Validating the integrity of single source condition monitoring data

JN de Meyer

orcid.org/ 0000-0002-9044-4523

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

Supervisor: Dr JF van Rensburg

Graduation: May 2020

Student number: 23452250

Acknowledgements

Firstly, I would like to thank God for helping me accomplish this goal. Without His blessing, I would not have been able to reach this point.

To my wife, Marcia, thank you for your love and support during this time. Your support means the world to me and I would not have been able to get through the long nights without you. You are my greatest motivation and allow me to accomplish more than I ever would have thought possible.

To my parents, thank you for all your sacrifice to make this a reality. I would not have made it this far if not for all the love and support. Thank you for always believing in me, encouraging me to keep working, and setting the best example for me to follow.

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

To Dr. J. F. van Rensburg and Dr. J. van Laar for all the time you put into helping me write up my dissertation.

To Dr. J. du Plessis and Mr. P. Goosen, thank you for your guidance and welcoming working environment.

To all my friends who helped me get through this, thank you for your support and encouragement.

Abstract

Vast amounts of data are generated daily and play an important role in decision-making and performance evaluation. Ill-informed decisions can have costly, negative effects. For this reason, confidence in the data used is important.

In the mining industry, bad decisions could lead to equipment failure and production losses, resulting in large financial implications. To minimise these risks, preventative measures are implemented.

Condition monitoring can reduce production losses by monitoring the status of equipment and determining maintenance needs. This minimises down-time and keeps the equipment in good operating condition. To optimise the efficiency of this process, the data needs to have high integrity.

Methods exist to validate the integrity of data. These methods differ depending on the context and use of the data. In literature, the need to validate the data integrity of single source condition monitoring data was identified.

A system was designed to calculate the data integrity of data streams in context to one another. The system was designed generically so it can be applied to any company which follows a described data layout. The system makes use of contextual data to classify each data point as having high or low integrity.

Generic temperature and vibration profile data sets were created using historical data from four major component types, namely fridge plants, compressors, fans, and pumps.

The system was verified with a clean data set in order to calibrate it. Subsequently, a test data set with manually introduced errors was used to judge the accuracy of the system.

The system was implemented on a deep-level mine case study to validate the integrity of the data for eight different components located across six different sites. The system was able to classify data integrity accurately, with some limitations being identified. It was seen that the case study had a recurring problem with data loss and faulty metering equipment.

The additional benefits of identifying human error in component configuration and faulty measurement equipment identification were identified. These benefits added value to the system.

Future work was recommended to address the limitations of the current system. It was recommended that calculating the data quality of individual data streams, verifying the power consumption measurements, taking transition states into consideration, and calibrating the profile data per component will increase the accuracy of the system. Incorporating the system results into existing condition monitoring systems and implementing a notification system to inform users of results will decrease the time to correct low integrity data.

The developed system met all the study objectives identified and succeeded in classifying the data integrity of single source condition monitoring data. The accuracy of the system can be improved by including an existing system which calculates the data quality of the individual data streams. Implementing a notification system which uses the results of the developed system could reduce the number of human errors in component configuration and increase the rate at which faulty equipment is repaired.

Keywords

Data integrity, Condition Monitoring, industrial data, data analysis, contextual data, single source data

Table of Contents

List of figures

List of tables

Nomenclature

Table 1: Nomenclature

Acronym	Definition
ZB	Zettabyte
CBM	Condition-Based Maintenance
CM	Condition Monitoring

Chapter 1 Introduction

1.1) Background

Vast amounts of data are generated daily. By 2010, around 1 ZB[1] of data had been generated [1]. In 2011, this amount had increased to 1.8 ZB, with manufacturing industries contributing close to 2 EB[2] [1], [2]. Modern industries, as a whole, currently generate more than 1 000 EB of data annually [3]. It was predicted that by 2020 the total amount of generated data would have increased to more than 35 ZB [1].

This was due, in large part, to technological advancement [4]. One of the most notable advancements was in the area of sensor technology [3]. Sensors can now be found in most electronic devices [5]. As of 2012, about 2.5 EB of data was generated daily, with the amount doubling every 40 months [6].

This data is used for a variety of applications, such as environmental monitoring, industrial applications, and business- and human-centric pervasive applications[1], [7]–[9]. If done correctly, big data can deliver a competitive edge as valuable insight can be extracted from it [9], [10]. However, the risk of low data quality increases as companies manage more extensive and complex information resources [11]. The large amount of data allows for data-driven decision-making, which leads to more timely decision-making [1], [6], [12]. Data-driven decision-making refers to making use of analytics data to promote more effective insights [8].

