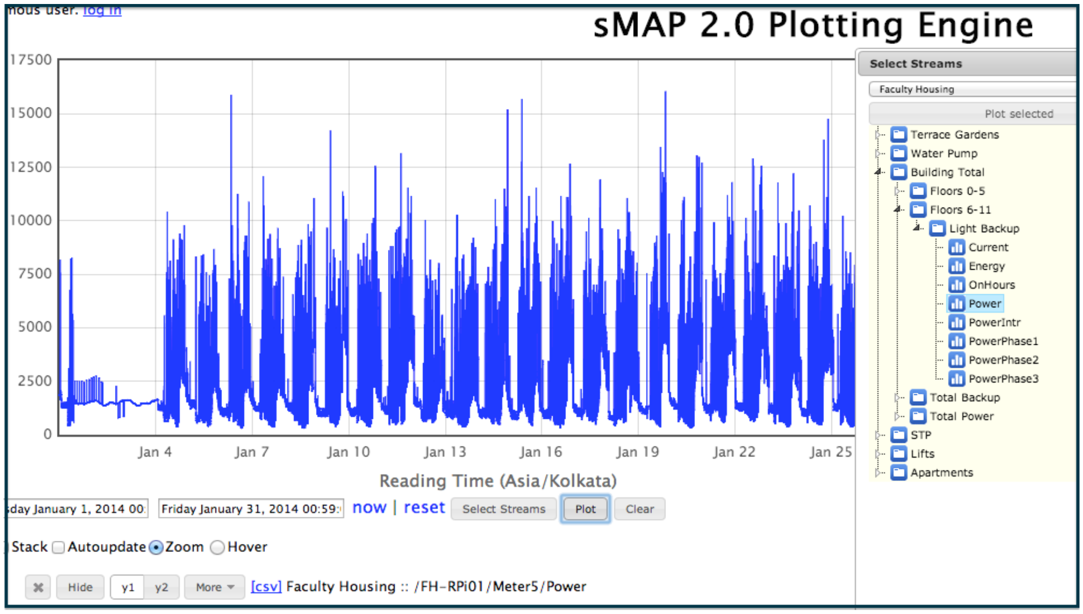
- **Code** It implements the technique described in Geoff Hinton's paper [Hinton, 2000].

- **Time Series :** The time series data of the energy consumption of floor 0 to 5, floor 6 to 11 and total power are plotted below in Figures 8, 9, 10.

- **Additional Details**

6.3 Experimental Setup

Each of the data stream is modelled as a HMM individually. There are three streams of data, the first stream D1 corresponds to the data from 0-5th floor, D2 corresponds to 6-11th floor and D3 represents the total power from the faculty housing which is represented by a fixed test set, T. Firstly, the stream D1 is used to train the model such that the contrastive divergence is minimized. The parameters (mixing component of each unigauss, means of gaussian bits, log precisions of axis-aligned gaussian bits) that are learnt during the training are provided to the test set T in order to obtain the conditional probability of the gaussians given the data D1 represented as pgauss1. Similarly, the second stream of data, D2 collected from floor 6-11, is used to learn the parameters of the model during the training phase which are then again provided to the test set T to obtain conditional probability of the gaussians given the data D2 as pgauss2. Finally the data D3 is used to learn the model and parameters which are then applied to the test set T to obtain the gaussian probability as pgauss3. Now, as we know that the total power consumption of the building should be approximately equal to the product of HMMs, which is the product of pgauss1 and pgauss2. If we can show that the value of pgauss3 is as close as possible to the product of pgauss1 and pgauss2.
The experiments performed in table 8 , shows the effect of varying samples on KL Divergence, convergence time and iterations keeping minimum threshold constant at 7.
The other experiment performed in table 9 shows that effect of varying experts on KL Divergence and convergence time.



