Machine Learning with domestic electricity use data: Prediction and Privacy

Sarah McGillion

April 3, 2014

Abstract

Smart meters are making their way into households across the country and there are concerns over the risks to personal information involved with having one in the home. This project aims to study the potential of using electricity readings from households to predict the number of residents. Modifying data from the Household Electricity Survey data, classifications based on the energy readings from 250 households using monthly or weekly collection intervals measured for 30 minutes or 1 hour were performed. The features and methods used in prior work are recreated on this new dataset. An exploration of features was performed to compare the performance of different consumption features for classification of residency numbers. Finally, the features were used for multi-class classification of residency numbers where the number of classes was increased to six to indicate the precise number of residents from 1 to 6+. Our evaluation showed that similar performance was achieved when using the recreated methods on our dataset. Simple consumption features could be used to predict the number of residents with up to 75% accuracy and increasing the complexity of the features used resulted in accuracy of up to 85% when performing two-class classification. Multi-class classification of household residency produces good results with the models and features used in general having around 60% accuracy but using certain consumption features accuracy of over 70% is possible

Acknowledgements

I would like to thank Dr. Nigel Goddard for all of his guidance and enthusiasm that helped me through this project. I would also like to thank Jonathan Kilgour who provided the database of information necessary for this project.

Contents

1 Introduction **5**
- 1.1 Introduction . . . 5
- 1.2 Smart Meters . . . 7
- 1.3 Related Work . . . 8
- 1.4 This Project . . . 8
- 1.5 My Contributions . . . 10

2 Data **11**
- 2.1 Overview . . . 11
- 2.2 Socio-Economic information . . . 11
- 2.3 Extracting the data . . . 12
- 2.4 Issues . . . 12
- 2.5 Comparison to previous work . . . 13

3 Feature Exploration **15**
- 3.1 The Features . . . 15
- 3.2 Time Periods . . . 15
  - 3.2.1 Measurement period . . . 15
  - 3.2.2 Collection Intervals . . . 16
  - 3.2.3 Half Hours . . . 16
  - 3.2.4 Hours . . . 16
  - 3.2.5 Days . . . 16
- 3.3 Features used to build input vectors . . . 16
  - 3.3.1 Recreated Features . . . 16
  - 3.3.2 Other Consumption Features . . . 17
  - 3.3.3 The use of averages and the standard deviations . . . 18
  - 3.3.4 Monthly Sum . . . 19
  - 3.3.5 Weekly Sum . . . 20
  - 3.3.6 Average Day and Standard Deviation . . . 21
  - 3.3.7 Average Hour . . . 23
  - 3.3.8 Average Half Hour . . . 24
  - 3.3.9 Day of the Week . . . 25
  - 3.3.10 Hourly . . . 27

3.4 Feature Selection . . . 28
3.5 Class Cardinality . . . 29
3.6 Balancing Classes . . . 29
3.7 Feature Preprocessing . . . 30

4 Models **31**
4.1 Overview . . . 31
4.2 K-Nearest Neighbour . . . 32
4.3 Support Vector Machine . . . 33
4.3.1 Multi class SVM classifiers . . . 34
4.4 Linear Discriminant Analysis . . . 34
4.5 Parametrisation of the Models . . . 34
4.5.1 Cross Validation . . . 34
4.5.2 K-NN . . . 35
4.5.3 SVM . . . 35
4.6 Evaluation Measures . . . 37
4.6.1 Two-class Evaluation Measures . . . 37
4.6.2 Multi-class Evaluation Measures . . . 38
4.7 Implementation . . . 39
4.7.1 KNN . . . 39
4.7.2 SVM . . . 40
4.7.3 LDA . . . 41

5 Results **43**
5.1 Overview . . . 43
5.2 Recreated . . . 43
5.2.1 Using the same features . . . 44
5.3 Exploration of Features . . . 45
5.3.1 Recreated Features . . . 46
5.3.2 Simple Features . . . 46
5.3.3 More Complex Featues . . . 48
5.3.4 More Complexity . . . 49
5.4 Observations From Two Class Classification . . . 51
5.5 Multi-class . . . 52
5.5.1 Recreated Features . . . 52
5.5.2 Simple Features . . . 53
5.5.3 More Complex Featues . . . 53
5.5.4 More Complexity . . . 54
5.6 Observations on Multi-class Classification. . . . 55
5.7 Classifier Comparison . . . 55
5.7.1 Two Class . . . 55
5.7.2 Multi-class . . . 55
5.8 Additional Interesting Findings . . . 56

5.8.1 Feature Selection Methods 56

6 Conclusion and Further Work **57**
6.1 Conclusion 57
6.2 Further Work 58

List of Acronyms

List of the Acronyms used in this project.

- **RWHH** - Recreated Week Half Hourly
- **RWH** - Recreated Week Hour
- **RMHH** - Recreated Month Half Hourly
- **RMH** - Recreated Month Hour
- **AD** - Average Day
- **AH** - Average Hour
- **AHH** - Average Half Hour
- **WS** - Weekly Sum
- **MS** - Monthly Sum
- **APD** - Average per day of the week, eg {the average reading for a Monday, average reading for a Tuesday, ... , average reading for a Sunday}
- **APH** - Average per hour of the day, eg {Average 00:00 readings, Average 01:00 reading,...,Average 23:00 reading}
- **APHH** - Average per half hour of the day, eg {Average 00:00 reading, Average 00:30 reading, Average 01:00 reading,...,Average 23:00 reading, Average 23:30 reading}
- **AHD** - Average per hour per day eg {Average 00:00 reading on Mondays, Average 01:00 on Mondays,...,,Average 23:00 reading on Mondays, Average 00:00 on Tuesdays,..., Average 23:00 reading on Sundays }
- **AHHD** - Average per half hour per day e.g.{Average 00:00 reading on Mondays, Average 00:30 on Mondays,...,,Average 23:30 reading on Mondays, Average 00:00 on Tuesdays,..., Average 23:30 reading on Sundays }
- **SAPD** - Standard deviation, Average Per Day
- **SAPH** - Standard deviation, Average Per Hour
- **SAPHH**- Standard deviation, Average Per Half Hour
- **SAHD** - Standard deviation, Average per Hour per Day
- **SAHHD** - Standard deviation, Average per Half Hour Per Day

Chapter 1

Introduction

1.1 Introduction

International promises have been made to reduce the carbon footprint of countries, and reducing energy use is one of the ways Governments will achieve these goals. One method that Governments have to encourage reduction in domestic energy use is the introduction of smart meters. These smart meters track energy use and encourage the end user to reduce their output by giving them a comprehensive breakdown of the way the energy is used. Energy companies have encouraged the introduction of smart meters as it allows them more accurate billings, less staff needed to travel to households to read meters and an ability to estimate load profiles of certain times and areas.

Previously personal energy data was only available in coarse granularity where energy readings were gathered monthly, or often quarterly. The smart meters will have the ability to constantly monitor the usage of households and relay that information to the interested parties for analysis. Data is collected in intervals of up to 10 seconds for electricity and 30 minutes for gas use.

With the advent of so much data becoming available, there is much interest in how this data can be used for statistical research. There is also potentially more interest in the potential risks that these smart meters may pose to households. There are concerns over the possible hidden risks of owning and having a smart meter in your home. There is a fear over the level of personal information attainable for these readings and to what extent the bubble of privacy is invaded. At what point is the appearance of privacy an illusion when so much data about the home is recorded?

Past research has attempted to discover the extent of the invasion of privacy if data from a smart meter were to be collected and analysed by trying to discover socio-economic information about households [1]. Others have suggested the nuisances of target advertising that can be achieved through energy readings [2]. Previous authors have raised serious questions about the risk to the individual that occur when the smart meter data is used such as a burglar knowing when the house is occupied [3] or the potential for stalkers to use this data to track their victims' movement [4].

This project has 3 main components:

1. **Recreate the features and methods of the work of Beckel** *et al* [1] and [5] to predict the number of residents.
   The work of Beckel *et al* uses energy data to predict socio-economic features about households, including categorising the number of residents of a household into 2 groups of few residents and many.
   This project aims to compare the results of these methods on a new and different dataset and showed that discovery of the number of residents is possible with comparable results.

2. **An exploration of features.**
   This was performed on a variety of consumption based features to gain an understanding of the kind of information that could lead to detection of the number of occupants in a household. Showing that the week total electricity reading is enough to predict residency with up to 75% accuracy when predicting two classes of residency numbers.

3. **An extension of the prediction of residency to a multi-class situation.**
   In this part of the project the number of classes of residents of a household is increased to six. This is to signify that this is a one person home, up to a 6+ resident household. The same features and similar models are applied when tackling the problem in a multi-class setting.

In breaking down the project into three tasks, a number of goals were achieved.

- Showing that the methods used by previous authors are suitable to be used on other datasets.

- Experiments with other consumption features enabled comparison of their performances on the level of information they can attain.

- Discovering how precise the personal information about households make up can be.

- Finding the exact number of residents of a household is an indicator of the extent of the risk attached to having a smart meter.

1.2 Smart Meters

The UK Government has a smart meter roll-out plan for household by the year 2020 [6]. They are following the example of other EU Countries including Italy, Sweden, Finland [7], Switzerland, and Germany [8] and are executing a plan to have them in all households in response to European Union demands and requirements [9].

The roll out of the smart meters to all households in the UK is aimed to begin in 2015 and be finished by the end of 2020 [6]. The role of smart meters will be to encourage consumers to take more control over their energy use and payments by providing them with a breakdown of costs and usage. This is an effort by Government to reduce the carbon-footprint of the country inline with international treaties. The information can be used by energy providers, third party and network operators. Examples of such uses are reading meters remotely, and sending real time energy information over a local communications network to a home display for customers [10]. The smart meters will be capable of taking readings every 30 mins and can hold a years worth of data on the device. There will be a Home Area Network which will send live information to home devices for consumers to see. In the future there will also be Wide Area Networks where the data can be sent to authorised third parties [11].

Smart meters send two kinds of energy use signals, firstly readings as fine as 5 seconds for electricity, and 15 minutes for gas, are sent across the Home Area Network (HAN) for devices attached to this network, including internal displays for end users to track their energy consumption [10]. Secondly, there are 30 minute readings send from the smart meters to the Data and Communications Company (DCC) via a Wide Area Network (WAN) [10], authorised users are allowed access to the DCC information for the households they have permission to. The DCC can query the smart meters for information on readings as far back as one year. It is unknown how strict the requirements are for an interested party to become an authorised user and how much personal information a party with access to this energy use data could find out.

There are concerns about the security of the data collected. There are worries about the type of personal information that could be attained from energy use data of households, whether by an authorised third party or an unauthorised one. Such personal information could be on what you own, what times you are in or out of the house, or whether there are children in the home. Data sharing with third parties is up to the customer, although it is not known how easy it will be for customers to know what data they are sharing or even how to stop it. Regarding the concerns about the safety of the signals being transmitted, if readings were able to be hacked by anyone, how much and how easily could they find out about your personal information? Users can limit the sharing of data, but is it an obvious process to stop the sharing of data to third parties? What about the security of the transmission itself?

As the 30 minute energy signals are sent out of the home area network and the information sent by this signals are collected into an information pool we have chosen this granularity for the work in our project, also using hourly readings rather than using the very fine grained small time step readings.

1.3 Related Work

The frequent readings will provide energy companies with large amounts of energy use data from households. Past studies have presented the benefits of this energy use data from smart meters not only for billing customers but also for various other reasons such as predicting peak-load times more accurately [12].

Non-intrusive appliance load-monitoring (NIALM) is an area of much active research which given household demand signal at fine grained intervals, attempts to determine which appliances were active and when [13]. Results of using NIALM approaches to discover personal data include the results from Lisovich *et al* [3]. In this study, readings were taken at 1 or 15 second intervals to find significant events that occur in households. The authors found that using this data collected over 3 days, or the larger 7 day dataset, that the presence of people in the home could be analysed with 90% accuracy. Other interesting events that could be observed from this fine grained data includes information on sleep/wake cycles, shower times, and when breakfast or dinner are eaten. With this sort of information a picture of your home life could be constructed and if the information were in the hands of a burglar it would be possible for them to find an opportune time in which to enter your property. This shows that live energy data streams could leave people at risk of personal information being discovered. As discussed above in Section 1.2, the smart meters are capable of logging energy use readings at less than 5 second intervals but it has been decided in this project to use 30 minute and 1 hour intervals as the energy use information is sent to the Data and Communications Company (DCC) across a Wide Area Network at 30 minute intervals.

Beckel *et al* [1] use supervised learning techniques to identify high level information about households including the size of a property, the age of a building, number of appliances and social class of the chief income earner The work of Beckel *et al* [5] identifies useful features for use in clustering households into different groupings. These include consumption figures, statistical information ratios and temporal data. When Beckel *et al* [1] attempt to use supervised learning methods to classify socio-economic information about households they use simple supervised classification techniques for their predictive modelling.

It is the work of Beckel et al [1] that we are replicating and extending in this project.

1.4 This Project

The aim of this project was to analyse the risk associated with the data collected from smart meters with a focus on classifying household residency. Using information collected by the Household Electricity Survey (HES), which was a DEFRA sponsored national survey of energy use collected over a period from 2010 to 2011, as our dataset we were able to complete the three goals mentioned above in Section 1.1. The work of Beckel *et al* [1] was reproduced on a new dataset for categorising the number of

residents in to two classes.

There was interest in recreating this project on a different dataset because we wanted to find out if the features and methods used by previous work produced similar results on a different dataset. Using features from Beckel *et al* [5] and three well known classification techniques we recreated the research in Beckel *et al* [1].

The models chosen to be used in this project were k-nearest neighbours [14], support vector machine [15] and linear discriminant analysis [16]. These models were chosen for many reasons including the general ease of implementation and their use in previous classification works. More details of the workings, advantages, and limitations of these models can be seen in Section 4.1.

The replication resulted in evidence that the previous experiments and results were replicable on other dataset. (The similarities of our HES [17] dataset and the Beckel *et al* "CER" [18] dataset are discussed in Section 2.5.) Resulting in over 80% accuracy of classification of homes into one of two classes of residency, ($few \leq 2$ or $many \geq 3$).

Once this had been established there was interest to compare the predictive power of these features with other features that we can calculate from the consumption data. Varying forms of consumption figures inspired from previous works, and through consideration of the type of information smart meters might emit were used. These ranged from simple features such as the total amount used in a month, or week to higher dimensional features such as the standard deviation and average energy use of each hour of each day of a week. This was an investigation of the information that could be gathered from features computed from energy signals.

Finally to asses the potential for more personal information the study of residency numbers was extended to a multi class problem where the exact number of residents was to be predicted. Using the same features and models as used in the two-class classification, with the exception of a modified SVM model which was extended to be used for multi-class purposes, it was found that the correct number of residents could be found with accuracy of up to 70%. This means that it might be possible for third parties to understand the composition of the house you live in just by the information gathered from smart meters.

