CS446 Project: Electric Load Identification using Machine Learning

Anand Deshmukh (adeshmu2@illinois.edu)
Danny Lohan (dlohan2@illinois.edu)

May 8, 2015

Abstract

In this project report we present the idea of electric load identification using supervised machine learning. The primary goal of this project is to identify the different devices running in the household using whole house electric consumption data and estimate the individual device energy consumption. We present a framework for creating the appropriate features and labels from the training data and use these features to predict the device status (on/off) and device energy consumption using a variety of classifiers. Finally, we identify a best performing classifier based on a thorough error analysis on test data. The best classifier was found to give sufficient predictions of device energy consumption.

1 Introduction

The goal of this project is to identify different devices running in the household using whole house electric consumption data and estimate the individual device energy consumption. Such an information can be immensely helpful for consumers to make an informed decision about their energy usage. The use of machine learning for this task is motivated by the need to have minimal number of sensors to monitor the device consumption (ideally only one sensor monitoring the whole house consumption). We would like to perform the task of energy desegregation for different devices in a household by using whole house power consumption signal and estimate their individual energy consumption for the duration of interest. Specifically, given the whole-home energy consumption signal we would like to predict which devices are being used. Our model consists of aggregated energy signals from difference devices, hence there will be no noise in the data. We propose to apply different machine learning techniques to this multi-class classification problem and perform a comparative analysis of such techniques.

The report is structured as follows. We present the background and literature review in this field in Section 2. In Section 3 we detail the task of the supervised machine learning and data necessary to accomplish that task. In Section 4 we describe the learning models considered in this project followed by the experimental results in Section 5. Finally we conclude the report in Section 6 and identify the future work.

2 Background

The typical 'intrusive' household load monitoring involves connecting multiple devices in the house to sensors. This procedure is typically done as a part of the energy audit to identify the highest power sinks in the house. Historically such an energy audit has been performed manually by skilled technicians after observing individual device power consumption periodically. The manual audit process even though most accurate is not the most cost- and time-effective.

To remedy this, the idea of using machine learning for household energy audits by intelligently disaggregating the whole-home energy consumption was proposed early on by Hart [1]. He coined the term non-intrusive load monitoring (NILM) to refer to this approach. In [1], Hart describes a procedure named non-intrusive appliance load monitoring (NALM[1]) to perform the classification, either by manual

[1]NALM is one specific procedure in the general field of NILM

or automatic setup. Manual setup describes the procedure where a technician would manually switch devices on and off to extract the characteristics of that device. By learning the on, off, and steady state signals of each devices, predictions can be made of which devices are active (on) by analyzing the whole house energy signal. The procedure as described by Hart [1] is presented in Figure 1.

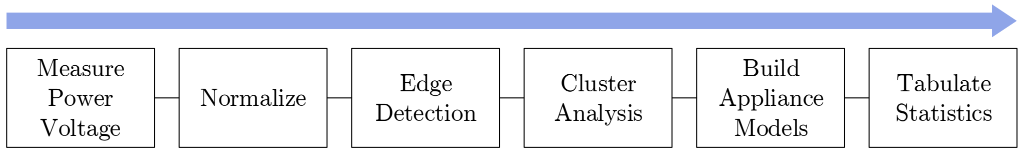

Figure 1: NALM Procedure

This NALM procedure formed a basis for further work in NILM [2, 3]. A variety of algorithms have since been proposed to improve the NILM, a survey of these algorithms can be found in [4]. The recent notable work in NILM has involved supervised [3] as well as unsupervised learning [5]. Another approach is taken using Hidden Markov models [6, 7, 5]. Beyond residential applications, several authors have looked to apply NILM to other areas. Kolter et. al. [8] extend this procedure to identify energy patterns relating to apartment complexes in a city sector where as Laughman et. al. [9] describe using NILM to identify load signatures in more complex industrial settings.

In this project we evaluate supervised learning approaches and perform a comparative analysis to identify the best performing algorithm. Using a supervised learning scenario, the algorithms try to identify characteristics of devices in an attempt to determine their status (on/off) and power consumption. The classifiers would then be used to predict device use throughout the day and a energy consumption can thus be estimated.

3 Task and Data

In this section we look at the machine learning tasks of this project and the data we use to perform the learning tasks on.

