**Interim Report**

**Full Unit Project  
Benedict Wilkins**

**Supervisor: Zhiyuan Lou**

**2016**

**Table of Contents**

[**Smart Meters and Energy Demand Management** 4](#_Toc467773926)

[**Energy Demand Management** 4](#_Toc467773927)

[**Techniques for Supply Management** 4](#_Toc467773928)

[**Demand Side Management and Smart Meters** 4](#_Toc467773929)

[**Data Collection in Context** 5](#_Toc467773930)

[**Motivations** 6](#_Toc467773931)

[**Introduction to Multi-agent Systems** 7](#_Toc467773932)

[**Intelligent Agents** 7](#_Toc467773933)

[**Multi-agent Systems** 8](#_Toc467773934)

[**Distributed agent environment** 8](#_Toc467773935)

[**Multi-agent Systems in Context** 9](#_Toc467773936)

[**Agent Architecture** 10](#_Toc467773937)

[**Environment Architecture (Simulation)** 11](#_Toc467773938)

[**House Environment** 11](#_Toc467773939)

[**Environment and Physics** 11](#_Toc467773940)

[**Global Environment and Physics** 11](#_Toc467773941)

[**Software Engineering in the Java based Multi-Agent System** 13](#_Toc467773942)

[**Design patterns** 13](#_Toc467773943)

[**State** 13](#_Toc467773944)

[**Observer/Observable** 13](#_Toc467773945)

[**Factory** 13](#_Toc467773946)

[**Singleton** 13](#_Toc467773947)

[**Check style** 13](#_Toc467773948)

[**Test Strategies and TDD** 14](#_Toc467773949)

[**UML** 14](#_Toc467773950)

[**Revision Control** 14](#_Toc467773951)

[**Data Generation** 15](#_Toc467773952)

[**Introduction** 15](#_Toc467773953)

[**Data analysis** 15](#_Toc467773954)

[**Low-Carbon-London-Dataset** 15](#_Toc467773955)

[**Seasonality** 16](#_Toc467773956)

[**Building a sampling model** 17](#_Toc467773957)

[**Proceedings** 18](#_Toc467773958)

[**Testing Prediction Models** 19](#_Toc467773959)

[**Introduction** 19](#_Toc467773960)

[**Bibliography** 20](#_Toc467773961)

# **Smart Meters and Energy Demand Management**

## **Energy Demand Management**

Energy Demand Management (EDM) is in essence about managing the consumption/production of energy to ensure an effective energy network and to minimize cost and environmental damage. One of the main issues that EDM faces is how to efficiently supply energy to consumers. Peaks in energy demand arise when consumers – domestic or industrial, have synchronized habits. These lead to energy demand fluctuations (EDF), daily, weekly and seasonally [15]. These fluctuations must be dealt with by energy provides so not to result in damage to the energy network, blackouts and unpredictable service.