The evaluation of the models in this project showed that when using two-class classification to discover the number of residents in a household it could be possible with accuracy up to 84% and a good measure of predictive power of the models, here F score is used to measure the performance of a model. When faced with multi-class classification it was more difficult to accurately classify the households into six separate classes, often achieving accuracy of around 60%, but it was found that with some consumption features accuracy of 70% was possible with a high performance of the models used.

1.5 My Contributions

This is a summarised list of the contributions I have made in this project.

- (Databases) Using the MySQL database provided I created a new database where each household was assigned its own table. I took the raw 2 minute or 10 minute per appliance per household readings and created tables for each household where the 2 minute or 10 minute sum of all appliances used in calculating the total profile were inserted.
- Using embedded SQL with python I wrote functions to create features based on the HES dataset, including recreating the types of features used by Beckel *et al* [1] [5] and various types of consumption based features.
- Implemented K-NN, LDA, and SVM (regular and 1vs1 for multi-class) in MATLAB using standard MATLAB Toolboxes. In the case of 1vs1, SVM multi-class model I used an online example of the implementation and changed it so that it performed 10 fold cross validation.
- Recreated the features and methods used by Beckel *et al* [1] [5] using the HES [17] dataset to perform two-class classification on household residency classes.
- Formed consumption features based on the energy readings from the HES [17] dataset and performed an exploration of their classification performances for two-class classification of residency numbers.
- Explored the performance of the features when applied to a multi-class classification situation of a six class problem of residency numbers.

Chapter 2

Data

2.1 Overview

The origins of the data used in this project come from The Household Electricity Survey (HES) [17] which was a nationally representative survey of energy use across homes in England between 2010 and 2011. This data was collected to give a picture of; the range and number of powered appliances in households, the pattern of usage, monitor total household electricity as well as appliance level monitoring and collecting habits of appliance owners.

The data has been edited for third parties by Cambridge Architectural Research Limited, they edited the data by taking out incorrect entires in the data, finding errors and either fixing them or removing them [19]. Such errors include the mislabeling of appliances with energy readings and normalising reading for time of the year.

The Household Electricity Survey monitored 250 households across England between 2010 and 2011. Twenty-six of these were monitored for one year and the remaining 224 were monitored for one month, with the months spread across the year. Households had between 13 and 85 appliances in the homes, with almost a third of them having between 30 and 40 appliances [19].

The households that took part in the survey had energy readings recorded in one of three ways; two minute intervals over a month, two minute intervals over a year or ten minute intervals over a year.

To calculate the total energy profile of a household the readings of particular appliances are summed together when the same time interval stamp matches, the particular appliances used to find the total profile were found by Cambridge Architectural Research Limited and vary from household to household.

2.2 Socio-Economic information

Along with information on appliance types, numbers and data readings there is information available on the occupants of the household. There is information on the

Experian social grade of the household, the occupation of the main income provider, information on if there are children present in the home and the number of residents of the household.

2.3 Extracting the data

The information from the HES was prepared into a MySQL database by Jonathan Kilgour for use by the IDEAL research group in The University of Edinburgh. Embedded SQL with python is how the information was extracted from the database so as to make features with the information.

All of the energy use data along with the appliance it was read from, the house that was measured and the time stamp attached to it were prepared into one MySQL database by Jonathan Kilgour. From this data, 250 new tables were designed by this project, one for each household, where energy readings that had the same date-time stamp, the same interval id (measured for 2 minutes over a month, 2 minutes over a year, 10 minutes over a year) and were included in the appliance list for total profile calculations were summed together. Once these base tables had been created, embedded MySQl in python scripts and functions were written that would sum these tables either hourly or half hourly, and different consumption measures were able to be calculated from these tables to provide the features as described in the Section 3.3.2.

2.4 Issues

The goal of this project is workout how many people live in a dwelling. This may be difficult because the models may not easily learn to distinguish households with certain occupancy, for example households with 4 and 5 people often have similar readings. Considerations about collapsing groups into one another were made, but through experimentation with models it was found that it was possible to differentiate between households of certain sizes.

There are only 250 households that were surveyed in the HES, this small number of households may not have enough information to accurately train and evaluate models. One solution to this was to split the 26 households measured for one year into 312 monthly household instances. Splitting these households resulted in 281 extra instances, as some houses were read for fewer than 12 months.

The data needed to be further segmented to work with weekly readings so as to recreate the methods used in Beckel *et al* [1]. When working with the weekly readings there were 1990 instances, and the class loads were not even so using accuracy alone as a measure would not be suitable to asses the performance of the work in this project. In order to balance classes when using weekly readings the number of instances with one, two, or four people were reduced to 340.

When working with weekly readings, classes became quite unbalanced especially when working with the two-class classification, (*few* vs *many*). Using current census data it was decided that during the modelling phase after outliers had been removed that the classes would be balanced on a 60/40 split in favour of few, as in England around 64 percent of households have one or two occupants [20]. The 60/40 split was chosen because if a model were to predict an energy reading as belonging to a *few* category class it would more often be correct in a real life situation, as dwelling occupancy changes throughout the country it was felt that a 3:2 split would provide more fair and accurate results.

A naive model that always predicted that instances belonged to the class *few* would still have an accuracy of 60%. To assess the performance of the models used in this project it was important to understand this limitation and to use other evaluation measures to assess the performance of the models, with the F score being used to measure the predictive power of the models used. When the data is broken into weekly readings it was decided that it would be interesting to discover the performance of the models on a larger dataset and this is why the project makes use of the 1300+ instances for weeks.

Occupants	Original	Expanded Data(Month)	Expanded Data(Week)	Balanced Data(Week)
1	70	125	454	340
2	87	214	814	340
3	30	41	146	146
4	49	104	391	340
5	8	19	73	73
6+	7	29	112	112
Total	250	532	1990	1351

Table 2.1: The number of instances

2.5 Comparison to previous work

As this project aims to recreate the work of Beckel *et al* [1] on this HES dataset it is worthwhile to note the similarities and the differences in the datasets used in this project and theirs.

The HES data is supposed to be a nationally representative survey of energy use across England [17]. The HES data was measured in periods of time between 2010 and 2011 where 26 households of the 250 surveyed were measured for one year and the others were measured only for a one month period. Ground-truth information on socio-economic information of the household was also collected, such information includes, the age of the building, the number of occupants, and the social grade of the household.

Beckel *et al* make use of the CER data set [18], a smart metering study conducted

by the Irish Commission for Energy Regulation.

This study used traces from 3,000 households measured at 30 minute intervals over 1.5 years, where ground truth information about socio-economic information is also provided. The data is supposed to be nationally representative for homes across The Republic of Ireland

The information provided is similar to the information collected by the HES. Beckel *et al* then identified households from the dataset and used only one week readings from the households [5] to identify interesting features and then use these features to predict socio-economic information on households [1]. The similarities in both data sets, such as the data being nationally representative, suggest that we may achieve similar results when using the methods of Beckel *et al* on our HES dataset.

Chapter 3

Feature Exploration

The features used in this project are focused around the recreation of past work [5] and by an exploration of other consumption based features. As such a variety of features are used and compared for use in determining the correct number of dwellers in a household.

3.1 The Features

Over 30 sets of features have been used in this project to predict the number of residents in a household. Consumption figures range from the total amount of energy used in a month or in a week to the per hourly average energy use per day of the week. There were many choices on how to build these features, such as using monthly or weekly collection interval data. Using hourly or half hourly energy reading intervals has resulted in the assortment of features described below in Table 3.2.

3.2 Time Periods

3.2.1 Measurement period

The length of measurement period for data to be used was the first choice for the project. Of the 250 households that have taken part in HES, 224 of them were measured for a period of a month. The remaining 26 were measured for up to a full year, and as such we can split these up into single monthly readings to acquire more data points.

As discussed in Section 2.5, previous work in this area has used week long periods of data measurement as the setting to take the data from. For a comparison of the results achieved in previous research this project has also used week long measurement periods. By separating the HES readings into week long readings 1990 instances were acquired, this resulted in unbalanced classes. The rebalancing of classes and methods used to do this have been discussed in Section 2.4.

3.2.2 Collection Intervals

Smart meters will have the capabilities of taking readings and sending them across a home network, as well as be able to send the information to energy providers and third parties every 30 minutes [10]. The devices will be capable of holding up to a years worth of information on the device itself [10].

3.2.3 Half Hours

It has been said by other authors that 30 minute time interval readings are appropriate for occupancy detection learning tasks [21]. The smart meters themselves are capable of taking readings every 30 minutes so there is a need to discover the information that can be found using this time interval. To provide further comparison between this work and that of previous research this project has used half hour readings in the making of features for the models.

3.2.4 Hours

Testing models using more coarse grained time intervals is an interesting step to discover how much data is needed and how fine the data needs to be. Features used include the average hour's energy use, and even the average per hour to discover if the changes per hour are more of an indication of residency numbers than just simple features.

3.2.5 Days

Daily readings are also used in this project to discover the performance of models that use daily consumption figures as features. For example we use the average day (AD) as a feature, it is a one dimensional feature that contains the average day's energy readings. It is of interest to discover if knowing this information will lead to a good classification of residency numbers. Also used are the average readings per day of the week, this is to explore the differences in energy usage depending on the day of the week to discover if there is a noticeable change in electricity use habits of a household.

3.3 Features used to build input vectors

3.3.1 Recreated Features

The following features in Table 3.1 represent the features as described in [5] and used for classification in [1]. The features are able to be separated into four groups: Consump-

tion figures; Ratios; Temporal Properties; Statistical Properties. With the temporal features it was found that the first time the mean was above 1 or 2 kW was similar among most households in our HES data set and so these were not seen to be useful and were not used in modelling.

We first use weekly 30minute interval readings on our data to compare the results achieved on our dataset to that of the previous work. We then also use weekly and monthly collection intervals with 30 minute or 1 hour energy readings to compare the results across all features used in this project.

$\bar{\text{P}}$ denotes the 30 minute,or 1 hour, mean power samples provided by the data.

Statistical properties		Temporal Properties
Variance		First time $\bar{\text{P}} > 1$kW
$\Sigma(\lvert\bar{P}_t - \bar{P}_{t-1}\rvert)$		First time $\bar{P} > 2$kW
Cross-correlation of subsequent days		First time $\bar{P}$ reaches maximum
# $\bar{P}$ with $(\bar{P}_t - _{t\pm1} > 0.2kW)$		Period for which $\bar{P} >$ mean
Consumption figures		Ratios
$\bar{P}$(daily)	c_day	Mean $\bar{P}$ over maximum $\bar{P}$
$\bar{P}$(daily, weekdays only)		Minimum $\bar{P}$ over mean $\bar{P}$
$\bar{P}$(daily, weekend only)		c_night/c_day
$\bar{P}$ for (6 p.m. - 10 p.m.)	c_evening	c_morning/c_noon
$\bar{P}$ for (6 a.m. - 10 a.m.)	c_morning	c_evening / c_noon
$\bar{P}$ for (1 a.m. - 5 a.m.)	c_night	
$\bar{P}$ for 10 a.m. - 2 p.m.	c_noon	
Maximum of $\bar{P}$		
Minimum of $\bar{P}$		

Table 3.1: Recreated Features. Features and descriptions taken from Beckel *et al* [5]

3.3.2 Other Consumption Features

The following features are created using week long and month long time measurement intervals. Where relevant they are created using half hourly readings or one hour readings.

Acronym	Name	Description
MS	MonthlySum	The total amount of energy used in a month per household
WS	WeeklySum	The total amount of energy used in a week per household
AD	Average Day	The average energy use of a day
ASD	AvgStdDay	The average and standard deviation of energy use in a day
APD	AvgPerDay	The average energy reading per day of the week
SAPD	StdAvgPerDay	The average and std energy reading per day of the week
AH	AvgHour	The average energy use in an hour
ASH	AvgStdHour	The average and standard deviation of energy use in an hour
AHH	AvgHalfHour	The average energy use in a half hour
ASHH	AvgStdHalfHour	The average and standard deviation of energy use in a half hour
AHD	AvgHourDay	Average reading per hour per day of the week
SAHD	StdAvgHourDay	Average and std reading per hour per day of the week (monthly readings only)
AHHD	AvgHalfHourDay	Average reading per hour per day of the week
SAHHD	StdAvgHalfHourDay	Average and std reading per hour per day of the week (monthly readings only)
APH	Average Per Hour	Average energy use per hour of the day
SAPH	Std Avg Per Hour	Standard Deviation and average energy use per hour of the day
APHH	Average Per Half Hour	Average energy use per half hour of the day
SAPHH	Std Avg Per Half Hour	Standard deviation and average energy use per half hour of the day.

Table 3.2: Own Features

3.3.3 The use of averages and the standard deviations

Averages were used in the features to determine what the normal behaviour of a household might be and to see if other households with similar occupancy act in a similar fashion.

The average can take away the variability of the data and might lose some interesting information regarding the behaviour. Adding in the standard deviation allowed the reintroduction of some of the information that is lost.

3.3.4 Monthly Sum

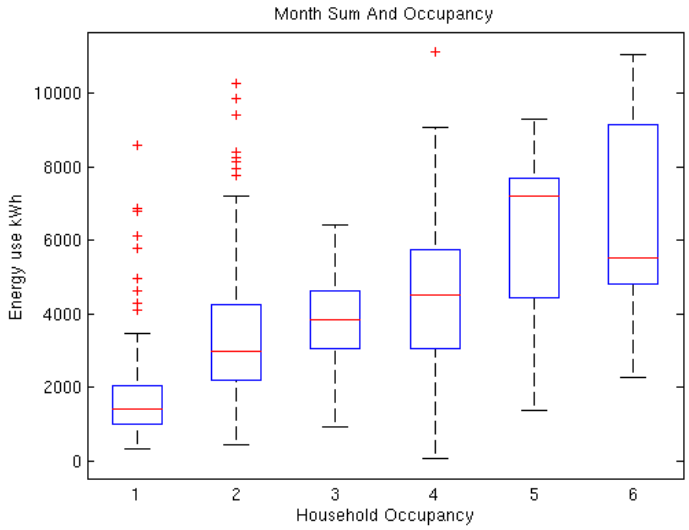


Figure 3.1: Total Montly Energy readings vs Occupancy. The vertical axis shows total energy use in kWh. The number of classes of occupants in a household is shown on the horizontal axis and ranges from 1 to 6 to indicate classes that have 1 occupant up to 6+ occupants.

We can see from the box plot in Figure 3.1 that the total readings in a month are not the best indicator of total household occupancy. Households with 3 or 4 occupants have readings that overlap with each other so the models may not be able to predict with great accuracy the difference between the two classes.

