Spring 2018

Big Data Quality Modeling And Validation

Khushali Yashodhar Desai
San Jose State University

BIG DATA QUALITY MODELING AND VALIDATION

A Thesis

Presented to

The Faculty of the Department of Computer Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Khushali Desai

May 2018

The Designated Thesis Committee Approves the Thesis Titled

BIG DATA QUALITY MODELING AND VALIDATION

by

Khushali Desai

APPROVED FOR THE DEPARTMENT OF COMPUTER ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2018

Dr. Jerry Gao	Department of Computer Engineering (Thesis Chair)
Dr. Jacob Tsao	Department of Industrial and Systems Engineering
Dr. Chandrasekar Vuppalapati	Department of Computer Engineering

ABSTRACT

BIG DATA QUALITY MODELING AND VALIDATION

by Khushali Desai

The chief purpose of this study is to characterize various big data quality models and to validate each with an example. As the volume of data is increasing at an exponential speed in the era of broadband Internet, the success of a product or decision largely depends upon selecting the highest quality raw materials, or data, to be used in production. However, working with data in high volumes, fast velocities, and various formats can be fraught with problems. Therefore, software industries need a quality check, especially for data being generated by either software or a sensor. This study explores various big data quality parameters and their definitions, and proposes a quality model for each parameter. By using data from the Water Quality U. S. Geological Survey (USGS), San Francisco Bay, an example for each of the proposed big data quality models is given. To calculate composite data quality, prevalent methods such as Monte Carlo and neural networks were used. This thesis proposes eight big data quality parameters in total. Six out of eight of those models were coded and made into a final year project by a group of Master's degree students at SJSU. A case study is carried out using linear regression analysis, and all the big data quality parameters are validated with positive results.

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Dr. Jerry Gao, for his tremendous support, always helping and guiding me whenever I was stuck. I would also like to thank San Jose State University for giving me the opportunity to do a thesis as part of my master's work. I am grateful for my father, mother and sister for proofreading my work and giving me input, a valuable perspective from people who do not belong to this field. Without them, my work would not be as fruitful. And special thanks to my thesis committee members, who always offered moral support, advice and technical reviews of the material for both this thesis and the original work. I would also like to mention Ariel Andrew and Jenn Hambly from the English writing center of SJSU for giving me essential input on my grammar mistakes. My friends were also my strength; I am indebted to Jayapriya, Chaitra, Sumana, Pranathi, and Nithya for being there with me during my research endeavor, and giving me moral support all the time. Finally, a huge thank you to David McCormick for copy editing the thesis to help reach the final version.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1
Introduction

Research Motivation

Due to advancements in technology like cloud computing, internet of things, social networking devices and more, use of mobile-applications is now generating greater quantities of data than ever before. According to the technology research firm Gartner, there will be 25 billion network-connected devices by 2020 (Vass, 2016). However, due to the huge volume of data generated, the high velocity with which new data are arriving, and the large variety of heterogeneous data, the current quality of data is far from perfect ("IDC Forecast," 2013). It is estimated that erroneous data cost US businesses about 600 billion dollars annually (Eckerson, 2012, pp. 1-36). At present, there is no standard method to measure the quality of data, so fully reliable benchmarks still need to be set. Therefore, there is a great need to address big data quality assurance, which can be defined as "the study and application of various assurance processes, methods, standards, criteria, and systems to ensure the quality of big data in terms of a set of quality parameters" (Gao, Xie, & Tao, 2016, pp. 433-441).

The following are the challenges and needs in big data quality assurance and validation (Gao et al., 2016, pp. 433-441):

1. Awareness of the importance of big data quality assurance needs to be raised.
2. There is a need for well-defined quality assurance standards.
3. Research needs to be done on big data quality models.

To address these needs, it is necessary to develop well-defined big data quality assurance and validation standards. To this end, appropriate big data quality assurance

programs need to be structured, and big data quality models must be defined and developed.

Why Big Data Quality Assurance?

One implicatdion of poor quality data is missed business. As pointed out in Cai and Zhu (2015), poor data quality could cause many tangible and intangible losses for businesses. The estimated costs could go as high as 8% to 12% of revenues for a typical organization and may generate about 40% to 60% of the service organization's expenses (Wigan & Clarke, 2013). Clearly, poor data may hinder revenue goals. They can also cause communication mistakes, which could result in dissatisfied customers (Gao et al., 2016, pp. 433-441).

Another negative effect of low quality data is greater consumption of resources. However, as organizations often do not know why data quality is important, 65% of businesses wait until there are problems with data before seeking solutions. In this way, they waste significant amounts of labor and time (Gao et al., 2016, pp. 433-441).

Lastly, poor service based on faulty data leads to poor decision-making and hence, low quality products. As a result, service will not be up to expected quality standards, so all the hard work, time, and labor invested may be of little to no value (Gao et al., 2016, pp. 433-441).

What Are Big Data Quality Issues?

The 5 Vs of big data (variety, volume, value, velocity, and veracity), although important, also lead to problems in measuring big data quality. As the volume of data is

high, it is challenging to maintain data quality in a given amount of time. It is also difficult to integrate data because of the multiple formats of data present.

Enterprise management for big data. Different organizations have varying needs for data, so they all require their own data processing techniques. They also need to have their own methods for big data management and quality assurance. Poor management in any of these areas will result in substandard data quality.

Big data processing and service. This includes factors like data collection, data conversion, data service scalability, and data transformation (Gao et al., 2016, pp. 433-441). Due to its inherently high volume, big data presents challenges in terms of collection, transformation, and conversion. Ultimately, this leads to poor quality data organization.

Chapter 2 presents a literature survey to cover the existing definitions, models, and methodologies adopted by various industries and institutions for big data quality. The third chapter describes key big data quality parameters, providing models and examples. The fourth chapter presents a case study. The concluding chapter provides suggestions for future work.

This thesis aims to model eight big data parameters to measure quality. With the help of either Monte Carlo or neural networks, composite data quality can be predicted. The aim of presenting various models is to improve the quality of big data and make better business decisions to make a business successful.

Chapter 2
Literature Review

With the emergence of big data and sensor networks, much attention has been placed on sensor data quality. This section outlines the current state of the art and explores any scope for improvement or innovation.

There have been many studies on the overall data quality parameters of big data (Askham, et al., 2013; Gao et al., 2016, pp. 433-441; Woodall, Gao, Parlikad, & Koronios, 2015, pp. 321-334). Laranjeiro, Soydemir, and Bernardino (2015), as well as Clarke (2014) and Loshin (2010), have noted different big data quality parameters and definitions. This thesis presents new models based on those definitions, such that they can be applied universally to big data. Cai and Zhu (2015) describe scorecard approaches that can be used to measure big data quality. Moreover, organizations have come up with their definitions, models or techniques to measure or predict quality.

Studies regarding data quality (e.g., Cai & Zhu, 2015) have been carried out since the 1950s. Industry experts have proposed many definitions and parameters for data quality. A group from MIT, Total Data Quality Management, has done major research in the field. They surveyed and identified four main categories that contain about fifteen data quality parameters.

A paper by Gao et al. (2016, pp. 433-441) presents useful ideas regarding big data quality assurance, including related challenges and needs. It addresses the extent to which big data quality is the same as that of normal data, ways to validate big data quality, and other key factors. It defines quality parameters such as accuracy, currency, timeliness,

correctness, consistency, usability, completeness, accessibility, accountability, and scalability. It also describes the big data quality validation process and proposes a comprehensive study of factors which cause problems with big data quality. This study by Gao et al. (2016, pp. 433-441) provides essential background knowledge required for this thesis. It also outlines available big data quality validation tools and major players.

Big Data Validation Process

The five main big data services in the big data validation process are (1) data collection, (2) data cleaning, (3) data transformation, (4) data loading, and (5) data analysis.

Data collection is the process of accumulating data and calculating various information on important variables, which improves understanding of data, resulting in better decision making. Data cleaning is, as its name suggests, the process of finding corrupt or inaccurate data and correcting them. Data transformation converts the format of the data from the source data system to the format of the destination's data system. Data loading is a process in which data are loaded into large data repositories. Depending on the requirements of the organization, this process varies widely. Data analysis refers to process of doing all the previously discussed big data services such as collecting, cleaning, transforming and loading with the primary intent of making better decisions and knowing more about the data itself. Data aggregation refers to the gathering of information from databases with the goal of preparing combined data sets for processing.

Big Data Quality Validation Tools

MS-Excel software, part of the Microsoft office package, is a data cleansing and validation tool. One can use it to rearrange and reformat data for analysis. One can also use it to generate charts and graphs that can illustrate the data well. It can support CSV, XLSX, and other data formats. However, despite performing well with small amounts of data, Excel cannot handle big data.

Zoho Reports is an online reporting and business intelligence service. It is a big data and analytics solution that allows users to create insightful reports and dashboards. It is a SaaS platform tool which is very easy to use. This thesis uses Zoho Reports to apply filters on the data set obtained to show an example of a created data model.

DataCleaner is an open source tool for data quality, data warehousing, data profiling, master data management, business intelligence, and corporate performance management. It is compatible with multiple platforms like Windows, Linux and IOS platforms. Its focus area is Apache Hive and Apache HBase connectivity. It can support data from TXT files, CSV and TSV files, as well as relational database tables, MS Excel sheets, MongoDB, and Couch DB. Major features of DataCleaner are as follows:

- It has a duplicate detection feature based on machine learning principles.
- It can easily check the integrity between multiple tables in a single step.
- It profiles and analyzes the database within minutes. However, it is slower compared to other big data validation tools.
- It serves as an efficient and scheduled data health monitor.

QuerySurge is a big data, ETL and data warehouse testing tool. It finds corrupt data

and provides insight into data's health.

Splunk is the leading tool for operational intelligence. Clients use this tool to monitor, search, analyze and visualize data. It can generate graphs, visualizations, reports, and create dashboards. Splunk is easy to use and works on both unstructured and structured data. It is available as both a software and cloud service.

Talend is a primer open source data validation tool. It consists of different modules such as big data integration, cloud integration and application integration. It runs in Hadoop and Spark. It supports multiple operating systems, including Windows, Linux, and Mac OS. It imports data from relational databases, NO SQL, and from CSV files. It also performs multiple data quality checks and generates graphs by analyzing certain criteria.

Tableau is a leading business intelligence and analytics tool. It can connect to various data sources like CSV files, Cloudera Hadoop, MySQL, and Google analytics. It has features to validate data type, conformity, and range checks. Data filters can be applied and customers can write their own filters as well. It is easy to use, and the facility of charts and graphs allows for clear analysis of data.

Pentaho is a platform for big data integration and business analytics. It consists of many tools such as data integration, embedded analytics, business analytics, cloud business analytics, Internet of things analytics, etc. Its data integration product delivers accurate data to customers from any data source. Pentaho has a parallel processing engine that gives high performance and scalability. It provides integrated debuggers for testing and job execution. It has a built-in library which has components that are used for data

transformation and validation.

Big Data Quality Process and Framework

Quality assessment process for big data. To perform quality assessment of big data, proper methodology should be followed. Cai and Zhu (2015) provide one such mechanism. This model (shown in Figure 1) specifies the goal of data collection and defines the parameters. Based on these parameters, the final step is to select various assessment indicators, all of which will require their own tools and techniques.

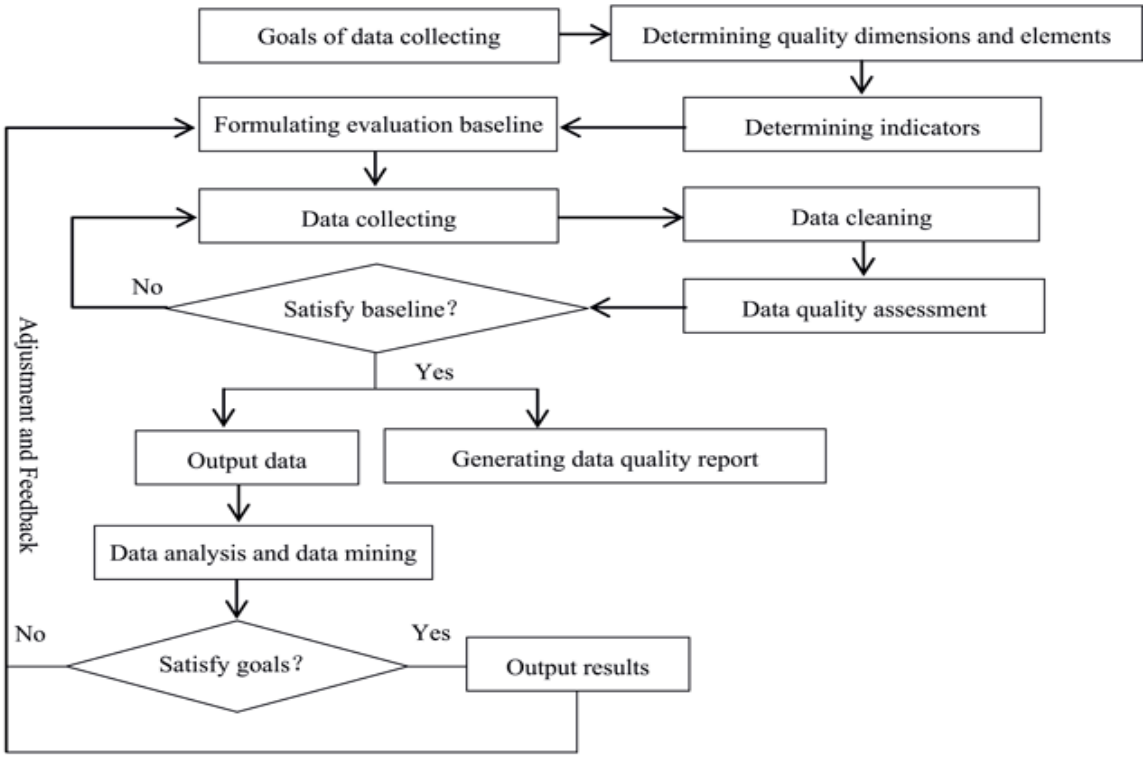


Figure 1. Quality assessment process for big data (Cai & Zhu, 2015).

