

A system to quantify industrial data quality

AE Goosen



orcid.org/0000-0003-3159-8669

Dissertation submitted in fulfilment of the requirements for the degree *Master of Engineering in Computer and Electronic Engineering* at the North-West University

Supervisor:

Dr JC Vosloo

Graduation Ceremony: May 2019

Student number: 29924774

Abstract

Title: A system to quantify industrial data quality
Author: Mrs A.E. Goosen
Supervisor: Dr J.C. Vosloo
Keywords: Data quality, industrial data, data analysis, data quality dimensions, data quality quantification

The digital universe is expanding at an exponential rate – both in size and data variety. This phenomenon, known as big data, is making an impact on all industries. By adopting the big data principle, organisations can become more profitable, and efficient, and deliver better services and products. The increase in the volume of data, however, also leads to an increase in factors that could lead to poor data quality.

Poor data quality has been shown to increase operational costs, decrease customer satisfaction, and lead to inefficient decision-making processes. However, the benefits of managing and investing in data quality include increased customer satisfaction, increased revenue, reduced costs and greater confidence in analytical systems.

There are only a limited number of existing systems to analyse industrial data quality. Most systems focus on customer relation management or healthcare applications. Very few complete systems are aimed at analysing industrial data. Therefore, a need exists to develop a method and system to quantify industrial data quality.

During this study, an industrial data quality analysis method was developed to analyse industrial data quality. The method builds on the fundamentals of industrial sensor data analysis and data quality measurement methods. Furthermore, an automated system was developed to quantify industrial data quality using the results of the analysis method. The functionality of the analysis method was verified using scenarios that simulated common errors in industrial data.

The method and system were implemented at an industrial company to prove the feasibility of a data quality analysis system. The system was validated using condition monitoring input data for a large deep-level gold mining operation in South Africa.

Further analysis was performed using the condition monitoring input data. Results indicated that fans were the component with the most data quality problems, that temperature data was the measurement that had the poorest data quality, and that static data was the most common error found. Most of the data problems were traced back to poor communication between relevant systems.

It was also found that the industrial company's condition monitoring data for the specific gold mining operation was approximately 16% erroneous. The most common errors found in the specific condition monitoring data were caused by static data, missing data and data that

exceeded limits. The most likely cause for the missing data and static data is once again communication failures between relevant systems. The exceeds limits errors occur due to pieces of equipment being run outside their operational limits.

The developed system met the objectives of the study and succeeded in quantifying industrial data quality. The accuracy of the results can be improved by expanding the analysis to include contextual knowledge. The design of the system also allows for a platform to which more metrics can be added as required, and a basis for the development of a cross-industry platform for quantifying industrial data quality.

Acknowledgements

First of all, I would like to thank God for helping me through this. Without His love and grace it would not have been possible.

To my husband, Pieter Goosen, thank you for your unending love and support. You are an inspiration to me every day. Without your unwavering support none of this would have been possible.

To my mother and father, thank you for your love and support. Thank you for always believing in me and for making me believe in myself. Your love and support are appreciated immensely.

To Prof. E. H. Mathews and Prof. M. Kleingeld, thank you for giving me the opportunity to do my master's at CRCED Pretoria. Thank you also to TEMM International (Pty) Ltd and ETA Operations for funding my research.

To Dr J. Vosloo, Dr J. du Plessis and Dr S. van Jaarsveld, thank you for all the valuable time and input you have provided during the writing of this dissertation.

Table of contents

Abstract	ii
Acknowledgements	iv
Table of contents	v
List of figures	vii
List of tables	ix
Nomenclature	xi
1 Introduction	1
1.1 Background	2
1.2 Data in industry	2
1.3 Data quality	3
1.4 Problem statement	8
1.5 Objectives of the study	8
1.6 Overview of the dissertation	9
2 Literature review	11
2.1 Introduction	12
2.2 Industrial data errors	12
2.3 Data quality overview	17
2.4 Data quality assessment	23
2.5 Data analysis systems	24
2.6 Conclusion	29
3 Method	30
3.1 Introduction	31
3.2 User requirements and specifications	31
3.3 Error identification method	32
3.4 System design	37
3.5 Verification	46
3.6 Conclusion	53
4 Implementation and results	54
4.1 Introduction	55
4.2 Validation: Case study	55
4.3 Results	59
4.4 Discussion	74
4.5 Conclusion	83

5 Conclusion	85
5.1 Summary	86
5.2 Study closure	87
5.3 Limitations and recommendations	88
Reference list	90
A Outlier identification methods	94
A.1 Preamble	94
A.2 Moving average	94
A.3 Density-based spatial clustering of applications with noise	95
A.4 Nearest neighbours regression	96
A.5 One-class SVM	97
A.6 Consolidating results from outlier detection methods	97
B Metric verification results	98
B.1 Introduction	98
B.2 Missing data results	98
B.3 Static data results	100
B.4 Exceeds limits results	102
B.5 Negative data results	104
B.6 Outlier results	106

List of figures

2.1	Stuck-at error	14
2.2	Abrupt error	15
2.3	Data analysis steps	24
3.1	System overview	38
3.2	JSON document example	39
3.3	Configuration database entity relationship diagram	41
3.4	Configure tags for analyses	41
3.5	Configure limits for tags	42
3.6	Main data analysis process	43
3.7	Update documents steps	44
3.8	Tag analysis scheduling	45
3.9	Verification – Overview results	48
3.10	Verification – Detailed results	49
3.11	AllMetrics tag – Data plot	50
3.12	AllMetrics tag – Data point count	50
3.13	AllMetrics tag – Pie chart	51
4.1	Complete system overview	56
4.2	Mine sites overview	57
4.3	Mine A – Overview results	60
4.4	Mine A – Overview results of ten worst tags	61
4.5	Missing – Overview and detailed results	62
4.6	Missing – Data point count	62
4.7	Compressor 2 gearbox pinion non-drive end temperature bearing data plots .	63
4.8	Compressor 2 gearbox pinion non-drive end temperature bearing data plots – Specific dates	63
4.9	Exceeds limits – Overview and detailed results	64
4.10	Exceeds limits – Data point count	65
4.11	Compressor 3 gearbox pinion drive-end bearing temperature data plots . . .	65
4.12	Static – Overview and detailed results	66
4.13	Static – Data point count	67
4.14	Compressor 2 non-drive end vibration data plots	67
4.15	Data plot – 2018-06-25	68
4.16	Negative – Overview and detailed results	69
4.17	Negative – Data point count	69
4.18	Fridge plant 3 compressor drive end vibration data plots	70
4.19	Fridge plant 4 motor bearing temperature – Overview results	71
4.20	Fridge plant 4 motor bearing temperature – Detailed results	71
4.21	Fridge plant 4 motor bearing temperature – Data point count	72
4.22	Fridge plant 4 motor bearing temperature data plots	72

4.23	Fridge plant 4 motor bearing temperature data plots	73
4.24	Components – Overview results	77
4.25	Component – Metric breakdown	77
4.26	Measurement – Overview results	78
4.27	Measurement – Metric breakdown	79
4.28	Measurement – Scaled overview results	80
4.29	Measurement – Scaled metric breakdown	81
4.30	Metric results	81
A.1	DBSCAN	96
B.1	Missing tag – Data plot	99
B.2	Missing tag – Data point count	99
B.3	Missing tag – Pie chart	100
B.4	Static tag – Data plot	101
B.5	Static tag – Data point count	101
B.6	Static tag – Pie chart	102
B.7	ExceedsLimits tag – Data plot	103
B.8	ExceedsLimits tag – Data point count	103
B.9	ExceedsLimits tag – Pie chart	104
B.10	Negative tag – Data plot	105
B.11	Negative tag – Data point count	105
B.12	Negative tag – Pie chart	106
B.13	Outlier tag – Data plot	107
B.14	Outlier tag – Data point count	107
B.15	Outlier tag – Pie chart	108

List of tables

1.1	Data quality dimensions	3
1.1	Data quality dimensions continued	4
1.2	Causes of data quality problems	5
1.3	Data quality impact on an organisation	6
1.3	Data quality impact on an organisation continued	7
2.1	Metrics according to data quality dimensions	16
2.2	Definitions for completeness	18
2.3	Methods for measuring completeness	19
2.4	Definitions for availability	20
2.5	Definitions for accuracy	21
2.6	Methods for measuring accuracy	22
2.7	Existing CRM data quality systems	25
2.7	Existing CRM data quality systems	26
2.8	Existing EHR data quality systems	26
2.9	Other existing data quality systems	27
2.10	Existing industrial data quality systems	28
3.1	System features	32
3.2	Metrics used to measure data errors	33
3.3	Metrics according to data quality dimensions	36
3.4	Key-value pair in tag documents	39
3.5	Key-value pair in value documents	40
3.6	DataQualityAnalysisResult subdocument	40
3.7	Data points for each test tag	47
3.8	Expected results for each tag	47
3.9	Results comparison of tags	51
3.10	System requirements verification	52
3.10	System requirements verification continued	53
4.1	Mine A – Overview of condition monitoring implementation	58
4.2	Mine A – Input tag parameters	58
4.2	Mine A – Input tag parameters continued	59
4.3	Results summary	73
4.4	Overview of condition monitoring implementation – All sites	74
4.5	Number of components chosen for investigations	75
4.6	Number of tags analysed	75
4.7	Analysis results by system	76
4.8	Overview results	82
5.1	System requirements verification	87

B.1	Expected results for each tag	98
-----	---	----

Nomenclature

Abbreviations:

BSON	Binary JavaScript object notation
CRM	Customer relation management
DB	Database
DBSCAN	Density-based spatial clustering of applications with noise
EB	Exabyte
EHR	Electronic health records
EMR	Electronic medical records
ETL	Extract, transform and load
GUI	Graphical user interface
ID	Identifier
IoT	Internet of things
IT	Information technology
JSON	JavaScript object notation
OPC	Open platform communications
PLC	Programmable logic controller
SCADA	Supervisory control and data acquisition
SQL	Structured query language
SVM	Support vector machine
ZB	Zettabyte

Definitions:

Array	An indexed set of related elements.
Big data	A term that is used to describe large volumes of structured and unstructured data.
Data cleansing	Data cleansing refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying or deleting the dirty data.
Data decay	The gradual corruption of data caused by an accumulation of non-critical failures in a data storage device.
Data point	A record or observation, at a specific time, for a specific event.
Data processing	This refers to any operations performed on data, such as the retrieval, transformation or classification thereof.
Data profiling	Examining of data for different purposes.
Data resolution	In number storage, the resolution is the reciprocal of the unit in the last place.
Data source	It refers to the where the data comes from such as a file, database or live data feed.
Data staging	Intermediate storage used to store data for processing during the extract, transform and load processes.
Data store	A stored collection of data sets.
Data warehouse	A collection of data captured from various sources and used for reporting or analysis.
Database	A structured set of data held in a computer.
ISODate	“The International Organization for Standardization (ISO) date format is a standard way to express a numeric calendar date.” ¹
MongoDB	“A cross-platform and open-source document-oriented database; a kind of NoSQL database” (see next definition). ²
NoSQL	“A NoSQL database provides a mechanism for storing and retrieving data that is modelled in means other than the tabular relations used in relational databases.” ³
ObjectID	“The default primary key for a MongoDB document, which is usually found in the <code>_id</code> field in an inserted document.” ⁴
Outlier	A value that is much smaller or larger than most of the other values in a set of data.
Relational database	A collection of data sets organised according to relations between stored items.
Tuple	An ordered set of data constituting a record.
Unit in the last place	The spacing between floating-point numbers.

¹ISO 86001 *Date and time format*, ISO 8601, 1988

²MongoDB, ‘*What is MongoDB?*’, 2018. [Online]. Available: <https://www.mongodb.com/what-is-mongodb>. [Accessed: 23-10-2018].

³MongoDB, ‘*NoSQL Databases Explained*’, 2018. [Online]. Available: <https://www.mongodb.com/nosql-explained>. [Accessed: 23-10-2018].

⁴MongoDB, ‘*ObjectId*’, 2008. [Online]. Available: <https://docs.mongodb.com/manual/reference/method/ObjectId>. [Accessed: 23-10-2018].

Chapter 1

Introduction



1.1 Background

Global data production is increasing at an exponential rate [1, 2]. According to the International Data Corporation, the data universe is expected to double in size every two years [2]. In 2010, the total amount of data on earth exceeded one zettabyte (ZB)¹ [3]. By the end of 2011, that number had grown to 1.8 ZB. It is expected that this number will reach 44 ZB by 2020 [1, 3].

According to McKinsey & Company, manufacturing industries generated close to 2 exabytes (EB)² of data in 2010 [4]. Today, modern industries as a whole generate more than 1000 EB of data annually, which is expected to increase twentyfold in the next ten years [1, 2].

The exponential growth of data is influenced by factors such as [2]:

- Increasing use of social media such as Facebook[®], Twitter[®] and Instagram[®],
- The widespread use of Internet of things (IoT) devices, and
- Affordability of sensors.

For data to be of value, it needs to be stored, processed and analysed. If done correctly, big data³ can deliver competitive advantages for businesses [5, 6]. Valuable insights can be and are being extracted from big data [6, 7]. Big data can help organisations create data transparency, improve performance, customise services for specific segments, automate decision-making and innovate [8].

Big data has had a beneficial impact on many sectors [8]. Search engine companies such as Google, Yahoo! and Bing have radically changed the way large collections of information are searched [8]. Their successes were enabled by advancing technologies directly linked to big data. These technologies include natural language processing and semantic technologies.

Data analysis and data processing are enabling organisations to make strategic and operational decisions. However, since organisations are managing more extensive and complex information resources, the risk of poor data quality has increased [9, 10]. Poor data quality can have significant negative impacts on an organisation, which presents a problem as high-quality data is crucial to a company's success [11]. However, according to several industry expert surveys data quality is an area that companies do not manage efficiently or give sufficient attention to [9].

1.2 Data in industry

One of the most popular definitions of data defines it as discrete, objective facts about events [11]. However, data can also be defined as providing a record of signs and observations collected from various sources. A data point can thus be regarded as a record or observation, at a specific time, for a specific event.

¹A unit of information equal to one sextillion bytes or, strictly, 2^{70} bytes.

²A unit of information equal to one quintillion bytes or, strictly, 2^{60} bytes.

³Big data refers to voluminous amounts of structured or unstructured data that organisations can mine and analyse for business gains.

Advances in technology have led to data growing at an exponential rate [1–3, 12, 13]. In the industrial sector, data can be generated automatically from different sources. Data can be measured on different pieces of equipment. For each piece of equipment, different parameters can be measured. Each measurement can be taken at different intervals. All these factors lead to a significant amount of data that must be analysed.

Industrial systems that generate data are also becoming more complex [13]. Industrial processes usually possess the following properties [14]:

- Generate tremendous amounts of data.
- Have a strong coupling between systems.
- Have a large amount of uncertainty.
- Consist of diverse data types.
- Have incomplete data sets

Data is used in every aspect of the industrial process. Industrial data can be used in applications and activities such as [13, 15]:

- Production activities,
- Condition monitoring,
- Maintenance scheduling,
- Cost estimation,
- Planning activities, or
- Environmental monitoring.

There are many more uses of industrial data. It is important to note that many of the activities that use data affect one another. This is because the industrial systems that generate data are strongly coupled complex systems [13, 14]. For instance, shutting down a system in a manufacturing process for maintenance means that the production output may need to be reduced, or other systems may need to work harder to produce the same production output level to compensate for the missing system in the process [15].

1.3 Data quality

1.3.1 What is data quality?

Data quality can be defined as the fitness or suitability of data to meet business requirements [9, 10, 16, 17]. Alternatively, data quality can be defined by characterising data according to data quality dimensions [9, 10, 17]. Table 1.1 summarises the data quality dimensions commonly used to characterise data.

Table 1.1: Data quality dimensions

Dimension	Description	References
Accuracy	Accuracy refers to whether the data is correct, objective, reliable, certified and validated.	[9–11, 16–18]
Accessibility	Accessibility refers to whether the data is easily accessible, usable and retrievable.	[9, 11, 16–18]

Table 1.1: Data quality dimensions continued

Dimension	Description	References
Completeness	Completeness refers to whether all the values that is supposed to be in a collection, are in the collection.	[9–11, 16–18]
Consistency	Consistency refers to whether the data is consistent and presented in the same format.	[9–11, 17, 18]
Integrity	Integrity refers to whether the data is coherent.	[9–11, 18]
Believability	Believability refers to whether the data is true and credible.	[11, 16, 18]
Compliance	Compliance refers to whether data complies with regulatory and industry standards.	[10, 11, 18]
Objectivity	Objectivity refers to whether the data is unbiased, unprejudiced and impartial.	[11, 16]
Relevance	Relevance refers to whether the data is applicable to the task at hand.	[11, 16, 16, 17]
Timeliness	“Timeliness refers to the delay between a change of a real-world state and the resulting modification of the information system state.”	[9, 16, 17]
Validity	Validity refers to whether data is within acceptable parameters.	[9–11, 18]
Reliability	Reliability refers to whether the data is correct and reliable.	[11, 16, 18]
Availability	Availability refers to whether the data is physically available.	[16–19]

Data quality can be constructed using a combination of the dimensions mentioned in Table 1.1. However, deciding which dimensions to use to determine the quality of data depends on the type of data and what it is used for.

Some of these dimensions are easy to quantify. For instance, completeness can be quantified by determining how many values are found in a collection compared with how many values are expected. However, others dimensions are more difficult to measure due to the subjective nature of the dimension. For instance, relevance is difficult to quantify because data that is relevant to one activity might not be relevant to another activity. The same is true for believability and reliability.

1.3.2 Causes of poor data quality

Some of the most common causes of data quality problems arise at the data source. However, it is worth noting that problems can also occur at the data profiling or data staging phases. Data profiling is the process of examining the data available. Data staging is the process of temporarily storing data during the extraction, translation and loading phase. Table 1.2 summarises the common causes of data quality problems.

Table 1.2: Causes of data quality problems [9, 10]

Source	Cause
Data source	Measurement errors.
	Measurement equipment errors.
	Missing values in data sources.
	Outliers.
	Different timeliness of data sources.
	Lack of validation routines.
	Unexpected system changes.
	Approximations used in data sources.
	Contradictory information in data sources.
	Different encoding formats.
	Orphaned or dangling data.
Profiling	Information derived manually about the data.
	Insufficient data analysis against reference data.
	Insufficient range and distribution of values for required fields.
	Insufficient threshold analysis for required fields.
	Insufficient pattern analysis for given fields within each data store.
Staging and extract, transform and load (ETL)	Different business rules for various data sources.
	The inability to schedule data extractions.
	Purging of data from the data warehouse.

The causes mentioned in Table 1.2 usually occur due to [9, 10]:

- External errors such as equipment failure,
- System errors such as measurement equipment failure,
- Changes to the source system,
- Improper data handling processes,
- Inconsistent data entry and maintenance procedures, and
- Errors when migrating data from one system to another.

In industrial processes, external errors can be caused by equipment failures, for example, pumps or compressors [15]. These external errors cause data problems such as missing data and varying timeliness of data, which can lead to contradictory information being present in data sources. Outlier values can also be created by large spikes in data when equipment fails.

One of the leading causes of system errors in industrial processes is measurement equipment failure. These types of systems error lead to different data problems such as measurement errors, outliers, missing values or contradictory information from data sources.

In industrial processes, poor data handling procedures can have severe consequences. It can lead to contradictory information between data sources or to missing data, which leads to manually derived information about data and varying timeliness of data sources, among others.

As can be seen, many causes mentioned in Table 1.2 can be produced by different circumstances. This means that there are many causes of poor data quality that should be analysed; however, due to the large volume of data generated (as discussed in Section 1.2), it is a tricky and time-consuming process.

1.3.3 Need for high-quality data

A crisis has been identified in data quality management by industry experts such as The Data Warehousing Institute [9], PwC [20] and Gartner Group [21].

“Organisations typically overestimate the quality of their data and underestimate the cost of errors.” [11]

It was found that up to 88% of data integration projects either fail or cost more than anticipated [22]. Additionally, up to 75% of organisations have identified costs stemming from dirty data, while poor data has caused up to 35% of organisations to delay or cancel new information technology (IT) systems [11, 22]. More than 50% of companies admit to not being confident in their data quality [9, 22]. Dirty data often cause business intelligence projects to fail, making it imperative that business intelligence-based projects are based on clean data. Additionally, only 15% of businesses are very confident in the quality of external data supplied to them [11, 22].

Another problem to address is that the information landscape is constantly changing. Customer expectations, compliance rules and business processes contribute to the ever changing information landscape. Data quality systems need to reflect and compensate for these changes. Usually large amounts of time and money are spent on custom coding and other traditional methods to solve immediate crises, however, these methods usually fail to address long-term problems.

Poor data quality can have severe negative effects on an organisation [9, 11, 16, 18, 22, 23]. Table 1.3 describes some of the ways an organisation can be affected by poor data quality.

Table 1.3: Data quality impact on an organisation

Dimension	Description
Operational costs.	It has been estimated that poor data quality could lead to business costs of 10–25 % of an organisation’s revenue. Up to 50% of an organisation’s IT budget might be spent correcting data quality issues [24].
Reduced customer satisfaction.	Many customers expect data to be correct and are especially unforgiving of data errors.

Table 1.3: Data quality impact on an organisation continued

Dimension	Description
Inefficient decision-making processes.	It is widely accepted that decisions are no better than the data they are based on. Since data is created and used in all daily operations, data is critical input to almost all decisions. Poor data quality leads to inefficient decision-making processes and hinders managers from making decisions [11, 16, 22, 23].
Lower performance and employee job satisfaction; difficulty building trust in company data.	Poor data quality often makes it difficult to build trust in company data [11, 22]. This leads to a lack of user acceptance of initiatives based on the data. This also leads to lower employee job satisfaction [16].
Negative impact on organisational culture.	Data captures an organisation's ideals and values. Data implicitly defines common terms used in an organisation and defines its internal language. This means that data is a significant contributor to organisational culture [16, 22]. Poor data can negatively affect an organisation's culture. Seemingly simple tasks such as defining common terms can become problematic, which divert management's attention from customers and the competition [16].

One of the most significant areas affected by poor data quality is operational cost [16, 23]. According to Larry English [24], the business cost of poor data quality may be as high as 10–25 % of revenue or total budget of an organisation. He also states that up to 50 % of an organisation's IT budget may be spent on trying to correct data quality issues. Poor data quality can lead to the following additional costs:

- Higher maintenance costs,
- Increased labour costs,
- Costs of time spent on irrelevant information,
- Lost revenue,
- Lost and missed opportunity costs,
- Increased data retrieval costs,
- Increased data administration costs,
- Process failure costs,
- Costs due to delays of product or service delivery,
- Costs due to a tarnished image (or loss of goodwill),
- Costs associated with personal injury, and
- Increased legal costs.

It is clear that poor data quality can be detrimental to an organisation's operations and success. However, data quality is not a one-way street. Companies have cited many tangible and intangible benefits by managing and investing in their data quality. These benefits include [9, 20]:

- Greater customer satisfaction,
- Creating a single version of the truth,
- Greater confidence in analytical systems,

- Reduced cost and increased revenue, and
- Reduced time spent reconciling data.

According to the PwC report [20], companies that manage their data as strategic resources and invest in its quality are gaining an advantage over competitors that fail to do so. It is clear that companies should do more to manage their data quality to reap the benefits associated with good quality data.

1.3.4 Data quality in industry

As discussed in Section 1.1, data is growing at an exponential rate [1, 2]. As shown in Section 1.2, industrial data is generated automatically from different sources. This leads to large amounts of data that need to be analysed, which becomes impossible without assistance. Therefore, systems are used to analyse the data automatically.

Many systems exist to analyse customer relation management (CRM) data [25–29] and electronic medical record data [30, 31], but very few systems exist to analyse industrial data. Some methods have been proposed to identify industrial sensor errors. However, these methods have not been incorporated into larger data management systems [32–37]. These methods also focus more on identifying the errors in data rather than quantifying the overall quality thereof. A more in-depth review of existing data quality analysis systems and methods can be found in Section 2.5.2.

1.4 Problem statement

The industrial sector generates vast amounts of data with different types, formats and resolutions. As mentioned in the previous section, analysing these large quantities of data is a time-consuming process. Because important decisions are made based on the data available, it is important that data be accurate and trustworthy. Some methods have been proposed to identify industrial data errors. However, there is a lack of systems that quantify industrial data quality. Therefore, a need exists to develop a system to quantify the quality of data from various sources, and with different types and resolutions.

1.5 Objectives of the study

The goal of this study is to develop a data quality analysis platform to monitor and quantify data quality in the industrial sector. Specific objectives have been identified in order to achieve this. These objectives can be divided into two sections: literature objectives and empirical objectives.

Literature objectives:

1. Review data quality dimensions as discussed in Table 1.1 and their application to industrial data.
2. Research how data quality is measured and quantified.

3. Research what types of error are found in industrial data.
4. Research how industrial data errors are identified.
5. Review existing data quality analysis systems.

Empirical objectives:

1. Develop a method to analyse data for errors. The method should be able to analyse industrial data with different types, resolutions and formats.
2. Develop and implement a system to quantify data quality. The system should use the method developed as per Point 1 to quantify industrial data quality.
3. Validate the method:
 - Create cases to test the method.
 - Analyse test cases using the developed method.
 - Compare results from the analysis to the expected results.

1.6 Overview of the dissertation

Chapter 1: Introduction

This chapter investigates the need for data quality analysis. Data and data quality are defined. Problems regarding poor data quality and the impact thereof on an organisation are investigated. The need for a data quality analysis system is identified. Finally, the problem statement and objectives of the study are stipulated.

Chapter 2: Literature review

This chapter investigates the various issues that can arise in industrial data and what methods are used to identify these problems. The various methods to analyse data quality according to data quality dimensions are investigated. The difficulties and shortcomings of existing data quality analysis systems are also discussed.

Chapter 3: Method

In this chapter, user requirements and specifications of the system are identified. A method is developed to analyse industrial data quality, and a system is designed to make use of this method. Both the method and system are verified using the user requirements and specifications defined earlier in the chapter.

Chapter 4: Implementation and results

The data quality analysis method and the system are implemented. A case study is discussed to demonstrate the impact of the data quality analysis system on data quality. Finally, the study is evaluated to ensure that it addresses the problem defined in Section 1.4 and meets the objectives stipulated in Section 1.5.

Chapter 5: Conclusion

Based on the results discussed in Chapter 4, conclusions are made regarding the data quality analysis system, and recommendations for future work are discussed.

Chapter 2

Literature review



2.1 Introduction

Section 1.2 defined data as discrete, objective facts about events [11]. Section 2.2 takes a more in-depth look at industrial signal data and the common faults that can occur therein.

As discussed in Section 1.3.1, data quality dimensions provide ways to measure and manage data quality. Different dimensions can be used depending on the application. Section 2.3 defines different data quality dimensions and analyses the impact that the specified data quality dimensions have on industrial applications and processes. The different methods used to analyse data quality are also discussed.

Section 2.4 evaluates the different methods that can be used to analyse data points for faults. Three functional forms are provided that can be used to create metrics to measure data quality using the various data quality dimensions.

Section 2.5 investigates various existing data quality analysis systems. Each of the systems is analysed to determine if the system quantifies data quality according to data quality dimensions, which dimensions the system analyses, and for which applications the system is used.

2.2 Industrial data errors

2.2.1 Common data errors

Industrial signal data covers all kind of physical values and conditions measured within an industrial process. The measured values could include temperature measurements; vibration measurements; flow rate of fluids; and much more. These conditions are measured as nominal values using sensors or actuators.

Industrial sensor measurements can be regarded as objective recordings of comparable recordings. Industrial signal data is also time discrete. The measurements are logged at specific points in time. Furthermore, the measured values are time dependent. This means that it is difficult to interpret the values without the historical data for context. In addition, industrial signal data comprises mostly continuous values, however, signal data can also appear binary in nature. For instance, the running status of a pump is either on (1) or off (0) [38].

Even though the data collection for an industrial plant is relatively simple, numerous errors can arise in industrial sensor data. An error can be defined as an unpermitted deviation of at least one parameter of the system from the acceptable, normal or standard condition [39]. There are four common data errors that can arise in sensor data [32]:

- Missing data,
- Stuck-at errors,
- Out-of-bounds errors, and
- Abrupt errors.

Each of these errors is discussed in the following sections.

Missing data

The most common causes of missing data is missing entries in a database or data that is entered incorrectly [38, 40, 41]. The process of handling missing data has been addressed extensively in literature. Many methods have been proposed to deal with missing data, however, these methods have been developed to deal with missing data in specific applications. These methods tend to have some drawbacks when applied in any other context. The methods that mostly appear in the literature can be divided into three categories [40].

- Case deletion: Case deletion uses only complete data sets. Fields with missing values are removed from the analysis. Many statistical packages impose this method. Case deletion can lead to a smaller subset of data. This means information in data fields may be eliminated because a value is missing. Case deletion can be a suitable method if the number of missing values is relatively small when compared with the size of the data set.
- Maximum likelihood estimation: Maximum likelihood is a method used to estimate the parameters of a statistical model. The method requires the complete data set for the statistical model.
- Simple imputation: The simple imputation procedure aims to fill in missing values with estimated values.

Since the incompleteness of data is usually the cause of missing or erroneous values, data incompleteness is quite common in large-scale industrial processes and the resulting computations. The methods classified into the above-mentioned three categories need to be evaluated carefully before being chosen to handle data incompleteness. Case deletion leads to very small subsets of data that can be used. Maximum-likelihood estimation requires near complete data sets. Simple imputation fills in missing values, and could lead to erroneous values in the data set. These considerations show that data incompleteness remains a challenging topic to deal with [41].

Stuck-at errors

For some types of sensor reading, stable readings can be an indication of a sensor fault as shown in Figure 2.1 [32, 33]. The Stuck-at rule declares an error whenever the standard deviation of a set of successive values stays below a certain threshold [32].

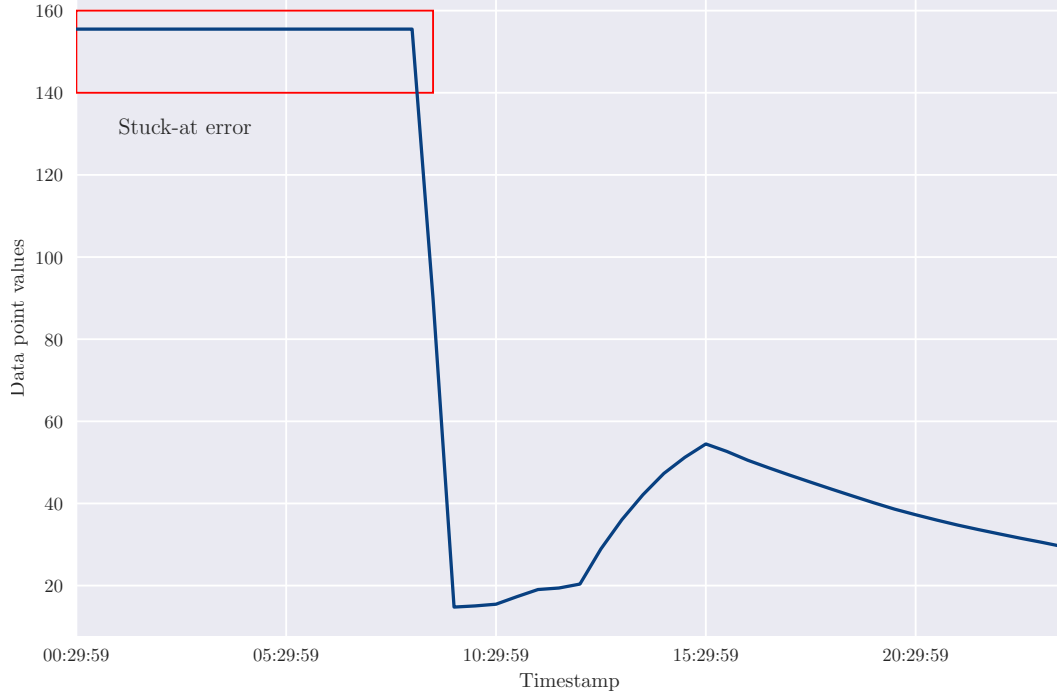
Equation 2.1 can be used to determine if a number of successive values have become stuck.

$$\sigma_V < \sigma_{min} \quad (2.1)$$

where:

σ_V = The standard deviation of a set of N successive values

σ_{min} = The minimum expected threshold

Figure 2.1: *Stuck-at error*

Out-of-bounds errors

Each sensor type has a range in which the sensor measurement is valid. Each sensor has a maximum and a minimum threshold wherein the measurement values are expected to be [32, 42]. The Out-of-bounds rule declares a fault whenever the values exceed the expected minimum and maximum thresholds. Equation 2.2 and Equation 2.3 can be used to determine whether a value exceeds the expected thresholds [18].

$$\bar{v} > v_{max} \quad (2.2)$$

$$\bar{v} < v_{min} \quad (2.3)$$

where:

- $\bar{v}$ = The value analysed
- v_{max} = The maximum expected values
- v_{min} = The minimum expected values

A special case of the minimum threshold test can be used to determine if certain sensor measurements are negative as shown in Equation 2.4.

$$\bar{v} < 0 \quad (2.4)$$

Abrupt errors

Abrupt errors are the most commonly seen sensor data errors. Data from sensors tend to show a pattern of change between successive measurements. However, a sudden change can occur due to erratic hardware behaviour or an environmental event [32, 33, 42]. Figure 2.2 shows an example of an abrupt error.