3.3.5 Weekly Sum

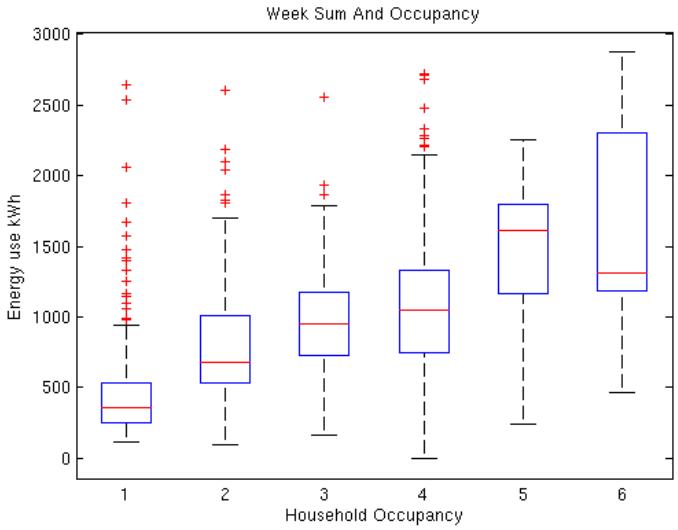


Figure 3.2: Total Weekly Energy readings vs Occupancy.The vertical axis shows total energy use in kWh. The number of classes of occupants in a household is shown on the horizontal axis and ranges from 1 to 6 to indicate classes that have 1 occupant upto 6+ occupants.

In Figure 3.2 we see that once again total energy reading may not be useful for multi-class classification as proposed in this project. Households with 2, 3, or 4 residents behave in a very similar way and have readings that overlap and models may confuse these three groups with each other.

3.3.6 Average Day and Standard Deviation

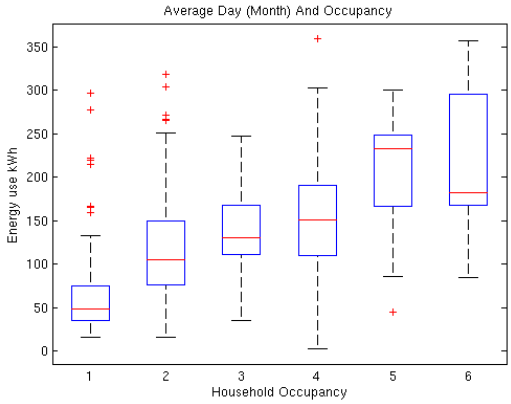


Figure 3.3: AD Measured Monthly

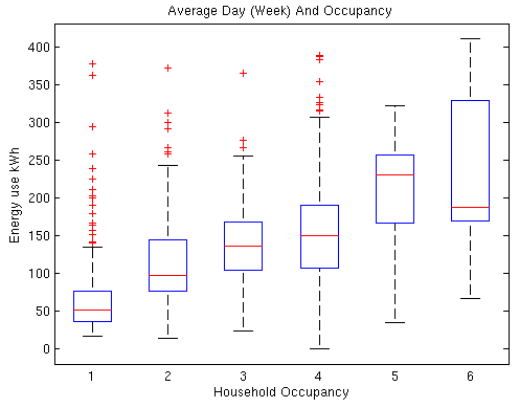


Figure 3.4: AD Measured Weekly

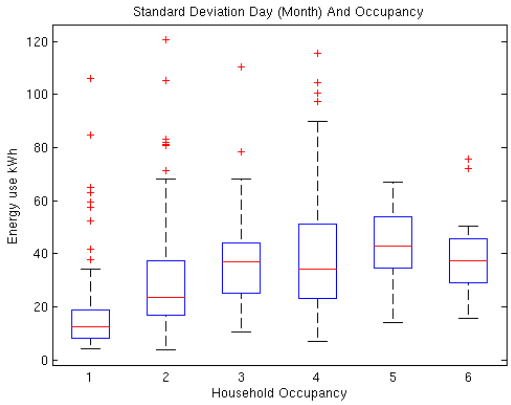


Figure 3.5: ASD Measured Monthly

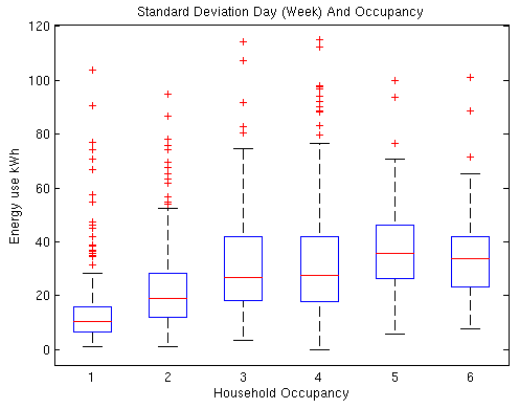


Figure 3.6: ASD Measured Weekly

The vertical axis in each of Figures 3.3, 3.5, 3.4, and 3.6 shows the average energy use in kWh. The horizontal axis indicated the number of resident of a household.

In Figure 3.3 we can see that households belonging to class 1, with one occupant in the household, can be easily distinguished from other households. When we consider the average day (AD) readings measured in the weekly dataset in Figure 3.4 we find that classes with 3 or 4 occupants have similar readings. When we view the standard deviations of the average day readings, Figure 3.5, in the monthly dataset we find that the classes with more than 2 occupants have the similar deviation from the mean, suggesting that the standard deviation alone may not be useful for classification. In

Figure 3.6 we see the standard deviation from the average day readings per class can once again be similar, classes with 2 or more residents appear to have the same changeability in average energy patterns. This set of features was a step in the right direction, although it has not performed best among all the features tested here we will see further in Chapter 5 that the performance of this type of feature is superior to the features we have discussed so far.

3.3.7 Average Hour

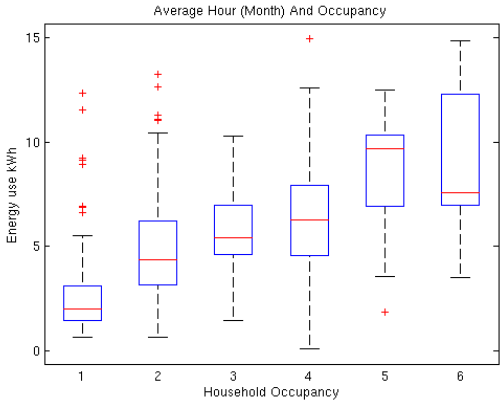


Figure 3.7: AH Measured Monthly

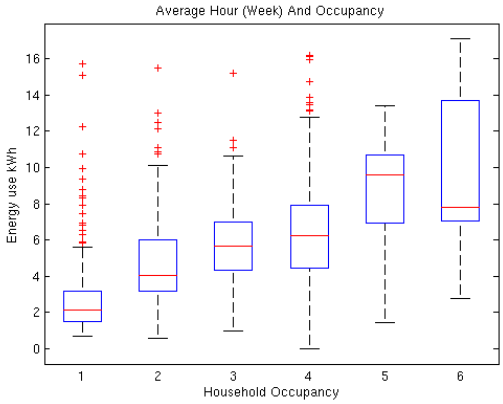


Figure 3.8: AH Measured Weekly

In Figures 3.7 and 3.8 the vertical axis shows the average energy use in kWh. The horizontal axis shows the number of the class.

Figures 3.7 and 3.8 depict the average hour's electricity use in the monthly dataset and weekly dataset respectively. In Figure 3.7 we can see there is a difference in the average hour's electricity use in households and we can clearly see more of a difference in the behaviour of classes than we have before. Class 1 is easily identifiable as different from the others. Class 5 is more distinct from classes 3 and 4 than when for example using the standard deviation for the average days usage in Figure 3.5 before. The average hour energy use in the weekly class, as seen in Figure 3.8, shows that there are many more outliers when measuring the average hourly reading in the bigger weekly dataset, this could be an indication that if the dataset size were to increase that the use of average hour energy use data may not be appropriate for multi-class classification.

3.3.8 Average Half Hour

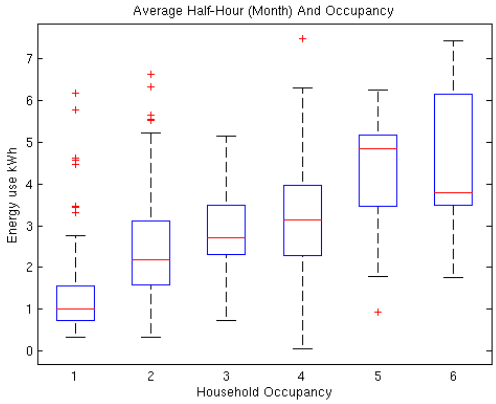


Figure 3.9: AHH Measured Monthly

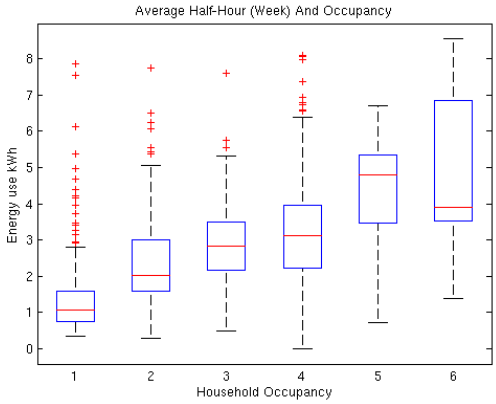


Figure 3.10: AHH Measured Weekly

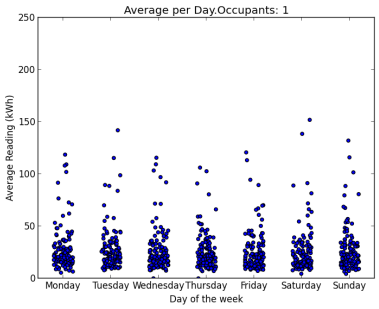
In Figures 3.9 and 3.10 the vertical axis shows the average energy use in kWh. The horizontal axis shows the number of the class. Figures 3.9 and 3.10 depict the average half hour's electricity use in the monthly dataset and weekly dataset respectively.

We can see from Figure 3.9 that homes with 4 or 5 occupants can have similarities in the amount of energy used on average in a half hour, but when we compare this to the weekly dataset as shown in Figure 3.10 we see that when we have more data we can separate the classes with four or five residents more easily.

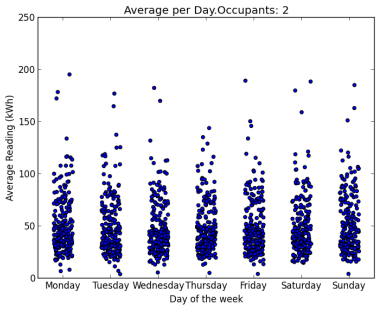
The average hourly reading and average half hourly readings show similar separation of classes, (although they are of different scale on the y-axis), suggesting that the using hourly or half hourly data could provide similar results.

3.3.9 Day of the Week

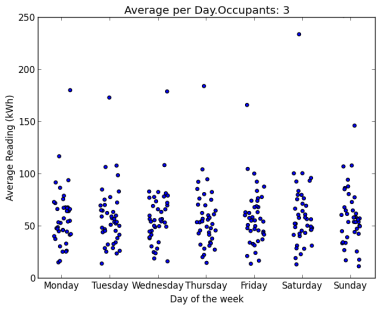
Average



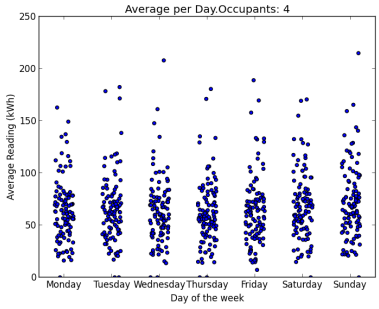
(a) Average Per Day vs Occupancy 1



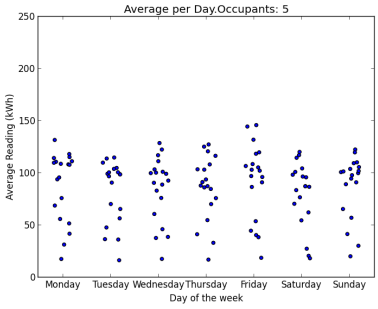
(b) Average Per Day vs Occupancy 2



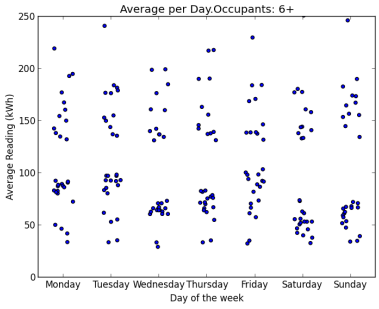
(c) Average Per Day vs Occupancy 3



(d) Average Per Day vs Occupancy 4



(e) Average Per Day vs Occupancy 5

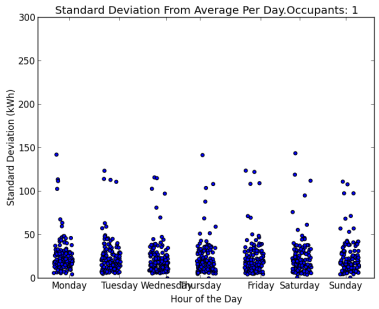


(f) Average Per Day vs Occupancy 6

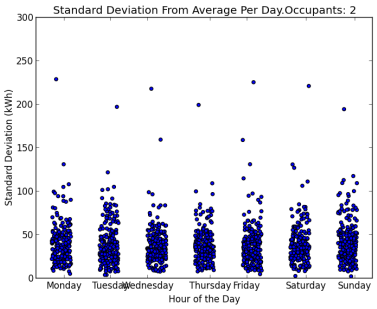
Figure 3.11: Daily Averages and Occupancy. The vertical axis shows the average energy use in kWh. The horizontal axis shows each day of the week from Monday on the left to Sunday at the end on the right. For each class a graph has been produced. In the figures the values for housholds have been jittered across the horizontal axis for ease of readings and detection of difference between classes. These are created from the monthly dataset

Scatter plots are now used to show more of the in-class differences and similarities. Using the average per day of the week it is hoped to discover is the day of the week changes the average energy consumption of a household.The average daily readings show a consistency that household tend to have every day with average readings remaining at similar levels regardless of whether it is a weekend or weekday. This indicates that if information we to be gathered and some form of daily feature were to be used, it would be unnecessary to know what day of the week it is from.