After gathering all the required information for data assessment, data are collected and cleaned. Then, data quality assessment is carried out by comparing results with the

baseline of the initial goals. Based on the results, either a quality report is generated or the whole process from "formulating evaluation baseline" is repeated.

Data quality framework. Gudivada et al. (2016, p. 33) propose the data quality framework (DQF) shown in Figure 2.

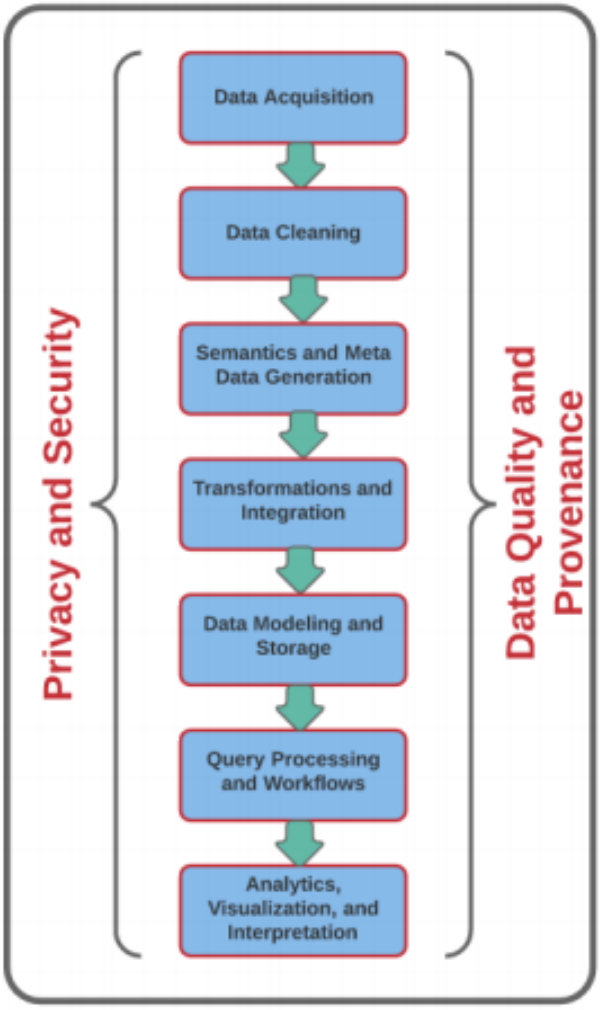


Figure 2. Data Quality Framework (Gudivada et al., 2016, p 33).

In the workflow presented, the process starts with data acquisition and is followed by data cleaning. In the third phase, semantics and meta data are generated. Here, unstructured data, like images, graphics, audio, video, and tweets are turned into semi or structured data. In the subsequent phases of data transformation and integration, data modeling, query processing, analytics, and visualization take place.

After comparing models by Cai and Zhu (2015) and Gudivada et al., (2016, p. 33), one can see that most of the phases are the same. What differs is the timeframe. In Cai and Zhu (2015), data gathering occurs at a much later stage. Whereas in Gudivada et al. (2016, p. 33), data gathering is the first step. Cai and Zhu (2015) emphasize the importance of making useful decisions to maintain quality assurance in the early stage. The current state of the art lacks big data quality models that can be applied based on parameters.

In summary, this thesis presents eight big data quality parameter models, all of which are based on clear definitions (Askham et al., 2013; Cai & Zhu, 2015; Gao et al. 2016, pp. 433-441; and Sharma, Golubchik, & Govindan, 2010). In addition, these models are modified to be suitable for use with big data. As such, they may become the starting point for generating protocols for big data quality standards.

Chapter 3
Big Data Quality Model and Evaluation

Big Data Quality Parameters

Big data quality assurance is carried out to assess the quality of data to ensure they are of high quality. According to Ludo (2013), data are of high quality if they are fit for their intended uses in operation, decision making, and planning. High-quality data are accurate, available, complete, consistent, credible, processable, relevant and timely. From the definition given above for high quality data, this thesis relies on eight quality parameters (Figure 3) that will be used to check quality standards for big data:

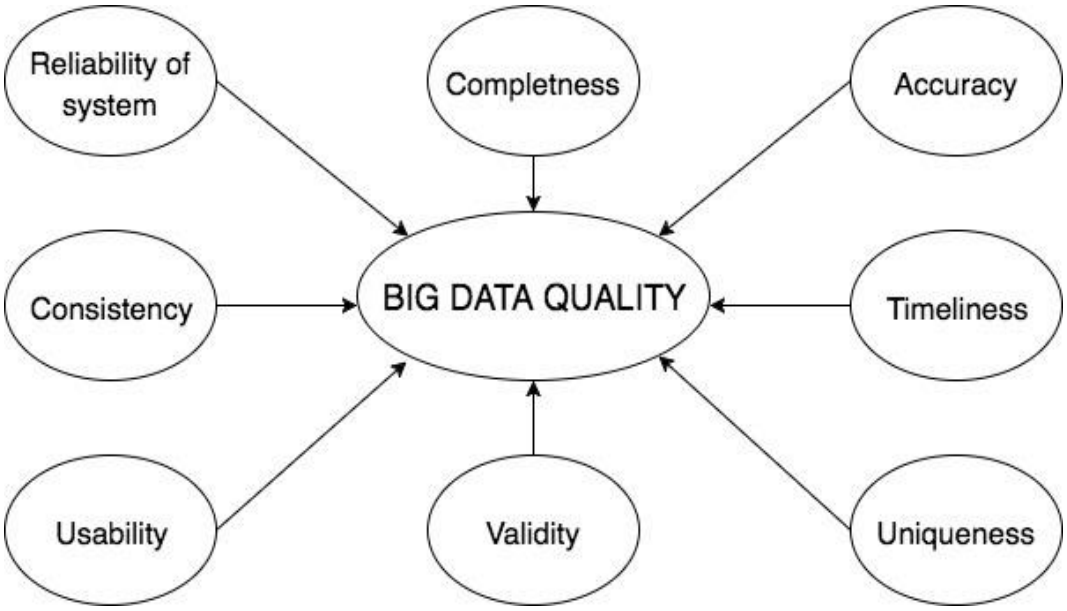


Figure 3. Big data quality parameters.

- Completeness: Are all the required values available in the dataset?
- Accuracy: Are data accurately describing events or objects?
- Timeliness: Do data arrive at the anticipated time?

- Uniqueness: Is there any redundancy in the data set?
- Validity: Do data follow specific rules?
- Consistency: Are there any contradictions in the data?
- Reliability of gauge/sensor: Is the state of machine gathering data reliable?
- Usability: Do data correspond to the given needs?

Big data completeness is a measure of the amount of data available against the desired amount for its intended purpose. Completeness is used to verify if deficiencies in the data will impact their usability. Big data completeness can be defined as the proportion of stored data against the potential of 100% complete data (Askham et al., 2013). For measuring completeness, this thesis takes the number of available values in the given data set and calculates its ratio against the total anticipated number of values. The unit of measure is percentage.

Big data accuracy can be defined as the degree to which data correctly describe the "real world" object or event being taken into consideration (Askham et al., 2013). To measure the accuracy of the data set or data item, data are compared with "real world" truths. It is common to use third party reference data, which are generally deemed trustworthy and of the same kind (Askham et al., 2013). The unit of measure is percentage of data entries that meet data accuracy requirements. In some cases, accuracy is easy to measure, for instance, distinguishing gender (i.e., male or female). Other cases might not be so clearly differentiated, making accuracy more difficult to measure. Accuracy helps to answer questions like whether the provided data are accurate, if they are causing ambiguity, and if they reflect the real state of the source of the data.

Big data timeliness is an important factor for big data quality assessment, as data change every second. Big data timeliness is measured by the degree of data which represents reality at the required point of time (Askham et al., 2013). To measure timeliness, one marks the time difference between when an event occurs and when it is recorded. In other words, this is the difference between when time data are expected and when they are readily available for use. The unit of measure is percentage of time difference. Timeliness helps determine whether data have arrived on time and whether data updates are regularly made.

Big data uniqueness is defined as the measurement of a data item against itself or its counterpart in another data set or database (Askham et al., 2013). The unit of measure is percentage. This parameter is used to confirm that a data set does not have duplicate values. In big data, checking this factor helps eliminate redundancies.

Big data validity is also known as data correctness. Data are valid if they conform to the syntax (format, type, and range) of their definitions (Askham et al., 2013). To measure validity, one compares the data with valid rules defined for them. The unit of measure is percentage. It helps to know whether data is valid for their intended use or not. This thesis models the validity at the transaction and parameter levels.

Big data consistency refers to the extent to which the logical relationship between correlated data is correct and complete (Cai & Zhu, 2015). Askham et al. (2013) define consistency as the absence of difference when comparing two or more representations of the same thing. To measure consistency, one measures a data item against itself or its counterpart in another data set (Askham et al., 2013). Suppose the same data arrive at two

different stations by coming from multiple paths and accumulating at a base station. To have consistency, both data sets should have the same value and the same meaning. For this reason, it is necessary to check the consistency between them. This thesis models the value and time consistency of data.

Big data reliability of the system is defined as the ability of the network to ensure reliable data transmission in a state of continuous change of network structure (Lavanya & Prakasm, 2014). To measure the reliability of system, one characterizes whether a component or system is properly working according to its specifications during a particular time. Sensors are checked to determine whether they are reliable.

Big data usability can be defined as whether the data are useful and meet users' needs (Askham et al., 2013). To measure usability, one calculates timeliness, accuracy, and completeness, as the value of this three-quality parameter defines whether data are usable or not. The unit of measure is percentage.

Referent Data Sets for Big Data Quality Models

To define big data quality models, two data sets (expected and received) are utilized as referents to help gauge big data quality parameters. Let S represent the k stations in the network such that $S = \{S_1, S_2 \ldots\ldots S_k\}$, where S_i presents the i^{th} sensor in the station. Suppose at sensor S_i, one expects the data set to arrive with m number of transactions, and each transaction consists of n number of parameters. Additionally, sensor S_i receives the data set with mr number of transactions, and each transaction has nr number of parameters. Let E be the expected data set, where m represents the total expected

transactions, and n is the total expected parameters for each transaction. Matrix E = {E_{11}, E_{12} ….. E_{mn}} can be given as follows:

$$\begin{bmatrix} E_{11} & E_{12} & \cdots & \cdots & E_{1n} \\ E_{21} & E_{22} & \cdots & \cdots & E_{2n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ E_{m1} & E_{m2} & \cdots & \cdots & E_{mn} \end{bmatrix},$$

where E_{ij} represents the value for the i^{th} transaction and j^{th} parameter.

Let R represent the received data set, where mr is the number of received transactions and nr is the total number of received parameters per transaction. Matrix R = {R_{11}, R_{12} ….. R_{mrnr}} can be expressed as follows,

$$\begin{bmatrix} R_{11} & R_{12} & \cdots & \cdots & R_{1nr} \\ R_{21} & R_{22} & \cdots & \cdots & R_{2nr} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ R_{mr1} & R_{mr2} & \cdots & \cdots & R_{mrnr} \end{bmatrix},$$

where R_{ij} represents the value for the i^{th} transaction and j^{th} parameter.

To measure data quality parameters, the total number of values for expected and received data sets must be calculated. Let E_{total} be the total number of expected elements with m transactions, and each transaction has n parameters. Hence, E_{total} can be determined as the following:

$$E_{total} = m \times n. \quad (1)$$

Let R_{total} be the total number of received elements with mr transactions, where each transaction has nr parameters. Hence, R_{total} can be given by the following equation:

$$R_{total} = mr \times nr. \quad (2)$$

With each parameter defined in this thesis, one example is also given to validate the models. The data set used to give an example is "Water Quality U. S. Geological Survey (USGS), San Francisco Bay" (Cloern & Schraga, 2016). To demonstrate a use of the model, manual calculation was carried out after defining each quality parameter. At various stages, it was required to make different filters and assumptions to show the example. Such filters and assumptions are mentioned separately at the start of each model example.

Various time measurements as transaction timestamps, the number of transactions per day, and the intervals between transactions are considered. Such measurements make it easy to calculate per day, per month, and per year values for the different parameters.

Data Sets Observed in Each Example

Data from USGS Measurements of Water Quality (San Francisco Bay, CA) for the duration of 1969-2015 are taken into consideration. The publication date of this data set is 2016, the start date for recording data was 04-10-1969 and the end date was 12-16-2015. The sensors are 2, 3, 4-36, 649, and 657 (Figure 4). Figure 5 depicts the data set, where the number of total rows is 210826, making this a big data set. A validation tool, Zoho Report, is used to apply filters on the data set and view the results.

Figure 4. Map showing all the sensors (Cloern & Schraga, 2016).

SFO_Water

Import Data

Edit Design | Filter | Save | Add | Delete | Column Properties | More | Search

Hide | Apply | Clear | Save (New*) | Delete Rows | Matches either full or any part of the date (E.g., 25 Dec 2008 or Dec 2008 or Dec)

	Date	Station_Number	Depth	Discrete_Chlorophy	Calculated_Chloro	Discrete_Oxygen	Calculat
	Is	Is	Is	Contains	Contains	Contains	Contains
1	04/10/69	4	0.5				
2	04/10/69	4	2				
3	04/10/69	4	4				
4	04/10/69	4	11				
5	04/10/69	5	0.5				
6	04/10/69	5	2				
7	04/10/69	5	4				
8	04/10/69	5	6				
9	04/10/69	6	0.5				
10	04/10/69	6	2				
11	04/10/69	6	4				
12	04/10/69	6	6				
13	04/10/69	7	0.5				
14	04/10/69	7	2				
15	04/10/69	7	4				
16	04/10/69	7	6				

Rows: 210826

Figure 5. Data set without any filters (Zoho Reports tool).

