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! — July 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

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:**	67
Professorship:	Data Communication Software	**Code:**	T-110
Supervisors:	Professor Matti Vartiainen Professor Jukka Nurminen		
Instructor:	Sanja Scepanovic M.Sc. (Tech.)		

A dissertation or thesis is a document submitted in support of candidature for a degree or professional qualification presenting the author's research and findings. In some countries/universities, the word thesis or a cognate is used as part of a bachelor's or master's course, while dissertation is normally applied to a doctorate, whilst, in others, the reverse is true.

!Fixme **Abstract text goes here (and this is an example how to use fixme).** Fixme! Fixme is a command that helps you identify parts of your thesis that still require some work. When compiled in the custom `mydraft` mode, text parts tagged with fixmes are shown in bold and with fixme tags around them. When compiled in normal mode, the fixme-tagged text is shown normally (without special formatting). The draft mode also causes the "Draft" text to appear on the front page, alongside with the document compilation date. The custom `mydraft` mode is selected by the `mydraft` option given for the package `aalto-thesis`, near the top of the `thesis-example.tex` file.

The thesis example file (`thesis-example.tex`), all the chapter content files (`1introduction.tex` and so on), and the Aalto style file (`aalto-thesis.sty`) are commented with explanations on how the Aalto thesis works. The files also contain some examples on how to customize various details of the thesis layout, and of course the example text works as an example in itself. Please read the comments and the example text; that should get you well on your way!

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
CIVIS	Cities as drivers of social change
VTT	Valtion Teknillinen Tutkimuskeskus (State Technical Research Center of Finland).
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

Contents

Chapter 1

Introduction

In modern era we have seen phenomenal increase in human dependency on information and communication technology. ICT enabled products and services has transformed the way of life on the planet. We need and depend on ICT to fulfil our needs from basic physiological level to the human desire of being 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 is a term that is being widely used to refer to these digital traces of human activity. Ubiquity of computing resources, fast and highly mobile connectivity and advent of social media usage has caused a great surge in volumes of data. Realizing the true potentials of data, businesses are not only utilizing it as source of decision making but as a new revenue stream. Large scale opportunities are emerging that 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 like Google and Yahoo has opened technologies and tools like MapReduce and Hadoop to facilitate these advancement and open source software communities like Apache Software foundation has further developed the tools to provide a complete ecosystem for handling big data and generate insights. The new specialized big data companies like Cloudera and Hortonworks has emerged as catalyst for this data revolution. In this research we try to formulate a model for 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 the insights based on business logic and then present the information using interactive visualizations. This thesis includes the development as well as implementa-

tion of the mentioned big data platform to perform analysis on real life use cases and generate useful insights. The model that we present in this thesis 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 they are not discussed in this scope of this thesis.

This thesis is inspired by European Union CIVIS- Cities as drivers of social change project under 7th framework. CIVIS project focuses on 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 functional unit to achieve global goal of low carbon emissions from energy ecosystem. The use of pervasive ubiquitous computing is driving the smart energy solutions. Combined with internet of things (IoT) for home/building automation, smart commuting, and remote monitoring is becoming the basis for energy conservation via gaining energy efficiency. All the smart energy devices as part of this ecosystems generates high volumes of data, that needs to be instantaneously transferred, stored, analysed and visualized for knowledge discovery and improvements of services for the goal of achieving high energy efficiency. The platform that was developed as part of this thesis has the capability to automate the whole process.

Energy usage pattern detection, classification of buildings on basis of energy efficiency and a prediction model for energy consumption per household will be the use cases for validating the developed big data analytics platform. These use cases also provide the basis for designing, planning and implementing schemes for improving energy related services for sake of achieving higher efficiency in both production and usage that contributes to cause of greener environment. The insight generated from these use cases can also help in educating the consumer about benefits of energy conservation and spread the awareness about behavioural changes that can benefit society as well as individuals.

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

In a nutshell, this thesis focuses on providing a solution for collecting, storing, analysing and visualizing 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. This thesis also provides the models for knowledge discovery that can be used to improve energy efficiency at both producers and consumers ends. The big data analytics platform developed as part of this thesis is not limited to be used only for energy efficiency. It has the capability of handling other big data uses cases as well. However, within scope of this document we shall only discuss its use for energy usage patterns detection and efficiency.

1.1 Problem statement

Energy conservation is required to reduce C02 emissions from energy production and usage. To achieve this goal we need to understand and improve the energy efficiency on both producer and consumer end. ICT enabled smart energy grids and devices are being rolled out globally to measure energy consumption and improve energy efficiency. These smart devices produce high volumes of data that may or may not be predicted and planned at time of setting up the infrastructure. The data generated by different devices comes in different formats. For knowledge discovery from this data it is required to collect, store and analyse the data and then visualize the generated insights in a way that the information can be understood efficiently. 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. So to cater for all this 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 scope of this research we provide a model for big data analytics platform that can provide the solution for 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.

1.2 Helpful hints

For referencing we shall be using Vancouver system[35]. When discussing from authors point of view we shall be using Author(s) name, year of publi-

cation as our format along with sequence numbering from Vancouver systm. In case of quotation from author we shall be using double quotes e.g. “quotation as it is”.

Throughout the document we shall be discussing about energy. Due to our main focus, The term energy in our research and this document will refer to electricity or electric power. In case of all other types of energy we shall be specifically mentioning the type name along with energy as a term.

In this document we shall be discussing about the concept, development and use of a big data platform as our main environment for our data analysis. The terms like platform, data platform, and big data platform will be used to refer to the same concept. In case of a specific need of any other platform like concept we shall be giving proper descriptions.

1.3 Structure of the Thesis

to be written at document finalization stage.

Chapter 2

Background

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

2.1 Smart grids

Energy industry across the globe is facing numerous challenges. There is a huge pressure from regulatory authorities and environmental organizations to reduce carbon foot print, expand their renewable energy portfolios, and 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 on electric

[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 jeopardized.

power"[14]. Traditionally, power system participants have been strictly producers or consumers of electricity. The demand response and reliability issue with conventional electric power distribution models on consumer side are causing a major trend in motivating consumers to produce electricity at domestic level mostly using the renewable energy production methods. " Prosumer" is an emerging term used for an economically motivated entity that: [20]

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

The current energy grids support unidirectional distribution models and are centralized in nature. They are very limited to handle the prosumer needs. Line loses and hierarchical topology makes them less reliable. They usually become bottle neck when rapid adaptations are required for demand response. Farhangi, 2010 define smarts 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, the smart grid needs to empower its stakeholders to define and realize new ways of engaging with each other and performing energy transactions across the system" [17].

In our research, we used data collected from smart metering devices as part of a pilot smart grid project. The data was used to generate analysis that recommends improvement for both demand and supply side to achieve energy efficiency as well as provide understanding to enable correct decision to adapt for demand response.