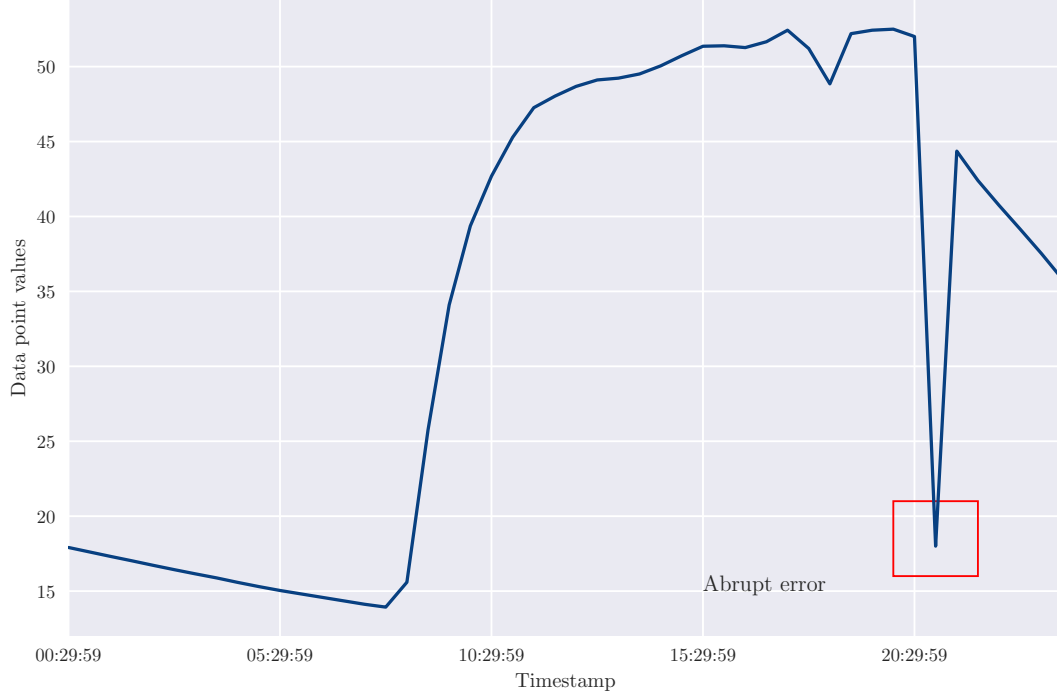


Figure 2.2: *Abrupt error*

The rate of change between consecutive readings can be used to determine if measurements are anomalous [18]. Equation 2.5 can be used to determine the rate of change between measurements. Alternatively, outlier detection or novelty detection can be used to identify abrupt changes in sensor data.

$$\frac{\Delta_V}{\Delta_t} > \Delta_{max} \quad (2.5)$$

where:

Δ_V = The variation of values

Δ_t = The variation of times

Δ_{max} = Threshold for the rate of expected changes

2.2.2 Industrial data errors and data quality dimensions

As discussed in Section 1.3.1, data quality dimensions provide a way of measuring the quality of data. Table 1.1 defined the various data quality dimensions. From these definitions it is possible to determine which data quality dimensions can be used to measure industrial data quality.

Table 2.1 summarises which dimensions can be used to quantify data quality according to which common industrial data errors. Table 2.1 lists the different data quality dimensions. Each data quality dimension is marked with an X where the dimension can be used to identify the data errors discussed in Section 2.2.1.

Table 2.1 shows that data with stuck-at errors, out-of-bounds errors and abrupt errors affect data quality in terms of accuracy. Missing data affects data quality in terms of completeness and availability. Data with out-of-bounds errors have an impact on the validity of data.

Many of the remaining dimensions are subjective. For instance, integrity refers to whether the data is coherent. Data could be logical and consistent to one person, and not for another. Believability is also subjective. Again, data could be believable to one person, and not to the next. The same is true for objectivity, relevance and reliability.

Compliance, timeliness and accessibility are dimensions that consider data as a whole. These dimensions are not suitable to measure data quality when considering errors in data on a per data point basis.

Table 2.1: Metrics according to data quality dimensions

		Data errors			
		Missing	Stuck-at	Out-of-bounds	Abrupt
Dimensions	Accuracy		X	X	X
	Accessibility				
	Completeness	X			
	Consistency				
	Integrity				
	Believability				
	Compliance				
	Objectivity				
	Relevance				
	Timeliness				
	Validity			X	
	Reliability				
	Availability	X			

2.3 Data quality overview

2.3.1 Effects of poor industrial data quality

Poor industrial data quality leads to a number of negative impacts. These impacts include:

- Inaccurate data outputs, which prohibit or contaminate data-driven decisions,
- Misleading results and bias,
- Impact on revenue, sales and profitability,
- Increase in operational costs,
- Operational inefficiencies,
- Schedule disruptions,
- Inaccurate forecasting,
- Poor resource utilisation, and
- Inaccurate planning activities and decisions.

Many of these impacts can be linked to numerous data quality dimensions. For instance, a reduced input data size can be influenced by both data completeness and availability. Accuracy, believability, reliability and compliance can all affect the data used to make data-driven decisions.

Many of the impacts also have a knock-on effect. For instance, a reduced size of input data can lead to misleading results, which lead to inaccurate forecasting, which can lead to increases in operational costs and reduced revenue and profitability.

Inaccurate data leads to inaccurate data outputs used for decision-making. Inaccurate data leads to misleading results. In statistical processes, missing data can cause complete data sets to be removed from the input data, thereby reducing the input data size and compromising the results of the processes.

In industrial processes, many decisions are data-driven. For instance, the decision to choose when to buy production materials is made based on data regarding current levels of materials. If the data is incorrect, materials can be bought too early or too late. This leads to operational inefficiencies.

Poor data can lead to inaccurate forecasting and planning activities. For instance, data could indicate that maintenance is not required on a piece of equipment when that piece of equipment does need maintenance. This could lead to the piece of equipment breaking, creating the need for unscheduled maintenance, which leads to schedule disruptions.

Any operational inefficiency has a negative impact on revenue and operational costs. Thus it is desired that an industrial site operate as efficiently as possible. By increasing the quality of industrial data, the negative impacts on revenue and profits can be avoided.

2.3.2 Dimensions of data quality

Data quality dimensions provide a way to measure and manage the quality of data [43]. However, each dimension requires different tools, techniques and processes to measure it. It is critical to differentiate between dimensions to:

- Match dimensions against project requirements,
- Better understand what results the assessments will deliver, and
- Better define and manage the sequence of activities within time and resource constraints of a project.

It is crucial to choose data quality dimensions based on the needs of the project. It is, therefore, important to understand the effort required to assess each dimension. A baseline can be set using initial assessments of data quality dimensions. Additional assessments can be added to operational processes as part of ongoing monitoring and information improvement [43].

As discussed in Section 2.2.2, certain dimensions are more suitable for measuring industrial data quality. For this study, the focus will be on the following dimensions that were shown to be measurable:

- Completeness,
- Availability,
- Accuracy, and
- Validity.

Each dimension will be discussed in more detail in the following sections.

Completeness

There are many definitions for completeness. Table 2.2 summarises the definitions for completeness from literature.

Table 2.2: Definitions for completeness

Definition	Reference
It refers to whether all the information is available.	[10]
“Refers to the scope of the information in the data.”	[16]
“The extent to which data is not missing and is of sufficient breadth and depth for the task at hand.”	[17]
The degree to which all the required information is provided.	[18]
“Completeness refers to the degree to which data are full and complete in content, with no missing data.”	[44]

By analysing the definitions, it can be seen that completeness refers to the number of data values that are not missing in a data collection. It is essential to understand why values can be missing in order to characterise completeness. A value can be missing because [40]:

- It exists but is not known,
- It does not exist, or
- It is not known whether it exists.

In industrial processes, the most common causes of data loss are sensor failure or when sensors become uncalibrated. The data loss continues until the sensors are replaced, fixed or recalibrated. Another cause of missing data is weak signal strength between the site and

the data warehouse.¹ Messages can become lost during data transfer, leading to missing data.

There are many different methods in literature for measuring data completeness. Table 2.3 summarises the most commonly used methods to analyse data completeness. Table 2.3 shows that the methods required to measure the completeness of data are dependent on the types of data warehouse used. For relational databases, it is a simple matter of counting the number of null values or empty cells in a data set [25, 45].

Table 2.3: Methods for measuring completeness

Method to measure completeness	Data warehouse	Citation
Constraint violations can be quantified simply by counting them.	Relational database	[25]
Percentage of complete cells: Indicates the percentage of cells that are not empty and have a meaningful value assigned.	Relational database	[45]
Percentage of complete rows: Indicates the percentage of rows that do not have any incomplete cells.	Relational database	
$Completeness = \frac{\text{Number of not null values}}{\text{Total number of values}}$	Data store	[46]
$Completeness = \frac{\text{Number of tuples}^2 \text{ delivered}}{\text{Expected number}}$	Relational database	
A simple ratio method is usually applied to measure completeness where: $Completeness(D, R) = \frac{ D \cap R }{ R } \in [0, 1] \quad (2.6)$ where: D = Data set under measure R = Reference data set	Relational database	[47]

However, a few organisations still use relational databases to store data. In order to measure the completeness of data, it is useful to look at methods such as calculating the ratio between the available data and the amount of data that should be in the data set as shown in Equation 2.7 [44, 47].

$$T_{Completeness} = \frac{N_{Available}}{N_{All}} \quad (2.7)$$

where:

$T_{Completeness}$ = Data completeness

$N_{Available}$ = Values that are recorded

N_{All} = Values that could have been recorded

¹Central repositories of integrated data from one or more disparate sources.

Availability

Availability is another data quality dimension that can be measured. Table 2.4 summarises the definitions for availability. Table 2.4 shows that availability generally refers to whether the data is physically available for use.

Table 2.4: Definitions for availability

Definition	Reference
The extent to which data is available, or easily and quickly retrievable.	[16, 17]
The extent to which information is physically accessible. The data must be current and available within a specified time frame.	[19]

Many factors can affect data availability. Causes of poor data availability include:

- Equipment failure,
- System malfunctions, and
- Corrupted data.

Due to the nature of the working environments in industrial processes, equipment failure is a common occurrence. Availability is also affected by system failures such as when the physical data store malfunctions or the connection to the data store is unavailable. Data can also become corrupted, making it unusable and unavailable.

The availability of data can be measured using the same method employed to measure data completeness. If several data points are missing, it can indicate that the data is unavailable, and the user should investigate the causes thereof.

Accuracy

Accuracy is an important data quality dimension to analyse regardless of the application. Table 2.5 summarises the definitions of data accuracy.

“Accuracy refers to the degree to which data are equivalent to their corresponding real values.” [44]

There are many sources and processes that cause data inaccuracies. Each of these contributes its part to the total data quality problem. Processes that cause data inaccuracies include:

- Using inaccurate sensors,
- Using data from outside sources,
- Changing data from within, and
- Using decaying data.³

One of the most common causes of inaccurate industrial data is sensor inaccuracies. Sensors could be installed incorrectly or become uncalibrated, which causes inaccurate measurements.

²An ordered set of data constituting a record.

³Data decay is the gradual corruption of data due to an accumulation of non-critical failures in a data storage device.

Bringing data in from an outside source is another cause of data inaccuracies. This applies to organisations that manage data for industrial companies. Most organisations that manage data for industrial companies have various processes in place to manage and organise the data. These processes include data processing and cleansing.

During data processing, data can become corrupt or unusable, or errors can occur during the processing steps. Data cleansing can remove data that should not be removed. Because data cleansing could lead to missing data, it contributes to the overall data quality problem.

Data decay is another area that leads to data inaccuracies. The most common decay-related problems occur when changes to measurements are not recorded, when systems are upgraded, when data is used differently than before, or when expertise in an organisation is lost.

Table 2.5: Definitions for accuracy

Definition	Reference
Data objects must accurately represent the real-world values they are expected to model.	[10]
Data contents or source are kept in high consideration.	[16]
The extent to which data represents the real-world values	[17]
The extent to which data is correct, reliable and error free.	[30]
It refers to whether the data is correct, objective and valid.	[18]
“Accuracy refers to the degree to which data are equivalent to their corresponding real values.”	[44]
Data is correct and error free.	[16]
One measure of accuracy consists of determining inaccurate values using functional dependence rules.	[48]

There are many different methods in literature that can be used to measure data accuracy. Table 2.6 shows that the most common method to measure the accuracy of data is calculating the number of correct values in the data set. However, none of these methods indicate how to determine if the data point is correct.

Table 2.6: Methods for measuring accuracy

Method to measure accuracy	Reference
Syntactic accuracy is the distance measured between the value stored in the database and the correct value: $Syntactic\ accuracy = \frac{Number\ of\ correct\ values}{Number\ of\ total\ values}$	[46]
Number of delivered accurate tuples.	[46]
Percentage of accurate cells in a data set that have correct values according to the domain of the data set.	[45]
Accuracy in aggregation indicates the ratio between the error in aggregation and the scale of the data represented. This metric only applies to data sets that have aggregation columns or where there are multiple data sets that refer to the same information but on a different granularity level.	[45]

Validity

Validity can be considered as the correctness and reasonableness of data [10]. Validity refers to whether data is within acceptable parameters for use by the business [18]. This means that a value must be in a collection of possible accurate values. It is possible for a value to be valid but inaccurate. To be accurate, the value must also be correct.

In database applications, a value can usually be verified or validated by cross-checking multiple data sources [49]. However, in most cases it is impossible to identify the real-world value of an attribute at the moment of observation.

Industrial data originates from measurements taken throughout an industrial site. Errors in measurements can lead to invalid data. Measurement errors can be caused by intrinsic sensor inaccuracy, poor sensor calibration or faulty data transmission.

Validity can also be considered as a set of constraints to apply to the data used in a certain application. All the data objects from all the data sources must satisfy a set of validity rules [50]. These rules can be presented as Boolean functions where $VR(o) = true$, if rule VR is satisfied by data object o , otherwise $VR(o) = false$. Validity can then be considered as the logical AND result of all the rules as defined in Equation 2.8 [50].

$$Validity = \bigwedge_{i=1}^m VR_i(o) \quad (2.8)$$

where:

- $VR_i(o)$ = Validity rule
- o = Data object being analysed
- m = Number of rules

This means that if the result of one rule equals false, the data object is considered to be invalid. The actual rules are dependent on the domain-specific properties of the data object and the application it is used for. In general, two types of rule can be applied [50]:

- Static rules that can be validated by checking a single data instance. For example, the environmental temperature of a city during a summer month is between 0°C and 35°C.
- Dynamic rules that are used to validate if the data changes are reasonable. For example, a 20°C change in environmental temperature cannot happen within a few minutes.

2.4 Data quality assessment

In order to measure the data quality, a standard of measurement is required. A metric can be defined as a standard of measurement. Some functional forms can be used to define metrics. Metrics can be derived from literature to measure and quantify industrial data quality.

When quantifying data quality, it is important to deal with both subjective and objective assessments of data. Subjective assessments reflect the experience and needs of data users. It is important to note that users' behaviour is influenced by their assessment of the data.

Objective assessments can be either task independent or task dependent. Task-independent metrics are developed to analyse data without contextual knowledge of what the data is used for. These metrics can be applied to any data set regardless of the tasks the data sets are used for. Task-dependent metrics are developed for specific applications. These metrics usually include various constraints, business rules and regulations [17].

In order to develop the metrics required, a set of principles should be followed. Three pervasive functional forms exist for metric calculations [17, 18]:

- Simple ratio,
- Minimum or maximum operations, and
- Weighted average.

Additional refinement of these functional forms can be incorporated with ease as required.

Simple ratio: The simple ratio measures the ratio of desired outcomes to the total number of outcomes. Equation 2.9 can be used to calculate the ratio. With this equation, 1 represents the most desirable outcome, and 0 represents the least desirable outcome. This is the preferred ratio since it can be used to illustrate trends of improvement over time.

$$S = 1 - \frac{N_e}{N_o} \quad (2.9)$$

where:

S = Simple ratio

N_e = Number of exceptions

N_o = Number of outcomes

Minimum or maximum operations: The minimum or maximum operations can be applied to dimensions that require aggregation. Another use is computing the minimum or maximum operator among measured data quality dimensions. The minimum operator is useful when a conservative approach is required. If a more liberal interpretation is needed, the maximum operator is more useful.

Weighted average: The weighted average is useful when dealing with multivariate data. A good understanding of the importance of each variable is required before using this operation.

2.5 Data analysis systems

2.5.1 Background

Data analysis is a crucial tool used by organisations to drive decisions. Figure 2.3 shows the typical steps of a data analysis process. It is important that data analysis systems enable users to progress through the data analysis steps swiftly. The data analysis steps include:

1. Define the problem to be solved. For this study, the problem was defined in Section 1.4.
2. Obtain data. Data forms the backbone of data analyses. Without data, it is not possible to perform the analysis. During this step, the data gathered is typically cleaned and transformed for the purposes of the analysis.
3. Explore the data. This typically involves plotting the data, and searching for anomalies or patterns.
4. Model data. During this step, models are built, fitted and validated.
5. Convey insight gained from the analysis to end users. This is a crucial part of data analytics. The insights gained from data analysis can be conveyed using various methods. However, the most effective method is displaying the results visually.

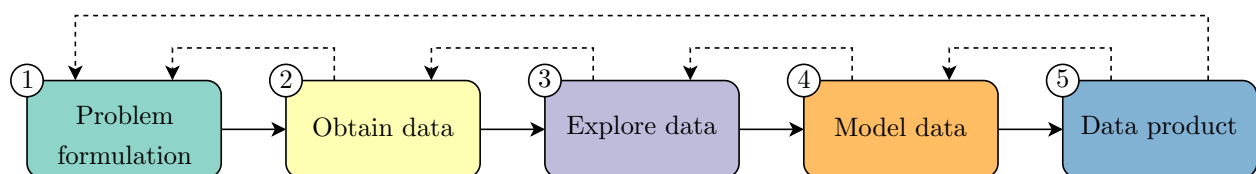


Figure 2.3: *Data analysis steps*

2.5.2 Existing data quality analysis systems

A critical analysis of literature revealed that various systems have been proposed and implemented to analyse and quantify data quality. However, each system is applied to different applications, analyses data in terms of different dimensions, and uses different techniques to analyse data quality.

Table 2.7, Table 2.8, Table 2.9 and Table 2.10 summarises the existing data analysis systems for various applications. Each table indicates whether an application-specific data analysis system has been implemented, and if it consists of a method only or is a theoretical framework that has not been implemented. The tables also indicate if the systems quantify data quality, and according to which dimensions data quality is analysed. Lastly, the tables summarise the different data applications used for each system.

The most popular application of data quality analysis is CRM. The data used in CRM has a high probability to be incorrect. Many systems have been proposed and implemented to analyse CRM data, as shown in Table 2.7. Systems have been implemented for use by telecommunications companies [25, 26] and automotive companies [27]. Methods have been used for enterprise information management and campaign management [28]. Methods have also been proposed and used for studies based on demographic and socio-economic information [29].

Many of the CRM data analysis systems analyse data quality in terms of dimensions [25–29]. Most systems analyse different dimensions. One system analyses currency [27] whereas another looks at timeliness and correctness [26]. Yet another system analyses completeness of data [29] while other systems analyse user-defined data quality dimensions [25, 28]. All systems look at dimensions specific to the application required.

The shortcomings of these systems vary. Some systems only analyse one or two dimensions. Some systems consist only of methods used to analyse data quality [28, 29]. Some systems do not quantify the data quality – whether in terms of data quality dimensions or not [29].

Table 2.7: Existing CRM data quality systems

Type of framework	Quantifies data quality	Evaluates dimensions	Application	Citation
Implemented	Yes	Dimensions defined by user	Customer database of telecommunications company.	[25]
Implemented	Yes	Timeliness Correctness	Campaign management of a major German mobile services provider.	[26]
Implemented	Yes	Currency	• German mobile services provider. • German automotive company. • Globally acting furniture manufacturing company. • Campaign management.	[27]

Table 2.7: Existing CRM data quality systems

Type of framework	Quantifies data quality	Evaluates dimensions	Application	Citation
Method	Yes	User-defined dimensions	• Customer information files. • Campaign management. • Compliance and transparency. • Enterprise information management.	[28]
Method	No	Completeness	Panel study of income dynamics data set.	[29]

Another popular application of data quality analysis systems analysing electronic medical and health records, which tend to be erroneous. Systems have been proposed to analyse the data quality of electronic health record (EHR) data and electronic medical record (EMR) data in terms of accuracy, completeness, consistency, relevance, etc. [30,31]. The dimensions analysed by these systems are chosen specifically for the data to be analysed. One of the shortcomings of these systems is that emphasis is placed on correcting and improving the data quality. The systems do not quantify the data quality in terms of the dimensions they analyse.

Table 2.8: Existing EHR data quality systems

Type of framework	Quantifies data quality	Evaluates dimensions	Application	Citation
Framework	No	• Accuracy • Objectivity • Believability • Timeliness • Appropriate amount of data	Data quality analysis and improvement in single-site and multisite clinical data (EHR)	[30]
Proposed framework	No	• Accuracy • Completeness • Consistency • Relevance • Timeliness • Usability • Provenance • Interpretability	Cloud-based health-care system	[31]

Other interesting applications are summarised in Table 2.9. These systems analyse data quality for ground solar radiation [51], experiments in biological crystallography [52], and open government data [45]. Many of these systems have been proposed or developed to help improve the data quality as required for the application.

For the ground solar radiation research, the historical ground solar radiation data was assessed [51]. The data was not assessed according to specific dimensions, nor was the quality of the data quantified.

A system was implemented to analyse open government data from the OpenCoesione project [45]. The system analysed and quantified data quality in terms of traceability, currentness, expiration, completeness, compliance, understanding and accuracy. The results were used to quantify the quality of the open government data in order to help researchers better understand the shortcomings of the data. The results also helped researchers to better choose which data sets to work with for specific projects.

Table 2.9: Other existing data quality systems

Type of framework	Quantifies data quality	Evaluates dimensions	Application	Citation
Implemented	Yes	• Accuracy • Traceability • Currentness • Expiration • Completeness • Compliance • Understanding	Open government data (OpenCoesione project)	[45]
Method	No	No	Assess quality of historical ground solar radiation data	[51]
Implemented	No	No	Biological crystallography	[52]

For the experiments in biological crystallography, a system was implemented to analyse the data quality [52]. The system was used to remove faulty data for the experiment from the baseline calculations. The data from the experiment was not analysed according to dimensions, nor was the quality thereof quantified.

A system was implemented to analyse open government data from the OpenCoesione project [45]. The system analysed and quantified data quality in terms of traceability, currentness, expiration, completeness, compliance, understanding and accuracy. The results were used to quantify the quality of the open government data in order to help researchers better understand the shortcomings of the data. The results also helped researchers to better choose which data sets to work with for specific projects.

Some methods have been proposed to identify sensor faults in industrial applications, as summarised in Table 2.10. These methods are usually used to identify sensor faults; they are not used to analyse the overall industrial data quality [32–37]. These methods are also rarely implemented as part of an overall data quality analysis system.

One of the methods has been proposed to analyse wireless sensor networks [32]. Data analysed by the system includes temperature, humidity and light measurements from sensors. The methods are not used to quantify data quality in terms of data quality dimensions.

Quite a few methods have been proposed to analyse data from industrial turbines [33–35]. These methods analyse flow, temperature, vibration and pressure sensors. All these methods analyse the data using different techniques to identify sensor faults. None of the methods quantify data quality according to data quality dimensions.

Methods also exist to analyse data from boilers [36] and pump systems [37]. These methods analyse data from a variety of sensors. These methods also do not quantify the data quality in terms of data quality dimensions.

Table 2.10: Existing industrial data quality systems

Type of framework	Quantifies data quality	Evaluates dimensions	Application	Citation
Method	No	No	Wireless sensor networks	[32]
Method	No	No	Sub-15 MW industrial gas turbine	[33]
Method	No	No	Industrial gas turbine system	[34]
Method	No	No	Aeroderivative gas turbine	[35]
Method	No	No	Industrial boiler	[36]
Method	No	No	Electro-pump system	[37]

While a significant amount of research has been done in terms of analysing, quantifying and improving data quality, whether in terms of data quality dimensions or not, not much work has been put into quantifying industrial data quality. Most data quality analysis systems have been implemented to analyse CRM data, or EHR and EMR data.

A multitude of methods have been developed and implemented to identify industrial sensor faults, but these methods fail to quantify industrial data quality in terms of data quality dimensions. Methods that do focus on data quality dimensions have been applied to different industries such as CRM data quality and do not focus on the dimensions desired for this study.

2.6 Conclusion

Industrial data is mostly generated by sensor measurements and calculations thereon. Four common data errors can be found in sensor data, namely: missing data, abrupt errors, noise errors, stuck at errors, and out-of-bounds errors. Different methods can be used to analyse the data for these errors.

Section 2.3.2 discussed the various data quality dimensions that can be analysed. Four dimensions were identified that have an impact on industrial applications and processes. The dimensions are completeness, availability, accuracy and validity. Different methods can be used to analyse the data quality dimensions.

Only a few existing systems can analyse data quality in terms of data quality dimensions. Most systems focus on analysing data used in CRM applications or healthcare data. Decidedly few complete systems are aimed at analysing industrial data. Many systems also only focus on a few specific dimensions. Few systems focus on analysing all four dimensions as discussed in Section 2.3.2.

Chapter 3

Method



3.1 Introduction

The literature discussed in Chapter 2 provided the need and a knowledge base upon which the data quality quantification system should be designed. The user requirements and specifications for the system are discussed in Section 3.2.

In Section 3.3, a methodology is developed to analyse data for specific errors and to quantify the quality of the data using the data quality analysis.

In Section 3.4, the system is designed to make use of the data analysis methodology. The existing systems, the data and the database designs are discussed.

The methodology and system are verified in Section 3.5. A test case is created to verify that the method and the system satisfy the user requirements and specifications defined in Section 3.2.

3.2 User requirements and specifications

3.2.1 User requirements

The user requirements are derived from the problem statement and objectives of the study defined in Section 1.4 and Section 1.5. The user requirements consist of data collection, data processing and user interface requirements.

1. Data collection:
 - Data must be collected from a data warehouse, the data must be analysed and the results of the analysis must be stored in the data warehouse.
2. Data processing:
 - The system must analyse the data and quantify the data quality according to the dimensions discussed in Section 2.3.2. These dimensions include completeness, availability, accuracy and validity.
3. User interface requirements:
 - A user interface is required to configure the analysis. The interface must allow the users to configure the required metrics to run on specific data sets.
 - The results of the analysis should be displayed using a user interface. The user interface must make it easy for the user to identify problematic data. The interface should also display which errors are found in the data, such as missing data, stuck-at errors, out-of-bound errors and abrupt errors.

3.2.2 Specifications

After the user requirements have been defined as described in Section 3.2.1, the specifications can be determined.. Table 3.1 summarises the core system features required. The specifications consist of analysing the data in terms of data quality dimension. Furthermore, automated processes are required to manage data retrieval. A system is also required to allow the user to manage configurations and show the results of the analyses to the user.

Table 3.1: System features

System feature	Description	Reference
1. Data collection		
Data retrieval from data warehouse	Retrieve data for a specified date range from the data warehouse.	Section 3.2.1
2. Data processing		
Completeness dimension	Quantify the completeness of data as discussed in Section 2.3.2.	Section 2.3.2
Availability dimension	Quantify the availability of data as discussed in Section 2.3.2.	Section 2.3.2
Accuracy dimension	Quantify the accuracy of data as discussed in Section 2.3.2.	Section 2.3.2
Validity dimension	Quantify the validity of data as discussed in Section 2.3.2.	Section 2.3.2
Overall score calculation	Determine the overall data quality score for a tag.	Section 2.5.2
Saving results of the quality analysis	Save the analysis results of each individual data point.	Section 3.2.1
3. User interface		
Tag configuration	Enable the user to manage the configurations.	Section 3.2.1
	Enable the user to configure which metrics to run.	Section 3.2.1
	Enable the user to configure metric specific settings.	Section 3.2.1
Display results	Display the overall score of the quality analysis to the user.	Section 3.2.1
	Display the metric scores of the quality analysis to the user.	Section 3.2.1
4. Other		
Data store	Use existing data store for information.	Section 3.2.1
Database	Use existing structured query language (SQL) database for configuration.	Section 3.2.1

3.3 Error identification method

3.3.1 Metric definitions

As discussed in Section 2.2, various problems can be found in industrial data. These errors include:

- Missing data errors,
- Stuck-at errors,
- Out-of-bounds errors, and
- Abrupt errors.

A number of metrics can be defined to measure data quality by analysing the data errors described in Section 2.2 and using the functional forms of the metrics described in Section 2.4. These metrics can be used to measure data quality in terms of data quality dimensions as discussed in Section 2.3.2. These metrics include:

- Missing metric,
- Static metric,
- Exceeds Limits metric,
- Negative metric, and
- Outlier metric.

Table 3.2 summarises which metrics can be used to analyse data for which data error. The table shows that each data errors can be measured using at least one metric. Missing data can be measured using the Missing metric, Stuck-at errors can be measured using the Static metric, Out-of-bounds errors can be measured using the Exceeds Limits metric and the Negative metric, and the Abrupt errors can be measured using the Outlier metric. The following sections define and discuss the above-mentioned metrics.

Table 3.2: Metrics used to measure data errors

		Metrics				
		Missing	Static	Exceeds Limits	Negative	Outlier
Data error	Missing data error	X				
	Stuck-at error		X			
	Out-of-bounds error			X	X	
	Abrupt error					X

Missing metric

Section 2.3.2 defined data completeness as the number of data values not missing in a data collection. Furthermore, it was determined that a value could be missing because [40]:

- It exists but is not known,
- It does not exist, or
- It is not known whether it exists.

It was found that the most common way of calculating data completeness is using Equation 2.7:

$$T_{Completeness} = \frac{N_{Available}}{N_{All}} \quad (2.7 \text{ revisited})$$

where:

$T_{Completeness}$ = Data completeness

$N_{Available}$ = Recorded values

N_{All} = Values that could have been recorded

Using the definition for data completeness and Equation 2.7, a metric can be defined to calculate the number of missing data points as shown in Equation 3.1.

$$T_{Missing} = \frac{N_{Available}}{N_{Expected}} \quad (3.1)$$

where:

$T_{Missing}$ = Metric score for the Missing metric

$N_{Available}$ = Number of data points in the date range from the database

$N_{Expected}$ = Expected number of data points in the date range from the database

Static metric

Similarly to the Stuck-at rule discussed in Section 2.2.1, the Static rule declares an error whenever the standard deviation of a set of successive values stays below a certain threshold. Equation 2.1 can be used to analyse successive data points to identify if the data points are static.

$$\sigma_V < \sigma_{min} \quad (2.1 \text{ revisited})$$

where:

σ_V = The standard deviation of a set of N successive values obtained by analysing the measurements

σ_{min} = The minimum expected threshold

A metric can be derived to calculate the number of data points that are static as shown in Equation 3.2.

$$T_{Static} = \frac{N_{Static}}{N_{Analysed}} \quad (3.2)$$

where:

T_{Static} = Static metric

N_{Static} = Number of static data points

$N_{Analysed}$ = Number of data points analysed

Exceeds Limits metric

As discussed in Section 2.2.1, the Exceeds Limits rule declares a fault whenever the values exceed the expected minimum and maximum thresholds. Equation 2.2 and Equation 2.3 can be used to determine whether a data point value has exceeded a limit.

$$\bar{v} > v_{max} \quad (2.2 \text{ revisited})$$

$$\bar{v} < v_{min} \quad (2.3 \text{ revisited})$$

where:

v_{max} = The maximum expected values

v_{min} = The minimum expected values

A metric can be derived to calculate the number of data points in sensor data that exceed the limits. Equation 3.3 is used to calculate the ratio between the data points that exceed set limits and the data points that do not.

$$T_{ExceedsLimits} = \frac{N_{ExceedsLimits}}{N_{Analysed}} \quad (3.3)$$

where:

$T_{ExceedsLimits}$ = Exceeds Limits metric

$N_{ExceedsLimits}$ = Number of data points that exceeds limits

$N_{Analysed}$ = Number of data points analysed

Negative metric

As discussed in Section 2.2.1, a negative data point is a data point that is less than 0. This can be considered as a special case of the Exceeds Limit metric. Equation 2.4 can be used to determine if a data point value is negative or not.

$$\bar{v} < 0 \quad (2.4 \text{ revisited})$$

Similarly to the Exceeds Limits metric, a metric can be derived to calculate the number of negative data points. Equation 3.4 is used to calculate the ratio between the negative data points and non-negative data points.

$$T_{Negative} = \frac{N_{Negative}}{N_{Analysed}} \quad (3.4)$$

where:

$T_{Negative}$ = Negative metric

$N_{Negative}$ = Number of negative data points

$N_{Analysed}$ = Number of data points analysed

Outlier metric

As discussed in Section 2.2.1, a sudden change can occur due to erratic hardware behaviour or an environmental event. This is referred to as an outlier. Various methods can be used to determine whether a data point is an outlier. The outlier identification methods are discussed in Appendix A.