The quality of data influences the accuracy of these decisions [12]. Poor quality data carry various negative effects, which include [13]:

- less customer satisfaction,
- increased operational costs,
- inefficient decision-making processes,
- lower performance, and
- lowered employee job satisfaction.

It is, however, difficult to estimate the extent of the monetary implications companies with low data quality experience. Companies typically overestimate the quality of their data and underestimate the cost of errors [13]. A company’s overall profit potential is affected when management regards data as being strategic, knowing some data is faulty, and not seeing the

[1] Zettabyte – 10^{21} bytes
[2] Exabyte – 10^{18} bytes

costs. Over 50% of companies are not confident in their data quality, with only 15% being very confident in the quality of externally-supplied data [13], [14].

This is problematic for companies who provide monitoring services as they are responsible to ensure that their clients remain in an optimal operational state. Clients rely on these services to remain profitable in tough economic times [15]. As an example, the mining sector is under pressure [16], [17]. In the South African mining industry, this can be contributed to various factors, including increases in labour and electricity costs [18], [19]. In order to remain profitable, expenses need to be minimised [20], [21].

In industrial processes, data quality can be affected by failing equipment [22]. Improving maintenance effectiveness is a potential source for financial savings [7]. One method of improving maintenance efforts is by implementing condition-based maintenance. CBM requires multiple sensors to be installed per piece of equipment [23]. Using the data generated by the sensors, along with the understanding of how the equipment operates, problems are detected before they escalate [7]. These problems can then be rectified with minimal impact to the company and prevent equipment from failing. This helps preserve the quality of the generated data [22].

Making recommendations based on faulty data may lead to serious consequences, such as misguided decisions and increased workload of human operators [23]. A lack of factual data could lead to as much as 33% of maintenance costs being wasted [7]. This is problematic as up to 40% of a large company’s operational budget is spent on maintenance, such as a mine [7].

Validating the integrity of sensor data before analyses could reduce erroneous decisions [24]. The largest portion of current approaches is rooted in machine learning and statistical models. However, these methods neglect that in the context of industrial equipment, additional information on the system exists [23]. By using this additional information, the data can be put into context, easing the process of validating the data integrity [25].

1.2) Data integrity & quality

Data integrity is the completeness of data compared to the integrity of the objective world, requiring all data values to be in an objective and true state and not absent [26]. In the context of this research, data integrity can be simply defined as the trustworthiness of data.

Decisions made based on low integrity data can have major monetary implications [13]. Thus, making decisions based on high integrity data instils confidence. As a result, data sets with low integrity have little to no use [12].

Data quality can be defined as the fitness or suitability for use of data [4], [8]–[12]. Alternatively, data quality dimensions can be used to characterise data [28], [29]. Combining the discussed characteristics in [4], [8]–[12] , Table 2 was compiled.

Table 2: Data quality dimensions

Dimension	Description
Accessibility	Is the data easy to access, use, and retrieve?
Accuracy	Is the data correct, objective, reliable, certified, and validated?
Availability	Is the data physically available?
Believability	Is the data true and credible?
Completeness	Is the collection complete?
Compliance	Does the data comply with regulatory and industry standards?
Consistency	Is the data consistent and presented in the same format?
Integrity	Is the data coherent?
Objectivity	Is the data unbiased, unprejudiced, and impartial?
Relevance	Is the data applicable to the task at hand?
Reliability	Is the data correct and reliable?
Timeliness	How long does it take between the change of a real-world state and the resulting modification of the information system state?
Validity	Is the data within acceptable parameters?

Using a combination of the data quality dimensions defined in Table 2, data quality can be constructed. The dimensions used will depend on the type of data.

Data integrity and data quality are sometimes used interchangeably. Data integrity and quality are not the same, though they are connected. Data quality can be seen as the building blocks used to make up data integrity, i.e. quality data produces trustworthy knowledge [24]. In short, data integrity refers to whether a data value is in a true state [26], while data quality refers to the degree to which data conforms to user’s specific requirements in a given context [30]. In order to determine the integrity of data, the data quality can be calculated and used as an indication.

As mentioned, depending on the type of data being analysed, different data dimensions will be used to determine the data quality. Studies that make use of these dimensions in practical applications are summarised in Table 3.