2.2 CIVIS project

CIVIS is the abbreviated name for "Cities as drivers of social change" project under European Union 7th framework. It is a part of the programme for optimising energy systems in smart cities. CIVIS project is a collaborative effort of 10 European universities) [2]. It aims to embed the social aspect

[2]1. Associazione Trento RISE, Italy 2. Aalto university, Finland 3. Imperial College London, UK 4. ENEL Foundation, Italy 5. Instituto Superior TÃ©cnico, Portugal

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 complex socio-techno-economic systems with multiple decision making layers that are in effect at the physical, cyber, social, and policy [16].

ICT acts as one of the main enabler of smart grids, distributed and bidirectional information flow models. On the other hand ICT also provides a lot of new mediums for social aggregation e.g. internet based social media. CIVIS projects tends to connect these two different dimensions with innovative ICT solutions. An integrated approach to energy efficiency is the basic manifesto of CIVIS project. [16]

Understanding energy usage patterns and benchmarking energy efficiency performance of small units within cities are some preliminary items in list of CIVIS objectives. Within scope of our research we analyse energy data to understand the consumption patterns and try to evaluate various factors that can effect directly or indirectly on the usage patterns. We also try to classify the building on basis of energy efficiency and try to test the sensitivity of energy efficiency with respect to 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 research.

2.3 Green campus initiative

Green campus initiative is a project by VTT "Technical Research Centre of Finland" . It is part of EcoCampus 2030 program. 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 [25]. 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 gain energy efficiency by building infrastructure in the building units that can make them self sustain for future requirements. The aim is build to build a performance

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

based ecosystem that can help both consumers and producers to adapt with demand response.

Green campus initiative is a pilot project for EcoCampus program in which VTT has installed smart devices inside Aalto University, Finland campus building in cities of Espoo and Helsinki. These specialized devices contained smart metering for energy consumption and indoor environment monitoring sensors. The data used for analysis in our research was collected from a subset of buildings as test sites of this project. The data includes hourly consumption of electricity and electricity used for heating. For one of the test sites VTT provided us the data with device level energy consumption details i.e. electricity used by different home appliances. This was achieved using smart NIALM [3][22] meters that can distinguish between different electric devices used on basis of their signal thumb print.

Apart from providing the data, VTT green campus researchers 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 advance data analytics techniques on large volumes of data. Advance analytics is a generalized term used for data analysis techniques like statistical analysis, data mining, machine learning, natural language processing, text mining and data visualization etc [37]. Although volume of the data is a widely used factor for qualification of a data set as big data but when it comes to big data analytics there few other important attributes i.e. variety, velocity, valuation and veracity. The concept of 3Vs (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"[28]. 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 below

Number 0 to 3 represents the scale of data that you perceive 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 is not providing 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. Valuation supports the

[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

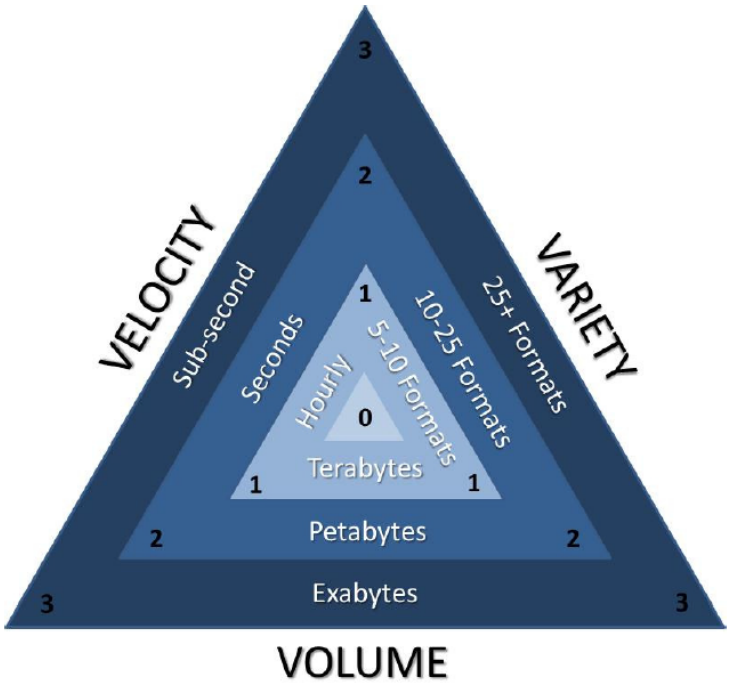

Figure 2.1: Gartner 3Vs of data and data magnitude index [28].

decision making by considering the value of outcomes against the efforts required to collect, manage, process and analyse huge 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 thesis or stating any criteria for qualification of a data set as big data. Instead we shall be proposing an advance analytics model that should be capable enough to handle big data as well other smaller data sets on need basis. The modular architecture of the model platform can be tweaked to handle volume, variety, velocity, and veracity on need basis while trying to maximize the valuation for the use case. In following subsections we shall discuss some of the relevant technological advancements that enables to handle 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 MapReduce and Hadoop

It is hard to predict the size of data and computing power required to pocess it 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 gets very expensive. So the viable option is to scale out [5] using required number of smaller machines with relatively low computing resources in parallel. We need a system that can handle large scale parallelization. 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 processing. 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" [15]. This paper describes MapReduce as "programming model 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 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 implementation of MapReduce 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 [15]. It was released by Apache foundation. Apache foundation has also built many supporting tool around Hadoop framework to support end to end big data analytics ecosystem e.g. Apache flume for data collection, Hadoop File system (HDFS) for storing, Apache Pig and Hive for processing, Apache Mahout for machine learning etc. We have used some of these tools within scope of our research.

MapReduce and Hadoop are batch processing frameworks that empower

[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.

[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.

processing of large volumes of data using commercial grade low cost computing infrastructure. So it supports volume and valuation directly. Variety can also be supported with support of all format files into associated files system e.g. 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 MapReduce and Hadoop cannot handle. The next subsection answers the question of velocity.

