Aalto University
School of Science
Degree Programme of Computer Science and Engineering

Hussnain Ahmed

Using Big Data Analytics for Measuring Energy Consumption Patterns:

Applying Big Data for energy efficiency

Master's Thesis
Espoo, June, 2014

DRAFT! — August 6, 2014 — DRAFT!

Supervisors: Professor Professor Matti Vartiainen, Aalto University
Professor Jukka Nurminen, Aalto University
Instructor: Sanja Scepanovic M.Sc. (Tech.)

Aalto University
School of Science
Degree Programme of Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Hussnain Ahmed

Title:
Using Big Data Analytics for Measuring Energy Consumption Patterns: Applying Big Data for energy efficiency

Date: June, 2014 **Pages:** 78

Professorship: Data Communication Software **Code:** T-110

Supervisors: Professor Matti Vartiainen, Professor Jukka Nurminen

Instructor: Sanja Scepanovic M.Sc. (Tech.)

Global energy requirements are continuously increasing. Conventional methods of producing more energy to meet this growth pose a great threat to the environment. CO_2 emissions and other bi-products of energy production and distribution processes have dire consequences for the environment. Efficient use of energy is one of the main tools to restrain energy consumption growth without compromising on the customers requirements. Improving energy efficiency requires understanding of the usage patterns and practices. Smart energy grids, pervasive computing, and communication technologies have enabled the stakeholders in the energy industry to collect large amounts of useful and highly granular energy usage data. This data is generated in large volumes and in a variety of different formats depending on its purpose and systems used to collect it. The volume and diversity of data also increase with time. All these data characteristics refer to the application of Big Data.

This thesis focuses on harnessing the power of Big Data tools and techniques such as MapReduce and Apache Hadoop ecosystem tools to collect, process and analyse energy data and generate insights that can be used to improve energy efficiency. Furthermore, it also includes studying energy efficiency to formulate the use cases, studying Big Data technologies to present a conceptual model for an end-to-end Big Data analytics platform, implementation of a part of the conceptual model with the capacity to handle energy efficiency use cases and performing data analysis to generate useful insights.

The analysis was performed on two data sets. The first data set contained hourly consumption of electricity consumed by a set of different buildings. The data was analysed to discover the seasonal and daily usage trends. The analysis also includes the classification of buildings on the basis of energy efficiency while observing the seasonal impacts on this classification. The analysis was used to build a model for segregating the energy inefficient buildings from energy efficient buildings. The second data set contained device level electricity consumption of various home appliances used in an apartment. This data was used to evaluate different prediction models to forecast future consumption on the basis of previous usage.

The main purpose of this research is to provide the basis for enabling data driven decision making in organizations working to improve energy efficiency.

Keywords: Big Data, energy, smart grid, energy efficiency, hadoop, analytics, machine learning, classification, CIVIS

Language: English

Acknowledgements

I wish to thank all students who use LaTeX for formatting their theses, because theses formatted with LaTeX are just so nice.

Thank you, and keep up the good work!

Espoo, June, 2014

Hussnain Ahmed

Abbreviations and Acronyms

ICT	Information and communication technology
VTT	Valtion Teknillinen Tutkimuskeskus (State Technical Research Center of Finland)
UEC	Unit Energy Consumption
MPP	Massively Parallel Processing
SN	Share Nothing
HDFS	Hadoop File System
SQL	Structured Query Language
ETL	Extract Transform and Load
RAM	Random Access Memory
HDD	Hard Disk Drive
VizQL	Visual Query Language
CDH	Cloudera Distribution including Apache Hadoop
VM	Virtual Machine
CPU	Central Processing Unit
VMDK	Virtual Machine DISK (format)
CSV	Comma separated values
FTP	File Transfer Protocol
API	Application programming interface
ANN	Artificial neural network
MAE	Mean absolute error
FMI	Finnish Meteorological Institutes

Contents

Chapter 1

Introduction

In the modern era, we have seen a phenomenal increase in human dependency on information and communication technology (ICT). ICT-enabled products and services have transformed the way of life on this planet. We need and depend on ICT to fulfil our needs from a basic physiological level to the human desire of being an effective part of society. There are many research areas and opportunities that are emerging as bi-products of this continuous transformation. One of them is the availability of digital traces of human activities. Every time we use these services, we produce digital traces that can be recorded and analysed. Big Data refers to these digital traces of human activity. Ubiquity of computing resources, fast and highly mobile connectivity and the advent of social media has caused a great surge in the data volumes. Realising the true potentials of data, businesses are not only utilizing it as a source of decision making, but as a new revenue stream. Emerging large scale opportunities are reshaping the business models of many companies around the globe.

To support this transfiguration, we have seen a rapid development in distributed parallel computing, data communication software and machine learning. Industry giants such as Google and Yahoo have open sourced technologies and tools e.g. MapReduce and Hadoop to facilitate these advancements. Open source software communities like Apache Software foundation have further developed these tools to provide a complete ecosystem for handling Big Data and generating useful insights. The new specialized Big Data companies such as Cloudera and Hortonworks have emerged as the catalyst for this data revolution. In this research, we try to formulate a model for an end-to-end Big Data analytics platform based on these technologies that can ingest data from heterogeneous sources, process it in an efficient way, mine the data to generate insights based on business logic and then present the information using interactive visualisations. This practical part of the research includes the development as well as implementation of the mentioned Big Data platform to perform the analyses on real life use cases and generate useful

insights. The model is based on open source software components available free of charge. There are other closed source software alternatives that can fit into the presented model, but the discussions about these solutions are not included in this document.

The topic of research is inspired by the European Union's *"Cities as drivers of social change"* (CIVIS) project under the seventh framework. The CIVIS project focuses on the adoption of ICT tools and techniques for integrating social aspects of city life into production, distribution and consumption of energy. It aims to make city life as a functional unit for improving energy efficiency. The use of pervasive ubiquitous computing is driving the smart energy solutions. The smart energy devices as part of this ecosystem generate high volumes of data. This data needs to be instantaneously transferred, stored, analysed and visualised for knowledge discovery and improvements of services. The platform that was developed as part of this endeavour has the capability to automate the whole process.

The data from smart energy devices was analysed to detect the usage patterns and classify buildings on the basis of energy efficiency. Evaluation of some prediction models for energy consumption of household appliances was also included in the scope of research. These use cases provide the basis for designing, planning and implementing schemes for improving energy related services for achieving higher efficiency in both production and usage. The insights generated from these use cases can also help in educating the consumer about the benefits of energy efficiency and spread awareness about behavioural changes from which the society and the individuals can benefit.

This research is also supported by Technical Research Centre of Finland (VTT) as part of their Green Campus initiative. This project focuses on use of ICT based solutions for management and control systems to optimize energy consumption without compromising the indoor environment of the buildings. VTT is also a supporter and partner of the CIVIS project. VTT has installed specialized smart devices in selected test sites. VTT has contributed to our research by providing the data generated by these smart devices. VTT has also helped in scoping the use cases for energy efficiency with the experience and the knowledge they have gained from the related projects and research.

In a nutshell, this thesis focuses on providing a solution for collecting, storing, analysing and visualising data generated by smart energy device for generating insights about energy consumption patterns and discovering the performance of different building units in terms of energy efficiency. The data analysis part of our research provides the models for knowledge discovery that can be used to improve energy efficiency at both producer and consumer ends. The Big Data analytics platform developed as part of this project is not limited to only being used for energy efficiency. It has the capability of handling other Big Data uses cases.

However, within the scope of this document we discuss its use for energy usage patterns' detection and efficiency.

1.1 Problem statement

Energy efficiency can help to curtail production of energy to meet growth in demand. This in turn can help to reduce CO_2 emissions. To achieve this goal we need to understand and improve the energy efficiency at both producer and consumer ends. ICT enabled smart energy grids and devices are being installed globally to measure energy consumption and improve energy efficiency. These smart devices produce large volumes of data. The data generated by different devices is in different formats. For the purpose of knowledge discovery, this data needs to be collected, stored and analysed. The extracted insights from the analysis need to be visualised for easy and effective understanding. The challenge gets even tougher when data needs to be collected and analysed in real time. Then with the time, volume of data and scope of analysis is expected to increase. In order to respond to the above mentioned challenges, a highly scalable and flexible data analysis platform is required that can automate the whole process. This platform needs to be very cost effective for global adaptation.

In the scope of this research we provide a model for Big Data analytics platform that can provide the solution to meet these requirements. We also implement the proposed model and test it with real life data from smart energy devices. The proposed solution is based on open source components that can be deployed on general purpose hardware that can be procured very easily and inexpensively. The proposed platform can be scaled according to data requirements and additional functional components can be integrated as per the scope of analysis. The data analysis within our research also provides advance analytics models to extract the information based on energy efficiency use cases from large volumes of data.

1.2 Helpful hints

For referencing, we have used the Vancouver system [39]. For discussing from authors' point of view, we use Author(s)' name(s) along with reference numbers that refer to the bibliography. In case of quoting the authors, we use double quotation marks and italic fonts e.g. *"quotation from the author"*.

Throughout the document, we discuss the energy. Due to our main focus, the term *energy* in our research and within this document refers to electricity or electric power. In the case of all other types of energy we specifically mention the type name along with energy as a generic term.

In this document we discuss about the concept, development and use of a Big Data platform as our main environment for data analysis within our research. The terms: platform, data platform, and Big Data platform refer to the same concept. In the case of a specific need of any other platform, we provide proper descriptions.

The mathematical text and the code snippets are differentiated from the rest of the text using variation in font such as $for\ mathematical\ text$ &

```
code snippets font.
```

1.3 Structure of the thesis document

The main body of this document is divided into seven chapters. First chapter provides the introduction about the topic of research and the problem that we are trying to solve. The second chapter explains the main theoretical concepts and motivation for this research. We thoroughly discuss the other similar research endeavours in this chapter and their respective linkages to our research. The third and fourth chapters describe our methodology and practical implementation steps respectively. The fifth chapter provides the details of the data analysis and the results that we produced. Chapter six presents a critical analysis of our approach and results. It also provides some possible directions for further research on this topic. The last chapter provides a summary of what we have achieved from our efforts and how can it be used in the real world applications.

Chapter 2

Background

This chapter describes the main motivation and theoretical background behind our research. In a systematic stepwise approach we list and describe the main topics. We start with the motivation, inspiration and the partners of this thesis and then we explain the theoretical concepts with reference to previous work done on the respective topics. For each topic we also describe how it has contributed to our research.

2.1 Smart grids

The energy industry across the globe is facing numerous challenges. There are huge pressures from regulatory authorities and environmental organizations to reduce their carbon footprint, expand their renewable energy portfolios, and to take energy conservation measures. The demand response (DR)[1] and its impacts on consumer behaviour requires rapid adaptations in energy service providers' business models. According to United States Federal Energy Regulatory Commission (FERC):

"Demand response can provide competitive pressure to reduce wholesale power prices; increases awareness of energy usage; provides for more efficient operation of markets; mitigates market power; enhances reliability; and in combination with certain new technologies, can support the use of renewable energy resources, distributed generation, and advanced metering. Thus, enabling demand-side resources, as well as supply-side resources, improves the economic operation of electric power markets by aligning prices more closely with the value customers place

[1]Demand Response(DR); Changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardised [13].

on electric power"[16].

Traditionally, power system participants have been strictly producers or consumers of electricity. The demand response and reliability issues with conventional electric power distribution models on the consumer side are causing a major trend in motivating consumers to produce electricity at a domestic level mostly using the renewable energy production methods. "Prosumer" is an emerging term used for an economically motivated entity i.e. [22]:

- Consumes, produces, and stores power,
- Operates or owns a power grid small or large, and hence transports electricity, and
- Optimizes the economic decisions regarding it.

The current energy grids support unidirectional distribution models and are centralized in nature. They have very limited ability to handle the prosumer needs. Line losses and hierarchical topology makes them less reliable. They usually become bottleneck when rapid adaptations are required for the demand response. In [19], Farhangi defines smart grids as:

"The next-generation electricity grid, expected to address the major shortcomings of the existing grid. In essence, the smart grid needs to provide the utility companies with full visibility and pervasive control over their assets and services. The smart grid is required to be self-healing and resilient to system anomalies. And last but not least, a smart grids needs to empower its stakeholders to define and realize new ways of engaging with each other and performing energy transactions across the system".

2.2 The CIVIS project

CIVIS refers to the European Union's project for *"Cities as drivers of social change"* under the seventh framework. It is part of the programme for optimising energy systems in smart cities. The CIVIS project is a collaborative effort of 10 European Universities [2]. It aims to embed the social aspect into the advancements of energy technology. To unleash the full potential of this vision, smart grids need to be coupled with broader social and cultural considerations and understood as

[2]1. Associazione Trento RISE, Italy 2. Aalto university, Finland 3. Imperial College London, UK 4. ENEL Foundation, Italy 5. Instituto Superior Tecnico, Portugal 6.Karlsruhe Institute of Technology, Germany 7.Kungliga Tekniska Hogskolan, Sweden 8.SANTER REPLY SpA Italy 9.Nederlandse Organisatie voor toegepast Natuurwetenschappelijkonderzoek, Netherlands 10. Delft University of Technology,Netherlands

complex socio-techno-economic systems with multiple decision making layers that are in effect at the physical, cyber, social, and policy making levels [18].

ICT acts as one of the main enablers for the smart grids. ICT also provides a lot of new mediums for the social aggregation e.g. internet based social media. The CIVIS project tends to connect these two different dimensions with innovative ICT solutions. An integrated approach to energy efficiency is the basic manifesto of the CIVIS project. [18]

Understanding energy usage patterns and benchmarking energy efficiency performance of small units within cities are some preliminary items in the list of the CIVIS project objectives. Within the scope of our research we analyse energy data to understand the consumption patterns and evaluate various factors that can effect directly or indirectly on the usage patterns. We also try to classify the buildings on the basis of their energy efficiency and try to test the sensitivity of energy efficiency with respect to the ecological factors that can cause shift in usage patterns. For the CIVIS project aim of social aspect integration, we also present an ICT application framework that can be used to collect and analyse social media data. However the analysis of that data is not within the scope of this document.

2.3 The Green Campus initiative

The Green Campus initiative is a project by "Technical Research Centre of Finland" (VTT). It is part of the EcoCampus 2030 program. The EcoCampus is an attempt to increase energy efficiency in districts and buildings by innovative management and control systems capable to optimize the local consumption without compromising the indoor environment, occupant comfort and building performance, and by introducing new ICT enabled business models [28]. The vision of the program is to realize a net zero energy model for a world class research, development and educational facility. Program focuses on co-designing this model with user by educating them and then collecting feedbacks for improvement. The main aim is to improve the energy efficiency of the building units and enable them to become self sustainable for the future. As a consequence, this performance based ecosystem can help both consumers and producers to adapt with the demand response.

The Green Campus initiative is a pilot project for the EcoCampus 2030 program. VTT has installed smart devices inside Aalto University campus buildings in the cities of Espoo and Helsinki. These specialized devices are equipped with smart meters for energy consumption and indoor environment monitoring sensors. The data used for analysis in our research was collected from a subset of buildings used as test sites for this project. The data includes hourly consumption of electricity and electricity used for heating. For one of the test sites VTT has provided

us with the data of device level energy consumption details i.e. electricity used by different home appliances. This was achieved using smart NIALM-[3][24] meters that can distinguish between different electric devices used on the basis of their signal thumb print.

Apart from providing the data, the researchers from VTT's Green Campus initiative have also helped us in formulating the use cases for this thesis research.

2.4 Big Data analytics

Big Data analytics is the application of advanced data analytics techniques on large volumes of data. Advanced analytics is a generalized term used for data analysis techniques: statistical analysis, data mining, machine learning, natural language processing, text mining and data visualisation etc. [41]. Although the data volume is a widely used factor for qualification of the Big Data, when it comes to Big Data analytics there are a few other important attributes i.e. variety, velocity, valuation and veracity. The concept of the 3V's (volume, variety and velocity) of data was first given by an analyst, Doug Laney from Gartner in a 2001 MetaGroup research publication, "3D data management: Controlling data volume, variety and velocity" [31]. Gartner used this concept to formulate a data magnitude index that can support decision making for selection of the solutions for tackling Big Data challenges. This concept is shown in Figure 2.1.

Numbers 0 to 3 represents the scale of the data that can be perceived on each dimension. Adding them together for a Big Data case can provide the data magnitude index. This method provides some basis for quantifying the data as Big Data, however it does not provide a definitive model as it allows presumptions to scale the data. Valuation and veracity are two other factors that are being used widely along with Gartner's 3V's. Valuation supports the decision making by considering the value of outcomes against the efforts required to collect, manage, process and analyse large amounts of data. While veracity refers to ambiguity in the data that can cause complexity. There is no standard definition of Big Data but most of the attempts to define Big Data can be associated with these five factors that we have discussed.

As a matter of fact, we are not attempting to provide a definition of Big Data as part of this study or stating any criteria for qualification of a data set as Big Data. Instead we are proposing an advanced analytics model that should be capable enough to handle Big Data as well other smaller data sets. The modular architecture of the model platform can be tweaked to handle volume, variety, velocity,

[3] NIALM stands for non-intrusive appliance load monitoring, is a process for analysing changes in the voltage and current going into a house and deducing what appliances are used in the house as well as their individual energy consumption

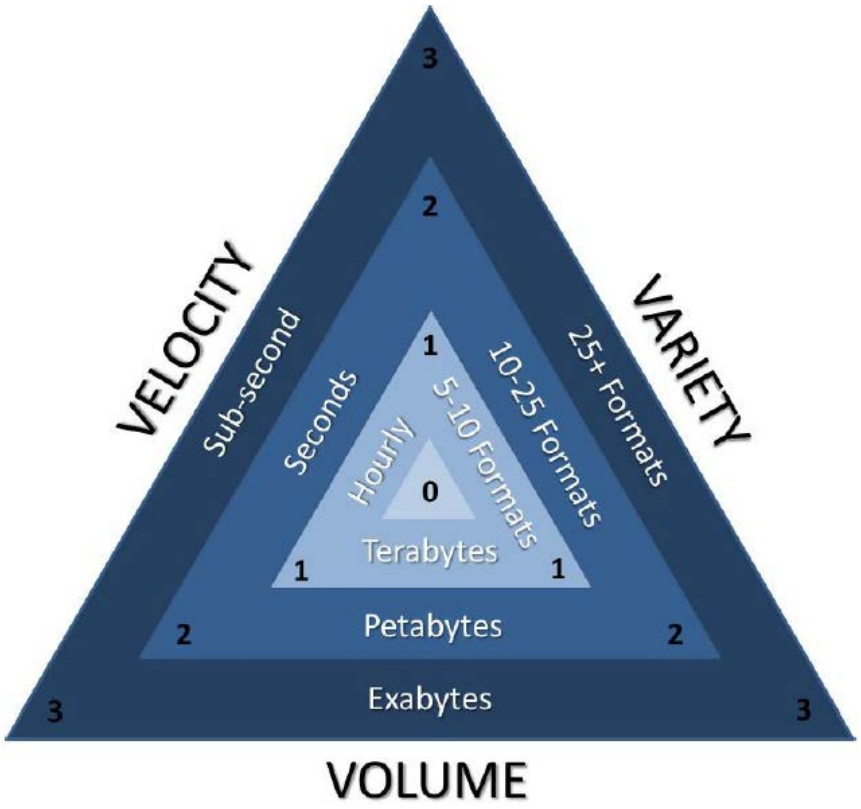


Figure 2.1: Gartner 3V's of data and data magnitude index [31].

and veracity based on the requirements while trying to maximize the valuation for the use case. In the following subsections we discuss some of the relevant technological advancements that enable handling of the mentioned challenges of Big Data analytics. These concepts, tools and techniques are also used in developing the data analytics platform and performing the analysis for our thesis research.