Table 3: Data quality dimensions on which focus was placed in other studies

	Mine condition monitoring	**Energy consumption**		**Condition-based maintenance**	**On-line monitoring**
Dimensions	[31]	[25]	[32]	[7]	[33]
Accessibility				✓	
Accuracy	✓	✓	✓	✓	✓
Availability	✓	✓	✓		
Believability		✓	✓		
Completeness	✓	✓	✓	✓	
Compliance		✓	✓		
Consistency				✓	✓
Integrity					✓
Objectivity					
Relevance		✓	✓	✓	
Reliability		✓	✓		
Timeliness					✓
Validity	✓	✓	✓		

The studies listed in Table 3 focused on different types of data. It is made clear that some dimensions can be used to quantify data quality for a greater range of data types, whilst others are rarely used. From this, it can be deduced that the following dimensions should be possible to calculate for most data types:

- Accuracy
- Availability
- Completeness

- Validity

However, when determining the trustworthiness of data, the following three dimensions should take priority as they stem directly from the definition of data integrity:

- Integrity
- Believability
- Reliability

The abovementioned dimensions can be calculated with relative ease, depending on the amount of data sources available. When multiple data sources are available, calculating the dimensions are relatively trivial, with implemented methods detailed in [7], [25], [32], [33]. This does, however, become more difficult when only a single data source is available.

1.3) Single source data

When a limited number of data sources are available for calculations, quantifying the quality of data becomes tedious. According to Gous et al. [25], it is vital to use multiple data sources to calculate the reliability and believability dimensions. Comparing different data sources identifies discrepancies in the data. These discrepancies negatively impact the overall integrity of the data set [25], [34].

Hayes et al. [34] discussed how Hill et al. [35] used a Bayesian detector algorithm to detect anomalies in single source data streams and how it did not use contextual data.

One way of identifying discrepancies in single source data is by calculating the accuracy, availability, completeness, and validity dimensions [25]. Doing so without taking data into context will quantify the data quality per data source, and not the data set as a whole. This can lead to falsely identified discrepancies [31], [34].

To address this, the integrity dimensions can be investigated and combined with the quality dimensions. This can be done by calculating the quality of each data stream, then adding context to those streams and calculating the quality of the streams in relation to one another. This is demonstrated in Figure 1 on the next page.

Only looking at individual data streams for quality will give insight that looking into contextual integrity might not identify, and vice versa [34]. For an optimal classification of the integrity of data sources, both the individual streams and the contextual data set need to be considered and compared. Goosen [31] investigated the quantification of individual data stream quality in a related study.

