Big Data Analytics

Feiyang Xue

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-128
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-128.html

May 15, 2015

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2015

Electrical Engineering & Computer Sciences

Signal Processing and Communications

Big Data Analytics

Feiyang Xue

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: _____ Date _____

Print Name/Department:

2. Faculty Committee Member #2:

Signature: _____ Date _____

Print Name/Department:

Big Data Analytics – Capstone Final Report

University of California, Berkeley

Lulu Huang, Sherrie Lim, Dylan Rhode, Subramanian Shankar, Feiyang Xue

ABSTRACT

With the use of Big Data gathered from over 50 buildings on the campus of University of California, Berkeley, and the SAP data analytics tool - HANA, our team has designed two solutions that seek to optimize energy and water usage. This document presents the problem we set out to solve, the results of an industry analysis carried out to determine the marketability of our solutions, our intellectual property strategy, our individual contributions to the technical aspects of the project, and our individual concluding reflections.

Table of Contents

1. Project Overview, Goals & Motivation

The objective of this big data analytics project is to identify a problem of public interest and solve it through the analysis of big data. The general approach we have taken as a team is as such: Identify the problems to be solved, gather relevant data, develop models, apply data to the models and demonstrate analysis results through the use of visualization tools.

Having gone through several rounds of iteration at the start of the academic year, we identified the problem to be solved as one that involves the optimization of energy and water consumption. Specifically, we analyzed large sets of weather, energy consumption and water consumption data, gathered from over fifty buildings on the campus of University of California, Berkeley, by a research group known as LoCal, to develop universal models that serve to introduce greater efficiency into electricity and water consumption on campus and beyond. Given the broad nature of this project goal, we further divided this project into two narrower sub-problems.

In the first project sub-problem, we created an optimized, water-efficient irrigation schedule for sprinklers. This project sub-problem required the analysis of existing weather data and the utilization of forecasting techniques. By making predictions for future weather patterns, we formulated recommendations for an irrigation schedule that ensures sufficient and yet non-excessive irrigation. The motivation for the development of this solution stems from the need to conserve water, an increasingly scarce resource given the drastic increase in global population over the last century, as exemplified by the severe drought in California. By reducing sprinklers' water output on rainy days or days when evaporation is high, the solution helps to minimize water wastage, thereby leading to the conservation of water.

In the second project sub-problem, we designed an algorithm that identifies malfunctioning equipment by spotting deviations from its predicted energy consumption trend. This project sub-problem required the application of machine learning algorithms, which learn from existing energy consumption data, and the classification of equipment as either functioning normally or malfunctioning, to perform new classifications based on future energy consumption data. The motivation for the development of this product stems from the need to conserve electricity. Often

times, equipment malfunctioning episodes are accompanied by drastic changes in the equipment's electricity consumption. Being able to identify the malfunctioning equipment allows for speedy intervention, therefore leading to a reduction in wastage of energy. There are also many cases in which malfunctioning equipment can lead to significant economic loss. Malfunctioning refrigerators in biology research laboratories, for instance, lead to the loss of viability of experimental samples, which might be costly or difficult to replicate. Early identification of malfunctioning refrigerators and any other equipment performing critical tasks minimizes the costs associated with such loss.

2. Industry Analysis

This section details the industry analysis performed to assess the marketability of the two solutions we have created. We studied the social, technological, economic, and regulatory trends to determine whether the trends favor the introduction of the solutions into the market at this point in time. We also utilized the Porter's Five Forces of Competitive Position Analysis to determine where the power lies in the network of buyers, suppliers, new entrants, rivals, substitutes, and suppliers in the market.[1]

a) Trend Analysis

The success of a product often depends on the macro-trends in society during the point at which the product is developed and commercialized. This section will detail the current social, technological, economic and regulatory trends that may influence the commercialization of the irrigation scheduling solution and the malfunction detection solution, and how the solutions can be positioned to leverage or mitigate them.

- **Social**

The last century has seen a drastic increase in the global population and a corresponding increase in the global demand for water and energy resources. The scarcity of water and energy resources, as a result of the high demand, has led many government and corporations down the path of developing means to optimize the usage of water and energy resources. The U.S. Environmental Protection Agency's Water Conservation Strategy, developed in 2008 (EPA's Strategies to Meet Its Federal Requirements), and the ongoing collaborations between the U.S. Department of Energy, academia, and national laboratories (Energy Efficiency) are but few of the many examples that testify to the society's growing commitment to maximize the efficiency of water and energy usage, a trend that favors the commercialization of the two software solutions. To leverage this favorable trend, the software solutions can be marketed as solutions that optimize

[1] Porter's Five Forces of Competitive Position Analysis is a framework developed by Michael E Porter of Harvard Business School in 1979. It is used to evaluate the competitive advantage a business organization has over the buyers, suppliers, new entrants, rivals, substitutes, and suppliers in the market.

water and energy usage, thereby increasing their relevance to the market and consequently, their demand.

- **Technological**

Digital revolution in the recent years has brought about the capabilities of generating, storing and handling copious amount of data. According to a McKinsey Global Institute published report titled "Big data: the next frontier for innovation, competition and productivity", enterprises around the globe stored more than 7 Exabytes of new data on disk drives and consumers stored more than 6 Exabytes of data on their personal computers and notebooks in the year 2010 (Manyika, Chui and Brown).[2] Despite this, the numbers are still rising, and much value can be generated from the use of big data in businesses and policy-making decisions. This hype around big data favors the commercialization of the software solutions, which are dependent on big data streams as input. In addition to lowering the price of the general infrastructure, such as networks of sensors, necessary for the functioning of the software solutions, the current technological trend also increases individuals' and enterprises' confidence in the capabilities of big data analytics, thereby allowing them to recognize the value of the software solutions.

- **Economic**

The 2014 Economic Report of the President provided a favorable outlook of the U.S. economy: the economy has been growing for 4 consecutive years and more than 8 million new private-sector jobs have been created. The government has cut its deficits by more than 50% and the housing market is showing signs of rebound (United States Government). These economic conditions provide a favorable backdrop against which the software products will be commercialized. A healthier economy implies that enterprises are more capable financially of making new investments and are more likely to purchase the software products to better optimize their water and energy usage, and increase profits.

- **Regulatory**

[2] One Exabyte of data is equivalent to more than 4000 times the data stored in the US Library of Congress, which has 235 terabytes of storage in April 2011.

Given that big data related technology is a rapidly developing field, the regulations governing the use of big data are patchy and have been a topic that is widely debated. Attempts at formulating formal regulations relating to the ownership of public data and the usage of personally identifying information were fraught with many challenges. In response to the lack of formal legislation, corporations in the industry have designed a series of self-regulatory frameworks to guide their usage of big data. In some cases, these frameworks have been inconsistent with the existing Federal Trade Commission's (FTC) guidelines, and have resulted in a number of civil actions brought by the FTC (Sotto and Simpson). This regulatory environment may complicate the process of commercializing the software products. While there is no personally identifying information involved in the use of the software products, the issue of ownership of the data used by the software solutions necessitates a closer look at the guidelines prescribed by the FTC. By ensuring that the frameworks guiding our use and ownership of data are aligned with that of the FTC, we will be able to avoid incurring unnecessary costs.

b) Porter's Five Forces Analysis

Besides the macro-trends identified in the previous section, the various forces present within the market also influence the success of the software solutions. Specifically, the forces presented by the buyers, rivals, substitutes, new entrants and suppliers will be discussed.