3.1 Task

The supervised learning task here is to train a multi-class classifier that can predict which group of devices are active (on) during the observation period and estimate the individual device energy consumption. The observation signal (i.e. power consumption signal) is obtained for the whole house and features are extracted from the signal and the instances $\mathbf{x}_i$ are created. Similarly, since this is a supervised learning task we also have access to individual device consumption data to create the labels $\mathbf{y}_i$. Models are evaluated based on their ability to correctly predict which devices are running given an aggregate signal at a point in time. Further metrics of success evaluate the accuracy estimating energy consumption of each device based on the predicted labels. The overall goal is to help a consumer identify which household devices are consuming the most energy.

3.2 Energy Consumption Data (REDD)

For this project we use REDD dataset [8]. This dataset consists of electric power consumption data for different houses in Cambridge, MA for the duration of approximately 45 days from 04/16/2011 to 05/31/2015. Each house participating in the study was fitted with multiple sensors to monitor individual device level power consumption data as well as the whole house power consumption data. The individual household devices considered in this study are listed in Table 1.

The energy signals are reported every second for the mains and every three seconds for the individual circuits. Data logging began on 04/16/2011 at 05:11:27 and continued until 05/31/2011 at 00:19:54. We will use REDD data for training and testing using a 80%-20% split between training and test data. The individual device power consumption data for four devices for a random time interval are shown in Figure 2.

Table 1: Energy consuming devices in the house considered for this study

Device #	Name
0	Electronics
1	Refrigerator
2	Dishwasher
3	Furnace
4	Washer Dryer 1
5	Washer Dryer 2
6	Microwave
7	Bathroom GFI
8	Kitchen Outlets

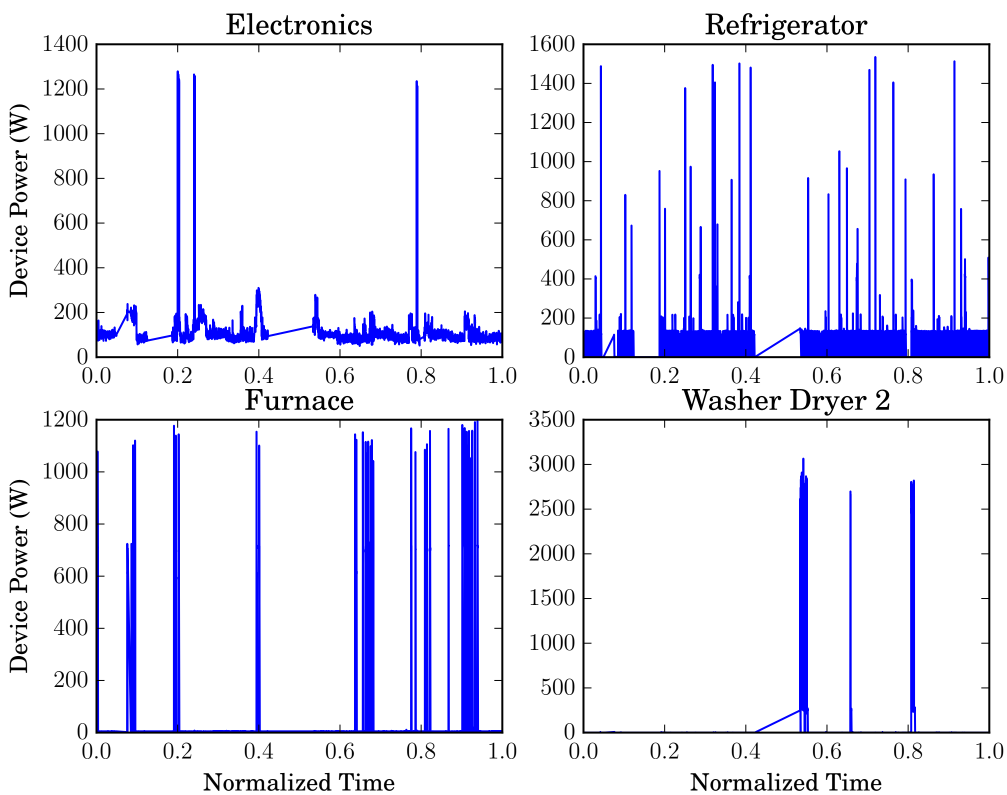

Figure 2: Typical device load signatures for four devices over approximately two weeks

3.3 Weather Data

It is anticipated that during the observation period, weather will also affect the power consumption in the house. To address that we also considered the hourly weather data for the Boston, MA area for the observation period. We acquired the weather data from the National Centers for Environmental Information [10]. This data consists of readings taken roughly every ten minutes. Out of all the weather parameters available in this data we use only the *Dry Bulb Temperature* information for this work as this temperature is most relevant for the operation of refrigeration and and HVAC systems in the house.