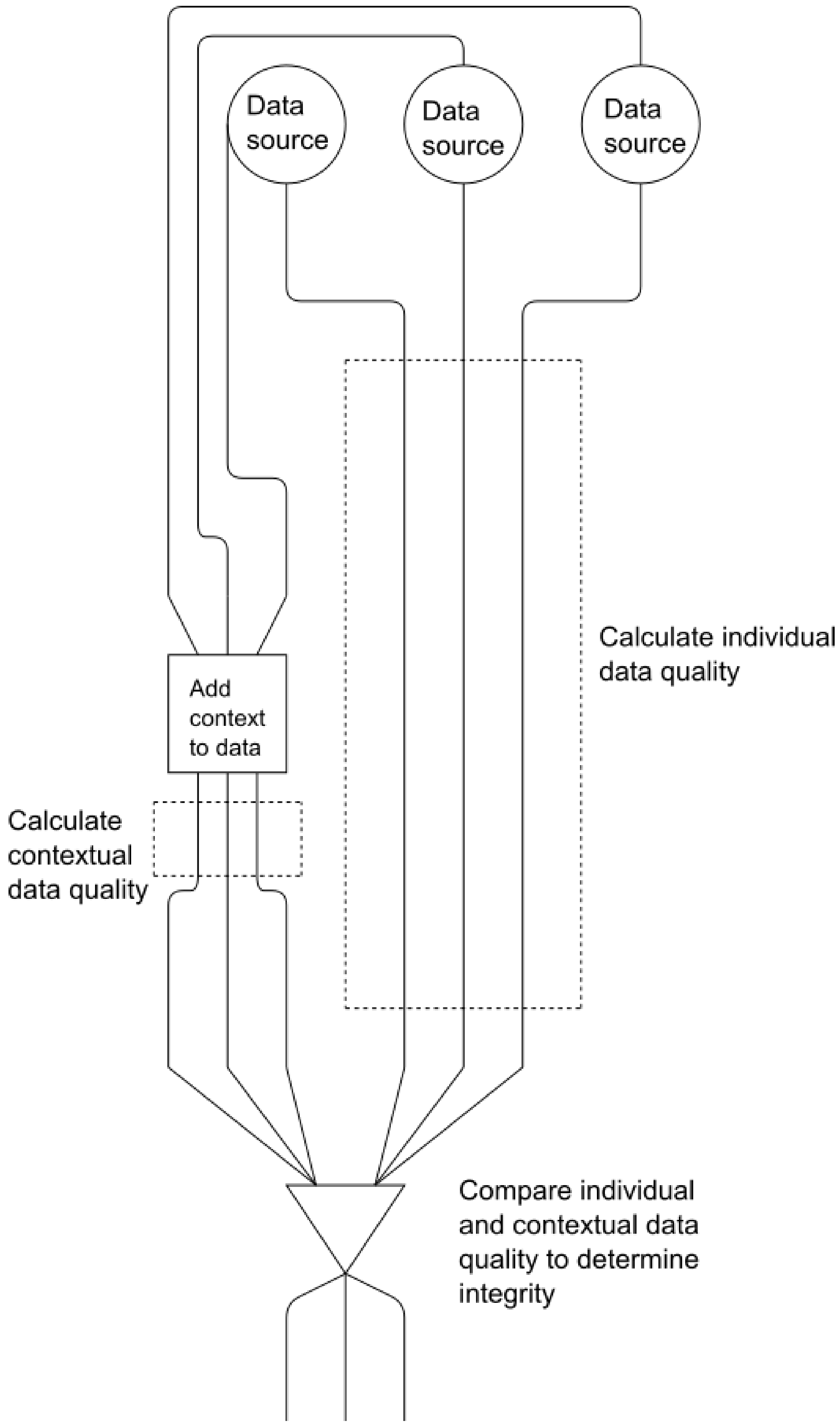

Figure 1: Calculating contextual data quality for a single data source

1.4) Condition-based maintenance

Condition-based maintenance (CBM) is a maintenance strategy in which equipment undergo maintenance based on their condition [15], [23], [36]–[39]. CBM focusses on fault detection, component diagnostics, degradation monitoring, and failure prediction [15], [38]. This helps with the identification and solving of problems in advance [38], [39]. CBM is used to reduce the risk of equipment failure before scheduled maintenance could be performed [38]. CBM attempts to avoid unnecessary maintenance tasks by only performing maintenance actions when there is evidence of abnormal behaviours [36].

CBM is divided into three steps, namely data acquisition, data processing, and maintenance decision-making [15], [36]. The data acquisition step obtains operational data from equipment. This data is then processed using various methods. The methods used vary depending on the type of data and what answers are desired. The processed data is then used to plan maintenance schedules and requirements.

Condition monitoring (CM) is a technique in which measurement equipment is installed onto equipment, measuring the operational conditions [39]. It is the process of constantly monitoring the state of the equipment [39].

The purpose of CM is to collect equipment condition data which can be used to detect incipient failure [12], [15], [39]. CM also assists maintenance supervisors with fault diagnostics and prognostics [15].

CM is considered a central part of CBM as it is used in the data acquisition step [39]. Not only does CM help with the optimisation of maintenance schedules, it also increases knowledge of failure cause and effect along with deterioration patterns [39].

If CBM is properly established and effectively implemented, maintenance costs can be significantly reduced [36], [39]. This is accomplished by reducing the number of unnecessary maintenance operations [36], [39]. Benefits observed in case studies implementing CBM are summarised in Table 4. Unless indicated otherwise, the results in Table 4 were obtained by comparing an implemented CBM system to a planned maintenance system.

Table 4: Summary of benefits obtained from implementing CBM

Benefit	Impact	References
Reduced maintenance costs	29 – 75%	[37]
	25 – 30%	[40]
Reduction in production losses	20 – 25%	[40]
	50% for wind turbines	[41]
	1 – 3% for gas turbines	[42]
	5% for oil and gas industry	[43]
Breakdown elimination	70 – 75%	[40]
Increase in return on investment	Mentions CBM maximises return on investment	[40]

From Table 4 it can be deduced that CBM adds significant value when it is adopted. In industry systems, any product damage can lead to serious results, making CBM an attractive method for high-valued assets [38]. However, high data integrity is desired for CBM [25]. As discussed in Chapter 1.2), there are various methods to determine data integrity. However, as discussed in Chapter 1.3), determining the data integrity of a data set becomes more complicated when only a single data source is available. In the mining industry, CBM mostly makes use of single source data. For more CM data sources to become available, more measurement equipment will need to be installed [40]. Other types of data sources, such as invoices for power consumption and calibration records for measurement accuracy to name a few, are also not always available [25]. However, with enough single source data and knowledge of the operational layout and working, relations between data can be established, putting the data into context [25].