2.4.2 Real time big data processing

Real time data processing is generally associated with live streams of data. Real time data can be processed and analyzed 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 context to large volumes of historic data. To apply advance analytics models like machine learning active feedback loops are also necessary. Even stored (non live data) big data, applications require data processing system to answer queries very fast. To fulfil these industry driven requirements technology is in rapid advance mode. In 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 system, frameworks and tools are being developed as the evolution path for MapReduce and Hadoop. All of them have their own purpose, strengths , and limitations. They are mostly used in combinations based on use cases. We shall not be discussing or comparing these systems and tool. Instead, in this article we shall be briefly discussing 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 system 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. Batch layer manages the historic data set and pre-compute the batch views. Serving layer indexes the batch views so the queries can be served with low latency as compared to traversing through complete data set. 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.[33][9]

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

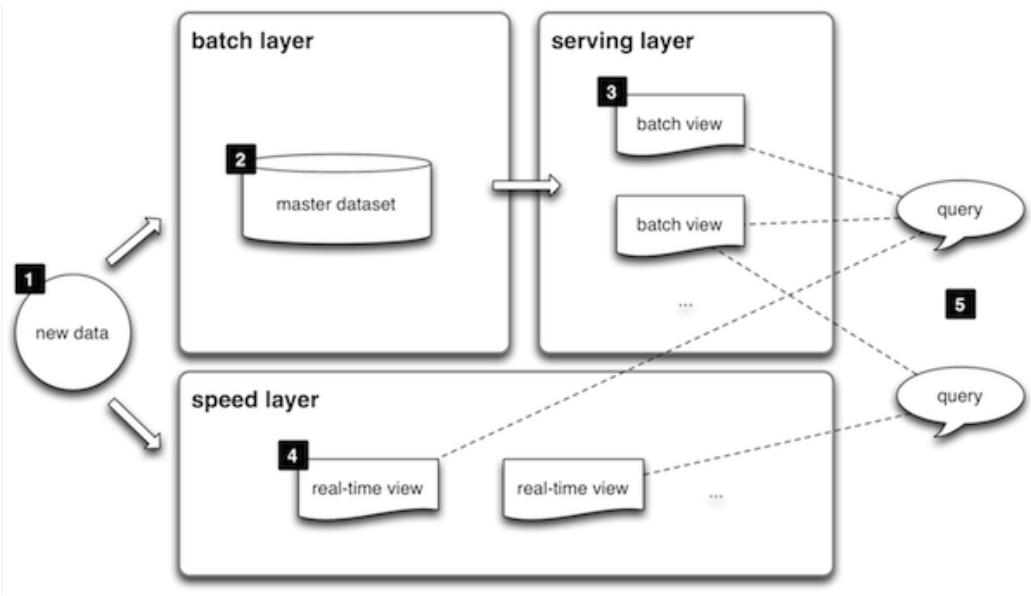

Figure 2.2: Lambda Architecture [9].

2.4.2.2 Massively parallel processing - MPP databases and query engines

MPP based architectures use multiple independent computing resources like severs, processors and storages to execute processing jobs in parallel. Most of the MPP based database approaches implements 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 University of California Berkeley in 1986 [38].. 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 source SN database.

Despite high scalability and other positive aspects, SN databases needs a lot of manual work in terms of partitioning the data, tuning the data and load balancing etc. So usually maintenance such database systems is expensive. MapReduce and Apache Hadoop ecosystem provides high level of automation along with scalability, flexibility and fault tolerance. However parallel batch processing is not as fast SN based MPP databases. Merging both the models solves can solve all these issues. Cloudera Imapala is one of the example of a MPP based on-line query engine that runs natively on top of Hadoop [5]. It can provide MPP like query response time performance with processing power and flexibility of 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 previous sections of this chapter, we have highlighted the importance of energy conservation. We discussed the advancements in pervasive smart energy device and 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 shall explain the main motivation and the theoretical concept behind data analysis part of our research.

Unprecedented challenges arising from increasing dependency on conventional energy are part of a global phenomenon. Improving energy efficiency is an important mean to tackle these challenges. Like other economies, European Union is 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 [6]. Member states agreed to reduce 20% of the EU's primary energy consumption by 2020 in European Union of council March, 2007. EU's Energy Efficiency Directive 2012 [6] defines energy efficiency as the ratio of output of performance, service, goods or energy, to input of energy. This definition was first discussed in 2006 in European commission 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 energy efficiency and to evaluate the savings, 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"[19]. 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

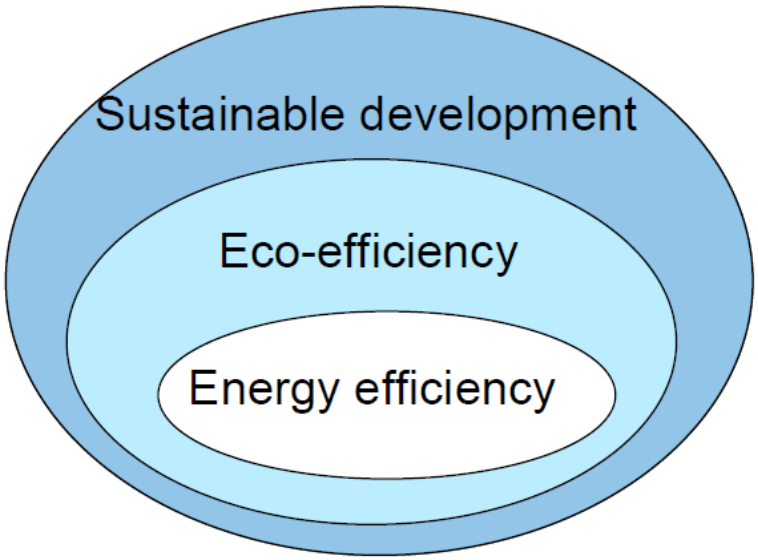

Figure 2.3: Energy efficiency, eco-efficiency and sustainability[19].

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 eco-efficiency model presented in VTT's report. We calculated energy efficiency of the buildings on basis of formula deduced in Chapter 5 (equation 5.1 and 5.2) of the VTT's report [19].

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

In case of a specific energy consumption (SEC) [19] 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 meter square. In subsequent sections we shall be referring to these equations when we try to identify the usage patterns on

building level, discuss the relevance of energy efficiency with these patterns and then discuss a model for classifying buildings on energy efficiency .

2.6 Daily consumption patterns, base load and user load

Daily consumption pattern of a building unit corresponds to the respective usage of the building. Understanding daily usage patterns can help in identifying the optimization point for improving the energy efficiency of that building unit. Base load of a building is one important metric that can be detected through observing the daily consumption. Base load is the consumption that takes place regardless of the actual use of the building and of the user's energy consumption[19]. 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 base load also include some energy consumption by functional components inside building like computer servers, lab equipments, and refrigerators etc. However VTT differentiate this load from user energy load that is characterized by the direct involvement of the users of a building. For example an office building that has peak load during day time because user are using various additional appliances like personal computers, coffee makers, lights etc compared to base load that is generated during night time when office building is not in use. Figure 2.4 illustrates the concept of base load and user load.