3.4 Data Processing

In this work we divide the training and testing data into snippets of five minute signals. Each five minute snippet contains the data for whole house power consumption as well the individual device consumption. The whole house data snippets are used to create the instances where as the the individual device snippets are used to create the labels for each of the instances.

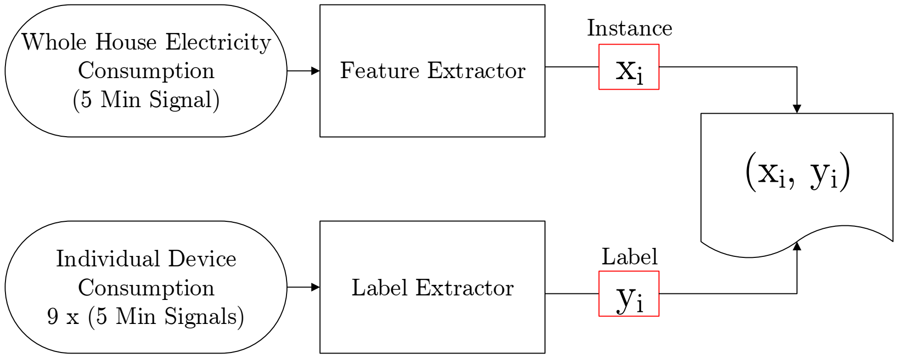

Figure 3: Preprocessing of data for feature and label extraction during training phase

Feature Extraction

Each five minute whole house data snippet is processed to extract seven features as shown in Table 2. These features for one five minute whole house data snippet are combined together to form an instance. The energy of signal feature is the power consumed over the observed duration: $E(t) = \int_0^t P(\tau)d\tau$.

Table 2: Features considered for this study

Feature	Description	Unit
μ_P	Mean power of the sampled signal	W
σ_P	Standard deviation of the sampled signal	W
t_d	Time of the day	Number
T	Ambient temperature	°F
P_m	Peak power	W
E	Energy of the sampled signal	W·s
D_w	Day of the week	Number

Label Assignments for Training Data

Using the five minute individual device snippets we then create labels to go with the instances. If the device is 'on' during that five minute snippet then a '1' is assigned for the corresponding device and the 9-bit vector (corresponding to 9 devices) is generated (see Table 3 for details). The integer corresponding to the 9-bit vector is used as the class label. We have such 512 labels in this work.

Table 3: Label assignment based on device status in the individual device signal

Device #									Label
0	1	2	3	4	5	6	7	8	
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	1	1	0	510
1	1	1	1	1	1	1	1	1	511

3.5 Learning Task

In this section we briefly mathematically formalize the prediction task at hand.

Label Prediction

As mentioned in the introduction, the prediction task in this project is two-fold: predicting which devices are on during a test instance (i.e. label prediction) and estimating the energy consumption of each device based on these label predictions. To that end, let $\tilde{A} \in \mathbb{I}^{n_i \times n_d}$ be the output matrix containing label predictions. n_i is the number of instances in the sampled data and n_d be the number of devices and $\mathbb{I} = \{0, 1\}$. $\tilde{A}$ must be generated as the output of the classifier such that:

$$\begin{aligned}\tilde{A}_{ij} &= 1, \text{ if device } j \text{ is predicted to be `on' in instance } i \\ &= 0, \text{ otherwise.}\end{aligned}$$

We then use $y_i = int(\tilde{A}_i)$ to get the integer value of label for instance i and $\tilde{A}$ to estimate individual device energy consumption as follows.