Once the outliers in a data set have been identified, the metric defined in Equation 3.5 can be used to determine the ratio between the number of outlier data points and the non-outlier data points.

$$T_{Outlier} = \frac{N_{Outlier}}{N_{Analysed}} \quad (3.5)$$

where:

$T_{Outlier}$ = Outlier metric

$N_{Outlier}$ = Number of outlier data points

$N_{Analysed}$ = Number of data points analysed

3.3.2 Combining metrics and data quality dimensions

Multiple metrics can be associated with each quality dimension. Table 3.3 summarises dimensions that have to be measured and the metrics that can be used to measure them.

Table 3.3: Metrics according to data quality dimensions

		Metrics				
		Missing	Static	Exceeds Limits	Negative	Outlier
Dimensions	Completeness	X				
	Availability	X				
	Accuracy		X	X	X	X
	Validity			X	X	

From the definition of missing data and the definition of data completeness, as described in Section 2.3.2, the Missing metric can be used to quantify data completeness. The availability of data can also be quantified using the Missing metric discussed in Section 2.3.2.

From the definition of data accuracy as described in Section 2.3.2, all of, or a combination of the following metrics can be used to quantify the data accuracy:

- Static metric
- Exceeds limit metric,
- Negative metric, and/or
- Outlier metric.

From the definition of data validity, as described in Section 2.3.2, the Exceeds Limit metric and the Negative metric can be used to quantify the validity of data.

3.3.3 Quantify overall data quality

Once the data points have been analysed, the overall data quality score can be determined. Equation 3.6 is used to determine the overall score for the data.

$$T_{Overall} = \frac{N_{Correct}}{N_{Expected}} \quad (3.6)$$

where:

$T_{Overall}$ = Overall score

$N_{Correct}$ = Number of correct data points

$N_{Expected}$ = Expected number of data points analysed

A data point is considered erroneous if it has been flagged by any of the metrics or if it is missing. The expected number of data points is used in this calculation, to ensure that any missing data points are also taken into consideration.

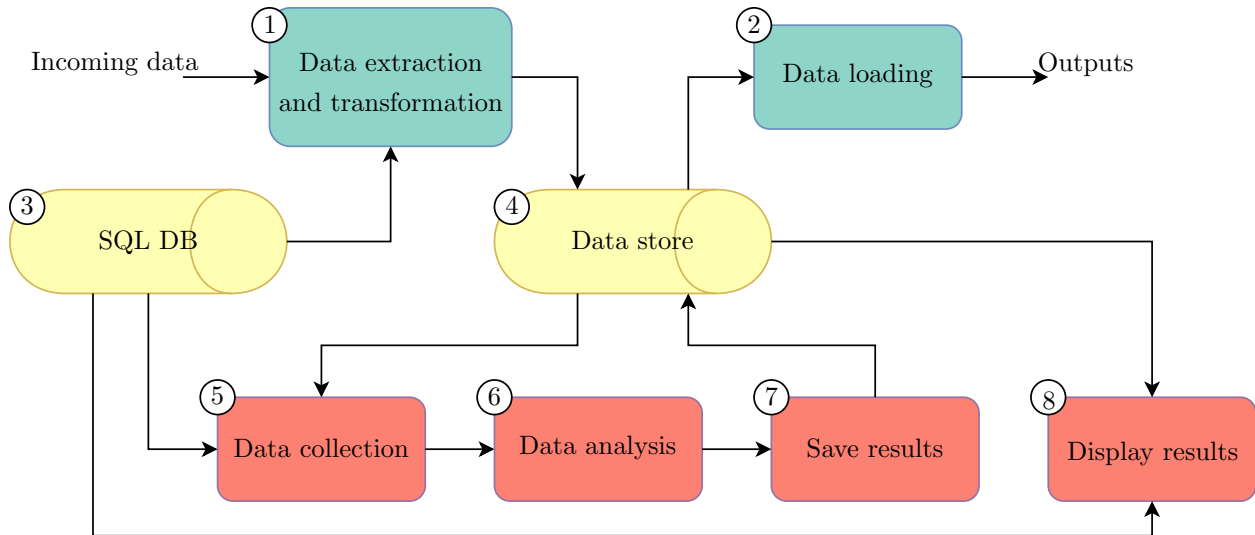
3.4 System design

3.4.1 System overview

An overview of the existing ETL system is shown in Figure 3.1. Data arrives in various formats at the extract and transform phase (1). The system processes the incoming data and stores it in a data store. During the load phase (2), data from the data store is loaded and sent to other processes as required.

The data is organised as data tags in the existing system. A data tag is a collection of related data. Each tag can have numerous measurements. A tag can also contain tag metadata, which describes the characteristics of the tag. The tag metadata and tag data are stored in a document-based database.

A data quality analysis system can easily be added to the existing system shown in Figure 3.1. The elements numbered 1 and 2 are existing components that will not be affected by the new system. The elements numbered 3 and 4 are existing components that must be modified. These elements are discussed in more detail in the following sections. The elements numbered 5 to 8 are the data quality analysis components that need to be developed and added.

Figure 3.1: *System overview*

The data quality analysis system consists of four main stages: the data collection stage (5), the data analysis stage (6), and the results stages (7 and 8) as shown in Figure 3.1.

In the data collection phase (5), the system determines which tags must be analysed, which metrics must be analysed for each tag, and for what time range the data must be analysed. Data is collected from the data store for analysis.

During the data analysis phase (6), the system analyses the data for each tag according to the metrics that must be analysed for each tag.

During the save results phase (7), the results of the analysis for each data point is saved to the data store. Lastly, a web-based interface is used to display the results of the analysis (8).

3.4.2 Database design

As discussed in the previous section, the data is organised as data tags, and each tag is a collection of related data. In order to store the tag information and measurements, the system makes use of two databases: a data store and an SQL database.

Each tag is saved in the data store as JavaScript object notation (JSON)¹-like documents. Tag configurations used by the ETL system are stored in the SQL database. In the following section, both types of database are discussed.

Data store

Information is stored as JSON-like documents in a document-based database. This means that each document can have varying fields and data structures. Figure 3.2 shows how to interpret a JSON document as a conceptual key-value pair. These key-value pairs consist of an attribute name and a value. Attribute values can consist of any data type such as strings, integers, floating-point values, collections or subdocuments.

¹<https://www.json.org/>

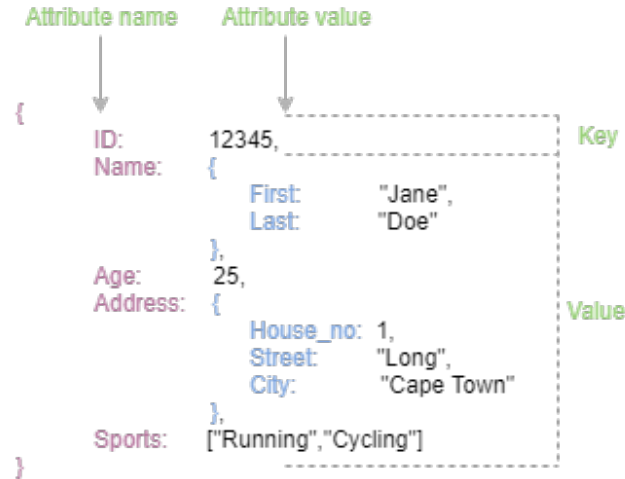


Figure 3.2: JSON document example

Two important documents are defined and used in the data store: a **tag** document and a **value** document. The **tag** document contains tag-specific information. Table 3.4 shows the key-value pairs stored in a **tag** document. The *_id* (identifier) field is a unique identifier for each **tag** document. The *TagNames* field is a collection of names used to identify the tags. The *IntervalMask* is a dictionary that indicates what the tag resolution is. Other fields are also found in the **tag** document; however, these fields are not used by the data quality analysis system.

Table 3.4: Key-value pair in tag documents

Key	Value	Description
<i>_id</i>	ObjectId()	Used to distinguish between tag documents.
<i>TagNames</i>	Array of strings	Contains the various tag names used for this tag.
<i>IntervalMask</i>	<pre> { "Year" : 0, "Month" : 0, "Day" : 0, "Hour" : 0, "Minute" : 30, "Second" : 0, "Millisecond" : 0 } </pre>	Used to indicate the resolution of the tag data.

A **value** document contains the measurement value for a specific tag at a specific date and time. Table 3.5 shows the key-value pairs stored in a **value** document. The *_id* is a unique identifier for each value document. The *Timestamp* field indicates when the value was measured. The *TagID* field is used to link the measurement to a specific **tag** document.

The *Value* field contains the measured value. The *WasValueInterpolated* field indicates if the value was interpolated.²

Table 3.5: Key-value pair in value documents

Key	Value	Description
<i>_id</i>	ObjectId()	Used to distinguish between value documents.
<i>TimeStamp</i>	ISODate()	Indicates what time the measurement was taken.
<i>TagID</i>	ObjectId()	Used to link value to a specific data tag.
<i>Value</i>	Double	Contains the measurement value.
<i>WasValueInterpolated</i>	Boolean	Used to indicate if the value is an actual measurement, or whether the value was interpolated.
<i>DataQualityAnalysisResult</i>	BSON document	Added to store the results of the data quality analysis for the specific data point.

The *DataQualityAnalysisResult* is an optional field. If a data point is flagged by one of the metric analyses, the results for the data point are stored in the *DataQualityAnalysisResult* field. The *DataQualityAnalysisResult* field is an embedded document. Table 3.6 lists the key-value pairs that can be found in the *DataQualityAnalysis* subdocument. It is important to note that in order to reduce the size of the *value* document, metrics that have not been flagged are removed from the *DataQualityAnalysisResult* field. It is, therefore, possible for a *value* document to not have a *DataQualityAnalysisResult* field.

Table 3.6: DataQualityAnalysisResult subdocument

Key	Value
Static	Boolean
ExceedsLimits	Boolean
Negative	Boolean
Outlier	Boolean

SQL database

Three tables are required in the SQL database for configuration purposes, as shown in Figure 3.3. These tables form part of a larger SQL database that is used by the existing ETL system. The *TagName* table shown in red exists as part of the ETL system. The *TagData* and *ExceedsLimits* tables shown in blue are new tables needed in the SQL database.

- The *TagName* table contains the basic tag details. Each tag in the data store is linked to a tag in the SQL database. The *tn_id* field coincides with the *tag* document *_id* in the data store.

²Insert (an intermediate value or term) into a series by estimating or calculating it from surrounding known values.

- The *TagData* table contains the tag-specific settings such as which metrics to analyse and how frequently the data for the tags must be analysed.
- The *ExceedsLimits* table contains the specified limits for tags if the Exceeds Limit metric is configured for a tag.

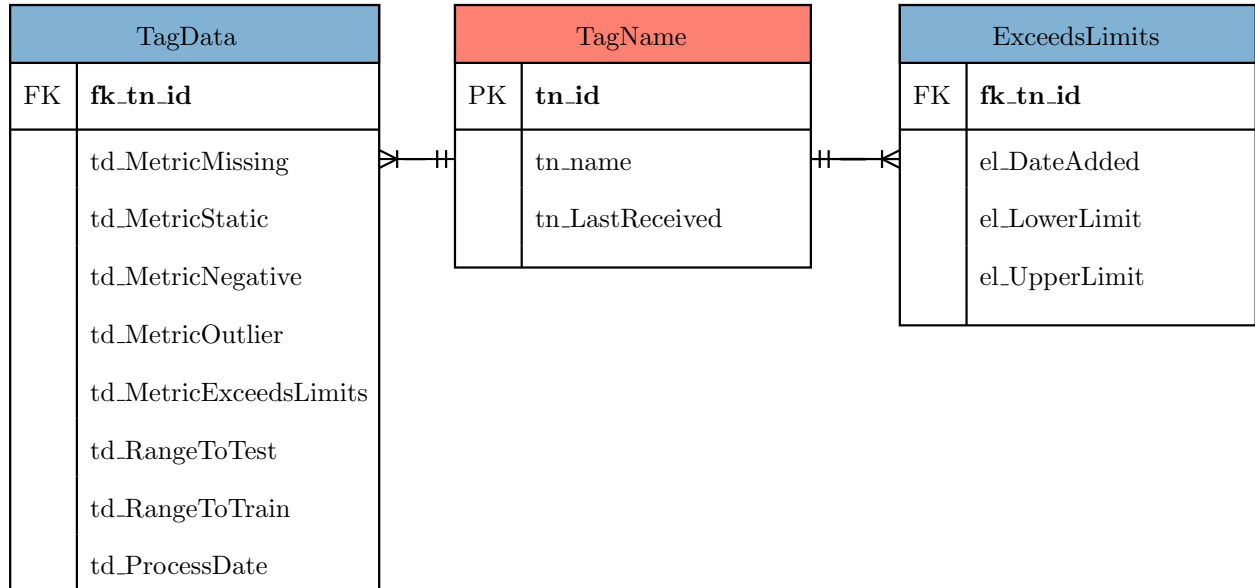


Figure 3.3: Configuration database entity relationship diagram

3.4.3 Data processing

Tag configuration

From the user requirements and specifications given in Section 3.2, a web-based graphical user interface (GUI) is required to configure the tags for analysis. A web-based interface is easy to maintain and gives users access to the latest version of the interface instantly.

The interface allows users to select which tags to analyse and which metrics to run on the specified tags, and gives users access to other metric-related settings. The configuration is saved to the SQL database. Figure 3.4 shows the GUI used to configure tags for analyses. As shown in the image, the user can select to analyse a tag and select which metrics to run on the tag using the checkboxes next to the tags.

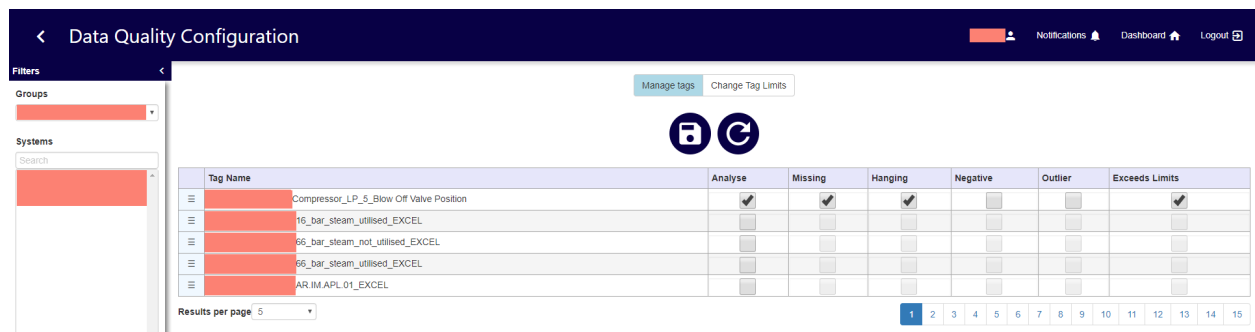


Figure 3.4: Configure tags for analyses

Figure 3.5 shows the user interface used to configure the limits for tags for the Exceeds Limit metric. The user can configure the limits for a tag if the Exceeds Limit metric is selected, as shown in the image.

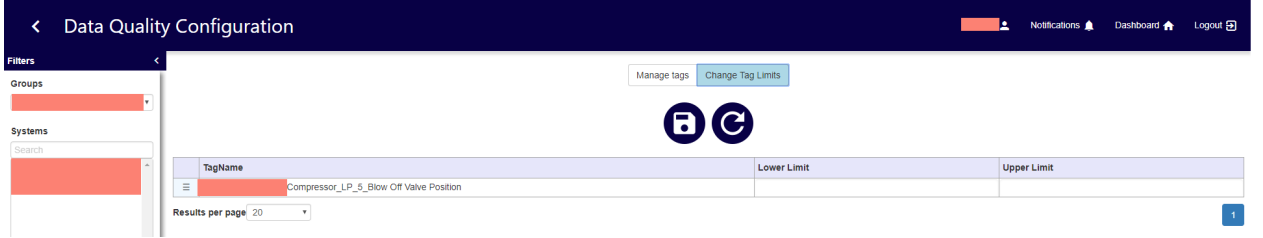


Figure 3.5: Configure limits for tags

Data collection

In order to analyse data quality, the system must know which tags to analyse, which metrics to analyse for which tags, and which data points to analyse. From the SQL database discussed in Section 3.4.2, the system can gather the various settings for each tag that must be analysed. The SQL database provides the IDs of the tags that must be analysed, along with the metrics that must be analysed for each tag and the date range for the data points to analyse.

Using the *tn_LastReceived* field from the *TagName* table, and the *td_ProcessDate* and *td_RangeToTest* fields from the *TagData* table as shown in Figure 3.3, it is possible to determine the start date and end date of the data to analyse. Equation 3.7 shows how the start date is calculated. The last processed date from the *td_ProcessDate* is used as the end date.

$$D_{Start} = D_{Processed} - D_{Testrange} \quad (3.7)$$

where:

D_{Start} = Start date

$D_{Processed}$ = *tn_ProcessDate* from *TagData* field SQL database

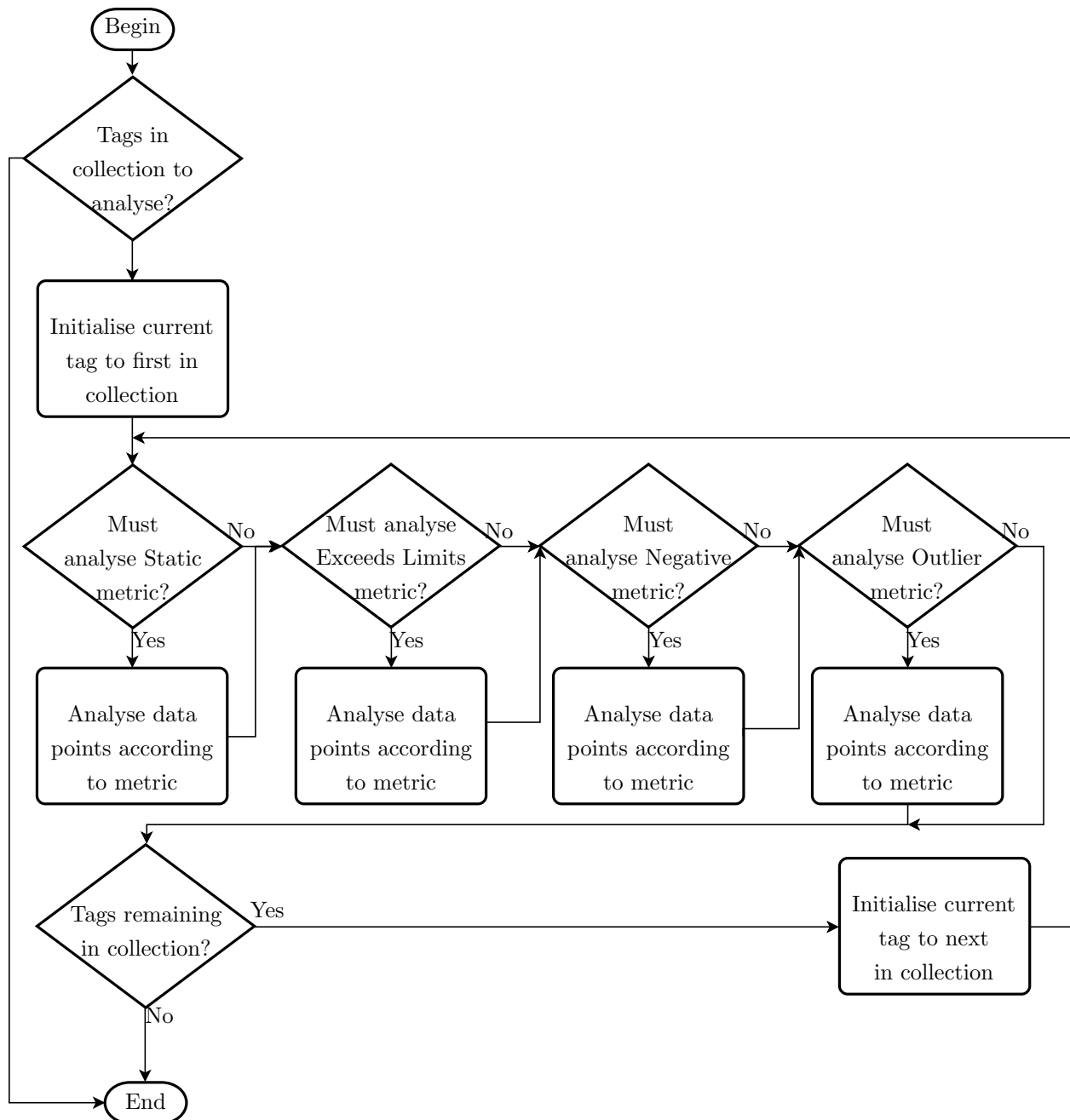
$D_{Testrange}$ = *tn_RangeToTest* from *TagData* field SQL database

Once the start and end dates of the date range to analyse have been determined, the data points in that range can be gathered from the data store as discussed in Section 3.4.2.

Data analysis

Data is analysed using the different metrics described in Section 2.4. The primary analysis process is shown in Figure 3.6. Each tag's data is analysed according to the metrics set in the SQL database.

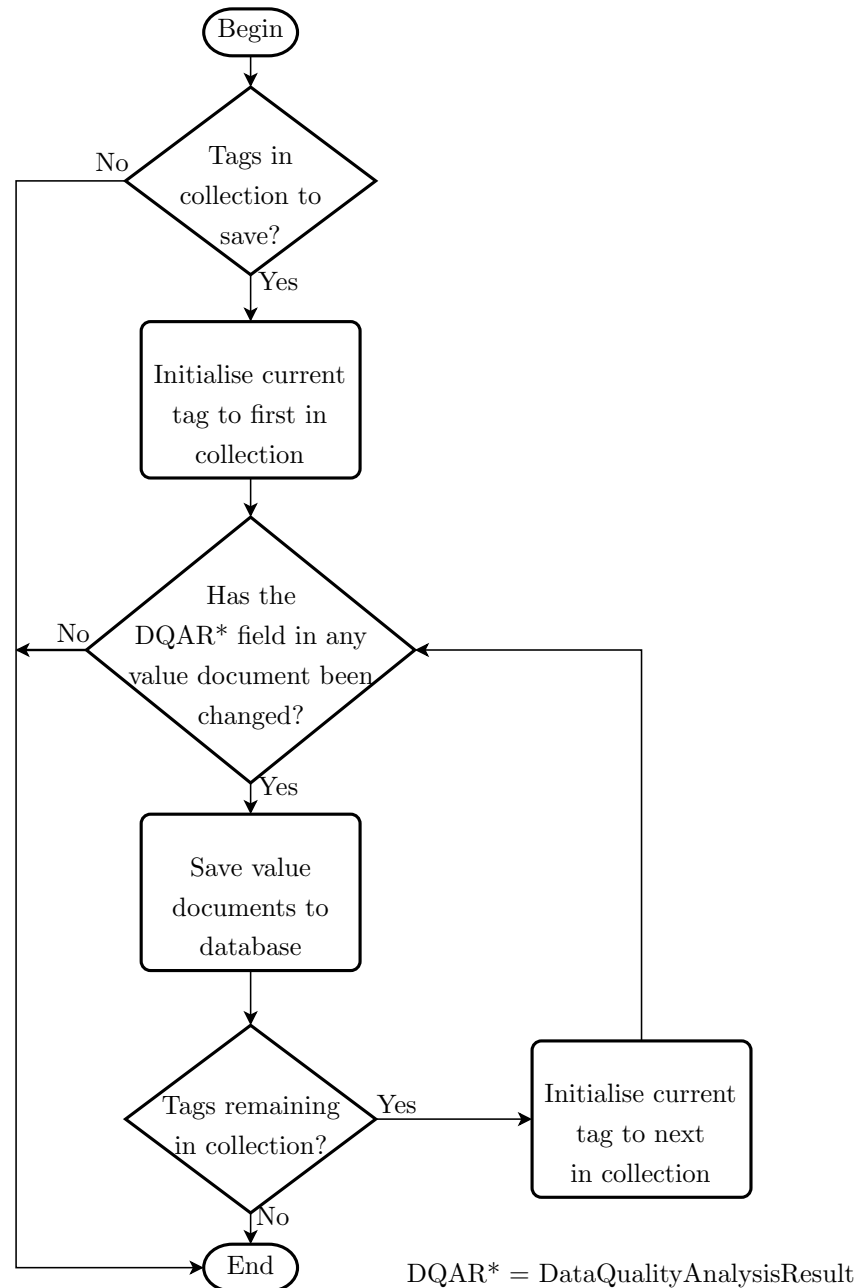
The data errors are identified for each tag's data using the methods described in Section 3.3. The Missing metric is not analysed at this stage. Missing data is data that does not have entries in the data set, making it difficult to analyse at this stage. The Missing metric is calculated as part of the results display stage discussed in Section 3.4.4.

Figure 3.6: *Main data analysis process*

Save results

Once all the data points have been analysed and the relevant flags set, the **value** documents can be updated in the data store. The document-based database was discussed in Section 3.4.2. Only **value** documents that have been modified in terms of the *DataQualityAnalysisResult* flags are updated in the data store.

Figure 3.7 shows the steps used to determine if a **value** document must be updated or not. Each **value** document is analysed to determine if any metric flags have been added or removed. If the **value** document metric flags have not been changed, the **value** document is not updated in the data store. If metric flags have been added or removed, the document is updated in the data store.

Figure 3.7: *Update documents steps*

Scheduling

The analysis of tags can be scheduled using the settings in the SQL database. Using the *tn_LastReceived* field from the *TagName* table, and the *td_ProcessDate* and *td_RangeToTest* fields from the *TagData* table, it is possible to determine if a tag is due to be analysed or not. Figure 3.8 shows the steps to determine whether a tag must be analysed.

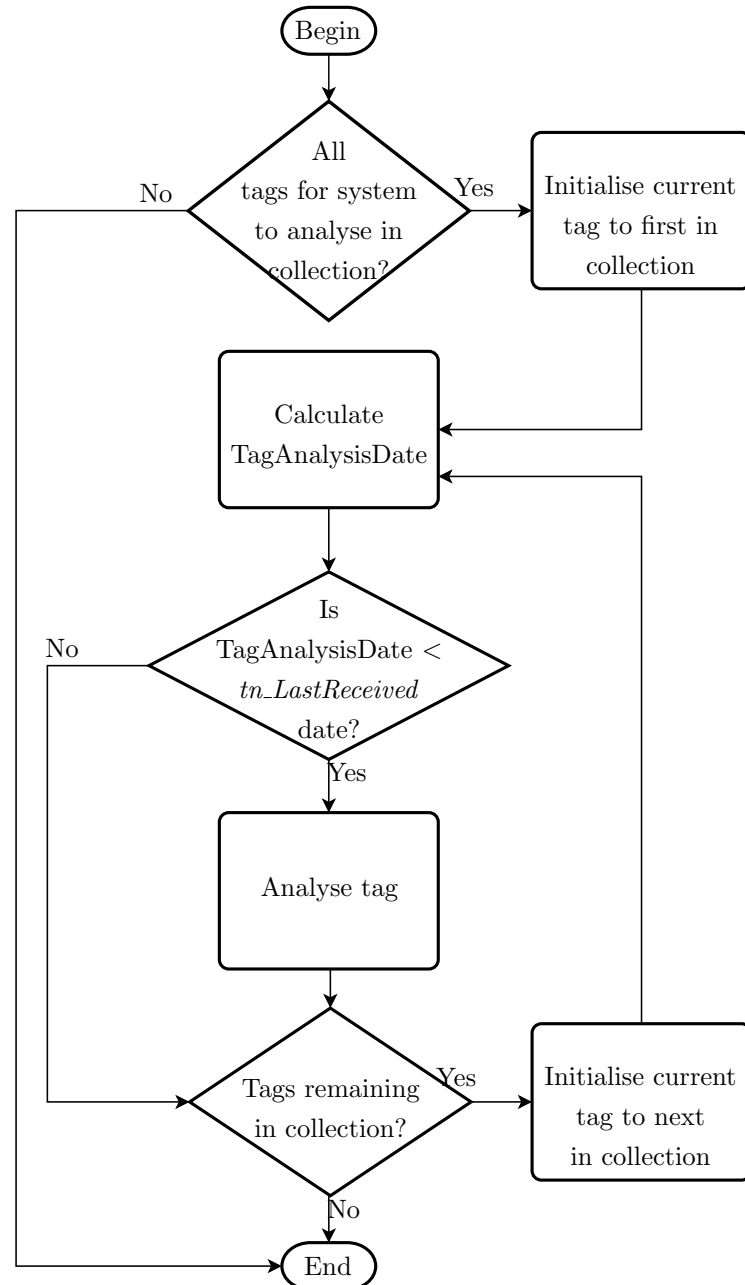


Figure 3.8: Tag analysis scheduling

3.4.4 Data analysis results

The web-based interface allows users to view the results of the analysis. Furthermore, the web-based interface makes it easy to restrict user access to only the projects the user works on. Users do not have access to other projects' tags and the tags' data.

The web-based interface allows users to view the results of the data analysis dynamically. A user can select a date range to view specific data. The web-based application calculates the scores for the tags for the given date range.

The tags are sorted based on the overall data quality score for the tags. Tags with lower data quality scores are sorted higher to make it easier to determine which tags are problematic. Equation 3.6 is used to determine the overall score for the data tags.

The individual metric scores for a tag can also be calculated. Each metric can be calculated as discussed in Section 3.3. The Missing metric can be calculated using Equation 3.1.

The other metrics are calculated using the methods described in Section 3.3:

- Static metric is calculated using Equation 3.2,
- Exceeds Limits is metric calculated using Equation 3.3,
- Negative metric is calculated using Equation 3.4, and
- Outlier metric is calculated using Equation 3.5.

3.5 Verification

3.5.1 Background

In order to verify the system works correctly, a number of tests were performed and the system was compared with the specifications. To configure the test environment, a test group was created. A test system was created within the group, and a number of tags were created and linked to the system. Each tag was used to test a specific metric:

- AllMetrics: Test if the system was able to identify erroneous data for multiple metrics.
- Missing: Test if the system was able to identify missing data points.
- Static: Test if the system was able to identify static data points.
- ExceedsLimits: Test if the system was able to identify which data points exceed the set limits.
- Negative: Test if the system was able to identify negative data values.
- Outlier: Test if the system was able to identify outlier data points.

For each tag, a number of data points were created. Table 3.7 summarises each tag and the data generated for that tag. Each tag's interval was set to 5 minutes.

Table 3.7: Data points for each test tag

Tag name	Metrics to test	Number of data points	Description of data
AllMetrics	All	1440	No values
		1440	1
		1440	Decreasing from -5
		1440	Increasing from 3000
		1	Outlier
		3168	Increasing values
Missing	Missing	2880	No values
		6048	Increasing values
Static	Static	7200	Increasing values
		1728	1
ExceedsLimits	Exceeds Limits	7200	1
		1152	-1
		576	2
Negative	Negative	7200	Increasing values
		1728	Decreasing negative values
Outlier	Outlier	8927	1
		1	Outlier

Table 3.8 summarises the expected results for each tag. These results were used to verify if the system analysed the data correctly and displayed the expected results.

For the AllMetrics tag, problems were created for all the metrics. The total number of erroneous data points was calculated by summing the number of erroneous points generated for a tag. The total number of data points expected was 8928. Equation 3.6 was used to determine the overall score for each tag.

Table 3.8: Expected results for each tag

Tag	Number of faulty data points	Number of data points expected	Overall score (%)
AllMetrics	5761	8928	34.6
Missing	2880	8928	67.7
Static	1728	8928	80.6
ExceedsLimits	1728	8928	80.6
Negative	1728	8928	80.6
Outlier	1	8928	99.9

3.5.2 Analysis results

The results of the analysis are divided into two sections, namely, overview results and detailed results. The overview results display the results for multiple tags to make it easier to identify problematic tags. The detailed results shows the results for a specific tag, making it easier to identify specific problems for a specific tag.

Overview results

The GUI displays the results for the tags. Figure 3.9 shows the overall results for all the tags for the entire date range of the data generated. As can be seen from the image, about 65% of the data for the AllMetrics tag is erroneous, 32% of the data for the Missing tag, and 19% for the rest of the metrics except the Outlier metric. The results displayed in Figure 3.9 are the same as the expected results in Table 3.8. The tags are sorted according to the overall score. It is easy to determine which tags have low quality based on the results displayed in this view.