- **Irrigation Scheduling Software Solution**

i. Buyers

'Buyers' is a collective term for end-users and distributors of the product. There exist several industries that could be the end-user of the irrigation scheduling software – the landscaping, agricultural, and horticultural industries are but a few of the examples. Thorough examination of the size of these industries has led our team to the conclude that the Precision Agriculture Systems and Services industry, with 296,303 farms that make up approximately 55.8 million irrigated acres of land in U.S., will make the most lucrative market by consideration of its sheer demand (United States Department of Agriculture).

Given the size and the extensiveness of the market – the top 10 agricultural producing states, which include California, Texas and North Carolina, span the entire country – the solutions can

only be brought to the market with the aid of distributors (United States Department of Agriculture). Utilizing distributors to reach the end-users helps to overcome numerous logistical challenges, including installing the software and sprinkler infrastructure and providing necessary technical support. By adding distributors to the value chain, we will, however, be exposed to greater risks associated with the power of both the distributors and the end-users.

To mitigate the buyers' power, we can first seek to increase the number of buyers. As the number of buyers increases, the percentage of sales that can be attributed to each individual buyer decreases, thereby giving them less leverage. In the case of the distributors, there are plenty that currently exist in the market. Examples include TWC Distributors, Imperial Sprinkler Supply and Atlantic Irrigation. By effectively engaging as many distributors as the manufacturing operations allow, we can ensure that our dependence on individual distributors is kept to the minimum. In the case of the end-users, the market can be segmented according to the size of the farms – defined by their annual sales, as shown in Figure 1 (U.S. Environmental Protection Agency). While the retirement farms and residential farms are high in numbers, they are likely to be small farms that do not require automation in optimizing their irrigation schedules. Limited resource farms and farming occupation (lower sales) farms, which make up approximately 560,000 farms and are likely to require automation in sprinklers' control, make an appropriate market segment to target for the purpose of maximizing the number of buyers.

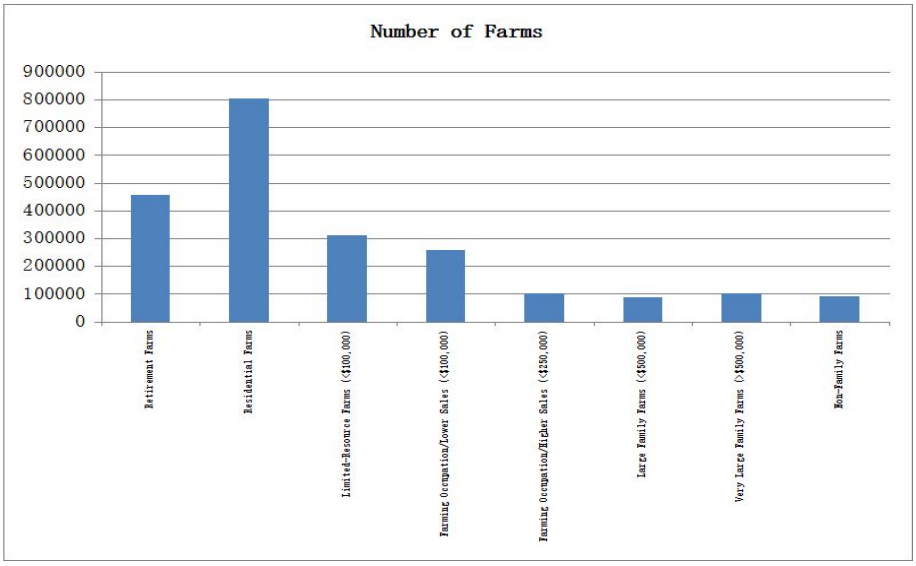

Figure 1. Farms Classified According to Their Sizes

Another factor that influences buyers' power is the switching costs involved in adopting the use of an alternative product. While it is difficult to mitigate the power of the distributors, given that many of them are entitled to the right to distribute multiple products, it is possible to decrease the power of the end-users. The switching costs for the farms are inherently high given the vast capital outlay involved in the installation of sprinklers with network capability. If the sprinklers and the software can be designed such that their compatibility is unique, the switching costs will be increased significantly, and the end-users will be less likely to switch to an alternative system.

In addition to mitigating buyers' power, improved marketing of the irrigation scheduling software to potential buyers is another way to ensure maximum profitability. The marketing strategy can be divided into four categories – Product, Price, Promotion and Place.

The product in this case is the sprinkler irrigation scheduling software. Marketing a product, however, goes beyond merely stating what the product is – it involves positioning the product in a way that satisfies a certain consumer need. As mentioned in the above section, people are now keen on using scarce resources in a more efficient manner. Marketing the software as a product that will result in less water wastage and lead to cost savings will therefore make the product more appealing to potential buyers.

The target end-users of the product are limited resource farms and farming occupation (lower sales) farms. Generally, smaller farms have fewer financial resources and may not purchase the product if priced out of their reach. Consequently, the product should be marketed as an affordable solution that will lead to cost savings. A more detailed economic analysis can be carried out to determine the optimal price such that the greater volume of sales achieved as a result of the lower price makes up for the loss in profit.

Targeted advertising in regions concentrated with farms will help promote the product to potential end-users. Also, we can hire a sales team to assist the distributors in prospecting for potential end-users and following up with established customers.

Finally, we will begin operations by selling domestically, with end-users scattered across the country in top agricultural producing states such as California, Texas and North Carolina. This will ensure that the operations remain tractable, while guaranteeing sufficient sales and capturing a sizeable portion of the market share upfront. Upon establishing our position domestically, we can then look to develop internationally.

ii. Rivals

The market identified for the irrigation scheduling software is moderately competitive. Within the market, Trimble Navigation Limited is the largest rival, capturing 22% of the market share. This is followed closely by Deere & Company at 12.9% (Neville). However, significant differences exist between our irrigation scheduling software and those offered by the two major rivals.

Of all Trimble's solutions, the Irrigate-IQ Precision Irrigation Solution offers functions that are most similar to ours. While it can be used to monitor water usage, the main purpose of Irrigate-IQ is to remotely control the irrigation pump systems, allowing the user to choose how much water is used and where in the field is irrigated (Irrigate-IQ Precision Irrigation Solution). Unlike our irrigation scheduling software, Trimble's solution offers no scheduling recommendation based on weather and environmental factors.