2.4.1 Parallel batch processing with Hadoop

It is hard to predict the size of data and computing power required to process the data when dealing with Big Data. Scaling up [4]is an option that is always bounded by some maximum capacity limits. Also specialized hardware to scale up for higher capacity usually cost much more than general purpose hardware. So

[4]When the need for computing power increases, a single powerful computer is added with more CPU cores, more memory, and more hard disks and used in parallel.

the viable option is to scale out [5] using the required number of smaller machines with relatively low computing resources in parallel. From programming point of view managing parallel running processes on different machines while ensuring low failure rate, is a tough job. So the desired system should provide programmers an abstraction from lower level system details to enable rapid and fault tolerant development for Big Data applications. MapReduce is a parallel batch processing framework developed at Google for the purpose of web indexing. The concept of MapReduce was published by Jeffrey Dean and Sanjay Ghemawat in 2008 within their research paper "MapReduce: simplified data processing on large clusters" [17]. This paper describes MapReduce as:

"a programming model that provides a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Programs written in this functional programming style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program's execution across a set of machines, handling machine failures, and managing the required inter-machine communication".

Hadoop is the open source software framework whose main components are drived from MapReduce. It was developed by Doug Cutting and Mike Cafarella. It was initially created in 2005 to support an open source search engine but then adapted to the published MapReduced framework [17]. It was released by the Apache foundation. Apache has also built various supporting tools around Hadoop framework to support end-to-end Big Data analytics ecosystems e.g. Apache flume for data collection, Hadoop File system (HDFS) for storing, Apache Pig and Hive for processing, Apache Mahout for machine learning.

Hadoop is a batch processing framework that empower processing of large volumes of data using commercial grade low cost computing infrastructure. So they support volume and valuation directly. Variety can also be supported by different file formats in HDFS. Veracity is subjected to supported tools like data collection or data mining tools. Support for such tools is available in Apache Hadoop e.g. Flume, Mahout etc. Velocity however is the only feature that a batch processing framework like Hadoop cannot handle. The next subsection answers the question of velocity.

[5] When the need for computing power increases, the tasks are divided between a large number of less powerful machines with (relatively) slow CPUs, moderate memory amounts, moderate hard disk counts.

2.4.2 Real time Big Data processing

Real time data processing is generally associated with live streams of the data. The real time data can be processed and analysed on arrival or it can be buffered for small intervals to provide near to real time analysis. However in many modern data applications instantaneous data needs to be analysed in the context of large volumes of historical data. To apply advanced analytics models such as machine learning, active feedback loops are also necessary. Even for stored (non live data) Big Data, applications require data processing systems to answer queries very fast. To fulfil these industry-driven requirements technology is in rapid advance mode. In the last twelve to eighteen months we have seen software like YARN (Hadoop 2.0), Storm, Spark, Shark, Cloudera Impala etc. with near to real time processing capabilities. On top of it, tools like Mlbase and Cloudera Oryx have started to enable real time advance analytics. Most of these systems, frameworks and tools are being developed as the evolution path for Hadoop. All of them have their own purpose, strengths, and limitations. They are mostly used in combinations depending on the use cases. We are not discussing or comparing these systems and tools. Instead, in this article, We briefly discuss the two prevailing architectural constructs that can enable real or near to real time Big Data processing.

2.4.2.1 Lambda architecture

Lambda architecture presents a hybrid model by using fast stream processing together with relatively slow parallel batch processing. It was developed by Nathan Marz on the basis of knowledge and experience he gained from his work with large data sets at Twitter Inc. His approach decomposes data processing systems into three layers i.e. a batch layer, a serving layer and a speed layer. The stream of data is dispatched to both the batch and speed layers. The batch layer manages the historic data set and pre-computes the batch views. The serving layer indexes the batch views so the queries can be served with low latency as compared to traversing through the complete data set. The speed layer deals with the recent data thus compensates for the change of data sets during updates of serving layer. An answer to the query is the merged view, batch view, and the real time view [37][5].

Figure 2.2 below shows the Lambda architecture. Lambda architecture can be implemented using combination of systems and tools e.g. Apache Hadoop along with Apache Storm.

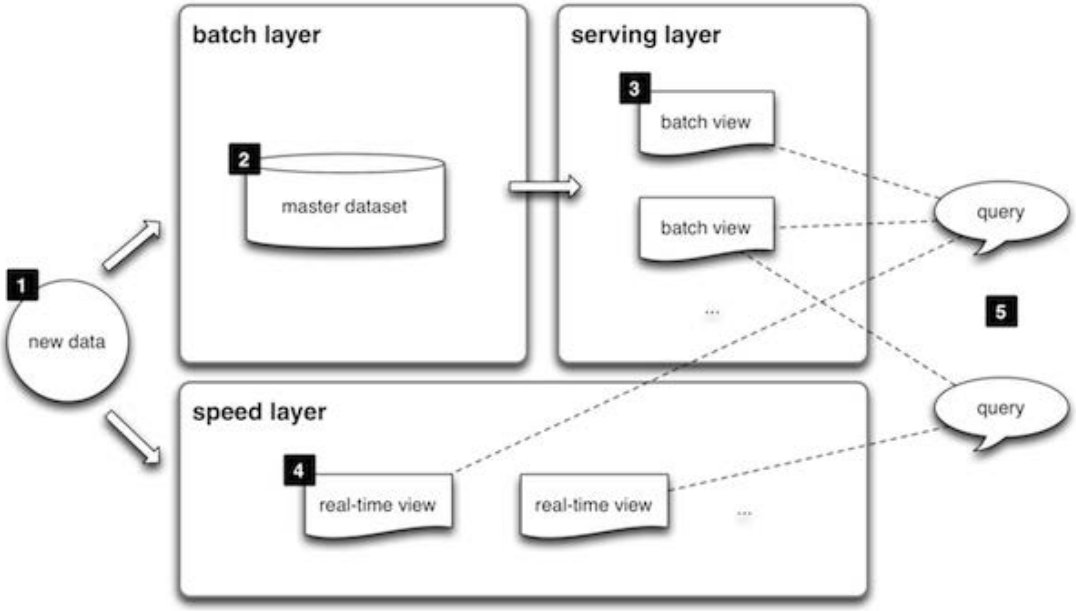


Figure 2.2: Lambda architecture [5].

2.4.2.2 Massively parallel processing - MPP databases and query engines

MPP based architectures use multiple independent computing resources like servers, processors and storages to execute processing jobs in parallel. Most of the MPP based database approaches implement shared nothing (SN) architecture i.e. a distributed computing architecture in which each node is independent and self sufficient and there is no point of contention across the system. The SN concept for databases was first presented by Michael Stonebraker at the University of California, Berkeley in 1986 [42]. The SN databases have been very popular in commercial application primarily because of the high scalability offered by this architecture. Teradata Warehousing Solutions has been using SN database architectures extensively. Greenplum is an example for open an source SN database.

Despite high scalability and other positive aspects, SN databases need a lot of manual work in terms of partitioning the data, tuning the data and load balancing etc. Maintaining such database systems can be expensive. MapReduce and Apache Hadoop ecosystem provide a high level of automation, along with scalability, flexibility and fault tolerance. However parallel batch processing is not as fast SN based MPP databases. Merging of both models can solve all these issues. Cloudera Imapala is one example of the MPP based on-line query engine that runs natively on top of Hadoop [1]. It can provide MPP like query response time with

processing power and flexibility of the Hadoop. For our research we have used Cloudera Impala for handling near to real-time velocity for Big Data processing.

2.5 Energy efficiency and eco-effeciency

In the introductory chapter and sections 2.1, 2.2, 2.3, we have highlighted the importance of energy efficiency. We discussed the advancements in pervasive smart energy devices, grids and their role in improving energy efficiency. We have also discussed the need for collecting and processing large volumes of data from smart energy devices and the available solutions. In this section we explain the main motivation and the theoretical concept behind data analysis.

Unprecedented challenges arising from increasing dependency on conventional energy are part of a global phenomena. Like other economies, countries in the European Union are also putting a lot of focus on energy efficiency to ensure energy supply security by reducing primary energy consumption and decreasing energy imports. It helps to reduce greenhouse gas emissions in a cost- effective way and thereby to mitigate climate change [2]. Member states agreed to reduce 20% of the EU's primary energy consumption by 2020 in the council of European Union in March, 2007. The EU's Energy Efficiency Directive 2012 [2] defines energy efficiency as the ratio between output of performance, service, goods or energy, and the input of energy. This definition was first discussed in 2006 in the European Commission's action plan for energy efficiency. This generic definition covers all major aspects of the energy efficiency i.e. production, distribution, consumption and the value created in comparison to the resources consumed during the whole process. However, to develop a methodology for measuring the energy efficiency and to evaluate the savings, the project "Measuring and potentials of energy efficiency (EPO)" was started in 2008 [12]. As part of this project VTT published a report: "Measuring energy efficiency Indicators and potentials in buildings, communities and energy systems"[21]. This report presents the model for calculating energy efficiency and its correlation with environmental factors. VTT's research presented in this report considers energy efficiency as a subset of larger eco-efficiency. The ecological factors that can affect energy efficiency are e.g. Temperature, CO2, NOx,SO2 etc. The ecological efficiency itself is a way of measuring sustainable development. VTT summarizes the whole ecosystem in Figure 2.3 below.

The concept of eco-efficiency provides the basis for data analysis in our research. We have applied basic and advanced analytics techniques on data sets collected from building units that are part of VTT's Green Campus initiative pilot project with consideration of the eco-efficiency model presented in VTT's report. We calculated energy efficiency of the buildings on basis of formula deduced in Chapter

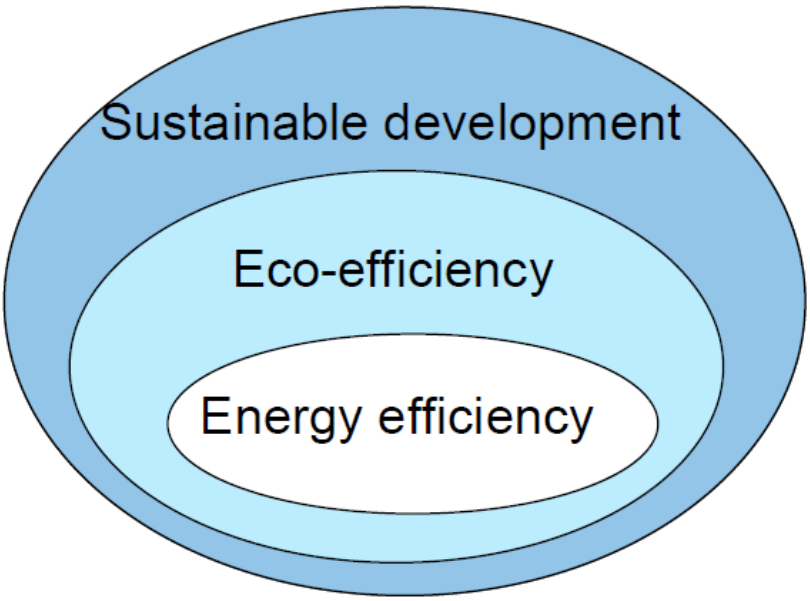


Figure 2.3: Energy efficiency, eco-efficiency and sustainability[21].

5 (equation 5.1 and 5.2) of VTT's report [21].

$$Energy\ efficiency\ of\ a\ building = \frac{Energy\ consumed}{Built\ area} \tag{2.1}$$

In case of a specific energy consumption (SEC) [21] equation 2.1 can be written as

$$SEC = \frac{Q}{A} \tag{2.2}$$

Where Q denotes the consumption for a single energy type for example electricity and A is the built area in square metres. In subsequent sections we shall be referring to these equations when we try to identify the usage patterns at building level, discuss the relevance of energy efficiency and then discuss a model for classifying buildings by energy efficiency .

2.6 Daily consumption patterns, base load and user load

Daily consumption patterns of a building unit corresponds to the respective usage of the building. Understanding daily usage patterns can help in identifying the optimization points for improving the energy efficiency of that building unit.

The base load of a building is one important metric that can be detected through observing the daily consumption. The base load is the consumption that takes place regardless of the actual use of the building and of the user's energy consumption [21]. It is the permanent minimum load that a power supply system is required to deliver. The base load is usually caused by the continuous consumption for building maintenance like air conditioning, ventilation, or night time lighting. Sometimes the base load also includes energy consumption by functional components inside building like computer servers, lab equipment, and refrigerators etc. However VTT differentiates the base load from the user energy load that is characterized by the direct involvement of the users of a building. For example an office building has the peak load during the day time because users are using various additional appliances like personal computers, coffee makers, lights etc. compared to the base load that is generated during the night time when the office building is not in use. Figure 2.4 illustrates the concept of base load and user load.

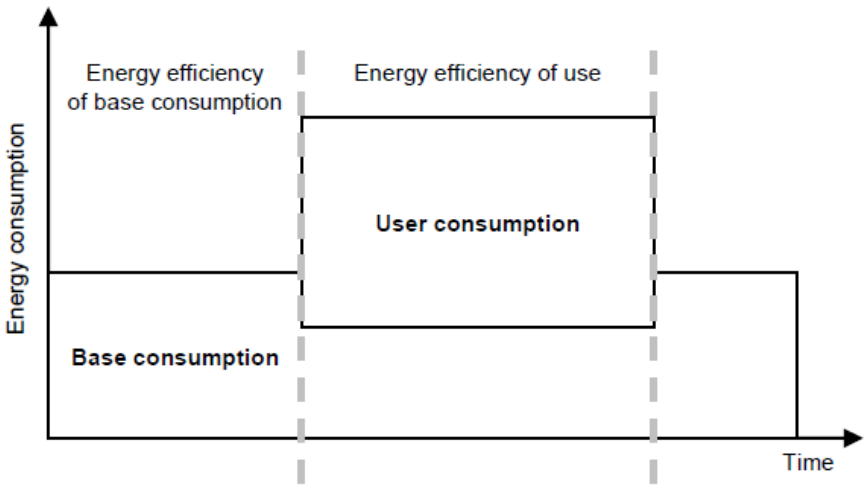


Figure 2.4: Base load, user load and energy efficiency [21].

In the Figure 2.4, the base load is written as base consumption. The energy efficiency of base consumption and the energy efficiency of user load can be calculated using equation 2.1 or 2.2. This provides a weighted metric that can be benchmarked and compared. It can help to narrow down the scope of research by referring to problematic buildings and their issues.

2.7 Energy consumption seasonal patterns

Energy consumption has high dependency on seasonal factors like the weather. Energy consumption trends vary with outside temperature. Among other things, the energy required for the air conditioning in the buildings is a major variable factor dictating the trends. Due to regional weather differences the seasonal energy consumption patterns are also different for different regions e.g. in cold regions of the world energy consumption surges in winter while in warmer regions energy consumption increase is expected in summer. Energy service providers usually conduct demand planning with consideration of seasonal trends. Consideration of seasonal trends is also very important while optimising for gaining energy efficiency.

In scope of our research we have also analysed the seasonal trends. It was not hard for us to perceive the trends while knowing the weather trend for localities of our test buildings. However, the interesting use case in our research was to check the sensitivity of other consumption patterns and analysis results against the seasonal trend. This will be more explained in the later part of this document where we shall discuss the results of our analysis.

Previously, there have been many studies of both daily and seasonal trends in energy consumption. Due to regional differences in trends, many of these studies focused on consumption patterns within a country. Geoffrey K.F. Tso et al.[43] and Yigzaw G. Yohanis et al.[45] study the energy consumption patterns in Hong Kong and the United Kingdom respectively. Building units e.g. residential houses, apartments and commercial offices etc were considered as basic units of analysis. Yigzaw G. Yohanis's methodology resembles most to our approach as he considered ecological factor along with energy efficiency in similar way. As discussed before, the main purpose of VTT's Green Campus initiative under the EcoCampus 2030 plan is to develop a highly efficient model ecosystem for energy production, distribution and consumption that can be expanded further to any scale. Aligned to this goal, we have attempted to provide a data analysis model that is not specific to certain geographic locations. However detailed study is required for adapting such generic models to region specific requirements. In our research we have also attempted to classify the buildings on the basis of energy efficiency, which is explained in next section.

2.8 Classification of buildings based on energy efficiency

Earlier we mentioned that quantifiable energy efficiency through equation 2.1 and 2.2 can be used as a metric for benchmarking and comparison. For energy service

providers, governmental energy regulatory agencies or research institute like VTT, it is very important to identify the problematic consumption units in group of other highly optimized or average performing consumption units. Classification of these units into similarly performing groups can help them to narrow down the focus on problematic units. Sometimes it can also help in understanding the good practices applied by certain consumption units that have improved their energy efficiency performance.

Classification for fault detection analysis of a building energy consumption has been used previously. Xiaoli Li et al. [33] used classification along with the outlier detection mechanism to identify the energy inefficient building. They provide a step wise approach to extract the features (types of energy, trends etc) from the data collected as a time series. Then detect the daily usage patterns using the auto regression technique and pass the results to benchmark against any outlying data point that can refer to faulty behaviour. Imran Khan [29] proposes different clustering techniques to group the buildings with similar level of energy efficiency together. In our research we used a hybrid method using feature extraction and trend detection techniques such as [33] and then applied a clustering technique proposed by Imran Khan [29]. The clustering technique that we use is called K-means clustering. It is explained in the next subsection.

2.8.1 K-means clustering

K-means is an algorithm for cluster analysis. In the context of the machine learning, cluster analysis is an unsupervised task of grouping a set of objects in a way that objects in the same group are similar to each other more than the objects in other groups. The K-means algorithm clusters the set of objects i.e. energy efficiency values in our case into a predefined number of classes. We term these values as data points. K represents the number of clusters or groups that we can set in start of the process. K-means means algorithm was first proposed by Stuart Llyod in 1957 [34] but the K-means term was first used by James Mcqueen in 1967 [35]. There have been many adaptations and optimizations in Lloyd's basic algorithm. K-means algorithm today has many variants like Fuzzy C-means clustering, K-medoids and Spherical means etc. Even for the Lloyd's original algorithm, there has been some modification in methodology. Two very commonly used methods are the Forgy method [20] and the Hartigan-Wong method [25]. In our approach we are using the Hartigon-Wong method. We shall also use some references from Forgy method when explaining the K-means algorithm.

The K-means groups the data points into clusters with logical centre points. The aim of the K-means algorithm is to divide data points within certain dimensions into K clusters so that the within-cluster sum of squares is minimized [25]. Let's assume if we want to have the K cluster for data points D = $\{x_1, x_2, \ldots, x_n\}$

in d dimensions then

$$x_i \in R^d$$

The K-means algorithm uses following steps to cluster data into groups [40].

1. Initialize the centroids randomly for each K i.e. for each group.
2. Data points are assigned to the closest centroid.
3. Move the centroids to the mean of the data points assigned to that centroid in step 2.
4. Repeat 2 and 3 till convergence. Convergence means that the values have stopped changing for further iterations.

Mathematically randomly initialized centroids are

$$\mu_1, \mu_2, \ldots, \mu_k \in R^n$$

If c^i is the distance of centroid to assigned data point then Step 2 and 3 with recursive distance minimization and mean adjustment can be explained as

For every i, set

$$c^i := \arg\min_j ||x^i - \mu_j||^2 \tag{2.3}$$

The equation above used the Euclidean distance formula for calculating distance between centroid and data point.