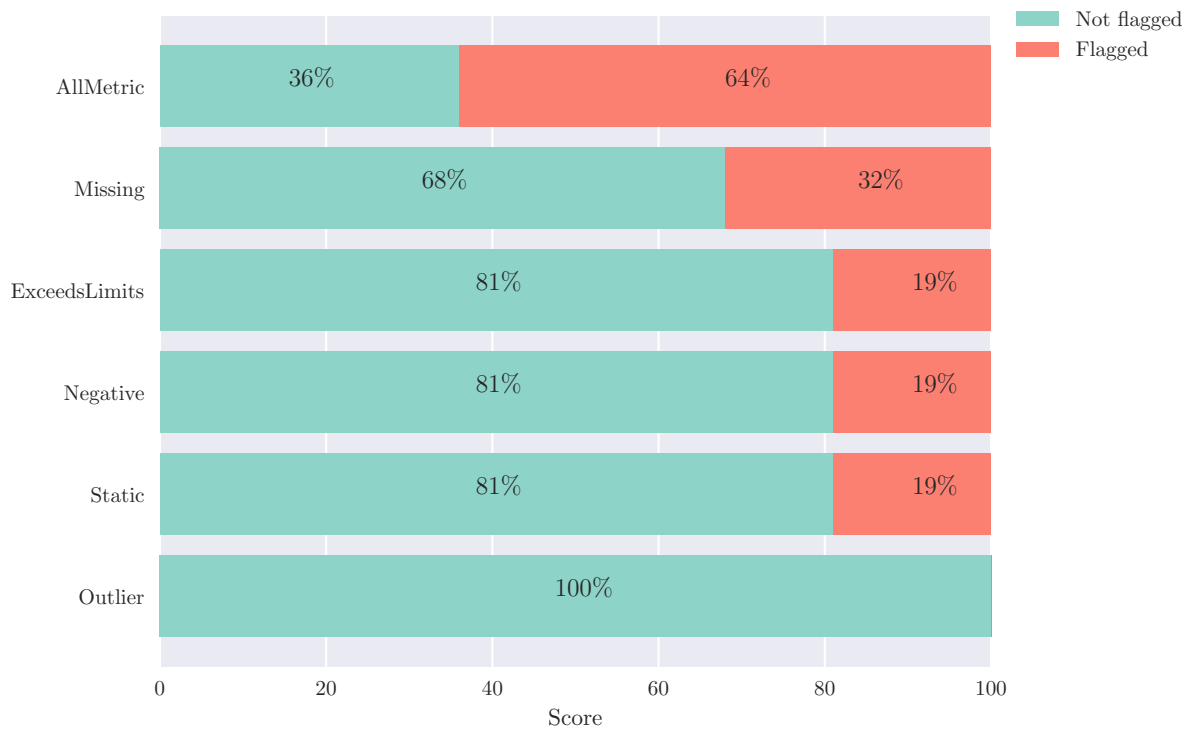
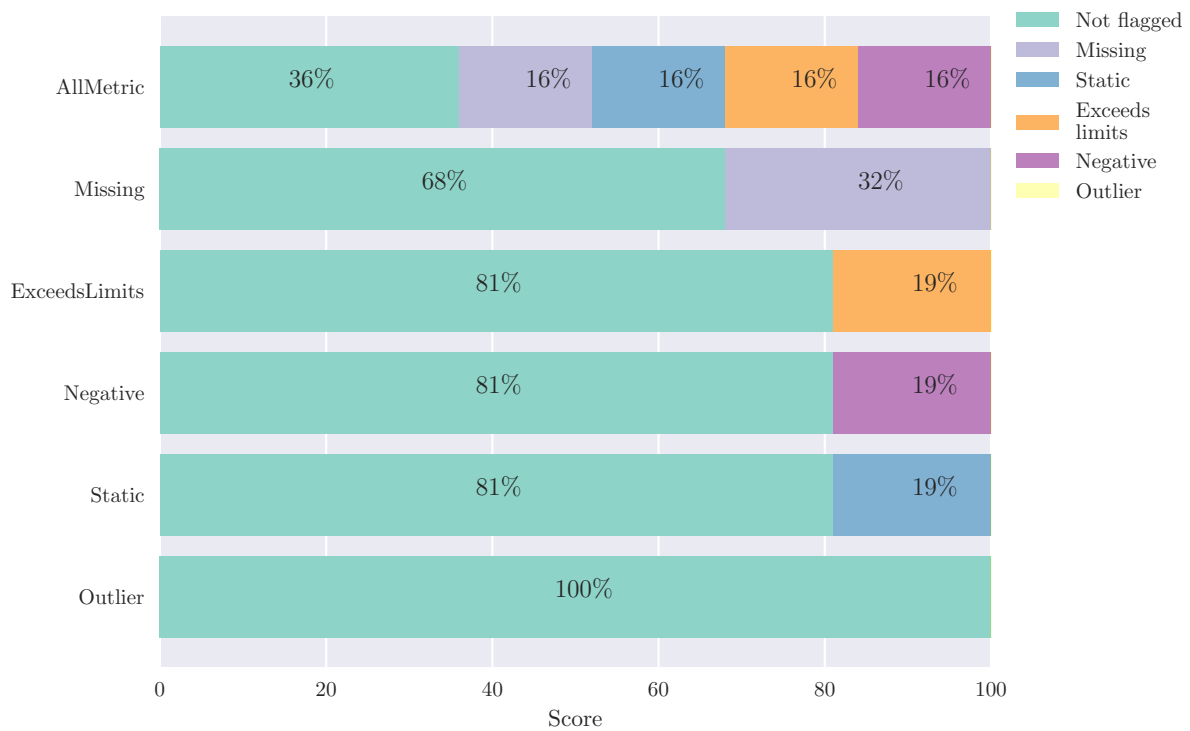


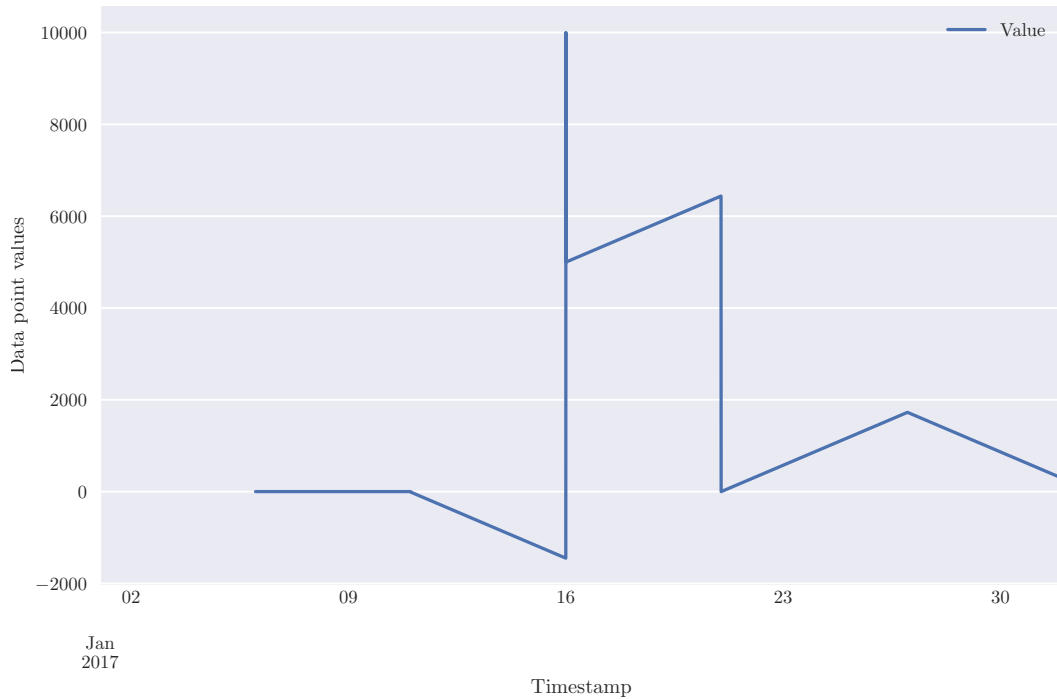
Figure 3.9: *Verification – Overview results*

Figure 3.10 shows the detailed results for the tags. From this view, it is possible to determine what the problems are for each data tag. For instance, the AllMetrics tag has about 16% missing data, 16% static data, 16% negative data and 16% data that exceeds limits. The other tags' faulty scores are completely generated as a result of a specific data problem, whereas, the AllMetrics score is generated from a variety of problems.

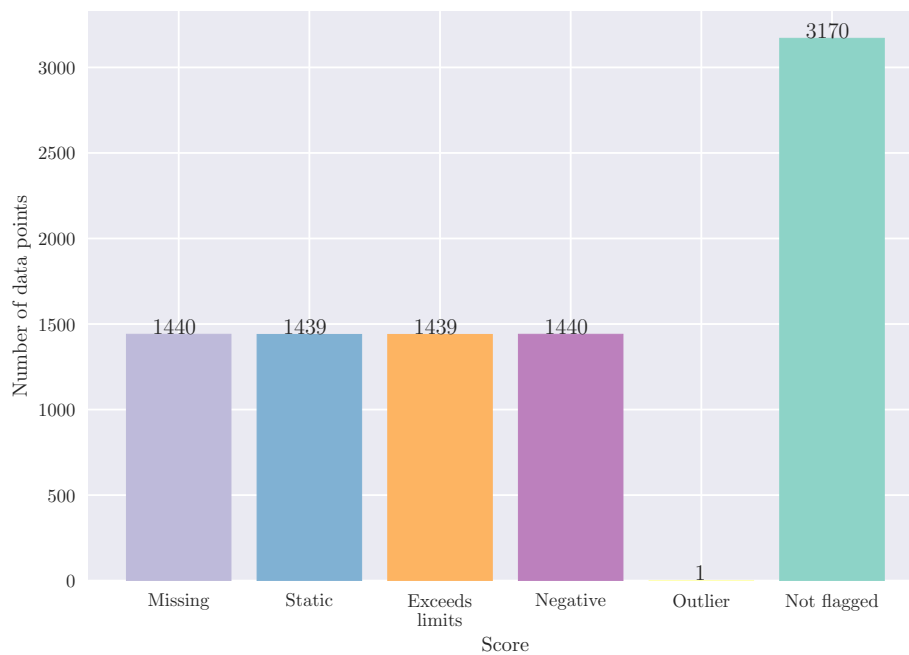
Figure 3.10: *Verification – Detailed results*

Tag detailed results

Results for each tag can also be analysed. Figure 3.11 shows the data plot for the data of the AllMetrics tag. It is possible to zoom into the graph to view a much narrower range of data. The interface also shows the results for each data point. From the graph, it is possible to see which data points are missing (the pink shaded area), which are negative, which exceed limits (limits are shown on the graph), and the potential outliers. Due to the large number of data points graphed, the outlier does not appear on the graph in Figure 3.11.

Figure 3.11: *AllMetrics tag – Data plot*

The bar graph in Figure 3.12 shows the total number of data points flagged by each metric. As can be seen, 1440 data points were flagged as either missing or negative, 1439 data points were flagged as either exceed limits or static, and one data point was flagged as an outlier. Furthermore, 3170 data points were correct. These results coincide with the expected results in Table 3.8.

Figure 3.12: *AllMetrics tag – Data point count*

The pie chart in Figure 3.13 shows the relation between the number of data points flagged for each metric and the number of correct data points. The chart shows the percentage of data points in the date range that has been flagged by each metric, and how many data points have not been flagged by any metric. This makes it easy to determine if the data has a more significant problem regarding a specific metric, which allows the user to take corrective action. A more detailed description of results for the individual metrics can be found in Appendix B.

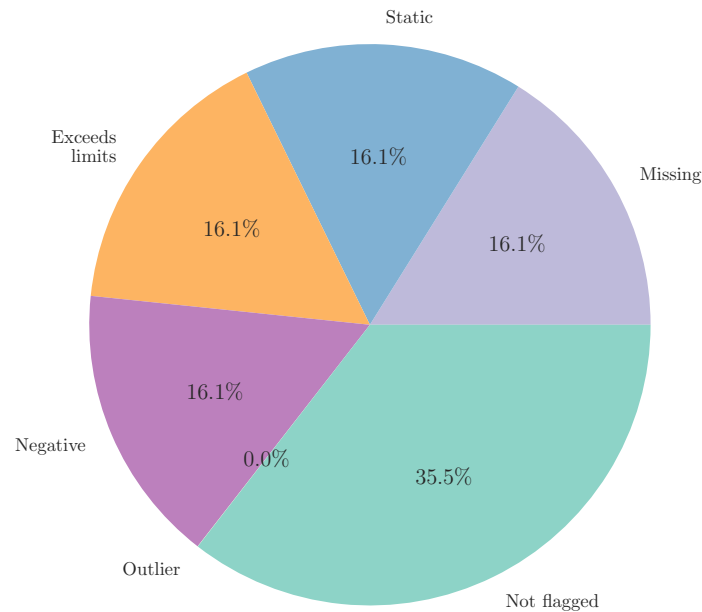


Figure 3.13: *AllMetrics tag – Pie chart*

Summary

Table 3.9 summarises the expected results and the actual results obtained from the system. From the table, it can be seen that the system rounds the overall scores. If the expected results for the overall scores are rounded, the results are the same.

Table 3.9: Results comparison of tags

Tag	Expected results		System results	
	Number of erroneous data points	Overall score (%)	Number of erroneous data points	Overall score (%)
AllMetrics	5761	35.5%	5758	36%
Missing	2880	67.7%	2880	68%
Static	1728	80.6%	1727	81%
ExceedsLimits	1728	80.6%	1728	81%
Negative	1728	80.6%	1728	81%
Outlier	1	100%	5	100%

Table 3.9 shows that the expected results and system results differ slightly for the data point counts. For the AllMetrics tag there is a three data point discrepancy. For the Static tag, there is a one data point discrepancy. A discrepancy is also found for the Outlier tag. The results for the ExceedsLimits tag and the Negative tag are the same as the expected results.

The discrepancy for the static analysis exists because the first data point in a series of hanging data points is not considered to be a static data point. The discrepancy for the outlier analysis exists due to the methods used to identify outliers. The support vector machine (SVM) and nearest neighbours regression methods identify the four data points surrounding the actual outlier as outliers as well.

3.5.3 Verification summary

In order to verify that the system is correct, we must determine if the system satisfies the requirements and specifications as described in Section 3.2.1 and Section 3.2.2. Table 3.10 summarises the requirements and the specification for the system, and indicates whether the system meets the requirements.

Table 3.10: System requirements verification

System requirement	Description	Design description	Requirement met
Missing metric	Determines how many data points are missing for a given tag for a given date range.	Section 3.4.4	✓
Static metric	Determines how many data points are static for a given tag for a given date range.	Section 3.4.3	✓
Exceeds Limits metric	Determines how many data points exceed the set limits for a given tag for a given date range.	Section 3.4.3	✓
Negative metric	Determines how many data points are negative for a given tag for a given date range.	Section 3.4.3	✓
Outlier metric	Determines how many data points are outliers for a given tag for a given date range.	Section 3.4.3	✓
Overall score calculation	Determines the overall score for a tag based on the metric calculation results.	Section 3.4.4	✓
Set up tag configuration	Enables user to configure which metrics to run on a given tag.	Section 3.4.3	✓
Manage tag configuration	Enables user to manage the configuration for tags.	Section 3.4.3	✓

Table 3.10: System requirements verification continued

System requirement	Description	Design description	Requirement met
Data retrieval from data warehouse	Retrieve data for a tag for a specified date range from the data warehouse.	Section 3.4.3	✓
Save results	Save the analysis results of each individual data point.	Section 3.4.3	✓
Display results	Display the quality analysis results to the user.	Section 3.4.3	✓
Use existing data store	Use the existing document-based database.	Section 3.4.2	✓
Use existing SQL database	Use the existing SQL database to store tag configurations.	Section 3.4.2	✓

3.6 Conclusion

Section 3.2 summarised the user requirements and specifications for the system. One of the most important specifications is the development of a method to quantify industrial data quality. Thereafter, a system is required to use the method, to analyse industrial data and to quantify the quality thereof.

A method was developed in Section 3.3 to analyse data quality according to four dimensions, namely, data completeness, availability, accuracy and validity.

Each of the data quality dimensions can be quantified using a simple ratio. To quantify completeness and availability, the ratio between missing data points in a data set and the expected number of data points in a data set can be determined.

To quantify accuracy and validity, a combination of metrics can be used as required. Four metrics were defined to analyse accuracy and validity. The metrics are: Static metric, Exceeds Limits metric, Negative metric and Outlier metric.

The system design was discussed in Section 3.4. An overview of the system's interactions with the existing ETL system was provided. The database designs were discussed. The different system processes such as data collection, data analysis, results storage and display were discussed.

In Section 3.5 a test group, a test system and test tags were created to verify that the method and system work as expected. Data was generated for each tag with a specific problem. The tags were analysed. The results showed that the system is able to identify the problematic data as expected.

Chapter 4

Implementation and results



4.1 Introduction

The system developed in Chapter 3 was implemented and deployed at an industrial company. The company has clients in the South African mining and industrial sectors. The company collects and processes client data to deliver various services. The gathered data is organised as tags. A tag is a collection of related data (as discussed in Section 3.4.2). Each tag can have numerous measurements. A tag can also contain tag metadata, which describes the characteristics of the tag. The tag metadata and tag data are stored in a document-based database. The measurement values for the tag are also saved in the database (as discussed in Section 3.4.2).

The data quality analysis system uses the relational database used by the existing ETL system (as discussed in Section 3.4.2). A schema is added to the existing database for the data quality analysis settings. The data quality analysis system also makes use of the existing data store to access data and store the results of the analysis.

The system is validated using condition monitoring input data for a mining site as a case study. The mining site is part of the operations of a large gold mining company in South Africa. Further data quality investigations are performed using data from the condition monitoring system implemented on the entire mining company's operations.

4.2 Validation: Case study

4.2.1 Data management system

The industrial company has a data management system in place. Figure 4.1 shows the entire data life cycle regarding the industrial company [53]. Each stage of the data management process is discussed below.

Measurement equipment (1): Data is created by measurement equipment [53]. Different measurement equipment is used on industrial sites to measure parameters such as temperature, vibration and flow.

Programmable logic controllers (PLCs) (2): The PLCs control the equipment installed on-site. The PLCs receive feedback signals from the various sensors. Control signals are sent to the PLCs from the supervisory control and data acquisition (SCADA) systems [53].

SCADA (3): A SCADA system is used to monitor and control the PLCs. An open platform communications (OPC) server is hosted on the SCADA [53].

OPC (4): An OPC server is hosted on the SCADA systems. This allows the industrial company's control system to establish an OPC connection with the SCADA system and access data from the various industrial systems [53].

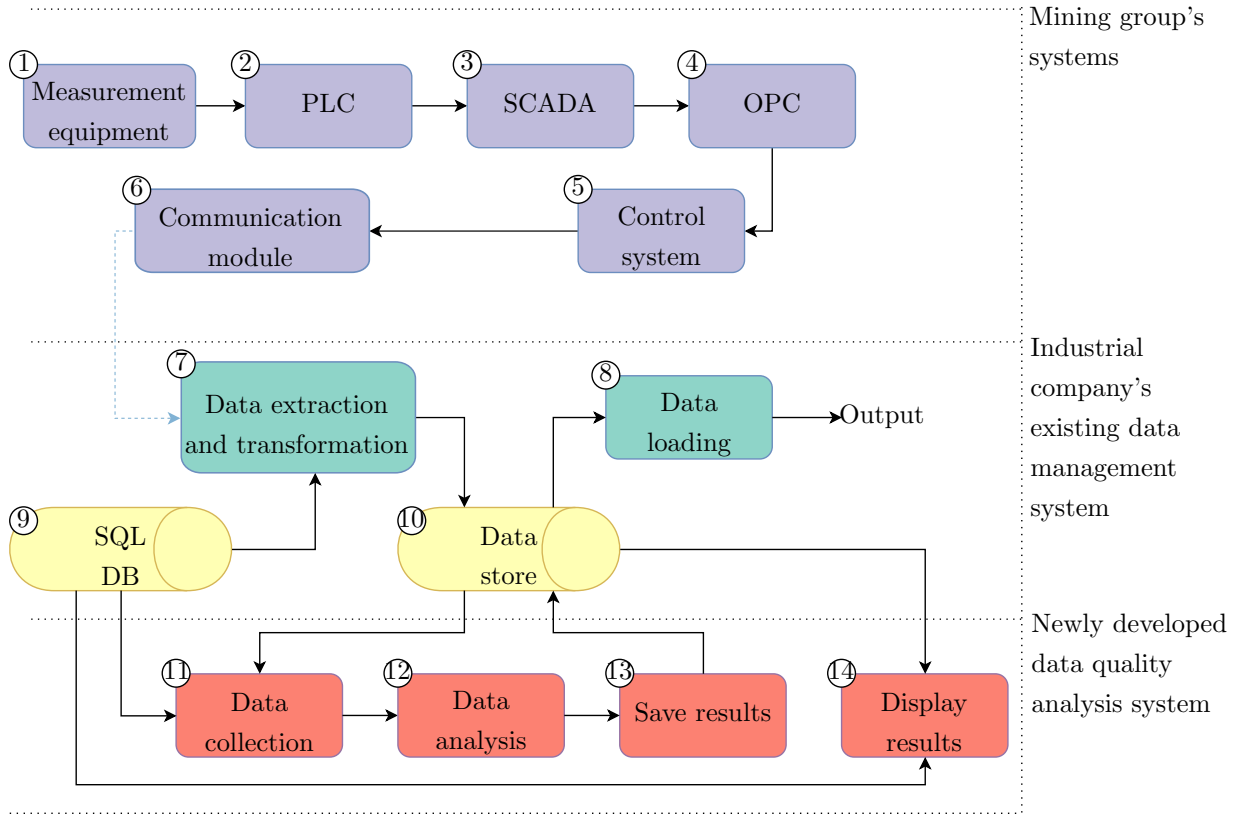


Figure 4.1: Complete system overview

Control system (5): As mentioned above, the industrial company's control system establishes an OPC connection with the client's SCADA systems. The control system uses this connection to receive data from the various industrial systems. The control system is able to establish OPC connections with various types of SCADA system [53].

Communication module (6): This module is an independent application that sends the data logged by the industrial company's control system to a hosted web server. The data is compressed and sent via email [53].

Data extraction and transformation (7): Data arrives in various formats at the extract and transform phase. The system processes the incoming data and stores it in a data store.

Data loading (8): During this phase, data is loaded from the data store and sent to other processes as required. The loaded data can be used in various outputs such as reports, condition monitoring analyses or simulations.

SQL database (DB) (9): The SQL database design was discussed in Section 3.4.2. The SQL database is used to store configurations for the data quality analysis.

Data store (10): The data store design was discussed in Section 3.4.2. The data store is used to store tag-specific information as well as the data for each tag.

Data collection (11): As discussed in Section 3.4 and Section 3.4.3, the system determines which tags must be analysed and which metrics must be analysed for each tag. The system also determines the date range for which data must be analysed. Data is collected from the data store for analysis.

Data analysis (12): The system analyses the data for each tag according to the metrics that must be analysed for each tag (as discussed in Section 3.4 and Section 3.4.3).

Save results (13): The results of the analysis for each data point is saved to the data store (as discussed in Section 3.4 and Section 3.4.3).

Display results (14): As discussed in Section 3.4 and Section 3.4.4, a web-based interface is used to display the results of the analysis.

4.2.2 Mine A

One of the industrial company's clients is a large gold mining company with numerous mining sites in South Africa. The company implemented a condition monitoring system on various of the mining group's sites [15]. Figure 4.2 shows the breakdown of the equipment monitored by the condition monitoring system on some of the mining group's sites.

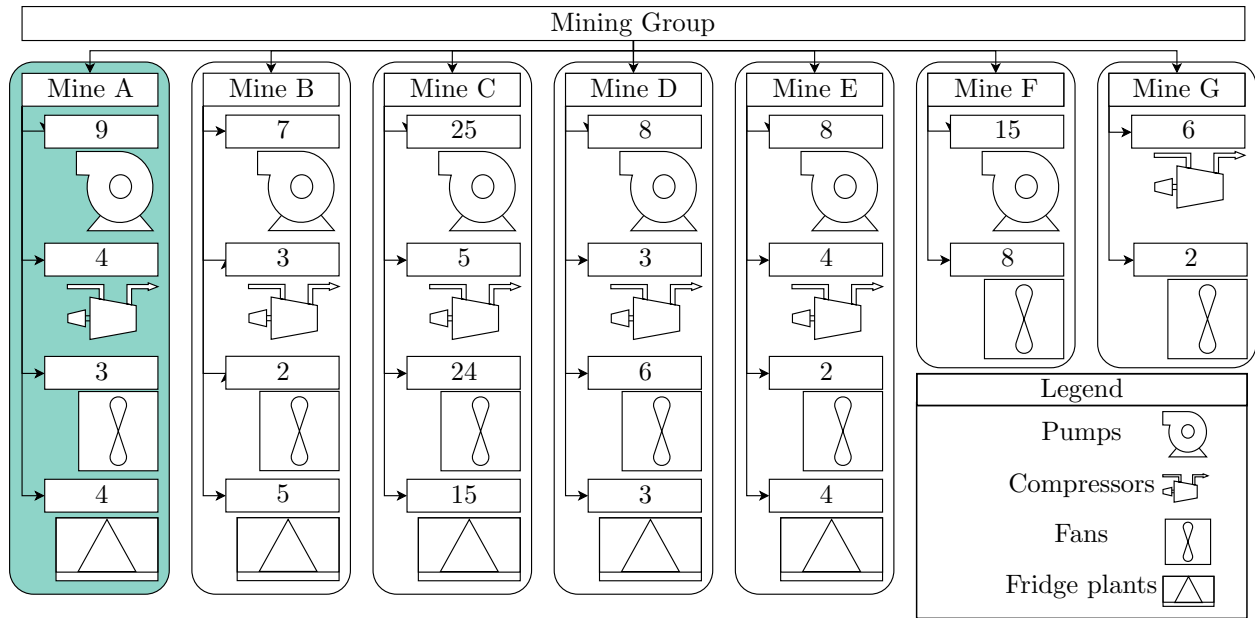


Figure 4.2: Mine sites overview

The mine sites have four major mining systems: compressed air, ventilation, water reticulation and cooling. Data loggers were configured for each of these systems as part of the condition monitoring based system implemented on the mining site [15]. The data loggers were used to log the required data from the four major mining systems.

The same input data used by the condition monitoring system from Mine A was used as a case study to validate the system. The input data for the condition monitoring system consisted of various measurements for different components on the mining site. Table 4.1 shows an overview of the condition monitoring system implementation on Mine A. The table summarises the number of components on the mining site, the number of tags and the corresponding number of data points analysed for the tags between 2018-06-01 00:00:00 and 2018-06-30 23:59:59.

Table 4.1: Mine A – Overview of condition monitoring implementation

Subsystem	Number of subsystems	Number of tags	Data points
Pumps	9	87	125 280
Compressors	4	64	92 160
Fans	3	21	30 240
Fridge plants	4	54	77 760

A data sample size of one month was chosen to ensure that enough data was analysed. As can be seen in Table 4.1, a large number of data points were analysed in the month for Mine A. A larger sample size would dilute the results, and a smaller sample size could cause the loss of some interesting patterns as shown in Section 4.3.6.

Table 4.2 lists the different measurements used in the evaluation. The tags for the measurements were configured where possible for analysis. Some measurements were not available for certain components. For instance, the winding temperature was only measured on some motors.

Table 4.2: Mine A – Input tag parameters

Tags	Pumps	Compressors	Fans	Fridge plants
Running status	X	X	X	X
Motor DE ¹ vibration	X	X	X	X
Motor NDE ² vibration	X	X	X	X
Motor DE temperature	X	X	X	X
Motor NDE temperature	X	X	X	X
Pump DE vibration	X			
Pump NDE vibration	X			
Pump DE temperature	X			
Pump NDE temperature	X			
Compressor DE vibration		X		X
Compressor NDE vibration		X		X
Compressor DE temperature		X		
Compressor NDE temperature		X		
Fan DE temperature			X	
Fan NDE temperature			X	
Cooling duty				X
Evaporator flow				X

¹DE – Drive end²NDE – Non-drive end

Table 4.2: Mine A – Input tag parameters continued

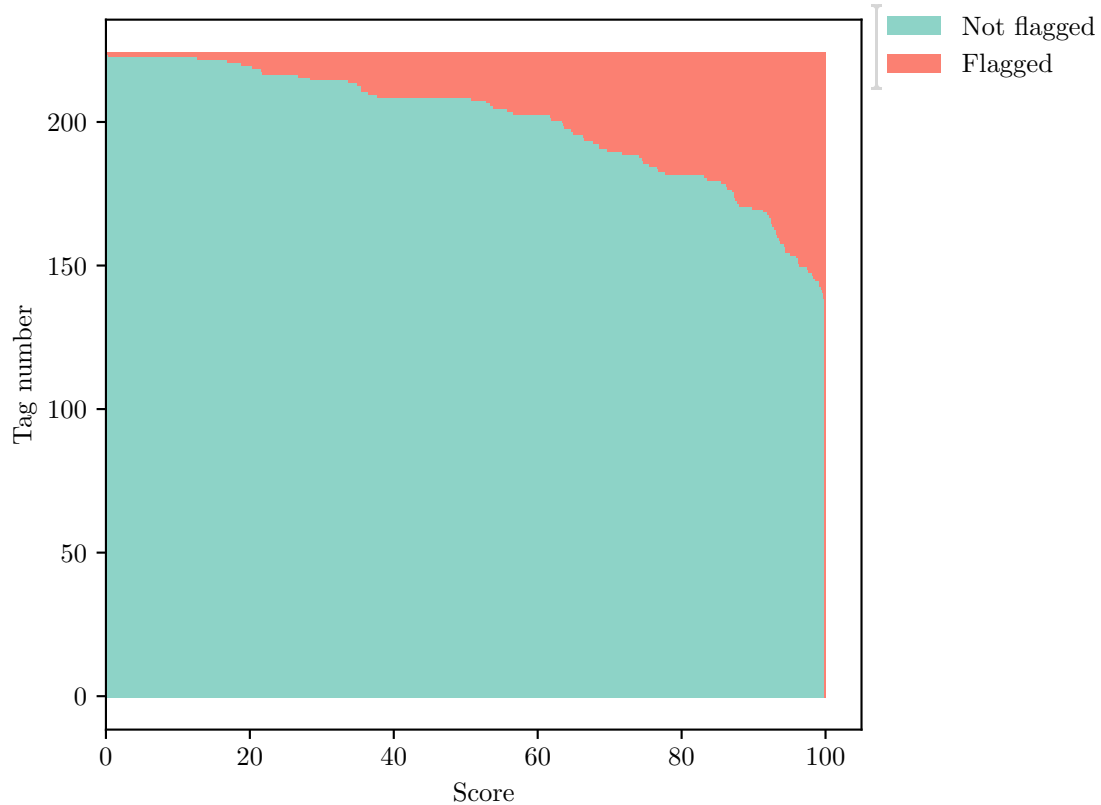
Tags	Pumps	Compressors	Fans	Fridge plants
Evaporator temperature out				X
Evaporator temperature in				X
Condenser flow				X
Condenser temperature out				X
Condenser temperature in				X
Motor power	X			
Winding temperature	X			
Gearbox vibration		X		
Gearbox DE temperature		X		
Gearbox NDE temperature		X		
Gearbox thrust DE temperature		X		
Gearbox thrust NDE temperature		X		
Gearbox pinion DE temperature		X		
Gearbox pinion NDE temperature		X		

All the tags were configured to be analysed using the Missing metric, Negative metric and Outlier metric. All the tags were configured to be analysed using the Static metric, except the running status tags for the different components. Only tags for which there were operational limits were configured to be analysed using the Exceeds Limits metric.

4.3 Results

4.3.1 Overview results

All the tags mentioned in Table 4.2 for the four subsystems for Mine A were analysed. In total, 226 tags were analysed for Mine A. All the data points between 2018-06-01 00:00:00 and 2018-06-30 23:59:59 were analysed for each tag. Figure 4.3 shows the overview results for all the tags for Mine A. The percentage of flagged data points is shown in red and the percentage of not-flagged data points is shown in green. The tags are sorted according to the calculated scores. Tags with lower scores are sorted higher in the list to make it easier to identify tags with poor data quality.

Figure 4.3: *Mine A – Overview results*

The overview results shown that it is straightforward to identify tags with poor data quality scores. From the Figure 4.3, it can be seen that most of the tags have good data quality scores. Some tags, however, have very low scores and require further investigation.

Figure 4.4 shows the overview results of the ten lowest scoring tags. From the graph, it can be seen that at least four of these tags have data quality scores below 20%.