Big Data Completeness Models and Examples

This section presents a model for big data completeness parameter. Models for completeness per transaction and completeness per parameter are given. For completeness per transaction, the model checks what percent of transaction is complete. For completeness per parameter, it checks what percent of data is available for one parameter during all the transactions in the given time span.

Model - completeness per transaction. This section defines big data completeness parameter in terms of transaction. To determine completeness, it is necessary to know how much data is expected to consider a data set as complete. One can find out the total number of expected data using Equation 1 as E_{total}. This section also defines a way to determine the total missing values in big data. M_{total} is the total number of missing data in the received data set R with mr transactions and nr parameters. Data set E is the expected data set. Also, there can be null values in the received data set R, where $Null_{\text{value}}$ is the total number of null values in the received data set. Therefore, M_{total} for the received data set R can be given as follows:

$$M_{\text{total}} = E_{\text{total}} - R_{\text{total}} + Null_{\text{value}}, \tag{3}$$

where E_{total} and R_{total} are derived from Equations 1 and 2, respectively. $Null_{\text{value}}$ is the number of null values in data set R.

In Equation 3, to obtain the total number of missing values, the total number of received values is subtracted from that of the expected values. Finally, null values are added to the total number of missing values. To measure the completeness per

transaction, substitute m = 1 for data set E of Equation 1 and mr = 1 for data set R of Equation 2.

$Completeness_{tran^i}$ is completeness per transaction for data set R and transaction number i. The subscript *tran* represents that completeness is measured in terms of the transaction. The $Completeness_{tran^i}$ can be determined as

$$Completeness_{tran^i} = \frac{(E_{total} - M_{total})}{E_{total}}, \quad (4)$$

where E_{total} and M_{total} are derived from Equations 1 and 3, respectively. In Equation 4, the total number of missing data in data set M_{total} is subtracted from the total expected number of data E_{total}. This whole value gives the actual number of elements available in data set R. Dividing this subtraction by E_{total} gives the completeness ratio.

Equation 5 determines the percentage value of Equation 4. Let $Completeness_{tran^i}\%$ be the percentage of completeness for transaction number i. It can be defined as

$$Completeness_{tran^i}\% = Completeness_{tran^i} \times 100, \quad (5)$$

where the value of $Completeness_{tran^i}$ can be substituted from the Equation 4.

To determine the per day measurement of data quality parameters, it is necessary to determine the total number of transactions per day. The $Transaction_{day^i}$ is the total number of transactions per day i. $Transaction_{day^i}$ is calculated using the time difference between two transactions and total hours of transaction. It can be defined as

$$Transaction_{\text{day}^i} = \frac{Total_{hr}}{Interval_{hr}}, \quad (6)$$

where $Interval_{\text{hr}}$ is the time difference between two transactions, and $Total_{hr}$ is the total hours for which transactions took place during the day.

Let $Completeness_{tran^{day^j}}$ represent average completeness for day j in terms of transaction. It can be determined as

$$Completeness_{tran^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Completeness_{tran^i}\%}{Transaction_{day^j}}, \quad (7)$$

where $Transaction_{day^j}$ are transactions for day j derived from Equation 6, and $Completeness_{tran^i}\%$ is the percentage completeness for transaction i considered from Equation 5. The summation is applied over $Completeness_{tran^i}\%$ for all the values of i which are equal from 1 to $Transaction_{day^j}$. To calculate completeness for all transactions that happened during day j, the summation value is divided by $Transaction_{day^j}$, producing the average transaction completeness for day j.

The $Numberof_{days^j}$ is the number of days in which transactions happened in month j. Let $Completeness_{tran^{month^j}}$ be the average completeness for month j in terms of the transaction. Hence, $Completeness_{tran^{month^j}}$ can be defined by

$$Completeness_{tran^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Completeness_{tran^{day^i}}}{Numberof_{days^j}}, \quad (8)$$

where $Completeness_{tran^{day^i}}$ is derived from Equation 7. The summation is applied over $Completeness_{tran^{day^i}}$ for all the values of i which are equal from 1 to $Numberof_{days^j}$. To calculate completeness for all the transactions that happened for $Numberof_{days^j}$ days in month j, this summation value is divided by $Numberof_{days^j}$ to determine the average transaction completeness per month.

Let $Numberof_{months^j}$ represent the number of months during which transactions happened in year j. Let $Completeness_{tran^{year^j}}$ represent the average completeness for year j in terms of the transaction. Hence, $Completeness_{tran^{year^j}}$ can be obtained by

$$Completeness_{tran^{year^j}} = \frac{\sum_{i=1}^{numberof_{months^j}} Completeness_{tran^{month^i}}}{Numberof_{months^j}}, \quad (9)$$

where $Completeness_{tran^{month^i}}$ is derived from Equation 8. The summation is applied over $Completeness_{tran^{month^i}}$ for all the values of i which are equal from 1 to $Numberof_{months^j}$ to calculate completeness for all the transactions that happened for $Numberof_{months^j}$ months in year j. This summation value, when divided by $Numberof_{months^j}$, yields average transaction completeness per year.

Example - completeness per transaction. To carry out an example, filters are applied to the data set explained in the previous section. Filters are applied as follows: For parameter Date = 12/16/15 and parameter Station_number = 2, the resultant data set based on these filters is depicted in Table 1. Let this data set be called "example data set" throughout all the examples explained in this thesis.

It is assumed that data are collected at two-hour intervals over all 24 hours of the total transaction. Therefore, as per Equation 6, $Transaction_{Day^j}$=24/2=12 transactions.

To calculate $Completeness_{tran^i}$ for the transaction 1 of the resultant data set, first find the values for E_{total}, R_{total}, and M_{total}. For the calculation of E_{total}, as this indicates completeness per transaction, the total number of expected transactions is one.

Hence, m = 1. As there are a total of 17 parameters in each transaction, the total number of expected parameters n = 17. From Equation 1, E_{total}= m × n = 1×17=17 values.

Table 1

Data After Applying Filters

Date	Station_ Number	Depth	Discrete_ Chlorophyll	Calculated_ Chlorophyll	Discrete_ Oxygen	Calculated_ Oxygen	Discrete_ SPM	Calculated_ SPM	Extinction_ Coefficient	Salinity	Temp.
12/16/15	2.0	2.0		5.3		10.1		36	2.61	4.05	10.32
12/16/15	2.0	3.0		5.2		10.1		36		4.1	10.3
12/16/15	2.0	4.0		5.3		10.1		37		4.16	10.29
12/16/15	2.0	5.0		5.1		10.1		38		4.14	10.28
12/16/15	2.0	6.0		5.5		10.1		39		4.14	10.27
12/16/15	2.0	7.0		5.1		10.1		38		4.15	10.27
12/16/15	2.0	8.0		5.4		10.1		38		4.15	10.27
12/16/15	2.0	9.0		5.4		10.1		38		4.17	10.28
12/16/15	2.0	10.0		4.9		10.1		37		4.23	10.28
12/16/15	2.0	11.0		4.7		10.1		35		4.34	10.28
12/16/15	2.0	12.0		3.3		10		33		5.0	10.32

Note: Here, there are five more columns named nitrate, nitrite, ammonium, silicate and phosphate, which have been deleted from the above data set due to space limitations. They are completely null. The blank box represents the null value.

For the calculation of R_{total}, as this equation involves completeness per transaction, the total number of received transaction mr = 1. There are in total 17 parameters received for transaction 1 of the example data set. Hence, the total number of received parameters nr = 17. From Equation 2, R_{total}= mr × nr = 1×17=17 values.

For the calculation of M_{total}, one needs values for E_{total}, R_{total} and $Null_{value}$. From the above calculations, values for E_{total} =17, and R_{total} = 17. For $Null_{value}$, there are eight

parameters that are completely null for transaction 1 of the example data set. These eight parameters are Discrete_Chlorophyll, Discrete_Oxygen, Discrete_SPM, Nitrate, Nitrite, Ammonium, Silicate and Phosphate. Therefore, $Null_{value}$= 8. Now, substitute all the values in Equation 3 as M_{total}= $E_{total} - R_{total} + Null_{value}$= 17-17+8=8 missing values.

To calculate $Completeness_{tran^i}$ for the 1st transaction of example data set, substitute values of E_{total} and M_{total} in Equation 4 as, $Comeleteness_{tran^1} = \frac{(E_{total} - M_{total})}{E_{total}}$ $= \frac{17-8}{17} = 0.52$. To get the percentage value, substitute $Completeness_{tran^1}$ into Equation 5 as $Completeness_{tran^1}\% = 0.52 \times 100 = 52\%$. The solution to Equation 5 is 52%, which means the 1st transaction of example data set is 52% complete.

Likewise, calculations for all the 12 transactions of example data set can be carried out. For the 2nd to the 11th transaction, $Completeness_{tran^i}\%$ is 47%, because these transactions have another parameter, Extinction_Coefficient, as null. The 12th transaction of the example data set is not received, which makes its $Completeness_{tran^{12}}\%$= 0.

Substituting all the values calculated above for Transactions 1 to 12 in Equation 7 can be given as $\sum_{i=1}^{Transaction_{day^{12/16/15}}} Completeness_{tran^i}\%$ = 522. From the assumption made earlier, $Transaction_{day^{12/16/15}}$ =12. Substituting all these values in Equation 7, $Completeness_{tran^{day^{12/16/15}}} = \frac{522}{12}$ = 43.5%. This means the average completeness for all 12 transactions that occurred on date 12/16/15 is 43.5%. The same calculation can be carried out for completeness for the month and year with the help of Equations 8 and 9, respectively.

Model - completeness per parameter. In defining big data, let X be the parameter for which completeness is calculated. Here, received data set R constitutes all the values in parameter X. With the help of M_{total} from Equation 3, calculate the total number of missing data in the received data set R with mr number of the transactions. In accordance with the earlier section, to calculate completeness, it is necessary to know the amount of data expected to consider the received data set as complete. With the help of E_{total} from Equation 1, find out the total number of expected values in the data set.

Let $Completeness_{param^i}$ be completeness per parameter for data set R and parameter i. The subscript $param$ signifies that completeness is measured in terms of parameter. The $Completeness_{param^i}$ can be determined as

$$Completeness_{param^i} = \frac{(E_{total} - M_{total})}{E_{total}}, \quad (10)$$

where E_{total} and M_{total} are derived from Equations 1 and 3, respectively.

Equation 11 determines the percentage value of Equation 10. Let $Completeness_{param^i}\%$ be the percentage completeness for parameter i. It can be defined as

$$Completeness_{param^i}\% = Completeness_{param^i} \times 100, \quad (11)$$

where the value of $Completeness_{param^i}$ can be substituted from Equation (10).

Let $Completeness_{param^{day^j}}$ represent average completeness for day j in terms of the parameter. It can be determined as

$$Completeness_{param^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Completeness_{param^i}\%}{Transaction_{day^j}}, \quad (12)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for Day j derived from Equation 6, and $Completeness_{param^i}\%$ is the percentage completeness for parameter *i* derived from Equation 11. The summation is applied over $Completeness_{param^i}\%$ for all the values of i which are equal from 1 to $Transaction_{\text{day}^j}$ to calculate completeness for all the transactions that happened during Day *j*. This summation value is divided by $Transaction_{\text{day}^j}$ producing the average parameter completeness for Day j.

Let $Completeness_{param^{month^j}}$ be the average completeness for Month *j* in terms of the parameter. Hence, $Completeness_{param^{month^j}}$ can be defined by

$$Completeness_{param^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Completeness_{param^{day^i}}}{Numberof_{days^j}}, \qquad (13)$$

where $Numberof_{days^j}$ is the number of days on which transactions happened in month *j*, and $Completeness_{param^{day^i}}$ is from Equation 12. The summation is applied over $Completeness_{param^{day^i}}$ for all the values of i which are equal from 1 to $Numberof_{days^j}$ to calculate completeness for all the transactions that happened for $Numberof_{days^j}$ days in month *j*. This summation value, when divided by $Numberof_{days^j}$, gives average parameter completeness per month.

Let $Completeness_{param^{year^j}}$ represent the average completeness for year *j* in terms of the parameter. Hence, $Completeness_{param^{year^j}}$ can be given by

$$Completeness_{param^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Completeness_{param^{month^i}}}{Numberof_{months^j}}, \qquad (14)$$

where $Numberof_{months^j}$ represents the number of months during which transactions happened in year j, and $Completeness_{param^{month^i}}$ is from Equation 13. The summation is applied over $Completeness_{param^{month^i}}$ for all the values of i which are equal from 1 to $Numberof_{months^j}$ to calculate completeness for all the transactions that happened for $Numberof_{months^j}$ months in year j. When this summation value is divided by $Numberof_{months^j}$, it produces average parameter completeness per year.

Example - completeness per parameter. The data set explained in the previous section (Model-completeness per transaction) is derived into consideration to carry out an example. To calculate $Completeness_{param^i}$ for parameter "calculatedSPM**"** example a data set, first find the values for E_{total}, R_{total}, and M_{total} in terms of the parameter. For the calculation of E_{total}, as this is completeness per parameter, the total number of expected parameters is one, Therefore n = 1. Since there are 210826 transactions in the data set, the total number of expected parameters m = 210826. From Equation 1,

E_{total}= m × n = 210826×1=210826 values.

To calculate R_{total}, as this is completeness per parameter, the total number of received parameter nr = 1. Since there are 210826 transactions in the data set, the total number of received transaction mr = 210826. From Equation 2,

R_{total}= mr × nr = 210826×1=210826 values.