For every j, set

$$\mu_j := \frac{\sum_{i=1}^{n} 1\{c^i = j\} x^i}{\sum_{i=1}^{n} 1\{c^i = j\}} \tag{2.4}$$

The input to the K-means is a set of feature vectors along with the number of clusters required. Before inserting data to the K-means, it is required to set the similar scale for features as well set the standard variance to avoid errors in the results. We were required to classify pilot site buildings into four groups with high efficiency, moderate efficiency, low efficiency and poor efficiency classes, So we have set K value as 4.

2.9 Forecasting the energy consumption

Estimating equipment-specific energy consumption has been a key focus area for energy service providers. It can help in demand planning, load forecasting, and understanding end user behaviour. Energy service providers can design better service offerings for their consumers. Unit Energy Consumption (UEC) is a term generally used for estimating equipment specific energy consumption. It is the average annual amount of energy consumed by a user device. As part of the Green Campus project VTT has used state of the art non-intrusive load monitoring (NIALM) [24] devices that can distinguish between the usage of different electric devices on the basis of changes in voltage and electricity.

We are using the data collected by a NIALM device installed in one of the residential apartment included in VTT's pilot test sites. We use auto regression along with the concept of the moving averages in the form of a model known as Auto-regression Integrated Moving Averages (ARIMA) to estimate the future consumption of a device depending on its previous usage. This is an example of quantitative forecasting. Before we go on to discuss about ARIMA models, it is important that we briefly discuss the basic conditions for quantitative forecasting and the time series analysis as the foundation for our prediction model based on ARIMA.

2.9.1 Main conditions and Steps for Quantitative Forecasting

Rob Hyndman et al [26] discuss two main conditions for application of quantitative forecasting in their book "Forecasting: Principles and Practice":

1. Numerical information about the past is available.
2. It is reasonable to assume that some aspects of the past patterns will continue into the future.

In case the conditions can not be met then qualitative forecasting is the only option. However, qualitative forecasting is not in the scope of our research. In the same book authors mention following five step approach for solving forecasting problems.

1. Problem definition.
2. Information gathering that includes statistical data collection.
3. Exploratory analysis of the data to evaluate the structure of the data and observing the relationship between different variables.

4. Choosing and fitting the forecasting model. The model depends upon the relationships between variables. Every model has its own construct. So data needs to be fitted to that construct before applying that model. We discuss it more in the data analysis part of this thesis.

5. Using and evaluating the forecasting model. It generally includes comparison of results after applying different models.

2.9.2 Time Series Analysis

Time series is the sequence of a random variable collected over time. Among other examples of time series data, energy consumption data from metering devices can also be collected periodically hence constituting a time series. Comparison of a single time series at different point in time is termed as time series analysis [15]. A time series usually consists of a deterministic component and a random component[38]. So if X_t is a time series data then we can have

$$X_t = d_t \ + \ \epsilon_t \tag{2.5}$$

Where d_t is the deterministic component and ϵ_t is the random component. The deterministic component itself can be in the form of trends, periods, and jumps etc. Figure 2.5 illustrates the example of different time series. In each illustration there is at least one stochastic random component with and without deterministic components.

In figure 2.5 illustrations 2, 3 and 4 contain a deterministic component with a random component. When forecasting for such cases, it is possible to predict even the random component by using the deterministic component. However for stochastic random time series data without any deterministic components it is very hard to predict anything accurately. The time series with no predictable pattern is generally termed as a stationary time series.

2.9.3 Autoregression, Moving Averages and ARIMA Models

Rob Hyndman's book "Forecasting: Principles and Practice"[26] is the main reference for this section.

2.9.3.1 Regression

The concept behind basic regression techniques for forecasting is that we try to forecast a variable 'y' on the basis of another variable 'x'. For example a linear

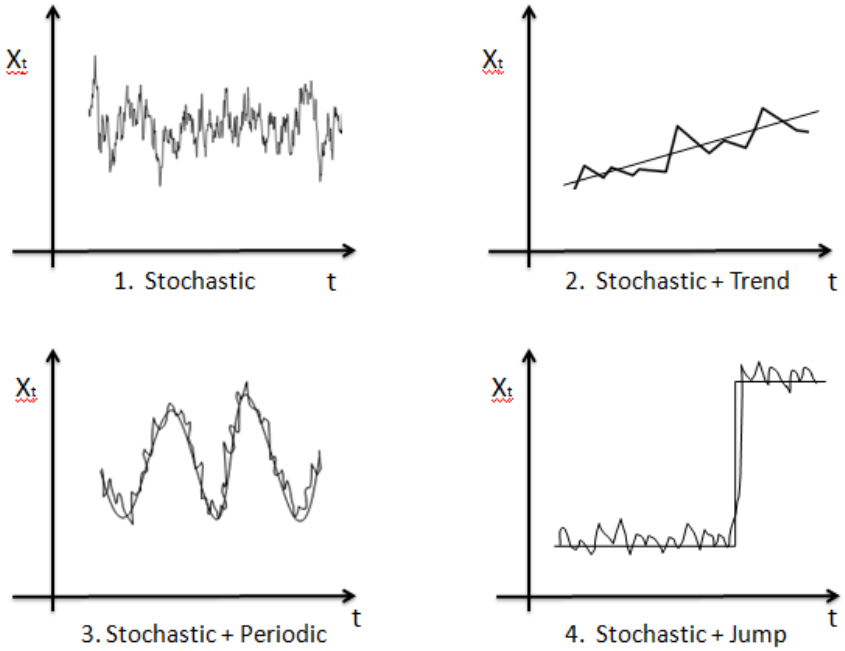


Figure 2.5: Time Series types[38].

regression model forecast y assuming it has a linear relationship with variable x e.g. as in equation below.

$$y = \beta_0 + \beta_1 x + \epsilon$$

Parameter β_0 and β_1 represents the intercept and slope respectively for the line representing the linear relationship. β_0 represents the predicted value when x is 0. Linear regression for time series analysis can be written as

$$Y_t = \beta_0 + \beta_1 x_{t-1} + \epsilon$$

Here Y_t is the estimate with past value of x_t i.e. $\{x_1, x_2, \ldots, x_{t-1}\}$. using differencing [6] error e_t in estimation can be calculated as

$$e_t = X_t - Y_t = x_t - \beta_0 - \beta_1 x_{t-1} - \epsilon \tag{2.6}$$

2.9.3.2 Auto-regression

The auto-regressive model is based on the concept of a variable regressing on itself. For auto-regression we can drive the equation as

$$x_t = \beta_0 + \beta_1 x_{t-1} + e_t + \epsilon \tag{2.7}$$

[6]The differences between consecutive observations

The aim for good estimation is to select values of β_0 and β_1 that can minimize the sum of the square of errors. The above equation can be used to estimate the value based on previous values. But in case we want to estimate based on multiple previous values e.g. 'p' values then we can write it as

$$x_t = c + \beta_1 x_{t-1} + \beta_2 x_{t-2} + \ldots + \beta_p x_{t-p} + e_t + \epsilon \tag{2.8}$$

We just replaced β_0 with a constant c as it is a constant value. Adding the summation to the historic values we can write

$$x_t = c + e_t + \sum_{i=1}^{p} \beta_i x_{t-i}$$

we have also taken out the random component ϵ that does not meet the basic conditions for forecasting as described in subsection 2.9.1. The model presented in equation 2.8 is referd to as AR(p) model.

2.9.3.3 Moving Averages

The moving averages model uses past forecast errors in regression like manner to forecast future time series values instead of using past time series values as in auto-regression. Mathematically, the model can be explained as

$$y_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \ldots + \theta_q e_{t-q} \tag{2.9}$$

OR

$$y_t = c + e_t + \sum_{i=1}^{q} \theta_i e_{t-i}$$

The model presented in equation 2.9 is termed as MA(q) model. In this model each value of y_t can be thought of as a weighted moving average of the past few forecast errors.

2.9.3.4 ARIMA Model

ARIMA stands for Auto-Regressive Integrated Moving Average. As the name suggests it is the combination of the auto-regression and moving average models. ARIMA is one of the most commonly used forecasting techniques e.g. ARIMA is being used widely in stock market prediction software solutions. ARIMA model can handle time series data with and without seasonality. So combining the auto-regression and moving averages using equations 2.8 and 2.9 we can have

$$y'_t = c + e_t + \phi_1 y'_{t-1} + \ldots + \phi_p y'_{t-p} + \theta_1 e_{t-1} + \ldots + \theta_q e_{t-q} \tag{2.10}$$

In this equation y'_t is the difference series. This constitutes ARIMA(p, d, q) model where

- p is the order of the auto-regression.
- d is the number of the non seasonal differences.
- q is the order of the moving averages.

Now to simplify the complex time series equation back-shift notations are usually used e.g. $y_t - 1$ can be denoted by By_t i.e.

$$By_t = y_{t-1}$$

&

$$B(By_t) = B^2yt = y_{t-2}$$

&

$$y_t \ - \ y_{t-1} \ = \ (1 \ - \ B)y_t$$

In general a dth order difference is written as

$$(1 \ - \ B)^d y_t$$

By rearranging equation 2.10 and using back-shift notations, we can have the following equation with labelled p,d and q for the ARIMA model.

$$\underbrace{1 \ - \ \phi_1 B \ - \ \ldots \ - \ \phi_p B^p}_{\text{AR(p)}} \quad \underbrace{(1 \ - \ B)^d y_t}_{\text{d differences}} \ = \ c \ + \ \underbrace{(1 \ + \ \theta_1 B \ + \ \ldots \ + \ \theta q B^q) e_t}_{\text{MA(q)}}$$

The explanation and the equations used in section 2.9.3 were cherry picked from Rob Hyndman's book "Forecasting: Principles and Practice" [26] as reference to theory related to our research.For further details please refer to chapter 5 and chapter 8 of this book.

Fitting the ARIMA model and estimating the future time series values needs intensive computation. We use software e.g. R to solve these equations for our use cases.

Chapter 3

Methodology

In previous chapters we introduced our research problem, listed and explained the solution options with the theoretical background. In this chapter we explain our practical approach for carrying out the research along with the software development required to support the experimentation and data analysis for our research. Following is the list of major tasks for the practical part of our research.

- Understanding energy efficiency, smart grids and available data.
- Requirement engineering and use case preparation.
- Understanding Data Analytics ecosystem, evaluating the Big Data tools and solutions.
- Exploratory data analysis and selection of algorithms and data analysis tools with respect to use cases.
- Development of an end-to-end Big Data analytics platform.
- Data collection, storage and preprocessing.
- Use case specific data analysis and evaluation of the results.
- Visualisation of the results
- Documentation of the research, process, software development and results.

Some of these tasks were required to be performed in a sequential way e.g. requirement engineering and evaluation of Big Data tools were required before developing the Big Data analytics platform or selecting the algorithms. Similarly we needed results before visualisations could be created. On the other hand some of the tasks could have been executed in parallel. For example the documentation was

an ongoing process along with all other tasks. Similarly the literature review for understanding each component of our research was also an ongoing process. Then the iterations were required for continuous improvement.

To tackle these challenges, we needed a methodology that could support sequential and parallel task execution with support for iterations to improve. Like most of scientific research, fail fast and small to move ahead for success was the key for us. Most of the tasks required conceptualization and rapid prototyping. Taking it as a software development task initially, we had some candidate models such as the water fall model, the agile development model, the spiral model and the incremental model etc. Here we shall briefly discuss the advantages and disadvantages in context to our research project.

- **The waterfall model** offered the simplest approach to requirement engineering, design, implement, test and operate our research. However, it is inherently sequential and had weak support for iterations [32].

- **The agile development model** Agile methodology [36] is rapid, iterative and supports quick prototyping but it requires additional communication and management overhead like scrum meetings. Managing it along with stakeholders like VTT and the CIVIS projects was very hard.

- **The spiral Model** is a risk driven process model. It supports prototyping, provides a good way of avoiding major failure risks, and it is iterative [14]. However, it needs a lot of resources during the planning phase especially when the spiral keeps growing in size. It is usually very successful for large projects but it has overheads for small projects like our thesis research. We shall be discussing more about using parts of the spiral model later in this chapter.

- **The incremental model** relies on small incremental steps with each step consisting of independent design, implementation and testing phases [27]. In the beginning, incremental model was the best fit among other candidate models. We were able to prototype small functional units of the Big Data analytics platform very quickly while independently working on the use cases. However during the platform development and data analyses parts it created integration overheads. For example by integrating two different data processing tools together for a single use case, it becomes difficult when they were configured in two different incremental steps.

Learning from the problems that we had faced while using incremental model, we altered our approach to an adapted version of another very flexible software research and development methodology known as "Kumiega-Van Vliet Trading System Development Methodology"[30].

3.1 Kumiega-Van Vliet model

The Kumiega-Van Vliet Trading System Development Methodology ($K|V$) was developed in 2008 for software development required specifically for trading systems. It is the combination of three general purpose software and new product development models i.e. the waterfall model, the spiral model and the stage gate model. We have already explained the waterfall and spiral models. The stage gate model consists of stages e.g. scoping, development, implementation, testing etc. Each stage or combination of stages can be controlled with an approval gate. The process can not move from a stage to another stage if the gate in between them is not approved.This model provides a good control over the development model to ensure quality. However it may cause delays because of the organizational hierarchies dictating the gates.

The ($K|V$) model tries to overcome the short comings of the three models by combining them to a single paradigm for trading system development [30]. In the spiral model, the spirals are divided into four basic steps i.e. research, planning, implementation and test. These four steps can be performed again and again in cycles. To avoid spirals growing too much after each cycle, a stage gate controls if a process can pass to next stage or if it needs to be sent back to perform another cycle in the same stage. Just like the waterfall model, there can be number of stages. However, unlike the traditional waterfall model, in this model we have an an iteration channel for continuous improvement.

3.2 Adaptation of the Kumiega- Van Vliet model

The ($K|V$) model is designed for software research and development in the domain of financial services. With the built in stage gate controls, it requires some scale of hierarchical organizational structure to support the model. For our highly academic research case we have made certain adjustments. The most notable adaptation was to use deliverables and team reviews of respective deliverables as the main control for moving from one stage to the next stage instead of stage gate approvals. The concept of stages like the waterfall model helped in keeping our focus on the solutions for our problem statement. The spiral model cycles enabled us to iterate within a stage and improve the deliverables quality. Typically the decision of additional cycles was based on the feedback during the team review sessions. The inter-stage iteration channels helped us in improving our overall quality. The lessons learnt or the new directions identified during one iteration were included in the scope for the next iteration. It also allowed us to include supplementary topics in our scope without losing focus on mandatory issues.

In our approach, we have divided the complete scope of research in four basic

stages. Within each stage we had four steps. These intra-stage steps were different for each stage. These steps were corresponding to the main tasks that we discussed in the beginning of this chapter. A typical intra stage cycle ended with a set of deliverables. The deliverables were reviewed in a team review session. If required, the other stockholders such as VTT, were also involved in some of the review meetings. We shall be discussing it in detail when we describe our stage wise proceedings. At the end of each review session a decision was made to either move to next stage or try to improve via an additional cycle. Using all four intra-stage steps for additional cycles was not a must. This was another minor adaptation to the $(K|V)$ model. Similarly iterations were mostly initiated after stage three. There were three major iterations. During the iterations change of deliverables were not mandatory. However in practice it was observed that each iteration had caused some major or minor changes in stage deliverables. Having a small and informal team structure reduced our management and communication overheads. This also helped in rapid processing during iterations. Figure 3.1 illustrates our approach with the adapted version of $(K|V)$ model. Stage by stage description of our methodology is explained in the next section.

3.3 Stages, steps and cycles

We have already mentioned that there were four stages with each having four respective steps. Each stage was controlled via deliverables review sessions in a stage gate manner. While inside a stage, steps were executed in spiral cycles. The first cycle of the spiral had to pass through the four steps. The additional cycles were initiated if the further improvements were decided for deliverables in review session. All four steps were not mandatory for additional cycles. In this section we list and describe the stages along with the respective steps. We highlight some major cycles and deliverables. However the iterations will be discussed in the following section. Figure 3.1 will be our main reference throughout this section. In this section we mention some functional components of our project e.g. logical architecture, data processing tools and algorithms.

3.3.1 Stage 1. Conceptualization

At the start, our research problem was mainly concerned about processing the large volumes of the data coming from smart metering devices and understand the consumption patterns. So the primary focus of the conceptualisation stage was to describe our problem in detail, to understand important factors related to it, find and evaluate methods and tools to solve the problem. This stage had the following four steps.

3.3.1.1 Step 1. Understanding

From the beginning, our research had two focus areas i.e. energy consumption and Big Data . The main purpose of this step was to understand important concepts related to these topics. Following were some main activities performed during this step:

- Intensive literature review.
- Participation in CIVIS project Helsinki- Use case workshop 26-27 January 2014. It gave good insights about ecological and social factors effecting energy production, distribution and consumption.
- Participation in VTT's Green Campus initiative introduction session.
- Discussions and informal interviews with VTT's project lead for Green Campus Initiative.
- Aalto University courses.
    1. Scalable Cloud Computing, as a good introduction to parallel batch processing and its uses for Big Data processing.
    2. Information Visualisation, as an introduction to effective communication through data visualisation.

Literature review had been a constant step through out this stage, its cycles, and the iterations it went through.

3.3.1.2 Step 2. Research quantitative methods

This step involved finding and evaluating the various quantitative methods used for measuring energy consumption and benchmarking energy efficiency. Data aggregation methods like daily, monthly consumption, and average consumption were evaluated. Identification and theoretical evaluation of advanced analytical methods was also performed during this step.

3.3.1.3 Step 3. Conceptual model

This step was dedicated for finding available open source solutions to make a conceptual model for an end-to-end Big Data analytics. This step was mandatory for the deliverable of the Big Data platform concept paper. This step was also repeated during various iterations.

3.3.1.4 Step 4. Evaluation of tools

This step was in pair with 3.3.1.3. All the tools listed in the conceptual model were tested during this step. A check-list of evaluated and selected tools was maintained. This list is available in AppendixA.

3.3.1.5 Delivearbles of stage 1

There were two deliverables of this stage

1. Problem Statement: First two steps of this stage were the main contributors for this deliverable.
2. Platform concept document. A document as result of step 3 and 4 of this stage.

3.3.1.6 Stage 1 cycles

In this stage we observed two cycles i.e. a cycle for producing the required deliverables and one additional cycle for the modification of platform concept document. The modification included changes in Big Data platform architecture and in ways of how to depict the conceptual model. Changes in architecture included additional components to handle data variety like MongoDB databases with Apache Pig.

3.3.2 Stage 2. Implementation

This stage mainly includes requirement engineering and intensive software development to prototype and test the Big Data platform described in the concept paper as a deliverable from stage 1. This stage had the following four steps.

3.3.2.1 Use case definition

In this step, based on the knowledge gained from stage 1. we decomposed our problem statement into lower level requirements that can be practically implemented using Big Data platform concept. Use cases went through several iterations. Details of iterations will be discussed later in section 3.4. However here we shall list the final list of use cases.

1. Understanding the seasonal energy usage patterns and their sensitivity with outside temperature.
2. Understanding characteristics of the buildings using daily energy consumption pattern.

3. Calculating the base load of the building to identify non user consumption of buildings

4. Classifying the buildings on the basis of the energy efficiency and analyse the seasonal shifts in this classification.

5. Predicting daily energy consumption of various household devices on the basis of previous consumption pattern.

3.3.2.2 Data analytics platform prototype

This step involved the practical implementation of the platform concept. It covered installation, configuration, customization and integration of selected components as a proof of concept for an end-to-end Big Data platform that can collect, store, process, analyse and visualise data. Details of the components and implementation will be discussed in the next chapter.