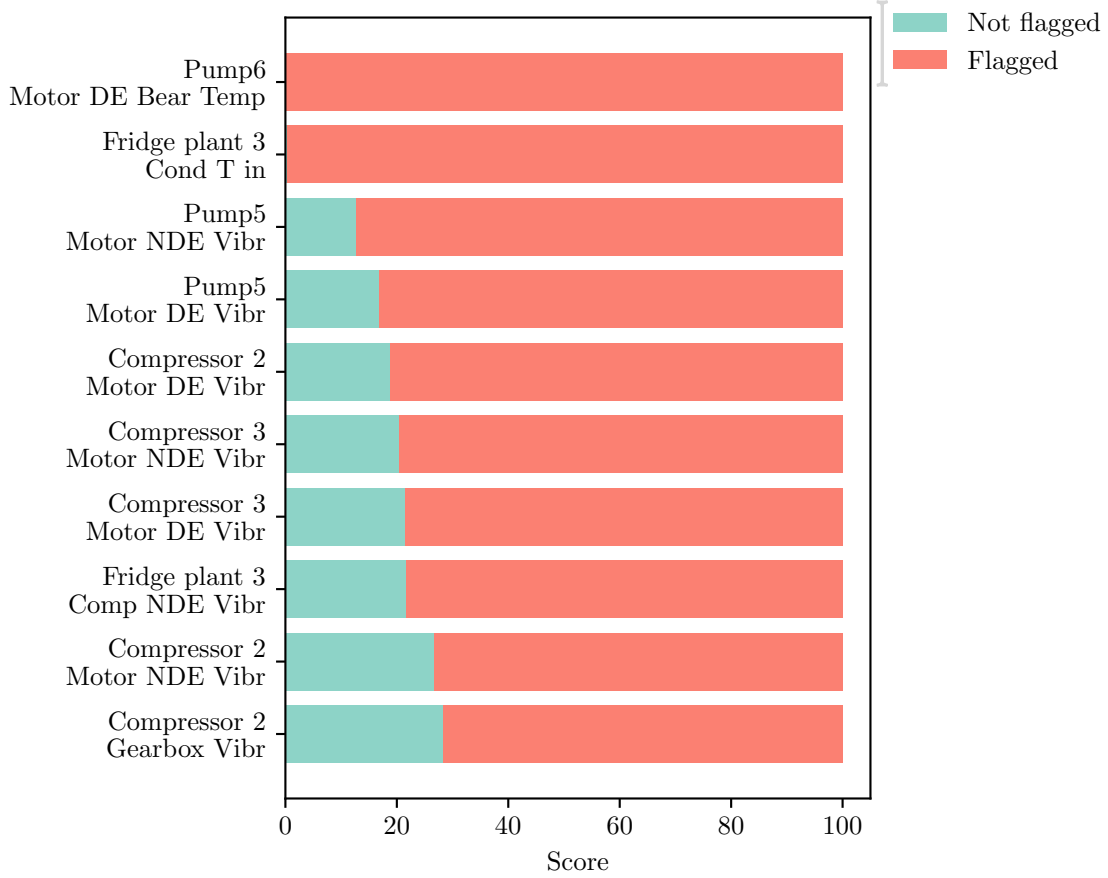


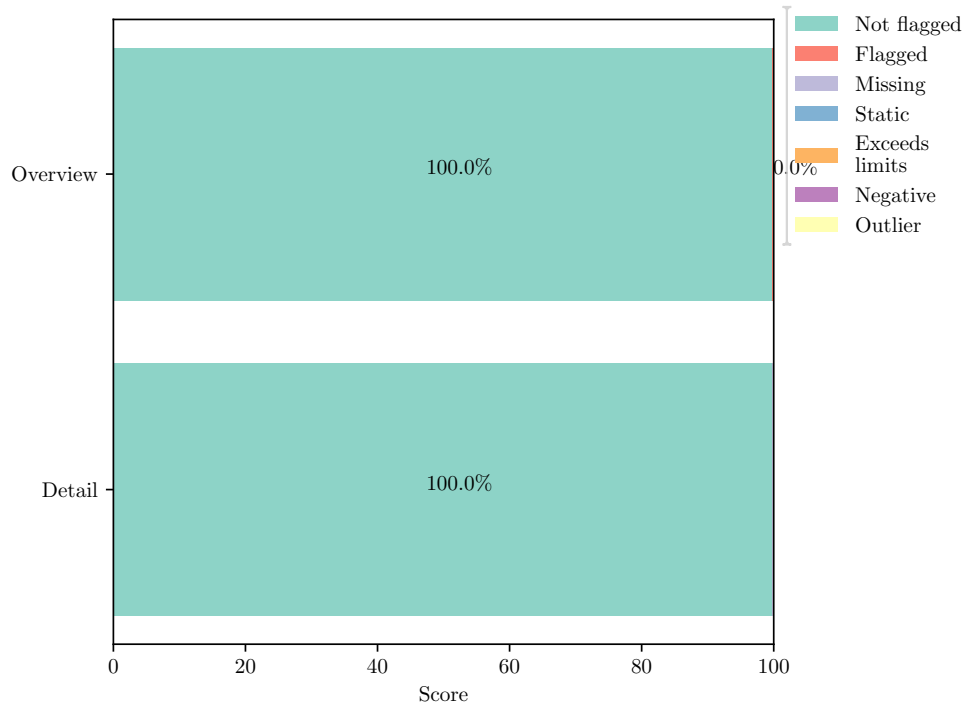
Figure 4.4: Mine A – Overview results of ten worst tags

The following sections discuss the results for tags with specific problems as identified by the different metrics. The tags are chosen at random to highlight the different problems that can arise in data.

4.3.2 Missing results

This section discusses the results for the Missing metric. The *compressor 2 gearbox pinion non-drive end bearing temperature* tag is used to show the system is able to identify missing data.

Figure 4.5 shows the overview and detailed data quality analysis results for the tag. Due to the very high score for data quality, it is difficult to determine from the graphs in Figure 4.5 if any errors have been identified in the data.

Figure 4.5: *Missing – Overview and detailed results*

Further analysis of the result for the tag reveals that three data points were missing as shown in Figure 4.6. The graph shows how many data points were flagged as erroneous by each metric. From the graph it is also possible to see that 1437 data points were not flagged.

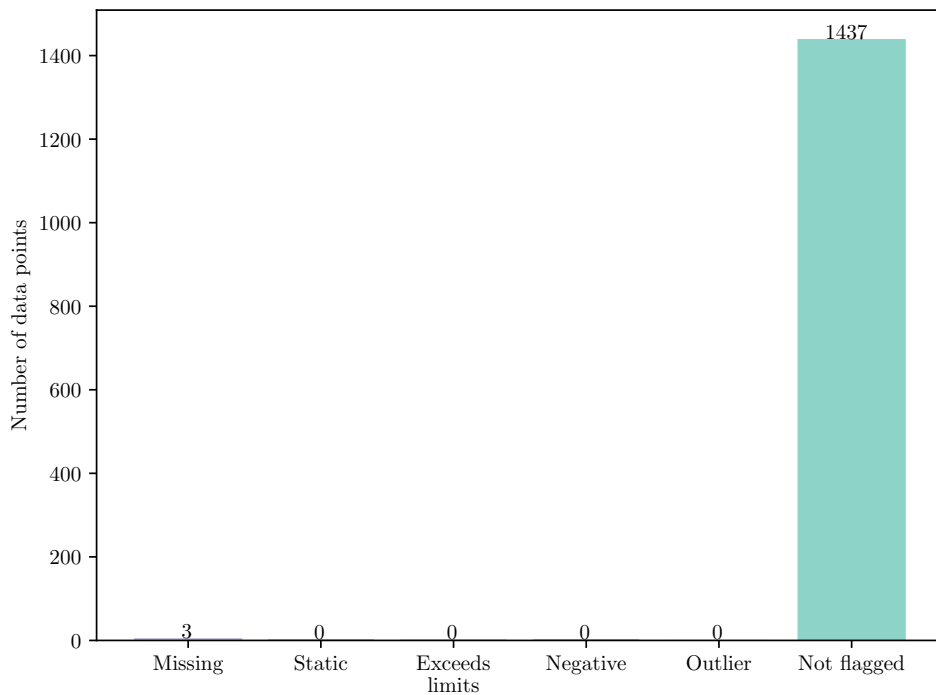
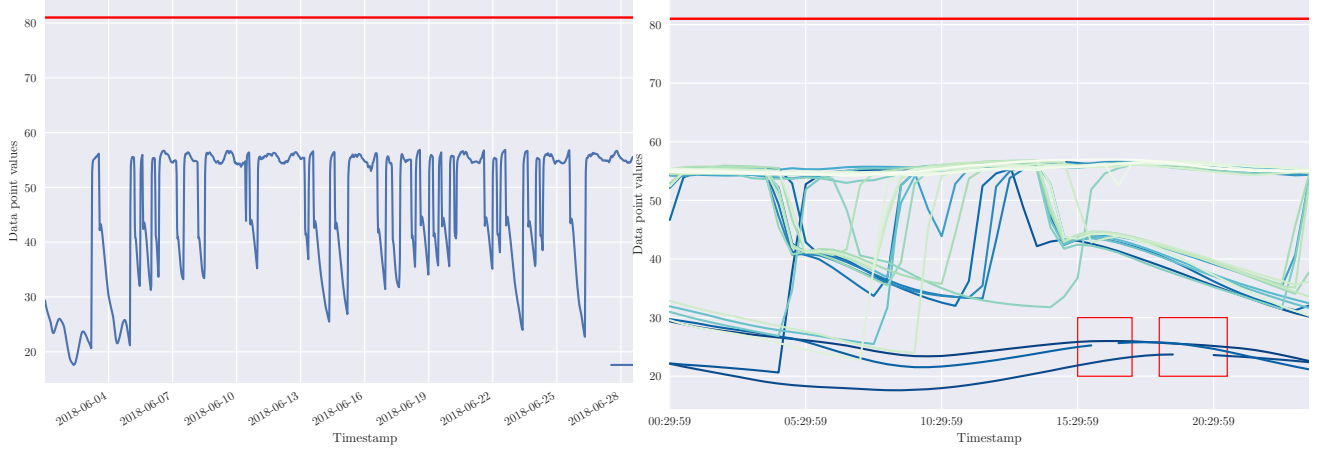
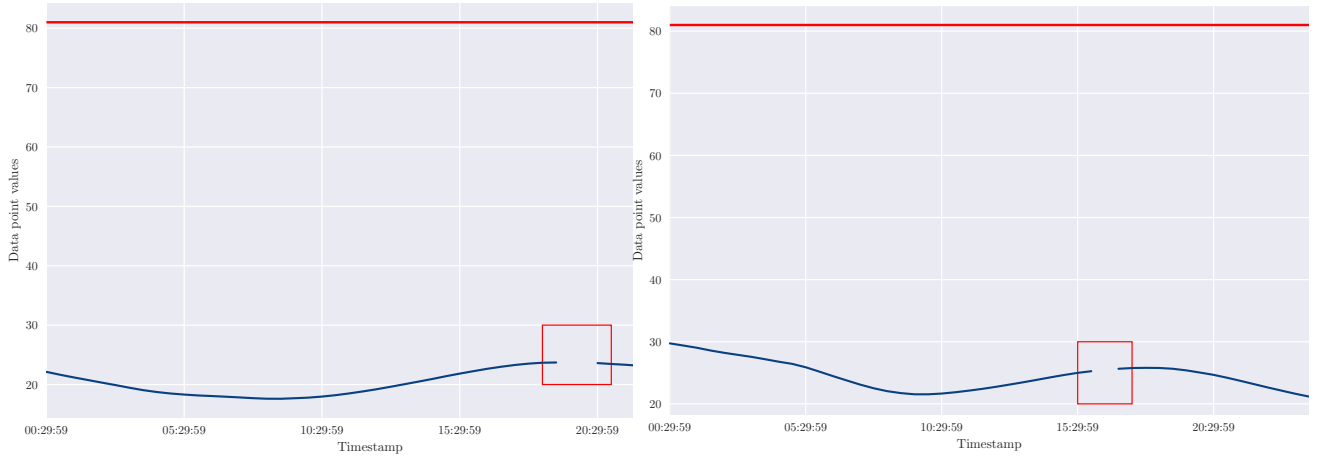
Figure 4.6: *Missing – Data point count*

Figure 4.7a shows the data plot for the tag for the entire time span. From the graph, it is difficult to identify any missing data points. Figure 4.7b shows the same data plotted in daily increments. Each line on the graph represents one day's data. This graph shows that there was missing data on two different days as shown by the red blocks on the graph.

(a) *Sequential*(b) *Daily*Figure 4.7: *Compressor 2 gearbox pinion non-drive end temperature bearing data plots*

The graphs in Figure 4.8 show the data plots for the days identified with missing data. From these graphs it is easy to identify where data is missing (highlighted by the red blocks on the graph). This highlights that data quality is a continuous problem that needs to be analysed and monitored regularly to identify and correct problems quickly.

(a) *2018-06-02*(b) *2018-06-04*Figure 4.8: *Compressor 2 gearbox pinion non-drive end temperature bearing data plots – Specific dates*

The same data points were missing for each tag analysed for Mine A. This indicates that the data was not received by the ETL system; therefore, no data points for the specific time periods exist in the database.

4.3.3 Exceeds Limits results

In this section, the results for the *compressor 3 gearbox pinion³ drive end bearing temperature* is discussed. The data quality analysis results are shown in Figure 4.9 and Figure 4.10. The overview results in Figure 4.9 show that roughly 10% of the data was erroneous. The detailed results in Figure 4.9 show that erroneous data mostly comprised data points that exceeded limits.

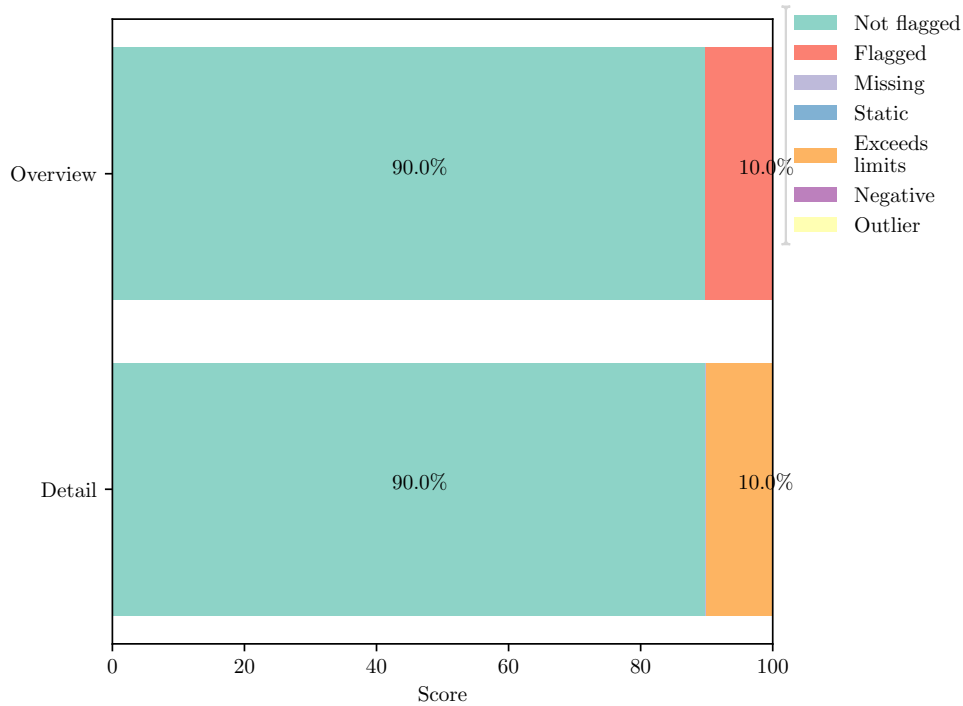


Figure 4.9: *Exceeds limits – Overview and detailed results*

The number of data points that satisfied the various metric conditions in the time frame are shown in Figure 4.10. From the chart it can be seen that three data points were missing, 145 data points exceeded limits, and 1292 data points were not flagged.

³A small cogwheel or spindle engaging with a large cogwheel.

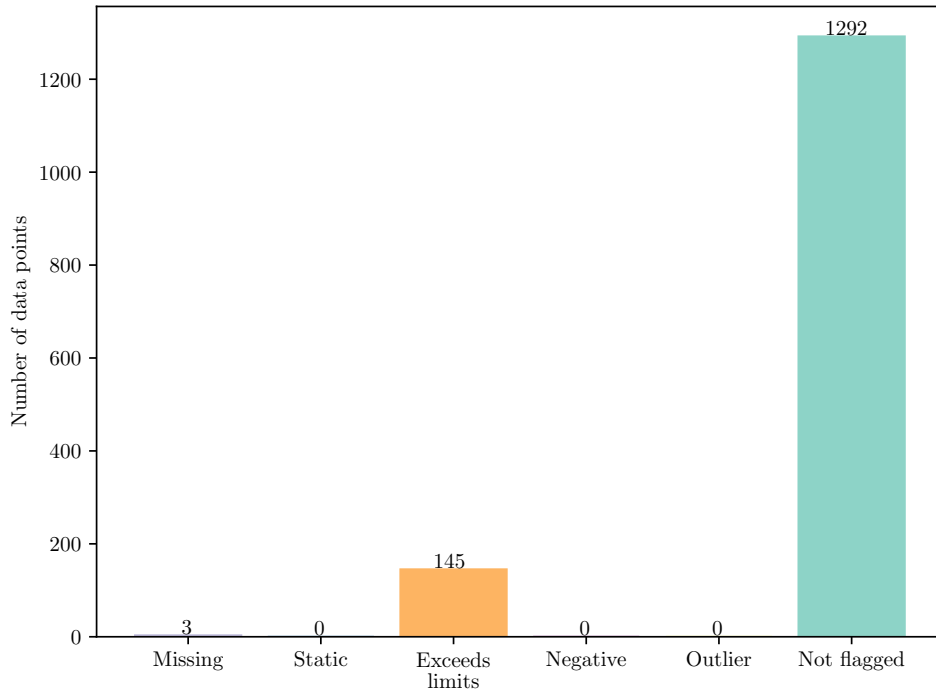
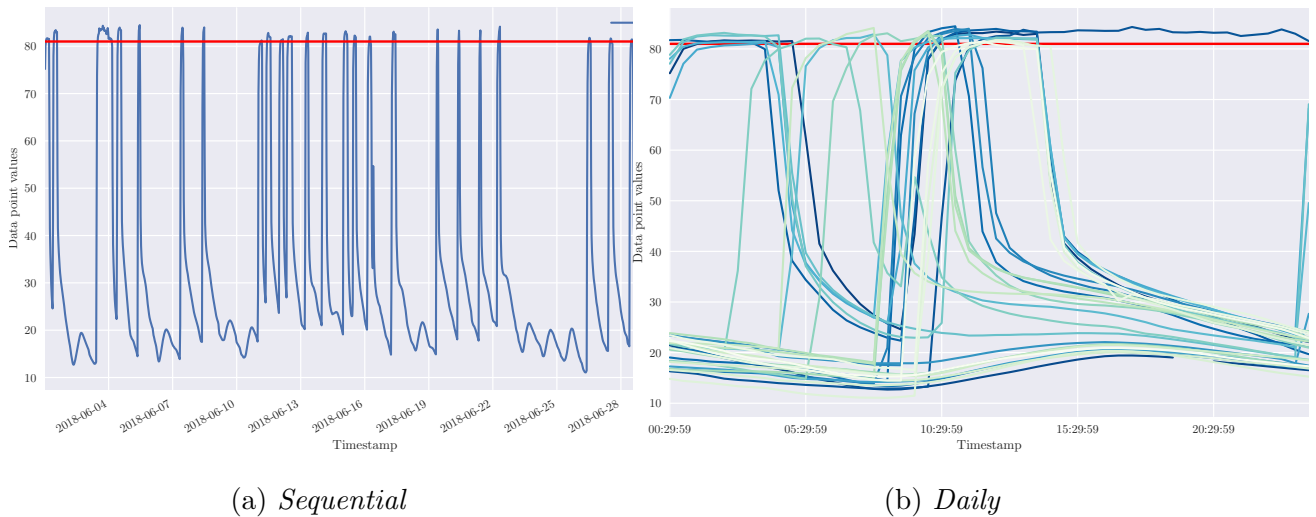
Figure 4.10: *Exceeds limits – Data point count*

Figure 4.11 shows the data plots for the tag. The red horizontal line on the graphs represents the limits configured for the tag. Figure 4.11a shows the data plotted sequentially. This graph shows that the data exceeded the specified limits frequently in the time span. Figure 4.11b shows the data plotted in daily increments. Each line represents one day's values for the tag. This graph shows that the data exceeded the specified limits almost daily.

Figure 4.11: *Compressor 3 gearbox pinion drive-end bearing temperature data plots*

The daily graph in Figure 4.11b shown that the daily data profiles for the tag followed a similar pattern.. It appears as if the compressor switches on, runs for some time above the limit and then switches off. From the profile of the data it is likely that the compressor tripped each time when it switched off. This indicated that the compressor required urgent maintenance.

4.3.4 Static results

In this section, the *compressor 2 non-drive end vibration* tag results are analysed. Figure 4.12 shows the overview and detailed data quality analysis results for the tag. The overview results show that more than 60% of the data was marked as erroneous. The detailed results show that almost all the erroneous data was due to static data.

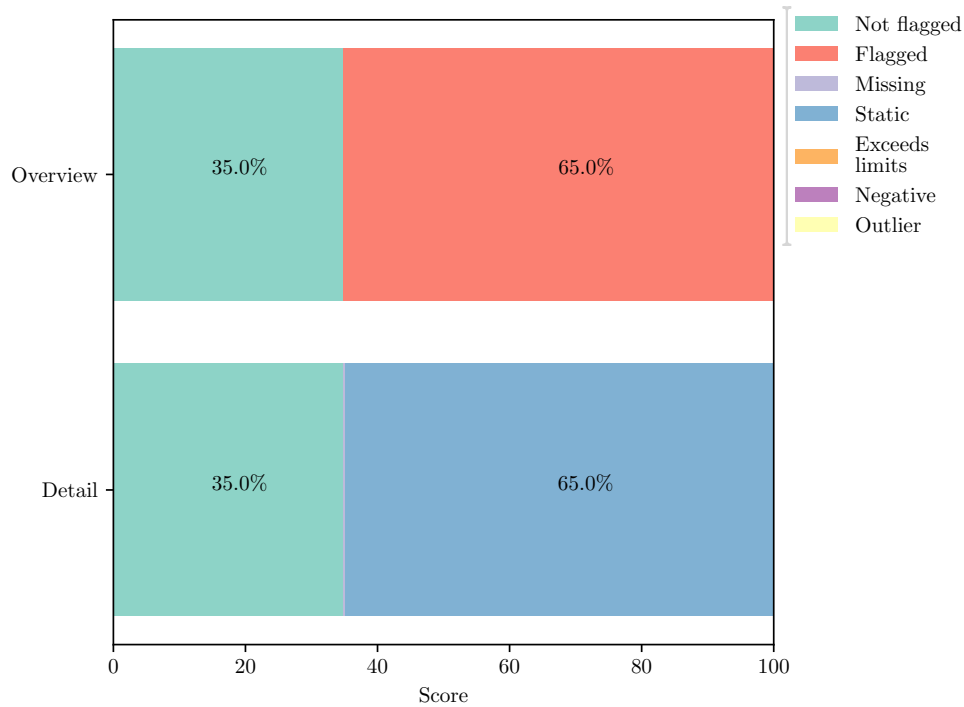
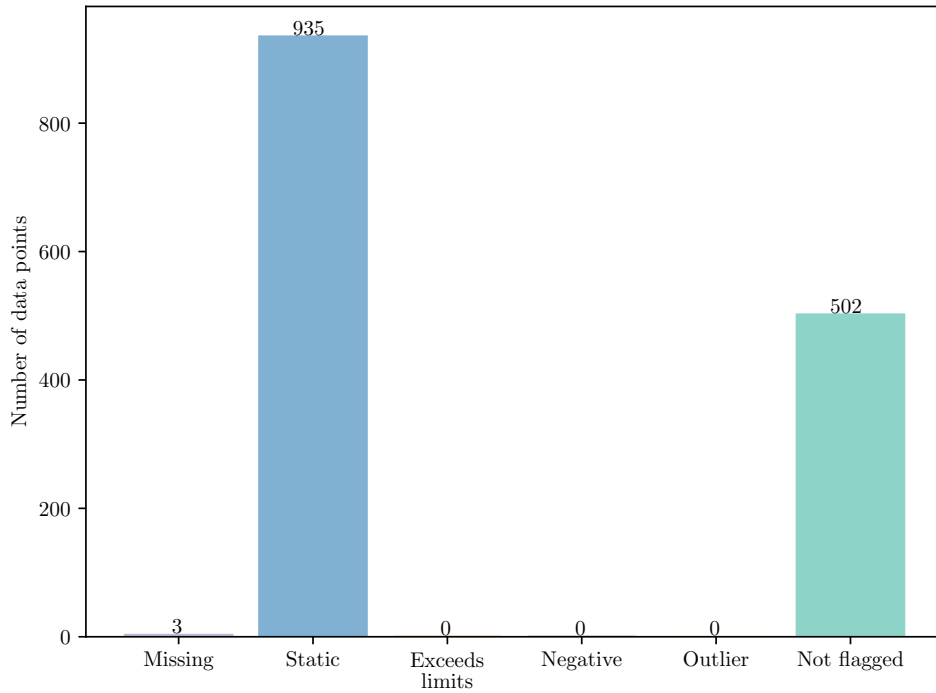


Figure 4.12: *Static – Overview and detailed results*

The graph in Figure 4.13 shows the number of data points marked according to each metric. It can be seen that three data points were missing, 935 data points were static, and 502 data points were not flagged.

Figure 4.13: *Static – Data point count*

The data plots for the tag are shown in Figure 4.14. The data is plotted sequentially in Figure 4.14a for the entire time span. The data is also plotted in daily increments as shown in Figure 4.14b. It is difficult to determine from these graphs where the data points were static.

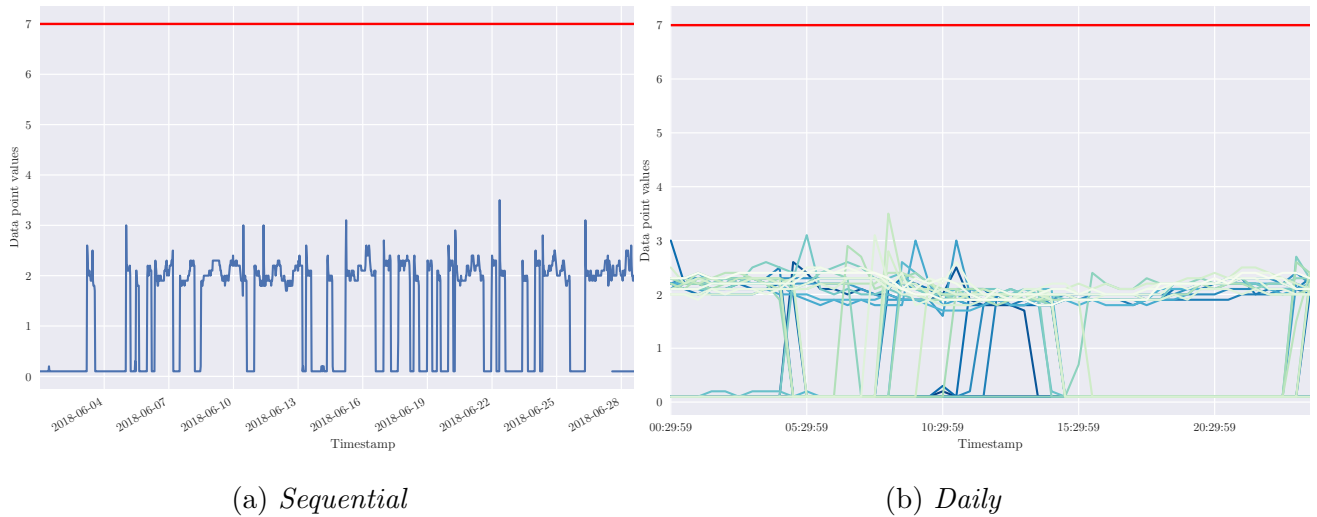
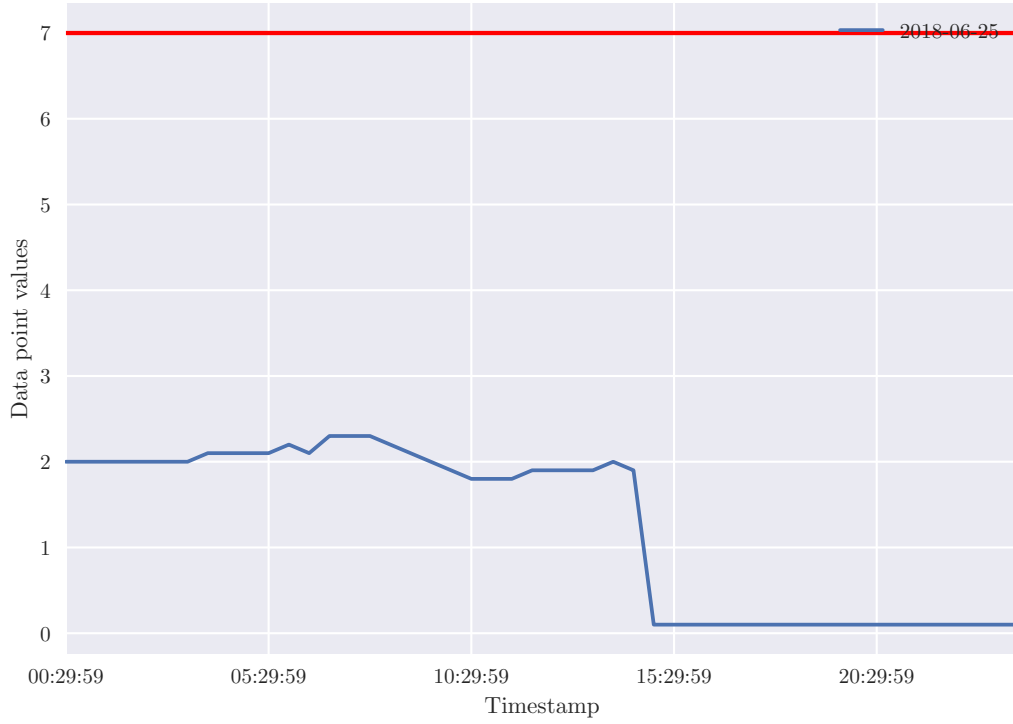
Figure 4.14: *Compressor 2 non-drive end vibration data plots*

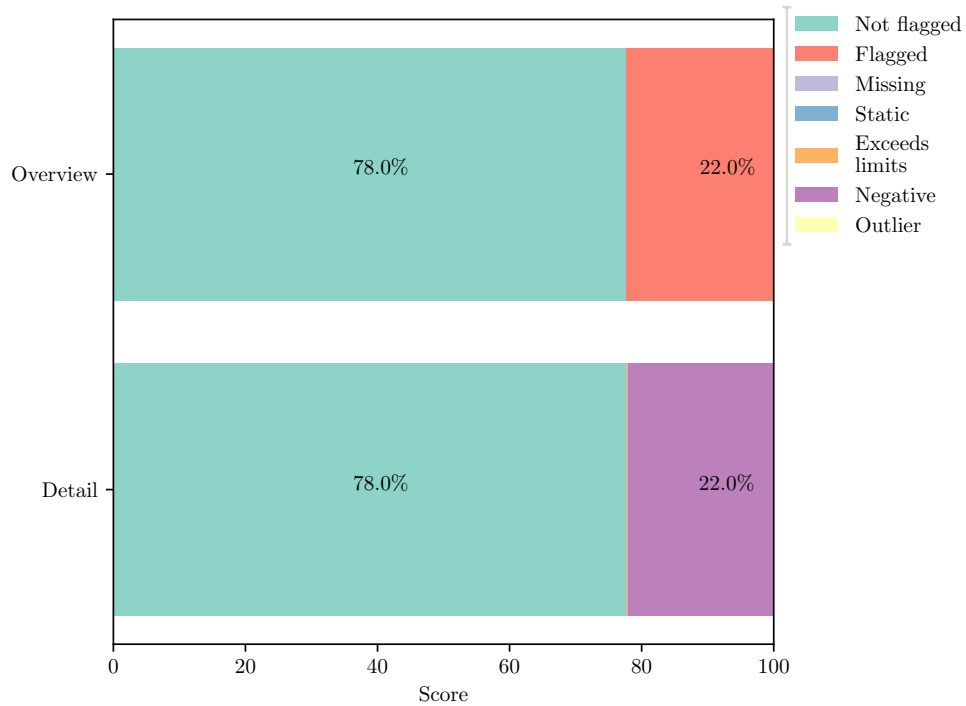
Figure 4.15 shows the data plot for 2018-06-25. From this graph it can be seen that the data became static at various times throughout the day. The Static metric was designed to not include 0 values in the analysis; however, as can be seen in Figure 4.15, the data did not become static at 0. The data did become static at 0.1 mm/s. This means that due to the way the Static metric was implemented, many values were marked as static that probably should not have been.

Figure 4.15: *Data plot – 2018-06-25*

These results highlight a need to improve the accuracy of the data quality analysis platform by using contextual knowledge. For instance, further investigation showed the fridge plant was inactive during the period that the vibration was 0.1 mm/s. This indicates that using other tags linked to a specific component can increase the accuracy of the data quality analysis.