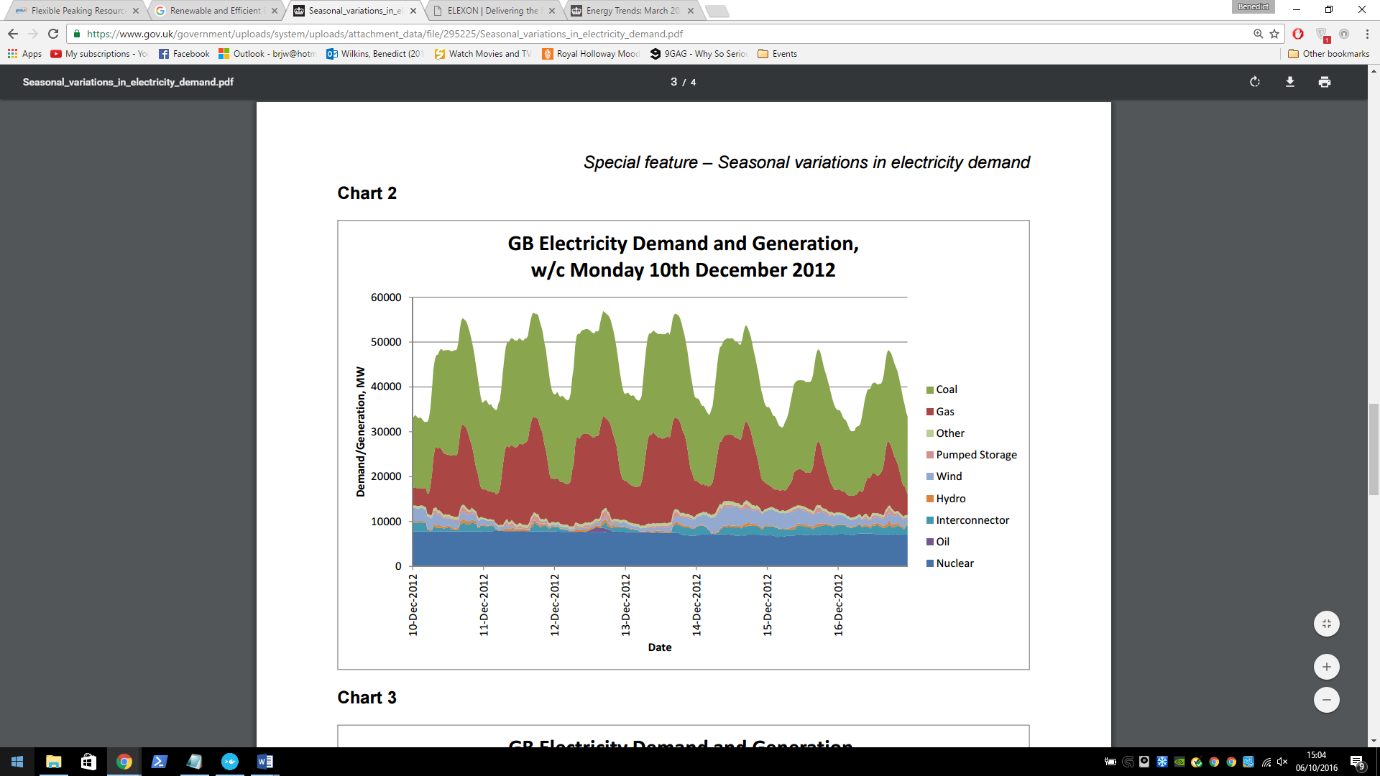
## **Techniques for Supply Management**

There are two general categories of power plant; base load and peaking. Base power plants are used to supply the base load power – the minimum requirement of energy over a period of time. They are usually nuclear, coal or large hydro-electric plants [24]. These plants are only turned off for maintenance or upgrades and usually provide power to a large area. Peaking power plants are one way to deal with EDF. They produce a variable amount of power which is matched to the current demand. Peaking power plants are expensive, heavily dependent on the fossil fuel market (as they are usually gas/coal burning plants) and usually only used as a late resort after employing the other methods mentioned below. See Figure 1.

Other techniques are used include energy storage and energy purchasing. Energy storage involves storing a large amount of energy and releasing it back to the grid during peak times. One of the most common examples of large-scale energy store are hydroelectric dams/pumps. Water may pumped during non-peak times or naturally build up behind a dam, later to be let through the turbines when it is required. A lot of energy is lost during this process so is not ideal. Energy purchasing is when an energy provider buys in energy from a separated grid. An example may be a British energy provider buying power from the French natation grid. Depending on the energy market this may be more desirable than using their own grid infrastructure

## **Demand Side Management and Smart Meters**

Moving to the consumer side of EDM. Another way of approaching the problem (which will be the focus of this project) is to alter the habits of the consumers to reduce EDF. This will result in reduced cost for energy providers and consumers. Demand response (DR) uses financial incentives to encourage consumers to alter their energy consumption habits. This method has been testing in the health sector, trying to alter habits that relate to health by heavily taxing cigarettes and alcohol for example and has been shown to be effective [16]. There have been various models that support DR in this projects context, see [18]. With the advent of smart meters and their increasing popularity in the market, it is now possible to collect house-hold specific data. The National Grid (NG) in the UK has previously had meters to monitor electricity demand on a larger scale (making the data in figure 1 possible to obtain). Now that data can be collected directly from the end point it is possible to implement DR by tailoring pricing in a way that reduces EDF and benefits the consumer.



**Figure 1**. Taken from Special feature – Seasonal variations in electricity demand. See bibliography entry [15].  
The above shows the power demand/generation on the GB National Grid from 10 – 16 of December 2012 starting at midnight (00:00). It illustrates daily energy demand fluctuations and shows the use of peaking and base load power plants. Nuclear power plants are shown here to be the base load power plants. It is clear to see that gas and coal fire power plants are being used to deal with the EDF and so are the peaking power plants.

## **Data Collection in Context**

It is important to look at the type of data that the system to be developed will be using. According to the Smart Metering Implementation Program [22], *‘GSME shall be capable of recording Consumption in each thirty minute period’.* (Gas Smart Metering Equipment – GSME), the same applies for Electric Smart Metering Equipment (ESME). There is also a daily data recording option however this option will not be suitable for the predictions that this project is concerned with. In principle the system will be able to support any reasonable time scale, but the most useful will be on an hourly time scale. The implementation details of the smart meter or what type of meter is irrelevant as long as the data is useable. With this in mind, the system isn’t necessarily limited to house-holds. Data may be collected from businesses/industrial settings. However it may be useful to make the distinction between the two as the scale of the data will be very different when comparing household to a factory. The relevance of this will be discussed in a later report on data pre-processing that will be presented with the data pre-processor program.

## **Motivations**

Machine learning is a heavily researched area in computing and is well suited to helping achieve this projects goal. Machine learning models require a few things to work – lots of data and a pattern. In this case data will be readily available from a large number of smart meters around the country. There is a pattern - looking a figure 1 gives a good indication that there is at least a day to day pattern. ANNs should be able to model this pattern effectively and provide good predictions as a result. ANNs have been used previous to forecast for arguably more complex system including the stock market.

Using a multi-agent framework will provide a flexible but robust structure for the system, it allows the system to be easily distributed. This will be key if the system is to scale well. The real smart meters can be considered agents – they operate in some environment and their goal is to record energy related measurements from their environment. It will be useful to model the intermediate ‘processing’ layers as agents within layered environments. It will be easier to experiment with different pre-processing methods by switching agent behaviours, agents could also automatically report errors or statistics using different behaviours.

DR may provide a cheaper alternative to the supply management solutions presented above. There are some problems it cannot solve – for example large EDF when weather/seasons change, but it shows potential in helping reduce EDF on a day to day scale. Utilising Smart meters and current computing technology to create a more efficient energy grid will help reduce our impact on the environment and reduce cost for everyone involved. Using the system to be developed will provide a means to this goal.