3.3.2.3 Data Collection

As mentioned before, real life energy consumption data was provided by VTT. This data was collected by VTT from the smart metering devices installed on test sites. We had prepared our prototype platform to collect this from VTT data repositories continuously in real time. However due to some constraints we were not allowed to integrate our platform with VTT's data repositories. The data was provided to us initially via file transfer from a FTP[1] server. In the later stages, a websevice was opened for us to collect the data. The details of the data will be provided later in this document, however two types of data were collected during different iterations.

1. Hourly consumption of electricity, electricity used for heating, water, and reactive power in a set of buildings as part of VTT's Green Campus initiative.

2. Device level electricity consumption data of home appliances used in two apartments of Aalto Univerity campus residential housing blocks as test cases for VTT's Green Campus initiative.

3.3.2.4 Prototype testing with sample data

This step was only used during the first cycle of this stage and the first iteration of the whole process. The purpose of this stage was to test the full work flow from data collection to data visualisation using the developed prototype. The

[1]The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.

sample data was the randomly selected records from the hourly consumption data set. Although in this step we started with a smaller sample and then kept on increasing it. The complete data set was also tested. During this testing the following functionalities were tested.

1. Data collection.
2. Raw data storage.
3. Data cleaning to produce tidy data set.
4. Data pre-processing. Reducing the large data volume without losing insights.
5. Storing pre-processed data into databases.
6. Testing of advanced analytics tools integrated within our prototype.
7. Data visualisation.

3.3.2.5 Stage 2 deliverables

The following were two deliverables of implementation stage.

1. Use case definition.
2. Working Prototype of Big Data platform concept.

3.3.2.6 Stage 2 cycles

This stage went through two additional cycles on top of the first mandatory cycle. Within the two additional cycles, all the steps were performed except data collection. Data was collected only in the first cycle of this stage. The major revisions inside cycles included; alteration in use cases e.g. effect of external temperature on seasonal energy pattern was identified during one of the review sessions. Within prototype and protype testing the alterations were required to adapt for changes in use cases

3.3.3 Stage 3. Data Analysis

In this stage we used the data platform to analyse the collected data and produce the insights based on the use cases. We applied the basic and advanced analytics techniques introduced in the sections 2.6, 2.7, 2.8, 2.9. This stage has the following four steps.

3.3.3.1 Tight integration of the platform components

In the section 3.3.2.4 we tested all the units of the platform by manually enforcing the process i.e. taking out the output of one module and manually feeding it to other module as the input. In this step we tried to automate the process by coupling the modules together in the form of a single process per use case.

3.3.3.2 Evaluation and selection of algorithms

In this step we tried to find and compare various options of advanced analytics algorithms available for supporting our use cases. It involved quantitative methods considered in section 3.3.1.2. However, the focus was more on the advanced analytics. The techniques explained in the sections; 2.6, 2.7, 2.8, 2.9 were selected during this step. For evaluating the algorithms we were using samples from collected data as our training data.

3.3.3.3 Applying Analytics

In this step we applied the selected algorithm on the complete data set. The insights generated from this step were the main results for our study. During the cycles of this stage, results from this step also affected the evaluation of algorithms in previous step, section 3.3.3.2. Details related to this step will also explained in chapter 5.

3.3.3.4 Result visualisation

For ease of understanding the extracted insights in te previous step, We visualised the results in form of data graphs. Different tools for visualisations were used in this steps. Visualisation tools will be discussed in chapter 4. For data visualisation we tried to implement the graphical practices discussed by Edward Tufte in his book "The Visual Display of Quantitative Information" [44].

3.3.3.5 Stage 3 deliverables

There were two deliverables of this stage.

1. Functional Platform. At the end of this stage, we had a fully functional platform capable of implementing the end-to-end data analytics.

2. Results and visualisation of the results. Providing required insight for the use cases.

3.3.3.6 Stage 3 Cycles

There were several cycles in this stage. However as deviation from our adaptation of the $(K|V)$ model, we just had one review session for this stage per iteration. Combination of different algorithms, their evaluation and then generating visualisations had to be repeated and tested many times. So reviewing in the intermediate cycles was very inefficient. This stage took more time because of many cycles and the wider spiral required to produce good quality results.

3.3.4 Stage 4. Documentation

Documentation was an ongoing process throughout the stages and the iterations. All of the stages had at least one deliverable in the form of some document e.g. stage 1 had the Big Data platform concept paper, stage 2 had the use cases document, in stage 3 we had data analysis and insights report. The purpose of the last stage was to consolidate all the information in different documents together in the shape of one single thesis report. The following were the four steps for this stage.

3.3.4.1 Problem statement vs. results review

This step was the check for the consistency of our results with the research problem that we had in the beginning. We earlier mentioned that documentation stage was not always part of the iterations. However this stage was used during all the iterations to keep track of the main focal points, the complimentary and supplementary parts of our research.

3.3.4.2 Document Integration

This step was concerned with the main task of this stage i.e. to consolidate the information together in the form of one consistent account. During this stage we tried to link the inter-stage documentation together along with theoretical background and the explanation of the process and the functional components of the project.

3.3.4.3 Process Review and discussion

The purpose of this step was to provide a retrospective view of the whole process. Also to highlight the main findings and discuss what could have been done better or more to improve the process and produce further results. This step also indicated some future directions for the relevant research areas. The considerations of this stage will be discussed further in chapter 6.

3.3.4.4 Document Finalization

This step controlled the final thesis report publishing aspects like formatting, sequencing of topics, proof reading and version control etc.

3.3.4.5 Stage 4 deliverables

The final thesis report document was the mandatory deliverable as the main output of the process and our research project. There were some supplementary deliverables like source codes, code books and procedures etc. that we intended to open source as part of our research.

3.3.4.6 Stage 4 Cycles

to be written after the document reviews.

3.4 Iterations

Iterative processes and work models do not require full specification right from beginning. Instead the implementation can start with a all part of specification. Then in a step wise approach the next scope is defined with consideration of lessons learnt and new directions found from previous iterations. This inherent characteristic of iterative processes had a vital role in our research. We started with a smaller scope i.e. two simple uses case. In earlier iteration we were able to focus on Big Data technologies and energy efficiency concepts more than the complex advanced analytics topics in later iterations. The findings and practical implementation in early phase enabled us to expand the scope later. We added more use cases with more focus on the data analysis and application of the Big Data for energy efficiency. Iterations also helped us in improving the quality of the research.

In our approach, we went through 3 main iteration cycles. As mentioned in the section 3.3 each iteration did not involve the complete four stages and their respective steps. The first three stages were the main contributors in the iteration with step 3.3.4.1 of stage 4 as the main source for reviewing our proceedings against our targets. Table 3.1 lists the main activities in each iteration against the respective stages and steps.

Stages	Steps	Iteration 1	Iteration 2	Iteration 3
Conceptualization	Understanding	Main Topics: Energy Efficiency, Eco-Efficiency, Demand Response, daily consumption, monthly consumption, smart grids, smart metering, NIALM, Big Data 3Vs, Parallel Batch Processing, MapReduce etc	Main Topics: Classification, clustering, K-means, Big Data Veracity and Valuation, Big Data Streaming, Lambda Architecture, Massively Parallel Processing etc.	Main Topics: Forecasting, Regression, Auto-regression, Moving Averages, ARIMA etc.
	Research quantitative methods	Sampling, Aggregation, Averages, Summation, standard deviation, distributions etc	Clustering,Centroid-based clustering; K-means, C-means, Distribution-based clustering; Cumulative distribution function, Density-based clustering; DBSCAN.	Time Series Analysis, Covariance, correlation, Regression, Auto-regression, Moving Averages, ARIMA, Random Forest etc.
	Conceptual Model	Model for parallel batch processing	Massively Parallel processing added for faster processing	Additional Machine learning modules (Forecasting)
	Evaluation of Tools	Apache; Hadoop, HDFS, Flume, Sqoop, oozie, Hive, Pig in Cloudera distribution. R, mahout, Tableau, D3.JS	Cloudera Impala, Spark, Hbase, MongoDB.	Weka, Cloudera Oryx, R (Forecast package)
Implementation	Use case definition	List of use cases: (1) Understanding the seasonal energy usage patterns and its sensitivity with outside temperature. (2) Understanding characteristics of building using daily energy consumption pattern.	List of use cases: (3) Calculate the base load of the building to identify non user consumption of buildings (4) Classify building on basis of energy efficiency and analyse seasonal shifts in this classification.	(5) Predict daily energy consumption of various house hold devices on basis of previous consumption pattern.
	Data analytics platform prototype	Parallel batch processing with capability to collect data from data from data servers and public social media streaming API s. Machine learning modules integration . Visualisation using Tableau Public.	Integration of on-line query engine with Cloudera Impala. This enabled near to real life Big Data processing.	Use of additional data mining and machine learning tools like Weka.
	Data collection	Hourly electricity and electricity for heating consumption data from VTT's smart metering devices on pilot sites for Green Campus Initiative.	Device level electricity consumption data from VTT's NIALM devices installed in two selected residential apartments.	One month twitter data collection for Green Hackathon using collection of energy related keywords.
	Prototype testing with sample data	Testing with samples from hourly consumption data. Testing with complete hourly consumption data.	Testing with NIALM device data.	Testing the prediction model using NIALM device data an additional data mining and machine learning tools. Performance comparison between non parallel executing, parallel batch processing and Masively parallel processing tools.
Data Analysis	Tight integration of the platform components	End-to-End workflow implementation i.e. from data collection, storage, preprocessing and analysis to visualisation of results. Limited to batch processing only	Integration of Impala.	
	Evaluation & Selection of Algorithms	Selected quantative methods: Basic aggregations e.g. averages , summations and groupings.	Selected quantative method: K-means clustering	Selected quantative methods: ARIMA, linear regression and Random Forest forecasting techniques.
	Applying Analytics	Applying basic aggregations according to use case 1 and 2.	1) Basic Aggregation for use case 3 2) K-means clustering for use case 4	ARIMA and Random Forest algorithms for use case 5.
	Result Visualisation	Using Tableau Public	Using Tableau Public	Using R plots and Weka
Documentation	Problem statement vs. results review	Results for use case 1 and 2 reviewed.	Results for use case 3 and 4 reviewed.	Results for use case 5 reviewed.
	Document Integration	Step not used	Step not used	Integration of platform concept paper, data analysis report and use documentation.
	Process Review and discussion	Step not used	Step not used	Theoratical background explainations and linkages to research. Future directions for related work.
	Document Finalization	Step not used	Step not used	Document formating.

Table 3.1: Details of the iterations

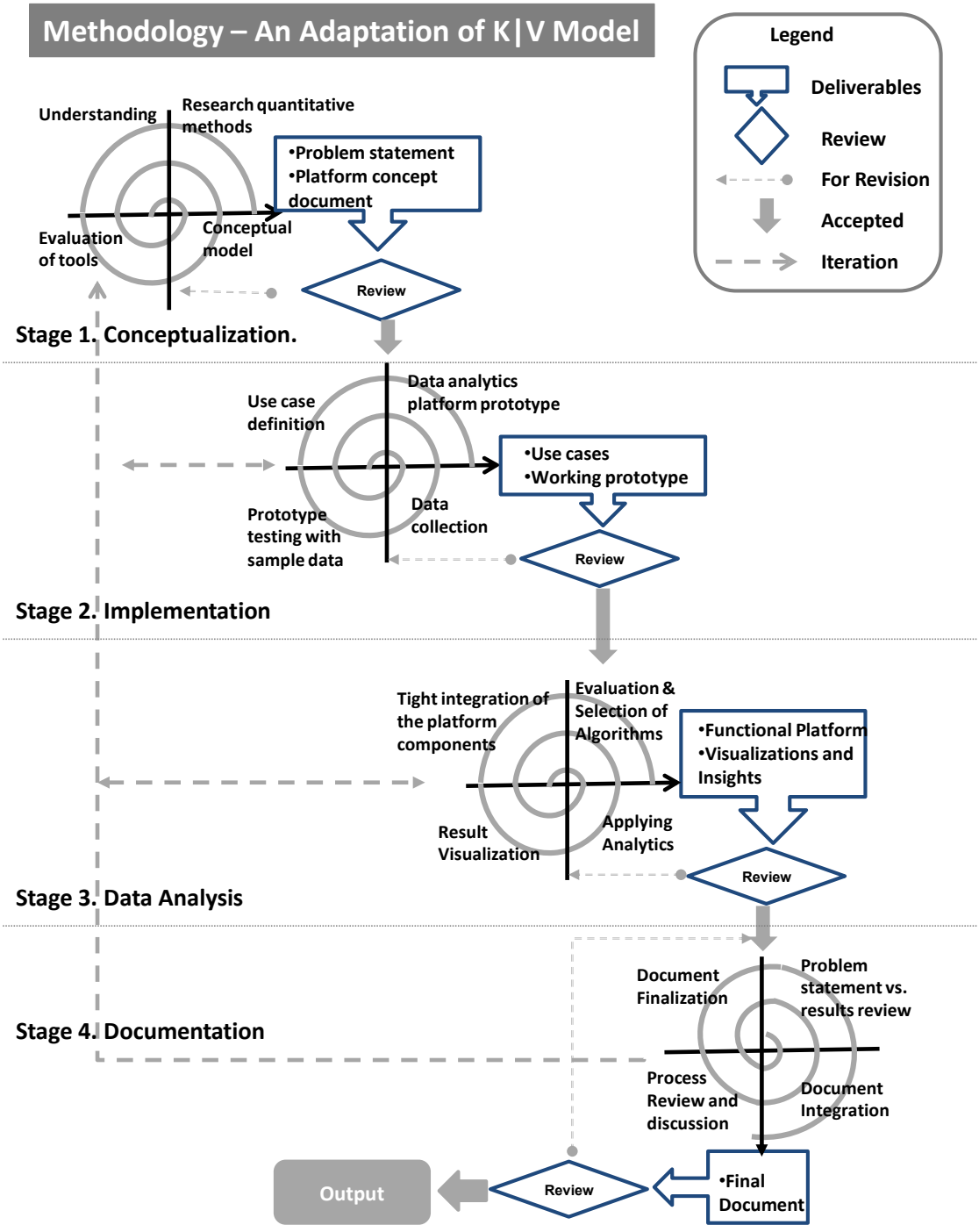


Figure 3.1: Methodology, an adaptation of $K|V$ model

Chapter 4

Concept and Implementation of Big Data Analytics Platform

We have referred to the Big Data platform concept and implementation on many occasions in all the previous chapters. In chapter 2 we have discussed the basic concepts of Big Data with various technological advancements and solutions available for handling Big Data. In chapter 3 we mentioned the functional components and their implementation during different stages, cycles and iterations. In this chapter we start with explaining the concept and a sample application framework for a Big Data platform based on available open source components. After that, we present the part of this concept that we have implemented for handling the energy and social media data for our energy efficiency use cases listed in section 3.3.2.1 of the previous chapter.

Before we move to our conceptual model of the Big Data platform it is important that we mention basic challenges that drive the design of a Big Data solution and briefly explain a typical Big Data analytics process.

4.1 Big Data challenges

As discussed in section 2.4 of chapter 2 there are five main challenges that influence the solution design criteria for Big Data analytic systems. These challenges are generally termed as the 5V's of Big Data.

1. **Volume** refers to the size of the data. Volume is the most commonly associated feature with Big Data. The Big Data analytics platform in our scope of work is based on Hadoop File System (HDFS) which is a highly scalable system. It has been tested with upto 4000 scaled out serving nodes capable of handling upto 10 Petabytes (PB) of data.

2. **Velocity** refers to the speed of data processing. Velocity is crucial for the business use cases that need to process huge volumes of data in real time to produce insights for decision making. Our model is designed for batch processing. However, we have integrated additional components that can process the data with near to real time capability.

3. **Variety** refers to the structure of the data. Traditionally the relational database systems can store data with fixed schemas. The fixed schema means that the stored data must have a definitive structure. Such databases are designed on basis of these data structures. In the context of Big Data, sometimes it is hard to perceive the structure of data so the storage systems needs to be designed for data in any structural format i.e. structured, unstructured or semi-structured. In our conceptual model we have added various components that can handle all formats of the data. However, in our implementation we processed the data that had fixed schema.

4. **Veracity** refers to the complexity due to the inconsistencies of the data. In real life scenarios for Big Data, it is very rare to find data in absolute consistent formats. Most of the time, some values will be missing or the data will be in the wrong format. For a good analysis, we need to take care of such inconsistencies and errors in data. Our conceptual model is capable of handling inconsistencies and in our implementation we had catered for some inconsistencies that we discuss in the next chapter.

5. **Valuation** compares the benefits of processing Big Data against the efforts required. It is an emerging feature for Big Data analytics design. Just like the other IT systems, organizations tend to decide about Big Data investments by looking at the business cases. In our concept, we present a model based on open source components. So there should be no cost of acquiring software. There is no specialized hardware requirements for implementing our model and any commercially available hardware with moderate specifications can be used to deploy this software. Hardware maintenance is required for running the service based on our model. There are Cloud alternatives that can be used within our model. However, we are not discussing such alternatives in the scope of this document or our research.

4.2 Data analysis workflow

The data analysis process involves collection of data from multiple heterogeneous sources including social media data, consumer data, sensors data, and already stored data from data servers or databases etc. The collected data in its original

form can be ingested directly into Hadoop File System (HDFS). If required, some filters can also be applied while collecting the data for efficient use of storage space. Collected data can be structured, semi structured or unstructured and some pre-processing can be done to format it i.e. form a schema or structure that can be stored and accessed for data mining in database(s). A data mining engine with flexibility to plug and use various quantitative and qualitative research tools can then be used to analyse the data as per use case requirements. The data mining engine requires a feedback loop to the pre-processing unit to adapt to the requirements of the use cases. Another feedback channel for processed data storage can be provided for direct data manipulations. Results of the data mining can be stored in the database. A RESTful API or data driver can be used to extract data from the database for visualisation front-ends. Figure 4.1 shows the high level process flow.

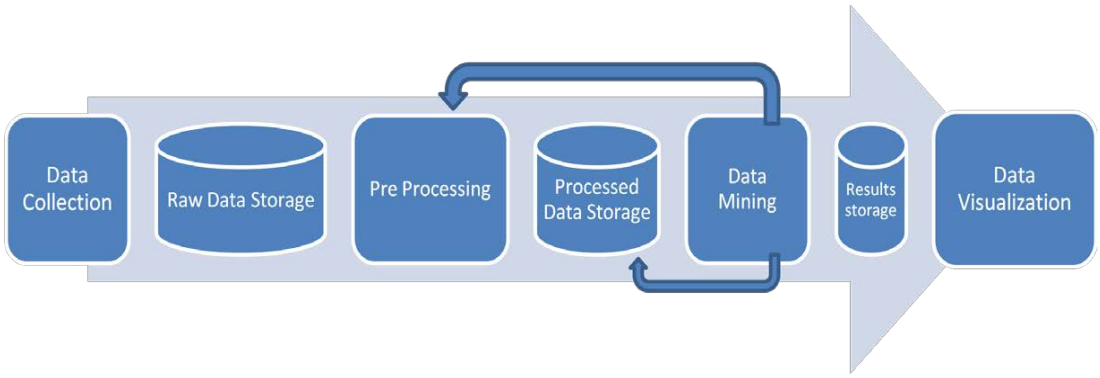


Figure 4.1: High level data processing flow.

4.3 Platform concept

This section presents an end-to-end Big Data analytics platform aligned with the data processing flow described in previous section. The proposed platform is based on software components that are available open source free of cost. However use of each software components is subjected to its respective license under a specific open source licensing scheme. There are closed source and paid Cloud services components that can be used as efficient alternatives for parts of this model. However this document does not contain information about these alternatives. Figure 4.2 illustrate the proposed concept.