Device Energy Consumption Estimation

The energy consumption $\tilde{E}_j$ of device j is estimated as:

$$\tilde{E}_j = \sum_{i=1}^{n_i} \frac{\tilde{A}_{ij} E_i}{\tilde{\eta}_i}, \quad \text{with} \quad \tilde{\eta}_i = \sum_{j=1}^{n_d} \tilde{A}_{ij} \tag{1}$$

where, E_i is the energy of the instance i. It should be noted here that due to the way $\tilde{\eta}_i$s are computed here, the energy of instance i is uniformly assigned to all the devices that are predicted to be 'on' by the classifiers. A more sophisticated approach would involve estimating these weights $\tilde{\eta}_i$ as well, that we anticipate would be much complex task.

4 Models

For this project we look to test the capability of different classifiers in a multi-class classification task. Our approach will involve following steps to solve this problem.

- Step 1: Pre-processing of data to create the features/instances/labels from the individual device and whole home energy consumption data.
- Step 2: Split the dataset in training and test data.
- Step 3: Train different classifiers using training data and evaluate accuracies of different classifiers.
- Step 4: Perform Step 1 on test data and use the best classifier from Step 3.

Central to the success of predicting device energy consumption is the accuracy of classification algorithm. The following sections will describe our baseline model, what is currently being used in research, as well as what algorithms we test.

4.1 Baseline Model(s)

Naïve Bayes Classifier: The Naïve Bayes Classifier is a generative model capable of handling multi-class classification problems. Due to its ease of use, this classifier was chosen as a baseline for testing. In order for the classifier to be used successfully, the features are 'binned' as shown in Table 4. The probability of each feature label being present given each classifier will calculated to generate the probability of the data set. This is given by the following equation,

$$\Theta_{ijk} = P(X = x_{ij} | Y = y_k) \tag{2}$$

To avoid the sparsity in the problem, we use add-one smoothing in the following equation to derive a maximum likelihood estimate;

$$\Theta_{ijk} = P(X = x_{ij} | Y = y_k) = \frac{\#D\left[X = x_{ij} \wedge Y = y_k\right] + 1}{\#D\left[Y = y_k\right] + n_x} \tag{3}$$

Table 4: Binning of features for Naïve Bayes classification

μ_p	σ_p	t_d	T	P_m	E	D_w
0 – 100	0 – 150	06 – 10	< 0	0 – 800	0 – 10000	1
101 – 200	151 – 200	10 – 15	0 – 10	801 – 1600	10001 – 20000	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
3901 – 4000	1351 – 1400	22 – 06	91 – 100	7201 – 8000	90001 – 100000	6
>4000	> 1500	06 – 10	> 100	> 8000	> 100000	7

4.2 Existing Models

As mentioned in the background, there exist variety of models in the literature to solve this machine learning problem. In this section we again briefly review the models that we found interesting.

NALM: Hart [1] uses the NALM technique to predict energy uses for all household devices [1]. Berges et al. adapts the NALM technique to determine the energy consumption for a refrigerator [2].

Hidden Markov Models: Kolter et al. works to predict device usage using a Factorial Hidden Markov Model [6]. Kim et al. proposes a new method based on Factorial Hidden Semi-Markov Models and Conditions Factorial Hidden Markov Models to predict device usage. This method combine the two and is named the Conditional Factorial Hidden Semi-Markov Model (CFHSMM) [5].

Support Vector Machines: Kolter et al. uses several different sparse coding techniques coupled with support vector machines to predict device energy consumption on the REDD data [3].

Regression: Kolter et. al compares Linear Regression and Guassian Process Regression in predicting energy consumption on large scale based on building features [8].

4.3 Proposed Model(s)

To facilitate the exploration of different classifers, we use the Python/scikit-kearn package. This package offers numerous tools to complete the tasks described above. As a preliminary study, we tried the Naïve Bayes classifier from scikit-learn package. The following section presents some algorithms available with scikit-learn that can handle multiclass classification.

Linear Discriminant Analysis: This classifier looks to fit a linear decision boundary. The boundary is generated by fitting densities to the data [11].

Support Vector Machines: The build in support vector machine package, SVC, utilizes a one vs. one scheme for classification [12].

Random Forest Classifier: The Random Forest Classifier builds decisions trees using a sample draw and replace method. The split for a decision tree is given by the best split of a randomly chosen subset of features [13].

Logistic Regression: The logistic regression package uses a one vs. all multi-class classification scheme. The classifier implements regularized logistic regression [14].

The proposed classifiers can handle continuous features and as a result the format of the feature vector changes slightly. Table 5 presents the change in type of each variable.