# **Introduction to Multi-agent Systems**

## **Intelligent Agents**

There are a variety of definitions for what a software agent actually is. A good way to look at an agent as a metaphor is to look at the real world. We may consider a human as a real world agent. We exist in an environment – the universe, we have beliefs, desires and intentions which lead us to make decisions and ultimately perform actions which may or may not affect our environment. In essence this is what a software agent does but in a software context – it and its environment will be running on one or more machines. [3]

We can model an agent using the real world metaphor. We as humans have a means to affect and observe our environment though use of our body. Parts of our body are used to perform actions in our environment – for example our hands can be used to pick up objects. In agent terms these parts are known as actuators or effectors. Other parts of our body are used for observation – for example our eyes, we look at the world and gather information about our environment. These parts are known as sensors. See figure 2 for an illustration of this idea. It is interesting to see how far this metaphor can be taken and how well it represents software agents – this will be discuss further in the *Starlite* section.

The environment is key aspect of agent based systems. An environment is just some space in which the agents in the system are housed. Agent environments have a number of properties [4], in the later section ‘MAS in context’ some attempt will be made to classify the environment of the system to be developed with these properties in mind.

In a lot of cases agents are in environment where other agents are also situated and so the need for communication between agents arises. These kind of systems are known as multi-agent systems.

Agent

Environment

Action

Perception

**Figure 2.** An illustration of an agent action on and observing an environment via is actuator and sensor respectively

## **Multi-agent Systems**

Agents interact with others agents in a variety of different ways depending on context. These interactions can be classified as a result from the following two agent behaviours; cooperative and competitive. Cooperative agents work together to achieve a common goal while competitive agents work to achieve their individual goals. It is not unusual for cooperative agents to display competitive behaviour and vice versa [14]. If an agent is *clever* enough it is easy to see how a competitive agent may behave like it is a cooperative. It may temporarily work with other agents if it finds that this is a more effective way of achieving its own goal.

Communication between agents is an important aspect of a multi-agent systems (MAS) and it provides the basis for agent interaction. Cooperative agents must have a means of communication in order to complete their goal as a group. They may need to dynamically organise themselves or communicate information that may be useful to the other agents. Competitive agents may also need to communicate. An agent based e-commerce trading system is a good example of a need for competitive communication. Each agent in the system will represent a real world entity such as a person or corporation. Their goal is to negotiate the best deal for the entity they are representing. They will require some communication protocol in order to negotiation [6]. It is up to the agent designer to provide these communication protocols in way that best suits the system.

## **Distributed agent environment**

The nature of the smart grid forces the system to utilise some distributed agent environment. Smart meters are physically separate entities each with an environment that they are responsible for – the household of the consumer. The smart meter agents (SMA) will need to be capable of communicating information to physically separate systems. This can be done using an IP protocol. Each SMA will be capable of communicating in this way with its manager agent. There are a few different possible configurations of management agents all with different associated positives and negatives and are to be discussed in this section.

The hierarchical structure of the MAS implies groupings of the agents. The SMAs already have a natural grouping, by geographical location or address which is already used by all household service providers. The manager agent’s job should be to forward data on behalf of its group as well as manage and control them by forwarding commands from higher in the hierarchy.

To keep the system as de-centralised as possible it may be desirable to have the SMAs doing management themselves. They could delegate one of the other agents in their group to be a manager. This configuration may avoid system failures; if the delegated agent fails, a different SMA can be delegated as the manager. There is a potential privacy issue with sending data to a Smart Meter in a household, some measures would have to be in place to prevent data theft or alteration at the management agent location.

The alternative configuration is to have the management agent reside further up the hierarchy. It would be more complicated to deal with system failures as the system would have no alternative management agent. However there will be no privacy issues with sending the data.

## **Multi-agent Systems in Context**

There have been various other studies involving multi-agent systems and the smart grid [2][5][10]. W. *Multi-Agent Systems (MAS) controlled Smart Grid - A Review* in particular gives a good summary of relevant studies. Most are concerned with managing operations and control of the smart grid and some address problems such as the centralisation of power plants and service restoration.

This project is less concerned with the technology that will be used to implement the control/management of the smart grid after predictions have been made. However from the studies mentioned previously it seems likely that MAS will play a key role. This motivates the use of an extendable MAS in this project.

**Classifying the agent environment**: (From afore mention standard environmental attributes (Russel and Norvig 1995)

To begin classifying the environment we must first define what exactly we mean in context. Each SMA is situated in its own ‘house’ environment in which it interacts (take readings). The house environment is however linked to the global environment; the environment containing all houses and any other agents in the system. This is the environment which we will try to classify as it contains all other environments.

**Inaccessible** – due to the nature of networking. Each SMA does not know what is happening in some other partition of the environment. It could ask for more information from the other agents but again there is no guarantee of a reply. Similarly the manager agents (whether they are central or not) do not have direct access to the house environments.

**Nondeterministic** – again due to the nature of networking. It is possible that the system will fail at some point, an SMA may fail and so no data will be sent up the hierarchy. It should be noted that the receiving agents should be able to handle not receiving any data. We do not want the employed machine learning method to fail because of this.

**Nonepisodic** – This attribute is a little harder to pin down. It could be said that the SMAs will reside in an episodic environment, all they will do it send data regardless of success or failure in the previous episode. In the simulated system the agents will receive an energy usage value every logical[[1]](#footnote-1)\* half hour interval. However it may be the case that in the real system the agent will be continuously monitoring the energy usage. Whether we consider the physical monitoring device part of the agent is up for debate and should be carefully considered in a fully deployable version of the system. The non-SMA (the agents higher up the hierarchy) are situated in a nonepisodic environment. Although data may be sent every half hour from the SMAs perspectives their time may be out of sync (this should be minimised as it will have an impact on prediction). The non-SMAs must be capable of continually receiving packets from the SMAs.