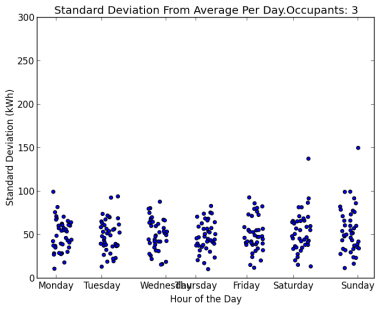
Standard Deviation



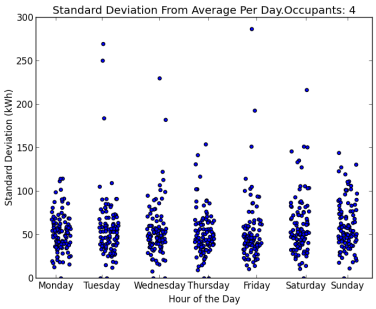
(a) Standard Deviation from Average Per Day vs Occupancy 1



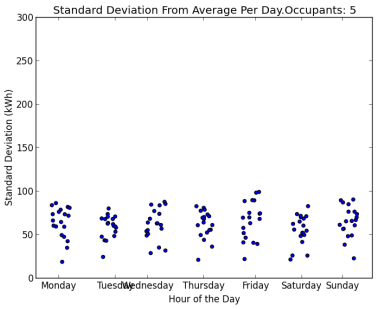
(b) Standard Deviation from Average Per Day vs Occupancy 2



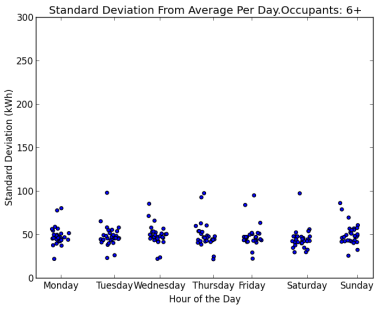
(c) Standard Deviation from Average Per Day vs Occupancy 3



(d) Standard Deviation from Average Per Day vs Occupancy 4



(e) Standard Deviation from Average Per Day vs Occupancy 5



(f) Standard Deviation from Average Per Day vs Occupancy 6

Figure 3.12: Standard Deviation from the Average vs Occupancy. The vertical axis shows the average energy use in kWh. The horizontal axis shows each day of the week from Monday on the left to Sunday at the end on the right. For each class a graph has been produced. In the figures the values for housholds have been jittered across the horizontal axis for ease of readings and detection of difference between classes. These are created from the monthly dataset

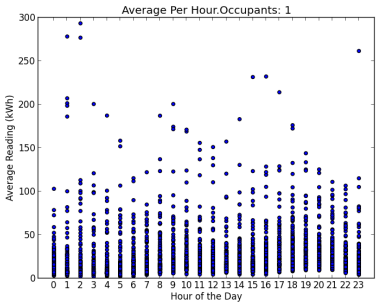
The standard deviation from the mean for daily readings displays that generally households consume similarly day to day.

Comments on days of the week

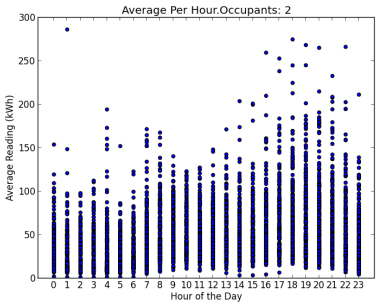
These graphs are an indication that daily readings will not result in accurate predictions because of the similar consumption behaviour of households day to day.

3.3.10 Hourly

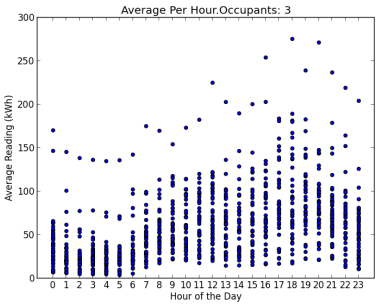
Average



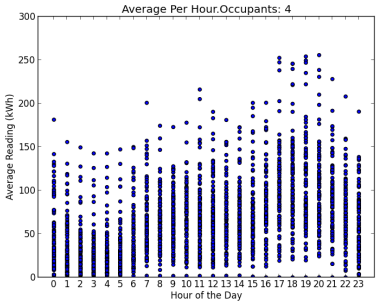
(a) Average Per Hour vs Occupancy 1



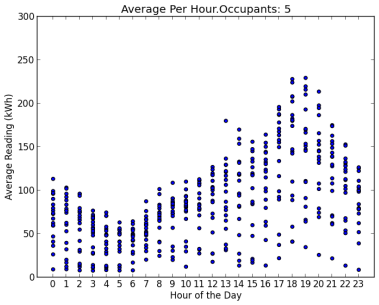
(b) Average Per Hour vs Occupancy 2



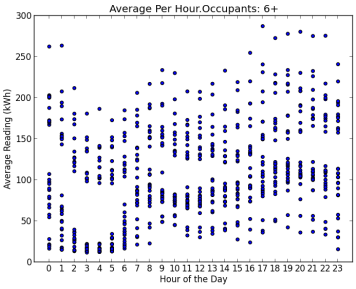
(c) Average Per Hour vs Occupancy 3



(d) Average Per Hour vs Occupancy 4



(e) Average Per Hour vs Occupancy 5

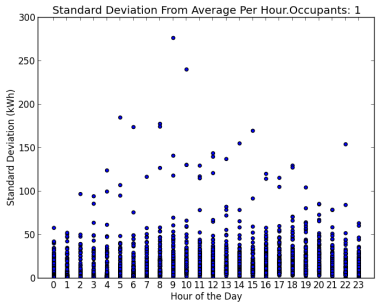


(f) Average Per Hour vs Occupancy 6

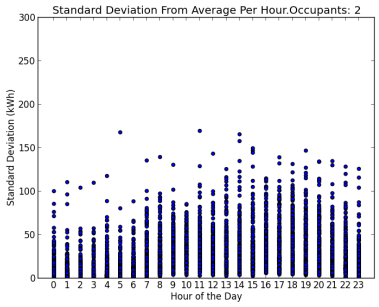
Figure 3.13: Hourly Averages vs Occupancy.The vertical axis shows the average energy use in kWh. The horizontal axis shows each hour of a day from midnight on the left to 11pm on the end right. These are created from the monthly dataset

We can see from these figures that the average per hour has more fluctuation at different times of the day and that the average for a particular set of households identify them as being part of a grouping. For example the overall average per hour for households with one occupant will be much lower than the overall average for households with two occupants, so these two groups have more potential for distinction from each other. Households with 3 occupants on the other hand tend to behave more like single occupancy in the earlier parts of the day (from midnight to 6am), but then more like a two person occupancy later on in the day.
Taking these into consideration it might be more worthwhile looking at certain parts of the day like waking hours or times of more regular activity.

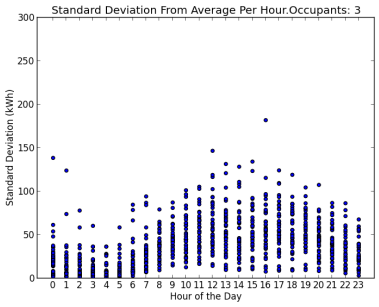
Standard Deviation



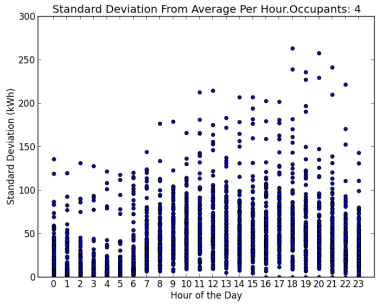
(a) Standard Deviation from Average Per Hour and Occupancy 1



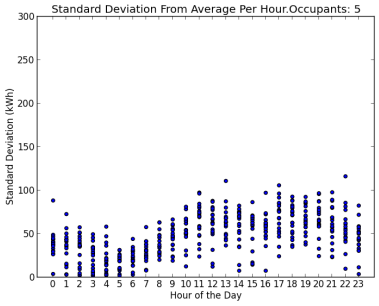
(b) Standard Deviation from Average Per Hour and Occupancy 2



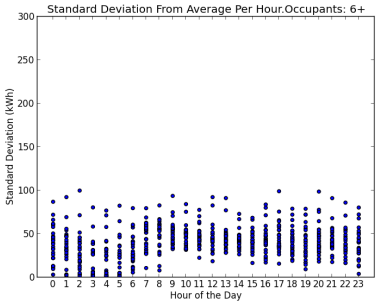
(c) Standard Deviation from Average Per Hour and Occupancy 3



(d) Standard Deviation from Average Per Hour and Occupancy 4



(e) Standard Deviation from Average Per Hour and Occupancy 5



(f) Standard Deviation from Average Per Day and Occupancy 6

Figure 3.14: Standard Deviation from the Average Hourly readings vs Occupancy. The vertical axis shows the average energy use in kWh. The horizontal axis shows each hour of the day from midnight on the left to 11pm on the end right. For each class a graph has been produced. These are created from the monthly dataset

Comments on Hours of the day

It can be seen from these observations that using hours of the day will be more informative to the predictions than days of the week because of the variability of the data and how households behave similarly within their groupings.

3.4 Feature Selection

Due to the large amount of features computed some of the vectors will be of high dimension. High dimensional feature data can lead to suboptimal performance of classifiers [22].

The first step in reducing the dimensionality is to use feature selection methods.

Feature selection aims to reduce dimensionality by removing irrelevant and unneeded data and increasing the learning accuracy of models.

Particularly we will focus on:

Fast Correlation Based Filter (FCBF) [23] [24] works by using normalised mutual information to select the best features. It has been shown to be effective in removing both irrelevant and redundant features.

Correlation-based Feature Selection (CFS) [25] uses correlation based methods and best first search to find the features that will be most useful in classification of the classes provided. It does this by evaluating the predictive power of individual features in a subset.

The fisher score is a method for determining the most relevant features for classification [26]. Using discriminative methods and generative statistical models the most relevant features are found. The fisher score is used here in this project to decide the top 30 percent of features in the data and to use these features in the modelling.

These three feature selection methods are used to reduce the number of features in the input vectors.

Implementations of these feature selection methods are available from Arizona State University (ASU) in association with DMML (Data Mining Machine Learning research group in ASU) [27], where a MATLAB toolbox has been made and is available for free under Gnu Public Licence.

3.5 Class Cardinality

The occupancy problem can be modelled in different ways depending on how to split the classes. To prove that the results are comparable this has been modelled as a two class problem, either few people in the home (≤ 2) or many (≥ 3). All of the features above will be used in two-class classification.

The interest of this project was to discover the import features required for occupancy number detection and as such the importance of modelling this as a multi-class classification problem is apparent. The above features will be used in multi class classification with classes 1, 2, 3, 4, 5, and 6+ representing houses with one, two and so on occupants.

3.6 Balancing Classes

Due to the method of expanding the data instances, through to expanding yearly readings into months and weeks, the imbalance of classes became extreme to a point where a models accuracy could be high by estimating the input to be of one particular class

(particularly a problem when the classes are binarized).

According to current census data [20], 30 percent of households in the UK have 1 occupant, a further 34 percent are occupied by only 2 occupants. The data was allowed to be unbalanced when there was two-class classification in the ratio 3:2 in favour of few houses as this is how the ratios fall in real life. The limit of 60 percent was chosen as obviously this rate of one and two occupied households is for the UK as a whole and there is a chance these numbers will vary throughout the country.

3.7 Feature Preprocessing

Outlier removal is used in this project. Outliers were defined as entries that were more than 3 standard deviations from the mean. A function was written to remove outliers. The function removes input features based on the number of outliers in the vector and the dimensionality. For example if the dimension of the input vector was less than 10 then it was only required to have one entry that was seen as an outlier to be removed. If on the other hand the dimensionality of the vector was more than 10 it was required that there be 2 or more outliers as entries. Outlier removal is an important part of the data clean up before the classification techniques are applied, K-NN can be sensitive to outliers where a mislabelled entry can make a large change to the decision boundaries. As such removing the outliers is an important step to improving performance.

Another method of preprocessing features used in this project is the standardization of the input features. Feature standardisation normalises features to have zero mean and unit variance. Feature standardization is often a recommended step in the use of SVM [28].

Chapter 4

Models

4.1 Overview

The models chosen for classification purposes in this project have been chose to be K- Nearest Neighbours [14] , Support Vector Machines [15] and Linear Discriminant Analysis [16]. As discussed in Section 1.4 these models have often been used for similar classification tasks and work well with this project because of the success of the models in previous work and the ease of experimentation with the models parameters. Many different modelling environments had been considered for use in this project, including WEKA [29] which is a software suite developed by The University of Waikato for use with machine learning tasks, while WEKA is freely available and can be fast and simple to use, the libraries and algorithms can often be out of date and more current methods are available elsewhere.

Another choice was scikit-learn [30], a python library for machine learning that is also freely available. It is an open source collection of a full set of machine learning tools including feature selection methods, regression and classification techniques. This was considered for use in the project but as scikit-learn does not have a vast use in academic research yet another software suite was chosen instead because of its growing popularity within industry and research to be used for machine learning.

All the models in this project have been implemented in MATLAB [31] (MATLAB 2013b) as MATLAB has gained much momentum in the world of machine learning among academics and many freely available toolboxes have been developed by universities for machine learning tasks. MATLAB has many purpose built toolboxes for machine learning and statistical analysis and there is much support available as well as many books and tutorials on the methods and the implementations of machine learning tasks.

4.2 K-Nearest Neighbour

K-Nearest Neighbour Algorithm (K-NN) is a non-parametric method that can be used in classification or regression tasks. This means that the model does not make assumptions on the underlying distribution of the data. It is also known as a lazy algorithm as there is no explicit training phase and all the training data is used in the testing phase. This makes K-NN computationally cheap and quick to run. K-NN has often achieved great results with multi-class classification methods and although it is a simple algorithm it has displayed it's effectiveness for resident number detection. This project is a class label based problem and hence the K-NN classifier is employed. K-NN uses a distance metric to decide the class a sample is most likely to be, a sample is classified by the most common class among it's K neighbours.

There are three parameters in the K-NN algorithm; the distance measure, the search method and the number of neighbours. The distance metric used in this project is Euclidean Distance. It is possible to use the Euclidean distance without much concern on bias of vector sizes because the vectors are normalised to have zero mean and unit variance, details of this normalisation are discussed in Section 3.7. It is a well used similarity measure in classification tasks similar to the task presented in this project. The search method chosen has been a K-D Tree.

Due to the large number of features used, an adjustment to the standard K-NN has been used. This is the use of a Nearest Neighbour Search Parameter. This proximity parameter aims to reduce the number of distance evaluations performed by K-NN and makes K-NN more computationally sound as the number of distances computed and stored in memory are reduced.

The K-NN implementation in this project uses K-D trees as search method [32]. K-D trees work by repeatedly bisecting the search space into regions.

These metrics can be varied in experimentation, but for this project these are the settings that have been chosen. The third parameter is the best number of neighbours used. The method the choosing the best number of neighbours is discussed below in Section 4.5.

Figure 4.1 shows an example of how the classification of an instance can change with the number of neighbours used to classify the points.