Deere's Field Connect measures moisture levels and presents that information in an online dashboard (John Deere Field Connect). Like Trimble's product, Field Connect does not provide any additional analysis. Consequently, the irrigation scheduling software solution stands apart because it will be targeting an underrepresented portion of the market: farmers that require a decision support system that will gather data and make recommendations for optimized irrigation schedules.

iii. Substitutes

There exist few acceptable substitutes in the Precision Agriculture Systems market. A substantial portion of farmers may consider simply doing without an irrigation scheduling software. Others may look into purchasing genetically engineered (GE) crops that require less water. Each of

these alternatives, however, has significant limitations. Completing the irrigation scheduling by hand, or by experience alone, is cheaper in the short term, but using big data to minimize the water required will greatly reduce costs in the long run. In addition, under drought conditions such as those currently found in California, any amount of wasted water has serious financial and social consequences. GE crops, on the other hand, will likewise reduce costs in the long term. Current trends towards organic and natural fruits and vegetables, however, as well as general public distrust of GE foods, may prohibit many farmers from taking that step.

iv. New Entrants

The threat of new entrants for both software solutions is high. This is because of the relatively low costs involved in development, which in turn can be attributed to the decreasing costs of equipment, the open source nature of data and software, and improvement in processing power. An IBISWorld Business Environment published report titled "Price of Computers and Peripheral Equipment" explains that the highly competitive nature of circuit and computer manufacturing will result in a 4.8% fall in prices of computers and peripheral equipment annually over the next five years (IBISWorld Business Environment Profiles Price of Computers and Peripheral Equipment). The open source nature of data also means that a potential new entrant may not need to spend a large amount of capital to acquire the data necessary for analysis. Improvements in processor speed over the recent years have also led to strong competition within the data processing and hosting services industry, such as those between IBM and Amazon Web Service (Diment). This competition translates to lower costs of computing and processing, thereby further lowering the barrier of entry into the software development industry.

In order to mitigate the threat posed by new entrants, the software solutions have to be developed and introduced to the market quickly. By establishing ourselves early, we will be able to capture a sizeable portion of the market. Given the relatively high switching costs involved in uninstalling existing software solutions to purchase a new one, it is likely that end-users will remain loyal to us once software solutions are set up.

v. Suppliers

The threat posed by suppliers is low. As mentioned in the *New Entrants* section, many factors that go into the development of the software solutions such as sensors, data and processing

capabilities are becoming increasingly cheap. The similarity in terms of supplies offered and the existence of competition within the suppliers' industry further keeps the suppliers' power in check, preventing them from charging exorbitant prices or limiting quality of supplies.

- **Malfunctioning Equipment Detection Software Solution**

i. Buyers

We identified the Machinery Maintenance and Heavy Equipment Repair industry as the main market for the malfunctioning equipment detection software solution. Given that most, if not all machines and equipment need to be powered by some energy source, the means through which malfunctioning equipment is identified is applicable for a broad range of machines.

Consequently, there are many potential users for this solution. We will, however, only be targeting the manufacturing companies as the main end-users for the initial launch. This decision was motivated by the fact that manufacturing companies often utilize heavy equipment on a relatively large scale, the malfunctioning of which can lead to significant wastage of energy. Given the size of the market, as in the case of the irrigation scheduling software solution, we will utilize distribution channels to reach the end-users.

To mitigate the powers of the buyers, we will seek to increase the number of buyers by engaging multiple software distributors such as Softchoice, TechXtend and Software Spectrum. Fortunately, the number of end-users is already high given the multitude of manufacturing companies in a $422.2 billion industry (Mataloni, Shoemaker and Aversa).

Buyers' price sensitivity also influences their power. By targeting the bigger companies in the manufacturing industry such as GlaxoSmithKline, 3M, and Ford, we can ensure that our end-users are not highly price sensitive. This is because big companies are likely to have greater financial resources and are more likely to pay a higher price for a software solution that they find valuable.

Just like the irrigation scheduling software, marketing the malfunctioning detection software to potential buyers well is essential to ensuring maximum profitability.

To enhance potential end-users perception of the product, it can be marketed as one that will allow for early intervention when equipment malfunctions, thereby leading to significant cost savings, both in terms of the reduction in power wastage as well as the damages resulting directly from equipment malfunction.

The target end-users of the product are large manufacturing companies. As mentioned in the above paragraph, these companies are likely to have deeper financial resources. Hence, the product can be marketed as a premium, robust software that comes at a higher price.

As in the case of the irrigation scheduling software, we can hire a sales team to assist the distributors in prospecting for potential end-users and following up with established customers. This is especially essential given that large manufacturing companies typically have managers in charge of overseeing the equipment. It is necessary that the sales team build a strong relationship with these managers to ensure the companies remain loyal to our software solution.

Finally, the software solution will be sold domestically, with end-users being scattered across the country. This will ensure that the operations remain tractable, while guaranteeing sufficient sales and capturing a sizeable portion of the market share upfront. Upon establishing our position domestically, we can then look to develop internationally.

ii. Rivals

The Machinery Maintenance and Heavy Equipment Repair Services industry, in which the malfunction detection product is most applicable, is highly fragmented. Over 90% of the businesses have less than 20 employees, and the four largest competitors have a combined 4% of the total industry revenue (Harris). There is also a wide range of products offered, from Wood Group's Integrity Management solutions that prepare manual inspection and maintenance plans (Wood Group), to Advanced Technology Services's Predictive Technologies that analyze the machinery with highly advanced sensors but still require operators (Advanced Technology Services, Inc.). Our malfunctioning detection software solution will reduce maintenance costs and electricity waste by capturing energy usage data on a continual basis, and then alerting users to electrical spikes or falls that may indicate equipment failure. It is uniquely positioned to

produce the same results as the competitive products, but constantly and with minimal human supervision.

iii. Substitutes

The substitutes in the Machinery Maintenance market are few but strong. Instead of taking the time and money to install sensors in each piece of equipment in case one eventually fails, many manufacturers are likely to take a reactive approach and only evaluate the machine after it is broken. At that point they will either require the assistance of technicians, or they will buy a new machine. These two responses are simple and require little initial investment or infrastructure. In the end, however, they create even bigger problems than they solve. Running the equipment to its breaking point is a good way to shorten its lifespan and close down production lanes with no warning. It will ultimately save much more money – through reduced maintenance costs, electrical waste, and lost sales from failing machines – by introducing continuous monitoring. This will allow quick and easy repairs the second a machine begins to break, rather than permitting it to continue working at partial capacity and causing further damage in the future.

iv. New Entrants

See *New Entrants* Section under Irrigation Scheduling Software Solution

v. Suppliers

See *Suppliers* Section under Irrigation Scheduling Software Solution

3. Intellectual Property Strategy

The two software solutions will not be patented. For an idea to be patentable, it is required to be novel, non-obvious, and useful. The ideas central to the two solutions are not entirely novel. In the case of the irrigation scheduling software, we are applying existing models to new, public data. In the case of the malfunction detection software, we are utilizing publicly available machine learning algorithms. Given that both the irrigation models and machine learning algorithms are not entirely developed by our team, we cannot claim them to be novel. Attempt at patenting the process of integrating the data assembly, data cleaning, model application and results visualization, is also fraught with challenges. This is because the core of our work involves codes written with SAP analytic software program. Consequently, SAP may attempt to assert a claim on the patent should it be granted.