Energy efficiency of base consumption and energy efficiency of user load are shown in figure 2.4 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 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 weather. The energy consumption trends vary with outside temperature. Among other things electricity or the other energy types required for the air conditioning in the buildings is 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 winters while in warmer regions energy consumption increase

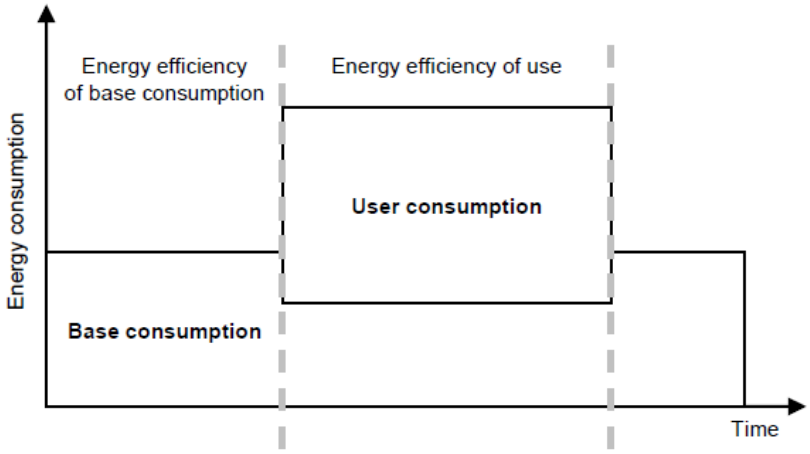

Figure 2.4: Base load , user load and energy efficiency [19].

is expected in summers because of the air conditioning requirements. Energy service providers usually conduct demand planning with consideration of seasonal trends.Considering 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 building. 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 document where we shall discuss the results of our analysis.

Previously, there have been many studies for 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. all, 2003[40] and Yigzaw G. Yohanis et. all 2007[42] study the energy consumption pattern in Hong Kong and United Kingdom respectively. Buildings units e.g. residential houses apartments and commercial offices etc were considered as basic unit of analysis. Yigzaw G. Yohanis et. all methodology resembles most to our approach as they considered ecological factor along with energy efficiency calculated in similar way as equation 2.1 and 2.2. As discussed before, the main purpose of VTT's green campus initiative under 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. Alligned 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 basis energy efficiency that 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 larger number of 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 consumptions unit that has improved their energy efficiency performance.

Classification for fault detection analysis of a building energy consumption has been used previously as well. Xiaoli Li et. all, 2010 used classification along with outlier detection mechnism to identify the energy inefficient building [29].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 identify the daily usage patterns using auto regression technique and pass the results to benchmark against any outlying data point that can refer to faulty behaviour. Imran Khan et all. 2013, proposes different clustering techniques to group building with similar level of energy efficiency together[26]. In our research we used a hybrid method using feature extraction and trend detection techniques like Xiaoli Li et. all[29] and then applied a clustering technique proposed by Imran khan et. all [26]. The clustering technique that we used is called K-means clustering. It is explained in the next subsection of this article.

2.8.1 K-means clustering

K-means is an algorithm for cluster analysis. In context to machine learning cluster analysis or clustering is an unsupervised task of grouping a set of objects in a way that objects in same group are similar to each other more than the objects in other group. K-means algorithm clusters the set of objects i.e. energy efficiency values in our case into predefined number of classes. We

shall term these values as data points. K represents the number of cluster and groups that we can set in start of the process. K-means means algorithm was first proposed by Stuart Llyod in 1957[30] but the k-means term was first used by James Mcqueen in 1967[31]. 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 original Lloyd's algorithm there has been some modification in methodology. Two very commonly used methods are Forgy method [18] and Hartigan-Wong method[23]. In our approach we are using Hartigon-Wong method. We shall also use some references from Forgy method when explaining the K-means algorithm.

K-means groups the data points in cluster with a logical centre point. The aim of the K-means algorithm is to divide data points in certain dimensions into K clusters so that the within-cluster sum of squares is minimized [23]. Lets assume if we want have K cluster for data points D = $\{x_1, x_2, \ldots, x_n\}$ in d dimensions then

$$x_i \in R^d$$

K-means algorithm uses following steps to cluster data into groups[36].

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

Mathematically randomly initialized centroid 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 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 k-means is a set of feature vectors along with the number of clusters required. In our case we shall be have two feature hence two dimensional matrix of energy efficiency values for electricity and electricity used for heating. Before inserting data to 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 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. Conditional demand analysis (CDA) model has been one the most commonly used method for UEC estimations. K. H. Tiedemann, 2007 explain CDA as a multivariate regression technique which combines utility billing data with weather information and customer survey data to produce robust end-use energy consumption estimates[39]. As part of the green campus project VTT has installed used state of the art nonintrusive load monitoring (NIALM)[22] devices that can distinguish between the usage of different electric devices on basis of changes in voltage and electricity.

We are using the data collected by a NIALM device installed in one of a residential apartment included in VTT's pilot test sites. In our analysis of this data we are not using CDA. However we shall be using auto regression along with concept of moving averages in form of a model known as ARIMA model to estimate the future consumption of a device depending on the previous usage. This is an example of quantitative forecasting. Before we go to discuss about ARIMA models it is important that we briefly discuss the basic conditions for quantitative forecasting and the time series analysis as foundation for prediction model based on ARIMA.

2.9.1 Main conditions and Steps for Quantitative Forecasting

Rob Hyndman et all, 2014 discuss two main conditions for application of quantitative forecasting[24] 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't be met then qualitative forecasting is the only option. However, the qualitative forecasting is not in scope of our research for this thesis. 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 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 shall discuss it more in data analysis part of tis document.
5. Using and evaluating 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 times is termed as time series analysis [13]. A time series usually consists of a deterministic component and a random component[34]. 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 form of trends, periods, and jumps etc. Figure 2.5 illustrates the examples of different time series. In each illustration there is atleast one stochastic random component with and without deterministic components.

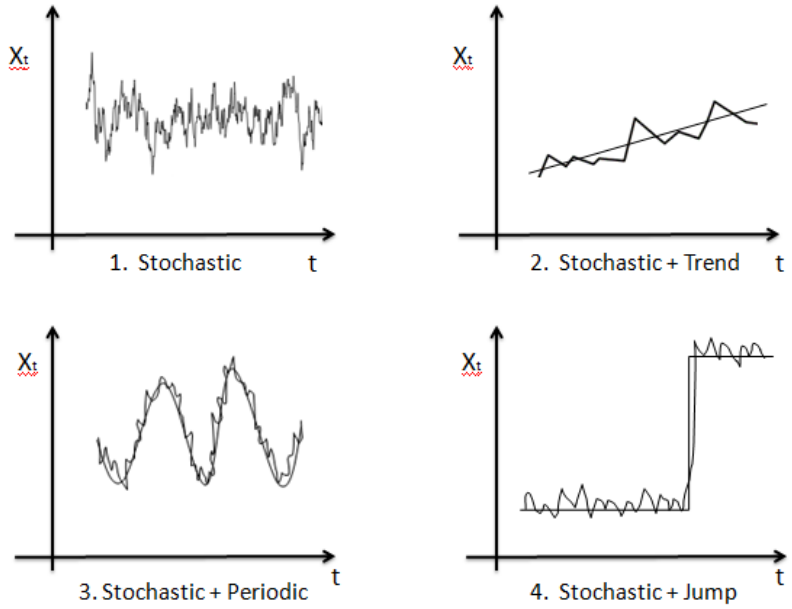

Figure 2.5: Time Series types[34].