Table 5: Types of features used in this project

μ_p	σ_p	t_d	T	P_m	E	D_w
Continuous	Continuous	Discrete	Continuous	Continuous	Continuous	Discrete

The items that are presented as continuous are used directly from the energy data. The items presented as discrete follow the same binning procedure as used to prepare data for the Naïve Bayes Classifier

5 Experiments

5.1 Experimental Hypothesis

In this work we look to accomplish two tasks. The first is to determine which devicess are running at any given time from an whole house power profile. This information will be used to complete the second task, predicting how much energy each device uses. We propose that we can use machine learning algorithms to predict what devices are running in a supervised learning scenario. Research in the domain shows that predictions are near 50% accurate when using NALM type approaches.

5.2 Experimental Setup

Our experimental setup consists of publicly available power consumption data [6], and the weather data for the corresponding city [10]. We use this data to create our features required for the learning task (for further details please refer to Section 3). The ambient temperature feature is obtained directly from weather source files. The rest of the features are calculated as described in Section 3. The majority of our algorithm run-time is spent creating features from the original data. When analyzing the power data, it was noted that there was a "dead segment" in the data where the occupants of the home may not have been home. We filter out this data when developing testing and training folds. We use the off-the-shelf classifiers available from Python/scikit-learn package [15] for this learning task. The scikit-learn package provides tunable parameters for each of the algorithms. We manually tune the classifier parameters in an attempt to refine classification algorithm performance. Using the predicted labels from each of these classifiers, we then estimate the energy consumption of each device as previously described using Eqn. (1).

Metrics

In order to asses the classification accuracy two separate metrics are used. The first metric measures the classifiers' ability to exactly identify which group of devices are running at a given time.

$$\textbf{Lumped Accuracy} = \left(1 - \frac{\sum_{i=1}^{n_i} \mathcal{I}(y_i, \tilde{y}_i)}{n_i}\right) \times 100\% \tag{4}$$

here, y_i is the actual label of the instance i, $\tilde{y}_i$ is the predicted label and,

$$\begin{aligned} \mathcal{I}(y_i, \tilde{y}_i) &= 0, \quad \text{iff} \quad y_i = \tilde{y}_i \\ &= 1, \quad \text{otherwise.} \end{aligned}$$

The second metric used is to assess how many times each device is correctly classified. Given that the classifier output is a 9-bit binary vector (corresponding to 9 devices), the error can be evaluated by comparing a given column j (corresponding to device j) of actual and predicted labels as:

$$\textbf{Device } j \textbf{ Accuracy} = \left(1 - \frac{\sum_{i=1}^{n_i} |y_{ij} - \tilde{y}_{ij}|}{n_i}\right) \times 100\% \tag{5}$$

where, $y_{ij}, \tilde{y}_{ij} \in \{0, 1\}$ and n_i is the number of test instances.

5.3 Experimental Results

In this section we report two sets of results: 1) the label prediction task, where-in a classifier is expected to predict which devices are 'on' in a given instance, and 2) device energy consumption estimation using the predicted labels.

Label Prediction

To address the first task, each classifier is used to predict which devices are running at a given point in time. Table 6 presents the accuracy of the base model, Naïve Bayes, and best performing algorithm, Random Forest. The lumped accuracy and device accuracy results for both the classifiers from the five-fold cross validation are shown in the Table 6. The μ_e represents the average lumped accuracy of the classifier and σ_e represents the standard deviation of lumped accuracy on each of the classifiers for five-fold cross validation. Only the results from base classifier (Naïve Bayes) and proposed classifier (Random Forest) are shown in Table 6 for brevity. For all the other classifiers we tested, please refer to Table 8 in the Appendix for their accuracy results. The proposed classifier (Random Forest) has the mean lumped accuracy of 74.4% and the standard deviation of 2.5%, which makes it fairly robust for this classification task.

It should be noted that some of devices which are not frequently active, are predicted with high accuracy and the devices which are frequently active are often misclassified. For example, a refrigerator is almost always on with a near constant power consumption thus making its load signature difficult to identify. On the other hand devices like washer/dryer are active for short duration with higher peak powers, thus making them stand out in terms of their load signature and hence can be picked up by classifiers relatively easily.