The patent offering a solution that comes closest to one of our software solutions is titled "Big Data Analytics System" (Watson Scott). Developed by Applied Materials, Inc., it is a method comprising the process of data collection, obtainment of manufacturing parameters, and identification of real-time data for storage both in-memory and in distributed forms. Like the patented process, our malfunction equipment detection solution requires the assembly of data from sensors installed in over 50 buildings in University of California, Berkeley, obtaining parameters of common equipment, and analysis of real-time data. Given that the data sources, data storage, and analytics methods utilized in our project are unique to our project, we do not need to license our technology.

A preferred alternative intellectual property strategy that will work for us is to keep the software solution designs as trade secrets, protected by the institution of access control, and technological and legal security measures such as non-disclosure agreements and non-compete classes in employee contracts.

Keeping the software designs as trade secrets is advantageous in several ways. First, obtaining trade secret status is cheaper and less onerous than obtaining a patent. The team does not have to conduct in-depth patent searches and will not require extensive attorney services. Second, the

protection of a trade secret can extend indefinitely, unlike that provided by a patent, which only lasts for a limited period of time.

Despite its advantages, there are several risks associated with keeping trade secrets. First it might be possible for competitors or end-users to dissect the integrated processes of data assembly, data cleaning, data analytics and results visualization. To mitigate this risk, the team could utilize asymmetric encryption to generate product licenses, thereby limiting outsiders' ability to dissect the code that goes into the product. Second, the level of protection granted to trade secrets is generally weak compared to the protection granted by the patent. Having the trade secret lie in the hands of certain individuals means that the team runs the risk of an individual breaking the employee contract and divulging the trade secrets for his or her personal benefits. While it might be possible to reclaim losses by filing civil suits in courts against this individual, the exposure of the trade secrets to outsiders means a permanent loss of the respective process.

4. Technical Contributions

a) Overview

Our team is divided into three sub teams. They are the irrigation scheduling team, malfunction detection team, and programming infrastructure team. I am in the programming infrastructure team. My focus is on the computer science part for all sub project teams. The tasks are in two main categories. One is providing computer science support to other teammates, including building scripts, code complexity analysis, method suggestion, etc. The other part is presenting other teammates' outcome, which includes building web app, demonstration of model, etc.

My work is greatly relevant to the success of our team. In the early stage of our project, I helped the team do the feasibility study of various data sources. Based on these studies, the team could make an informed decision of which direction should the project go. In the next phase, I focused on data extraction and and measuring of the speed of extraction. By my doing these tasks, the team gathered required data to develop models and made a advised schedule of project progress. I also manage the data uploading to SAP's HANA system. This part includes the data table designing, building data upload scripts, and database management. My final task is building a web app demonstrating other teammates' results. The specific tasks are building an event log for the malfunction detection team, building a repeatedly updating prediction graph for the irrigation scheduling team, and building a web app that incorporates the two previously mentioned systems. My work pushes the overall progress and gives the team a visual way to exhibit the findings.

b) Knowledge domains

My tasks requires knowledge in many domains, and they are mainly in three parts. The three parts are data collection, data processing, and web app building. These knowledge are mainly leaned from other researchers, SAP engineers, and public forums online. In this section, I will focus on how I get the knowledge rather than the ways I take to execute. The detailed explanation about the methods will be discussed in the next session.

The data collection is the key part of building a model. Without training with enough amount of data, the model would have only trivial parameters which would prevent the model from outputting meaningful outcomes. The data in our project is from a Berkeley research team called Software Defined Buildings (SDB). Beginning in 2011, the SDB team, which was called LoCal project team at that time, started to collect the data from the buildings in the campus of UC Berkeley (LoCal). The data covers a variety of information of a building, such as the power consumption, water usage, temperature and humidity, wind speed, and cloud coverage. The data could be accessed in the openbms website using either graphical user interface (GUI) or SQL query. There is also a python library called sMap developed in the computer science department in UC Berkeley to help simplifying the data extraction process (SMAP 2.0 Documentation). The data collected by SDB team and the documentation of sMap library enables the execution of data collection part. Using the data from SDB team could help in transforming the data to solution that creates value, and potentially help the SDB team to discover the corrupted data and keep the database alive.

The data processing part is also a key component of this project. We did an estimation of the data size at the beginning. The SDB has metadata corresponding to each data source they stored in their database. Thus, counting the number of metadata and download part of data is the reasonable way to estimate the size. The specific method will be talked about in the "methods and materials" section below. The estimated size of all the data stored in SDB's database is about 120 GB. Giving the size of the data, it would not be a feasible choice to do analysis on a single computer. The project is supported by the company SAP with their technology of HANA system (SAP HANA). The HANA system provides an simple SQL console for executing query command on a relatively huge database in a reasonable amount of time. Most of the needed knowledge in this part are from consulting SAP engineers and HANA user guide or documentation of specific libraries. HANA system is a commercial product which has many components, and the engineers in SAP do not necessarily knows how to operate every part of the system. The components we use are python drive which can connect to the HANA database, HANA studio which provides a GUI, and the predictive analysis library (PAL) that does the prediction using machine learning. In using these components, we also helped SAP catch bugs in their products. By doing this, we contribute in building a stable commercial product.

The web app building part is for the visualizing of the team's outcome. The web app is built using the Django web framework. The documentation of Django makes the most part of the literature review in this part (Django Documentation). Another important source of knowledge is from the online forum, for example, stackoverflow website (Stack Overflow). These online forums act as platforms for the programmers to exchange ideas, ask for help, and share solutions. In participating in discussion and up vote the effective solution, the team helps the follow programmers who have the same problem resolve issue more efficiently.

The knowledge domains in this project are in data collection, data processing, and web app building. Due to the nature of this project and the resource the team has, most of the time are used not in reading papers, but in interacting with other computer science related person and manual of systems or libraries. In this process, our project also helps in exploring the value from data, discovering bugs in other systems, and leaving feedback for future programmers

c) Methods and Materials

There are different methods and materials used in different stages of the project. The project can be separated into four stages -- defining problem, gathering data, building model, and demonstrating results. These stages overlap with each other sometimes, but the general chronological order is like this. There are unique methods in each part related to specific tasks. These methods will be detailed in the chronological order above.

- **Defining Problem**

In the defining problem stage, The team needs to check the data to find out what does the data contains, as well as the size and extraction speed of the data. By understanding the data, the team can figure out what kind of problems the team is capable of solving.

i. Metadata

The data from SDB team is stored in a database that accepts SQL query. However, the query needs information from the metadata to be built. Thus, examining the metadata is one of the

priority in this stage. The first task is to do an estimation of the data. After talking to the members in SDB team, our team learned that the metadata about other data is stored in a metadata called 'buildingmanageronline archive'. I took a few samples from this metadata, and figured out that the structure of the description of data is like a JSON object. The metadata of each data source contains a tag called 'UUID', which is needed by the sMap python tool to extract data. I gathered 100 random 'UUID' from the metadata, and extract the data from SDB's database. For each data source, I extract the data from the year 2010 to 2011, and write the data to file to estimate the size. I also repeated this process for accuracy. For each year from 2011 to 2015, I use another randomly selected 100 data source to extract data, and record the size and time accordingly. The table below shows the values recorded.