**Dynamic** – The environment will change independently of the each agent. Network failures and other agents may affect the environment.

**Discrete** – SMAs may only take an energy reading and send it. Non-SMAs may only receive messages, send messages.

## **Agent Architecture**

The agent architecture was taken from the …

The arrows represent the forwarding of events, results and actions.

Body

* The mind generates an action based on some reasoning, this is forwarded to the brain, which forwards it to the body. The body then generates an event from the action which is forwarded to the actuator. The actuator then forwards the event to the environment.
* The sensor receives a result from the environment, this is the agent’s perception. The sensor forwards it to the body and in the same fashion as above it reaches the mind as a result. The mind can then process the result as it sees it.

**In Java:** Each part of the agent is represented as a class, each class is an observer observable pair (each class can communicate via observer/observable design pattern with its respective parts). The environmental also uses the observer/observable design pattern in order to send and receive results and events respectively. The body class contains all of the parts of the agent and so is essentially the complete agent. A body may also have an appearance which is defined as being; the external appearance of the body – what the agent looks like to other entities in the environment.

The framework for the agents is given by the GAWL (Generic Agent World Library) package. This framework also includes packages for, actions, events, perceptions, environment physics and the observer/observable implementation.

## **Environment Architecture (Simulation)**

### **House Environment**

**Data Generator**

**­­­­**

The SMA will receive a reading from the Data Reader agent – who in the simulation will be set to read a value from the Data Generator at a given time frame. The time frame will be decided at the beginning of the simulation, each frame will represent a half hour interval in the deployable system. The house environment/physics will be responsible for forwarding the Data Reader -> SMA message. Once the message is received the SMA will forward the data up the agent hierarchy. The Data generator module will be global and will depend on the set up of the simulation. It will provide data reading from all houses based on their properties e.g. what financial positon the house is in.

### **Environment and Physics**

Each house has its own physics and environment, these are responsible for message forwarding within a house and will evaluate events that pass through them making sure they are valid in context. For example a house environment will receive an event from the Data Reader agent, it will forward it to the house physics which will evaluate it in context, make sure it is possible and valid. It will then execute the action (in this case executing the action means giving it back to the environment enabling a forward to the SMA).

### **Global Environment and Physics**

The arrows represent the transfer of events. Each SMA in its respective House Environment will send a message to the Global Environment. This will be processed (sent to the physics) and given to the Group agent. The Group agent may be a neighbourhood, region, country or other defined at the start of the simulation. The Group agent will aggregate the data in some way and format it so that it may be useful to the Predictor agent. The aggregated data will be sent in the same manner from the Group agent to the Predictor agent.

**Global Environment**

**House Environment**

**House Environment**

**House Environment**

**Global Physics**

There will be room in the architecture to provide multi group agents for different aggregation if needed. As an example we may have multiple neighbourhoods about which we want to make an aggregated prediction (a prediction regarding all houses in each neighbourhood). This can be done by simply having two Group (neighbourhood) agents with their respective House Environments. There is an obvious bottle neck at the Global environment/physics, this may be avoided by making a logical partition of the global environment e.g. for each group; a version of the global environment will be created to handle their communication specifically. In the real system the idea of Global environment will not hold as strong. The SMAs will forward their messages via IP communication – the machine who receives the message will essentially be the Global Environment.

# **Software Engineering in the Java based Multi-Agent System**

## **Design patterns**

### **State**

Allows an object to change its behaviour depending on its internal state by effectively changing its class.

The state design pattern has been used in many areas of my code. One example, the *SmartMeterAgentBody* class *brainHandler* attribute. The *brainHandler* attribute can be in one of two states – *NormalBrainHandler* or *IPBrainHandler*. This depends on the *Actuator* given to the agent at instantiation. One state allows the agent to handle *IPCommunicationActions* the other allows any non-IP related action e.g. *RecordAction*.

### **Observer/Observable**

The observer design pattern is used for event parsing between objects without explicit method calls. In java the *update* and *notifyObervers* methods are used to do this.

A custom version of the java implementation is used in GAWL for event parsing between agent parts, agents and environments, and environments and their associated physics. For of these concepts there is one or more classes in the system that represent them e.g. *NationalGridUniverse, SmartMeterAgentBody, HouseEnvironmentPhysics* etc.

### **Factory**

The factory design pattern provides an abstraction from object creation. No knowledge of the creation of the object is required, only the method in the factory for the desired object.

The *HouseModelFactory* class is one example of the factory design pattern in the system. It abstracts from house model creation, only an error term is required for creation.

### **Singleton**

The singleton pattern is used to ensure that only one instance of a class can be created. This instance can be accessed in a static way, the class is responsible for the creation of this single instance.

The HouseModelFactory class is one example of the singleton design pattern. Factories are often singletons as there should only ever be one instance.

## **Check style**

During the Java development process the google coding standards/check style was used to keep the formatting of code consistent and readable and also to make it easier to follow proper java coding standards.

## **Test Strategies and TDD**

TDD is a software development strategy in which the requirements of the system are broken up into essentially atomic tests. The tests are written and then code is written so that tests succeed. All tests are re-run each time the system is tested. The system is built up in this way so that (hopefully) by the end everything will work. In java JUnit is used for TDD, JUnit and a TDD strategy was used when testing some important utility classes such as *TestArgumentUltilities*, *TestTimeDateTracker*, and *TestMathUtilities* all of which can be found in the *test* package.