In figure 2.5 illustrations 2,3 and 4 contain a deterministic component with a random component. In terms of prediction even the random component can be estimated using the deterministic component. However for stochastic random time series data without any deterministic component it is very hard to predict anything accurately. The time series with no predictable pattern is generally termed as stationary time series. We shall be discussing this in details during the analysis part of this document.

2.9.3 Autoregression, Moving Averages and ARIMA Models

Rob Hyndman's book "Forecasting: Principles and Practice"[24] 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 liner 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 Autoregression

Autoregressive model is based on the concept of a variable regressing on itself. For autoregression we can drive equation as

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

The aim for good estimation is to select value of β_0 and β_1 that can minimize the sum of square of errors. Above equation can be used to estimate the value based on first previous value. 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.

[6]The differences between consecutive observations

2.9.3.3 Moving Averages

Moving averages model use past forecast errors in regression like manner to forecast future time series values instead of using past time series values as in autoregression. Matehmatically 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 autorgresion and moving average models. ARIMA is one of the most commonly used forecasting technique. ARIMA model can handle time series data with and without seasonality. We shall be discussing non-seasonal ARIMA because of the nature of the data we shall be processing in our analysis. The nature of the data will be discussed in data analysis part of this document. So combining the autoregression and moving averages using equation 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 autoregression.
- d is the number of non seasonal differences.
- q is the order of 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.

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

AND

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

AND

$$y_t - y_{t-2} = (1 - B)y_t$$

In general a dth order difference is written as

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

Rearranging equation 2.10 and using backshift notations we can have following equation with labeled p,d and q for ARIMA model.

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

Explanation and the equations used in section 2.9.3 were cherry picked from Rob Hyndman's book "Forecasting: Principles and Practice"[24] as refernce to theorey 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 need intensive computation. We shall be using 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 shall try to 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. On a practical level following were some major tasks that were required to fulfil scope 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.
- Developing an end to end big data analytics platform.
- Data collection, storage and preprocessing.
- Use case specific data analysis and evaluation of results.
- Visualization of 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 need to have results before visualizations 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 literature review for understanding each component of our research was also an ongoing process through the time line for this thesis. The regardless of sequential and parallel tasks we need to iterate for continuous improvement.

To tackle these challenges we needed a methodology that can support sequential and parallel task execution with support for iterations to improve. Like most of scientific researches, fail fast and small to succeed was the key for us. In the list of tasks mentioned above. Most of them requires conceptualization and tested quickly using rapid prototyping. Taking it as a software development task initially, we had some candidate models such as water fall model,agile development model, spiral model and incremental model etc. Here we shall briefly discuss the advantages and disadvantages in context to our research project.

- **Waterfall model** offered the simplest approach of requirement engineering, design, implement, test and operate our research. How ever it is inherently sequential and had weak support for iterations.

- **Agile development model** Agile methodology[32] 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 CIVIS projects was very hard.

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

- **Incremental model** relies on small incremental steps with each step consist of independent design, implement and test phase. In the beginning, Incremental model was the best fit among other candidate models for our thesis research.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 platform development and data analysis part. It has started creating integration over heads. For example integrating two different data processing tools together for a single use case becomes difficult when they were configured in two different incremental steps.

Learning from the problems that we face from incremental model we altered our approach to adapted version of another very flexible software research and development methodology known as "Kumiega-Van Vliet Trading System Development Methodology"[27].

3.1 Kumiega-Van Vliet Model

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. waterfall model, spiral model and stage gate model. We have already explained the waterfall and spiral models. 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. Process can not move from a stage to other 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.

($K|V$) model tries to overcome the short comings of these three models by combining them to a single paradigm for trading system development[27]. In spiral model in start smaller time is allocated to four basic steps i.e. research planning implementation and test. These four steps can be performed again and again in cycles. To avoid spiral to grow two much after each cycle a stage gate controls if process can be passed to next stage or it needs to be sent back to perform another cycle in same stage. Just like waterfall there can be number of stages. But for continuous improvements process there is an iteration channel available unlike traditional waterfall model.

3.2 Adaptation of Kumiega- Van Vliet Model

($K|V$) model is designed for software research and development in 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 other instead of stage gate approvals in ($K|V$) model. The waterfall model alike stages helped in keeping our focus on the solution 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 was include in the scope of research for 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 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 start 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 like VTT were also involved in some of the review meetings. We shall be discussing it in details when we describe our stage wise proceedings. At end of each review session a decision was made to either move to next stage or try to improve via 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 iteration had caused some major or minor changes in stage deliverables as well. Small informal team structure reduced management and communication efforts. 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 research methodology is explained in 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. First cycle of the spiral had to pass through the four steps. The additional cycles are initiated if the further improvements are decided for deliverables in review session. All Four steps were not mandatory for additional cycles. In this section we shall be listing and describing the stages along with respective steps. We shall be highlighting some major cycles and deliverables. However the iterations will be discussed in next section. Figure 3.1 will be our main reference through out this section. In this section we shall mention some functional components of our project e.g. logical architecture, data processing tools and algorithms etc. Details for these functional