Year*	**09-10**	**10-11**	**11-12**	**12-13**	**13-14**	**14-15**
Size (MB)	0.0	14.1	15.4	13.9	10.0	13.4
Time to extract (s)	29.1	92.2	112.6	100.7	96.8	126.2

*Year starts at January 1st.

Table 1: Data Size and Extraction Time For 100 Data Sources in Each Year

It is worth noticing that the data has no even distribution on years. I also noticed that many data source are just empty for particular years. Given the above values, I estimate that one data source may contain 0.67 MB data. Counting the number of entries of the Metadata, we know there are 187,726 existing data sources. Thus, the total data size would be about 122 GB. However, it would not be necessary to download all the data. Finding out needed data source is the next step. The other task in this stage is to transfer the metadata to a human-readable form, so the other teammates could read and find what they are interested in. The problem is that a single metadata does not contain all possible fields, a field many contain either a value or a few sub fields, and the values of fields are usually not in a fixed length. Below is a sample of a metadata in the metadata file. Notice that each metadata has its 'UUID' as its own key in the file.


```
"00002f84-f73a-534e-8cfb-3905797dcb91": {
    "Metadata/Extra/Type": "priority",
    "Metadata/Location/Building": "Sutardja Dai Hall",
    "Metadata/Location/Campus": "UCB",
    "Metadata/PointName": "SDH.S7-08:CTL FLOW MAX:PRIORITY",
    "Metadata/SourceName": "Sutardja Dai Hall BACnet Archive",
```

```
    "Path": "/Siemens/SDH.PXCM-09/SDH/S7-08/CTL_FLOW_MAX_PRI",
    "Properties/ReadingType": "long",
    "Properties/Timezone": "America/Los_Angeles",
    "Properties/UnitofMeasure": "Level",
    "uuid": "00002f84-f73a-534e-8cfb-3905797dcb91"
}
```


ii. Two Human-readable Form

Giving these constrains, there are two options to transfer the data into human-readable form. The first is to build JSON object and write it to text file with correct indentation, and the other is to build a table from the data with each field corresponds to a column in the table. The JSON object method is quick to build, and it would clearly shows what each data source do in half of screen size. The problem of this method is the readers cannot compare between data sources, and would not have a overall view of all the possible fields. The table method would provide a overall view of all possible fields and enables the reader to compare between data source. The disadvantages of table method would be more coding needed and inability to read full description of a data in one screen. The number of all possible fields presented is over two hundreds. For a single description, there will be many fields marked as blank in the table, and the readers need to scroll to the very end of the table to see all values of existing fields.

iii. Final Decision

The teams prioritized the being comparable feature after a team meeting since we also want to aggregate the similar data source in the future, so we adapted the table method. I build a script to scrape through the data from ‘buildingmanageronline archive’. For each item in the query result, I explore its key, and put the keys into a set. I keep the set for the next item and union the newly acquired key set with the old one. After first scraping through the entire results, the column name of the table is extracted one by one from the set. The script then scrapes the metadata again and assign corresponding value or ‘NULL’ string to every column. After doing this, our team has built a table with full description of all data stored in SDB team’s database. Other teammates aggregates the rows they are interested in from the table, and entered into the next stage.
Below is the code for the process of transforming, and a sample for the resulting table.

```
data = open("metadata.json").read(os.path.getsize("metadata.json"))
loaded = json.loads(data)
finalKeys = set()
for i in loaded:
        if loaded[i] == {}: #in case of an empty item.
                continue
        else:
                newS = set(loaded[i].keys())
                finalKeys = finalKeys.union(newS)
fieldnames = list(finalKeys)
for i in fieldnames:
        tempDict[i] = ''
with open("Everything.csv","w") as csvfile:
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
        writer.writeheader()
        for i in loaded:
                if loaded[i] == {}:
                        continue
                else:
                        for j in fieldnames:
                                if j in loaded[i]:
                                        tempDict[j] = loaded[i][j]
                                else:
                                        tempDict[j] = ''
                        if 'uuid' not in loaded[i]:
                                tempDict['uuid'] = i
                        writer.writerow(tempDict)
```

DZ	EA	EB	EC	ED	EE	EF	EG
a/moteid	uuid	Metadata/metadata/location/desk	Metadata/Location/Country	Metadata/Extra/StationType	Properties/UnitofMeasur	Metadata/cooling_tower	Path
	42e3a82e-f00a-57ee-b54a-ff3c8cc736db				percent		/94129/1300humidity
	b39072dc-0c69-5b88-a360-e2fa4196f2c7				C		/InternalMass/EP_I-T_0-W_0-H_0-O_1-W_2-A_1-T_0-M_0/SUPPLYPLENUM:ZoneMeanRa
	f61500db-c957-518a-ab03-e4d26daffb64				m/W		/ttpc_acmes_SDB02/925/true_power
	c434aa93-658b-5df3-b7c6-229e06088c63				degrees true		/93314/1900wind-dir
	8b84c46a-a975-57ae-ad2e-d2b6dc6142ec				Fahrenheit		/95982/0800temperature
	25b20f21-4f8f-5fa0-a5e3-2f69361540eb				percent		/92101/1400humidity
	b9b291fa-a06d-552c-926b-14f683cc9f0c				98		/Siemens/SDH.PXCM-08/SDH.S6-04:VLV POS
	3060d547-10c9-5b02-b197-15d837f4b357				kW		/r/c36ae3bb-61e2-5cc6-8c6f-0225e6cf9ca2/3060d547-10c9-5b02-b197-15d837f4b357
	336a508f-d20b-52c2-b858-39d21dce91bd				Fahrenheit		/92164/1000temperature
	f93528a4-654d-5f81-a4f9-58f54b859b52				$/MWh		/r/6c0080f8-44ff-5130-8571-afebf438a51b/f93528a4-654d-5f81-a4f9-58f54b859b52
	92745c5d-ede3-5aae-a90c-33ee343dd90e				kWh		/m252-2/elt-A/ABC/true_energy
	17bc05a8-915c-5475-be38-97414fe3d978				$/MWh		/r/a6c3c273-ed5d-589b-87f5-bc04324e2873/17bc05a8-915c-5475-be38-97414fe3d978
	5b3086e3-219c-524a-b8aa-6f9009757a29				knots		/94587/1000wind-speed
	321beb27-a8df-5176-a001-3483ec53a476				knots		/90248/0100wind-speed
	a1eb51c3-9ef4-5eea-8533-e719b347f706				W		/InternalMass4/EP_I-T_0-W_0-O_0-M_1-T_3/ZONESUBSURFACE1:SurfaceOutsideFaceC
	b7b01196-03d1-58f9-9a7c-7a3b3b06e786				64		/Siemens/SDH.PXCM-04/SDH.S1-04:CTL STPT
	c66f8282-c4d1-5d4c-b668-a48b83f74aab		USA		$/MWh		/ISO-NE/LMP/LD.ROLF_AVE13.8 NETWORK NODE/Forecasted Energy Component
	85328ecc-03b1-5ed1-b0bf-6a1421ceeeb8				Level		/r/7412e29e-df14-5f9a-a0bf-89a41e8e54da/85328ecc-03b1-5ed1-b0bf-6a1421ceeeb8
	7601d7dd-3a20-55a8-b998-64a2b87df068				percent		/94550/1700cloud-cover
	770523cb-d825-581a-a18c-78ea04327e44				inches		/91016/0200rain
	03253f47-1240-508f-8c61-c063d9352ce3				percent		/95387/0200humidity
	ccafb4cf-9468-5b98-ae5a-0fb7273d940c				cfm		/r/63c6bba6-fac9-5adc-800c-c122563cde45/ccafb4cf-9468-5b98-ae5a-0fb7273d940c
	423e83ee-aa01-511e-be79-6d1a66bd1910				degrees true		/95210/1000wind-dir
	b8c7fdce-ca89-5bc0-a65e-227818be5d29				$/MWh		/r/6c002cc5-7130-5655-af7b-7e3c40b411d5/b8c7fdce-ca89-5bc0-a65e-227818be5d29
	44ca3e23-c750-54a5-9b31-63cf8414cdd8				inches		/90670/2000rain
	85b0352b-5544-5dfb-accd-c40b8d21bed0				103256016		/r/8497a2a1-9d30-52e5-93b5-459968263c66/85b0352b-5544-5dfb-accd-c40b8d21bed0
	68b41164-308b-5548-ac57-aa0ee1481baa						/r/6b2fe50e-4bdf-50ae-9c5f-266a40d5d604/subsample-3600
	4b135c0e-d2da-5d39-b46e-2168bb4c49ac				W		/InternalMass4/EP_I-T_0-W_0-O_2-M_1-T_0/GEOMETRICMASSSURFACE:SurfaceInside
	24a381ca-1df2-5d12-975b-0803acfacce5				$/MWh		/r/0d50ef13-b9ed-51c1-a2d9-4d706ad65f75/24a381ca-1df2-5d12-975b-0803acfacce5
	46324bbd-f457-50b1-9090-f0b5e89bb14f				percent		/96074/0200humidity
	5b448c56-78fd-5f06-9f07-ec65bef63d2b				Fahrenheit		/90703/0800dew-point
	9092ea71-a758-5086-bfc3-6c2ac0962696				Fahrenheit		/94038/1400dew-point
	fcc9fc91-4306-5977-8939-ba5fdc2ccdbe				47		/Siemens/SDH.PXCM-06/SDH.SW.MSA.CD41A.PWR REAL 3 P
	426463fd-b0dd-58dd-b85d-c09c14757114				inches		/92028/0500rain
	b9b0856a-65b7-535a-84ed-0daa05aae6b5				Fahrenheit		/91801/0500dew-point