For the calculation of M_{total}, one needs values for E_{total}, R_{total} and $Null_{value}$. From the above calculations, values for E_{total} =210826 and R_{total} = 210826. For $Null_{value}$, apply the filter in the tool Zoho report as "Is Empty" for parameter Calculated_SPM. Figure 6

depicts this scenario. There are 36175 values, found null for parameter Calculated_SPM. Hence, $Null_{value}$ = 36175. Next substitute all the values in Equation 3 as

$M_{total} = E_{total} - R_{total} + Null_{value}$ = 210826-210826+36175 = 36175 missing values.

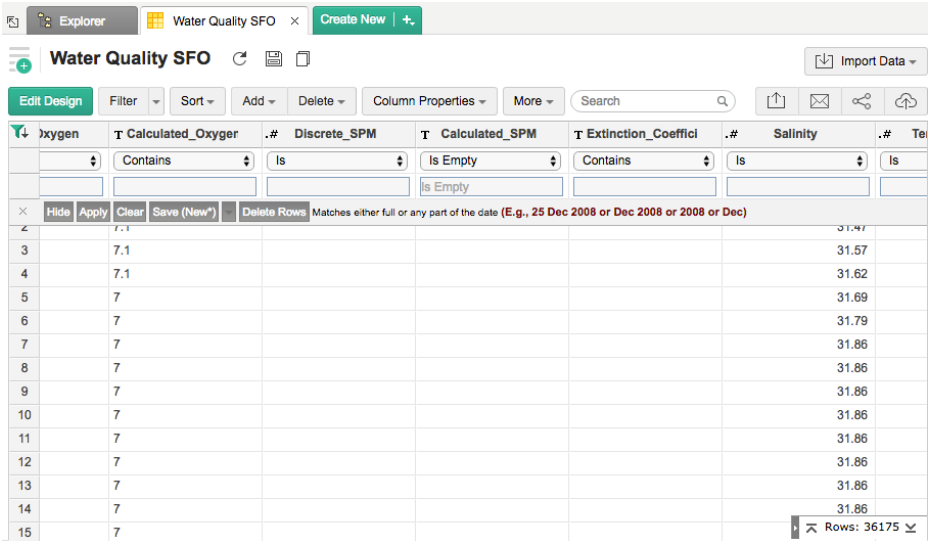


Figure 6. Filter applied is empty on parameter "calculatedSPM" (Zoho Reports tool).

To calculate $Completeness_{param^i}$ for the 1st transaction of the example data set, substitute values of E_{total} and M_{total} in Equation 10 as

$$Completeness_{param^{\text{Calculated\_SPM}}} = \frac{(E_{total} - M_{total})}{E_{total}} = \frac{210826 - 36175}{210826} = 0.8284.$$

To get the percentage value, substitute $Completeness_{param^1}$ into Equation 11 as

$$Completeness_{param^{\text{Calculated\_SPM}}}\% = 0.8284 \times 100 = 82.84\%.$$

The solution for Equation 11 is 82.84%, which means parameter Calculated_SPM is 82.84% complete.

Big Data Accuracy Models and Examples

Here, models for accuracy per transaction and accuracy per parameter are given. The accuracy per transaction model checks accuracy of each element in one single

transaction. The accuracy parameter model checks each element in parameter during all transactions for the given time. Both use percentage as the unit of measurement.

Model - accuracy per transaction. To calculate accuracy, a reference data set is required. The expected data set described in an earlier section is the reference data set for all calculations. For calculating accuracy per transaction, substitute m = 1 in Equation 1 and mr = 1 in Equation 2. The received data set is R. The distance between both the data sets selected gives their accuracy. Here, n will be the maximum number of parameters per transaction between the reference and received data sets.

Equation 16 defines accuracy per transaction as $Accuracy_{tran^k}$ for transaction k where, $Accurate_{ij}$ is the difference between the reference and received data sets for transaction *i* and parameter *j*. This is calculated as

$$Accurate_{ij} = 1 \text{ if difference does not exist between } E_{ij} - R_{ij}, \quad (15)$$

where i represents the number of transactions and j represents the number of parameters.

Let $Accuracy_{tran^k}$ be accuracy for transaction *k*. $Accuracy_{tran^i}$ can be defined as

$$Accuracy_{tran^k} = \frac{\sum_{j=1}^{n} Accurate_{kj}}{n}, \quad (16)$$

by substituting $Accurate_{ij}$ from Equation 15 with n as the number of parameters per transaction. The summation is applied over $Accurate_{ij}$ for all values of j equal to 1 to n number of parameters.

Equation 17 determines the percentage value of Equation 16. Let $Accuracy_{tran^i}\%$ be the percentage accuracy for transaction i. It can be defined as

$$Accuracy_{tran^i}\% = Accuracy_{tran^i} \times 100, \quad (17)$$

where the value of $Accuracy_{tran^i}$ can be substituted from Equation 10.

Let $Accuracy_{tran^{day^j}}$ represent average accuracy for day *j* in terms of the transactions occurring on that day. It can be determined as

$$Accuracy_{tran^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Accuracy_{tran^i}\%}{Transaction_{day^j}}, \quad (18)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6 and $Accuracy_{tran^i}\%$ is the percentage accuracy for transaction *i* derived from Equation 17. The summation is applied over $Accuracy_{tran^i}\%$ for all values of *i* which are equal from 1 to $Transaction_{\text{day}^j}$ to calculate accuracy for all transactions that happened during day *j*. This summation is divided by $Transaction_{\text{day}^j}$, producing the average parameter accuracy for day j.

Let $Accuracy_{tran^{month^j}}$ be the average accuracy for month *j* in terms of the transaction. Hence, $Accuracy_{tran^{month^j}}$ can be defined by

$$Accuracy_{tran^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Accuracy_{tran^{day^i}}}{Numberof_{days^j}}, \quad (19)$$

where $Numberof_{days^j}$ is the number of days in which transactions happened in month *j*, and $Accuracy_{tran^{day^i}}$ is from Equation 18. The summation is applied over $Accuracy_{tran^{day^i}}$ for all the values of i which are equal from 1 to $Numberof_{days^j}$ to calculate accuracy for all the transactions that happened for $Numberof_{days^j}$ days in month *j*. This summation value, when divided by $Numberof_{days^j}$, yields average transaction accuracy per month. Let $Accuracy_{tran^{year^j}}$ represent the average accuracy for year *j* in terms of each transaction. Hence, $Accuracy_{tran^{year^j}}$ can be given by

$$Accuracy_{tran^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Accuracy_{tran^{month^i}}}{Numberof_{months^j}}, \quad (20)$$

where $Numberof_{months^j}$ represents the number of months during which transactions happened in year j, with $Accuracy_{tran^{month^i}}$ derived from Equation 19. The summation is applied over $Accuracy_{tran^{month^i}}$ for all the values of i which are equal from 1 to $Numberof_{months^j}$ to calculate accuracy for all transactions that happened for $Numberof_{months^j}$ months in year j. This summation value, when divided by $Numberof_{months^j}$, produces average transaction accuracy per year.

Example - accuracy per transaction. To calculate an example, take the example data set described in an earlier section as received data set. For accuracy, a reference data is required. Table 2 is the reference data set used to show assumed example calculations.

In calculating accuracy for transaction number three, the total number of parameter n per transaction is 17. It is also observed that in Transaction 3 of the reference data set, two values are different from the example data set. Hence, in Equation 16, $Accurate_{3j}$=15. This is because two values in Transaction 3 are different from the example data set. Moreover, there are a total of 17 parameters per transaction. Hence, $Accuracy_{tran^3} = \frac{\sum_{j=1}^{n} Accurate_{3j}}{n} = \frac{15}{17} = 0.8823$. The accuracy for transaction number 3 comes out to be 0.8823. This value is substituted in Equation 17, giving a percent value of 83.23%.

Table 2

Reference Data Set (Assumed)

Date	Station_ Number	Depth	Discrete_Chlorophyll	Calculated_Chlorophyll	Discrete_Oxygen	Calculated_Oxygen	Calculated_ SPM	Extinction_ Coefficient	Salinity	Temp.
12/16/15	2.0	2.0		5.3		10.1	36	2.61	4.05	10.32
12/16/15	2.0	3.0		5.1		10.1	36		4.1	10.3
12/16/15	2.0	4.0		5.3		**10.2**	**30**		4.16	10.29
12/16/15	2.0	5.0		5.1		10.1	38		4.14	10.28
12/16/15	2.0	6.0		5.5		10.1	39		4.14	10.27
12/16/15	2.0	7.0		5.1		10.1	38		**5.1**	10.27
12/16/15	2.0	8.0		5.4		10.1	38		4.15	10.27
12/16/15	2.0	9.0		5.4		10.1	38		4.17	10.28
12/16/15	2.0	10.0		4.9		10.1	37		4.23	10.28
12/16/15	2.0	11.0		4.0		10	35		4.34	10.28
12/16/15	2.0	12.0		3.3		10	33		5.0	10.32

Note: Blank boxes represent null value in Table 2. Bold values represent changes from Table 1. Due to space limitation column, Discrete_SPM was deleted as it was completely null.

Model - accuracy per parameter. For accuracy per parameter, substitute n = 1 in Equation 1 and nr = 1 in Equation 2. Received data set is R. Let expected data set E described in the earlier section be the reference data set. Here, number of transactions will be shown as m, denoting the maximum number of transactions per parameter between reference and received data sets.

Equation 21 defines accuracy per parameter as $Accuracy_{param^k}$ for transaction. It is calculated as

$$Accuracy_{param^k} = \frac{\sum_{i=1}^{m} Accurate_{ik}}{m}, \quad (21)$$

where $Accurate_{ij}$ is substituted from Equation 15 and m is the number of parameters per transaction. Let $Accuracy_{param^i}\%$ be the percentage completeness for parameter i. It can be defined as

$$Accuracy_{param^i}\% = Accuracy_{param^i} \times 100, \quad (22)$$

where the value of $Accuracy_{param^i}$ can be substituted from Equation 10. Let $Accuracy_{param^{day^j}}$represent average accuracy for day *j* in terms of parameter. It can be determined as

$$Accuracy_{param^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Accuracy_{param^i}\%}{Transaction_{day^j}}, \quad (23)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6, and $Accuracy_{param^i}\%$ is the percentage accuracy for parameter *i* derived from Equation 22. The summation is applied over $Accuracy_{param^i}\%$ for all the values of i which are equal from 1 to $Transaction_{day^j}$ to calculate accuracy for all the transactions that happened during the day *j*. This summation value is divided by $Transaction_{day^j}$, producing the average parameter accuracy for day j.

Let $Accuracy_{param^{month^j}}$ be the average accuracy per month *j* in terms of parameter. Hence, $Accuracy_{param^{month^j}}$can be defined by

$$Accuracy_{param^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Accuracy_{param^{day^i}}}{Numberof_{days^j}}, \quad (24)$$

where $Numberof_{days^j}$ is the number of days in which transactions occurred in month j, and $Accuracy_{param^{day^i}}$ is from Equation 23. This summation is applied over $Accuracy_{param^{day^i}}$ for all the values of i which are equal from 1 to $Numberof_{days^j}$ to calculate accuracy s for all transactions that happened for $Numberof_{days^j}$ days in month j. This summation value, when divided by $Numberof_{days^j}$, gives average parameter accuracy per month.

Let $Accuracy_{param^{year^j}}$ represent the average accuracy for year j in terms of parameter. Hence, $Accuracy_{param^{year^j}}$ can be given by

$$Accuracy_{param^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Accuracy_{param^{month^i}}}{Numberof_{months^j}}, \qquad (25)$$

where $Numberof_{months^j}$ represent the number of months during which transactions happened in year j, with $Accuracy_{param^{month^i}}$ derived from Equation 24. The summation is applied over $Accuracy_{param^{month^i}}$ for all the values of i which are equal from 1 to $Numberof_{months^j}$ to calculate accuracy for all the transactions that happened for $Numberof_{months^j}$ months in year j. This summation value, when divided by $Numberof_{months^j}$, produces average parameter accuracy per year.

Example - accuracy per parameter. To calculate examples for accuracy per parameter, take parameter as salinity. The "salinity" row from the example data set in Table 1 is the received data set. The reference data set is Table 2's "salinity" row.

Only one value differs between the reference and received data sets. Hence, as per Equation 15, $Accurate_{m11}$=10, m=11 transactions as there are 11 transactions in total for the example data set. Hence, from Equation 21, $Accuracy_{param^{11}} = \frac{\sum_{i=1}^{m} Accurate_{i11}}{m} = \frac{10}{11} = 0.9090$. Substituting the above value in Equation 2, final the percentage value will be $Accuracy_{param^{11}} = 90.90$ %. Further calculations can be done to find this assessment for each day, month and year as per Equations 23, 24, and 25, respectively.

Big Data Timeliness Model and Example

Model - timeliness. According to the definition of timeliness, one should measure the time difference between the arrival and received times. To measure timeliness, one needs to store a time stamp for each transaction. Let $Record_{time}$ represent an array of timestamps for each record's start and end time. Hence, $Record_{time} = \{t_{1e}, t_{1r}, t_{2e}, t_{2r}, 2_r, t_{me}, t_{mr}\}$, where tie represents expected time for transaction i to arrive, and t_{ir} indicates actual received time for transaction i.

Let $Timeliness_{tran^i}$ be the timeliness for transaction i. It can be defined as

$Timeliness_{tran^i}$=1 if no difference between t_{ie} and t_{ir} else 0. (26)

Let $Timeliness_{tran^i}\%$ be the percentage timeliness for transaction i. It can be defined as

$$Timeliness_{tran^i}\% = Timeliness_{tran^i} \times 100, \quad (27)$$

where the value of $Timeliness_{tran^i}$ can be substituted from Equation 26.