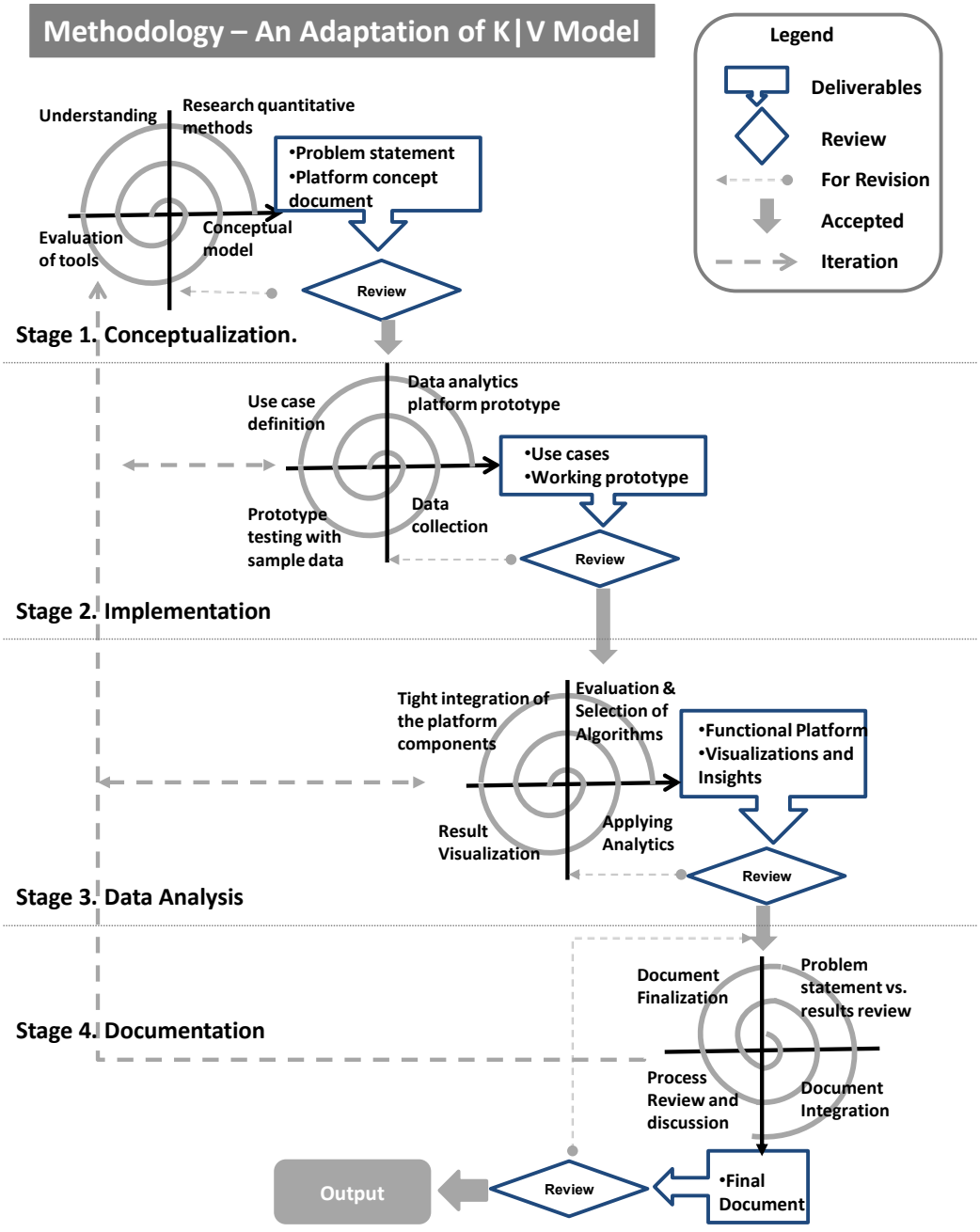

Figure 3.1: Methodology, An Adaptation of $K|V$ Model

components will be given in later part of this document.

3.3.1 Stage 1. Conceptualization

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

3.3.1.1 Step 1. Understanding

From the start, 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. Follwoing are 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 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 Visualization, as an introduction to effective communication through data visualization.

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

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 etc were evaluated. Identification and theoretical evaluation of advance 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 big data platform concept paper deliverable. This step was also repeated during various iterations, whenever change was required in data platform.

3.3.1.4 Step 4. Evaluation of Tools

This step was in pair with 3.3.1.3. All the tools listed in conceptual model were tested during this step. A checklist of evaluated and selected tools was maintained. This list is available as annex???

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. cycle for producing the required deliverables and one additional cycle for modification of platform concept document. The main modification in additional cycle was the replacement of application frame work with an architecture diagram to clarify.

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 concept paper as a deliverable from stage 1. This stage had 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 this chapter. However here we shall list the final 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.
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.

3.3.2.2 Data Analytics Platform Prototype

This step involved the practical implementation of 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 visualize data. Details of the components and implementation will be discussed in 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 policy 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 later a web sevice was opened for us to collect 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.

data. the details of the data will be provided later in the 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 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. Real time data was also collected using the same platform to support some CIVIS projects objectives. However the real time data was not included in scope of analysis presented in this document.

3.3.2.4 Prototype testing with sample data

This step was only used during first cycle of this stage and first iteration of whole process. The purpose of this stage was to test the full flow from data collection to data visualization using the developed prototype. The sample data was the randomly selected records from hourly consumption data set. Although in this step we started with smaller sample and then kept on increasing it. The complete data set was also tested. In testing 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 advance analytics tools integrated within our prototype.

7. Data visualization.

3.3.2.5 Stage 2 deliverables

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

Stage went through two additional cycles on top of the first mandatory cycle. Within two additional cycles all the steps were performed except data collection that was collected once during the this stage for first cycle. However data collection was repeated within iteration that will be discussed in later in this chapter. The major revisions inside cycles includes 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 the 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 use cases. We applied the basic and advanced analytics techniques introduced in 2.6,2.7,2.8,2.9 sections. This stage has following four steps.

3.3.3.1 Tight integration of the platform components

In 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 manually as in input to the other module on requirement basis. In this step we tried to automate the process by coupling the modules together in 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 advance analytics algorithms available for supporting our uses cases. It involved quantitative methods considered in section 3.3.1.2. However the focus was more on the advance analytics. The techniques explained in 2.6,2.7,2.8,2.9 sections were selected during this step. The selection criteria for each algorithm will be discussed in chapter 5. 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 sets. The insights generated from this steps are 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 Visualization

For the sake of ease to understand, the extracted insights in te previous step were visualized in form of data graphs. Different tools for visualizations were used in this steps. Visualization tools will be discussed in chapter chapter:platform. For data visualization we tried to implement the graphical practices discussed by Edward Tufte in his book "The visual display of quantitative information"[41]

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 visualization 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 $(K|V)$ model, we just had one review session for this stage per iteration. Combination of different algorithms, their evaluation and then generating visualizations had to be repeated and tested many times. So reviewing in the intermediate cycles was very inefficient. This stage took longer time because of many cycles and the wider spiral required to produce good quality results.

3.3.4 Stage 4 Documentation

Documentation was an on going process throughout the stages and iterations. All of the stages had at least one deliverable in form of some document e.g. stage 1 has big data platform concept paper, stage 2 had 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 shape of one single thesis report. Following were four steps for this stage.

3.3.4.1 Problem statement vs. results review

This step was the check for 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 form of one consistent story. During this stage we tried to link the inter stage documentation together along with theoretical background and explanation of the process and 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. 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 indicate 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 in end

3.4 Iterations

Iterative processes and work models do not require full specification right from beginning. Instead the implementation can start with small part of specification. Then in a step wise approach the next scope is defined with consideration of lesson learnt and new directions found from previous iterations. This inherent characteristic of iterative processes had a vital role in our research. We started with 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 advance analytics topics. Findings and practical implementation in early phase enabled us to expand our scope later. We added more use cases with more focus on data analysis and application of big data for energy efficiency. Iterations also helped us in improving the quality of research.

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

Chapter 4

Concept and Implementation of Big Data Analytics Platform

We have referred to big data platform concept and implementation on many occasions in 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 **??** we mentioned the functional components and their implementation during different stages, cycles and iterations. In this chapter we shall first explain the concept and a sample application framework for a big data platform based on available open source components. Then we shall 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 **??** of 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 **??** 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 5V s of big data.