Figure 2: Sample of Description Table

- **Gathering Data**

In the gathering data stage, the team focused on getting a reasonable amount of data from the SDB database, and checking the possibility of building a model upon it. This part contains a few tasks including data extraction, data schema designing, data cleaning, and data uploading.

i. Universally Unique Identifier (UUID) and Description Tables Design Options

From defining problem stage, we have built a description table for data. There is one column of the data which has the tag of 'uuid'. The data in this column is unique and would never be blank for any data source. This 'uuid' is also used in the query to extract data from the database.Since the data would be pushed to the HANA system, the schema of the data tables in HANA need to be designed prior to uploading. There are many ways to design the data table. Since there are two sub-teams doing modeling in our project, it is reasonable to build two description tables in the HANA database with 'uuid' being the primary key. In the meantime, each table would only contain columns that are meaningful to the specific team. It is also reasonable to load the original description table to the database. This method would requires less coding but more query time comparing to the previous method.

ii. Data and Data Table Design Options

The extracted data are all in the same format: each row contains a UNIX time stamp and a reading. Thus, there are also two reasonable ways for the data schema. One is to put the data into multiple two-columns data tables with the table names being the 'uuid' value from the description table. The other one is to make one three-columns data table with 'uuid' values being the extra element in the row. The two-columns data table requires less storage space, but the users would need to search through table names to get the data. The three-columns data table requires more space, introduces more redundancy, and potentially increases the query time, but this design simplifies the building of a query, for the 'uuid' value in the description table can also act as the foreign key to the data table.

iii. Final Decision

After consulting the SAP engineers, we learned that the database in HANA system is column-oriented, which means a large table with many blanks will be shrank in size while having little impact on query time in HANA. The SAP engineers also guaranteed on the processing power of their platform, and encouraged us to take the three-columns approach. Considering the resources the team has, I build the schema with loading full description table and building three-columns data table approach. The sample of the data table can be found in figure 3 below. After designing the schema, the other sub teams requires data to be cleaned to fit the schema. The amount of the data needed is discussed in the next stage.

1357057768000	49	336a508f-d20b-52c2-b858-39d21dce91bd
1357058668000	49	336a508f-d20b-52c2-b858-39d21dce91bd
1357059568000	49	336a508f-d20b-52c2-b858-39d21dce91bd
1357060468000	49	336a508f-d20b-52c2-b858-39d21dce91bd
1357061368000	49	336a508f-d20b-52c2-b858-39d21dce91bd
1357062268000	49	336a508f-d20b-52c2-b858-39d21dce91bd
1357063168000	49	336a508f-d20b-52c2-b858-39d21dce91bd
1357064068000	49	336a508f-d20b-52c2-b858-39d21dce91bd
1357064968000	54	336a508f-d20b-52c2-b858-39d21dce91bd
1357065868000	54	336a508f-d20b-52c2-b858-39d21dce91bd
1357066768000	54	336a508f-d20b-52c2-b858-39d21dce91bd
1357067668000	54	336a508f-d20b-52c2-b858-39d21dce91bd
1357068568000	54	336a508f-d20b-52c2-b858-39d21dce91bd
1357069468000	54	336a508f-d20b-52c2-b858-39d21dce91bd
1357070344000	54	336a508f-d20b-52c2-b858-39d21dce91bd
1357071268000	54	336a508f-d20b-52c2-b858-39d21dce91bd

Figure 3: Sample of Three-columns Data Table

- **Building Model**

The building model stage relies heavily on other two sub teams. In this stage, my tasks is to consult on the time span of data samples for modeling, and to provide technical support such as fixing MATLAB bugs. I also make suggestions on the feature selections for support vector machine based on my knowledge of the data. The team finally decided to look into two years of data. A possible way is to use one year's data as training data for the model, and the other year's data as the validation data. This approach works for the irrigation scheduling team, but not the malfunction team, since malfunction is binary and we cannot obtain logs of failure in each building. The other contribution of mine is translating the MATLAB code from the other two sub-teams into Python and the query language that the SAP HANA system accepts. My technical contribution in this stage is mostly theoretical, and the actual coding portion is mainly for the data cleaning, debugging, and translating purposes.