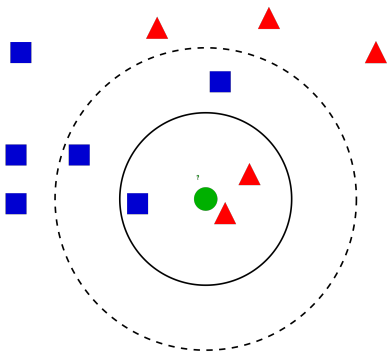


Figure 4.1: KNN

4.3 Support Vector Machine

Often considered a state-of-the-art classifier [33] , this model can perform well even when given a small training set. The standard Support Vector Machine (SVM) classification model as proposed by Cortes and Vapnik [15] is between two classes. It is a binary classifier that finds the optimal boundary between the two classes, called support vector. The algorithm attempts to maximise the margin between the boundary and the support vectors if the data is linearly separable leading to similar results as can be seen in Figure 4.2. In the event that the data is not linearly separable other kernel methods may be applied.

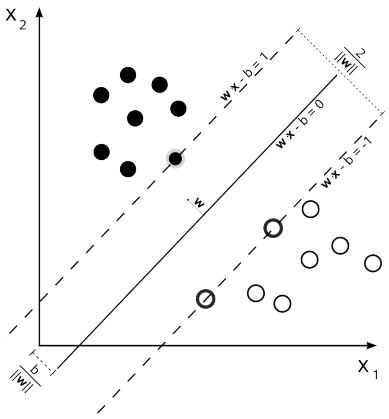


Figure 4.2: Linear SVM

A major draw back of SVM is that they have a computationally expensive training phase. As explained later in Section 4.5 on parameter optimisation, the computational cost and time necessary to run the SVMs resulted in linear kernel choice, which may not be the most optimal for this project.

4.3.1 Multi class SVM classifiers

The standard SVM algorithm is a binary classifier and as such when used in this project for multi-class classification of households into the 6 different categories of residency labels another implementation was required. There are many ways to implement multi-class SVM classification the two most frequently used are 1-vs-the rest implementation and a 1vs1 approach.

The first works by construction of as many models as classes and then checking through them to find the correct classification. Each model is trained taking all examples from one class as positive and all examples from the other classes as negative then a function is outputted and the classifier with the highest function value for a sample is decided to be the class to which it belongs [34] .

The 1vs1 approach works by constructing one binary classifier for each pair of distinct classes. There will be a total of M(M-1)/2 classifiers constructed. In our case there will be 15 models constructed. A new sample is tested in every one of the M(M-1)/2 classifiers and a score is awarded to a model if the sample is to be classified as that positive, else the negative class gets a score of one. Once all iterations are finished the sample is assigned to the class with the highest score [34].

In this project the 1vs1 approach has been implemented. It has been shown in practice to outperform 1vs rest implementation and may be more suitable for practical use [35].

4.4 Linear Discriminant Analysis

Discriminant analysis is a classification technique that makes the assumption that classes generate data based on different Gaussian distributions [16].

Linear Discriminant Analysis is used to find a linear combination of features which separate classes, LDA attempts to model the difference between classes and once the features have found the features a linear classifier may be used.

When using the LDA classifier in this project it is assumed a the data is based on a Gaussian Distribution, this means that the discriminant function parameters which partition the feature space do so based on the mean and covariance of the distributions for each class. This is a major drawback of using LDA. The upside of using LDA in this project is that is computationally cheap and quick to run and has shown comparable results to the other methods used in this project as explained in the Chapter 5.

4.5 Parametrisation of the Models

4.5.1 Cross Validation

Cross-validation is a model validation technique. It use to analyse how well a model will perform on an unseen data set [14].

Cross-validation is a method used to predict the performance of a model to a sample validation set when there is not a validation set readily available. This project makes use of cross-validation due to the small HES data set being used. The whole data set can be used for training and testing rather than separating the dataset into distinct sets.

When using k-fold cross validation the original input data is randomly partitioned into k equal subsamples, k-1 of these subsamples are used as training data and one of the subsamples is used as a validation sample. Cross validation is repeated k times so that each sub sample is used to validate exactly once. The results from these k runs are averaged and a single performance value can be returned.

4.5.2 K-NN

The number of neighbours parameter need to be optimised for the classification task of this project. Large values of K are used to reduce the affect of outliers in the data while lower values of K can lead to overfitting and error due to outliers in the data. The optimal value of K for each set of features is calculated using cross validation to find the best value for K that gives the overall lowest error measure on the data.

4.5.3 SVM

Linear Classification

Linear SVM used when the data is linearly separable to maximise the distance between different classes. Linear SVM classification is less likely to overfit the data compared to Non-linear SVM.

Non-Linear Classification

When the data is not linearly separable a Non-linear kernel can be used with SVMs to map the data to a linearly separable feature space.

Kernel Choice

Kernel methods are a way of transforming data into high dimensional feature space to extract nonlinearity of data. Kernel methods map data to higher dimensional feature space, kernel methods use kernel features and this operation is computationally cheaper than using explicit methods, this approach has been called the kernel trick.

1. Linear $u' * v$.
   Planes separate the classes

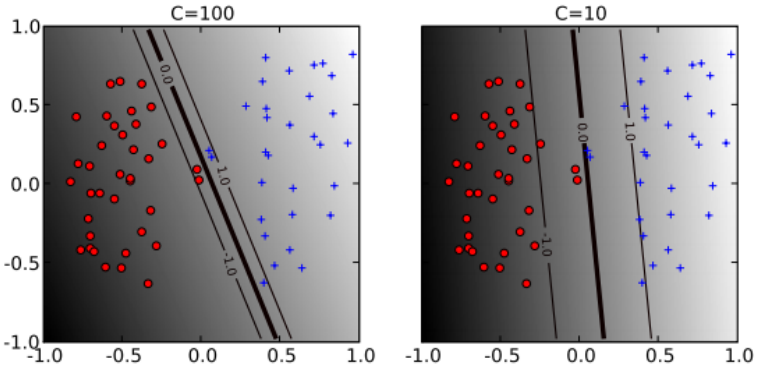


Figure 4.3: SVM Linear Kernel,change in parameter C, the boundary parameter. The effect of the change of the soft-margin constant C on the decision boundary. Taken from Ben-Hur and Weston [28]

2. Radial Basis Function $exp(-gamma * |u - v|^2)$
   Gamma controls the curvature of the hyperplane

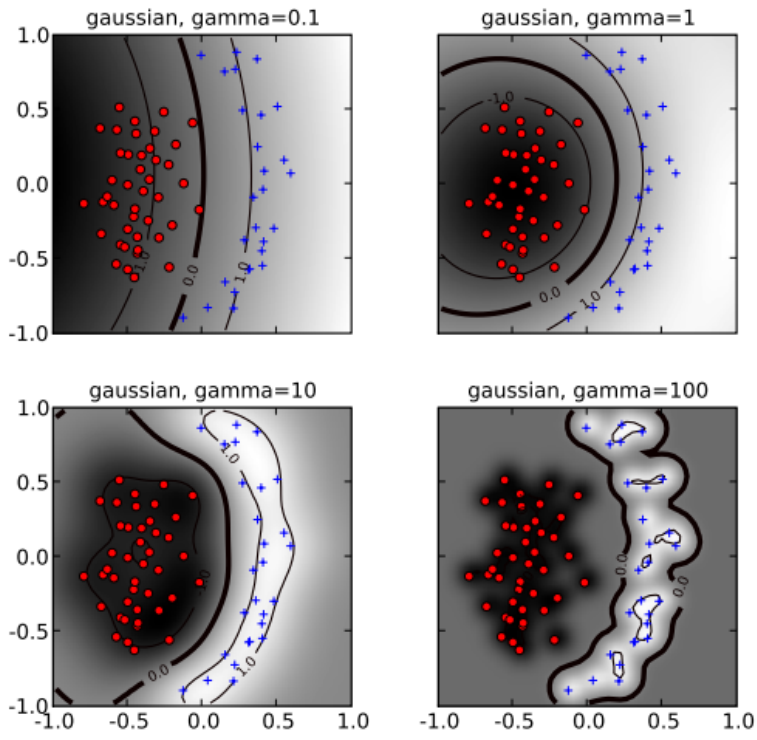


Figure 4.4: RBF Linear Kernel, change in parameter gamma for a set C. Small values of gamma the boundary is almost linear. Large values of gamma can lead to overfitting. Taken from Ben-Hur and Weston [28]

A linear kernel is a special case of RBF and under certain conditions, such as high dimensional feature space,it has shown similar performance as RBF kernel in predictions [36].

A two-class and a multi class version of SVM was created in MATLAB for use in this project using a RBF kernel, although due to compute time constraints not all features could be evaluated using this RBF kernel. As such and to maintain a standard method in experimentation a linear kernel was used in modelling all of the features in order to fairly compare the performance.

4.6 Evaluation Measures

Due to the unevenness of classes and the reuse of certain households in modelling accuracy alone cannot be used as a measure of good performance in the model. Methods for multi class precision and recall along with F-score are used to asses the worth of a model.

Data class	Classified as *pos*	Classified as *neg*
pos	true positive (*tp*)	false negative (*fn*)
neg	false positive (*fp*)	true negative (*tn*)

Table 4.1: Confusion Matrix

Confusion Matrix for two class classification.

4.6.1 Two-class Evaluation Measures

The accuracy of a classifier is the ratio between the number of correct classifications to the total number of test samples [37]. Further, the evaluation measures of precision, recall and F-score are introduced [38].

Precision for a two class problem as defined below in Table 4.2 denotes the probability that a sample classified as class 1 truly belongs to class 1. For example if the *few* class has a precision of 70% it indicates that if the number of residents is classified as *few* it truly is a *few* household with 70% probability.

Recall for a two class problem as defined below in Table 4.2 denoted the probability of a sample being classified as class 1 given that the sample belongs to class 1. For example a recall of 70% in the classification of *few* indicates that 70% of the households that actually have 2 or fewer residents are also classified that way.

Choosing to evaluate the classification models with either precision or recall could be misleading about the performance of the models. Instead, here the F-score is used. F-score, as defined below in Table 4.2 when β =1, is the harmonic mean of precision and recall where the importance of precision and recall are measured equally. Other F-scores exist where the value of β favours precision or recall over the other.

For example $F_{0.5}$ score weighs precision higher than recall. F_2 score puts more importance on recall than precision

Measure	Formula	Evaluation focus
Accuracy	$\frac{tp+tn}{tp+fn+fp+fn}$	The achievement of a classifier
Precision	$\frac{tp}{tp+fp}$	Class agreement of the data labels with positive labels given by the classifier
Recall	$\frac{tp}{tp+fn}$	Achievement of a classifier to identify positive labels
F_β score	$\frac{(\beta^2+1)Precsion*Recall}{\beta^2 Precision+Recall}$	Relationship between data's positive labels and those given by a classifier

Table 4.2: Two class Evaluation Measures.
Measures for two class classification using the notation in Table 4.1

4.6.2 Multi-class Evaluation Measures

We use the following evaluation measures for multi-class classification. We extend the evaluations used above in Table 4.2 to multi-class Accuracy, Precision, Recall, and F-score [39].

Measure	Formula	Evaluation focus
Average Accuracy	$\frac{\Sigma_{i=1}^{l} \frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class effectiveness of a classifier
Precision$_M$	$\frac{\Sigma_{i=1}^{l} \frac{tp_i}{tp_i+fp_i}}{l}$	An average per-class agreement of the data class labels with those of a classifiers
Recall$_M$	$\frac{\Sigma_{i=1}^{l} \frac{tp_i}{tp_i+fn_i}}{l}$	An average per-class effectiveness of a classifier to identify class labels
F$_\beta$ score$_M$	$\frac{(\beta^2+1)Precsion_M * Recall_M}{\beta^2 Precision_M + Recall_M}$	Relations between data's positive labels and those given by a classifier based on per-class average

Table 4.3: The number of instances. Table and descriptions taken from Marina Sokolova and Guy Lapalme [39]
Measures for multi-class classication based on a generalization of the measures of Table 4.2 for many classes C_i: tp_i are true positive for C_i, and fp_i false positive, fn_i false negative, and tn_i true negative counts respectively. M indices represent macro-averaging, which is the the average per class.

4.7 Implementation

All the models in this project have been developed in MATLAB [31] (MATLAB 2013b) using standard MATLAB Toolboxes.

4.7.1 KNN

10 fold cross validation is used to find the best number of neighbours for the input vectors that classify the problem both with high accuracy and F1-score Macro, which measures all of the classes equally regardless of size. Once the best k is found the model is run one last time on this k number of neighbours using cross validation and the final evaluation measures were outputted to a text file.

The function used to fit the model is *ClassificationKNN* from the Statistics Toolbox. This is used to fit the model to the input features and class labels using a KD-Tree as a search method. Then, *crossval* is used to cross validate this model using kfold cv with k = 10. Next, *kfoldPredict* returned the predicted class labels and used these and the correct class labels to create a confusion matrix. Once the confusion matrix

was created there was need to develop the evaluation measures based on the equations above. This was done using custom functions.

4.7.2 SVM

Standard C-SVC SVM is used for the SVM model in this project.

It is suggested that when the number of features is large it may not be necessary to map to multiple dimensions and using a linear kernel and just searching for the cost parameter may suffice [40].

For the linear kernel implementation of C-SVM the soft margin cost value is a regularisation parameter. Large values for C result in smaller margin hyperplane if that hyperplane performs well at classifying the training points. Small values of C will result in larger margin separating hyperplanes even if it will misclassify more instances.

Due to computational time constraints, an RBF kernel is not used in this project. One had been implemented using the following procedure, but the results are not used in this project. When a radial basis function kernel is used where the cost and sigma (the kernel width) parameters (as mentioned above in the definition of the RBF kernel) are found by finding the best combination of cost and sigma where
cost $\in 2^{-5}, 2^{-3}, ..., 2^{15}$ and sigma $\in 2^{-15}, 2^{-13}, ...2^{3}$.
This is the method used to find the best parameters [40].

This is a standard grid search method to find the best parameters for cost and sigma where the best are decided to be judged by the F-score evaluation measure, as accuracy alone cannot be relied upon as a performance measure. Once the best parameters have been found, the SVM is trained once more on these parameters and the final evaluation measures are outputted to a text file.

Two-class SVM

Cross validation is performed using *crossvalind* to split the input matrix into 10 sets of training and testing sets. A performance tracker, *classperf*, is then initialised and updated after the model has been trained and tested. Standard MATLAB functions for training and testing SVM are used, *svmtrain* and *svmclassify* which train the SVM model on the training set and then classify instances in the testing set. The performance tracker then allows a confusion matrix to be created, and once again custom functions were used to return the evaluation measures as mentioned above.

Multi-class SVM

A 1vs1 method of multi class SVM has been implemented for this project. As with all the models 10 fold cross validation is used to fine tune the parameters.