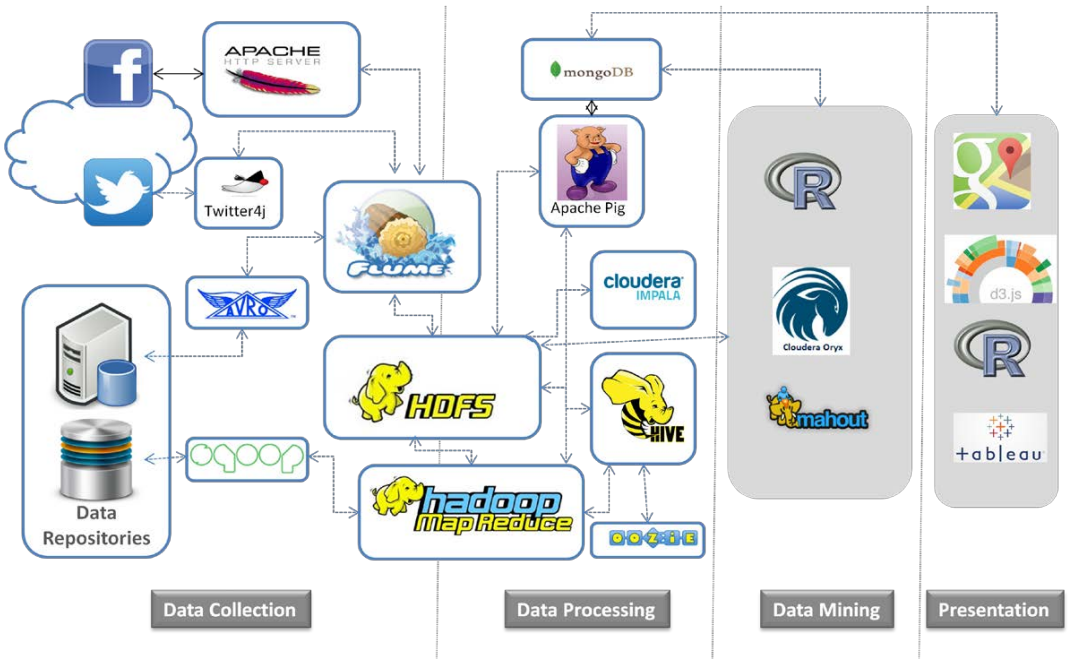


Figure 4.2: Conceptual model of Big Data analytics .

4.3.1 Data core

Before we go into the details of each and every process step and respective components, it is beneficial to discuss the data core of the platform that is based on Apache Hadoop and the Hadoop File System (HDFS).These two components are also shared between data collection and data processing steps. Apache Hadoop is a framework that allows the distributed processing of the large data sets across clusters of computers. HDFS is a distributed file system that provides high throughput access to application data[8] thus providing a highly efficient and scalable solution for handling Big Data. We have described Hadoop and MapReduce in section 2.4.1.

4.3.2 Data collection

The proposed platform is capable of collecting and aggregating data from the multiple data streams i.e. social media data, consumer data, or server log files etc. The data can be live streaming data or data residing in server file systems or databases. Data can be collected as it is so there are no dependencies on format or structure of the data. Some filtering can also be applied while collecting the data

e.g. collecting only the geo tagged tweets or collecting logs with error notifications only. Following two components are recommended for data collection

4.3.2.1 Apache Flume

Apache Flume is a distributed service for efficiently collecting, aggregating and moving large amounts of data [7]. Multiple flume agents can be configured to collect data from heterogeneous sources, channel the data to configurable destinations and store on desired locations. In the proposed model Apache flume is using Twitter4j library to stream data from Twitter, Apache HTTP REST API for collecting Facebook data and Apache Avro[6] data serialization system to collect log data from file systems of remote servers. Flume can then ingest data directly into the HDFS. Flume can also read from databases and it is particularly useful while reading from document stores (NoSQL databases). However for reading from relational databases the Apache Foundation has another useful tool called Sqoop.

4.3.2.2 Apache Sqoop

Apache Sqoop [11] is designed for efficiently transferring bulk data between relational databases and Hadoop. So in most of the consumer data cases Sqoop can be used to collect data and feed it into the HDFS through running multiple parallel Hadoop MapReduce jobs.

4.3.3 Data pre-processing

Once the data is available in HDFS then it can be normalized to structured formats. Furthermore, certain filtering can be applied e.g. the tweet text can be separated from other information for qualitative analysis and then natural language processing techniques can be applied to get it ready for further text mining. Pre-processing of the data also helps in filtering out the unwanted information to make the data lighter for the mining process. In the proposed model, Apache Pig and Hive are used to pre-process the data. Apache Pig and Hive both use Hadoop Mapreduce as parallel batch processing.

4.3.3.1 Apache Hive

Apache Hive [9] is dataware house software that offers a way of providing schema to the stored data with SQL based query language to extract data. Apache Hive is further supported by another Apache Hadoop ecosystem tool called Oozie. Oozie is acting as a workflow scheduler for Hadoop i.e. while loading the data into Hive it can create partition for tables to arrange the data for optimized querying.

4.3.3.2 Apache Pig

Apache Pig [10] provides a high level scripting language to analyze the data stored in HDFS using MapReduce. Apache Pig provides a complete Extract , Transform and Load (ETL) model for data processing.

4.3.3.3 Cloudera Impala

In section 2.4.2.2 we have already introduced Cloudera Impala as a massively parallel processing database engine. We also explained the concept of massively parallel processing databases. In this section we shall emphasize on how Cloudera Impala fits into the Hadoop ecosystem.

Cloudera Impala uses the HDFS as its main data source. It can read data directly from the HDFS directories. The HDFS directory path can be configured before each run. Data can be read from a single or multiple files stored within the configured directory. A schema needs to be created inside Cloudera Impala. Based on the schema it can project the data in the HDFS directory files as a table. The data can then be queried using a subset of SQL (structured query language). Cloudera Impala provides much faster processing than Hive and Pig. However Cloudera Impala currently supports limited data structures. For handling some complex data formats e.g. Avro serialized JSON format data, it can be configured to work together with Hive [4].

4.3.3.4 Databases

For storing the pre-processed data various choices of available open source databases can be applied in this platform. The document stores like MongoDB seems a natural choice because of the flexibilities to handle any structure data and easy maintainability but the relational databases can still be integrated and used within this platform. The availability of tools like Cloudera Impala can fit very well with SQL based databases like MySQL to build on-line query engines. Apache Hive in this model is also acting as a projection on top of MySQL.

4.3.4 Data Mining

The real value of any analytics platform lies in its ability to make sense of the data. Data collection and data pre-processing modules of the platform can bring and normalize data from multiple streams to be further analysed by the data mining modules in the platform. There are various open source data mining and statistical analysis tools available on-line with the power of most advanced machine learning, text mining and statistical modelling algorithms built in them. The proposed platform can provide a 'plug n play' environment for most of these tools to be

applied on use case requirements. Some example of these tools as presented in figure 2 i.e. Apache Mahout, R Project, and Cloudera Oryx.

R does not provide scalability or parallelism without special configurations. It processes data in memory so it requires large RAM (Random Access Memory) for large data sets. The real power of R is in the availability of a large number of statistical and advanced analytics algorithms. With proper data pre-processing using Big Data tools like Apache Hive, Pig and Cloudera Impala the amount of data processing can be reduced for R without loosing key insights.

Apache Mahout and Cloudera Oryx can fit well with Hadoop ecosystem tools to provide scalability and parallelism for applying data mining and machine learning on Big Data. Cloudera Oryx is designed particularly for velocity. However it is still in the development phase and has support for less number of data mining and machine learning algorithms. It is expected that it will take over Mahout completely if it can match Mahout's algorithm library.

4.3.5 Presentation

For visualisation of the results, this model presents some tools to build the interactive dashboards. These dashboards should be able to zoom in and out of data. They should provide a flexible way of managing visualisations based on use cases. Also they can be integrated into user interfaces of the web based applications or the mobile applications. The suggested components are Tabelau Public, d3.js, google maps and R project.

Tableau Public provides the easiest solution to visualise data through the static and interactive graphs. Tabelau Public is a free service. The paid version of Tableau can give additional tools like connecting directly to data bases and Hadoop ecosystem tools like Apache Hive. Tableau software are based on the VizQL (Visual Query Language) paradigm [23]. Once connected to the data source, VizQL provides a very flexible way of interacting with graphs. The idea is to focus on the insights that are required rather than spending more efforts on programming the queries to generate those insights. Tableau automatically generates the queries and visualises the results.

D3.JS provides a comprehensive library in Java Script for making the customized information graphics. It is very flexible and powerful. However it needs programming skills in Java scripts and a reasonable effort is required to prepare or change the graph formats.

Google maps is a powerful tool for geo-spatial info graphics. In many Big Data use cases geo spatial mapping provides a smart way of presenting the information.

R - project has additional packages and libraries like "ggplot" to visualise the results after applying statistical analysis and data analytics. Like D3.JS, R infographics also need programming to visualise the results.

4.4 Implementation

In this section we discuss our implementation of the Big Data analytics platform for analysing the energy consumption data of our energy efficiency use cases and collecting social media data for supporting the CIVIS project. The implemented model is the subset of the conceptual model presented in the previous section with some additional component. Figure 4.3 shows the implemented model.

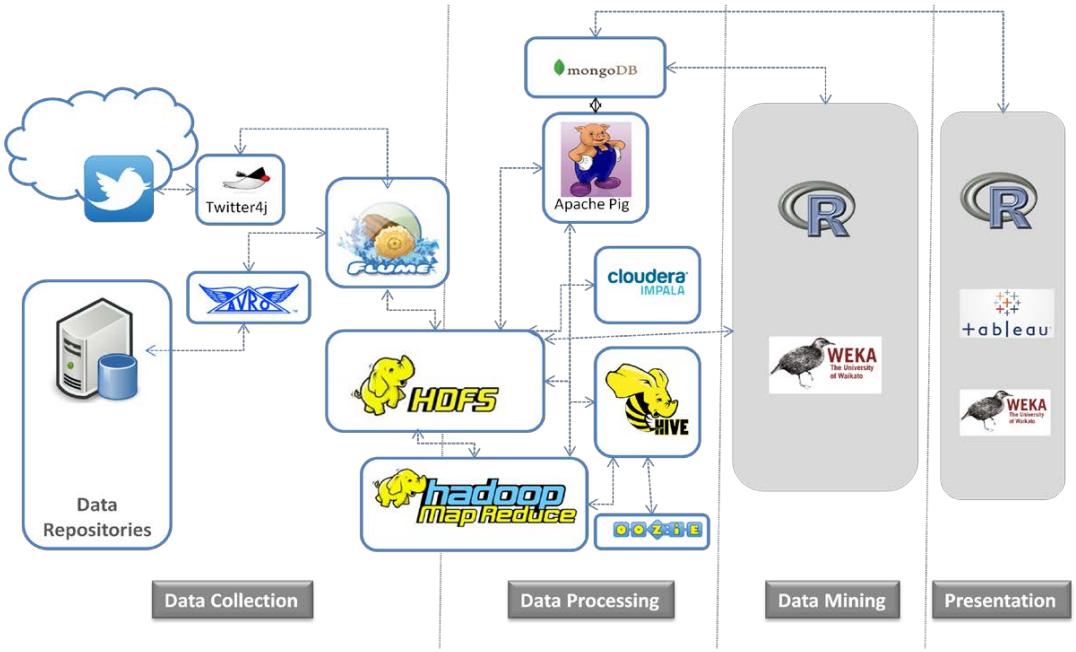


Figure 4.3: Implemented Big Data analytcis platform .

4.4.1 Implementation Environment

Cloudera distribution including Apache Hadoop (CDH) version 4.7 was used for implementation of our Big Data analytics platform. The following is the list of the components used with their respective version numbers.

Apache Flume	flume-ng-1.4.0+97
Apache Hadoop (MapReduce+HDFS)	Hadoop-2.0.0+1603
Apache Hive	hive-0.10.0+258
Apache Oozie	oozie-3.3.2+102
Apache Pig	pig-0.11.0+43

Cloudera Impala	Impala 1.3.1

Cloudera CDH 4.7 was used in the form of a pre-configured quick start virtual machine (VM) capable of running on top of any known operating system e.g. Microsoft Windows 7/8 , Linux RHEL/Centos, and Ubuntu etc. Cloudera CDH 4.7 quick start VM itself was running on Centos 6.2 operating system. The following Hardware resources were allocated to run the VM for our analysis.

- 3 x 1.8GHz intel i7 4500 CPU
- 4GB RAM
- 64 GB VMDK storage

During the testing phase, we have also tested multi node Cloudera CDH 4.7 configuration on the Cloud. However there was cost associated to running the set up on the Cloud and the requirement of our use cases were also fulfilled by the single node quick virtual machine. So we preferred not to run our analyses using the multi node Cloud deployment.

In addition to pre-configured components in Cloudera CDH 4.7 we also added some additional components as part of our platform. The following is the list of those components with their respective versions.

Apache Avro	v 1.7.6
R	v 3.0.3
Tableau	Public , v 8.0
Weka	v 3.7
Twitter4J	v 3.0.3

For ease of use, we had been using R, Weka and Tableau outside the quick start VM environment. Typically we were using Windows 7 with similar dedicated hardware. Tableau Public is a web service running in the public Cloud.

4.4.2 Implemented data processing work flows

As mentioned before, we utilized the implemented platform for two purposes.

- Analysing energy data for energy efficiency use cases.
- Collection of social media data to support the CIVIS project activities.

In this section we explain the data processing work flows for both scenarios. These work flows are aligned with the process we explained in section 4.2.

4.4.2.1 Data processing for energy efficiency use cases

For analysing the energy consumption data, we designed and implemented the platform to automatically collect data from VTT's data servers and ingest it into the Hadoop Files System. For data collection we configured Apache Flume to collect and aggregate the data. Apache Avro was configured within Apache Flume to serialize the data. Unfortunately, due to some policy issues at VTT, we were not allowed to integrate our platform to their data servers. Instead data was provided to us through an FTP server. In later stages we were also given access to a web service from which we could download the data in an off-line mode. Here off-line mode means that the data was not collected automatically and human intervention was required to collect the data.

The collected data was then ingested into the HDFS manually. Once the data is available in the HDFS then data pre-processing tools can access it. Selection of the pre-processing tool was based on use case requirements, format and volume of data. As discussed before, VTT provided us with two types of data i.e. hourly electricity consumption data from smart metering devices and a second set of NIALM device level data. For the first data set of hourly electricity consumption data, we used Apache Pig to pre-process the data.

The data was processed to prepare inputs for data mining and advance analytics modules as per use case requirement e.g. for classification on the basis of energy efficiency, the input matrix for K-means algorithm was produced. For producing this input matrix 1.2 Million records (*rows*) were reduced to 343 rows. Making it very simple for R to apply K-means algorithm. Then from R we saved the results of classifications to a comma separated file. This file was then read by Tableau Public. We then generated the visualisations for our analysis. Tableau can connect to data files in live mode so the updates in files can directly be imported and projected on created visualisations. However this was not required for our case.

For the second data set of NIALM device data, volume of the data was too small. So we analysed it in R without pre-processing. We used both R and Weka to apply and evaluate different analytics techniques. Weka was used only in the evaluation step as described in section 3.3.3.2 as it provides a quick mechanism for algorithm testing. It can not be used in tight integration within a platform so our conceptual model does not include Weka as an option. For visualisations we used R plots and Weka graphs.

4.4.2.2 Social media data collection for supporting CIVIS project

To support the CIVIS project activities we configured our platform to collect and store the Twitter data using the Twitter streaming API. The implementation procedure for achieving this case was based on a the Cloudera Tutorial "How-to

Analyze Twitter Data with Apache Hadoop" [3].

We configured Apache Flume with Twitter4j Java library to capture the Twitter streaming data. A Twitter application was registered with our Twitter account to get the access to streaming API. The keyword filtering was applied using Twitter4j to target the concerned tweets only. Tweets were then stored to the HDFS using the Apache Flume to the HDFS sink mechanism. A Hive table was created with the a subset of Twitter provided schema. This helped us in shedding the unwanted header data to reduce the storage size. An Apache Ozie workflow was implemented to archive the Hive data in a manageable format i.e. creation of Twitter data partitions on basis of hourly data. This is a very helpful feature in reducing the query processing time. Within our scope we did not analyse the Twitter data.

Chapter 5

Data Analysis and the Results

In the section 3.3.2.1, we have listed the energy efficiency use cases. In chapter 2, we have explained the relevance of these use case to energy efficiency and also introduced and explained the basic concepts of suitable statistical and advanced analytics techniques to extract insights from data for these use cases. In chapter 4, we have explained the implementation of a model Big Data platform for providing an environment to perform these analysis. In this chapter, we explain how we have performed our data analysis using the knowledge and capabilities developed through our work, which is explained in the previous chapters. We also present our results and their prospective applications. Before we go into the details of the use cases, analysis and the results, it is important that we explain the data sets.

5.1 Data sets

In our analysis, we have used two distinct data sets provided by VTT. Both the data sets were collected by specialized devices installed on the test sites as part of the Green Campus initiative. The description of data sets are as follows.

5.1.1 Data set 1: Hourly energy consumption data

This data set contains hourly readings of the energy consumption taken by specialized meters installed by VTT (Technical Research Centre of Finland) on 40 different test sites in the cities of Espoo and Helsinki. Each site has at least one meter installed. Each installed meter on a particular site can record consumption of a specific energy type on an hourly basis. In some cases, a test site may have more than one meter even for same energy type. There are four energy types considered for test sites i.e. Electricity, Heating, Water and Reactive power. We consider only electricity and heating in the scope of this thesis. The heating usage

is explained in terms of the electricity consumed to produce heating. In the rest of this chapter, energy types will be termed as "features" and each record representing hourly consumption of a feature will be called an "observation". There are approximately 1.2 million observations taken from the 144 installed meters on 40 different sites, during the 11 months period between 1st January 2013 till 30th November 2013. Each row in the data represents an observation with the following information in exact sequence separated by commas:

"Device ID","Destination Address","Building Name", "Meter", "type", "date", "hour", "Consumption"

A "Device ID" is the unique id for each installed device. To avoid confusion, the sites are being termed as buildings in the rest of the document, For sake of anonymity both "Destination Address" and the "Building Name" fields are masked as BuildingXX (where XX ranges from 01 to 40). "Meter" shows the nth number of the meter inside the same building. The field "type" labels the feature, while "date" refers to the calendar day in the format of YYYYMMDD, "hour" is the hour number of the day (0 to 23) and the "Consumption" is the consumption of the feature in respective units. Only electricity and electricity consumed for heating is considered in this paper so units are 10xWh (Watt hour). To fix the scale of consumption to Kilo Watt Hour, each consumption value should be divided by 100.

The data set is not consistent and has an unequal number of the observations for some buildings per energy type per day. Figure 5.1 illustrates the summary of this data set and shows the inconsistencies. From the figure 5.1, we can see that the number of the records for each building are not equal. In an ideal data set they should be the same. The inconsistencies occur because of the multiple number of devices collecting data, however exploratory analysis reveals that for some of the buildings, the consumption values were totally missing. In some other cases, all the days in the 11 months period were not available. Similarly all hours in a day were not fully captured for some buildings.