1.5) Contextual data

Data quality is subjective to the data used and the context [7]. Thus, determining the quality of data requires a wider approach that includes data context and representation, not only technical aspects [7].

Better understanding of an industrial process can lead to more efficient monitoring and control [44]. Contextual data will add more meaning and value to CM data [1]. As CM data are time-dependent values, calculating their quality will be difficult without context [31].

Contextual data is used in monitoring systems to identify certain phenomena and react to specific events. These events are related to critical aspects such as the violation of predefined constraints [45]. Contextual data may introduce new information which diminishes or enhances the abnormality of anomalies [34].

A platform was developed by Goosen et al. [31] to quantify the quality of industrial data and, as described in the results, often miscalculated the quality due to the lack of contextual knowledge. Contextual data can enhance computer systems and applications by broadening the input in comparison to classical standalone solutions [44]. Incorporating contextual knowledge will ease the validation of data integrity, especially when only single data sources are available.

By doing so, models can be created. These models include all relevant context to data points. Relevant context depends on the type of data and its application. In the case of CM data, models can be created by including data points from different data streams from the same timestamps, e.g. temperature, vibration, energy consumption. These models should include the following context-related data as proposed for industrial process monitoring [44]:

- Expert knowledge – understanding of how different data streams related and influence one another, and
- Peripheral sensor inputs – sensor values that represent different measurements. These vary depending on the application.

1.6) Need for study

A literature study was conducted in which related literature was identified. Literature was found by using relevant keywords including, but not limited to:

- Condition Monitoring,
- Data Quality,
- Data Integrity,
- Single Source Data, and
- Contextual Data.

The literature found in accredited search portals was further limited to studies that were written in the past 15 years and focus was placed on journal articles and conference proceedings.

The literature identified was read and categorisation criteria could be formulated as follow:

1. Was the data integrity validated?
2. Did the study have access to only a single data source per data stream?
3. Did the study look at the data in context?
4. Was the study done using CM data?

It was seen that data integrity is often validated for stored data in databases, and rarely done for CM data as focus is rather placed on data quality. This lead to two of the categorisation criteria, namely 1 and 4. As stated in section 1.1), sensor technology has seen a lot of advancement in recent years. This leads to multiple sensors being installed in operations to obtain as much information as possible. However, typically, multiple sensors aren’t installed to measure the same characteristics. This lead to the criteria of whether multiple sensors or different data sources (such as billing records) were available (2) and whether multiple data streams were put into context (3).

A research matrix was compiled and can be seen in Table 5.

Table 5: Literature review

	Data			Application
Reference	**Integrity Validation**	**Single source**	**Context**	**Condition Monitoring**
[33]	✓		✓	
[7], [12]	✓			✓
[23]		✓	✓	
[13], [25], [32]	✓		✓	
[46]–[48]	✓	✓	✓	

Hongxun et al [33] looked into data integrity classification for on-line monitoring systems. They investigated methods to validate the integrity of data streams in context, which is applicable to this study. However, they did not assess the integrity of CM data and they had multiple data sources available to help with the integrity validation.

Ratnayake et al [12] suggested an empirical approach to quantify the integrity of CM data. They had multiple data sources which were used to validate the integrity. The CM data streams were not considered in context.

Madhikermi et al [7] identified the various factors contributing to low integrity data. They conducted two case studies in which both made use of CM data. Various data sources were used to obtain the CM data. These data streams were not put into context and were rather analysed independently.

Solomakhina et al [23] quantified the quality of single source data by analysing data sets which put the data streams into context. They analysed manufacturing data generated by complex machinery during operation.

Haug et al [13] defined an optimal data maintenance effort by quantifying the data quality of contextual data streams by using the data quality characteristics discussed in Table 2.

Gous et al [25] evaluated the quality of energy consumption data from deep-level mines by compiling contextual data sets. They identified the impacts of common data anomalies and their possible causes.

Hamer et al [32] developed a practical approach to quantify tax incentives for South African industry. In his approach, he quantifies the quality of energy consumption data by evaluating contextual data sets.

Decker et al [46] quantified the integrity of stored data in a database. Single source data was compiled into contextual data sets on which their analysis was performed.

Hiremath et al [47] suggested an auditing technique to quantify the integrity of data stored in the cloud environment. The study uses meta data as context to more accurately calculate the data integrity.