The 1vs1 method of multi-class SVM develops one binary model for every pair of classes using only the class labels belonging to these classes, this is done using *nchoosek* to create the combinations of all pairs of classes, then the elements belonging to a particular set of two classes are set to binary features and *svmtrain* is used to train the instances belonging to these models and *svmclassify* is used to classify the instances in the testing set. In the testing phase an instance is tested in each of the 15(M(M-1)/2) models an instance is decided to be in a particular class if it receives the most votes.
The 1 vs 1 model used in this project is based on an entry on an online forum, it was edited to use 10 fold cross validation for use in this project. A confusion matrix is returned and the custom evaluation measure functions were used to decide on the best parameters.

The models are then run on the best parameters and the final evaluation measure results are outputted to a text file.

4.7.3 LDA

LDA is available in MATLAB within the statistics toolbox. 10 fold cross-validation was used in this classification to assess the performance of the model. The function *ClassificationDiscriminant.fit* is used to fit a Linear Discriminant Model to the data. Using *crossval* with 10 fold cv the accuracy of the model can be inspected by finding the KFold Loss, this is the average error value of the model over all of the number of folds of cross validation. Once again custom functions were used to find the Confusion Matrix of the classification of the model and to retrieve the values for the evaluation measures as mentioned above.

Chapter 5

Results

5.1 Overview

The results of this project can be broken down into five main areas.

Firstly, we will show the performance of the recreated features on our dataset and compare it to the performance from Beckel *et al* [1].

Secondly, the results from the feature exploration will be shown and discussed. In this section we will discuss how each of the features from Table 3.2 perform in two-class classification.

Thirdly, we will explain the results returned from the multi-class classification on this dataset using all of the features from Table 3.1 and Table 3.2 where we gauge the potential for gathering information on the precise number of residents.

Next, we will also comment on the performance of particular methods used in this project and how the performance of the three methods was affected by multi-classification.

Lastly, other interesting findings from the experiments will be discussed, such interesting findings include the performance of the various feature selection methods and about the features that performed well.

5.2 Recreated

We will now show the results of the experiments run by Beckel *et al* compared with the results we attain from recreating their features and running the models on our HES dataset. We use the features as described in Table 3.1 from [5]. These recreated features contain four categories of feature types base on 30 minute interval energy readings from households measured for a week. The four categories can be separated into consumption figures, ratios, temporal properties and higher-order statistical properties.

Consumption figures are simple statistics of the consumption data for a household such as the maximum or minimum power sample. These features allow comparison of

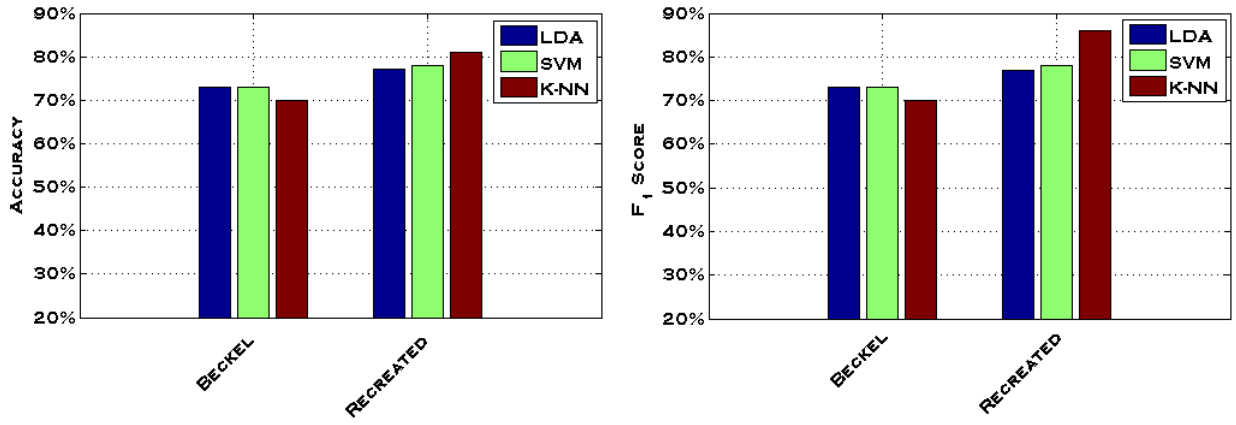
households given the average readings for particular portions of the day and identify similarities between households of the same class.

Ratios of average consumption values of different periods of the day are the second category that these features can lie in. These ratio figures can allow us to capture relevant patterns of energy usage throughout the day. For example, we may be able to use the ratio of average evening usage over the average afternoon usage to tell whether cooking happens in the afternoon the evening or both [1].

Temporal features correspond to time relevant features such as the amount of time where the readings captured were above the mean energy use in the household.

The final category of features is statistical properties. Here we measure the variance of the readings in a household the the absolute difference between successive readings as well as the cross correlation of subsequent days and the number of peak readings for a household.

5.2.1 Using the same features



Beckel vs Recreated. Accuracy **Beckel vs Recreated. F_1 Score.**

Figure 5.1: Performance comparison of Beckel i*et al* to this projects. Two-class classification

The graphs in Figure 5.1 show the performance that Beckel *et al* [1] achieved on their dataset and the comparison of their features recreated and modelled on our HES dataset. We can see that their results are indeed replicable on another dataset. These results proved with confidence that the results gathered in Beckel *et al* [1] are replicable on other data sets. The similarities in results can be attributed to the similarities between the CER [18] data set used by Beckel *et al* [1] and the HES [17] data set we use here in this project as discussed in Section 2.5.

The results from Beckel *et al* [1] achieve classification accuracy of between 70% and 73% with the models that they are using on their dataset to predict that the number of residents is few or many.

Using the same models and recreating the features used we found that the results on the HES dataset (taken with weekly readings) achieve classification of between 77% and 81% depending on the model used.

The F score achieved in Beckel *et al* [1] ranges between 70% and 73% with LDA and SVM achieving the highest scores. When we compare the F score from our models we see that the scores range between 77% and 86% where once again LDA and SVM perform similarly, and KNN has the highest F score. The high F score indicates that the precision and recall of the models predictions are quite high, and that the model performs well in predicting the correct class of a sample.

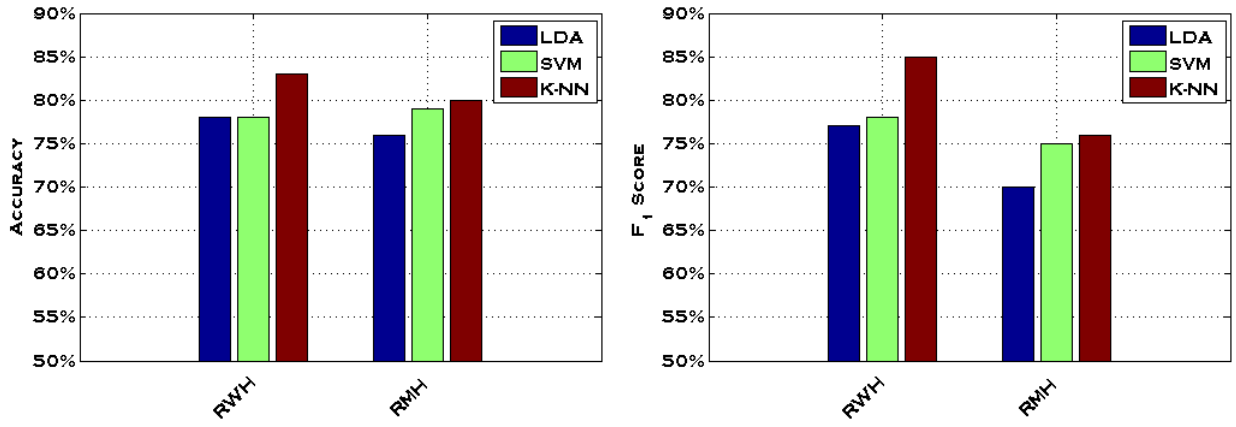
5.3 Exploration of Features

This section shows the results obtained from the exploration of features that was performed. The results of modelling the features as described in Table 3.2 are shown below. All of the features as described in that table were modelled and the full set of results is available in the accompanying data files with this project if the reader wishes to further analyse the results. The following graphs will only show the hourly versions of the features as described in Table 3.2. This is because the performance of the models only varied slightly when using hourly readings over half hourly readings, using either results in comparable performances. The results for each of the features were taken from the best performing of the feature selection methods as discussed in Section 3.4.

We would like to remind the reader about the list of acronyms at the beginning of this report and invite them to revisit Table 3.2 to familiarise themselves with the features used in this project.

A note on the display of the results.
The accuracy results of models are shown on the left hand side of the page and the accompanying F score results are shown on the right hand side of the page. Where appropriate and when comparing monthly and weekly results the top graphs will show weekly results and the bottom graphs will represent the results when modelling the monthly features.

5.3.1 Recreated Features



Recreated Features Accuracy **Recreated Features. F Score.**

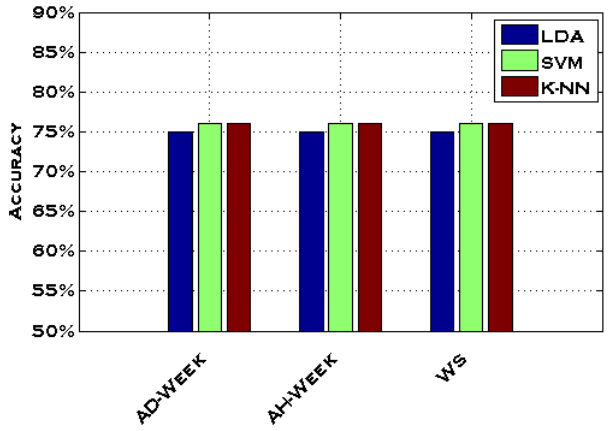
Figure 5.2: Recreated Features. Two-class classification performance.

Using the features as described in Table 3.1 on our data set but using hourly intervals we attain the results in Figure 5.2. We see that good accuracy can be achieved when estimating the number of residents of a household in the two class case. K-NN scores best in both weekly and monthly readings in terms of both accuracy and F score. We see that weekly readings achieve a better F score compared to the monthly readings.

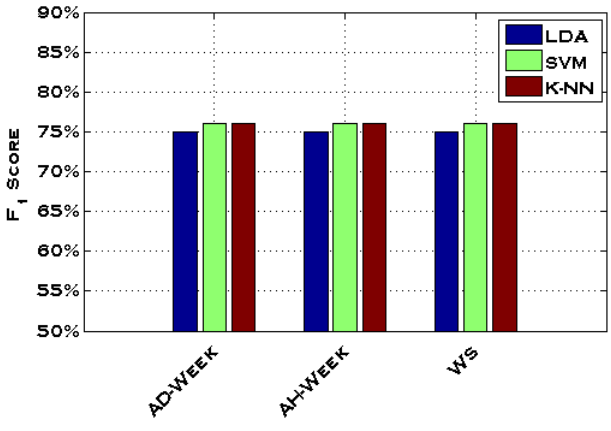
The F score is an indicator of the predictive power of a model. A model can often have good accuracy if it predicts one class more than the other. A high accuracy with a high F score indicates that a model performs well, a high accuracy but a low F score can indicate that a model is not predicting well. Here we find that the monthly readings do not perform as well as the weekly readings and this could be an indication of a need for more data to predict with more accuracy the data about households.

5.3.2 Simple Features

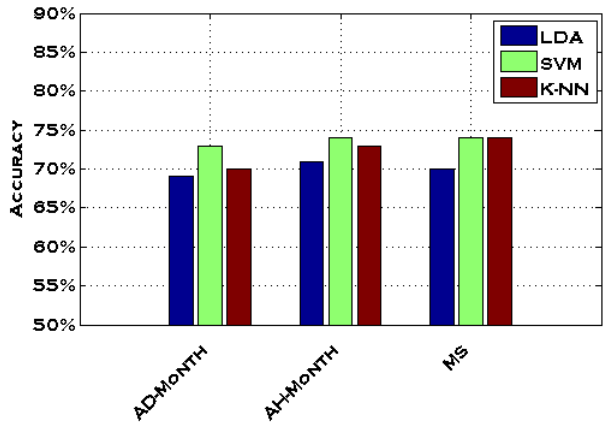
The following features are referred to as simple features because they are of single dimension. Features in this simple feature category include the monthly sum (MS) the weekly sum (WS) which is the total energy used in a single week, the average day (AD) feature represents the average energy used in a single day which could be measured over a week or a month, and average hour (AH) which is the average hourly reading which once again could be measured over different collection intervals.



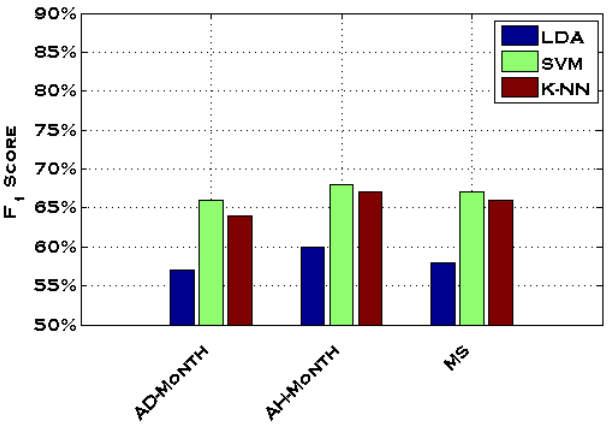
Simple Weekly Accuracy



Simple Features.Week.FScore



Simple Montly Accuracy



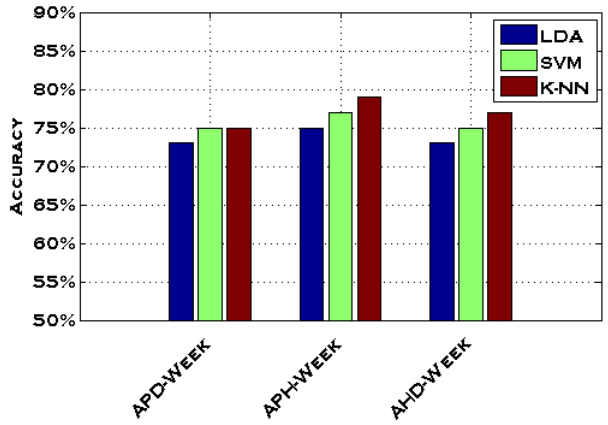
Simple Features .Month. FScore

Figure 5.3: Simple Features. Two-class classification performance

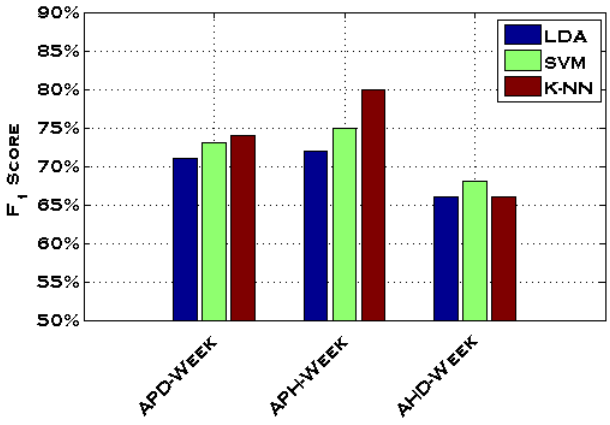
Results for simple features are shown in Figure 5.3. Weekly readings result in somewhat higher accuracy (upper left) than that of the monthly dataset (bottom left). The F score for the monthly readings is much lower than those of the weekly readings the difference in sample size could have some impact and could tell us that using more data will results in better results showing again that the amount of data needed to improve the predictive power of the classification would need to be larger. This indicates that if some nefarious party wishes to discover personal data that they would need to have a large dataset of information on which to base their model.