Apart from the inconsistencies shown by number of the records, some building names and addresses were also missing. To handle inconsistent data, we had to adapt our analysis using aggregation techniques like averaging. We shall also explain other data cleaning and pre-processing techniques in section 5.3.1. The implementation level details of data are available in Appendix B

To support this data set for the analysis, we have also collected the real estate data that includes floor area of each building included in this data set. The Real Estate data was collected using the following Espoo city information website. `http://https://arska.espoo.fi/`

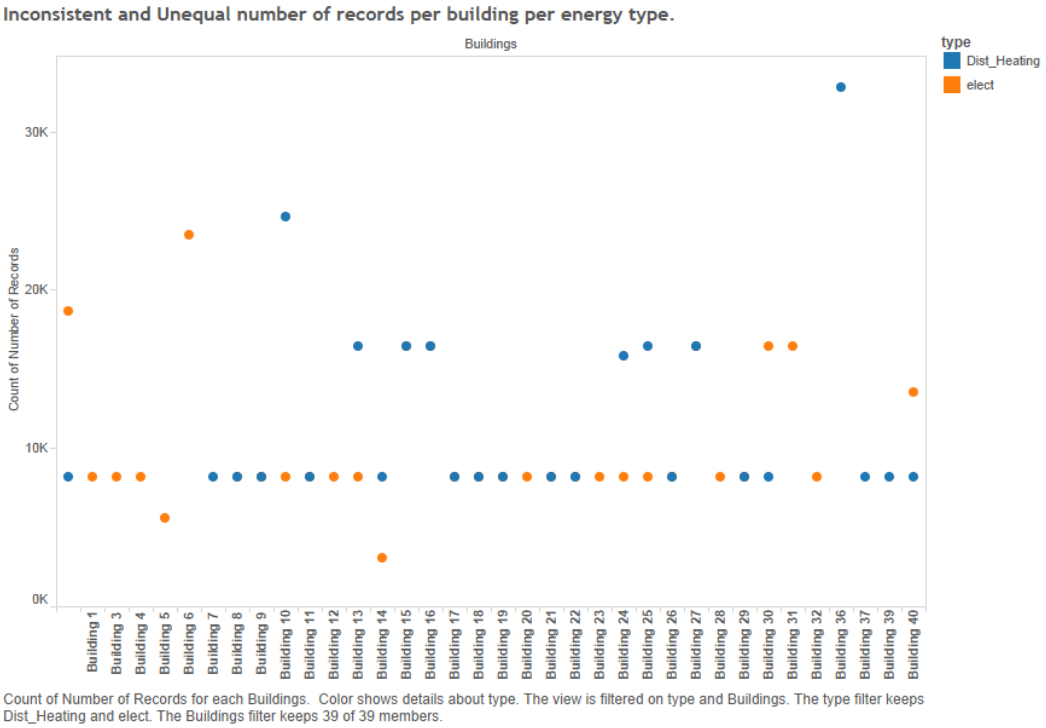


Figure 5.1: Data Inconsistencies in hourly energy consumption data.

5.1.2 Data set 2: Device level data

This data set contains device level consumption of the electricity collected from a residential apartment included as a test site for the VTT's Green Campus initiative. Device level means that the electricity consumption was collected for various home appliances used in that apartment. These appliances were differentiated on the basis of their respective electrical signal thumbprint using NIALM metering devices described in section 2.3. The data was collected from 1st May 2013 to 30th April 2014 in form of a text file from the VTT's data webservice. The different fields in the data were separated by ";". The data set has the following fields.

"Devic";"Timestamp";"Consumption(Wh)"

The "Device" field is a label for the respective home appliance e.g. refrigerator, freezer, TV, and stove etc. "Timestamp" contains the date, hour, minutes and seconds of the recorded data in the YYYY-MM-DD HH:MM:SS format. The "Consumption(Wh)" field contains the value of electricity consumed and recorded

at that instance of time. The devices that are used on demand for example a TV, stove, coffee maker etc. records the consumption values right after the end of the respective usage session. For continuously used devices like refrigerators and freezers etc. NIALM devices have their inbuilt mechanism to keep recording the data. For our research we take the data record as it appears in the data set.

5.2 Use Case Categories

As listed previously in the section 3.3.2.1, the following are the main use cases for our research.

Use Case 1: Understanding the seasonal energy usage patterns and their sensitivity with outside temperature.

Use Case 2: Understanding the characteristics of the buildings using daily energy consumption pattern.

Use Case 3: Calculating the base load of the buildings to identify off-peak hours usage when users are not in the building.

Use Case 4: Classifying the buildings on the basis of energy efficiency and analyse seasonal shifts in this classification.

Use Case 5: Forecasting the daily energy consumption of various household appliances on the basis of the previous consumption patterns.

Data set 1 was used for the use cases 1,2,3 and 4, while data set 2 was used only for the use case 5. In a similar way we grouped the use cases into two categories for analysis. Use cases 1 to 4 were referred to as "Energy Consumption patterns and classification of the buildings on the basis of energy efficiency". While use case 4 was labelled as "Prediction model for forecasting the energy consumption of the household devices "

5.3 Energy consumption patterns and energy efficiency classification

In the section 2.5, we have discussed the concept of energy efficiency in context of the ecological factors. In the section 2.7, we have discussed the impact of outside temperature on energy usage. Similarly we have established the relationship between daily load, base load and energy efficiency in the section 2.6. Now using all these concepts together, we try to analyse data set 1 and detect the daily and seasonal trends for energy usage. We also try to find out the base loads for the buildings to see which building is consuming more energy during the off peak hours when users are not present in the building. Finally we attempt to classify the building on the basis of energy efficiency as per section 2.8. The following

subsections will describe the each step along with explanation of the analysis and results.

5.3.1 Data cleaning and pre-processing

For preparing a tidy data set for the analysis, the following steps were performed:

1. We intended to perform the analysis for electricity and heating features, so the relevant observations were extracted from the data. In the rest of the document, we only consider these extracted observations.

2. To check the consistency and quality of the data, a quick statistical analysis was performed. The result of the analysis is illustrated in figure 5.1.

3. The consumption scale was set to KWh (Kilo Watt Hour).

4. For each feature, the distinct buildings were listed to see how many buildings have each type of feature available. The result shows that 32 out 40 building has observations available for electricity while 24 has observation for heating. The observations for those building were extracted which have records available for both the features. In some of the observations there were empty "Destination Address" and "Building Name" fields. Such observations were tagged as "Unknown".

5. There were some other pre-processing steps that are explained with their respective use cases.

5.3.2 Seasonal variation in energy consumption

Seasonal variation in the use of electricity is a very obvious phenomenon. However the important aspect for our analysis was to first see the sensitivity of energy usage with change in outside temperature as per use case 1 and then see the impact of the seasonality on the classification of building on the basis of energy efficiency i.e. to check if a building of a particular class shifts to another class with change in the external temperature. This provides the basis for use case 4. Since all of the test buildings are in cities of Helsinki and Espoo which are geologically located close to each other, so it can be safely assumed that both city has similar temperatures throughout the year. The upper graph in figure 5.2 shows the aggregated consumption for all the buildings. Two separate lines represent electricity and electricity used for heating respectively. While the lower graph shows the average temperature of Helsinki and Espoo during the same time of the year. The temperature data was collected using the Finnish Meteorological Institutes's (FMI) data API.

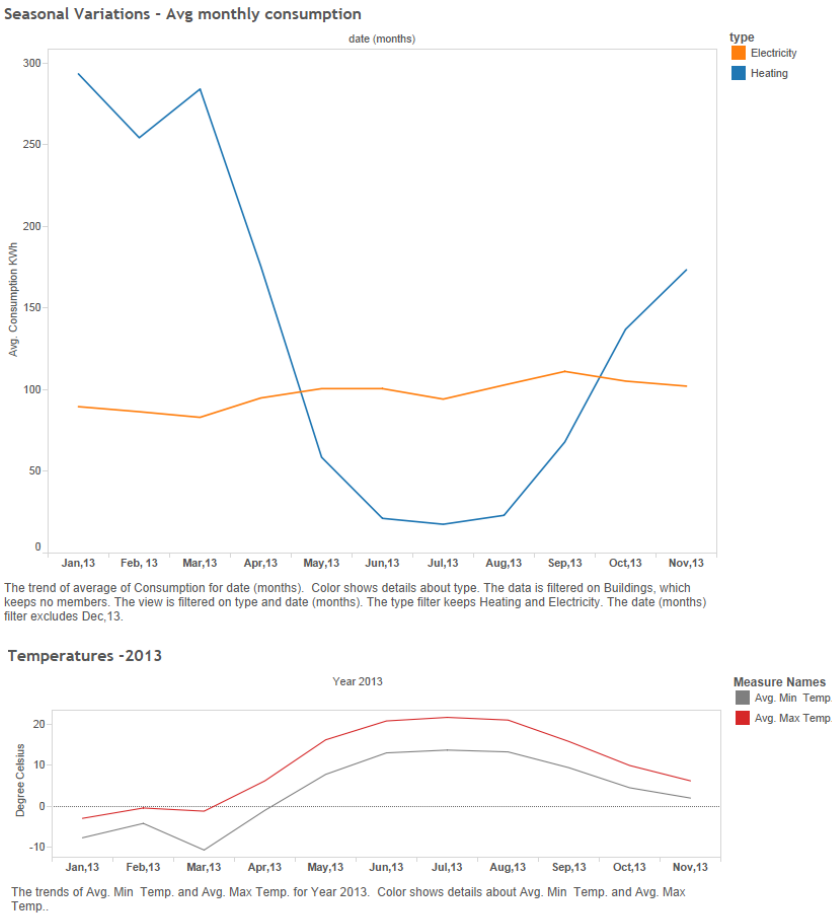


Figure 5.2: Seasonal patterns in usage of electricity and electricity used for heating.

Figure 5.2 shows that electricity used for heating is more sensitive to temperatures than electricity used for the other purposes. The general purpose electricity consumption shows a relatively stable trend through out the 11 months period. For the service providers, improving heating distribution and usage systems can contribute more to energy efficiency.

5.3.3 Daily trends

Detecting and projecting the daily trends from data set 1 can provide us two very important insights that corresponds to use cases 2 and 3. It can help to suggest the characteristics of a building without having any prior information of the building. For example if the use of energy is higher during the work hours of the working

days and lesser during night times and on the weekends, then we can suggest that the building is an office building. Secondly, the base load analysis can refer us to building where there could be possible electricity leakages. Rectifying such issues can improve the overall energy efficiency of the buildings.

In our analysis we detected the daily trends of the buildings by averaging the consumption of each building separately for each hour of the day. Normalizing the data for missing values was very important, so instead of averaging with the total number of days for each building, we took averages for the number of days for which data is available.

Figures 5.3a and 5.3b show the average daily electricity consumption of the two buildings from our data i.e. Building 6 and 13 respectively. While figures 5.3c and 5.3d show the electricity consumption for the heating of the same buildings. Both type of consumption are higher during office hours suggesting the purpose of building. While each building has different base loads. Base load hours are very visible in the daily trends graphs. An interactive dashboard to observe the trends for all the buildings is available on the following website:

`http://https://arska.espoo.fi/`

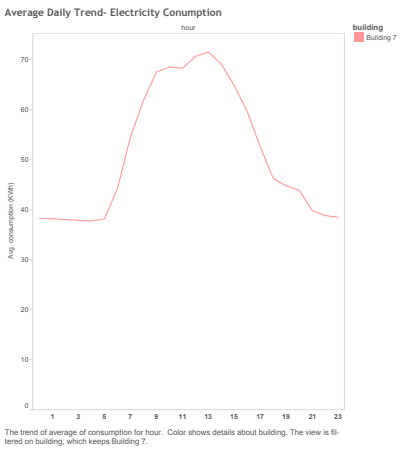
A quick exploratory analysis of the trends suggests that all buildings are office buildings. Secondly the daily base load hours are hour 0, 1, 2, and 23. The base loads can then be calculated by averaging the consumption for these hours. Appendix C contains the list of calculated base loads for each of the buildings in each month of the year within available data.

5.3.4 Classification of buildings on basis of energy efficiency

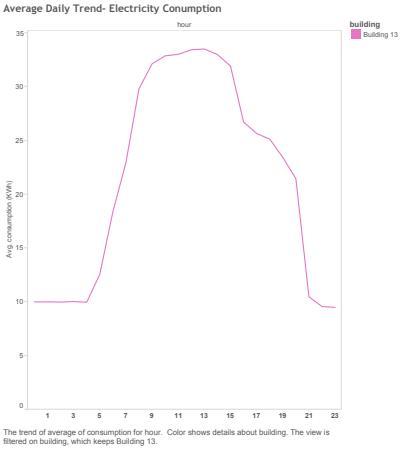
Classification on the basis of energy efficiency can be used as a tool to benchmark and segregate the inefficient energy consumption units from the efficiently performing buildings. Such classification can narrow down the scope of research for finding the possible energy leakages and faults. We discussed the main concepts and the K-means clustering technique, we used for our analysis in section 2.8. The existence of some similarity is a pre-condition for any cluster analysis technique. To test this on our data set we calculated the average hourly energy efficiency using equation 2.2 for each building. Figure 5.4 confirms the availability of similarly behaving buildings in terms of energy efficiency.

5.3.5 Data processing for cluster analysis

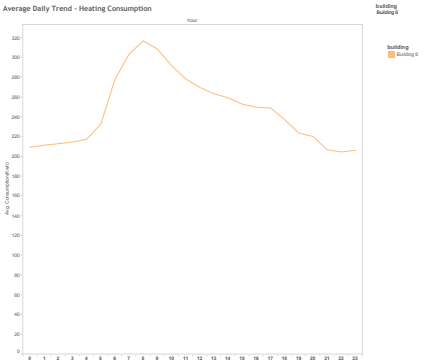
The K-means algorithm needs input data in the form of a matrix, where the values of each feature are defined in their respective columns, while each row represents



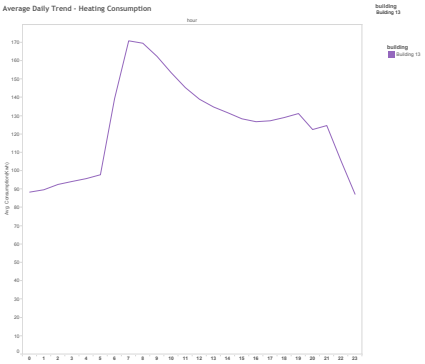
(a) Building 6: Daily electricity consumption trend



(b) Building 13: Daily electricity consumption trend



(c) Building 6: Daily electricity for heating consumption trend



(d) Building 6: Daily electricity for heating consumption trend

Figure 5.3: Daily energy consumption patterns

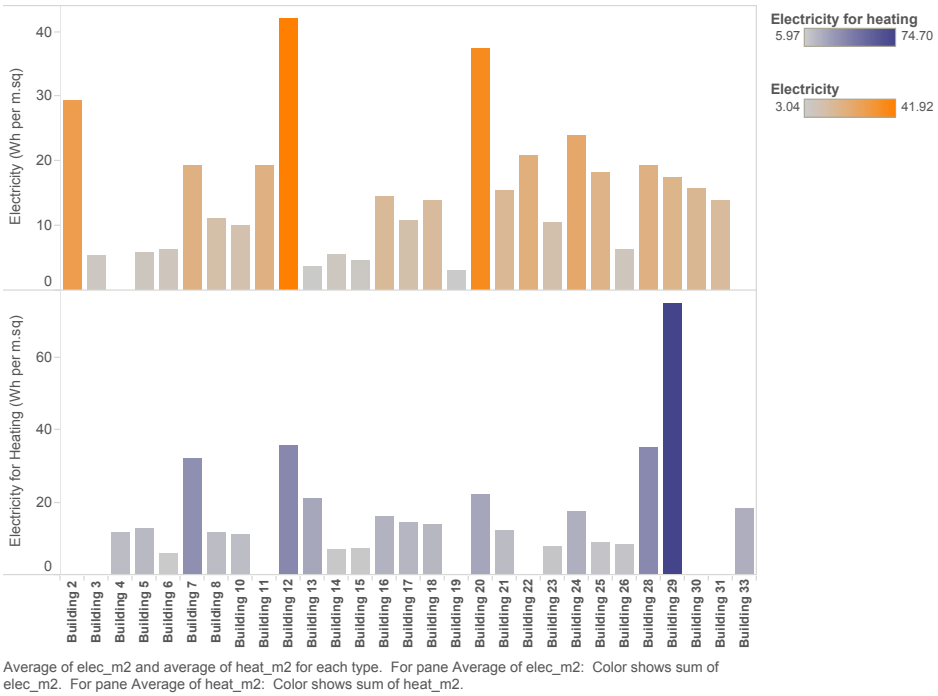


Figure 5.4: Energy efficiency of buildings with hourly average consumption.

a data point that will be grouped together in the form of clusters. In our case we have two energy features i.e. electricity and the electricity used for heating. While the average monthly energy efficiency value for each month and each building is represented by a row in the input matrix. The reason for considering the average monthly values for buildings is to analyse the energy efficiency of the building throughout the 11 months period. So in this way our input matrix consisted of:

$$\text{Number of rows, } i = \text{Number of buildings x } 11$$
$$\text{Number of coulmn, } j = 2$$

Following are the main data processing steps for our cluster analysis. These steps were performed on the data set that we have as output of the preliminary data pre-processing described in section 5.3.1.

1. We segregated the electricity and electricity used for heating records from each other.

2. We calculated the average daily consumption for each building and each en-

ergy type first. Then calculated the average monthly consumption in similar way. It is important to normalize the data by taking the averages on the basis of number of records available for the hours within a day and then the days within a month.

3. We then arranged calculated average monthly consumption values for each energy type in form of a matrix labelled with building names and month names as the separate columns. We called this matrix as the "Energy Consumption Matrix".

4. At this stage our Energy Consumption Matrix had few buildings for which consumption values were missing for either type of energy feature. We removed these building to avoid inconsistency in data for the cluster analysis.

5. We then introduced the real estate data i.e. ground floor area of the respective buildings into our energy consumption matrix. Using equation 2.2, we calculated the energy efficiency values for each energy feature. We termed the resulting matrix as the "Energy Efficiency Matrix".

6. Until this point, we had energy efficiency in units of Kilo Watt hour per square metre. We then converted the values into Watt hour per square metre. This was an optional step and it was performed just to avoid handling small decimal values.

7. To prepare the final input matrix we removed the labels and left two columns of energy efficiency values for the two target energy features.

8. To finalize the K-means input matrix, we used the R programming **scale()** function to set the unit variance for all the matrix elements. This function could have also normalized the scale of the measurements, however in our case it was already a similar unit for both energy types i.e. (Wh/m^2).

5.3.6 K-means clustering analysis and results

Our use case requirement was to classify the buildings into four categories of high efficiency, moderate efficiency, low efficiency and poor efficiency buildings. So we had the pre-defined K value of 4. We applied K-means clustering on our input matrix using R programming **kmeans()** function.

As a post processing step we combined the resulting cluster values to the energy efficiency matrix in front of their respective labels (Building Name + Month Name) and energy efficiency values (electricity and electricity used for heating). We termed this matrix as the "Clustered Matrix'. The Clustered Matrix was

then fed into Tableau public as a CSV file to create an interactive dashboard and visualisations. Figure 5.5 visualises the result of clustering.