4.3.5 Negative results

The results for the *fridge plant 3 compressor drive end vibration* tag is discussed for the time period from 2018-06-01 00:00:00 to 2018-06-31 23:59:59. Figure 4.16 shows the overview and detailed data quality analysis results for the tag. The overview results show that more than 20% of the data was marked as erroneous. The detailed results show that almost all the erroneous data was due to negative data.

Figure 4.16: *Negative – Overview and detailed results*

The graph in Figure 4.17 shows the number of data points marked according to each metric. It can be seen that four data points were missing, 318 data points were negative, and 1118 data points were not flagged.

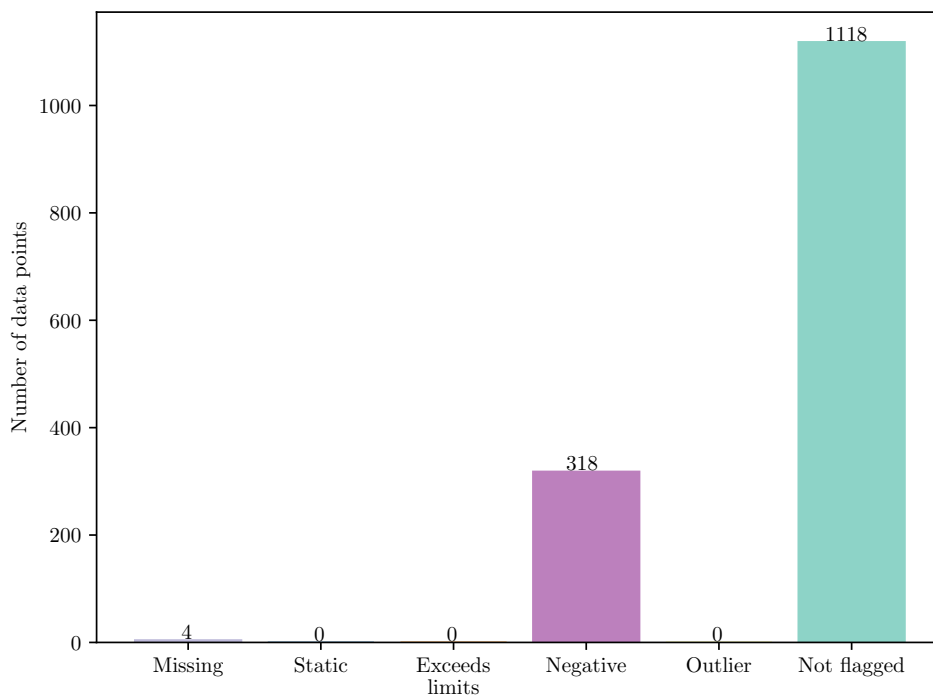
Figure 4.17: *Negative – Data point count*

Figure 4.18a shows the data plot for the tag. From the graph it can be seen that the values fluctuated between -0.1 and 0 . No data points were above 0 for the time span. Figure 4.18b shows the data plotted in daily increments for the selected time span.

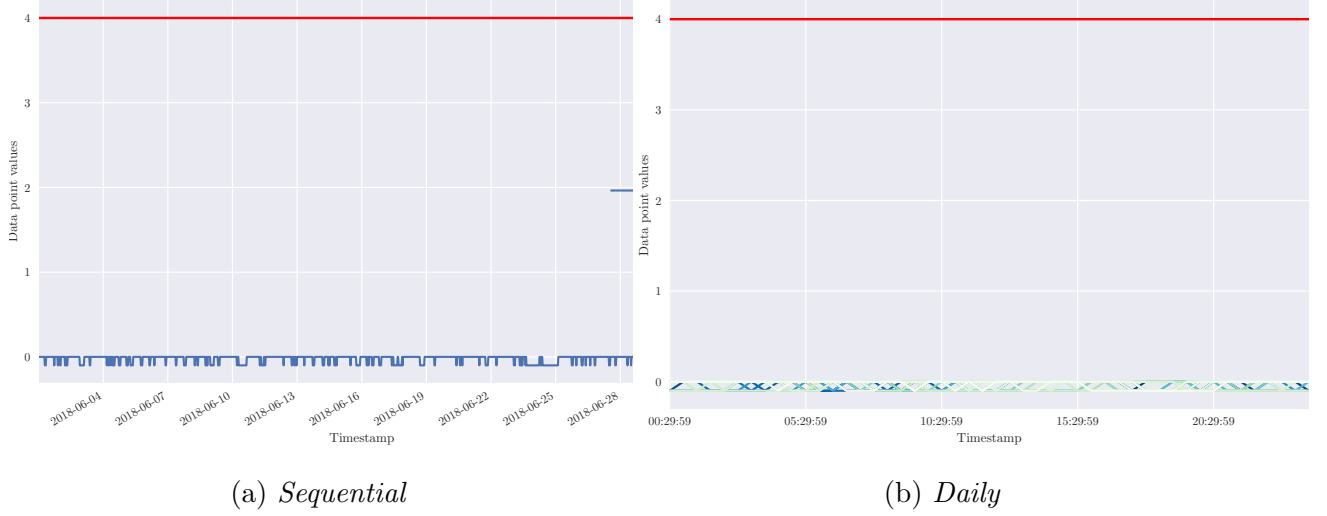


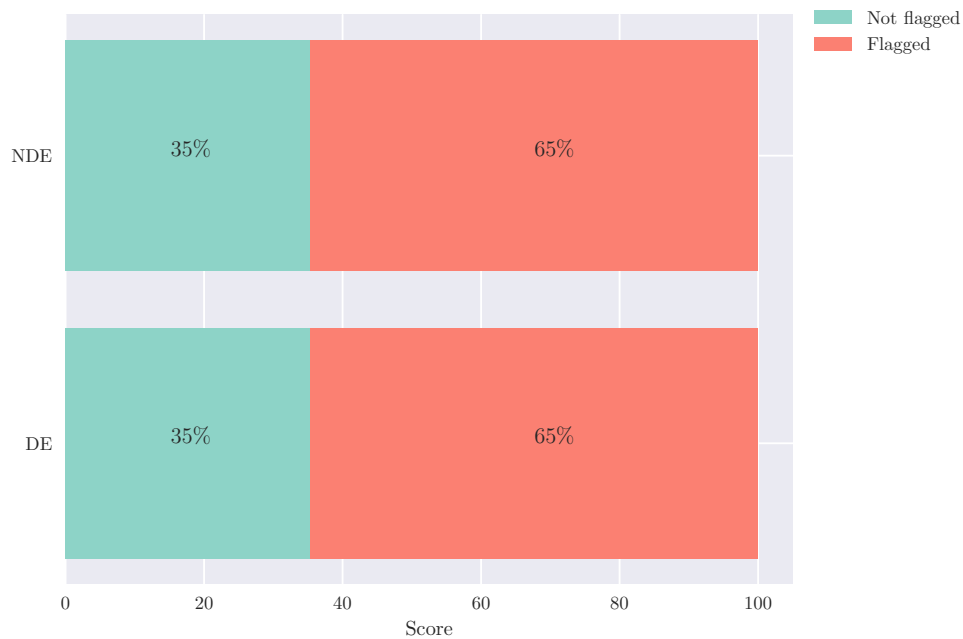
Figure 4.18: *Fridge plant 3 compressor drive end vibration data plots*

A vibration sensor was used to measure displacement for the compressor drive end bearing. The sensor was not supposed to measure negative values, indicating that the sensor was calibrated incorrectly.

4.3.6 Other results

The *fridge plant 4 motor bearing temperature* tags had interesting data quality analysis results. Both the *drive end bearing temperature* and *non-drive end bearing temperature* had the same problems, and the data of both tags had more than one specific problem.

Figure 4.19 shows the overview results for the tags. From the figure it can be seen that the tags had similar data quality scores. Each tag had a data quality score below 40% for the period between 2018-06-01 00:00:0 and 2018-06-30 23:59:59.

Figure 4.19: *Fridge plant 4 motor bearing temperature – Overview results*

The detailed data quality results for the tags are shown in Figure 4.20. The graph shows that the two tags had similar types of error in the data. Each tag had the same amount of static data as well as data that exceeded limits.

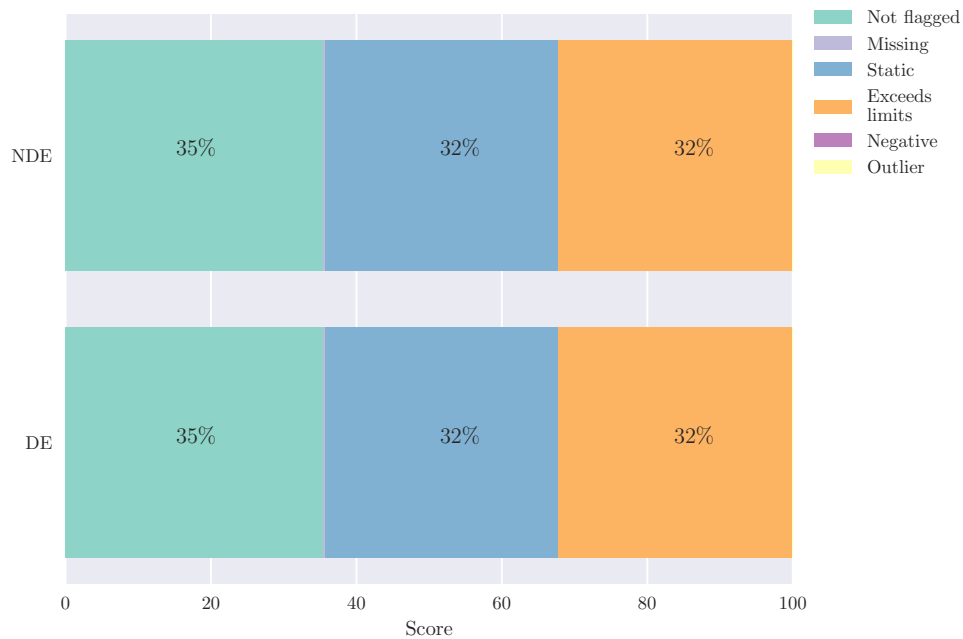
Figure 4.20: *Fridge plant 4 motor bearing temperature – Detailed results*

Figure 4.21 shows the number of data points that satisfy each metric condition for each tag. From Figure 4.21 it can be seen that three data points were missing, 926 data points were static, 927 data points exceeded limits, and 510 were not flagged for both tags.

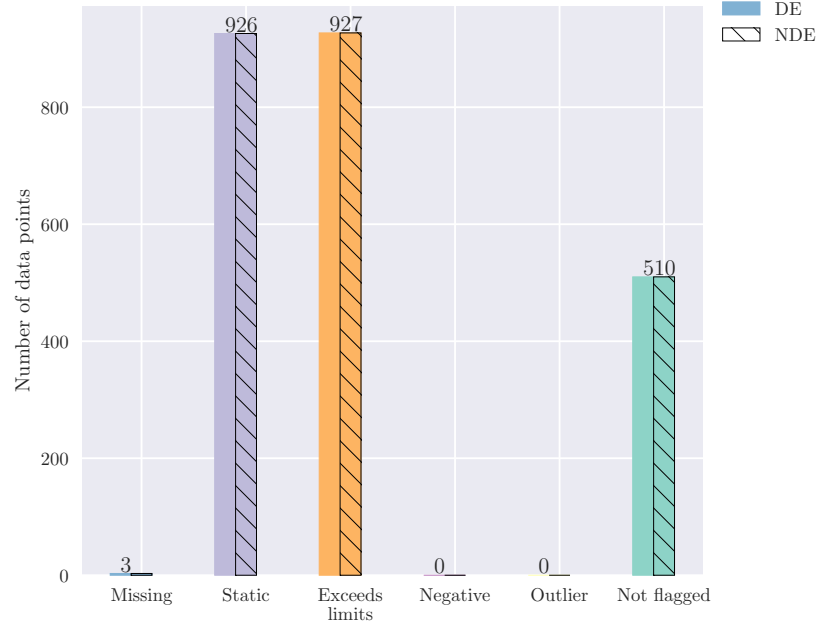


Figure 4.21: *Fridge plant 4 motor bearing temperature – Data point count*

Figure 4.22 shows the data plot for the *drive end bearing temperature* and Figure 4.23 shows the data plot for the *non-drive end bearing temperature* tags. The plots show that the data appears to exceed the limits indicated by the red line on the graph, and the data appears static at numerous timestamps, confirming the results from the data quality analysis.

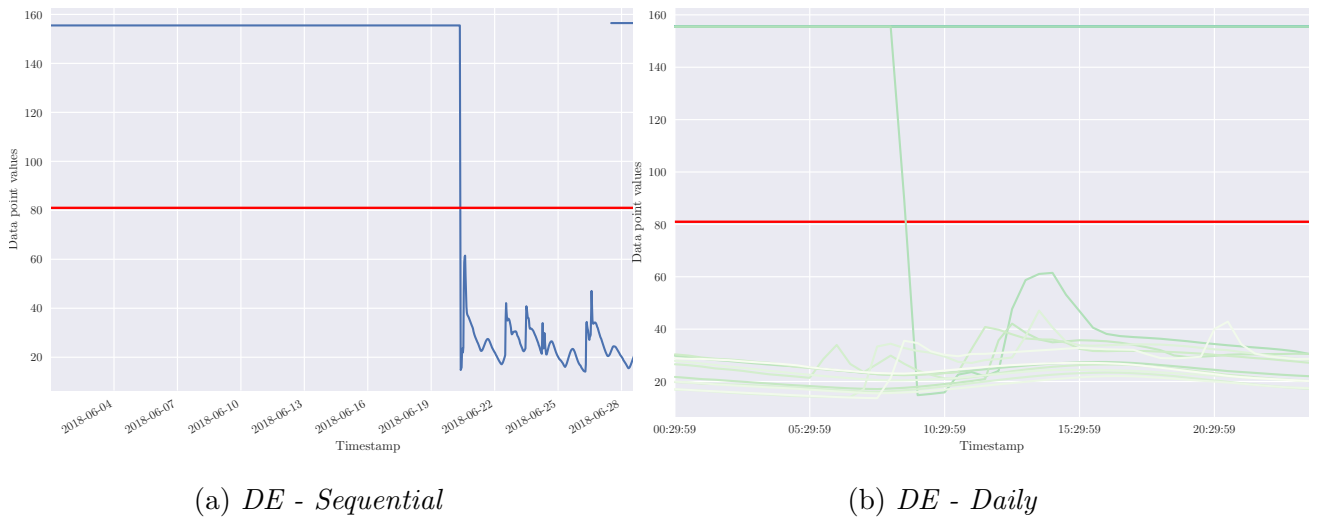


Figure 4.22: *Fridge plant 4 motor bearing temperature data plots*

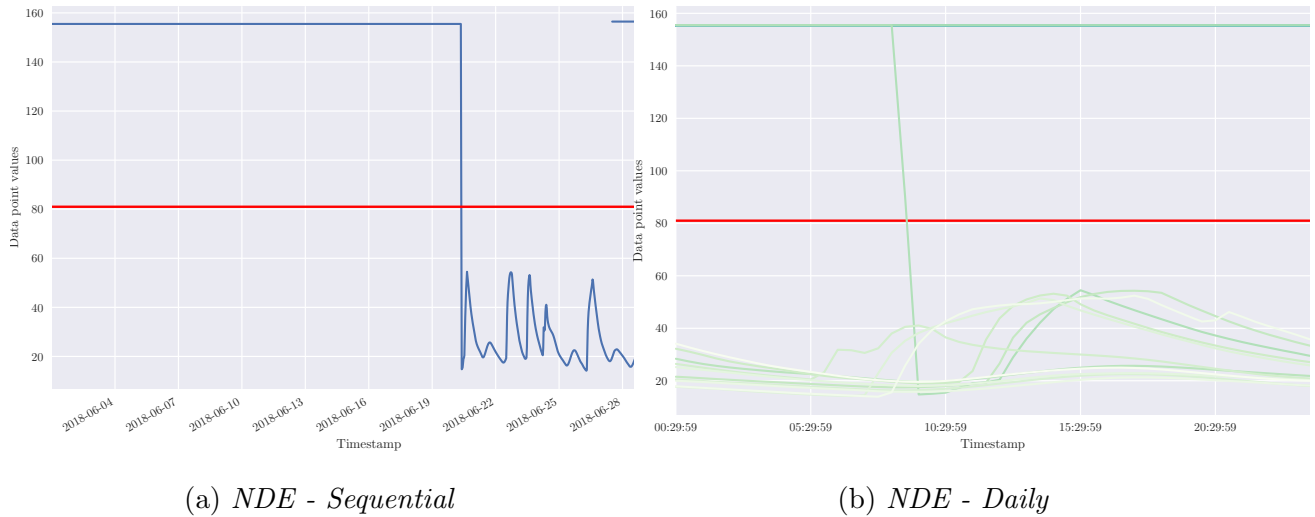


Figure 4.23: Fridge plant 4 motor bearing temperature data plots

Figure 4.22 and Figure 4.23 show that the data was static and above its limit for both tags for a large part of the month. This appeared to be a significant problem. Further investigation showed that the fridge plant was not running during this period as maintenance was being performed. In this case, the data was correctly flagged as erroneous when viewed in isolation; however, when looking at the larger picture, the data was wrongly flagged as erroneous. This highlights a need to expand the data quality analysis system to use contextual information to increase the accuracy of the data quality analysis.

4.3.7 Summary

In the previous sections, the results for each of the metrics were discussed. The results showed that the system is able to analyse industrial data and quantify the quality thereof by calculating an overall data quality score for each tag using the method developed in Section 3.3.

From the results discussed in the previous sections it was found that different problems cause poor data quality for the tags. Table 4.3 summarises the results for each metric as discussed in the previous sections.

Table 4.3: Results summary

Result for metric	Cause	Caused by data problem?	Section
Missing	Communication fault	Yes	Section 4.3.2
Exceeds limits	Maintenance required	No	Section 4.3.3
Static	Sensor calibration	Yes	Section 4.3.4
	Contextual knowledge required	No	Section 4.3.4
Negative	Sensor calibration	Yes	Section 4.3.5
Other	Contextual knowledge required	No	Section 4.3.6

Missing data was caused by communication problems between the mining site and the industrial company's ETL system. The Exceeds Limits error were caused by components that required maintenance. The static data showed that it is important for the data quality analysis system to use contextual knowledge to analyse industrial data quality. The negative results were caused by improperly calibrated sensors. The other results discussed also highlighted the need to use contextual knowledge to increase the data quality analysis accuracy.

4.4 Discussion

4.4.1 Background

Further investigations were done using the results of the data quality analysis for condition monitoring data. These investigations were performed in addition to the study objectives defined in Section 1.5. These investigations were performed to determine:

- The component with the lowest data quality.
- The measurements with the lowest data quality.
- The most common type of error in condition monitoring data (as discussed in Section 4.2).

Figure 4.2 shows the breakdown of the equipment monitored on the mining group's sites. The components monitored by the condition monitoring system for the different mining sites is summarised in Table 4.4.

Table 4.4: Overview of condition monitoring implementation – All sites

Mine site	Pumps	Fans	Fridge plants	Compressors	Total	Distribution
Mine A	9	3	4	4	20	10.98%
Mine B	7	2	5	3	17	9.34%
Mine C	25	24	15	5	69	39.91%
Mine D	8	6	2	3	19	10.44%
Mine E	8	2	3	4	17	9.34%
Mine F	15	8	9	0	32	17.58%
Mine G	0	2	0	6	8	4.39%
Total	72	47	38	25	182	
Distribution	39.56%	25.82%	20.88%	13.74%		

As shown in Table 4.4, 182 different pieces of equipment are monitored by the condition monitoring system across nine mining sites. A sample of 52 components was selected for further analysis. The sample size selected is $\frac{52}{182} \approx 29\%$. Table 4.5 summarises how many of each component were chosen from each mining site.

Table 4.5: Number of components chosen for investigations

Mine site	Pumps	Fans	Fridge plants	Compressors	Total	Distribution
Mine A	2	1	1	1	5	9.62%
Mine B	2	1	1	1	5	9.62%
Mine C	7	7	4	1	19	36.54%
Mine D	2	2	1	1	6	11.54%
Mine E	2	1	1	1	5	9.62%
Mine F	4	3	2	0	9	17.31%
Mine G	0	1	0	2	3	5.77%
Total	19	16	10	7	52	
Distribution	36.54%	30.77%	19.23%	13.46%		

Table 4.6 shows the number of tags analysed for each mining site's components. Different parameters were measured for each component, as summarised in Table 4.2. Not all the parameters were available for each component.

Table 4.6: Number of tags analysed

Mine site	Pumps	Fans	Fridge plants	Compressors	Total
Mine A	20	7	12	14	53
Mine B	22	7	18	21	68
Mine C	69	39	79	17	204
Mine D	22	21	6	25	74
Mine E	9	5	3	19	36
Mine F	33	25	52	0	110
Mine G	0	11	0	39	50
Total	175	115	171	134	595

The data quality analysis was performed on the tags summarised in Table 4.6. The results of the analysis performed on all the tags are summarised in Table 4.7. The results were calculated by determining how many data points were analysed for each mining site. The number of data points flagged by each metric was also determined and is summarised in Table 4.7.

Table 4.7: Analysis results by system

Mine site	Tags analysed	Correct	Faulty	Missing	Static	Exceeds limits	Negative	Outlier
Mine A	53	75 907	413	160	250	141	0	0
Mine B	68	97 180	740	216	431	47	92	0
Mine C	202	272 921	25 159	23 540	1 616	141	0	0
Mine D	70	146 310	570	376	194	0	0	0
Mine E	36	51 593	244	74	123	0	47	0
Mine F	110	157 423	977	177	512	188	232	0
Mine G	50	60 920	11 080	10 757	322	47	0	0
Total	589	862 257	39 183	35 300	3 448	564	371	0

4.4.2 Component analysis

An investigation was performed to determine which components measured by the condition monitoring system had the lowest data quality. For the investigation, the data quality of the tags summarised in Table 4.7 for 2018-06-01 00:00:00 to 2018-06-30 23:59:59 was analysed.

The results were calculated by determining the ratio between the flagged data points and the non-flagged data points using Equation 3.6. The results are shown in Figure 4.24. The bar chart shows that fans appeared to be the component with the highest amount of erroneous data. Pumps and compressors appeared to be among the best components regarding data quality.

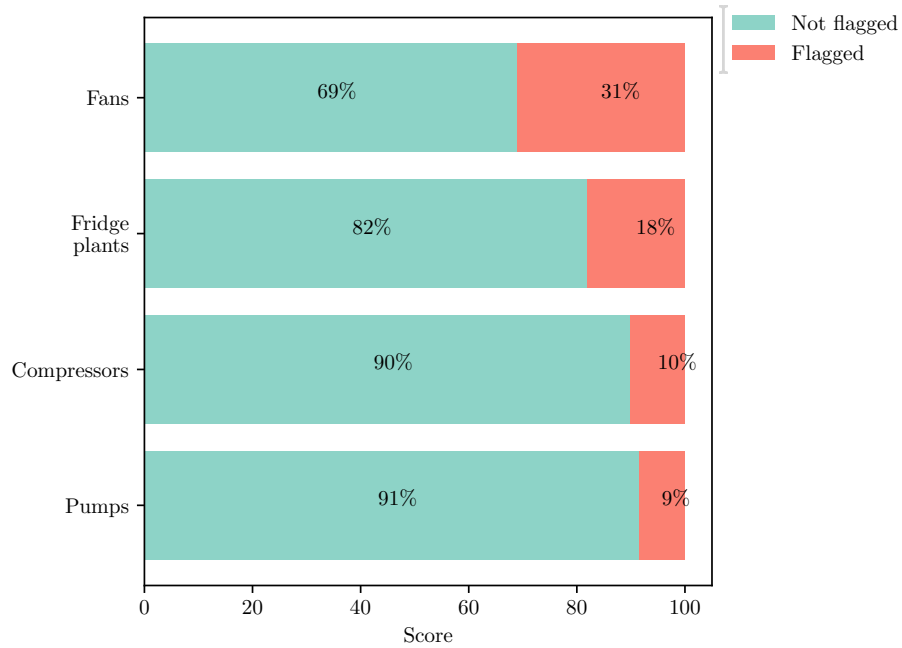
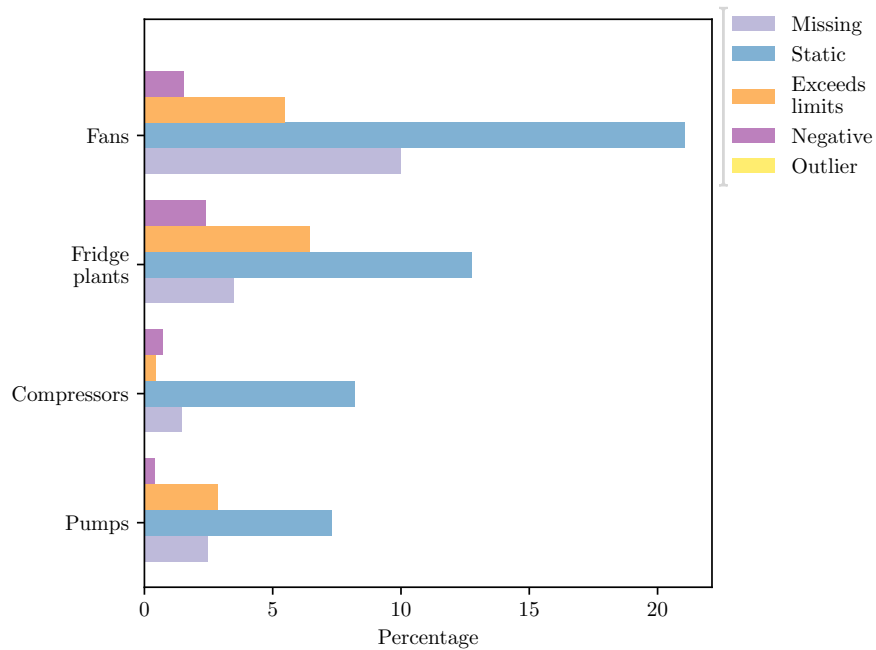
Figure 4.24: *Components – Overview results*

Figure 4.25 shows the percentage of each component’s data that was flagged as erroneous and by which metric. The chart shows that static data was the largest contributor to erroneous data for each component. Missing data was the second-largest contributor to erroneous data for fans and compressors. Data that exceeded limits was the second-largest contributor to erroneous data for fridge plants and pumps.

Figure 4.25: *Component – Metric breakdown*

Further investigation of the tags of the fan components revealed that most of the data quality problems could be traced to various temperature and vibration tags. Many of these tags became static, leading to poor data quality scores for the component. The likely causes of the static data for the various components were sensor faults or poor OPC connections between the mine's SCADA systems and the industrial company's control systems.

4.4.3 Measurement type analysis

An investigation was performed to determine which type of measurement had the lowest data quality. The condition monitoring system monitors different measurements for each component such as:

- Temperature,
- Vibration,
- Flow,
- Pressure,
- Power,
- Position, and
- Running status.

Figure 4.26 shows the results of the data quality analysis for the different measurements. The graph shows that positional data had the lowest data quality. This was followed by flow and vibration. The data quality scores of these three measurements were below 80%. The data quality of the remaining measurements scored higher than 80%.

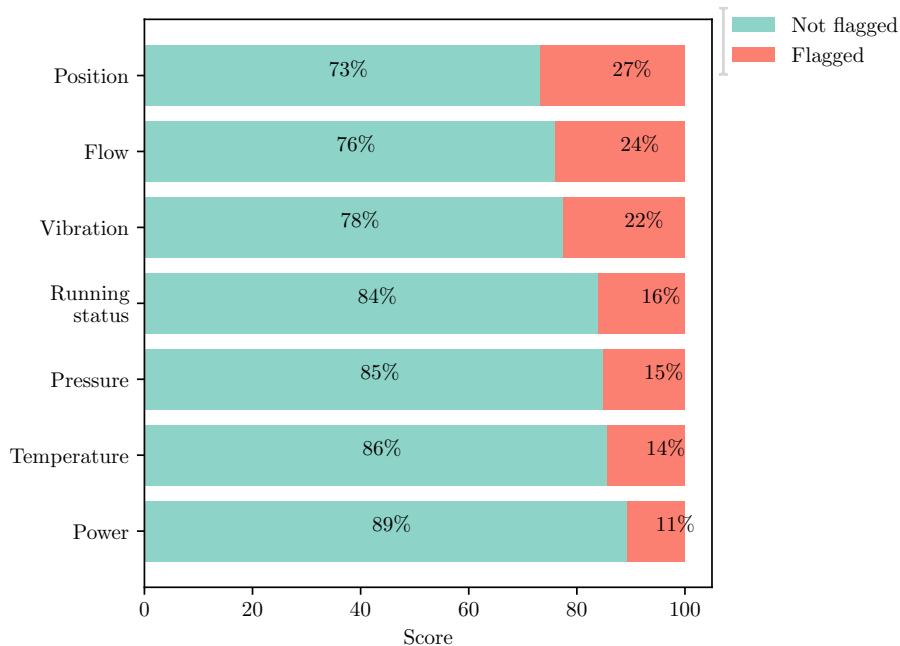


Figure 4.26: *Measurement – Overview results*

Figure 4.27 shows the percentages of each type of error for each measurement. From the chart it can be seen that static data was the highest problem for all the measurement types except running status. Running status tags were not analysed for static data because running status data were expected to be static for long periods of time.

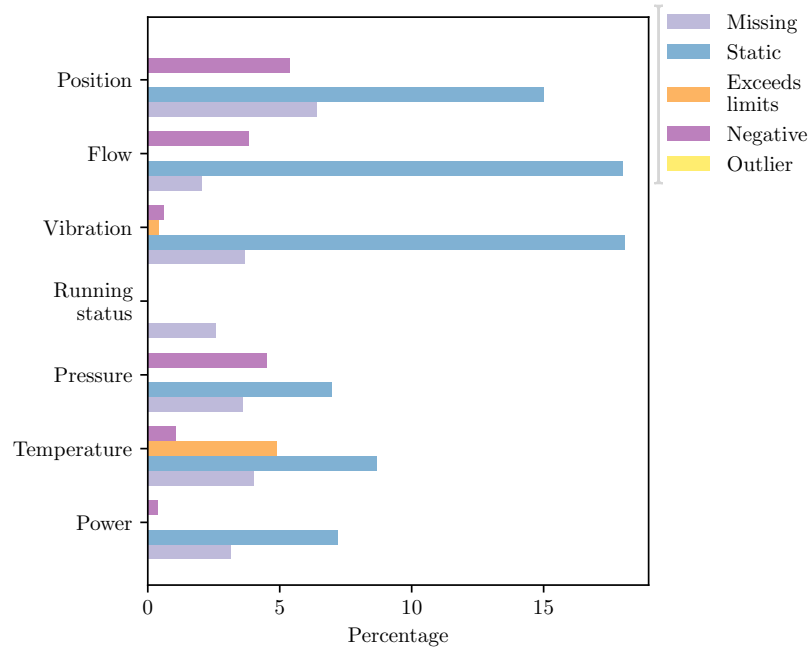


Figure 4.27: *Measurement – Metric breakdown*

Figure 4.27 also shows that data exceeding limits was only found for vibration tags and temperature tags. Further investigation revealed that the Exceeds Limits metric was only used to analyse vibration and temperature data.

The results in the above graphs, however, only considered the number of data points analysed for each measurement type. The results did not take all the data points into account. This made it seem that the positional data was a large contributor to overall poor data quality.

Figure 4.28 shows the data quality results for the different measurements when all the data points were considered. The chart shows that temperature measurement made up almost 50% of the data. The most data quality problems were found in the temperature measurement.