5.3.3 More Complex Featues

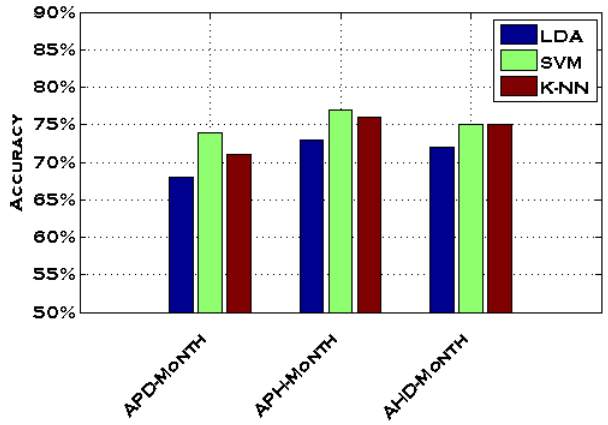
These more complex features are labeled as such because of the increase in dimensionality of these features compared to those we have seen before. Here we are using features such as the average per day of the week (APD) which has 7 consumption features, one for each day of the week. Also used are average per hour (APH) which measure the average energy reading per hour of the day, resulting in 24 features in this feature vector. The third type of feature used is the average per hour per day (AHD) which are the average energy consumption readings for each hour for each day of the week this results in 168 (24*7) features in this vector one for each hour in a week.



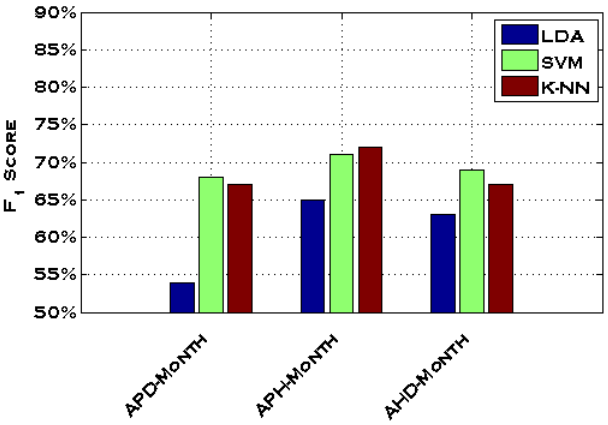
Complex Weekly Accuracy



Complex Features.Week.FScore



Complex Montly Accuracy



Complex Features .Month. FScore

Figure 5.4: More Complex Features. Two-class classification performance

Using these more complex features the models achieved the results above in Figure

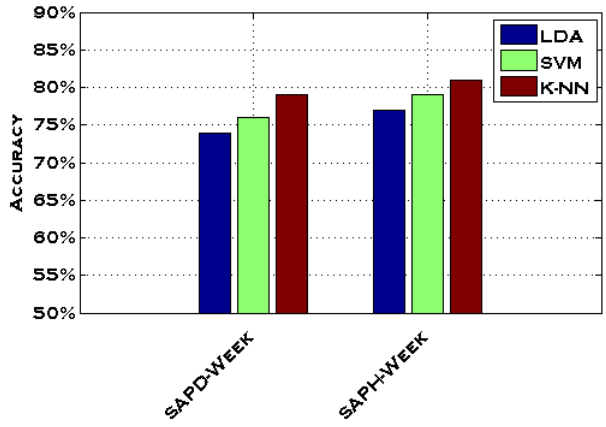
5.5. We see that a good performance can be achieved with using the average per hour (APH) readings, especially in the weekly dataset where accuracy of 79% with corresponding F score of 80% can be achieved. When we compare these results to those of the recreated features we find that similar accuracy can be achieved. Recreated features achieves 80% accuracy with an F score of 85% at its best, compared to the 80% we see here. This could be an indication that a set of features which are more simple than in Beckel *et al* [5] could achieve good results.

In both the weekly and monthly datasets we see that the average per day of the week (APD) does not perform as well as the others. When using LDA we find the F score to be as low as 54% suggesting that the correct class label would be assigned to a household only around 54% of the time.

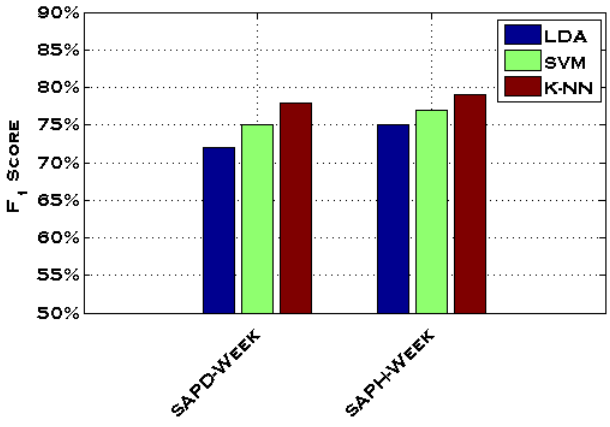
When we compare the results of these more complex features to those of the simple features we find that the performance in terms of both accuracy and F score improves with the average per hour (APH) and average per hour day (AHD) in both the monthly and weekly cases compared to the simple features performance. We find though that the average per day (APD) performs similarly in the weekly case, not improving from the simple one dimensional features, and even decreasing the performance when using LDA to classify household occupancy.

5.3.4 More Complexity

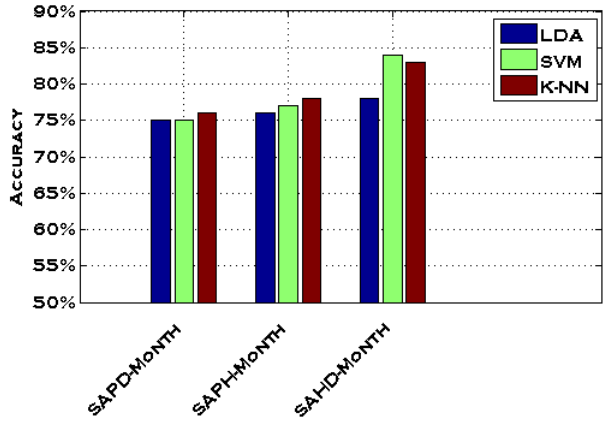
These features add higher order statistics into the features mentioned above. These features include the standard deviation of readings. There is the average and standard deviation of readings per day of the week (SAPD) which results in 14 features. For the standard deviation and average per hour (SAPH) there are 48 features and we also make use of the standard deviation and average reading per hour per day (SAHD) where this results in 336 different features. Due to the requirement of standard deviation of an hour of a particular day these readings can only be used with readings taken over a period of more than 2 weeks, here we are using the monthly readings where we can find the standard deviation over at least four points.



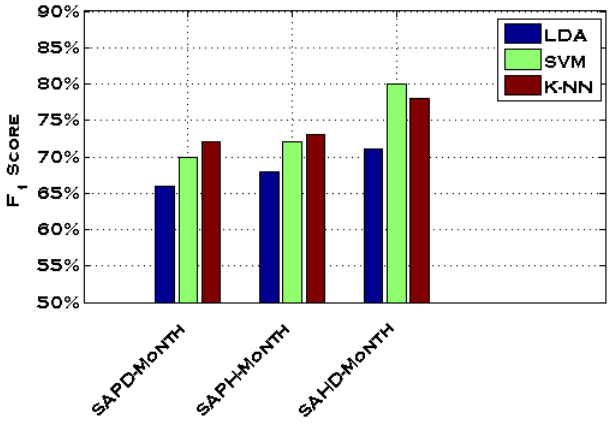
More Complex Weekly Accuracy



More Complex Weekly F_1 Score



More Complex Montly Accuracy



More Complex Features .Month. FScore

Figure 5.5: More Complex Features. Two-class classification performance

From Figure 5.5 we find with these features we achieve the best accuracy performance of all of the features we have explored so far. When we include the standard deviation into these readings we find that we can improve classification accuracy up to 4% more than without using the standard deviation, as is the case with APD and SAPD. We find the best performance of all of the features used when we evaluate the performance of the SAHD features. We achieve accuracy of 84% compared to the accuracy and F score of the recreated features at 82%. This set of features with added complexity also have the largest F scores indicating an increase, albeit a small one, in the predictive performance of the model.

5.4 Observations From Two Class Classification

The recreated features performed best on the weekly dataset. Using only 1 week's data and a large dataset the recreated features could be used to determine the number of dwellers in a household with accuracy of up to 83%.

If the amount of data collected was more than just one weeks worth of readings, other features such as SAHD could be used to predict the number of dwellers with up to 84% accuracy, or a probability of around 80% using a smaller dataset.

The models used for two-class classification all perform similarly throughout the feature exploration. We find that for simple features such as the average day (AD) that performance is within 1% for all of the models. Once there are more dimensions and levels of complexity we find that LDA performs worst with SVM and KNN achieving the better results. SVM works well here in binary classification of residency numbers due to the nature of the problem being 2 class, which is what SVM was originally designed for.

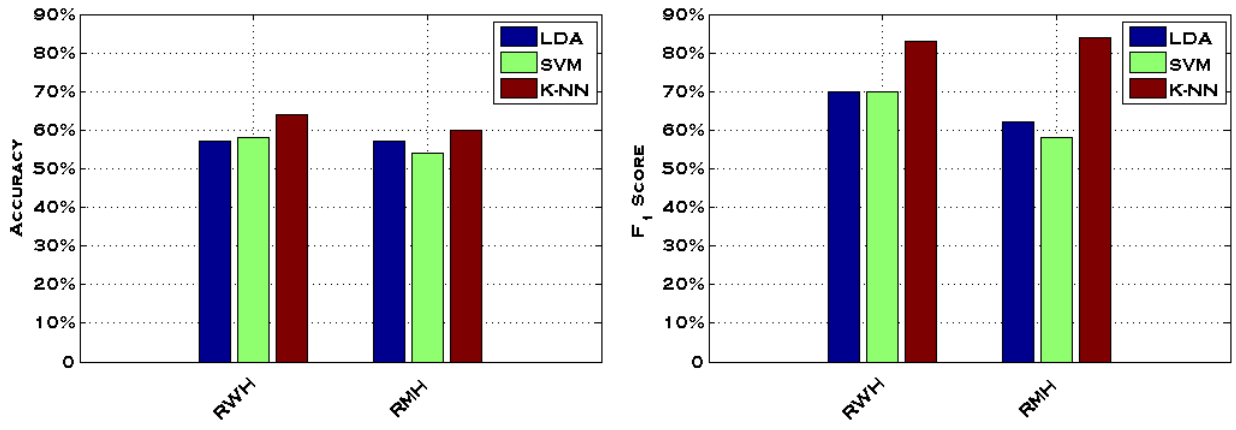
5.5 Multi-class

The features were then used to model the multi class problem of finding the precise number of residents of a household. These ranged from 1 person in a home to 6+ residents per household resulting in a 6 class problem. Once again LDA, KNN and SVM are used to model these predictions. Where the SVM method used is a 1vs1 SVM model as discussed in Sections 4.3.1 and 4.7.2. The following results show that there is indeed some form of risk for households with smart meters having personal information about them identified. There is up to 70% accuracy in correctly identifying the number of residents in household. The maximum F Score for these features has been 83% showing that some models have high predictive performance.

Rather than explore the results gathered from all of the features again graphically as in for two class classification we discuss the results, which can be found in the accompanying data files for this project.

5.5.1 Recreated Features

These are the features as described in Table 3.1 where the features can be split into the four categories of consumption, ratios, temporal information, and statistical values.



Recreated Features Accuracy **Recreated Features. F_1 Score.**

Figure 5.6: Recreated Features. Multi-class classification performance

Here in Figure 5.6 the recreated features on the weekly dataset using hourly interval readings (RWH) are compared to the recreated features on the monthly dataset using hourly intervals (RMH)

This shows that with these features the precise number of residents in a house can be predicted with accuracy of up to 64% That means that without any *a priori* knowledge and just with one weeks worth of half hourly readings the number of people

in a household can be estimated. This shows the potential risk of smart meters where particular and precise information can be determined from energy readings. The high F scores indicate the models are more likely to classify the households correctly.

These results also show what will become a trend in the results pattern where K-NN is the superior classifier for predicting the number of residents in a household.

5.5.2 Simple Features

The features that fall into the category of simple features are the average reading of a day (AD), the average hour energy use (AH) and the weekly and monthly sums of the total energy used in those periods (WS, MS). As discussed in Section 3.3.2 we expected poor performance of these simple features when used for multi-class classification because of the similarities in the energy use between groups at this coarse level of granularity.

We found that in both the monthly and weekly datasets achieve an accuracy of around 50% with KNN. The accuracy of LDA varied from 45 to 50% when used on the weekly and monthly datasets respectively and SVM achieved the lowest accuracy with a score of 40%. We found that the F score of these models reached a maximum of 50% (KNN with weekly data) and as low as 11% (AH month) This means that the classifiers were not able to accurately apply the correct label to a household, or a household to a class with a reasonable probability of labelling it correctly. This tells us that if data in the form of these features were captured from a smart meter that it would be unlikely that anyone would accurately be able to predict the number of residents in a household.

5.5.3 More Complex Featues

These more complex features correspond to features that have more than one dimension and use the average readings for particular time intervals. For example here we use the average reading per day (APD) the average per hour of a day (i.e. the average per each of the 24 hours in a day) (APH) and average per hour per day. Also included here are half hourly counterparts to the hour measured features.

We see a good improvement in the performance of these features, achieving over 60% accuracy with KNN using APH, APHH, AHD and AHHD. The average per day (APD) results vary from 45% to 55% depending on the dataset used. Average per hour (APH) preforms best with KNN accuracy of 63% and corresponding F score of 75%. The F scores for the features in this category are high, often achieving over 75%, meaning that the predictive power of the models built on these features is able to apply the correct class label to the households in a multi-class situation.