Let $Timeliness_{tran^{day^j}}$ represent average timeliness for day j in terms of transaction. It can be determined as

$$Timeliness_{tran^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Timeliness_{tran^i}}{Transaction_{day^j}}, \quad (28)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6, and $Timeliness_{tran^i}\%$ is the percentage timeliness for transaction *i* derived from Equation 27. The summation is applied over $Timeliness_{tran^i}\%$ for all values of i which are equal from 1 to $Transaction_{\text{day}^j}$ to calculate timeliness for all transactions that happened during day j. This summation value is divided by $Transaction_{\text{day}^j}$, yielding average parameter timeliness for day j.

Let $Timeliness_{tran^{month^j}}$ be the average timeliness for month j in terms of the transaction. Hence, $Timeliness_{tran^{month^j}}$ can be defined by

$$Timeliness_{tran^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Timeliness_{tran^{day^i}}}{Numberof_{days^j}}, \quad (29)$$

where $Numberof_{days^j}$ is the number of days during which transactions happened in month j, and $Timeliness_{tran^{day^i}}$ is derived from Equation 28. The summation is applied over $Timeliness_{tran^{day^i}}$ for all the values of i which are equal from 1 to $Numberof_{days^j}$ to calculate timeliness for all the transactions that happened for $Numberof_{days^j}$ days in month j. When this summation value is divided by $Numberof_{days^j}$, average transaction timeliness per month is determined.

Let $Timeliness_{tran^{year^j}}$ represent the average timeliness for year j in terms of the transaction. Hence, $Timeliness_{tran^{year^j}}$ can be given by

$$Timeliness_{tran^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Timeliness_{tran^{month^i}}}{Numberof_{months^j}}, \quad (30)$$

where $Numberof_{months^j}$ represents the number of months when transactions happened in year j, and $Timeliness_{tran^{month^i}}$ is from Equation 29. The summation is applied over $Timeliness_{tran^{month^i}}$ for all values of i which are equal from 1 to $Numberof_{months^j}$ to calculate accuracy for all the transactions that happened for $Numberof_{months^j}$ months in year j. This summation value, when divided by $Numberof_{months^j}$, produces average parameter accuracy per year.

Example – timeliness. Filters are the same as per the example explained in the previous sections of this thesis. Timestamps are assumed as below. Here, timeliness is calculated per day in terms of transaction. For example, purposes "12/16/15" date is removed from the timestamp.

Data is expected to arrive at timestamps as follows:
00:00, 02:00, 04:00, 06:00, 08: 00, 10:00, 12:00, 14:00, 16:00, 18:00, 20:00, and 24:00.
Data are received at timestamps as shown below:
00:25, 02:00, 04:00, 06:05, 08:00, 10:00, 12:00, 14:00, 16:00, 19:00, 20:00, and 24:00.
Hence, record time can be given as below:
$Record_{\text{time}}$= {00:00, 00:25, 02:00, 02:00, 04:00, 04:00, 06:00, 06:05, 08:00, 08:00, 10:00, 10:00, 12:00, 12:00, 14:00, 14:00, 16:00, 16:00, 18:00, 19:00, 20:00, 20:00, 24:00, and 24:00}.

Timeliness for Transaction 1 as per Equation 26 will be 0, because there is a difference in the timestamps. From Equation 27, $Timeliness_{tran^1}\% = 0\%$. There are

three timestamps which differ from the excepted timestamp. Out of a total of 12 transactions, only 9 transactions are in time. Hence,

$\sum_{i=1}^{Transaction_{day^j}} Timeliness_{tran^i}\% = 900$. Moreover, 12 transactions happened in total. Hence, $Transaction_{day^j} = 12$.

From Equation, 28 instances of timeliness per day in terms of the transaction can be calculated as $Timeliness_{tran^{day^{12/16/15}}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Timeliness_{tran^i}\%}{Transaction_{day^j}} = \frac{900}{12}$=75.

Hence, for day 12/16/15, timeliness is 75%.

Big Data Uniqueness Model and Example

Model - uniqueness. Big data uniqueness is measured by comparing the data with their counterpart in the same data set to check redundancy. This section presents the uniqueness for each transaction made in one day. Suppose there is one transaction; to calculate its uniqueness, compare it with the rest of transactions.

Let $Uniqueness_{tran^i}$ be the uniqueness for transaction i. To define uniqueness of the transaction, compare that transaction with the rest of the transaction in the data set. Hence, $Uniqueness_{tran^i}$ can be defined as

$$Uniqueness_{tran^i} = 1 \text{ if no match found within data set else } 0. \quad (31)$$

Equation 32 determines the percentage value of Equation 31. Let $Uniqueness_{tran^i}\%$ be the percentage uniqueness for transaction i. It can be defined as

$$Uniqueness_{tran^i}\% = Uniqueness_{tran^i} \times 100, \quad (32)$$

where the value of $Uniqueness_{tran^i}$can be substituted from Equation 10.

Let $\text{Uniqueness}_{\text{tran}^{\text{day}^j}}$ represent average uniqueness for day j in terms of transaction. It can be determined as

$$Uniqueness_{tran^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Uniqueness_{tran^i}\%}{Transaction_{day^j}}, \quad (33)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6, and $Uniqueness_{tran^i}\%$ is the percentage uniqueness for transaction *i* derived from Equation 32. The summation is applied over $Uniqueness_{tran^i}\%$ for all the values of i which are equal from 1 to $Transaction_{\text{day}^j}$ to calculate uniqueness for all transactions that happened during the day *j*. This summation value is divided by $Transaction_{\text{day}^j}$, yielding average parameter uniqueness for day j.

Let $Uniqueness_{tran^{month^j}}$ be the average uniqueness for month *j* in terms of the transaction. Hence, $Uniqueness_{tran^{month^j}}$ can be defined by

$$Uniqueness_{tran^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Uniqueness_{tran^{day^i}}}{Numberof_{days^j}}, \quad (34)$$

where $Numberof_{days^j}$ is the number of days during which transactions happened in month *j*, and $Uniqueness_{tran^{day^i}}$ is from Equation 33. The summation is applied over $Uniqueness_{tran^{day^i}}$ for all values of i which are equal from 1 to $Numberof_{days^j}$ to calculate the uniqueness for all transactions that happened for $Numberof_{days^j}$ days in month *j*. This summation value, when divided by $Numberof_{days^j}$, gives average transaction uniqueness per month.

Let $Uniqueness_{tran^{year^j}}$ represent the average uniqueness for year j in terms of the transaction. Hence, $Uniqueness_{tran^{year^j}}$ can be given by

$$Uniqueness_{tran^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Uniqueness_{tran^{month^i}}}{Numberof_{months^j}}, \quad (35)$$

where $Numberof_{months^j}$ represents the number of months during which transactions happened in year j, and $Uniqueness_{tran^{month^i}}$ is from Equation 34. The summation is applied over $Uniqueness_{tran^{month^i}}$ for all the values of i which are equal from 1 to $Numberof_{months^j}$ to calculate the uniqueness for all transactions that happened for $Numberof_{months^j}$ months in year j. This summation value, when divided by $Numberof_{months^j}$, produces average transaction uniqueness per year.

Example – uniqueness. For example, take the first transaction from the example data set. Check with the rest of the data set transactions for redundancy. From Equation 31, $Uniqueness_{tran^1} = 1$, as there is no match found. To get percent value, substitute Equation 31 into 32, with $Uniqueness_{tran^1}\% = 100\%$.

Big Data Validity Models and Examples

The definition of big data validity correctly suggests that it involves a measure of validity. It is important to have rules, or syntax, with which one can assess accuracy. This section proposes validity at the transaction and parameter levels.

Model - validity per parameter. To validate data, there should be certain defined rules, which allow those data to be deemed valid. Suppose the received data set is R with mr transactions, and each transaction has nr parameters. For each parameter present in the

data set R, suppose validation criteria V= {v1, v2…vk}. To define validity per parameter, keep nr = 1 as in Equation 2. Check validity of each value item in parameter to determine which ones' validity need to be calculated. To validate parameter, each value of the parameter is measured against its rules to check validity.

Equation 36 defines validity for each value in the data set as $Validity_{value^i}$ for value *i*. It is calculated as

$$Validity_{value^i} = 1 \text{ if all validity rules passed else } 0. \quad (36)$$

Now, apply the summation of all the $Validity_{value^i}$ present in the parameter. Hence, $Validity_{param^i}$ for parameter i can be defined as below:

$$Validity_{param^i} = \frac{\sum_{j=1}^{mr} Validity_{value^j}}{mr}, \quad (37)$$

where mr is the total number of transactions, and $Validity_{value^i}$ is derived from Equation 36. Let $Validity_{param^i}\%$ be the percentage completeness for parameter i. It can be defined as

$$Validity_{param^i}\% = Validity_{param^i} \times 100, \quad (38)$$

where the value of $Validity_{param^i}$ can be substituted from Equation 36. Let $Validity_{param^{day^j}}$ represent average validity for day *j* in terms of parameter. It can be determined as

$$Validity_{param^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Validity_{param^i}\%}{Transaction_{day^j}}, \quad (39)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6, and $Validity_{param^i}\%$ is the percentage validity for parameter *i* derived from Equation 38.

The summation is applied over $Validity_{param^i}$% for all values of i which are equal from 1 to $Transaction_{\mathrm{day}^j}$ to calculate the validity for all transactions that happened during day *j*. This summation value is divided by $Transaction_{\mathrm{day}^j}$, producing the average parameter validity for day j.

Let $Validity_{param^{month^j}}$ be the average validity for month *j* in terms of parameter. Hence, $Validity_{param^{month^j}}$ can be defined by

$$Validity_{param^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Validity_{param^{day^i}}}{Numberof_{days^j}}, \qquad (40)$$

where $Numberof_{days^j}$ is the number of days on which transactions happened in month *j*, and $Validity_{param^{day^i}}$ is from Equation 39. The summation is applied over $Validity_{param^{day^i}}$ for all values of i which are equal from 1 to $Numberof_{days^j}$ to calculate validity s for all transactions that happened for $Numberof_{days^j}$ days in month *j*. This summation value, when divided by $Numberof_{days^j}$, gives average parameter validity per month.

Let $\mathrm{Validity}_{\mathrm{param}^{\mathrm{year}^j}}$ represent the average validity for year j in terms of parameter. Hence, $\mathrm{Validity}_{\mathrm{param}^{\mathrm{year}^j}}$ can be given by

$$Validity_{param^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Validity_{param^{month^i}}}{Numberof_{months^j}}, \qquad (41)$$

where $Numberof_{months^j}$ represents the number of months during which transactions happened in year *j*, and $Validity_{param^{month^i}}$ is from Equation 40. The summation is

applied over $Validity_{param^{month^i}}$ for all values of i which are equal from 1 to $Numberof_{months^j}$ to calculate validity for all transactions that happened for $Numberof_{months^j}$ months in year *j*. This summation value, when divided by $Numberof_{months^j}$, produces average parameter validity per year.

Example - validity per parameter. To find validity for parameter Station _number of example data set, assume validity rules as defined below for parameter Station _number:

- It should be between 2, 3, 4-36, 649, and 657.
- It should be a number.

Validity per parameter can be calculated as follows:

In Equation 36, the total number of transaction mr = 11 with $\sum_{j=1}^{mr} \text{Validity}_{\text{value}^i}$=11, as all values are valid and conform to the validity rule. From Equation 37, $\text{Validity}_{\text{param}^i} = \frac{\sum_{j=1}^{mr} \text{Validity}_{\text{value}^i}}{mr} = \frac{11}{11} = 1$. Hence, final $\text{Validity}_{\text{param}^i}$% is 100% for parameter Station _number.

Model - validity per transaction. To measure validity per transaction, it is necessary to have validity rules, or syntax, for each value in the transaction. That means each value needs to be compared with its rules. Suppose the received data set is R with mr transactions, and each transaction has nr parameter. For each value in the transaction, check its validity as per Equation 36.

Now apply the summation of all the $Validity_{value^i}$ present in the transaction. Hence, $Validity_{tran^i}$ for transaction i can be defined as below:

$$Validity_{tran^i} = \frac{\sum_{j=1}^{nr} Validity_{value^j}}{nr}, \quad (42)$$

where nr is the total number of parameters per transaction, and $Validity_{value^i}$ is derived from Equation 36. Equation 43 determines the percentage value of Equation 42. Let $Validity_{tran^i}\%$ be the percentage validity for transaction i. It can be defined as

$$Validity_{tran^i}\% = Validity_{tran^i} \times 100, \quad (43)$$

where the value of $Validity_{tran^i}$can be substituted from Equation 42.

Let $Validity_{tran^{day^j}}$ represent average validity for day *j* in terms of the transaction. It can be determined as

$$Validity_{tran^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Validity_{tran^i}\%}{Transaction_{day^j}}, \quad (44)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6, and $Validity\%$ is the percentage validity for transaction *i* derived from Equation 43. The summation is applied over $Validity_{tran^i}\%$ for all the values of i which are equal from 1 to $Transaction_{\text{day}^j}$ to calculate the validity for all the transactions that happened during day *j*. This summation value is divided by $Transaction_{\text{day}^j}$, producing the average parameter validity for day j.

Let $Validity_{tran^{month^j}}$ be the average validity for month *j* in terms of the transaction. Hence, $Validity_{tran^{month^j}}$can be defined by

$$Validity_{tran^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Validity_{tran^{day^i}}}{Numberof_{days^j}}, \quad (45)$$

where $\mathrm{Numberof}_{\mathrm{days}^{\mathrm{j}}}$ is the number of days on which transactions happened in month j, and $\mathrm{Validity}_{\mathrm{tran}^{\mathrm{day}^{\mathrm{i}}}}$ is from Equation 44. The summation is applied over $Validity_{tran^{day^i}}$ for all the values of $_i$ which are equal from 1 to $Numberof_{days^j}$ to calculate the validity for all the transactions that happened for $Numberof_{days^j}$ days in month j. This summation value, when divided by $Numberof_{days^j}$, gives average transaction validity per month.