TDD was not a development strategy that would provide any advantage in the rest of the project. It becomes difficult to create unit tests for classes that interact heavily with one another and in a multi-threaded system it was nearly impossible. Instead, I used an integration based test strategy, testing the interacting between Observer/Observable pairs in a chain using simple print statements. For example to test event parsing between all parts of an agent – place a meaningful print statement in each *update* method and evaluate their order/correctness manually.

## **UML**

## **Revision Control**

# **Data Generation**

## **Introduction**

Data will be generated by a Data Generation module which will be global – it will be accessible to all House Environments. When generating data we want it to represent a real scenario as much as possible. The best way to do this is to acquire some real data; the energy usage of different households, and fit a representative model to it. We can then sample from the fitted model to generate realistic data. It is necessary to perform some analysis of the data (see below section Data Analysis) to find a good representative model. The analysis of the data gives some insight into desired properties of a prediction model e.g. linearity.

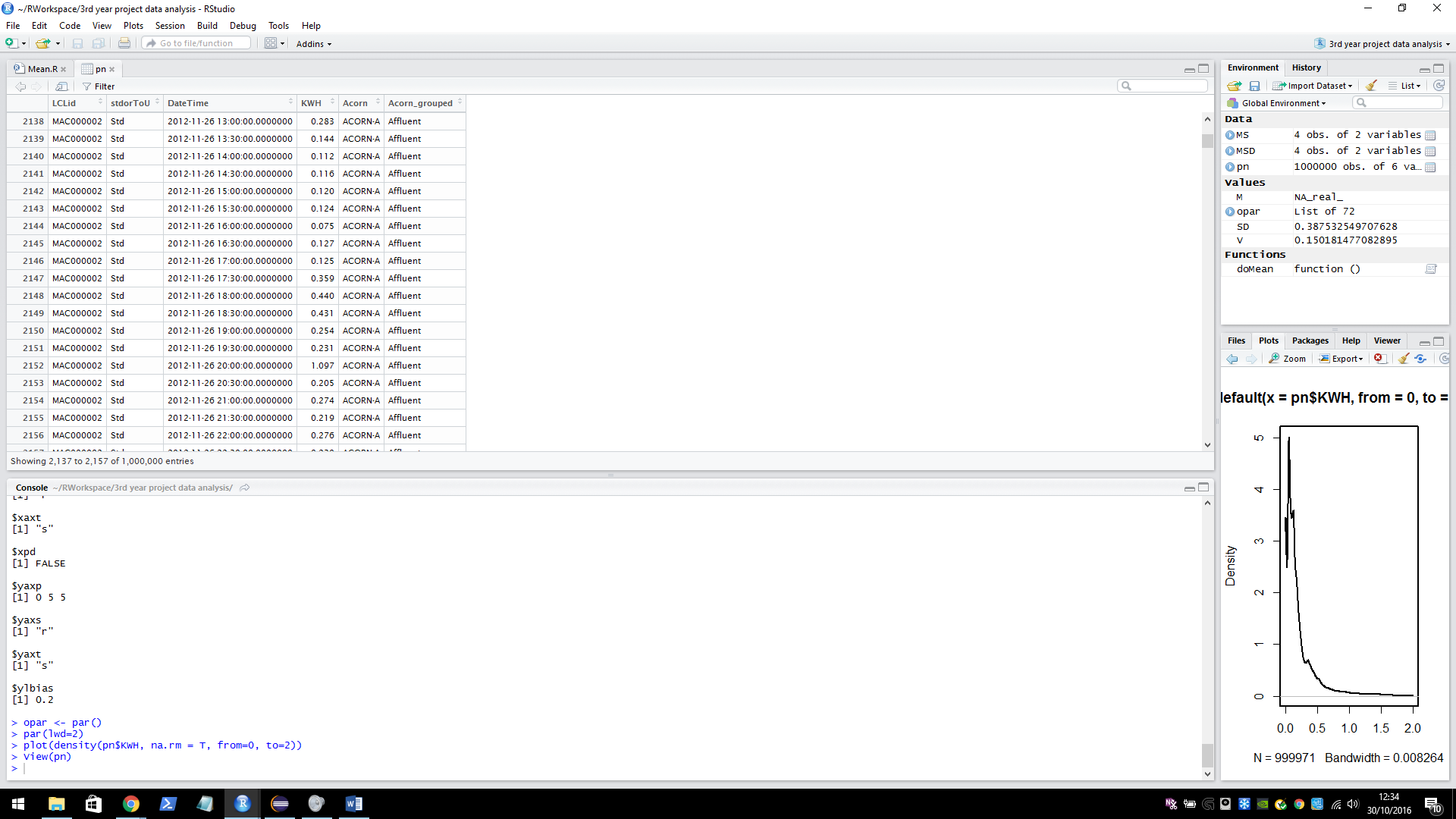
The below section contains analysis of the Low-Carbon-London dataset. This data set almost exactly the kind of data that the simulation will be using – half hourly data in different households for different financial situations.

## **Data analysis**

### **Low-Carbon-London-Dataset**

The data set includes KWH per half hour readings for a number of households. The data set is grouped into Affluent, Comfortable, Adversity and ACORN-U depending on the customer status.

The first section from this containing 1 million entries was used in the analysis. The complete dataset contains 167 million entries. The sample below shows 8 rows of the data set with the column names.