day January 1, 2014 00 | Friday January 31, 2014 00:59

Stack □ Autoupdate ● Zoom ○ Hover

[csv] Faculty Housing :: /FH-RPi01/Meter5/Power

Figure 7: Screen shot of the webpage

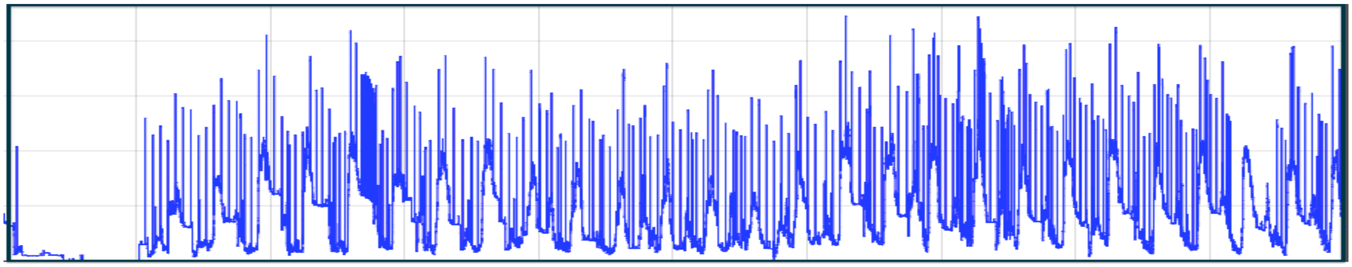


Figure 8: Stream 1: Power consumption of floors 0-5

6.4 Results

Table 8 shows that the error was minimum when the sample size was 300. With respect to the number of experts, the error was minimum when there were 5 experts as shown in table 9.

7 Conclusion & Future Work

The conclusion goes here. this is more of the conclusion

References

[Akyildiz *et al.*, 2002] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.

[Brown and Hinton, 2001] Andrew Brown and Geoffrey Hinton. Proceedings of artificial intelligence and statistics 2001. In *Products of Hidden Markov Models*, number GCNU TR 2000-008, 2001.

[Brown, 2001] Andrew Dennis Brown. Product model for sequences, 2001.

Samples	$KLDiv$	$T(sec)$	$Iterations$
100	2.6219e-05	257	45100
300	1.9753e-05	222	43200
500	5.5493e-05	260	44800
700	3.2847e-05	249	44000
900	3.9486e-04	221	42600
1100	4.9274e-04	317	44700
1300	3.0425e-04	276	43100
1500	3.1128e-04	303	44400
2000	1.9192e-04	306	44400
2500	1.7122e-04	370	44100
3000	1.4686e-04	331	43300
3500	1.2663e-04	370	43200
4000	1.0793e-04	403	43200