Let $Validity_{tran^{year^j}}$ represent the average validity for year j in terms of transaction. Hence, $Validity_{tran^{year^j}}$ can be given by

$$Validity_{tran^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Validity_{tran^{month^i}}}{Numberof_{months^j}}, \tag{46}$$

where $Numberof_{months^j}$ represents the number of months during which transactions happened in year j, and $Validity_{tran^{month^i}}$ is from Equation 45. The summation is applied over $Validity_{tran^{month^i}}$ for all values of i which are equal from 1 to $Numberof_{months^j}$ to calculate the validity for all transactions that happened for $Numberof_{months^j}$ months in year j. This summation value, when divided by $Numberof_{months^j}$, produces average transaction validity per year.

Example - validity per transaction. For an example of validity per transaction, Transaction 1 from the example data set is derived into consideration. For each transaction in data set R, validation criteria are to be defined. Check these criteria for the first transaction's data value from Table 1. Different criteria for each parameter, like data, should be in MM/DD/YY format, and year should be between 69 to 15.

In Equation 42, put total number of parameter per transaction as nr = 17, where $\sum_{j=1}^{mr} \ Validity_{value^i}$=17, as all the data are valid and conform to the validity rule. From Equation 42, $Validity_{tran^i} \ \frac{\sum_{j=1}^{mr} \ Validity_{value^i}}{nr} = \frac{17}{17} = 1$. Therefore, final $Validity_{tran^i}$% = 100% for the first transaction of example data set.

Big Data Consistency Models and Examples

Here, this thesis presents two kinds of consistency; one is parameter based and another is time based. In parameter consistency, each value is compared against the value from a different data set. Whereas in time based, time stamps are compared to both data sets.

Model - consistency per parameter. This section defines consistency per parameter for parameter i of sensor X's data set and takes sensor Y's data set as the reference data set.

$$[Y]_{m\times 1} = \begin{bmatrix} Rp_{1y} \\ Rp_{2y} \\ \vdots \\ Rp_{my} \end{bmatrix} \quad [X]_{m\times 1} = \begin{bmatrix} Rp_{1x} \\ Rp_{2x} \\ \vdots \\ Rp_{mx} \end{bmatrix}$$

The dimension of X's data set should be equal to dimension Y's data set. If not, then substitute null in the absent dimension to make it equal so that mr is the total number of the transaction and is equal to the maximum of both data sets' transaction number. Consistency at station X with respect to Y can be given as Equation 47 for parameter i. Now, compare each data item present in parameter i as,

$Consistency_{value^i}$ =1 if no difference found in both data set else 0. (47)

Equation 48 defines consistency per parameter as $Consistency_{param^i}$ for parameter *i*. It is calculated as

$$Consistency_{param^i} = \frac{\sum_{j=1}^{mr} Consistency_{value^j}}{mr}, \quad (48)$$

where $Consistency_{value^j}$ is from Equation 47 and mr is the total number of transactions.

Let $Validity_{param^i}\%$ be percentage completeness for parameter *i*. It can be defined as

$$Consistency_{param^i}\% = Consistency_{param^i} \times 100, \quad (49)$$

where the value of $Consistency_{param^i}$ can be substituted from Equation 48.

Let $\text{Consistency}_{\text{param}^{\text{day}j}}$ represent average consistency for day j in terms of the parameter. It can be determined as

$$Consistency_{param^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Consistency_{param^i}\%}{Transaction_{day^j}}, \quad (50)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6, and $Consistency_{param^i}\%$ is the percentage of consistency for parameter *i* derived from Equation 49. The summation is applied over $Consistency_{param^i}\%$ for all values of i which are equal from 1 to $Transaction_{\text{day}^j}$ to calculate consistency for all the transactions that happened during day *j*. This summation value is divided by $Transaction_{\text{day}^j}$, producing average parameter consistency for day j.

Let $Consistency_{param^{month^j}}$ be the average consistency for month *j* in terms of the parameter. Hence, $Consistency_{param^{month^j}}$ can be defined by

$$Consistency_{param^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Consistency_{param^{day^i}}}{Numberof_{days^j}}, \quad (51)$$

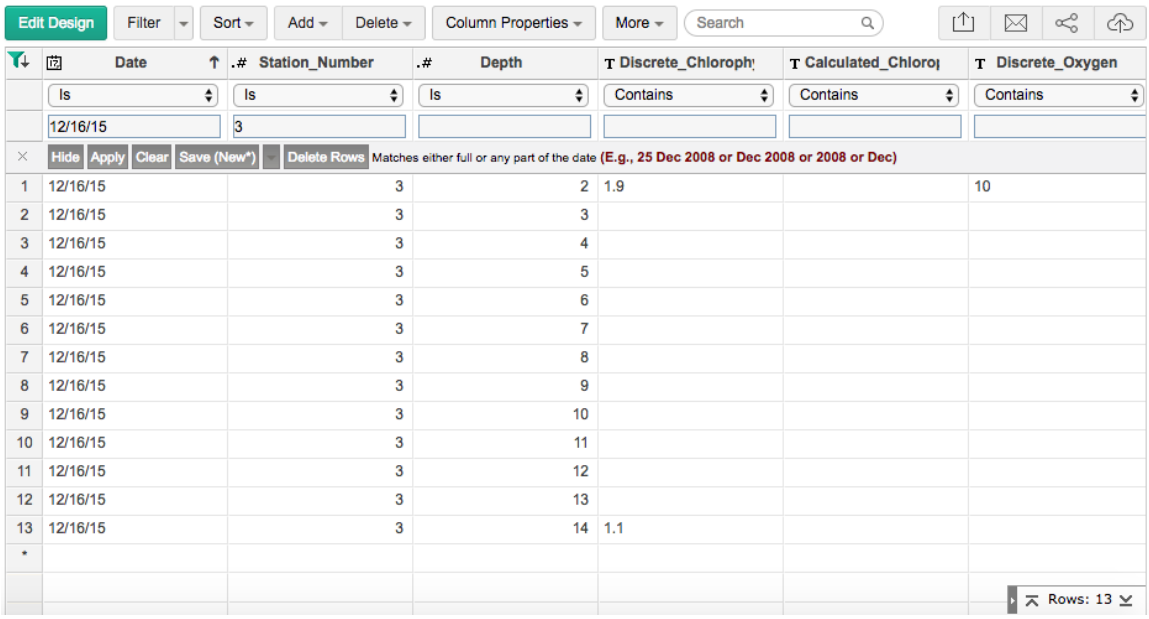
where $Numberof_{days^j}$ is the number of days on which transactions happened in month *j*, and $Consistency_{param^{day^i}}$ is from Equation 50. The summation is applied over $Consistency$ for all values of i which are equal from 1 to $Numberof_{days^j}$ to calculate consistency s for all the transactions that happened for $Numberof_{days^j}$ days in month *j*. This summation value, when divided by $Numberof_{days^j}$, gives average parameter consistency per month.

Let $Consistency_{param^{year^j}}$ represent the average consistency for year *j* in terms of the parameter. Hence, $Consistency_{param^{year^j}}$ can be given by

$$Consistency_{param^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Consistency_{param^{month^i}}}{Numberof_{months^j}}, \quad (52)$$

where $Numberof_{months^j}$ represents the number of months during which transactions happened in year j, and $Consistency_{param^{month^i}}$ is from Equation 51. The summation is applied over $Consistency_{param^{month^i}}$ for all values of i which are equal from 1 to $Numberof_{months^j}$ to calculate consistency for all transactions that happened for $Numberof_{months^j}$ months in year j. This summation value, when divided by $Numberof_{months^j}$, produces average parameter consistency per year.

Example - consistency per parameter. Take the example data set's depth parameter to calculate an example (Table 1). Reference data set Y indicates the depth parameter's values with filter date = 12/16/15, and Station_number is 3 (as shown below Figure 7).



	Date	Station_Number	Depth	Discrete_Chlorophy	Calculated_Chloro	Discrete_Oxygen
	Is	Is	Is	Contains	Contains	Contains
	12/16/15	3				
1	12/16/15	3	2	1.9		10
2	12/16/15	3	3			
3	12/16/15	3	4			
4	12/16/15	3	5			
5	12/16/15	3	6			
6	12/16/15	3	7			
7	12/16/15	3	8			
8	12/16/15	3	9			
9	12/16/15	3	10			
10	12/16/15	3	11			
11	12/16/15	3	12			
12	12/16/15	3	13			
13	12/16/15	3	14	1.1		

Figure 7. Data after applying filters as Station_number = 3, Date = 12/16/15 (Zoho Reports tool).

Station 2 (Figure 6) has two fewer transactions than Station 3 (Figure 7). Station 3 has 13 total transactions. Hence, mr = 13, taking maximum number of transactions among both

data sets. After comparing all the values present in both data sets, $Consistency_{value^j}$ from Equation 47, $\sum_{j=1}^{m} Consistency_{value^j}$ =11, as all the values are consistent except two null. Hence, as per Equation 48,

$Consistency_{param^{depth}} = \frac{\sum_{j=1}^{mr} Consistency_{value^j}}{mr} = \frac{11}{13} = 0.846$, and for percentage value from Equation 49, $Consistency_{param^{depth}}$ % = 84.61%.

Model - time consistency. Time consistency is measured to show time consistency between two data sets. For both sensors explained in an above section, X and Y measure the time transactions that were received to see if they maintain time consistency between the same transactions. Let $Record_{time}^{x}$ be defined as an array of the received timestamps for sensor X. For sensor X, $Record_{time}$ can be given as the following:

$Record_{time}^{x} = \{t_1^x, t_1^x, \ldots, t_{mr}^x\}$, where t_i^x represents the timestamp for the i^{th} transaction and mr represents the total number of transactions.

Let $TimeConsistency_{value^i}$ represent time consistency for transaction *i*. To define time consistency of the transaction, compare that transaction's timestamp with its reference data set's timestamp. Take Sensor X and Sensor Y to check time consistency between them. Hence, $TimeConsistency_{tran^i}$ for sensor X against sensor Y can be defined as

$TimeConsistency_{tran^i} = 1$ if no difference found between t_i^x and t_i^y else 0, (53)

where t_i^j represents the timestamp for the i^{th} transaction of sensor j.

Equation 54 determines the percentage value of Equation 53. Let $TimeConsistency_{tran^i}\%$ be the percentage time consistency for transaction i. It can be defined as

$$TimeConsistency_{tran^i}\% = TimeConsistency_{tran^i} \times 100. \quad (54)$$

Let $TimeConsistency$ represent average time consistency for day j in terms of the transaction. It can be determined as

$$TimeConsistency_{tran^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} TimeConsistency_{tran^i}\%}{Transaction_{day^j}}, \quad (55)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6, and $TimeConsistency\%$ is the percentage time consistency for transaction i derived from Equation 54. The summation is applied over $TimeConsistency_{tran^i}\%$ for all the values of i which are equal from 1 to $Transaction_{\text{day}^j}$ to calculate time consistency for all transactions that happened during day j. This summation value is divided by $Transaction_{\text{day}^j}$, producing the average parameter time consistency for day j.

Let $TimeConsistency_{tran^{month^j}}$ be the average time consistency for month j in terms of the transaction. Hence, $TimeConsistency_{tran^{month^j}}$ can be defined by

$$TimeConsistency_{tran^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} TimeConsistency_{tran^{day^i}}}{Numberof_{days^j}}, \quad (56)$$

where $Numberof_{days^j}$ is the number of days during which transactions happened in month j, and $TimeConsistency_{tran^{day^i}}$ is from Equation 55. The summation is applied over $TimeConsistency_{tran^{day^i}}$ for all the values of i which are equal from 1 to

$Numberof_{days^j}$ to calculate time consistency for all the transactions that happened for $Numberof_{days^j}$ days in month j. This summation value, when divided by $Numberof_{days^j}$ gives average transaction time consistency per month.

Let $TimeConsistency_{tran^{year^j}}$ represent the average time consistency for year j in terms of the transaction. Hence, $TimeConsistency_{tran^{year^j}}$ can be given by

$$TimeConsistency_{tran^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} TimeConsistency_{tran^{month^i}}}{Numberof_{months^j}}, \qquad (57)$$

where $Numberof_{months^j}$ represents the number of months during which transactions happened in year j, and $TimeConsistency_{tran^{month^i}}$ is from Equation 56. The summation is applied over $TimeConsistency_{tran^{month^i}}$ for all the values of i which are equal from 1 to $Numberof_{months^j}$ to calculate time consistency for all the transactions that happened for $Numberof_{months^j}$ months in year j. This summation, value when divided by $Numberof_{months^j}$, produces average transaction time consistency per year.

Example - time consistency. For both sensors X and Y, measure the time transactions received and see if they follow time consistency between the same data. Here, timestamp is assumed to show the following numerical calculation:

$Record_{time}$ for X =

{00:25, 02:00, 04:00, null, 06:00, 08:00, 10:00, 12:00, null, 14:00, 16:00, 19 :00, 20:00, 24:00}

$Record_{time}$ for Y =

{00:00, 02:00, 04:00, 05:00, 06:00, 08:00, 10:00, 12:00, 13:00, 14:00, 16:00, 18:00, 20:00, 24:00}

When assessing time consistency of the first transaction per Equation 53,

$$TimeConsistency_{tran^1} = 0,$$

there is a difference between the 1's transaction's timestamp of sensor X and sensor Y. From Equation 54, $TimeConsistency_{tran^1}\% = 0\%$.

Big Data Reliability of System Model and Example

Model - reliability of system. This parameter is indirectly connected to big data quality. It is important because data quality may get degraded if the system acquiring the data itself is faulty. Suppose station S has sensors as S= {S1, S2 , Sn} during the time interval with the help of finding the reliability of sensors. Different Techniques to do so can be seen in Zhu, Lu, Han, & Shi (2016). These techniques are beyond the scope of this thesis. All other parameters defined in this thesis are at the sensor level. Big data reliability is defined at the station level.