**LCLid**: the unique house identifier.

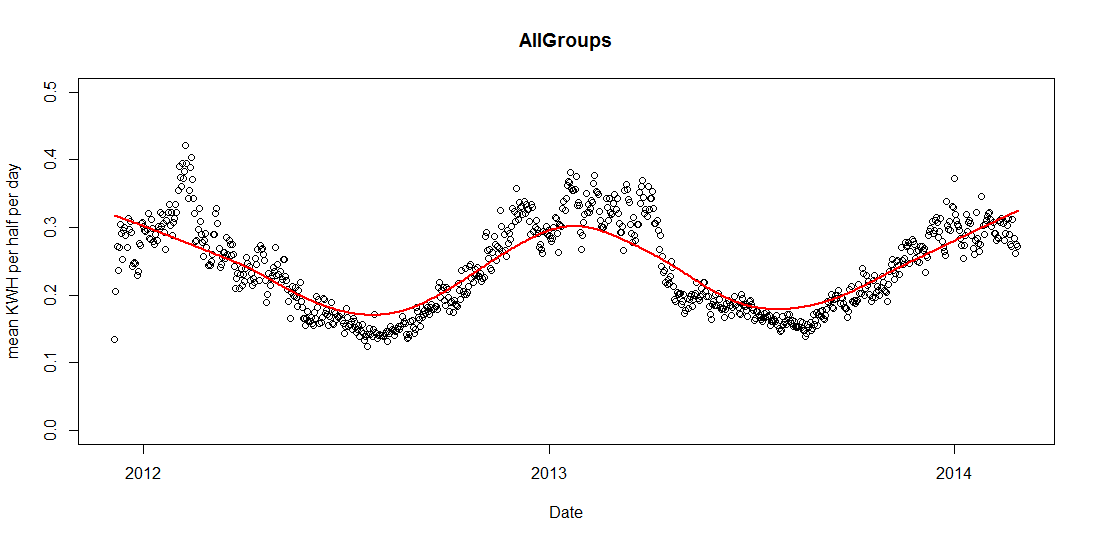
**stdorToU**: Tariff.

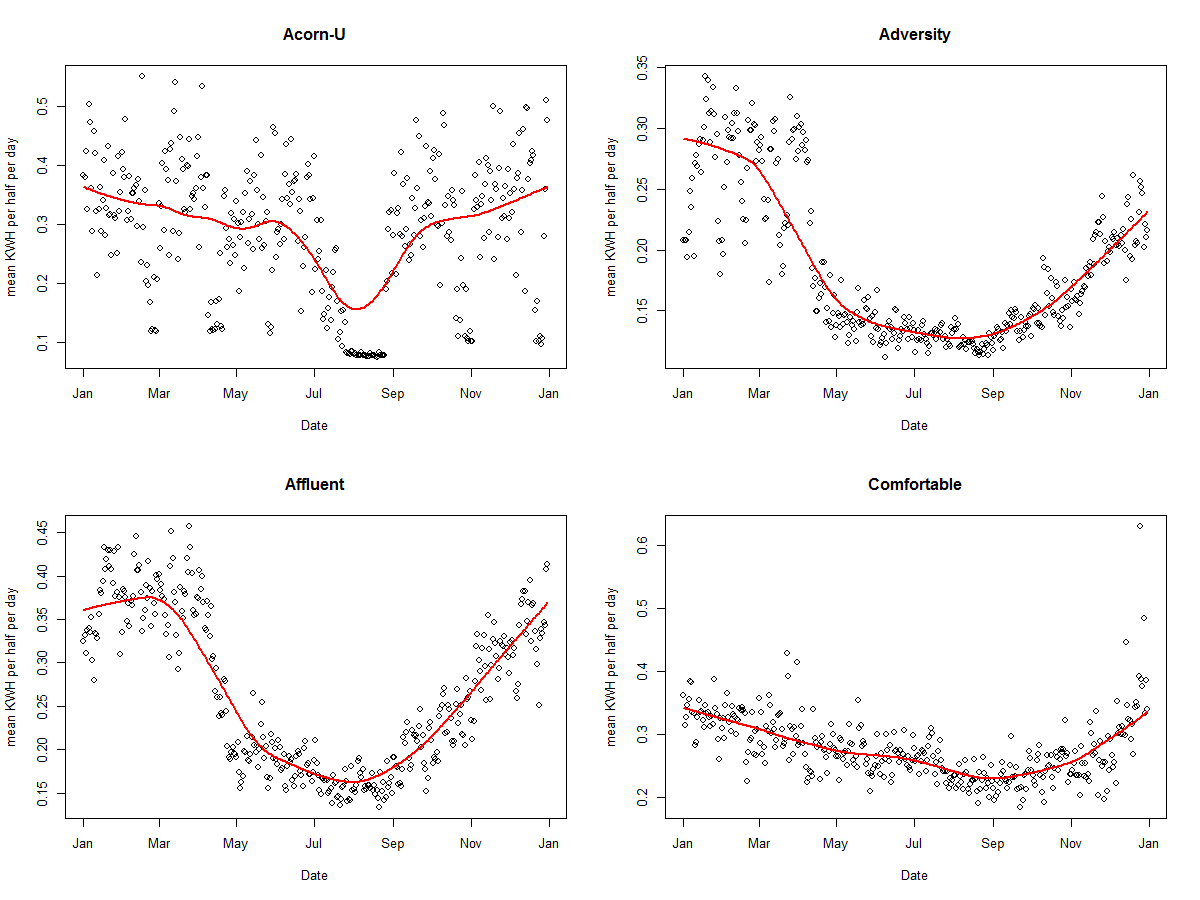
**KWH**: energy used KWH per half hour.  
**Acorn/Acorn grouped**: the grouping of the customer.