In the data cleaning part, I helped the team to take the average of weather data in an hour, and use the data for forecasting. The cleaning and forecasting is done on HANA. The cleaning part utilize the time-stamp in the data table. I divided the time-stamp by 3,600,000, which is the amount of milliseconds in an hour, then round the result and group the data by the rounded number, and take the average over each group. I use the time-stamp of the beginning of the particular hour as the time value, and put the averaged weather data value accordingly. Usually the two years of data for one type of weather, for example, humidity, is about 5.4 GB. HANA finished the above process in few seconds and the resulting table is on the order of KB level.

The forecasting part also utilize the HANA's library. The team uses exponential smoothing for forecasting. The figures below showed the team has tried single exponential smoothing, double exponential smoothing, and triple exponential smoothing. The red-line indicates the actual humidity value from our data, and the blue line is the smoothed and predicted value. The actual value contains 400 data points, and each smoothing method takes in first 300 points and predicts the value across all 400 point of time. Although single and double exponential smoothing captures the trend in the given 300 points, they cannot predict well in the last 100 points. The single exponential smooth gives NaN value in the prediction part, while the double exponential smoothing gives only a straight line. Thus, the team decided to use the triple exponential smoothing. The graph and data is generated through Matlab, which is only for testing.

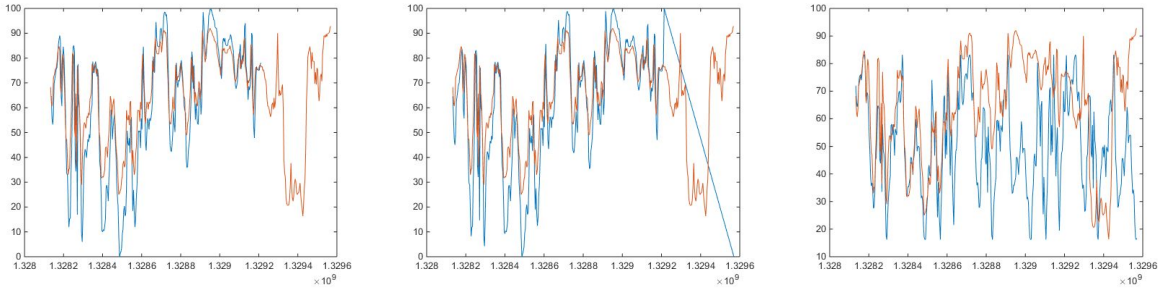

Figure 4: Sample Humidity Data with Single/Double/Triple Exponential Smoothing

The optimization of irrigation scheduling is also done on HANA's optimization modules. The output of this part is an advised schedule for irrigation. The output will be parsed and plotted in

our dashboard in the next part. The optimization is linear programming problem, and below is the description of the particular problem cited from the work of the team member Dylan Rhode.

- **Decision Variables**
- $x_d = amount\ to\ water\ on\ day\ d$
- **Constants**
- $e_d = expected\ evapotranspiration\ on\ day\ d$
- $S = number\ of\ days\ water\ lasts\ in\ soil$
- $T = number\ of\ days\ to\ schedule$
- **Linear Program**
- $minimize\ \sum_d^T e_d * x_d$
- $subject\ to \quad \sum_{i=d}^{d-S} x_d - e_d \geq 0\ \forall\ d = S+1, \ldots, T$
  $x_d, e_d \geq 0$

Figure 5: Linear Programming Model of Irrigation Schedule

- **Demonstrating Results**

The last stage is demonstrating results, which requires a visual way of presenting the outcomes of the other two sub teams. There are also several methods in designing the visual presentation. The simplest one would be creating a static web page showing the resulting data of predicted schedule and possible malfunction. The figure below shows the resulting page of a particular query for the irrigation schedule on the dashboard.

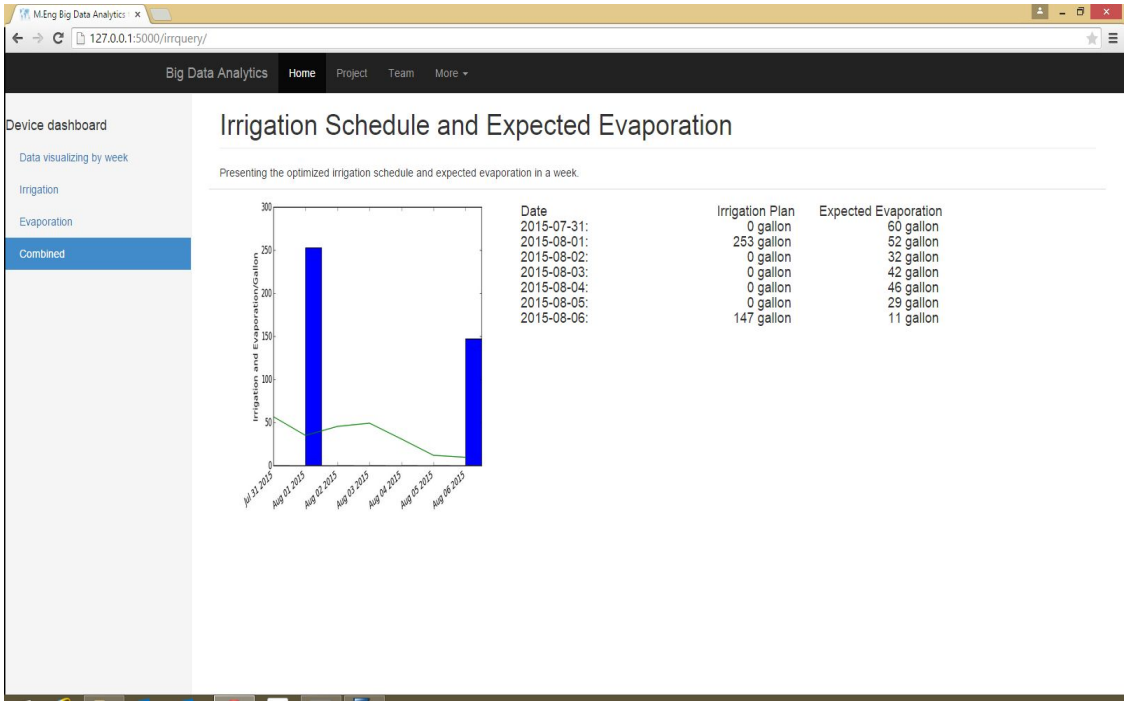

Figure 6: Result of Irrigation Plan in the Week of 07/31/2015

d) Results and Discussion

The final results of my part would be a dashboard on the web app demonstrating the results of the whole project team. The dashboard could be a prototype of a control panel in the future. For example, the irrigation scheduling team generates the prediction of irrigation schedule of future. The ideal situation would be using the output data to control the irrigation in the campus, and later achieving a decreasing in water usage while the system maintains the campus as before. However, due to the limitation of accessing school equipment, the team can only provide a graph of the predicted schedule. The graph will be demonstrated on the dashboard, which may be used for actual control later. The malfunction team would have a dashboard with the notification of possible malfunction. This can also be further developed into a notification system for the maintenance team on campus. If the data can be acquired in real time and be processed, the dashboard could warn members in the maintenance team for a possible malfunction in the building. Also, interface of this dashboard could be generalized such that any campus that has similar sensors could adapt this system. The dashboard is only a demonstration of project now, but it can be extended to save water and power, and detect real-time malfunction in the future.