1. **Volume** refers to the size of the data. Volume is the most commonly associated feature with the big data. The big data analytics platform in our scope of work is based on Hadoop File Systsem (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 data procesing speed.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 structure of the data. Traditionally the relational data base system can store data with fixed schema. The fixed schemas mean that the stored data must have a definitive structure. Such databases are designed on basis of these data structures. In context to 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 formats. In our conceptual model we have added various components that can handle all formats of the data. However, In our implementation we shall be processing the data that had known fixed schema.

4. **Veracity** refers to the complexity due to noises and inconsistencies of the data. In real life scenarios for big data it is very rare to find data in absolute consistent. Most of the time some values will be missing or the data will be in wrong format or there will pollutions in data. For good analysis we need to take care of such inconsistency and errors in data. Our conceptual model is capable of handling inconsistencies and in our implementation we had catered for some inconsistencies that we shall discuss in next chapter 5.

5. **Valuation** refers to the benefits of processing big data against the efforts required. It is an emerging feature for big data analysis design. Just like the other IT systems, organizations tends to decide about big data investments by looking at the business cases. In our concept we shall be discussing 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. However their are cloud alternatives that can be use within our model with some costs. However we shall not be discussing such alternatives in scope of this document or our research.

4.2 Data Analysis work-flow

The data analysis process involves collection of data from multiple heterogeneous sources including both social media data, consumer data, sensors data, and already 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. Data mining engine requires a feedback loop to 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 database for visualization frontends. Figure 4.1 shows the high level process flow.

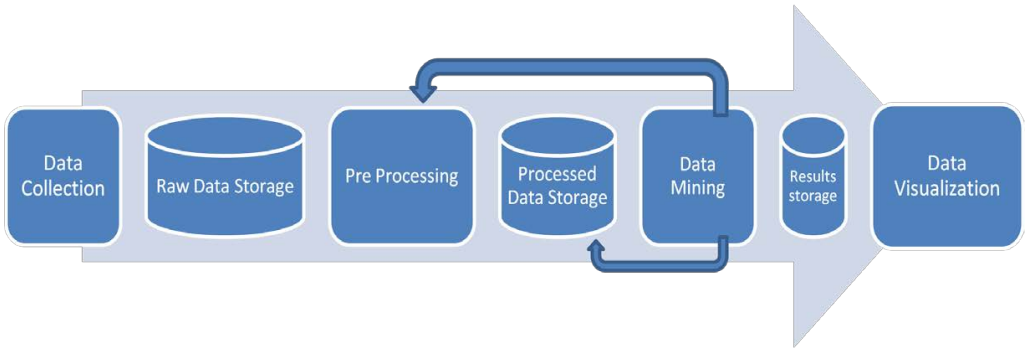

Figure 4.1: High level data processing flow.

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 available that can be used as efficient alternatives for parts of this model. However this paper does not contain information about these alternatives. Figure 4.2 illustrate the proposed concept

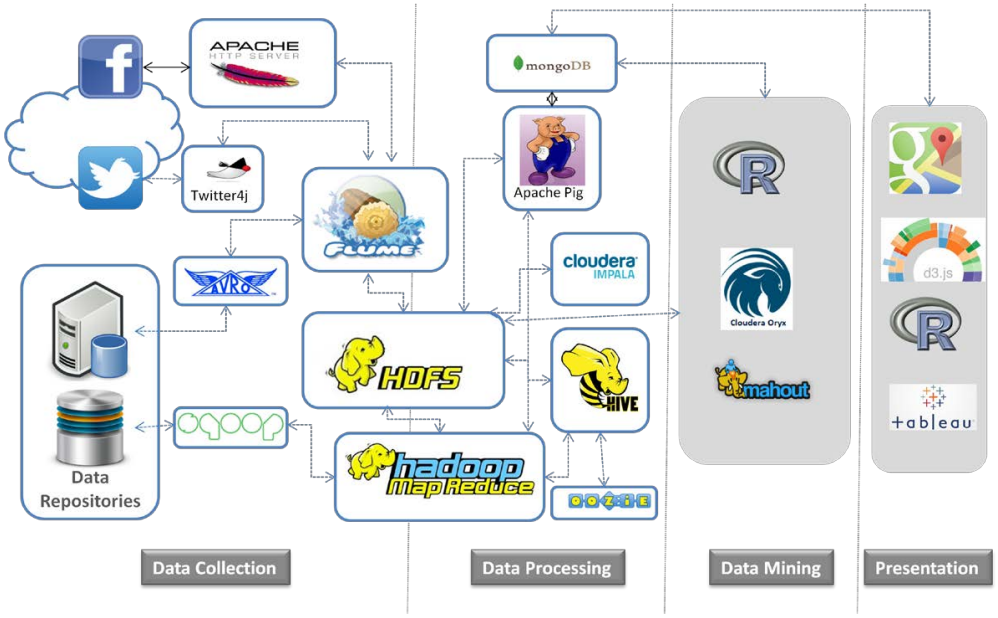

Figure 4.2: Conceptual model of big data analytics .

4.3.1 Data Core

Before we go into 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 MapReduce 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 large data sets across clusters of computers and HDFS is a distributed file system that provides high throughput access to application data[11] 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 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 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 amount of data [1]. 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[10] data serialization system to collect log data from file systems of remote servers. Flume is then ingesting data directly into hadoop file system (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 Apache foundation has another useful tool called Sqoop.

4.3.2.2 Apache Sqoop

Apache Sqoop [4] 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 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 definitive structured forms e.g. schema. Furthermore, certain filtering can be applied for example in case of tweets, 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. Usually pre-processing also helps in filtering out the unwanted information and make data lighter for 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. For sake of fast pre-processing Cloudera Impala is also also added to the model.

4.3.3.1 Apache Hive

Apache Hive [2] is data ware house software that provides a way of providing schema to the stored data with SQL based query language to extract, transform and load data (ETL). Apache hive is further supported by another Apache Hadoop ecosystem tool called Oozie. Ooozie is acting as a workflow scheduler for Hadoop i.e. while loading the data into Hive it can create partition for tables arrange the data for optimized querying.

4.3.3.2 Apache Pig

Apache Pig [3] provides a high level scripting language to analyze the data stored in HDFS using MapReduce.

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 Hadoop ecosystem.

Cloudera Impala uses HDFS as its main data source. It can read data directly from a 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 schema it can project the data in 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 that uses parallel batch processing. However Cloudera Impala currently supports limited data structures. For handling some complex data formats e.g. Avro serialized JSON format data. For such cases it can be configured to work together with Hive and Hbase etc.[8]

4.3.3.4 Databases

For storing the pre-processed data various choices of available open source databases can be applied in this platform. While 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 online 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 analyzed by the data mining modules in the platform. There are various open source data mining and statistical analysis tools available online with 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 process data in memory so it requires large RAM (Random Access Memory) for large data sets.The real power of R is in availability of large number of statistical and advance 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 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 the Mahout's algorithm library.