Table 6: Classification results for a five-fold cross validation

	Lumped Accuracy %		Device Accuracy %								
Classifier	μ_e	σ_e	0	1	2	3	4	5	6	7	8
NaïveBayes	40.0	3.0	72.1	83.1	98.7	96.3	99.5	97.6	99.0	97.4	94.3
Random Forest	74.4	2.5	82.0	87.9	98.8	96.9	99.4	97.4	99.2	97.8	94.5

Device Energy Consumption Estimation

Using the device predictions from the first task, the energy consumption of each device is estimated. The Figures 4 and 5 show the estimated percentage of energy consumption of each device using the respective classifiers and comparing them with actual percentage of energy consumption of each device.

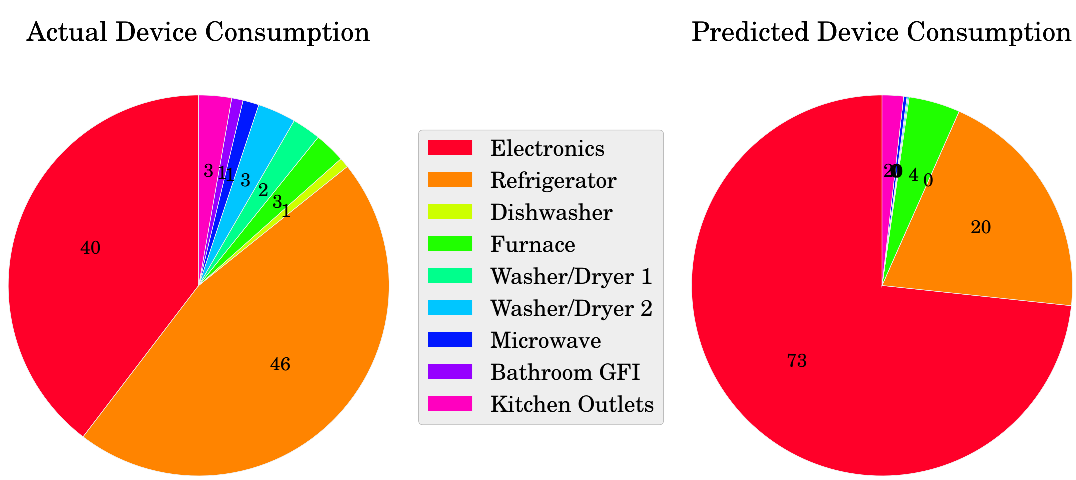

Figure 4: Individual device energy consumption (% of total consumption): prediction using Naïve Bayes Classifier

Our base case, the Naïve Bayes classifier, has a lumped accuracy of about 40% with a standard deviation of 3.0% in correctly labeling data. Moreover from Figure 4, it can be seen that this algorithm does a poor job of predicting the amount of energy consumed by each device. It assigns a very high percentage to electronics alone.

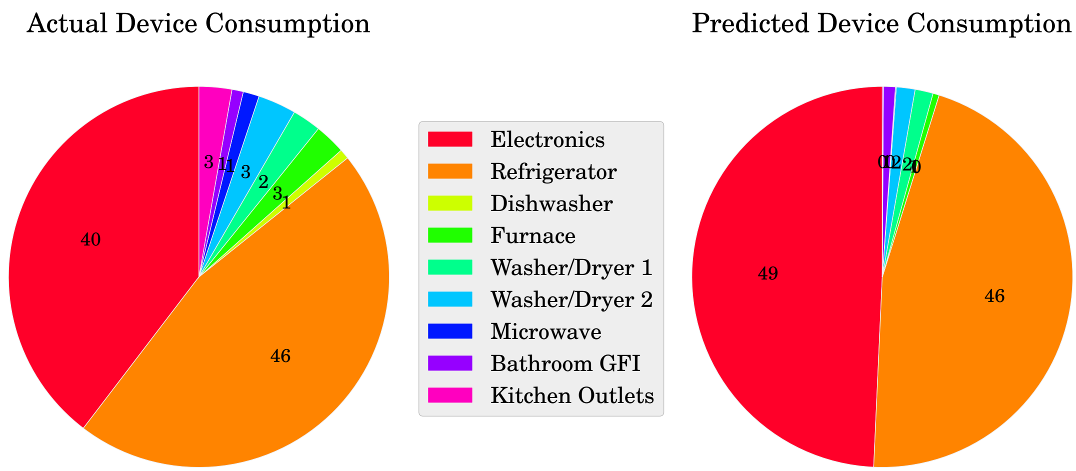

Figure 5: Individual device energy consumption (% of total consumption): prediction using Random Forest Classifier