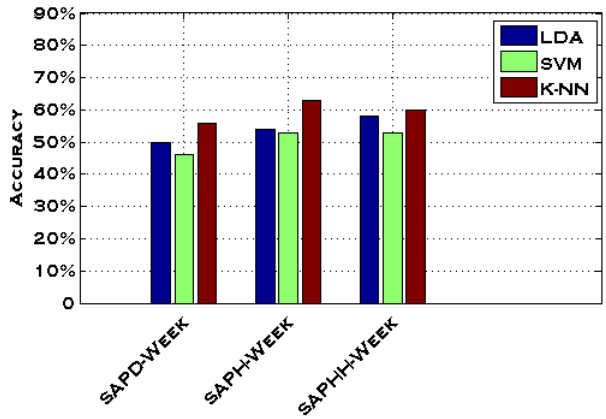
The classification performance of these best features, APH, AHD, are almost as good as the recreated features we used from Beckel *et al* [5] which results in an accuracy of 65%.

5.5.4 More Complexity

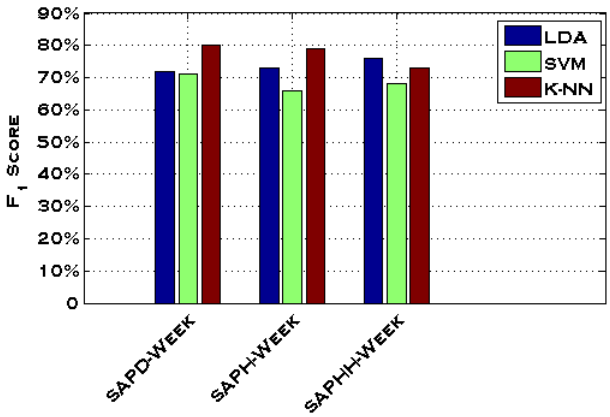
These features are similar to the complex features discussed above where they use the average consumption figures over a certain period. Now though we include the standard deviation to these features.

We are using here the standard deviation and average per day of the week (SAPD), standard deviation and average per hour of the day (SAPH), and the standard deviation and average per hour per day of the week (SAHD).

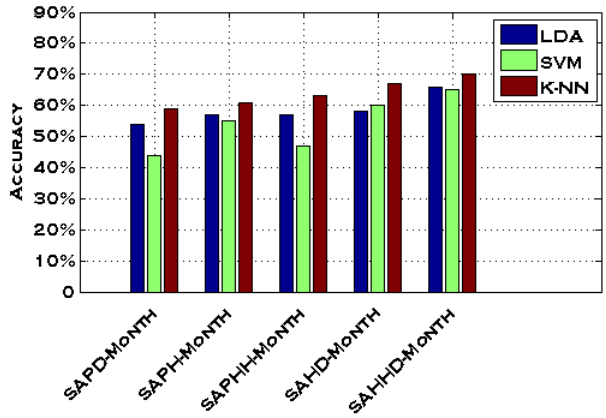
When using these features with the multi-class problem we find not much improvement from just the average readings counterparts to these features. We see that SAPH reaches accuracy of 63% and a corresponding F score of 80%.



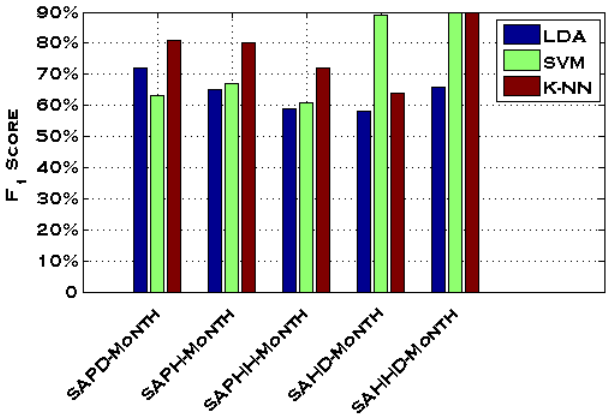
More Complex Weekly Accuracy



More Complex Weekly F_1 Score



More Complex Montly Accuracy



More Complex Features .Month. FScore

Figure 5.7: More Complex Features. Multi-class classification performance

In Figure 5.7 it can be seen that with these even more complex features we can achieve accuracy of predicting the correct number of residents in household of up to 70%. This is using SAHHD, which is a feature set that can only be applied to monthly data as we are taking the standard deviation of the hour per day across at least 4 data points when using monthly readings.

5.6 Observations on Multi-class Classification.

With either monthly or weekly readings the number of residents in a household can be predicted with over 60% accuracy and with a probability of predicting the correct number of residents in a household of over 70% across the different features used.

The features used in this project perform comparably across the weekly and monthly datasets indicating that the features are useful consumption figures that can be used in multi class classification even when not dealing with large datasets. The high F scores indicate that even when dealing with smaller dataset we can achieve better classification performance when performing multi-class classification compared to two-class classification when using the smaller monthly reading dataset.

5.7 Classifier Comparison

5.7.1 Two Class

The three methods achieved comparable accuracy scores across the features used for two-class classification. K-NN did often perform the best, although the differences were quite small indicating that no one model trumps the other on accuracy.

When judging the performance of the models on their F scores we find that LDA frequently performs worst, especially in the case of the simple features used, like weekly sum (WS) or average day (AD). SVM and KNN often had F scores results in the same neighbourhood, indicating the performance of these models performed similarly.

5.7.2 Multi-class

SVM performs worst at multi-class classification. We are using a linear kernel, and perhaps a different kernel should be used as the data may not be linearly separable when faced with multi-class classification. KNN is once again best of the three classification models, but LDA performs comparatively on both accuracy and F-Score despite the underlying assumptions of LDA on the data.

5.8 Additional Interesting Findings

5.8.1 Feature Selection Methods

There is not much difference in the performance of the three feature selection algorithms we use, CFS, FCBF and Fisher. When dealing with two-class classification the evaluation measures are close in size across the different feature selection methods used, we find that CFS tends to improve the performance by a small amount, around 3% accuracy performance such as in the case of average per day (APD), compared to FCBF. When dealing with the multi-class classification though the increase in performance of the models built using the CFS method is larger, and the decrease in performance of those models built on FCBF is also larger.

Overall the performance of the features selection methods over just using the entire feature set is not that different. This suggests that the use of other feature selection methods should be employed and the change in performance of the feature selection methods evaluated.

Chapter 6

Conclusion and Further Work

6.1 Conclusion

This project set out to support the theory that the information collected by smart meters is a potential information security risk, where personal and precise information about households and those who live in them can be detected with just the use of the energy signal.

In particular this project has shown that the work of Beckel et al [1] can be used on other datasets with similar results. This was achieved by recreating the features from Beckel et al [5] on our dataset by using python with embedded SQL. These feature files were then imported into MATLAB where various feature selection methods were applied and then three predicative models were used to classify households into *few* or *many* class given the energy data. This resulted in a two-class classification accuracy of around 80% with an F score of 85% indicating that these features performed well on our dataset and that two-class classification of household residency numbers was possible on this dataset. Following on from this there was an exploration of features, where varying consumption figures, such as the monthly sum (MS) and the average and standard deviation of the hours of the day were used for two-class classification. This project showed that other features performed as well as the work of the recreated features for two-class classification.

A multi-class expansion of the classification of household residency was conducted. We used all the features from the two-class classification in a attempt at multi-class classification of households into one of 6 classes, from 1 person households to 6 or more person households. Features were created using a MySQL database and embedded SQL in python, where once these features were created they were imported into MATAB. Various feature selection methods were then used and three classification models were built using these features. When attempting multi-class classification the SVM model needed a modification. Here a 1vs1 implementation of multi class SVM was built using MATLAB toolboxes. Multi-class classification was then attempted on all features with varying results. When using simple features accuracy of 50% and F score of as low as

10% were found. Suggesting that these simple features alone may not be informative enough to find the number of residents in a household using multi-class classification on our dataset. With a further exploration of features it was found that accuracy of up to 70% and F score of over 80% was possible. Showing that it is indeed possible to work out more detailed personal information of households given their energy readings, in the case of occupancy number detection on our dataset.

6.2 Further Work

This project shows it is possible to classify the residency numbers of households in both two-class and multi-class situations. Though, it is recognised that there may be room for improvement of the classification.

- Using more statistical information from the energy use data to create different features. Some of these would be based on finer collection intervals such as days to compare the changes in classification performance when using finer data.
- To assess the change in performance of applying some *a priori* knowledge to the households being classified, this information could be found in various ways by unauthorised groups. Examples of this include finding the household that the information came from on maps to discover the type of dwelling, and where on a street it is situated even the floor area can be collected from tax records and building plans.
- To apply these features that we already have to larger and different datasets to see if the performance can carry across into different data sets and to discover if the setup used in this project can be improved with even more data.
- To use the different features selection techniques to improve the performance. This could be achieved through the using of sequential feature selection methods to select the best performing features.
- To apply these features to multi-class classification of other characteristics to further assess the extent of security risks to households with smart meters. Other interesting classes to classify could be the social class categorisation of the household, this in particular could be useful for advertisers if they attempted to directly advertise to households.
- When using SVM for multi-class classification, it may be useful to experiment with the use of other non-linear kernels that could be used to partition the classes.
- When comparing the performance of weekly data and monthly data use data sets of the same size to better judge the change in performance brought about by the change in collection interval.

- To perform an investigation of the changes in energy usage in day of the week. Discover if they are as similar as in our dataset.

Bibliography

[1] Christian Beckel, Leyna Sadamori, and Silvia Santini. Automatic socio-economic classification of households using electricity consumption data. In *Proceedings of the Fourth International Conference on Future Energy Systems*, e-Energy '13, pages 75–86, New York, NY, USA, 2013. ACM.

[2] Elias Leake Quinn. Privacy and the new energy infrastructure (february 15, 2009). `http://ssrn.com/abstract=1370731` or `http://dx.doi.org/10.2139/ssrn.1370731`.

[3] M.A. Lisovich, D.K. Mulligan, and S.B. Wicker. Inferring personal information from demand-response systems. *Security Privacy, IEEE*, 8(1):11–20, Jan 2010.

[4] E. Jones U. Shankar P.S. Subrahamanyam, D. Wagner and J. Lerner. Network security architecture for demand response/sensor networks, june 2006.

[5] Christian Beckel, Leyna Sadamori, and Silvia Santini. Towards automatic classification of private households using electricity consumption data. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '12, pages 169–176, New York, NY, USA, 2012. ACM.

[6] Office of Gas and Electricity Markets (OfGEM). Transition to smart meters. `https://www.ofgem.gov.uk/electricity/retail-market/metering/transition-smart-meters`.

[7] Jorge Vasconcelos. Survey of regulatory and technological developments concerning smart metering in the european union electricity market. `http://hdl.handle.net/1814/9267`, 2008.

[8] European commission. directive 2009/72/ec of the european parliament and of the council of 13 july 2009 concerning common rules for the internal market in electricity and repealing directive 2003/54/ec, 2009.

[9] European commission. directive 2006/32/ec of the european parliament and of the council of 5 april 2006 on energy end-use efficiency and energy services and repealing council directive 93/76/eec, 2006.

[10] Office of Gas and Electricity Markets (OfGEM). Smart metering implementation programme: Statement of design requirements. https://www.ofgem.gov.uk/ofgem-publications/63543/smart-metering-statement-design-requirements.pdf.

[11] Office of Gas and Electricity Markets (OfGEM). Smart metering implementation programme: Data privacy and security. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/42730/232-smart-metering-imp-data-privacy-security.pdf.

[12] M. Ghofrani, M. Hassanzadeh, M. Etezadi-Amoli, and M.S. Fadali. Smart meter based short-term load forecasting for residential customers. In *North American Power Symposium (NAPS), 2011*, pages 1–5, Aug 2011.

[13] G.W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, Dec 1992.

[14] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[16] T. W. Anderson. *An introduction to multivariate statistical analysis / T.W. Anderson.* Wiley series in probability and mathematical statistics. New York ; Chichester : Wiley, [1984], 1984., 1984.

[17] Household electricity survey. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/278809/10043_R66141HouseholdElectricitySurveyFinalReportissue4.pdf, January 2014.

[18] Commission For Energy Regulation. Smart metering project. http://www.ucd.ie/issda/data/commissionforenergyregulationcer/.

[19] Cambridge Architectural Research Limited. Household electricity survey data briefing. http://www.carltd.com/article/Household-Electricity-Survey, 2013.

[20] Uk 2011 census. how we live. http://www.ons.gov.uk/ons/dcp171778_290685.pdf, 2011.

[21] Eoghan McKenna, Ian Richardson, and Murray Thomson. Smart meter data: Balancing consumer privacy concerns with legitimate applications. *Energy Policy*, 41(C):807–814, 2012.

[22] A.W. Whitney. A direct method of nonparametric measurement selection. *Computers, IEEE Transactions on*, C-20(9):1100–1103, Sept 1971.

[23] H. Liu and L. Yu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Correlation-Based Filter Solution". In Proceedings of The Twentieth International Conference on Machine Leaning (ICML-03)*, pages 856–863, Washington, D.C., 2003. ICM.

[24] Hussain F. Tan C.L. Liu, H. and Manoranjan Dash. Discretization: An enabling technique. In *Data Mining and Knowledge Discovery*, pages 393–423, Netherlands, 2002. Springer Netherland.

[25] Mark A. Hall and Lloyd A. Smith. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper, 1999.

[26] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2 edition, 2001.

[27] Arizona State University in association with DMML. Feature selection algorithms. `http://featureselection.asu.edu/software.php`.

[28] Asa Ben-Hur and Jason Weston. A user's guide to support vector machines. `http://pyml.sourceforge.net/doc/howto.pdf`.

[29] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[31] MATLAB. *version 8.2.0.701 (R2013b)*. The MathWorks Inc., Natick, Massachusetts, 2013.

[32] Andrew Moore. A tutorial on kd-trees. Extract from PhD Thesis, 1991. Available from http://www.cs.cmu.edu/*sim*awm/papers.html.

[33] Boriana L. Milenova, Joseph S. Yarmus, and Marcos M. Campos. Svm in oracle database 10g: Removing the barriers to widespread adoption of support vector machines. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pages 1152–1163. VLDB Endowment, 2005.

[34] Kai-Bo Duan and S.Sathiya Keerthi. Which is the best multiclass svm method? an empirical study. In NikunjC. Oza, Robi Polikar, Josef Kittler, and Fabio Roli, editors, *Multiple Classifier Systems*, volume 3541 of *Lecture Notes in Computer Science*, pages 278–285. Springer Berlin Heidelberg, 2005.

[35] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, Mar 2002.

[36] S. Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Comput.*, 15(7):1667–1689, July 2003.

[37] Ethem Alpaydin. *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2004.

[38] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[39] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing Management*, 45(4):427 – 437, 2009.

[40] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2010.