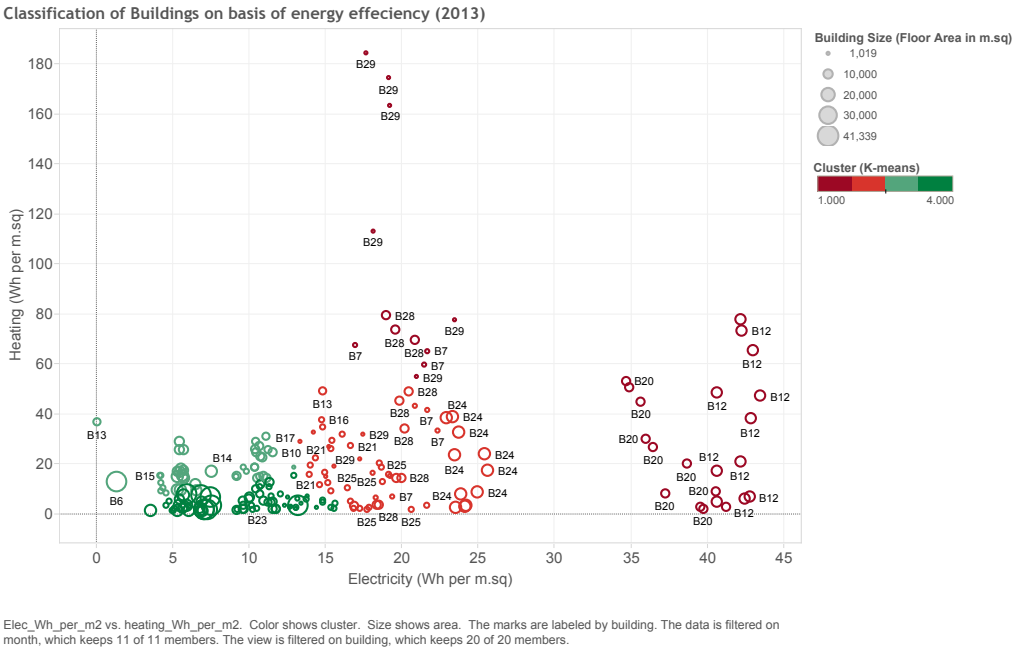


Figure 5.5: K-means clustering, Average monthly energy efficiency per each building.

Each bubble on the graph in Figure 5.5 represents a month's average energy efficiency value for a particular building. The colour of the bubble represent the respective cluster or class. While size of the bubble represents the size of the building. The cluster numbers range from 1 to 4, where 4 represents the highly efficient class and 1 is for the most energy inefficient class of the buildings. Figure 5.6 shows the one month subset of the clustered values. Each buble represent average energy efficiency in the month of January.

Insight: Figures 5.5 and 5.6 reflect a very important insight; that some of the bigger size buildings are in high or moderate efficiency clusters while some of the smaller buildings are in inefficient clusters. Such extreme cases can be good targets for case studies. The low efficiency buildings may have some energy leakages, faults or inefficient usage practices while the high efficiency buildings may suggest good practices for using energy.

As part of the use cases, we also studied the change behaviour of the buildings

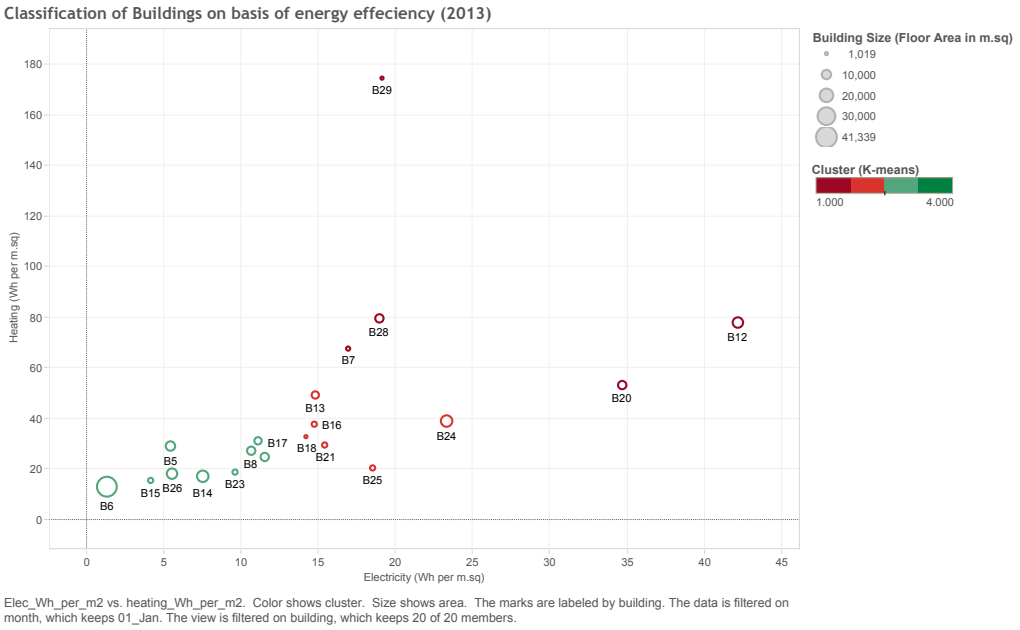


Figure 5.6: K-means clustering, A one month view.

with the external temperatures, while using the same cluster analysis. Figure 5.7 illustrates the behaviour of different buildings during the 11 month period. Figures 5.7a and 5.7b present the behaviour of Building 29 and Building 7 respectively. Both the buildings shift among three different clusters. While figures 5.7c and 5.7d represent the buildings 16 and 24 that show shift between two clusters and no clusters respectively.

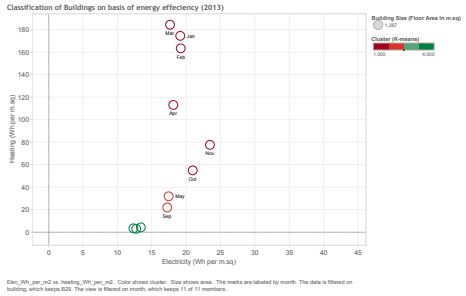
Insight Figure 5.7 shows the fact that buildings are not equally energy efficient or inefficient throughout the year.

An interactive dashboard is available for viewing the behaviour of all the buildings in the 11 months of collected data via following web address:

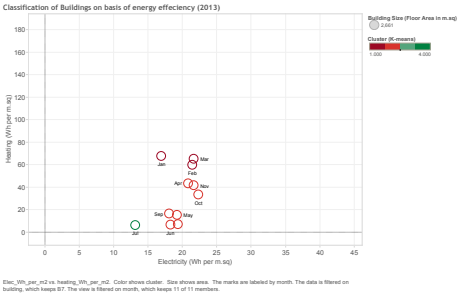
`http://catalyc.net/`

5.4 Forecasting energy consumption of household devices

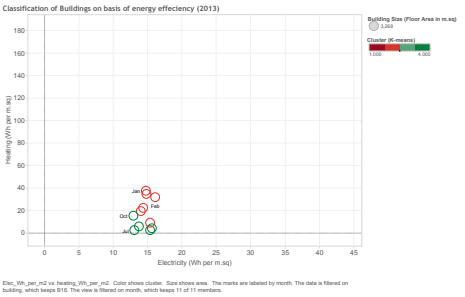
We discussed the data set collected by NIALM devices in section 5.1.2. Using this data set we tried to evaluate different prediction models using limited amount of



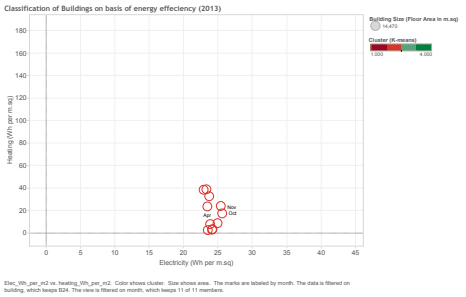
(a) Building 29: Triple shift



(b) Building 7: Triple shift



(c) Building 16: Double shift



(d) Building 24: No Shift

Figure 5.7: Energy effeciency cluster shift during 11 months

our data set. There were nine different appliances included in our data set i.e. Refrigerator, Freezer, Dishwasher, Laundry machine, Coffee maker, Stove, TV or PC, and Microwave oven. Some of the devices are generally in the continuous use e.g. freezer and refrigerator. While others are used on the need basis e.g. stove, coffee maker, laundry machine etc. Even in such devices, frequency of usage can be different e.g. stove, coffee maker etc. are usually used on daily basis while laundry machine can be used once in a week. The limited use of the data set means that we are not dealing with all this diversity. Instead we forecast only for those devices which are in continuous usage e.g. freezer and the refrigerator. The predicted values are on the daily consumption basis for a particular device.

5.4.1 Important considerations for forecasting

Following were some of the important considerations while evaluating the prediction model.

1. We collected NIALM data in two phases. The first data set contained consumption values from 1st May 2013 to 3rd February 2014. This data was used as training data for prediction models. While the seconds data was collected with consumption values from 4th February 2014 till 30th April 2014. This set was used as the test data to measure the accuracy of the prediction model.

2. Predictions models were evaluated using consumption data of the freezer usage.

3. A time window of previous 30 days was used to predict the values for the next 30 days with 80% confidence interval.

4. The accuracy of the prediction models were compared on the basis of Mean Absolute Error (MAE).

5.4.2 Data processing steps

Following are the main steps that were performed to evaluate the prediction models.

1. Data was reshaped to form continuous time series. The missing daily values were filled with zeros.

2. Data was aggregated to calculate the daily consumption for each device.

3. The freezer data was extracted from the processed data set to apply the prediction models.

4. Three forecasting models were applied to the training data i.e. Linear Regression, ARIMA, and Artificial Neural Networks (AAN). We used R for the ARIMA modeling and Weka for the Linear Regression and AAN. We shall discuss about the usage of these tools in chapter 6.

5. We calculated the accuracy using test data and mean absolute error formula.

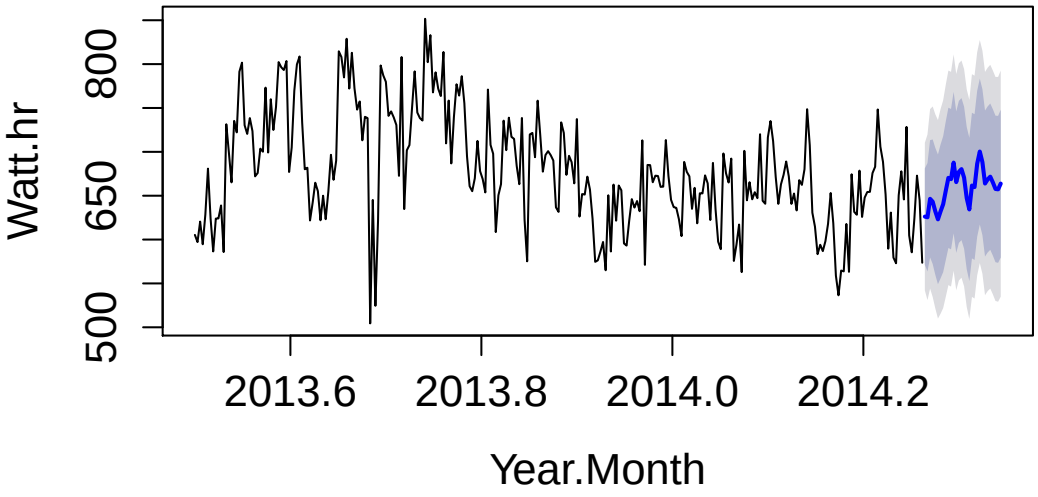
5.4.3 Forecasting results

Figure 5.8 shows the resulting forecasts from the three prediction models. In figure 5.8a, predicted values using ARIMA model with 80% confidence interval are represented by the bold blue line, while shaded geom represent the higher and lower 80% and 95% confidence intervals respectively. The figures 5.8b and 5.8c illustrate the predictions using linear regression and artificial neural networks (ANN) models respectively using the 80% confidence interval. Circular dots in the graphs show the predicted daily values, while the doted lines represent the higher and lower 80% confidence interval prediction ranges. Due to the use of different analysis tools, the ARIMA graph is different from the linear regression and ANN graphs. Both the X axis and Y axis in all the graphs represents days and electricity (Watt.hour) respectively. However label of the days in the X axis is in the figure 5.8a is showing respective months with year while the label in the figures 5.8b and 5.8c represents the number of the day starting from the first day in the data.

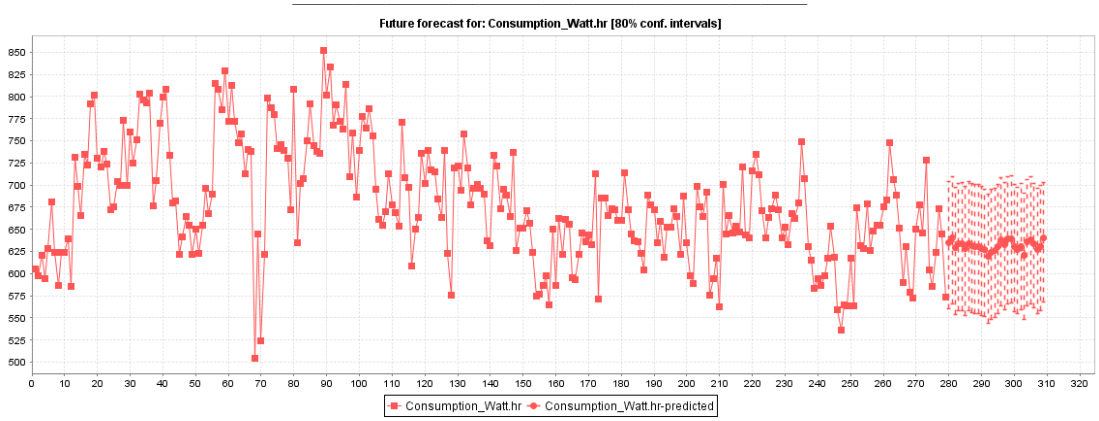
Table 5.1 lists the mean absolute error (MAE) values for each prediction model using 80% confidence interval. Forecasting with the ARIMA model shows the lowest MAE values. For this reason, we included ARIMA forecasting as part of our implemented platform. We use R to perform calculations described in the section 2.9.3 using packages like “Forecast” for fitting and forecasting with the ARIMA model and “zoo” for handling time series data structures. The p,d,q values for predicting monthly values based the previous 30 days were 30,0,30 respectively.

	ARIMA	ANN	Linear Regression
Mean Absolute Error (MAE)	32.6	47.5	42.4

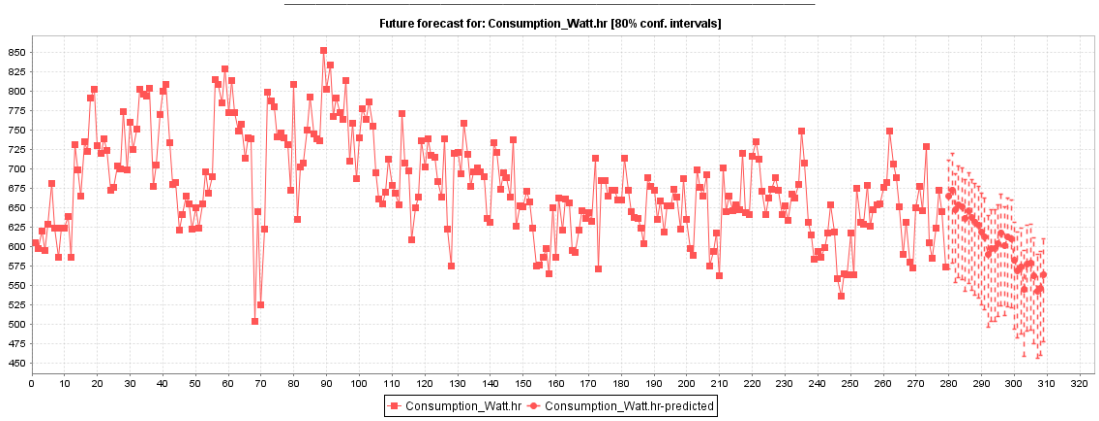
Table 5.1: Mean absolute error values for prediction models.



(a) ARIMA with 30 days AR and MA



(b) Linear Regression Model



(c) Artificial Neural Network Model

Figure 5.8: Forecasting monthly energy consumption for the home appliances based on previous monthly consumptions

Chapter 6

Discussion

In this chapter we attempt to summarize our work with a critical analysis of our approach and results while highlighting some possible future directions related directly or indirectly to our research. Our research has three main constituents i.e. (i) Big Data tools and techniques (ii) Big Data analytics (iii)Using Big Data analytics to facilitate in improving energy efficiency. We focus on these topics for our discussion in this chapter.

6.1 Big Data tools and techniques

In our approach, we tried to cover data volume, velocity, variety, veracity, and valuation as the five major aspects of Big Data. The concept and implemented platform are horizontally scalable to meet the large data set requirement. We performed most of our analysis on small size hardware described in section 4.4.1. For the proof of concept we also tested the implemented platform in a multi server environment by scaling out to four serving nodes of similar hardware specifications as explained in section 4.4.1. We used both public cloud and dedicated on-premises hard to test multi server implementation. But for the purpose of data processing during our analyses, we used single hardware server node in pseudo distributed mode i.e. using resources within a single server in distributed parallel mode. This single node configuration in pseudo distributed mode provided us enough computing power to handle the data we collected from VTT.

The largest data set collected for our research was an approximately 250MB file with around 1.2 Million records in it. This is a small volume compared to the volumes that are generally associated with Big Data. It is important to mention the fact that this data was collected from only 40 buildings for a period of 11 months as a pilot project. The real life commercial scale use of such projects can include thousands of buildings and data processing may require including of the

data recorded during several years. To test our system's readiness for large data sets we replicated the collected data several times in the form of larger file sizes in multiple numbers.

In terms of velocity, we analysed our data using distributed parallel batch processing. For the sake of fast and near to real-time data processing speed, we also tested Cloudera Impala. The performance of Cloudera Impala in terms of speed, was many times faster than the batch processing tools like Apache Pig and Apache Hive. However, Cloudera Impala has very limited support for handling complex and composite data structures. There are some new emerging tools like Apache Spark and Apache Shark that can enable much faster data processing and offers support for complex data structures. Adaptation of our data platform with these new emerging tools can further enhance its data processing capabilities.

Data variety and veracity was managed on the basis of our use case requirements and collected data sets. We built our data platform on top of Cloudera CDH that provides basic Apache Hadoop ecosystem data processing tools like Hive and Pig. On top of it we also integrated database systems like MongoDB that can ingest schema-less data to give flexibility for handling unstructured data. Statistical programming capabilities using R were also available to clean and process data. The whole platform was based on open source software and the model has the inbuilt flexibility for use in any kind of deployment models e.g. public or private cloud, own premises general purpose hardware, or combination of them.

In an ideal data platform, the processes and workflows should be fully automated. This means that after configurations, the platform should be able to collect, process, mine, and visualise data automatically. Our data platform covered the end-to-end process for data analytics. All the components were integrated to implement the required workflows. However some manual interventions were required to execute the workflows e.g. extracted insights were fed to visualisation tools in the form of off-line CSV files. This can be improved to provide full automation using some paid services like database connectors to visualisation. Another automation can be added for operations and maintenance of the platform using tools like Opscode Chef and PuppetLabs AutomateIT tools. It is worth mentioning here that Cloudera CDH provides a management control panel to manage resources. But this control panel is limited to tools within CDH.

6.2 Big Data Analytics

The real value of processing data lies in revelation of hidden valuable knowledge. Big Data analytics is about processing large volumes and diverse varieties of data on a need basis to discover the hidden patterns, underlying correlations and other useful information. Statistical analysis tools like R, SPSS and Matlab have a

diverse range of powerful advance analytics libraries and packages that can be used to extract the information from data. But these tools allow scalability only in terms of scaling up. Such scalability is always limited and requires specialized expensive hardware. Combining the data processing power of Apache Hadoop and MapReduce with these statistical tools can enable us to process huge amounts of data and mine the required information with tools like R.