Let $Reliability_{Station^S}$ is reliability for Station S. To define reliability of station S with k unreliable sensor and n as the total number of sensor,

$$Reliability_{Station^S} = \frac{(n-k)}{n}, \tag{58}$$

where k is the number of the unreliable sensor and n is the total number of sensors.

Equation 59 determines the percentage value of Equation 58. Let $Reliability_{Station^S}\%$ be the percentage reliability for Station S. It can be determined as

$$Reliability_{Station^S}\% = Reliability_{Station^S} \times 100, \tag{59}$$

where the value of $Reliability_{Station^S}$ can be substituted from Equation 58.

Example - reliability of system. In Equation 58, n = 37 sensors as data set are derived into consideration from the previous section, with a total of 37 sensors. And assuming k = 4 sensors, reliability of Station S can be given by substituting n and k, as $Reliability_{Station^S} = \frac{(n-k)}{n} = \frac{33}{37}$ 33/37 = 0.89. From Equation 59, Station S is 89% reliable.

Big Data Usability Model and Example

Model - usability. Big data usability can be modeled by simply measuring three different quality parameters such as completeness, accuracy, and timeliness.

Let $Usability_{tran^i}$ be usability for transaction i. It can be determined as

$$Usability_{tran^i} = \frac{(Completeness_{tran^i} + Accuracy_{tran^i} + Timeliness_{tran^i})}{3}, \quad (60)$$

where $Completeness_{tran^i}$, $Accuracy_{tran^i}$ and $Timeliness_{tran^i}$ are from Equation 5, 18, and 29, respectively.

Equation 61 determines the percentage value of Equation 60. Let $Usability_{tran^i}\%$ be the percentage usability for transaction i. It can be defined as

$$Usability_{tran^i}\% = Usability_{tran^i} \times 100, \quad (61)$$

where the value of $Usability_{tran^i}$ can be substituted from Equation 60.

Let $Usability_{tran^{day^j}}$ represent average usability for day j in terms of the transaction. It can be given as

$$Usability_{tran^{day^j}} = \frac{\sum_{i=1}^{Transaction_{day^j}} Usability_{tran^i}\%}{Transaction_{day^j}}, \quad (62)$$

where $\text{Transaction}_{\text{day}^j}$ are transactions for day j derived from Equation 6, and $Usability_{tran^i}\%$ is the percentage usability for transaction *i* obtained from Equation 11. The summation is applied over $Usability_{tran^i}\%$ for all the values of i which are equal from 1 to $Transaction_{\text{day}^j}$ to calculate usability for all the transactions that happened during day j. This summation value is divided by $Transaction_{\text{day}^j}$, producing the average parameter usability for day j.

Let $Usability_{tran^{month^j}}$ be the average usability for month j in terms of the transaction. Hence, $Usability_{tran^{month^j}}$ can be determined as

$$Usability_{tran^{month^j}} = \frac{\sum_{i=1}^{Numberof_{Days^j}} Usability_{tran^{day^i}}}{Numberof_{days^j}}, \tag{63}$$

where $Numberof_{days^j}$ is the number of days during which transactions happened in month j, and $Usability_{tran^{day^i}}$ is from Equation 62. The summation is applied over $Usability_{tran^{day^i}}$ for all the values of i which are equal from 1 to $Numberof_{days^j}$ to calculate usability for all the transactions that happened for $Numberof_{days^j}$ days in month j. This summation value, when divided by $Numberof_{days^j}$, gives average transaction usability per month.

Let $Usability_{tran^{year^j}}$ represent the average usability for year j in terms of the transaction. Hence, $Usability_{tran^{year^j}}$ can be given by

$$Usability_{tran^{year^j}} = \frac{\sum_{i=1}^{Numberof_{months^j}} Usability_{tran^{month^i}}}{Numberof_{months^j}}, \tag{64}$$

where $Numberof_{months^j}$ represents the number of months transactions happened in year j, and $Usability_{tran^{month^i}}$ is from Equation 63. The summation is applied over $Usability_{tran^{month^i}}$ for all the values of i which are equal from 1 to $Numberof_{months^j}$ to calculate usability for all the transactions that happened for $Numberof_{months^j}$ months in year j. This summation value, when divided by $Numberof_{months^j}$, produces average transaction usability per year.

Example – usability. In Equation 60, substitute values of

$Completeness_{tran^i}, Accuracy_{tran^i}$ and $Timeliness_{tran^i}$ as follows:

$Usability_{tran^i} = \frac{(Completeness_{tran^i} + Accuracy_{tran^i} + Timeliness_{tran^i})}{3} = \frac{(52+100+0)}{3} = 0.5066$,

where the values for $Completeness_{tran^i}, Accuracy_{tran^i}, and\ Timeliness_{tran^i}$ are calculated in examples given in sections of respective parameters.

Hence, $Usability_{tran^i}\% = 50.66\%$ from Equation 61.

The Composite Outcome of Data Quality Parameters

This section presents how to calculate the composite outcome out of measurements made for each data quality parameter in this thesis with the help of two well-known methods Monte Carlo and Neural Networking. In Monte Carlo weight technique, certain predefined weightage (%) is applied to the data calculated based on the model discussed above to generate composite outcome to evaluate the data quality at station level. The estimation of weightage requires special attention and will be based on the relationship between data and the results. Sometimes the results may vary if the weightages are not defined correctly. Normally, a point for a relatively unknown system is the equal

weightage, and once more, real time data make available the weightage and can be changed. A regression analysis modeling can be used to decide the next set of weightages. This variable weightage technique is very effective and more practical to implement. On the other hand, neural networking involves a multi-level technique. The requirement of some levels (layers) and their neurons need careful selection. Training of the network is also very important and requires a lot of data and time. Improper training and methods used to estimate weights can generate errors as high as 40%. For data with high internal relationships, neural networking techniques are highly effective. However, if the data are discrete and have minimal relations to other data, neural networking techniques may become expensive. Without any internal layer or inter-relationship (between the data), this technique generates a result very close to the result obtained from the Monte Carlo variable weightage technique. For both techniques, eight factors are used to determine the accumulated result of the data quality at the sensor level: completeness, accuracy, timeliness, uniqueness, validity, consistency, reliability, and usability.

Monte Carlo. Below, Table 3 illustrates the basic calculation using the Monte Carlo method for evaluation of data based on defined models in the previous section. Here, weightage can be given as per requirement of the data. Supposing that completeness is not the prominent feature of data to assess quality, then put W1 as 0. Normally, the sum of the weight-age (weight factor) is 100. Hence, W can result from the set of weight-age. If the desired result is known and it is R, then a regression analysis can be performed using the least square method to re-estimate weight-age (w1, w2, etc.).

Table 3

Summation of The Parameters

	Data Point	Weightage (weight factor)	Data calculated based on thesis	Weight
1	Completeness Per Transaction	w_1	D_1	$w_1 \times D_1$
2	Accuracy Per Transaction	w_2	D_2	$w_2 \times D_2$
3	Timeliness	w_3	D_3	$w_3 \times D_3$
4	Uniqueness	w_4	D_4	$w_4 \times D_4$
5	Validity Data Time/ Record level	w_5	D_5	$w_5 \times D_5$
6	Value Consistency	w_6	D_6	$w_6 \times D_6$
7	Reliability of System	w_7	D_7	$w_7 \times D_7$
8	Usability	w_8	D_8	$w_8 \times D_6$
	Total	$\sum_{i=1}^{8} w_i = 100$	-NA-	$W = \sum_{i=1}^{8} wi \times Di$

Neural networks. Figure 8 below shows the graphical representation of a neural network, for the present data modeling system and considering only one hidden layer with three neurons. i.e. 8-3-1. To make the network readable not all the weights are displayed in Figure 8.

Here only one hidden layer is assumed with three neurons, but it can be changed. Normally for a less complicated system, one hidden layer yields strong results. The system is trained using output value versus the desired result R. One hidden layer with three neurons estimate 8*3+3*1 = 27 weights as against eight in the previous method (Monte Carlo). If more neurons are added to the hidden layer, there will be more weights

to be estimated and optimized. Hence, more training equals more data. After removing the hidden layer, the system becomes like the previous method (Monte Carlo).

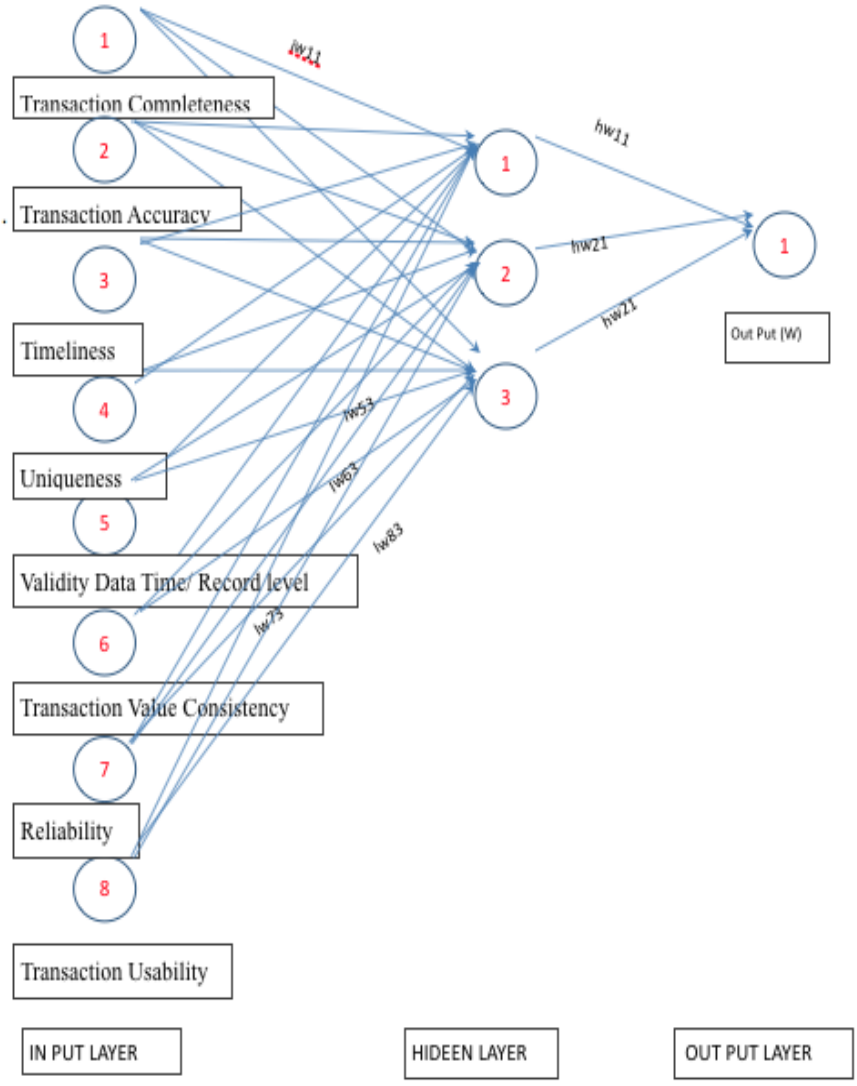


Figure 8. Neural network depicting all data quality parameters.

Chapter 4
Case Study – Predictive Analysis of Quality Parameters

Case Study Design

This case study aims to test the correctness of the quality parameter proposed in this thesis by applying predictive analysis to the quality parameters of "water quality data" collected from the National Estuarine Research Reserve System (NERRS). Correctness of the models can be checked by calculating the value of parameters with the help of proposed models in this thesis compared with predicted values by regression analysis. Six out of eight of those models were coded and made into a final year project by a group of Master's degree students at SJSU, Sampada Khandekar, Heen Mohare, and Spandana Boppana. The data set for this case study was collected from their software. The quality parameters implemented in the project are completeness, correctness, accuracy, timeliness, validity, uniqueness, and usability. The values are calculated for the years 2001 to 2014 and predictions for the year 2015 and 2016 are carried out.

Data Analysis

Data analysis involves a process of inspecting, cleaning, transforming, and modeling data in order to discover useful information that one can use to support the decision-making process (Jorge, 2017). The dataset for case study consists of one sensor's daily data transactions throughout many years. The data set consists of structured data, and it is downloaded in CSV format. After applying an Extract Transformation Load (ETL)

process, data were stored in MongoDB in a structured format. Quality parameter models were then applied to data and values for each quality parameter were calculated. The calculated values were stored in CSV files and provided for the case study.

Predictive Models

Before discussing the case study and its findings, this section explains prediction analysis and its various algorithms. Predictive modeling is the process of creating and validating a model to best determine the probability of an outcome (Jorge, 2017). Several modeling methods from machine learning, artificial intelligence, and statistics are available in predictive analytics. Each of them has its own weaknesses and strengths, so each is best suited for certain kinds of problems. These models fall into three categories defined in Table 4.

Table 4

Validation Model Categories

Category	Definition
Predictive Models	They analyze past performance for predicting the future.
Descriptive Models	They quantify relationships in data to classify datasets into groups.
Decision Models	They depict relationships between all variables of a decision to predict the results of decisions involving many variables.

Comparison of Prediction Models

Table 5 defines and provides examples for various algorithms which perform statistical analyses and data mining for predicting patterns and trends in data.