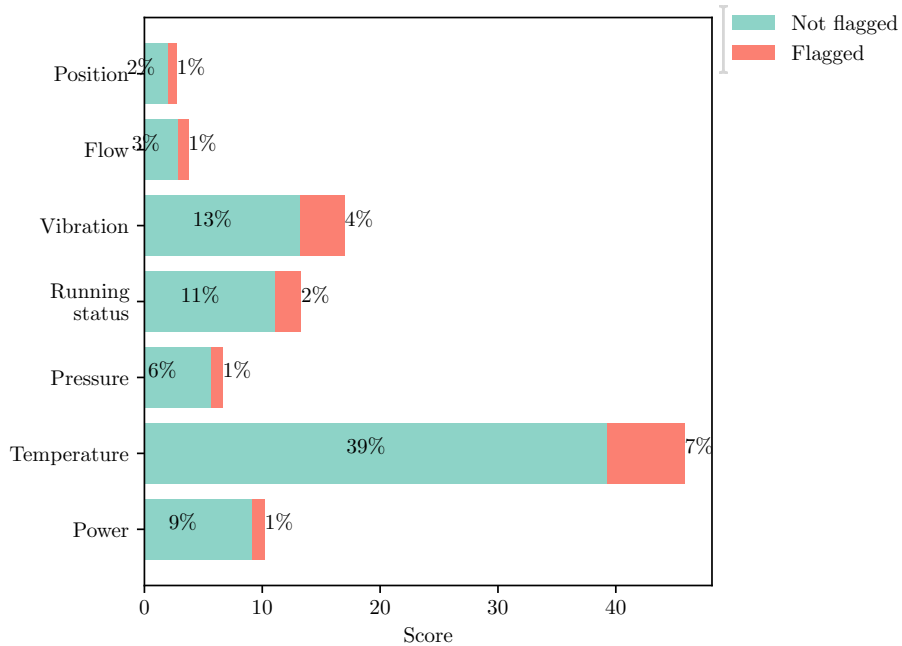
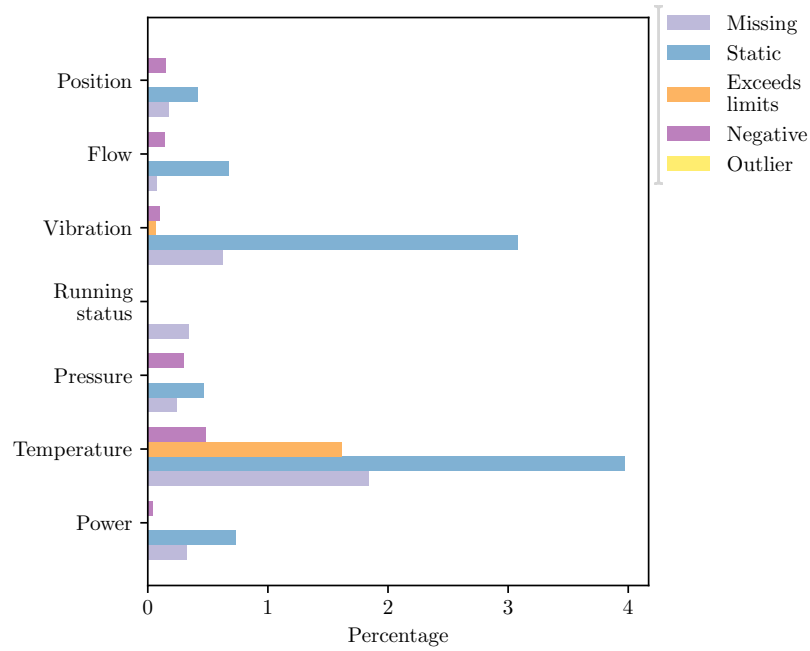
Figure 4.28: *Measurement – Scaled overview results*

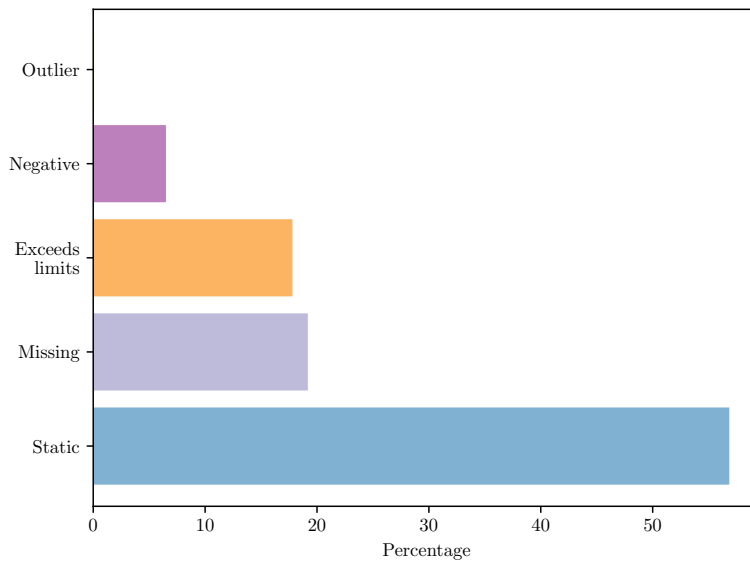
Figure 4.29 shows the results of the data quality analysis for the types of measurement broken down according to the different metrics. The graph shows that static data accounted for the most problems in all of the measurements except the running status measurements. Missing data was also a large contributor to poor data quality for the various types of measurement.

The likely cause for the static data was poor OPC connections between the mine’s SCADA systems and the industrial company’s control systems. The missing data was also likely caused by poor communication between different systems.

Figure 4.29: *Measurement – Scaled metric breakdown*

4.4.4 Metric analysis

An investigation was performed to determine which faults contributed the most to poor quality data. The chart in Figure 4.30 shows the percentage of errors that was caused by each metric. From the chart it can be seen that static data was the largest contributor to poor data quality. Missing data and data exceeding limits were nearly equal contributors to poor data quality.

Figure 4.30: *Metric results*

As shown in the previous sections, static data was the largest contributor to poor data quality when considering either components or types of measurement. The most likely cause of static data was faulty instrumentation or poor OPC connections between the mine's SCADA systems and the industrial company's control systems. The missing data was probably caused by communication faults. These communications faults can occur anywhere in the industrial process, which makes it difficult to determine the exact cause of missing data.

4.4.5 Overview results

The results of the analysis are summarised in Table 4.8, which shows that the data quality for the entire mining group is approximately 84% correct. This means that approximately 16% of the data is regarded as erroneous. This is quite a large subset of the data that is considered erroneous and untrustworthy.

Table 4.8: Overview results

Data points analysed	Correct	Erroneous	Overall score
Missing	888 480	32 169	96.38%
Static	770 400	82 878	89.24%
Exceeds limits	444 960	14 943	96.64%
Negative	888 480	10 805	98.78%
Outlier	888 480	0	100%
Overall	888480	144345	83.75%

Further results in Table 4.8 show that missing data, data exceeding limits and static data were the largest contributors to the overall data quality of the mining group. This is as expected from the results in the previous sections.

4.4.6 Organisational benefits

From the results of the analyses in the previous sections, the industrial company can decide where to focus data quality driven efforts. It is clear that temperature measurements are the most problematic and that static data is a large contributor to poor data quality regardless of measurement type.

The industrial company now has the opportunity to focus attention on the errors that occur most frequently. For instance, less time needs to be spent trying to improve data quality for positional data because it accounts for a small percentage of the errors found in the data.

It is expected that with the system, the employees at the industrial company should have a measure to trust the data. This means that employees can make better decisions based on the data available.

4.4.7 Summary

Investigations were performed in the previous sections to determine:

- The component with the lowest data quality.
- The measurements with the lowest data quality.
- The type of metric flagged the most.

Data from a condition monitoring implementation on the sites of a client from the industrial company was used for the investigation. Components were selected according to the distribution thereof across the mining group's sites. Table 4.5 summarises how many components from each mining site were selected. All the tags were used for the investigations of each selected component.

It was found that fans had the lowest data quality of the four different components. The most common problem for all components was static data. Data exceeding limits was the second-largest contributor to poor data quality for the different components. Static data was caused by OPC connection problems between the mine's SCADA systems and the industrial company's control systems. Data exceeding limits errors were caused by pieces of equipment running outside their operational limits.

The investigation revealed that temperature data and vibration data had the poorest data quality scores. Each of these measurements had large amounts of static and missing data. The errors were likely caused by OPC connection failures between the mine's SCADA systems and the industrial company's control systems.

The Static metric was flagged most often. This was expected from the results shown in the previous two investigations. The Static metric accounted for nearly 60% of the problems identified in the data. The Exceeds Limits metric and Missing metric each accounted for nearly 20% of the problems identified in the data. The large amount of static data was caused by unstable OPC connections between the mine's SCADA systems and the industrial company's control systems. The control system logged the last value it received when the OPC connection failed, thereby causing large amounts of static data.

The industrial company has the opportunity to focus data-driven initiatives in the areas where they will have the most significant impact. For instance, trying to improve positional data is not a worthwhile endeavour compared with trying to improve temperature data.

4.5 Conclusion

The data quality analysis system was implemented and deployed at an industrial company. As discussed in Section 4.2, one of the company's client's mining sites was used as a case study to validate the system.

It was found that the system was able to identify problems in data according to the metrics configured to analyse the data. The results of the data analysis for each metric were discussed in Section 4.3.

It was found that not all the errors in the data were caused by poor data quality. Some problems indicated that components urgently needed maintenance. Other problems highlighted the need to expand the data quality analysis platform to improve its accuracy. These results were summarised in Table 4.3 in Section 4.3.7.

Further investigations were done using a number of tags from all the mining group's sites to determine which components had poorer data quality, which types of measurement had poorer data quality and which metrics were flagged most commonly.

The results found that fans had the poorest data quality. Temperature data and vibration data were among the types of measurement that had the poorest data quality. It was also found that the most commonly flagged metric was the Static metric, followed by the Missing and the Exceeds Limits metrics.

The industrial company can use the developed system to determine which types of measurement lead to poor data quality for their different clients. The results of such analyses can be used to improve client data, save money by focusing data-driven initiatives in the correct areas, and help clients make better decisions using the available data.

Chapter 5

Conclusion



5.1 Summary

The objectives of this study were to investigate current methods and systems used to quantify industrial data quality. Further objectives included developing a method to analyse industrial data and developing a system to quantify industrial data quality using the method as mentioned earlier.

Literature revealed that data quality can be defined as the fitness for the use of data. Alternatively, data quality can be defined according to different dimensions. Each dimension analyses different properties of data.

For this study, four data quality dimensions were analysed: completeness, availability, accuracy and validity. Completeness was used to determine if all the data points were in a data set. Availability was used to determine if all the data was available. Accuracy was used to determine if the data was correct. Lastly, validity was used to determine if the data was within acceptable parameters.

Industrial data covers all kinds of physical values and conditions measured within technical processes. These measurements include but are not limited to temperature, vibration and flow rate. Various errors can occur in industrial data. The four most common industrial data errors include missing data, abrupt error, stuck at error, and out of bounds error.

Further research showed that there is a lack of data quality systems that quantify industrial data quality. Most data quality analysis systems have been developed to analyse CRM data or electronic health data. A few systems have been developed for other applications. Furthermore, a few methods have been developed to analyse industrial data; however, these methods are not used to quantify industrial data quality.

A method was developed to analyse industrial data quality using five metrics. The metrics were used to analyse industrial data for missing data, static data, data that exceeds limits, negative data and outliers. The system then used the results of the method to quantify the data quality.

The system was implemented and deployed at an industrial company. The company has clients in the South African mining and industrial sectors. Data was collected and processed from clients to deliver various services. The system was validated using data from a condition monitoring implementation on a mining system. From the results, it was found that the system can analyse and identify problems in data using the five metrics developed in Section 3.3.

Further analysis was performed using a sample of tags from all the mining group's sites. The results showed that fans had the lowest overall data quality compared with pumps, compressors and fridge plants. Fans had 31% erroneous data compared to 18% for fridge plant, 10% for compressors and 9% for pumps. It was also found that temperature data was more problematic than other types of measurement with 15.2% of temperature data being faulty. Lastly, it was found that the most common fault in data was static data which accounted for more than 50% of the errors found. The most likely cause for errors in the data is OPC connection failures between the mine's SCADA systems and the industrial company's control systems.

The industrial company can use the developed system to determine which types of measure-

ment lead to poor data quality for their different clients. The results of such analyses can be used to improve client data, save money by focusing data-driven initiatives in the correct areas, and help clients make better decisions using the available data.

5.2 Study closure

Despite the limitations of the study as discussed in Section 5.3, all research objectives were met. Each of the specific objectives was addressed in detail as summarised in Table 5.1. The table shows that all the literature objectives were met in Chapter 2. The empirical objectives were met and discussed in Chapter 3.

Table 5.1: System requirements verification

Objective	Section	Objective met
Literature objectives		
Review data quality dimensions as discussed in Table 1.1 and their application to industrial data.	Section 2.3	✓
Research how data quality is measured and quantified.	Section 2.3 and Section 2.4	✓
Research what types of error are found in industrial data.	Section 2.2	✓
Research how industrial data errors are identified.	Section 2.2	✓
Review existing data quality analysis systems.	Section 2.5	✓
Empirical objectives		
Develop a method to analyse data for errors. The method should be able to analyse industrial data with different types, resolutions and formats.	Section 3.3	✓
Develop and implement a system to quantify data quality. The system should use the method developed to quantify industrial data quality.	Section 3.4	✓
Validate the method: • Create cases to test the method. • Analyse test cases using the developed method. • Compare results from the analysis with the expected results.	Section 3.5	✓

5.3 Limitations and recommendations

The developed system and method have three identified and discussed limitations three recommendations. This section discusses recommendations to counter the limitations found, or for further studies to improve on the knowledge gained.

Scheduling limitations and recommendations

The system requires a scheduler to run continuously to analyse the tags. This requirement wastes computing resources. A more efficient approach would be to analyse the data as it is added to the database, or for the ETL to trigger the quality analysis process once the data is received.

Identifying good data

Historical data is often used to calculate baselines for projects or to create simulations of existing systems. These processes require a large amount of good quality data that is as accurate as possible. Further research can be conducted on developing a system to use the results of the developed method to search for good quality data. Results from such a study can decrease the time it takes to implement accurate simulations and calculate accurate baselines for industrial projects.

Additional metrics

Additional metrics can be added to the system as required. Currently, the system is implemented to make it easy for developers to include additional metrics as required. However, it is essential to analyse the merit of any additional metric before adding it as too many metrics can confuse users.

The system currently analyses the most commonly found problems in data used by the industrial company's employees, and as such, no new metrics have been requested since the system has been implemented.

Contextual knowledge

Currently, each tag is analysed in isolation. No contextual information is used to analyse the data for a tag. Contextual information can include what type of component is being analysed or what other tags are linked to the component. A further study can be conducted to increase the accuracy of data quality analysis using contextual information for tags as discussed in Section 4.3.6.

Application to other systems

The method can be incorporated into systems such as the condition monitoring system implemented for the mining group discussed in Section 4.2. The results of the method can

be used to quickly identify problems in components. An example is discussed in Section 4.3.3.

Application to other industries

The method can easily be used to analyse numerical time series data. The method has been designed in a way to analyse any numerical data without context. The design makes the analysis generic, which means the method can be used in different industries.

The system can be used to analyse data from other industries with no modifications given that the data is stored in JSON documents. If the data is stored in a different data structure to which the results of the method's analysis can easily be saved, minimal system modifications are required. This means both the method and the system can easily be adapted for use in other industries.

Reference list

- [1] P. Ffoulkes, “The intelligent use of big data on an industrial scale,” Hewlett Packard Enterprise, Tech. Rep., 2017.
- [2] S. Yin and O. Kaynak, “Big data for modern industry: Challenges and trends [point of view],” *Proceedings of the IEEE*, vol. 103, no. 2, pp. 143–146, 2015.
- [3] A. Zaslavsky, C. Perera, and D. Georgakopoulos, “Sensing as a service and big data,” in *Proceedings of the International Conference on the Advances in Cloud Computing (ACC)*, 2013.
- [4] J. Manyika and M. Bailly, “Is manufacturing ‘cool’ again?” McKinsey Global Inst., Tech. Rep., 2013.
- [5] D. Feinleib, *Big data bootcamp: What managers need to know to profit from the big data revolution*, 1st ed. Berkeley, CA: Apress, 2014.
- [6] O. Kwon, N. Lee, and B. Shin, “Data quality management, data usage experience and acquisition intention of big data analytics,” *International Journal of Information Management*, vol. 34, no. 3, pp. 387–394, 2014.
- [7] V. Grover, R. H. L. Chiang, T. Liang, and D. Zhang, “Creating Strategic Business Value from Big Data Analytics : A Research Framework,” *Journal of Management Information Systems*, vol. 35, no. 2, pp. 388–423, 2018.
- [8] S. Mohanty, M. Jagadeesh, and H. Srivatsa, *Big data imperatives: Enterprise big data warehouse, BI implementations and analytics*, 1st ed. Berkeley, CA: Apress, 2013.
- [9] W. Eckerson, “Data quality and the bottom line,” The Data Warehouse Institute, Tech. Rep., 2002.
- [10] R. Singh and K. Singh, “A descriptive classification of causes of data quality problems in data warehousing,” *IJCSI International Journal of Computer Science Issues*, vol. 7, no. 2, pp. 41 – 50, 2010.
- [11] A. Haug, F. Zachariassen, and D. van Liempd, “The costs of poor data quality,” *Journal of Industrial Engineering and Management*, vol. 4, no. 2, pp. 168–193, 2011.
- [12] M. Aly, S. Yacout, and Y. Shaban, “Analysis of massive industrial data using MapReduce framework for parallel processing,” in *Annual Reliability and Maintainability Symposium*, 2017, pp. 1–6.
- [13] B. Esmailian, S. Behdad, and B. Wang, “The evolution and future of manufacturing: A review,” *Journal of Manufacturing Systems*, vol. 39, pp. 79–100, 2016.
- [14] C. Gamarra, J. M. Guerrero, and E. Montero, “A knowledge discovery in databases approach for industrial microgrid planning,” *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 615–630, 2016.

-
- [15] S. van Jaarsveld, S. van Heerden, and J. van Rensburg, “Development of a condition monitoring information system for deep level mines.” presented at the SAIIE28 proceedings, 2017.
 - [16] N. Laranjeiro, S. N. Soydemir, and J. Bernardino, “A Survey on Data Quality : Classifying Poor Data,” in *The 21st IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2015)*, 2015.
 - [17] S. Neumaier, J. Urgen, and A. Polleres, “Automated Quality Assessment of Metadata across Open Data Portals,” *Journal of Data and Information Quality*, vol. 8, no. 1, 2016.
 - [18] C. Batini and M. Scannapieco, *Data and Information Quality*. Switzerland: Springer International Publishing, 2016.
 - [19] M. A. Jabar and A. S. M. Alnatsha, “Knowledge management system quality: A survey of knowledge management system quality dimensions,” in *2014 International Conference on Computer and Information Sciences (ICCOINS)*, June 2014, pp. 1–5.
 - [20] PricewaterhouseCoopers, “Global data management survey,” PricewaterhouseCoopers, London, Tech. Rep., 2002.
 - [21] T. Friedman, “Gartner Dataquest.”
 - [22] A. Haug, J. S. Arlbjørn, A. Haug, and J. S. Arlbjørn, “Barriers to master data quality,” *Journal of Enterprise Information Management*, vol. 24, no. 3, pp. 288–303, 2011.
 - [23] R. Silvola, J. Harkonen, O. Vilppola, H. Kropsu-Vehkapera, and H. Haapasalo, “Data quality assessment and improvement,” *International Journal of Business Information Systems*, vol. 22, no. 1, pp. 62–81, 2016.
 - [24] L. English, *Improving data warehouse and business information quality*, 1st ed. New York, NY: John Wiley and Sons, 1999.
 - [25] H. Decker, “Quantifying the quality of stored data by measuring their integrity,” *2009 2nd International Conference on the Applications of Digital Information and Web Technologies*, pp. 796–801, 2009.
 - [26] B. Heinrich, M. Kaiser, and M. Klier, “How to measure data quality?” in *Proceedings of the 16th International Conference on Information Quality (ICIQ-2011)*, 2011, pp. 1–15.
 - [27] B. Heinrich and M. Klier, “Metric-based data quality assessment: Developing and evaluating a probability-based currency metric,” *Decision Support Systems*, vol. 72, pp. 82–96, 2015.
 - [28] D. Boyadzhieva and B. Kolev, “An extension of the relational model to intuitionistic fuzzy data quality attribute model,” in *2008 4th International IEEE Conference Intelligent Systems, IS 2008*, vol. 3, 2008, pp. 1314–1318.
 - [29] A. Lazar, L. Jin, C. A. Spurlock, A. Todd, K. Wu, and A. Sim, “Data quality challenges with missing values and mixed types in joint sequence analysis,” in *IEEE International Conference on Big Data (BIGDATA)*, 2017, pp. 2620–2627.
 - [30] M. G. Kahn, M. A. Raebel, J. M. Glanz, K. Riedlinger, F. Steiner, S. M. Care, S. Electronic, D. Methods, and E. D. M. Forum, “A pragmatic framework for single-site and multisite data quality assessment in electronic health record-based clinical research,” *Medical Care*, vol. 50, no. 7, pp. S21–S29, 2012.
-

-
- [31] O. Almutiry, G. Wills, A. Alwabel, R. Crowder, and R. Walters, "Toward a framework for data quality in cloud-based health information system," *International Conference on Information Society (i-Society)*, pp. 153–157, 2013.
 - [32] D. Hamdan, O. Aktouf, I. Parissis, B. El Hassan, and A. Hijazi, "Online data fault detection for wireless sensor networks: Case study," in *2012 International Conference on Wireless Communications in Underground and Confined Areas*, 2012, pp. 1–6.
 - [33] Y. Zhang, C. Bingham, Z. Yang, M. Gallimore, and W. Ling, "Sensor fault detection for industrial gas turbine system by using principal component analysis based y-distance indexes," in *2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, 2012, pp. 1–6.
 - [34] Y. Zhang, C. M. Bingham, M. Gallimore, Z. Yang, and J. Chen, "Sensor fault detection for industrial systems using a hierarchical clustering-based graphical user interface," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2012, pp. 389–394.
 - [35] M. Navi, M. Davoodi, and N. Meskin, "Sensor fault detection and isolation of an industrial gas turbine using partial block-wise adaptive kernel PGA," in *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, apr 2017, pp. 1054–1059.
 - [36] A. M. Bardawily, M. Tamazin, and A. Nasser, "Sensors fault estimation , isolation and detection using MIMO extended Kalman filter for industrial applications," in *10th International Conference on Electrical and Electronics Engineering (ELECO)*, 2017, pp. 2–6.
 - [37] M. Rezaee, "Kalman filter based sensor fault detection and identification in an electro-pump system," in *2017 5th International Conference on Control, Instrumentation, and Automation (ICCIA)*, 2017, pp. 12–17.
 - [38] I. Kirchen, D. Schutz, J. Folmer, and B. Vogel-Heuser, "Metrics for the evaluation of data quality of signal data in industrial processes," in *2017 IEEE 15th International Conference on Industrial Informatics*, 2017, pp. 819–826.
 - [39] R. Isermann and P. Ballé, "Trends in the application of model based fault detection and diagnosis of technical processes," *IFAC Proceedings Volumes*, vol. 29, no. 1, pp. 6325–6336, 1996.
 - [40] A. Bennane and S. Yacout, "Processing missing and inaccurate data in a condition based maintenance database," in *40th International Conference on Computers and Industrial Engineering: Soft Computing Techniques for Advanced Manufacturing and Service Systems*, 2010, pp. 1–5.
 - [41] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6414–6428, 2014.
 - [42] Z. Gao, C. Cecati, and S. X. Ding, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques — Part I : Fault Diagnosis With Model-based and Signal-based Approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.
-

-
- [43] P. Woodall, A. Borek, and A. K. Parlikad, “Data quality assessment : The Hybrid Approach,” *Information & Management*, vol. 50, no. 7, pp. 369–382, 2013.
- [44] B. T. Hazen, C. A. Boone, J. D. Ezell, and L. A. Jones-farmer, “Data quality for data science , predictive analytics , and big data in supply chain management : An introduction to the problem and suggestions for research and applications,” *Intern. Journal of Production Economics*, vol. 154, pp. 72–80, 2014.
- [45] A. Vetrò, L. Canova, M. Torchiano, C. O. Minotas, R. Iemma, and F. Morando, “Open data quality measurement framework: Definition and application to Open Government Data,” *Government Information Quarterly*, vol. 33, no. 2, pp. 325–337, 2016.
- [46] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, “Methodologies for data quality assessment and improvement,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–52, 2009.
- [47] N. A. Emran, “Data completeness measures,” in *Pattern Analysis, Intelligent Security and the Internet of Things*, A. Abraham, A. K. Muda, and Y.-H. Choo, Eds. Cham, Switzerland: Springer International Publishing, 2015, pp. 117–130.
- [48] M. Rashid, G. Rizzo, M. Torchiano, N. Mihindukulasooriya, O. Corcho, and R. García-Castro, “Completeness and consistency analysis for evolving knowledge bases,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 54, pp. 48–71, 2019.
- [49] D. Vatsalan, Z. Sehili, P. Christen, and E. Rahm, “Privacy-preserving record linkage for big data: Current approaches and research challenges,” in *Handbook of Big Data Technologies*. Springer, 2017, pp. 851–895.
- [50] F. Li, S. Nastic, and S. Dustdar, “Data quality observation in pervasive environments,” in *2012 IEEE 15th International Conference on Computational Science and Engineering*, dec 2012, pp. 602–609.
- [51] G. Y. Shi, T. Hayasaka, A. Ohmura, Z. H. Chen, B. Wang, J. Q. Zhao, H. Z. Che, and L. Xu, “Data quality assessment and the long-term trend of ground solar radiation in China,” *Journal of Applied Meteorology and Climatology*, vol. 47, no. 4, pp. 1006–1016, 2008.
- [52] P. Evans, “Scaling and assessment of data quality,” *Acta Crystallographica Section D: Biological Crystallography*, vol. 62, no. 1, pp. 72–82, 2006.
- [53] P. Goosen, J. Swanepoel, and J. du Plessis, “The need for a comprehensive energy management information system for industries,” *South African Journal of Industrial Engineering*, vol. 27, no. 3, pp. 1–11, 2016.

Appendix A

Outlier identification methods

A.1 Preamble

One of the metrics used in the data quality analysis system is the Outlier metric as discussed in Section 3.3. The metric is used to identify outliers in data. Various methods exist to identify outliers in data.

For the data quality analysis system, four methods were implemented:

- Moving average,
- Density-based spatial clustering of applications with noise,
- Nearest neighbours, and
- One-class SVM.

Each of the methods used are discussed in the following sections.

A.2 Moving average

In statistics, a moving average (rolling average or running average) is a calculation to analyse data points by creating series of averages of different subsets of the full data set.¹ Given a series of numbers and a fixed subset size, the first element of the moving average is obtained by taking the average of the initial fixed subset of the number series. Thereafter, the subset is modified by excluding the first number of the series and including the next value in the subset.

Mathematically, a moving average is a type of convolution; therefore, it can be viewed as an example of a low-pass filter used in signal processing. When used with non-time series data, a moving average filters higher frequency components without any specific connection to time, although typically some kind of ordering is implied.²

¹E. A. Omran and W. A. Murtada, “Fault Detection and Identification of spacecraft reaction wheels using Autoregressive Moving Average model and neural networks,” in *2016 12th International Computer Engineering Conference (ICENCO)*, Cairo, 2016, pp. 77-82.

² B.P. Lathi, *Linear systems and signals*, 2nd ed. New York, NY, USA: Oxford university press, 2010.

The following python code can be used to calculate the moving average of time series data.

```
import numpy as np
def DetermineMovingAverage(aInterval , aWindowSize):
    window = np.ones(int(aWindowSize))/float(aWindowSize)
    return np.convolve(aInterval , window , 'same')
```

In order to determine the outliers using the moving average, the distance between the moving average and the data point is calculated. If it exceeds a specified maximum distance, the data point is marked as an outlier.

A.3 Density-based spatial clustering of applications with noise

Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering algorithm.³ Given a set of points in some space, it groups points that are closely packed together. Points that lie alone in low-density regions are marked as outliers.³

Consider a set of points in some space to be clustered. For the purpose of DBSCAN clustering, the points are classified as core points, (density-)reachable points and outliers as follows:³

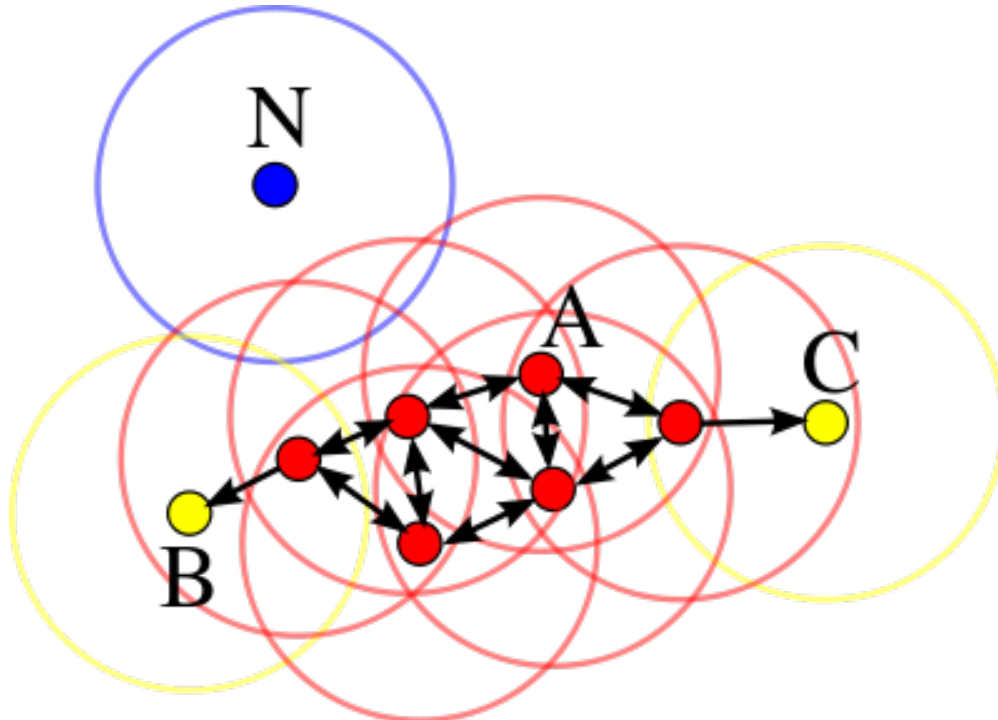
- A point p is a core point if at least $minPts$ points are within distance ε (ε is the maximum radius of the neighbourhood from p) of it (including p). Those points are said to be directly reachable from p .
- A point q is directly reachable from p if point q is within distance ε from point p ; p must be a core point.
- A point q is reachable from p if there is a path $p_1, \dots, p_n$ with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i (all the points on the path must be core points, with the possible exception of q).
- All points not reachable from any other point are outliers.

If p is a core point, it forms a cluster together with all points (core or non-core points) that are reachable from it. Each cluster contains at least one core point. Non-core points can be part of a cluster, but they form its "edge" since they cannot be used to reach more points.

In Figure A.1, $minPts = 4$. Point A and the other red points are core points because the area surrounding these points in an ε radius contains at least four points (including the point itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points, but they are reachable from A (via other core points) and thus belong to the cluster as well. Point N is an outlier point that is neither a core point nor directly reachable.

Reachability is not a symmetric relation since, by definition, no point may be reachable from a non-core point regardless of distance (so a non-core point may be reachable, but

³M. Celik, F. Dadaser-elik and A. S. Dokuz, "Anomaly detection in temperature data using DBSCAN algorithm.", in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, Istanbul, 2011, pp. 91-95.

Figure A.1: *DBSCAN*

nothing can be reached from it). Therefore, a further notion of connectedness is needed to formally define the extent of the clusters found by DBSCAN. Two points, namely, p and q are density-connected if there is a point o such that both p and q are reachable from o . Density-connectedness is symmetric. A cluster then satisfies two properties:

1. All points within the cluster are mutually density-connected.
2. If a point is density-reachable from any point of the cluster, it is part of the cluster as well.

A.4 Nearest neighbours regression

The principle behind nearest neighbour methods is to find a predefined number of training samples closest in distance to the new point and predict the label from these.⁴ The number of samples can be a user-defined constant for k -nearest neighbour learning, or vary based on the local density of points for radius-based neighbour learning. The distance can, in general, be any metric measure. Neighbour-based methods are known as non-generalising machine learning methods since they remember all of their training data.

Neighbour-based regression can be used in cases where the data labels are continuous rather than discrete variables. The label assigned to a query point is computed based on the mean of the labels of their nearest neighbours.

The scikit-learn python library implements two different neighbour regressors, namely, *KNeighborsRegressor* and *RadiusNeighborsRegressor*.⁴

⁴ Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, no. 12, pp. 2825-2830, 2011

- *KNeighborsRegressor* implements learning based on the k -nearest neighbours of each query point, where k is an integer value specified by the user.
- *RadiusNeighborsRegressor* implements learning based on the neighbours within a fixed radius r of the query point, where r is a floating-point value specified by the user.

Once the nearest neighbour regression has been calculated, the outliers can be determined. The outliers are determined by calculating the distances between the *RadiusNeighborsRegressor* prediction and the data point value. If the distances are larger than a specified distance, the data point is considered to be an outlier.

A.5 One-class SVM

In machine learning, SVMs are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis.⁵ Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.⁵

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.⁵

One-class SVM is used for novelty detection, that is, given a set of samples, it will detect the soft boundary of that set so as to classify new points as belonging to that set or not. The class that implements this is called *OneClassSVM*. In this case, as it is a type of unsupervised learning, the fit method will only take as input an array X , as there are no class labels.

A.6 Consolidating results from outlier detection methods

Once the outliers have been identified using the various methods, the outliers are compared. Any data point that has been identified by more than two methods as an outlier is deemed a valid outlier and the data point is marked as such in the database. Any data point not identified by more than one outlier detection method is considered to be a valid data point.

⁵Y. Chen, J. Qian and V. Saligrama, “A new one-class SVM for anomaly detection,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, 2013, pp. 3567-3571.