There are many other results in other stages described in the section above. The transferring from metadata to human-readable table enables teammates to quickly compare between data sources, and simplifies the process of aggregating necessary data sources. The design of data schema fully utilizes the SAP HANA platform's computing power, and makes teammates' query building more efficient. These results in the intermediate stages pushed the project forward and also made some side discoveries.

Some of the discoveries may help other groups in their researches. For example, there are many corrupted or empty data sources in SDB team's database. Since our team has scrapped most of the data sources, we kept a log of the problematic data sources along the scrapping. The log could be helpful to the SDB team to manage their database better. Another example is the discovery of a bug in SAP's data loader. When I upload the description table to the HANA system, the system always encounters an error at a little bit over a thousand rows. It turns out the loader would first sample a few hundred rows, and automatically generate a table with the data types detected in these rows and the length of value from the maximum length in each column detected. This scheme does not work for our description data table, since the values are in different length. This bug report can also help SAP build a better data loader to scan the whole table first, then build a table accordingly. Another solution would be to incorporated this issue in their error message, so the people who encounter the same problem in the future can discover the solution more efficiently.

Overall, my results greatly correspond to the success of the project. The final results present the project outcome in a visual way. In the other stages of the project, my work helped other teammates in optimizing computing resources, extracting and managing data, and writing and debugging code. My results also generate some side discoveries that may prove to be helpful to other groups related to our project.

5. Concluding Reflections

As mentioned in the results and discussion section in technical contribution part, my outcome is the dashboard and the code for the modeling. In the original plan, the team aims at building a system that can access the equipment control panel and apply our model in the campus. Due to the limitation, we cannot access the control panel, so we switched to only demonstrating our outcome and model. Another issue is the security check from SAP. Our dashboard need to access the HANA system, and the HANA system can only be accessed using SAP's laptop. Thus, our dashboard is not pushed online, but pushed to the local host in the SAP's laptop. This is an authorization issue, which would give the team more flexibility if it is well taken care of.

This project requires making both the school and industry ends meet. The team spent a lot of time managing the conversation with other research teams in the university and the engineering team in SAP. It usually takes a week to solve a technical issue if it is related to other party, which was not expected at the beginning. Thus, a graduate project with the industry would require subtle project management and time control.

There are still many parts in our project that can be further improved. The future team can try to solve the authorization issue with SAP, and make an online dashboard so the other user can access it through the internet. They can also gain access to the school equipment and maintenance log to further verify and improve our model. Our team has not fully use the computing power that SAP HANA system can provide, so the future team can also transfer more computation to their system, and build a real-time event processing unit upon our dashboard. The future team could also recruit more student from computer science department to help with technical issue and system building.

This project provides me with a chance to work with people in different disciplines, and I truly value this experience. The process of learning other major's language and understanding their needs has enabled me to think in many different ways. I hope the graduate students would also have the opportunity to participate in a project involves people with other background in the future.

6. Works Cited

Advanced Technology Services, Inc. "Sharpen Your Competitive Edge." 2015. Advanced Tech. February 16 2015 <http://www.advancedtech.com/services>.

Diment, Dmitry. "IBISWorld Industry Report 51821: Data Processing & Hosting Services in the US." 2015.

"Django Documentation" Django. Django Software Foundation, n.d. Web. 14 Apr. 2015. <https://docs.djangoproject.com/en/1.8/>.

"Energy Efficiency." 2015. Energy.gov. 2 March 2015 <http://www.energy.gov>.

"EPA's Strategies to Meet Its Federal Requirements." 5 November 2012. U.S. Environmental Protection Agency. 2 March 2015 <http://www.epa.gov/greeningepa/water/strategies.htm>.

Harris, Zachary. "IBISWorld Industry Report 81131: Machinery Maintenance & Heavy Equipment Repair Services in the US." 2014. February 16 2015.

"IBISWorld Business Environment Profiles Price of Computers and Peripheral Equipment." 2015.

Irrigate-IQ Precision Irrigation Solution. 2015. 16 February 2015 <http://www.trimble.com/agriculture/irrigate-iq>.

John Deere Field Connect. 2015. 16 February 2015 <https://www.deere.com/en_US/products/equipment/ag_management_solutions/field_and_crop_solutions/john_deere_field_connect/john_deere_field_connect.page>.

"LoCal." LoCal (A Network Architecture for Localized Electrical Energy Reduction, Generation and Sharing). University of California, Berkeley, n.d. Web. 14 Apr. 2015. <http://local.cs.berkeley.edu/wiki2/index.php/Main_Page>.

Manyika, J., et al. "Big Data: The Next Frontier for Innovation, Competition, and Productivity." McKinsey Global Institude Industry Report. 2011.

Mataloni, Lisa, Kate Shoemaker and Jeannine Aversa. "Gross Domestic Product: First Quarter 2014 (Third Estimate)." 25 June 2014. Bureau of Economic Analysis. 2 March 2015 <http://www.bea.gov/newsreleases/national/gdp/2014/pdf/gdp1q14_3rd.pdf>.

Neville, Antal. "IBISWorld Industry Report OD4422: Precision Agriculture Systems & Services." 2014.

"SAP HANA" SAP HANA. SAP SE, n.d. Web. 14 Apr. 2015. <http://hana.sap.com/abouthana.html>.

"SDB" SDB - Software Defined Buildings, UC Berkeley. University of California, Berkeley, n.d. Web. 14 Apr. 2015. <http://sdb.cs.berkeley.edu/sdb/>.

"SMAP 2.0 Documentation" SMAP: The Simple Measurement and Actuation Profile —. University of California Regents, n.d. Web. 14 Apr. 2015. <http://www.cs.berkeley.edu/~stevedh/smap2/>.

Sotto, L.J. and A.P. Simpson. "Data Protection & Privacy." 2014. Hunton & Williams LLP. 2 March 2015.

"Stack Overflow" Stack Overflow. Stack Exchange Inc, n.d. Web. 15 Apr. 2015. <http://stackoverflow.com/>.

U.S. Environmental Protection Agency. Demographics. 14 April 2013. 2 March 2015 <http://www.epa.gov/agriculture/ag101/demographics.html>.

United States Department of Agriculture. FAQs. 10 February 2015. 2 March 2015 <http://www.ers.usda.gov/faqs.aspx#10>. "Land in Farms, Harvested Cropland, and Irrigated Land, by Size of Farm: 2012 and 2007." 2012.

United States Government. Economic Report of the President. Washington: United States Government Printing Office, 2014.

Watson Scott, Jamini Samantaray, John Scoville, and James Moyne. Big Data Analytics System. U.S.A: Patent WO2014005073A1. 3 Jan 2014.

Wood Group. "Integrity Management." 2015. Wood Group. February 16 2015 <http://www.woodgroup.com/products-services/view-by-product-service/specialist-services/integrity-management/pages/default.aspx>.