Table 5

Predictive Models

Model	What it does	Examples
Clustering	It clusters results into groups of similar groups.	Kohonen, K-means, and TwoStep.
Regression	Predicts relationships among variables.	Linear, Exponential, Logarithmic, Geometric, and Multiple Linear.
Time series	Time based prediction	Single, double, and triple exponential smoothing.
Association	To determine association rules, this algorithm finds the patterns in large transactional data sets.	Apriori
Decision Tree	Classifies and determines one or more discrete variables based on other variables.	C 4.5 and CNR Tree
Neural Network	It predicts, classifies and performs statistical pattern recognition.	NNet Neural Network, and MONMLP Neural Network

Regression Analysis

In this case study, regression analysis was carried out for predictive analysis. Regression analysis helps to estimate the relationship between the dependent and independent (explanatory) variables. If there is only one explanatory variable, then it is called simple linear regression, while if multiple explanatory variables are present, it is called multiple linear regression.

Linear Regression Analysis - Method

To conduct linear regression analysis on each quality parameter, observations were obtained for one quality parameter's measurements from the year 2001 to 2014 and plotted on the graph in Excel. Excel also provides the option to checkmark whether one wants to show the value of R^2 and equations on a graph or not. With the help of that, a value of R^2 is known. Figure 9 presents scatter plot for the completeness parameter; it gives the equation, with the help of this equation values for year 2015 and 2016 was predicted. Here, the value of R^2 is 0.822, which indicates that the regression equation can explain 80% of the variability of the data.

Years	Calculated	Predicted
2001	97.6	
2002	98.6	
2003	98.7	
2004	97.3	
2005	97.6	
2006	97.6	
2007	98.3	
2008	98.2	
2009	97.5	
2010	99.4	
2011	98.6	
2012	98.5	
2013	97.9	
2014	98.3	
2015	97.5	98.39
2016	98.7	98.42

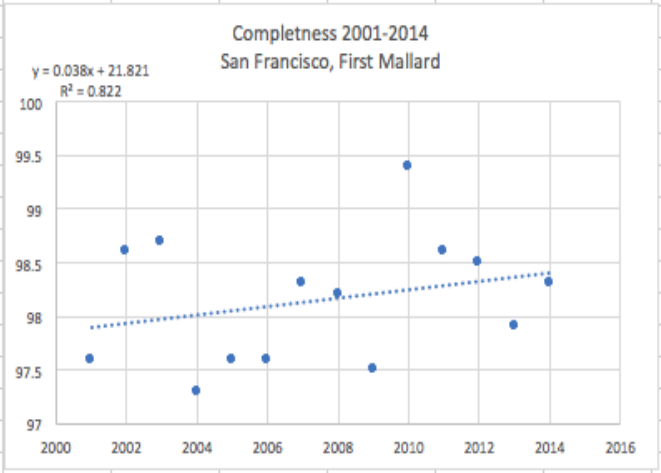


Figure 9. Scatter plot for parameter completeness (Year 2001- 2014).

Findings of the Case Study

Figures 10 and 11 depict a radar chart plotting calculated and predicted values for years 2015 and 2016, respectively. The radar chart is used to show the values for all parameters calculated and predicted values. The plotted lines are almost overlapping, and

the confidence interval for all quality parameters is around 95%. These are good indicators that the models proposed in this thesis are acceptable.

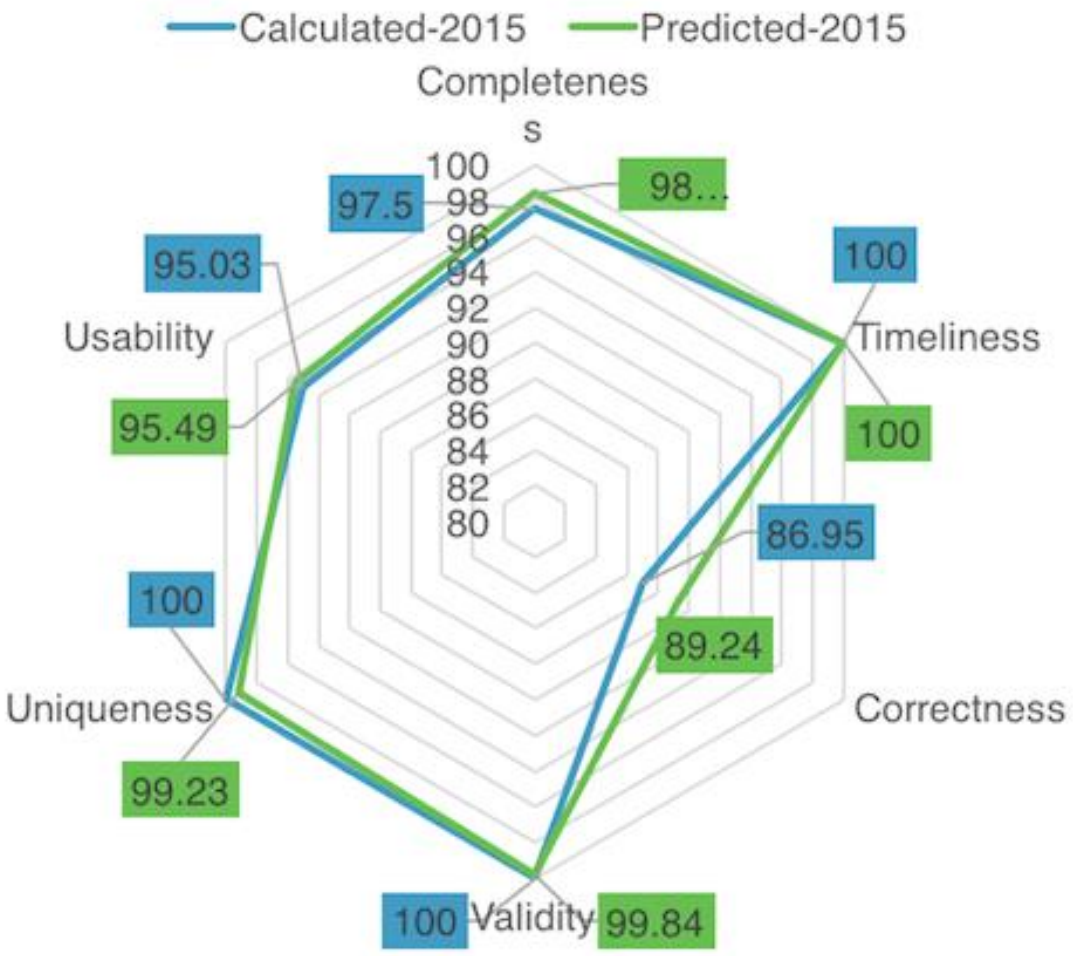


Figure 10. Data quality parameters predicted versus actual values for the year 2015.

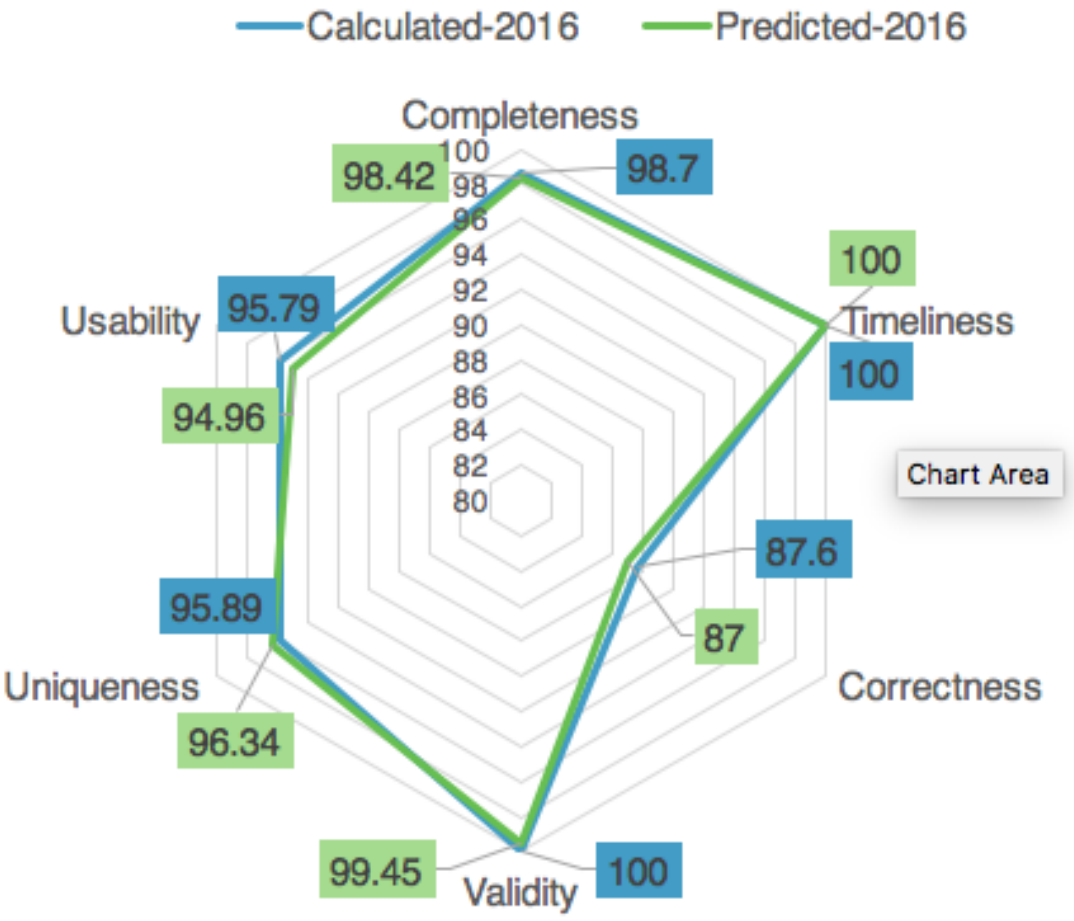


Figure 11. Data quality parameters predicted versus actual values for the year 2016.

Chapter 5
Conclusion and Future Work

This thesis has presented eight data quality parameters and proposed models for each that can be useful for measuring and predicting data quality. These models can be a starting point for developing more advanced modeling. In turn, these advanced models could then be used to generate benchmarks and protocols for assessing and optimizing data quality on larger scales. These measuring and predictive tools are helpful when comparing various data, as benchmarked data can be used for reliable decision making. A student group at San Jose State University (SJSU) used these proposed models to create a software tool for big data quality assessment as part of their master's project. The case study was carried out using the values acquired from the tool developed by the SJSU students. Predictive analysis was conducted with the help of linear regression analysis. Ideally, these results and proposed models can be extended in the future if they are studied and further developed by experienced professionals from industry and researchers from academic institutions.

References

Askham, N., Denise, C., Martin, D., Helen, F., Mike, G., Ulrich, L., Rob, L., Chris, M., Gary, P., and Julian, S. (2013). The six primary parameters for data quality assessment. Technical report, *DAMA UK Working Group*.

Cai, L., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data Science Journal, 14*, 2.

Clarke, R. (2014). Quality factors in big data and big data analytics. Xamax Consultancy Pry Ltd. Retrieved from http://www.rogerclarke.com/EC/BDQF.html.

Cloern, J.E., & Schraga, T. S. (2016). USGS Measurements of water quality in San Francisco Bay (CA). 1969-2015, version 2: U. S. Geological Survey data release, doi:10.5066/F7TQ5ZPR.

Eckerson, W. W. (2002). Data quality and the bottom line: Achieving business success through a commitment to high quality data. *The Data Warehousing Institute,* 1-36.

Gao, J., Xie, C., & Tao, C. (2016, March). Big data validation and quality assurance--Issues, challenges, and needs. In *Service-Oriented System Engineering (SOSE), 2016 IEEE Symposium* (433-441). IEEE.

Gudivada, V. N., Rao, D., & Grosky, W. I. (2016). Data quality centric application framework for big data. *ALLDATA 2016*, 33.

IDC Forecast: Big data technology and services to hit $32.4 billion in 2017. (2013, December 18). Retrieved from http://www.hostingjournalist.com/cloud-hosting/idc-forecast-big-data-technology-and-services-to-hit-32-4-billion-in-2017/.

Kang, G., Gao J. Z., & Xie, G. (2017, April). Data-Driven water quality analysis and prediction: A survey. *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService),* San Francisco, CA, 2017, 224-232.

Laranjeiro, N., Soydemir, S. N., & Bernardino, J. (2015, November). A survey on data quality: Classifying poor data. In *Dependable Computing (PRDC),* 2015 IEEE 21st Pacific Rim International Symposium on (179-188). IEEE.

Lavanya, S., & Prakasm, S. (2014). Reliable techniques for data transfer in wireless sensor networks. *International Journal of Engineering and Computer Science, 3 (12)*.

Loshin, D. (2010). The practitioner's guide to data quality improvement. Amsterdam, Netherlands: Elsevier.

Ludo, H., & Beek, J. (2013, December 16). Open data quality. Presentation published in *Technology*. Retrieved from https://www.slideshare.net/OpenDataSupport/open-data-quality-29248578.

Sharma, A. B., Golubchik, L., & Govindan, R. (2010). Sensor faults: Detection methods and prevalence in real-world data sets. *ACM Transactions on Sensor Networks (TOSN), 6(3),* 23.

Vass, L. (2016, 28 June). Terabyte terror: It takes special databases to lasso the Internet of Things. *Ars Technica*. Retrieved from https://arstechnica.com/information-technology/2016/06/building-databases-for-the-internet-of-data-spewing-things/.

What is predictive modeling? (2017) *Predictive Analytics Today*. Retrieved from http://www.predictiveanalyticstoday.com/predictive-modeling/.

Wigan, M. R., & Clarke, R. (2013). Big data's big unintended consequences. *Computer, 46(6),* 46-53.

Woodall, P., Gao, J., Parlikad, A., & Koronios, A. (2015). Classifying data quality problems in asset management. *In Engineering Asset Management-Systems, Professional Practices and Certification* (321-334). New York, NY: Springer, Cham.

Zhu, X., Lu, Y., Han, J., & Shi, L. (2016). Transmission reliability evaluation for wireless sensor networks. *International Journal of Distributed Sensor Networks, 12(2).*