Appendix B

Metric verification results

B.1 Introduction

A number of tests were performed in Section 3.5 to determine if the system works as expected. In the section, a number of tags were created and configured for data quality analysis. The expected results for the analyses are summarised in Table B.1.

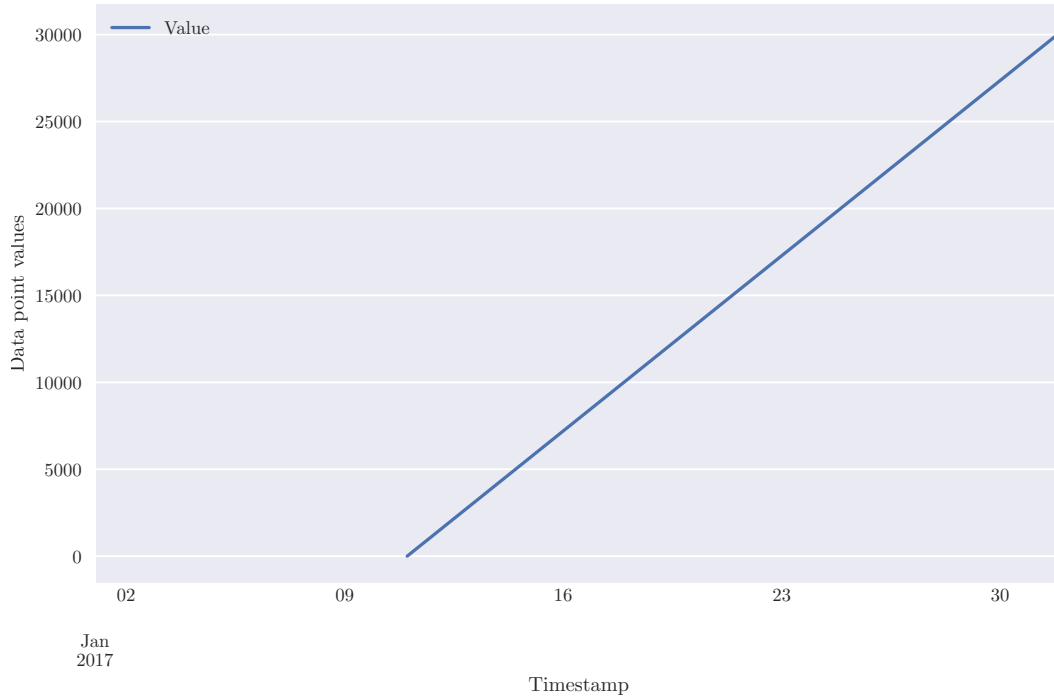
Table B.1: Expected results for each tag

Tag	Number of erroneous data points	Total number of data points expected or generated	Percentage erroneous data	Overall score (%)
AllMetrics	5760	8928	64.5	35.5%
Missing	2880	8928	32.3	67.7%
Static	1728	8928	19.4	80.4%
ExceedsLimits	$= 576 + 1152$ $= 1728$	8928	19.4	80.4%
Negative	1728	8928	19.4	80.4%
Outlier	1	8928	$= 1.12 \times 10^{-4}$ ≈ 0	99.9%

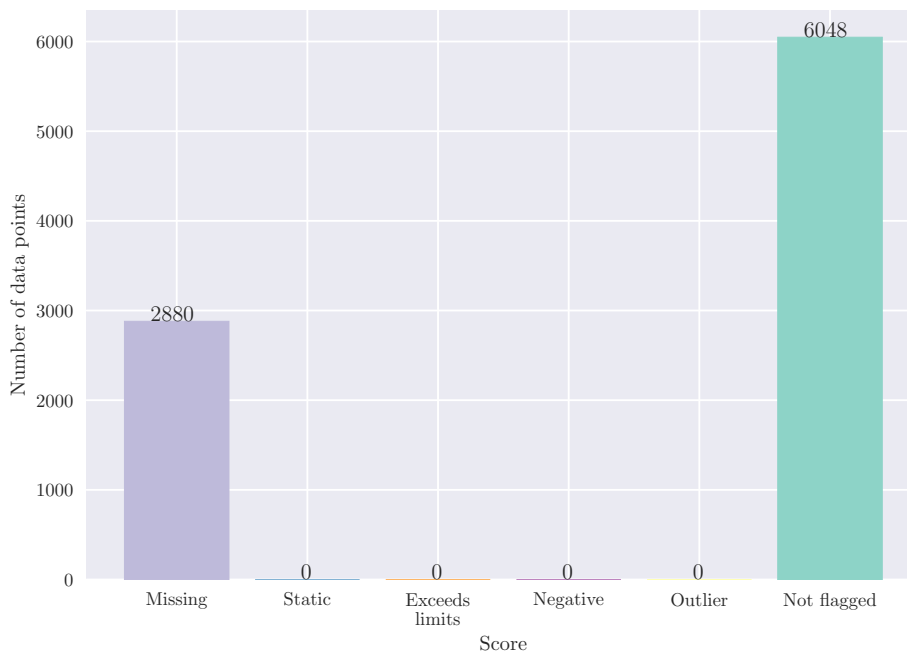
The data quality analysis results for AllMetrics tag were discussed in Section 3.5.2. The results for the remaining tags are discussed in the following sections.

B.2 Missing data results

The results from the Missing metric show that the system is able to identify missing data points. The data plot is shown in Figure B.1, which shows the values and analysis results for individual data points. From the chart it is possible to see that data is missing between 2017-01-01 00:00:00 and 2017-01-11 00:00:00.

Figure B.1: *Missing tag – Data plot*

The bar chart in Figure B.2 shows the number of data points flagged by each metric and the number of data points not flagged by any metric. In this case, the system identified 2880 missing data points. The total number of data points found in the date range is 6048. This coincides with the expected results for the Missing tag as shown in Table B.1.

Figure B.2: *Missing tag – Data point count*

The pie chart in Figure B.3 shows the ratio between the Missing metrics and the not flagged

results. As can be seen in the chart, approximately 32% of the data is missing. This coincides with the expected results in Table B.1.

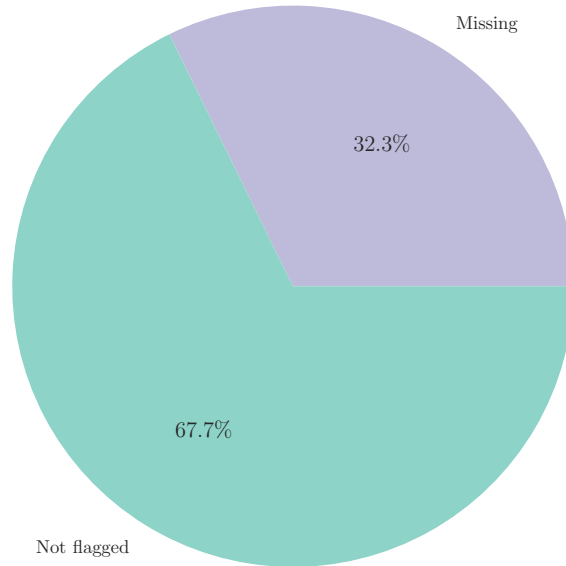
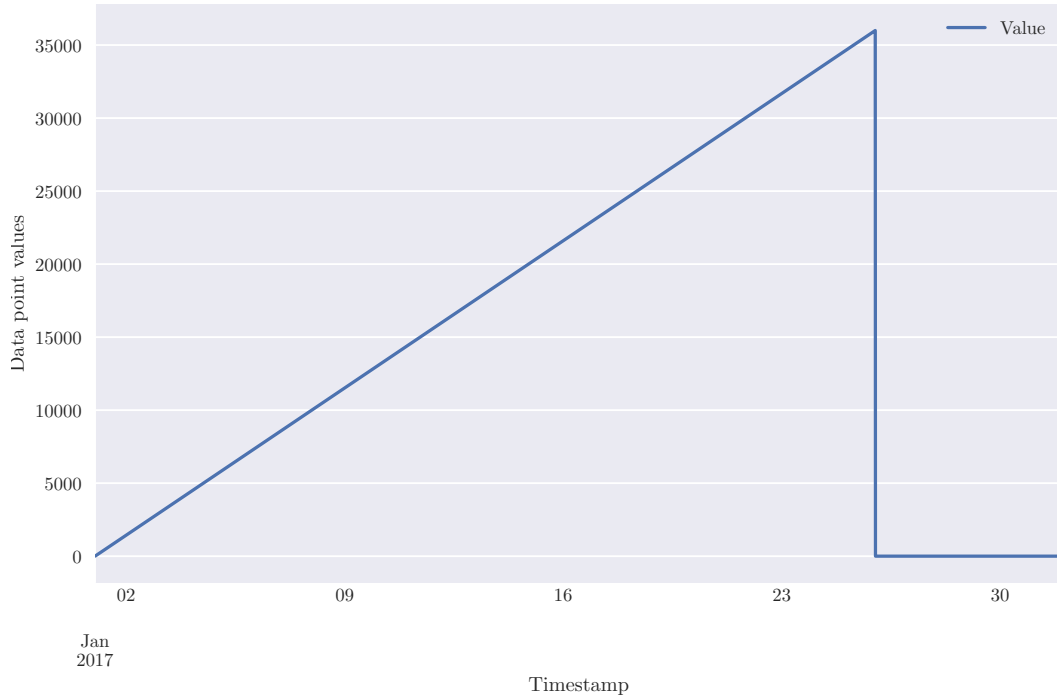


Figure B.3: *Missing tag – Pie chart*

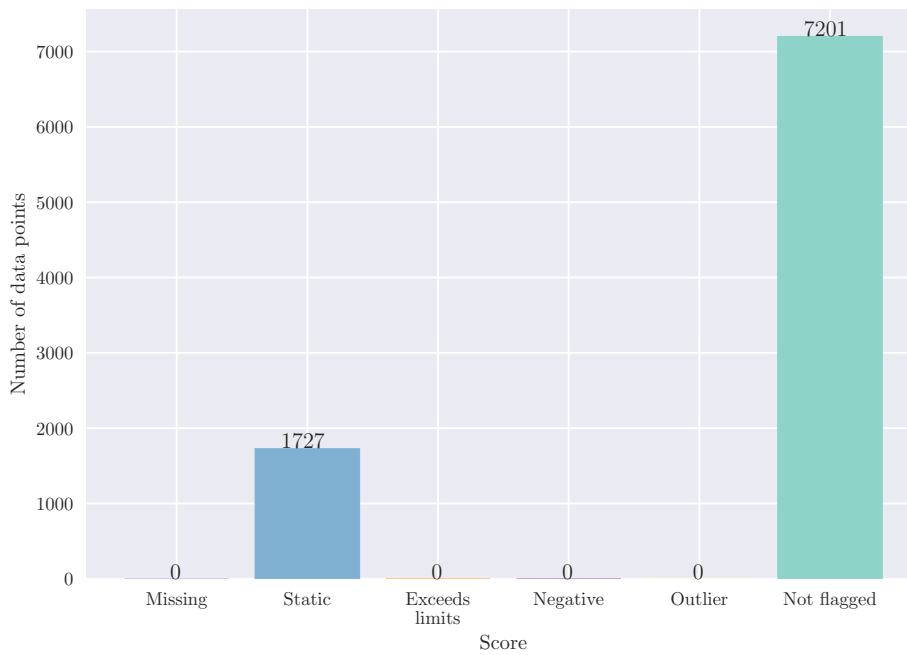
None of the other metrics are configured for this tag and thus no data points are flagged for any other metric. All the results for the tag coincide with the expected results in Table B.1; thus, the Missing metric works as expected.

B.3 Static data results

The results from the Static metric show that the system is able to identify static data points. The data plot is shown in Figure B.4, which shows the values and analysis results for individual data points. From the chart it can be seen that the data drops off after 2017-01-25 and then becomes static.

Figure B.4: *Static tag – Data plot*

The bar chart in Figure B.5 shows the total number of data points flagged by each metric. In this case, the system identified 1727 static data points. This coincides with the expected number of static data points in Table B.1.

Figure B.5: *Static tag – Data point count*

The pie chart in Figure B.6 shows the ratio between the Static metric and the not flagged

results. According to the chart, approximately 19% of the data is identified as static data points. This coincides with the expected results in Table B.1.

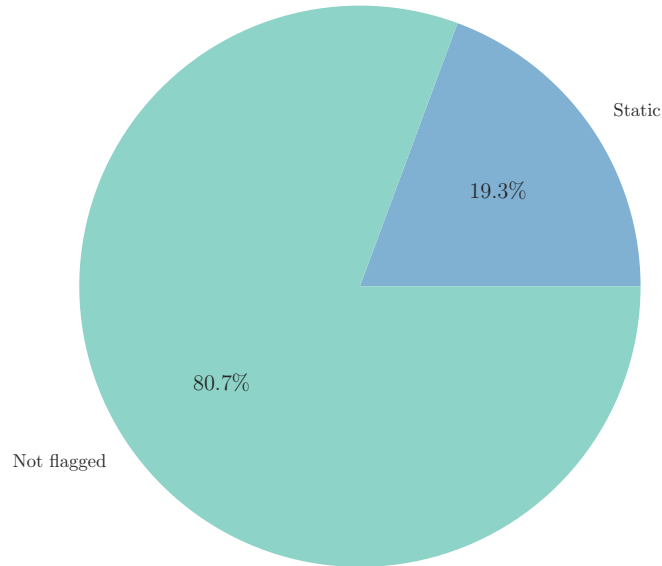
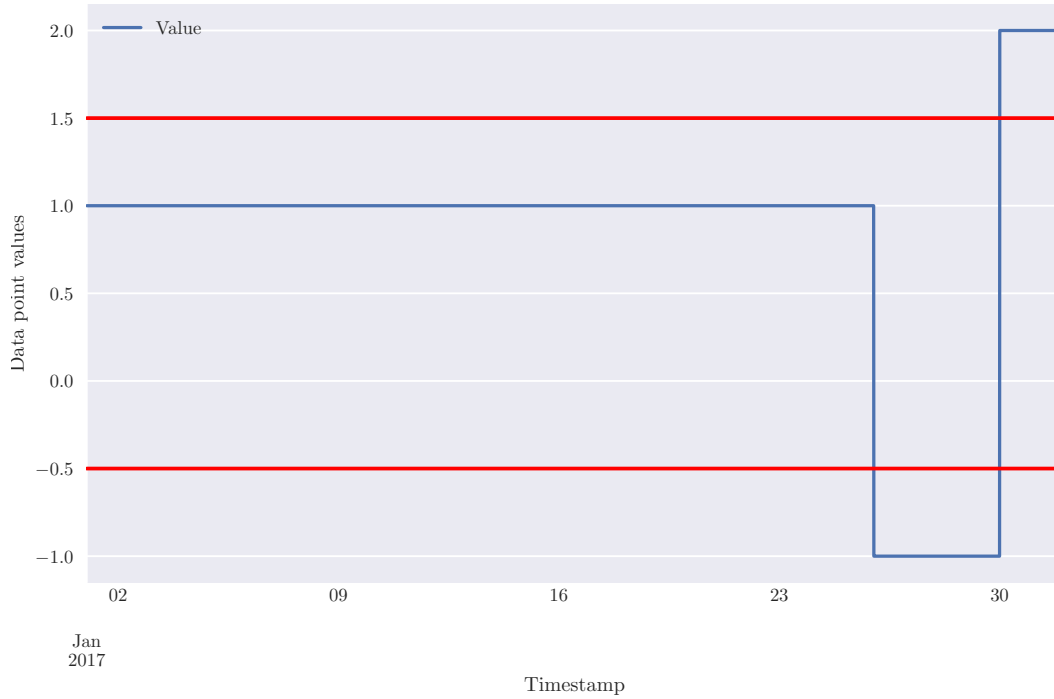


Figure B.6: *Static tag – Pie chart*

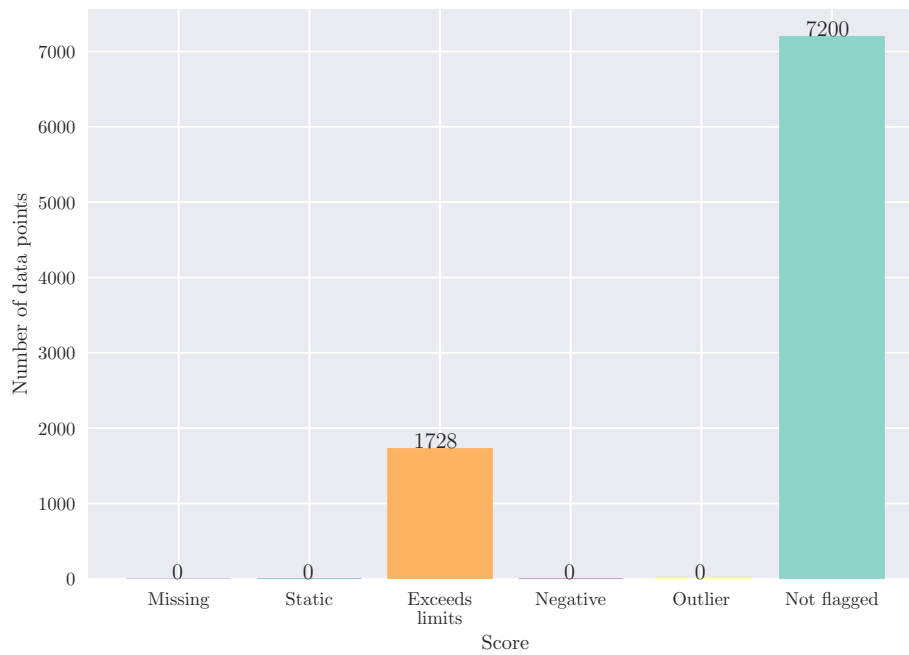
None of the other metrics are configured for this tag and thus no data points are flagged for any other metric. All the results for the tag coincide with the expected results in Table B.1; thus, the Static metric works as expected.

B.4 Exceeds limits results

The results from the Exceeds Limits metric show that the system is able to identify data points that exceed set limits. The data plot is shown in Figure B.7, which shows the values and analysis results for individual data points. From the chart it can be seen that the data drops below the lower limit on 2017-01-26, and then increases to above the upper limit on 2017-01-30.

Figure B.7: *ExceedsLimits tag – Data plot*

The bar chart in Figure B.8 shows the total number of data points flagged by each metric. In this case, the system identified 1728 data points that exceeded set limits. This coincides with the expected number of erroneous data points in Table B.1.

Figure B.8: *ExceedsLimits tag – Data point count*

The pie chart in Figure B.9 shows the ratio between the Exceeds Limits metric and the not

flagged results. From the chart it can be seen that approximately 19% of the data is identified as exceeding limits. This coincides with the expected results in Table B.1.

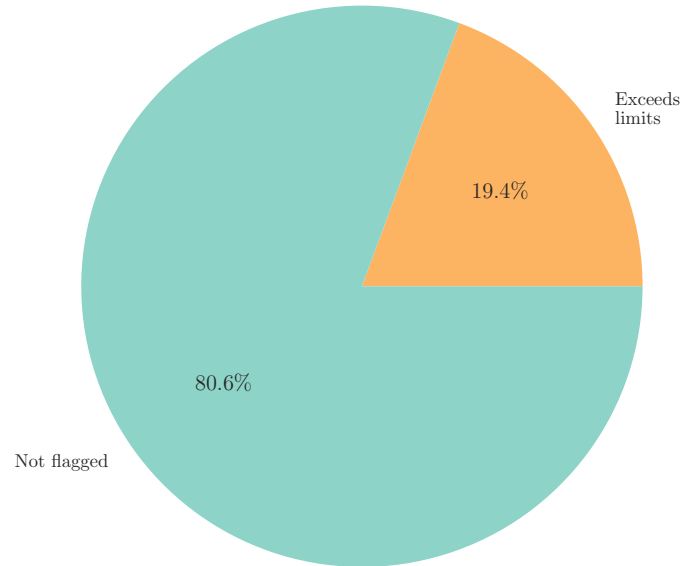
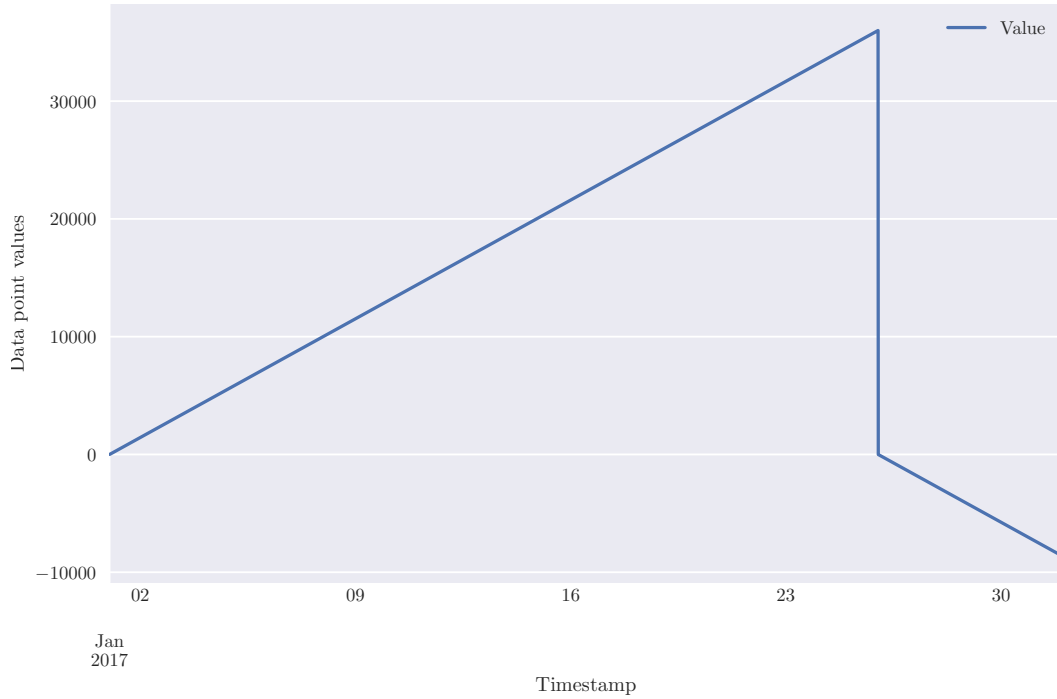


Figure B.9: *ExceedsLimits tag – Pie chart*

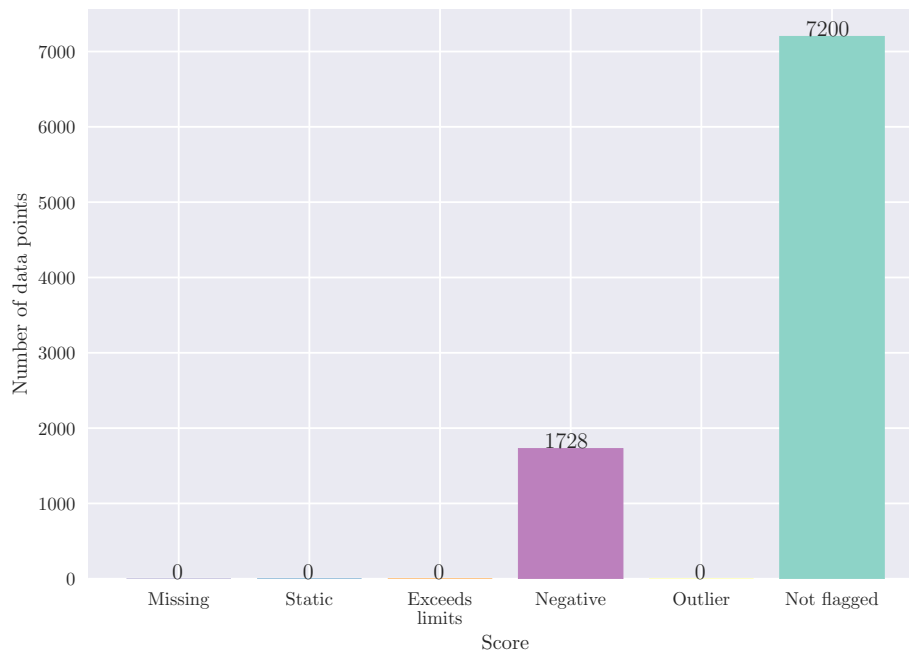
None of the other metrics are configured for this tag and thus no data points are flagged for any other metric. All the results for the tag coincide with the expected results in Table B.1; thus, the Exceeds Limits metric works as expected.

B.5 Negative data results

The results from the Negative metric show that the system is able to identify negative data points. The data plot is shown in Figure B.10, which shows the values and analysis results for individual data points. From the chart it can be seen that the data drops off and steadily decreases from 2017-01-26 onwards.

Figure B.10: *Negative tag – Data plot*

The bar chart in Figure B.11 shows the number of data points counted for the Negative metric or if the data points are not flagged. In this case, the system identified 1728 negative data points. This coincides with the expected number of negative data points in Table B.1.

Figure B.11: *Negative tag – Data point count*

The pie chart in Figure B.12 shows the ratio between the Negative metric and the not flagged

results. From the chart it can be seen that approximately 19% of the data is identified as negative. This coincides with the expected results in Table B.1.

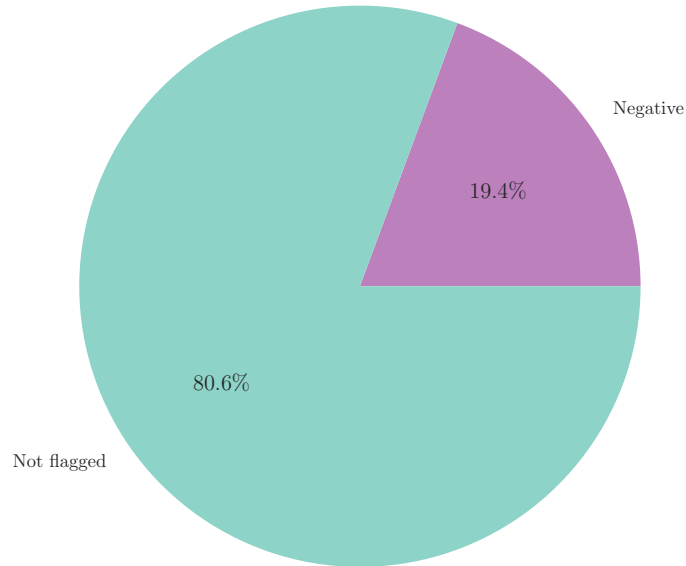
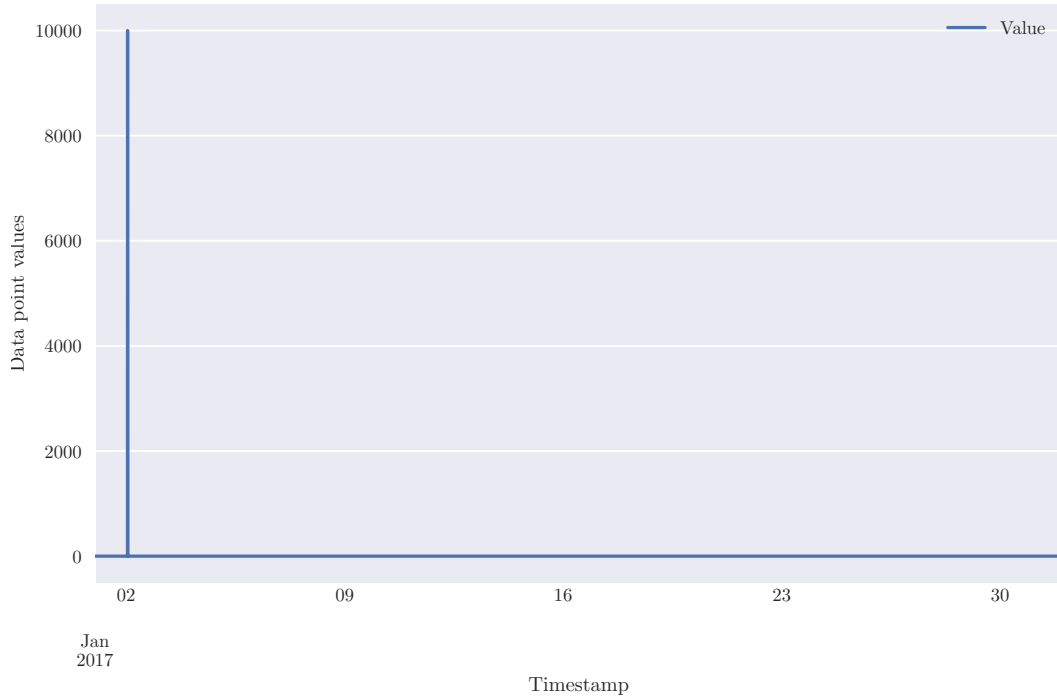


Figure B.12: *Negative tag – Pie chart*

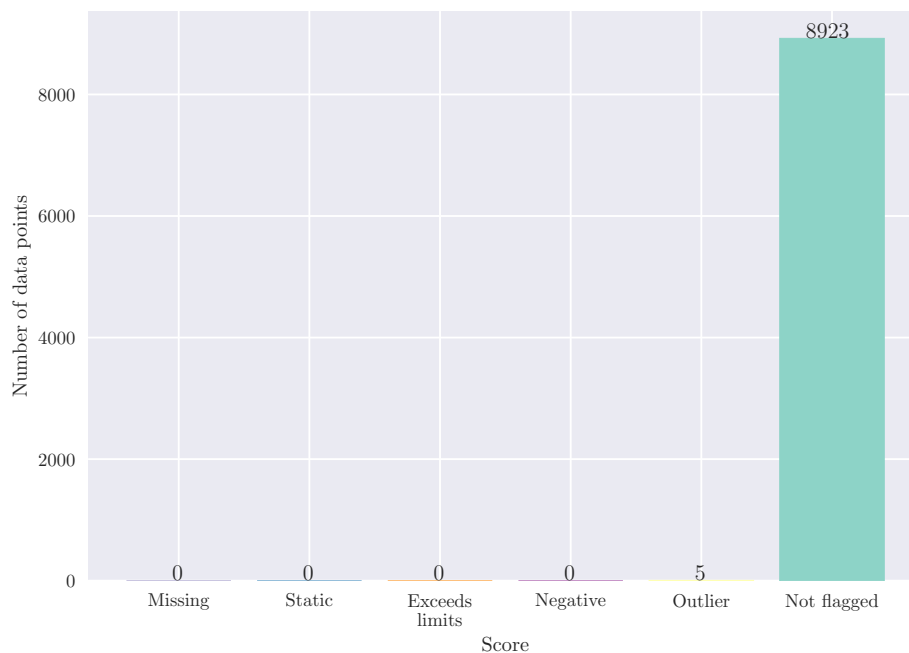
None of the other metrics are configured for this tag and thus no data points are flagged for any other metric. All the results for the tag coincide with the expected results in Table B.1; thus, the Negative metric works as expected.

B.6 Outlier results

The results from the Outlier metric show that the system is able to identify outliers. The data plot is shown in Figure B.13. The chart shows the values and analysis results for individual data points. From the plot it can be seen that one outlier was present.

Figure B.13: *Outlier tag – Data plot*

The bar chart in Figure B.14 shows the total number of data points counted for each metric and if the data points are correct. In this case, the system identified five points as outliers. The expected results are only one outlier for the tag. This means that four extra data points were identified as outliers. The investigation revealed that the *NearestNeighborsRegressor* and the *SVM* methods identified the four data points surrounding the outlier as outliers as well.

Figure B.14: *Outlier tag – Data point count*

The pie chart in Figure B.15 shows the ratio between the Outlier metric and the not flagged results. In this case almost all of the data is correct. From the chart it can be seen that only 0.1% of the data are identified as negative.

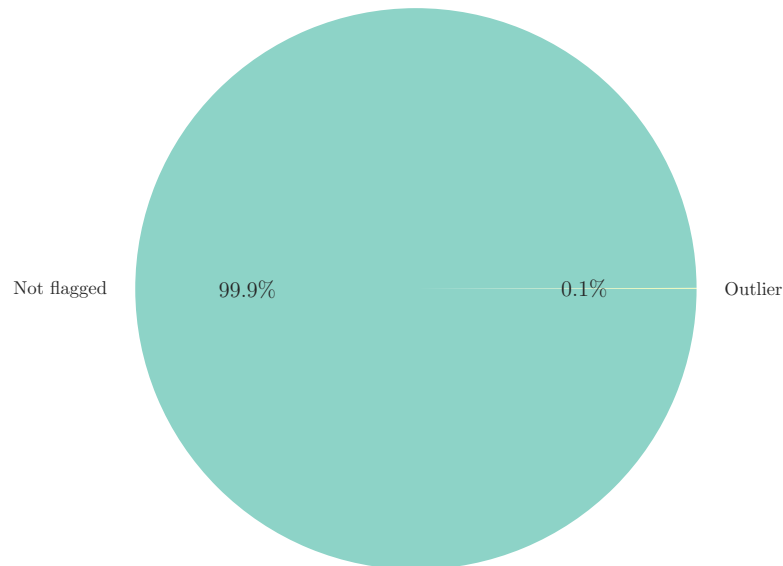


Figure B.15: *Outlier tag – Pie chart*

None of the other metrics are configured for this tag and thus no data points are flagged for any other metric.