We used this concept for data processing and analysis in our reserach. We leveraged on the power of Hadoop to pre-process our bulk data and transformed it into a smaller size while keeping the useful information intact. We then applied the advance analytics using R to generate the required insights. This is a very useful approach. But in order to build a continuous data processing and mining pipeline it has its limitations e.g. manual or off-line mode for transfer of processed data between the data processing and the data mining phases.

There is a new emerging paradigm of Big Data analytic tools that can execute variety of advance analytics techniques directly on top of the Hadoop File System. Apache Mahout is one example that can be integrated within the Apache Hadoop ecosystem and provides powerful advance analytics capabilities. Mahout uses parallel batch processing, so inherently it is limited for use in on-line streaming analytics. Cloudera Oryx and Mlbase are two new emerging tools in this ecosystem that can integrate with Apache Spark and Hadoop 2.0 to provide on-line analytics on streaming Big Data. Integration and use of these tools can improve the capabilities of platform discussed in the scope of our thesis research.

6.3 Using Big Data analytics for energy efficiency

We used R in combination with Apache Hadoop in our data analysis for energy efficiency use cases. We described data sets in the details in section 5.1. In the same section we mentioned about the respective use cases of each data set. For the hourly energy consumption data set we analysed the daily and seasonal usage patterns using simple aggregation methods like summation and averaging. These patterns provided the basic information about the impacts of ecological factors on consumption of some specific types of energies. We also observed the sensitivity of consumption patterns to the variations in these ecological factors. Identified seasonal trends in the data also laid the basis for our next important use case i.e. the classification of buildings on the basis of energy efficiency.

We tried to quantify the energy efficiency of the buildings in our data set using VTT's previous research on the topic. Then we attempted to group the similarly performing (in terms of energy efficiency) buildings together and classify them into four categories ranging from highly energy efficient to highly inefficient buildings. Within our classification we also tried to observe the seasonal trends on

the classification. We used the cluster analysis technique with K-means algorithm to classify the buildings. K-means algorithm requires a predefined number of clusters. In our case we had our basic requirement to categorises building into four categories. In other possible use cases, where there may not be any predefined categories, other advance analytics techniques e.g. the Hierarchical Agglomerative Clustering can be used. The use of our classification technique is to identify the possible opportunities to improve the energy efficiency and learn from good practices. As an obvious next step to our research, the respective buildings with lower energy efficiency should be explored to identify the possible energy leakages, faults and bad usage practices. Similarly the good practices can be learnt from the energy efficient buildings. Our energy efficiency classification is limited to the general purpose usage of the buildings. This classification model may raise false red flags for buildings with specialized usage like data centre buildings, laboratories and factories etc.

For the NIALM data set with device level energy consumption information, we tried to compare different prediction models that can help energy service providers to plan for demand response, production and distribution. Our comparison was limited to the prediction model for continuously used devices like refrigerators and freezers. We predicted the future month's usage on the basis of previous monthly usage. This comparison could actually be the first step for building a prediction model for energy service providers, while a recommendation engine for users that can learn from user behaviour. The energy tariff data can be included to provide the recommendation to user to improve energy efficiency, as well as to save money from reduction from the reduction on energy spendings. This model can be improved further by adding more historic consumption data that can further affirm the seasonal and usage specific trends.

Chapter 7

Conclusion

Global energy needs are continuously growing. The conventional methods for producing more energy to meet the demand pose a great threat to the environment. Among other solutions, energy efficiency has become a major tool for minimizing the need for producing more energy to cater for the growing demand. Inherently, the cause of improving energy efficiency relies on understanding the usage patterns, identifying the problematic areas, establishing good energy consumption practices and to rectify the faults for reduce energy leakages. The advancement in sensors, ubiquitous computing and communication technologies has provided the basis for effectively collecting the usage data to understand energy usage. The collected data needs to be processed to generate leads for improving energy efficiency. The quality of insights generated from data improves if we consider the current data in context to historic data. This means that data volume for processing will keep on increasing. There can be multiple sources of data so the data formats can also vary. On the use case basis, data processing requires flexibility for customization and variation in speed of data processing. All of these data features refers to application of Big Data technologies for energy efficiency.

Distributed parallel computing programming models like MapReduce provide the basic environment for handling Big Data. We leveraged on the power of MapReduce using Apache Hadoop ecosystem tools to present an end-to-end Big Data analytics tool. Hadoop supports scalability to meet large volumes of data sets while the there are other tools that can integrate with Hadoop to process complexity in data. We used the model platform to process real life energy data and generated insights that can be used to improve energy efficiency. The proposed model provides a 'plug and play' environment for many other analytic tools to integrate on a need basis. It is based entirely on open source software components and can be deployed using general purpose hardware or any cloud based model.

We observed the strong sensitivity in the consumption of specific energy types to the ecological factors like external temperature. The visualisation of the average

daily consumption of each building suggested the respective use of the buildings. It also provided the information about base loads of the building. Optimizing the base load of the building improves their energy efficiency. We also calculated the energy efficiency of the buildings in our sample data set and classified them on the basis of calculated energy efficiency. This classification provided us a reference point to identify the buildings to focus on and locate the possible faults, energy leakages and bad consumption practices. This classification can be an important tool for the organization working to improving the energy efficiency, as it can help to isolate the problematic buildings. Our results also show a dynamic behaviour of buildings' energy efficiency performance during different months in the year. This particular insight can be used as another reference to isolate the causes for energy inefficiency in the target buildings.

We also compared different prediction models to forecast the future energy usage of different home appliances on the basis of their previous usage. The prediction model itself can be useful for energy service providers to understand and plan for user specific demand. However, this can also be used as the first step towards building a decision support system for users to effectively use home appliances. The decision support system could predict the usage pattern and then recommend the best options on the basis of configurable parameters e.g. best tariffs or best time to use green energy etc.

In a nutshell, our research provides a proof of concept of how emerging Big Data technologies can be applied in the energy industry to improve energy efficiency and data driven decision making. We presented and demonstrated the concept of our Big Data analytics platform and applied it to solve the real-life use cases from the energy industry. The output of our research is a working Big Data analytics platform and the results generated from advance analytics techniques applied specifically to solve energy efficiency problems.

Bibliography

[1] The cloudera impala - open source, interactive sql for hadoop. http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html. Accessed: 2014-06-09.

[2] Directive 2012/27/eu of the european parliament and of the council of 25 october 2012 on energy efficiency. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2012:315:0001:0056:EN:PDF. Accessed: 2014-06-10.

[3] How to analyze twitter data with apache hadoop. http://blog.cloudera.com/blog/2012/09/analyzing-twitter-data-with-hadoop/. Accessed: 2014-07-05.

[4] Impala concepts and architecture. http://www.cloudera.com/content/cloudera-content/cloudera-docs/Impala/latest/Installing-and-Using-Impala/ciiu_concepts.html. Accessed: 2014-07-05.

[5] Lambda Architecture what is the lambda architecture. Accessed: 2014-06-09.

[6] Welcome to apache avro. http://flume.apache.org/. Accessed: 2014-07-04.

[7] Welcome to apache flume. http://flume.apache.org/. Accessed: 2014-07-04.

[8] Welcome to apache hadoop. http://hadoop.apache.org/. Accessed: 2014-07-04.

[9] Welcome to apache hive. https://hive.apache.org/. Accessed: 2014-07-04.

[10] Welcome to apache pig. http://pig.apache.org/. Accessed: 2014-07-04.

[11] Welcome to apache sqoop. http://sqoop.apache.org/. Accessed: 2014-07-04.

[12] Arundel, A., and Kemp, R. Measuring eco-innovation. *United Nations University Working Paper Series*, 2009/017 (2009), 1–40.

[13] Balijepalli, V. M., Pradhan, V., Khaparde, S., and Shereef, R. Review of demand response under smart grid paradigm. In *Innovative Smart Grid Technologies-India (ISGT India), 2011 IEEE PES* (2011), IEEE, pp. 236–243.

[14] Boehm, B. A spiral model of software development and enhancement. *SIGSOFT Softw. Eng. Notes 11*, 4 (Aug. 1986), 14–24.

[15] Box, G. E., and Jenkins, G. M. *Time series analysis: forecasting and control, revised ed.* Holden-Day, 1976.

[16] Commission, F. E. R., et al. Assessment of demand response and advanced metering.

[17] Dean, J., and Ghemawat, S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM 51*, 1 (2008), 107–113.

[18] European Union 7th framework programme. Proposal part b - cities as drivers of social change civis project - ict-2013.6.4 optimising energy systems in smart citiese, 2013.

[19] Farhangi, H. The path of the smart grid. *Power and Energy Magazine, IEEE 8*, 1 (2010), 18–28.

[20] Forgy, E. W. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics 21* (1965), 768–769.

[21] Forsström, J., Lahti, P., Pursiheimo, E., Rämä, M., Shemeikka, J., Sipilä, K., Tuominen, P., and Wahlgren, I. Measuring energy efficiency.

[22] Grijalva, S., and Tariq, M. U. Prosumer-based smart grid architecture enables a flat, sustainable electricity industry. In *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES* (2011), IEEE, pp. 1–6.

[23] Hanrahan, P. Vizql: a language for query, analysis and visualization. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (2006), ACM, pp. 721–721.

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

[25] Hartigan, J. A., and Wong, M. A. Algorithm as 136: A k-means clustering algorithm. *Applied statistics* (1979), 100–108.

[26] Hyndman, R. J., and Athanasopoulos, G. *Forecasting: principles and practice*. 2014.

[27] Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., and Booch, G. *The unified software development process*, vol. 1. Addison-Wesley Reading, 1999.

[28] Janne Peltonen. Presentation on vtt otaniemi greencampus summary, 2013.

[29] Khan, I., Capozzoli, A., Corgnati, S. P., and Cerquitelli, T. Fault detection analysis of building energy consumption using data mining techniques. *Energy Procedia 42* (2013), 557–566.

[30] Kumiega, A., and Van Vliet, B. A software development methodology for research and prototyping in financial markets. *arXiv preprint arXiv:0803.0162* (2008).

[31] Laney, D. 3d data management: Controlling data volume, velocity and variety. *META Group Research Note 6* (2001).

[32] Laplante, P. A., and Neill, C. J. The demise of the waterfall model is imminent. *Queue 1*, 10 (Feb. 2004), 10–15.

[33] Li, X., Bowers, C. P., and Schnier, T. Classification of energy consumption in buildings with outlier detection. *Industrial Electronics, IEEE Transactions on 57*, 11 (2010), 3639–3644.

[34] Lloyd, S. Least squares quantization in pcm. *Information Theory, IEEE Transactions on 28*, 2 (1982), 129–137.

[35] MacQueen, J., et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (1967), vol. 1, California, USA, p. 14.

[36] Martin, R. C. *Agile software development: principles, patterns, and practices*. Prentice Hall PTR, 2003.

[37] Marz, N., and Warren, J. *Big Data: Principles and best practices of scalable realtime data systems*. O'Reilly Media, 2013.

[38] Mujumdar, P. Stochastic hydrology-video course.

[39] Neville, C., Gatti, P. J., Sekhon, B. S., Patil, J., Korachagaon, A., Shiralashetti, S., Marapur, S., Archana, R., Basavaraj, B., Bharath, S., et al. Referencing: Principles, practice and problems. *RGUHS J Pharm Sci 2* (2012), 1–8.

[40] Ng, A. Cs229 lecture notes. *CS229 Lecture notes 1*, 1 (2000), 1–3.

[41] Russom, P., et al. Big data analytics. *TDWI Best Practices Report, Fourth Quarter* (2011).

[42] Stonebraker, M. The case for shared nothing. *IEEE Database Eng. Bull. 9*, 1 (1986), 4–9.

[43] Tso, G. K., and Yau, K. K. A study of domestic energy usage patterns in hong kong. *Energy 28*, 15 (2003), 1671–1682.

[44] Tufte, E. R., and Graves-Morris, P. *The visual display of quantitative information*, vol. 2. Graphics press Cheshire, CT, 1983.

[45] Yohanis, Y. G., Mondol, J. D., Wright, A., and Norton, B. Real-life energy use in the uk: How occupancy and dwelling characteristics affect domestic electricity use. *Energy and Buildings 40*, 6 (2008), 1053–1059.

Appendix A

List of evaluated tools

Tools	Evaluation	Learning	Selected
Web Crawling and data collection			
Twitter API (streaming) V1.1	Done	Done	Yes
Tweepy	Done	Done	Yes
Apache Flume			
Apache Kafka	Done	Required	Not
Apache Sqoop	Done	Done	Yes
Batch Processing - MapReduce Toolkits			
Apche Hadoop	Done	Done	Yes
Cloudera CDH4	Done	Done	Optional
Apache Pig	Done	Done	Yes
Apache Hive	Done	Done	Yes
Databases			
MongoDB	Done	Done	Yes
mySQL	Done	Done	Yes
Online Query Engine			
Cloudera Impala	Done	Done	Yes
Apache Spark	Done	Required	For Future Use
Cloudera Storm	Done	Required	Not
Presto	Limited	Required	Not
Data Mining			
Apache Mahout	Done	In Progress	Yes
Data Visualization			
Tableau	Done	Done	Yes
Google Maps	Done	Done	Yes
D3.JS	Done	Done	Optional
Advanced Analytics			
R	Done	Done	Yes
Apache Mahout	Done	Done	Yes
Cloudera Oryx	Done	Done	Yes
MLBase	Not Started	Required	Not

Figure A.1: List of evaluated platform components.

Appendix B

Data Descriptions

B.1 Hourly consumption data

Contains the hourly utility consumption data recorded by the specialized meters, installed by VTT in test buildings as Green Campus initiative test sites. These devices measures consumption of following utility types.

1. SÄKHÖ (Electricity);(consumption /100) will be KWh values
2. KAUKOLÄMPÖ(District Heating-);same unit as above
3. LOISTEHO (Reactive-Power) -KVar
4. VESI(water) in m^3
5. Data header is not available in data, please follow the description.

There are 8 comma separated fields in the each record / row. Description is as the following,

1. Device ID: are the unique codes to identify the metering device installed in buildings. One building may have more than one device.
2. Building Names in Finnish language. You may require the appropriate encoding e.g utf-8 to read the names
3. Building Address in Finnish language. You may require the appropriate encoding e.g utf-8 to read the addresses.
4. Meter Type is related to utility type value 1 to 4
5. Utility type. Four types explained above. Text in Finnish language. You may require the appropriate encoding e.g utf-8 to read the names

6. Date YYYYMMDD of collected record
7. Hour of the day, Value from 0 to 23. 0 hour mean utility consumed from 00:00 to 00:59 hr of the day.
8. Consumption value of utility. Units as explained above.

B.2 NIALM Device Data

This data is collected from the NIALM device installed in one of the test site- a residential student-family apartment.

1. First row describes the time zone setting.
2. Second row is the header.
3. Data fields are separated by ;
4. Each record contains the name of the device used, time stamp of usage and the amount of electricity consumed in Watt hour unit.
5. Please note that there are certain devices that are being logged into records after their respective usage however there are devices like refrigerator and freezer that are being used continuously and NIALM device uses internal mechanism to get consumption value of these devices after certain period of time.

Appendix C

Detailed Results

C.1 K-means clustering

	Month										
Building	01_Jan	02_Feb	03_Mar	04_Apr	05_May	06_Jun	07_Jul	08_Aug	09_Sep	10_Oct	11_Nov
B5	3	3	3	3	4	4	4	4	4	3	3
B6	3			4	4	4	4	4	4	4	4
B7	1	1	1	2	2	2	4	2	2	2	2
B8	3	3	3	3	4	4	4	4	4	4	3
B10	3	3	3	3	4	4	4	4	4	4	3
B12	1	1	1	1	1	1	1	1	1	1	1
B13	2	3									
B14	3	3	3	3	4	4	4	4	4	4	3
B15	3	3	3	3	4	4	4	4	4	3	3
B16	2	2	2	2	2	4	4	4	4	4	2
B17	3	3	3	3	4	4	4	4	4	3	3
B18	2	2	2	3	4	4	4	4	4	2	2
B20	1	1	1	1	1	1	1	1	1	1	1
B21	2	2	2	2	4	2	4	2	4	2	2
B23	3	3	3	4	4	4	4	4	4	4	4
B24	2	2	2	2	2	2	2	2	2	2	2
B25	2	2	2	2	2	2	2	2	2	2	2
B26	3	3	3	3	4	4	4	4	4	3	3
B28	1	1	1	2	2	2	2	2	2	2	2
B29	1	1	1	1	2	4	4	4	2	1	1

Figure C.1: Details of K-means clustering results. Four classes of buildings are represented as high efficiency (4), moderate efficiency (3), low efficiency (2), and poor efficiency (1).

C.2 Base loads

Building	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov
Building 12	406.2	410.5	414.5	395.9	399	426.3	422.2	431.2	417.2	419.9	431.2
Building 24	289.3	292.3	289.3	291.8	306.1	313.2	310.5	316.8	321.7	322.1	319.3
Building 2	251.7	254.1	255.3	257.39	255.5	279.7	271.8	275.3	268.1	256.1	245.3
Building 20	227.9	235.8	234.4	239.5	247.7	259.1	255.9	265.6	268.3	258.1	244.9
Building 31	170.3	175.7	163.1	159.6	182.8	195.4	190.4	176	177.9	164.6	164.7
Building 14	78.2	57.1	54.4	50.4	52.2	49.4	45.3	46.9	50.9	52.4	56.1
Building 11	77.8	83.7	83.7	83.4	74	66.5	59.7	55.2	51.7	72.5	75
Building 28	62.3	66.8	68	71.9	73.2	73.90	67.7	67.5	68.5	65.7	69.2
Building 8	60.9	65	63.3	68	72.3	75.7	75.59	73.5	67.5	68	65.5
Building 25	60.3	61.4	61.7	60.6	71.2	69.59	60.6	62.1	51.4	50	45.5
Building 10	59.3	54	53	45.2	50.8	53.9	56.5	55.9	57.6	59	57.5
Building 6	56.2	0	0	183.9	231.7	239.9	232.8	228.6	400.2	223.6	205.3
Building 17	48	48.1	47	44.2	45	46.3	44.8	48.2	48.6	48.6	47.4
Building 26	44.3	44.7	44.7	56.1	59.1	63.1	59.8	59.9	60.9	56.3	49.5
Building 5	43.6	45.8	45.6	44.6	46.8	50.9	50.5	52.7	85.4	47.1	47.6
Building 21	41.9	45.8	42.4	38.5	39.2	47.2	42.7	46.5	40.5	40.2	40.4
Building 16	38.5	43.3	42.2	40.2	40.7	41.2	36.1	41.9	33.9	31.8	35.7
Building 22	31.1	27.8	32	36.1	37	34.9	32.7	33.6	35.5	36.4	39.1
Building 7	30.9	42.1	44.1	41.7	38	37.2	26.1	39.5	35.4	44.4	43.3
Building 1	25.2	26.2	23.4	21.6	22.7	24.7	22.2	24.6	23.1	22.8	29.4
Building 23	19.8	20.3	21.2	23.1	20.6	18.8	18.7	20.5	23.9	23.7	22.7
Building 29	16.6	14.7	13.6	12	10	8.6	10.7	9.6	10.4	10.9	12.7
Building 30	14.7	14.6	13.8	14.3	14.2	14	13.3	13.9	13.5	11.7	11.9
Building 18	12.5	13.2	12.8	13	13.1	11.9	12.7	13.5	13.1	13.1	13.4
Building 15	9.2	10.2	10.4	10	12.6	15.8	16.6	16.2	13.2	10.4	9.8
Building 19	4.5	5.3	5.5	4.8	4.4	3.6	4.0	4.5	4.5	4.8	4.9

Table C.1: Base loads of the buildings in KWh.