4.3.5 Presentation

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

Tableau public provides the easiest solution to visualize data through 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 VizQL (Visual Query Language) paradigm [21]. 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 visualize the results.

D3.JS provides a comprehensive library in Java Script for making customized information graphics. It is very flexible and powerful. However it need 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 information.

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

4.4 Implementation

In this section we shall be discussing our implementation for analysing the energy consumption data for our energy efficiency use cases and collecting social media data for supporting CIVIS project. The implement model is the subset of the conceptual model presented in previous section with at least one additional component that we shall discuss later in this section. Figure 4.3 shows the implemented model. The details of configurations are available in Appendix B.

4.4.1 Implementation Environment

Cloudera distribution including Apache Hadoop (CDH) version 4.7 was used for implementation of our big data analytics platform. 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 form of a pre-configured quick start virtual machine (VM) capable of running on top of any known operating system e.g. Microsoft Windows xp/7/8 , Linux RHEL/Centos, and Ubuntu etc. Cloudera CDH 4.7 quick start VM itself was running on Centos 6.2 operating system. 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 testing phase we have also test multi node cloudera CDH 4.7 configuration on cloud. However there was cost associated to running the set up on cloud and the requirement of our use cases were also fulfilled by single node quick virtual machine. So we preferred not to run analysis using 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. 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 quick start VM environment. Typically we were using Windows 7 with similar dedicated hardware resources as for quick start VM. Tableau public is a web service running in 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 CIVIS project activities.

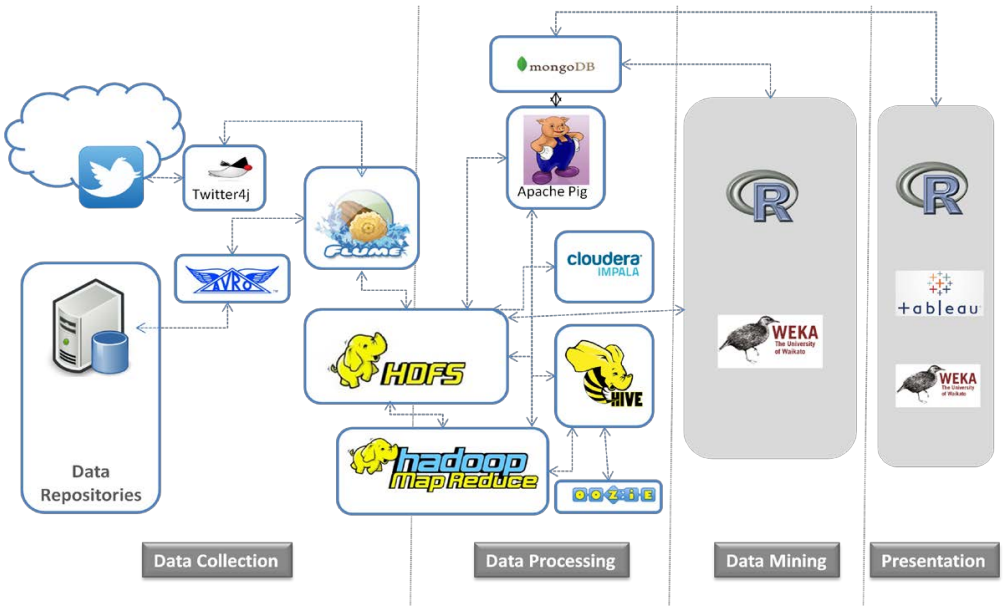

Figure 4.3: Implemented big data analytcis platform .

In this section we shall 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 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 there 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 can download data in off-line mode. Here off-line mode means that data was not collected automatically and human intervention was required to collect the data.

The collected data was then ingested to HDFS manually. Once the data is available on HDFS then data pre-processing tools can access it. Selection of the preprocessing tool is based on use case requirements, format and volume of data. As discussed before, VTT provided us two types of data i.e. hourly

electricity consumption data from smart metering devices and second set of NIALM device level data. For first data set of hourly electricity consumption data we used Apache Hive to pre-process the data. The decision was based on the fact that data had fixed schema and Hive has good support for such data types. Alternatively, Apache Pig and Cloudera Impala could have also been used but a choice was made on basis of preference. For performance test purposes we had used Cloudera Impala and Apache pig. We shall discuss details of that in next chapter. For data analysis purpose following schema was created in hive and data was loaded as per following schema.

```
CREATE EXTERNAL TABLE hourly_data
(
    devid string,
    building string,
    meternumb int,
    type string,
    day string,
    hour int,
    consumption int

)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
```

After loading the data was processed to prepare inputs for data mining and advance analytics modules as per use case requirement e.g. for classification on 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 comma separated file. This file was then read by Tableau Public. We then generated the visualizations 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 visualizations. 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 evaluation step as describe in section 3.3.3.2 as it provides a quick mechanism for analytics prototyping. It can not be used in tight integration within platform so our conceptual model does not include Weka as an option. For visualizations we used R plots and Weka graphs.

4.4.2.2 Social media data collection for supporting CIVIS project

To support CIVIS project activities we configured our platform to collect and store Twitter data using Twitter streaming API. The implementation procedure for achieving this case was based on a Cloudera Tutorial "How-to Analyze Twitter Data with Apache Hadoop" [7]. We configured Apache Flume with Twitter4j Java library to capture the twitter stream 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 HDFS by Apache Flume HDFS sink mechanism. 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. Apache Ozie workflow was implemented to archive hive data in 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 Results

Chapter 6

Discussion

Chapter 7

Conclusion

At this point, you will have some insightful thoughts on your implementation and you may have ideas on what could be done in the future. This chapter is a good place to discuss your thesis as a whole and to show your professor that you have really understood some non-trivial aspects of the methods you used...

Chapter 8

Conclusions

Time to wrap it up! Write down the most important findings from your work. Like the introduction, this chapter is not very long. Two to four pages might be a good limit.

Bibliography

[1] Apache flume. http://flume.apache.org/. Accessed: 2014-07-04.

[2] Apache hive. https://hive.apache.org/. Accessed: 2014-07-04.

[3] Apache pig. http://pig.apache.org/. Accessed: 2014-07-04.

[4] Apache sqoop. http://sqoop.apache.org/. Accessed: 2014-07-04.

[5] Cloudera impala.

[6] 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.

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

[8] 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.

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

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

[11] Welcome to apache hadoop. http://hadoop.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] Box, G. E., and Jenkins, G. M. *Time series analysis: forecasting and control, revised ed.* Holden-Day, 1976.

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

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

[16] 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.

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

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

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

[20] 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.

[21] 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.

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

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

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

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

[26] 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.

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

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

[29] 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.

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

[31] 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.

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

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

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

[35] 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.

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

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

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

[39] Tiedemann, K. Using conditional demand analysis to estimate residential energy use and energy savings. *Proceedings of the CDEEE* (2007).

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

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

[42] 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 Platform Components

Appendix B

Platform Configurations