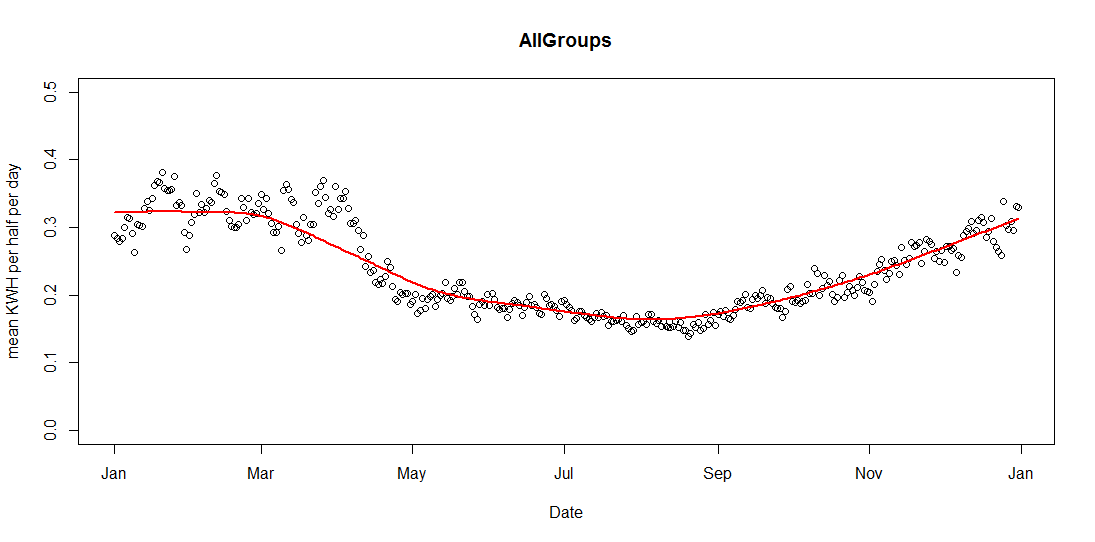
The data set can be found here <https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>. Details on the ACORN groups can be found here <http://acorn.caci.co.uk/downloads/Acorn-User-guide.pdf>.

### **Seasonality**

The plot below illustrates seasonality over the two years that data was collected. We can see that the average KWH usage is higher in the winter than in the summer. On the y axis: the mean KWH per half hour usage per day, on the x axis: the date (in days).

The seasonality of each group varies, however for the sampling model one seasonality functions will be developed that represents all groups. This will be similar to the above graph (the average of all groups). The graphs below show the seasonality of each group in the 2013 year.





### **Building a sampling model**

When building the sampling model the equations for each group will take the form:

b­1\*normal(μ1, σ1) + b­2\*normal(μ2, σ2) + c

Manually fitting two normal curves to the mean KWH per half hour of the ACORN-U group. The blue line shows the fitter combined normal curve, the curve loops around the boundary, this is because hours are continuous – it is time series data. The red line shows a calculated mean values of the data group per half hour. The Y axis shows KWH per half hour, the X axis shows 48 half hourly time intervals which covers one day (24 hours) starting at 00:00:00 and ending at 11:30:00. (The graphs below were generated in the java class DataFitter using the JFreeChart graphing library).

|  |  |
| --- | --- |
|  | **Group: Acorn-U**  b­1 = 3.8  μ1 = 40.0  σ1 = 3.0  b­2 = 6.0  μ2 = 25.0  σ2 = 7.0  c = 0.1 |
|  | **Group: Adversity**  b­1 = 1.0  μ1 = 1.5  σ1 = 1.5  b­2 = 2.0  μ2 = 20.0  σ2 = 14.0  c = 0.12 |
|  | **Group: Affluent**  b­1 = 6.0  μ1 = 48.0  σ1 = 9.0  b­2 = 1.5  μ2 = 24.0  σ2 = 5.0  c = 0.15  discrepancy in normal curve \* |
|  | **Group: Comfortable**  b­1 = 5.0  μ1 = 40.0  σ1 = 5.0  b­2 = 4.0  μ2 = 22.0  σ2 = 6.5  c = 0.1 |

**\*** In the Affluent graph there is a small discrepancy in the normal line. This is because the equation wraps around the boundaries up to half of the range in each direction. It occurs when there is a significant different in the middle values (for each wrapped half range), in most cases the difference is negligible because the normal line tends to 0, it has no effect when added. In this case however the stand deviation of the curves is sufficient enough so that the line is not close to 0 and so the addition is noticeable. It will have a very small effect on the data sampling at the centre point but as the curve is an approximation and some error term will be added anyway it is not something to be too concerned about.

### **Proceedings**

After looking at a good representative example of the data that the system will be dealing with (above). It has come to attention that an ANN prediction model may not be the most suitable. Changing it in favour of a simpler and more intuitive model will benefit the final result of the project. Having a simulation of the system including the representative model formed above will allow testing of different prediction models. The selection of models to test does not necessarily exclude an ANN model.