Table 8: Effect of varying samples on KL div

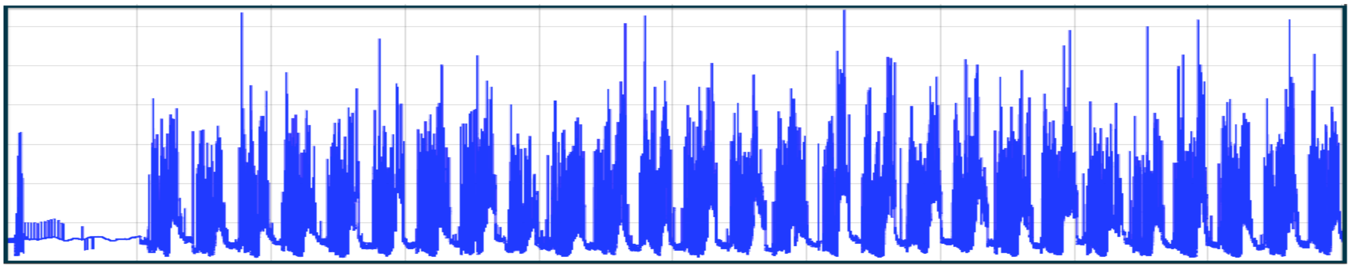


Figure 9: Stream 2: Power consumption of floors 6-11

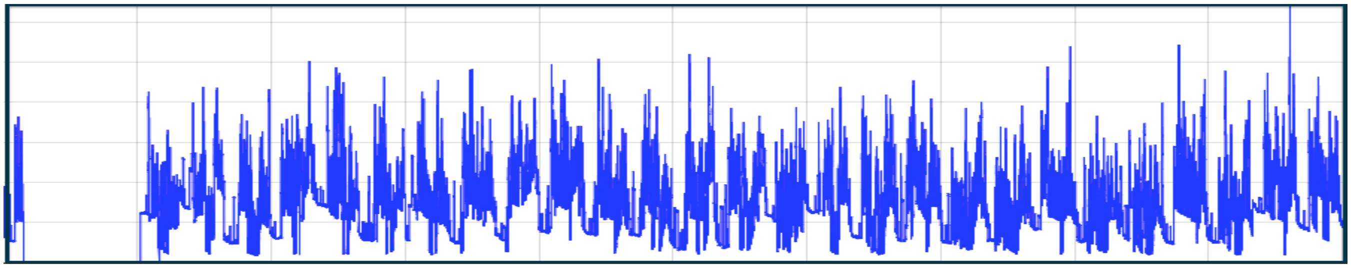


Figure 10: Total Power of the building

Experts	$KLDiv(e-05)$	$T(sec)$
3	1.9780	229
4	3.5897	217
5	1.9753	228
6	4.3488	238
7	4.9111	245
8	5.6564	241
9	5.4290	258
10	5.5163	267
12	4.4504	262
14	6.9006	296
16	6.8666	300
18	6.2872	313
20	5.3842	267
25	5.8970	326
30	5.9962	327
35	5.2716	346
40	5.0955	320

Table 9: Effect of varying experts on KL div and time

[Chen *et al.*, 2008] Huifang Chen, Hiroshi Mineno, and Tadanori Mizuno. Adaptive data aggregation scheme in clustered wireless sensor networks. *Computer Communications*, 31(15):3579 – 3585, 2008.

[Fabre *et al.*, 2000] Eric Fabre, Stefan Haar, Albert Benveniste, and Albert Benveniste. Hidden markov models for distributed and concurrent systems, 2000.

[Ghahramani and Jordan, 1997] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. *Mach. Learn.*, 29(2-3):245–273, November 1997.

[Heinzelman *et al.*, 2000] Wendi Rabiner Heinzelman, Anantha Ch, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. pages 3005–3014, 2000.

[Hinton, 2000] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. Technical Report GCNU TR 2000-004, Gatsby Computational Neuroscience Unit, University College London, 2000.

[Liao *et al.*, 2008] Wen-Hwa Liao, Yucheng Kao, and Chien-Ming Fan. Data aggregation in wireless sensor networks using ant colony algorithm. *Journal of Network and Computer Applications*, 31(4):387 – 401, 2008.

[Liao *et al.*, 2011] Wen-Hwa Liao, Yucheng Kao, and Ru-Ting Wu. Ant colony optimization based sensor deployment protocol for wireless sensor networks. *Expert Systems with Applications*, 38(6):6599 – 6605, 2011.

[Lin *et al.*, 2012] Chi Lin, Guowei Wu, Feng Xia, Mingchu Li, Lin Yao, and Zhongyi Pei. Energy efficient ant colony algorithms for data aggregation in wireless sensor networks. *Journal of Computer and System Sciences*, 78(6):1686 – 1702, 2012. {JCSS} Multidisciplinary Emerging Networks and Systems (MENS).

[Liu *et al.*, 2007] Chuan-Ming Liu, Chuan-Hsiu Lee, and Li-Chun Wang. Distributed clustering algorithms for data-gathering in wireless mobile sensor networks. *Journal of Parallel and Distributed Computing*, 67(11):1187 – 1200, 2007.

[SHA *et al.*, 2010] Chao SHA, Ru chuan WANG, Hai ping HUANG, and Li juan SUN. Energy efficient clustering algorithm for data aggregation in wireless sensor networks. *The Journal of China Universities of Posts and Telecommunications*, 17, Supplement 2(0):104 – 122, 2010.

[Yuea *et al.*, 2012] Jun Yuea, Weiming Zhang, Weidong Xiao, Daquan Tang, and Jiuyang Tang. Energy efficient and balanced cluster-based data aggregation algorithm for wireless sensor networks. *Procedia Engineering*, 29(0):2009 – 2015, 2012. 2012 International Workshop on Information and Electronics Engineering.