Our proposed classifier, the Random Forest classifier on the other hand is reasonably accurate in labeling data. From Figure 5, it can be seen that this algorithm does a better job at predicting the energy consumption of each device. It reasonably estimates with a small error, the percentage energy consumption for electronics and refrigerator both. Another interesting result to look at is the ranking assigned by each of the classifiers to devices based on their estimated energy consumption. This result is shown in Table 7.

Table 7: Device energy consumption ranking (ascending from L to R)

Classifier	Device Order (Ascending L to R)								
Actual Ranking	2	7	6	4	3	8	5	0	1
Naïve Bayes	2	7	4	5	6	8	3	1	0
Random Forest	2	6	8	3	7	4	5	1	0

6 Conclusion

In this project report we presented the idea of electric load identification using supervised machine learning. This task was accomplished using six different classifiers out of which the Random Forest algorithm performed the best based on the classifier error analysis on the test data. The device energy estimation procedure using Random Forest classifier was able to estimate the individual device energy consumption with sufficient accuracy. This result is particularly promising since with further improvements, it can be directly used to provide the consumers with the feedback about which devices in the house are biggest energy sinks and thus helping them take proactive actions to save energy. An area for further improvement is to learn accurate weights for use in device energy consumption estimation. This will allow for a more accurate prediction of energy for devices that have relatively low load signature in the whole house power signal.

References

[1] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, 1992.

[2] M. E. Berges, H. S. M. Goldman, and L. Soibelman, "Enhanced electricity audits in residential buildings with nonintrusive load monitoring," *Journal of Industrial Ecology*, vol. 14, no. 5, pp. 844–858, 2010.

[3] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Neural Information Processing Systems*, 2010.

[4] M. Ziefman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.

[5] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *SIAM Conference on Data Mining*, 2011.

[6] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *SustKDD workshop on Data Mining Applications in Sustainability*, 2011.

[7] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation." La Palma, Ganary Islands: Proceedings of the 15th International Congress on Artificial Intelligence and Statistics, 2012.

[8] J. Z. Kolter and J. J. Ferreira, "A large-scale study on predicting and contextualizing building energy usage."

[9] C. Laughman, D. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong, "Advanced nonintrusive monitoring of electic loads," *IEEE Power and Energy Magazine*, pp. 56–63, March 2003.

[10] N. C. for Environmental Information, "U.s. local climatological data," April-May 2012.

[11] scikit-learn. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.lda.LDA.html#sklearn.lda.LDA

[12] scikit-learn. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

[13] scikit-learn. [Online]. Available: http://scikit-learn.org/stable/modules/ensemble.html#forest

[14] scikit-learn. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression

[15] scikit-learn. [Online]. Available: http://scikit-learn.org/stable/index.html

Appendix

Overall Results

Classification Results for All Classifiers

Table 8: Classification results for all the classifiers considered in this project

Classifier	Lumped Accuracy %	Device Accuracy %								
		0	1	2	3	4	5	6	7	8
Naïve Bayes	40.0	72.1	83.1	98.7	96.3	99.5	97.6	99.0	97.4	94.3
Logistic Regression	07.4	44.5	42.4	99.2	72.5	8.3	71.9	99.6	5.2	62.0
SVM	20.6	44.5	61.6	99.2	99.5	98.5	96.8	99.6	99.2	89.6
LDA	10.3	50.3	62.8	99.2	98.3	99.9	27.6	99.7	98.7	89.1
Random Forest	74.4	82.0	87.9	98.8	96.9	99.4	97.4	99.2	97.8	94.5

Table 9: Testing Data Energy Consumption Ranking

Classifier	Device Order (Ascending)								
Actual Ranking	2	7	6	4	3	8	5	0	1
Naïve Bayes	2	7	4	5	6	8	3	1	0
Logistic Regression	2	6	4	3	5	8	7	1	0
SVM	1	2	3	4	5	6	7	8	0
LDA	2	6	8	3	7	1	4	5	0
Random Forest	2	6	8	3	7	4	5	1	0