# **Testing Prediction Models**

## **Introduction**

ANNs have their advantages – mention above, but they also have their disadvantages. They are essentially a black box meaning that the interpretability of their results is very bad. I will be advantageous in a business oriented environment to have at least some interpretability. There is an opportunity to test the use different machine learning techniques with the success of the data generation module. If a prediction model can be found that works as well or better than an ANN model then it should be seriously considered. Taking this opportunity, the next section will begin to test some alternative machine learning models.

It should be noted that this testing is completely separate from the multi-agent system as generating data through a simulation is unnecessary. Data will be generated similarly to the *TestCombinedNormalHouseModel* class in the *test* package – it will be generated on mass. A dedicated mass data generator will provide enough data to perform the tests. Different combinations of house models will be used to see if they will affect the prediction performance of the alternative methods. The testing will be done using R and using time series packages (more information on the specific package used in each respective section).

# **Bibliography**

|  |  |
| --- | --- |
| [1] | Smart Energy GB, “About the rollout,” [Online]. Available: https://www.smartenergygb.org/en/the-bigger-picture/about-the-rollout. |
| [2] | C. Yilmaz, S. Albayrak and M. Lutzenberger, “Smart Grid Architectures and the Multi-Agent System Paradigm,” in *Proceedings of the 4th International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies.*, Chamonix, 2014. |
| [3] | M. Wooldridge, An introduction to multiagent systems, John Wiley & Sons, 2009. |
| [4] | S. Russell and P. Norvig, Artifical Intelligence. A modern apprach, Prentice Hall, 1995. |
| [5] | A. Rogers, S. D. Ramchurn and N. R. Jennings, “Delivering the Smart Grid: Challenges for Autonomous Agents and Multi-Agent Systems Research,” in *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, Toronto, 2012. |
| [6] | I. Rahwan, R. Kowalczyk and H. H. Pham, “Intelligent Agents for Automated One-to-Many e-Commerce Negotiation,” *Autralian Computer Science Communications,* vol. 24, no. 1, pp. 197-204, 2002. |
| [7] | PyData, “Alex Rubinsteyn: Python Libraries for Deep Learning with Sequences,” 4 December 2015. [Online]. Available: https://www.youtube.com/watch?v=E92jDCmJNek. |
| [8] | B. Oancea and S. C. Ciucu, “Time series forecasting using neural networks,” *arXiv,* p. 1, 2014. |
| [9] | nelsonminar, “Energy usage notes,” 31 Janurary 2012. [Online]. Available: https://nelsonslog.wordpress.com/2012/01/31/energy-usage-notes/. |
| [10] | M. S. Narkhede, S. Chatterji and S. Ghosh, “Multi-Agent Systems (MAS) controlled Smart Grid - A Review,” in *Proceedings on International Conference on Recent Trends in Engineering and Technology*, 2013. |
| [11] | T. M. Mitchell, Machine Learning, MIT Press, The McGraw-Hill Companies Inc, 1997. |
| [12] | S. Miller, “Mind: How to Build a Neural Network (Part One),” 10 August 2015. [Online]. Available: http://stevenmiller888.github.io/mind-how-to-build-a-neural-network/. |
| [13] | MachineLearner, “CS231n Lecture 10 - Recurrent Neural Networks, Image Captioning, LSTM,” 14 June 2016. [Online]. Available: https://www.youtube.com/watch?v=iX5V1WpxxkY. |
| [14] | P. J. '. Hoen, K. Tuyls, L. Panait, S. Luke and J. A. La Poutré, “An Overview of Cooperative and Competitive Multiagent Learning,” *Lecture Notes Computer Science,* vol. 3898, pp. 1-46, 2006. |
| [15] | C. Gaven, “Seasonal variations in electricity demand,” Department of Energy & Climate Change, 2014. |
| [16] | S. R. E. M. F. F. S. J. A. Emma L. Giles, “The Effectiveness of Financial Incentives for Health Behaviour Change: Systemtic Review and Meta-Analysis,” 2014. |
| [17] | B. Davito, T. Humayun and U. Robert, “The smart grid and the promise of demand-side management,” McKinsey and Company, 2009. |
| [18] | C. D. A. L. S. P. A. M. Antonios Chrysopoulos, “Agent-based small-scale energy consumer models for energy portfolio management,” *2013 IEEE/WIC/ACM International Joint Conferences on (Vol. 2, pp. 94-101). IEEE.,* 2013. |
| [19] | B. Alrayes, O. Kafali and K. Stathis, “RECON: A Robust Multi-agent Environment for Simulating COncurrent Negotiations,” *Advances in Agent-base Complex Automated Negotiation,* pp. 157-174, 2016. |
| [20] | H. Allende, C. Moraga and R. Salas, “Artifical neural networks in time series forcasting: A comparative analysis,” *Kybernetika,* vol. 38, pp. 685 - 707, 2002. |
| [21] | British Gas, “Smart Meters,” 2016. [Online]. Available: https://www.britishgas.co.uk/smarter-living/control-energy/smart-meters.html. |
| [22] | Department of Energy & Climate Change, “Smart Metering Implementation Programme, Smart Metering Equipment Technical Specifications,” Department of Energy and Climate Change, 2014. |
| [23] | “BBC News,” [Online]. Available: http://news.bbc.co.uk/1/shared/spl/hi/guides/456900/456991/html/. |
| [24] | European Nuclear Society, “Base load power plants,” [Online]. Available: https://www.euronuclear.org/info/encyclopedia/baseloadpowerplants.htm. |

1. \* The half hour interval will not be in real time, it will be scaled to simulation speed. [↑](#footnote-ref-1)