Tan et al [48] developed an auditing service to verify the integrity of data stored in the cloud computing environment. This service makes use of meta data to add context when calculating the integrity of stored data.

From the research matrix, it can be seen that few studies focus on CM data in context. There are also limited studies in which the data integrity of CM data is validated. In most studies where data integrity was calculated, despite contextual data being used, multiple data sources are available. This leads to the need for the study, namely to validate the integrity of single-source CM data by making use of data context.

1.7) Problem statement & objectives

From the literature reviewed, a few shortcomings have been identified in the related research. These shortcomings are:

- Few studies focus on single source data,
- From the studies that focus on data integrity and use contextual data, very few apply this on CM data, and
- None of the above studies consider single source CM data.

Combining the shortcomings identified above led to the problem statement, namely that a need exists to validate the integrity of CM data when no supporting data streams are available.

As stated earlier, Goosen [31] investigated the quantification of individual data stream quality. For this study, however, the focus will be placed on the quantification of integrity for data streams in a contextual data set. It will be assumed that data used by the system will be of a high quality. This assumption is made as data quality calculations and corrections form part of data preparation pre-processes. This allows the focus to be placed on the data integrity calculations.

To satisfy the need for the study, the following objectives should be addressed:

- Develop a software system to validate the integrity of CM data. This system should make use of data models in its decision-making [45].
- Create models that represent the different pieces of equipment found in the CM network. These models should be representative of the equipment on a basic level. They should adhere to basic rules and limitations experienced by the actual component [34], [44].
- Link data streams as inputs to the models. This will give context to the data by creating links between the various data streams [44].
- Calibrating the models requires the basic rules and limitations to be tuned. This will enable the model to behave similarly to the actual component. If the models are not calibrated, the analysis results will be of little to no value.
- Determine data integrity using the calibrated models, calculating the integrity of the data streams.
- Flag low integrity points so they can be further examined. This will create a record of low integrity points which can be used to find errors in the sensor network.

1.8) Dissertation layout

Chapter 1: Introduction

This chapter contains all of the research for the dissertation. The background, Chapter 1.1, painted a picture of a problem. This picture was then expanded by the literature review, Chapters 1.2 – 1.5. In Chapter 1.6, a need for the study was identified by comparing literature. Chapter 1.7 constructed a problem statement and corresponding objectives for the dissertation. Chapter 1.8 concluded Chapter 1 by defining the layout for the dissertation.

Chapter 2: Design of software to validate data integrity

Chapter 2 describes the design process and methodology followed to address the problem identified in Chapter 1.7. This chapter will provide more details on a possible solution to the problem, along with the method verification and implementation.

Chapter 3: Results

Chapter 3 contains the results after the solution derived in Chapter 2 was implemented on a case study. The results are interpreted and discussed, validating that the problem identified was addressed and concludes with a summary of the results.

Chapter 4: Conclusion

Chapter 4 discusses the dissertation as a whole and ends with recommendations for future work and the closing statements.

Chapter 2
Design

2.1) Introduction

As stated in Chapter 1.7), there is a need to validate the integrity of single source CM data. This chapter describes the system, as well as the methodology used to create it, to address this need.

In Section 2.2) the system requirements are discussed. These requirements give a guideline to how the system should work. This section also describes the data flow and how measurements are taken.

Section 2.3) describes the methodology used to address the problem defined in Chapter 1.

Section 2.4) describes the design process to address the issue identified in the previous chapter. This includes the design of models and system.

Section 2.5) describes how the methodology, models, and system are verified by implementing them on a test case.

2.2) System requirements

There are three main shortcomings identified in Table 5, namely:

- Little focus on single-source data,
- Contextual data is not used to determine data integrity, and
- Data integrity is not applied widely to CM data.

To address these issues, a system will be designed to validate the integrity of single source CM data. In order for the system to do this, some high-level constraints to the system determined from the literature study and Table 5 can be created, namely it should:

- be generic so it can be applied to multiple case studies,
- only use single source data as inputs,
- use contextual links to determine data integrity,
- save the results for future use,
- indicate low integrity data.

2.2.1) Data layout

A data layout used throughout the rest of the document will be defined here. This data layout refers to the structure in which data is grouped. This layout is used to give context to different data streams in relation to one another. This layout is generic and consists of six levels, of which the most important levels are the bottom two (discussed in Section 2.2.2). The data layout was developed by combining the company structures described by van Jaarsveld et al [15] and Goosen et al [31] and is displayed in Figure 2.

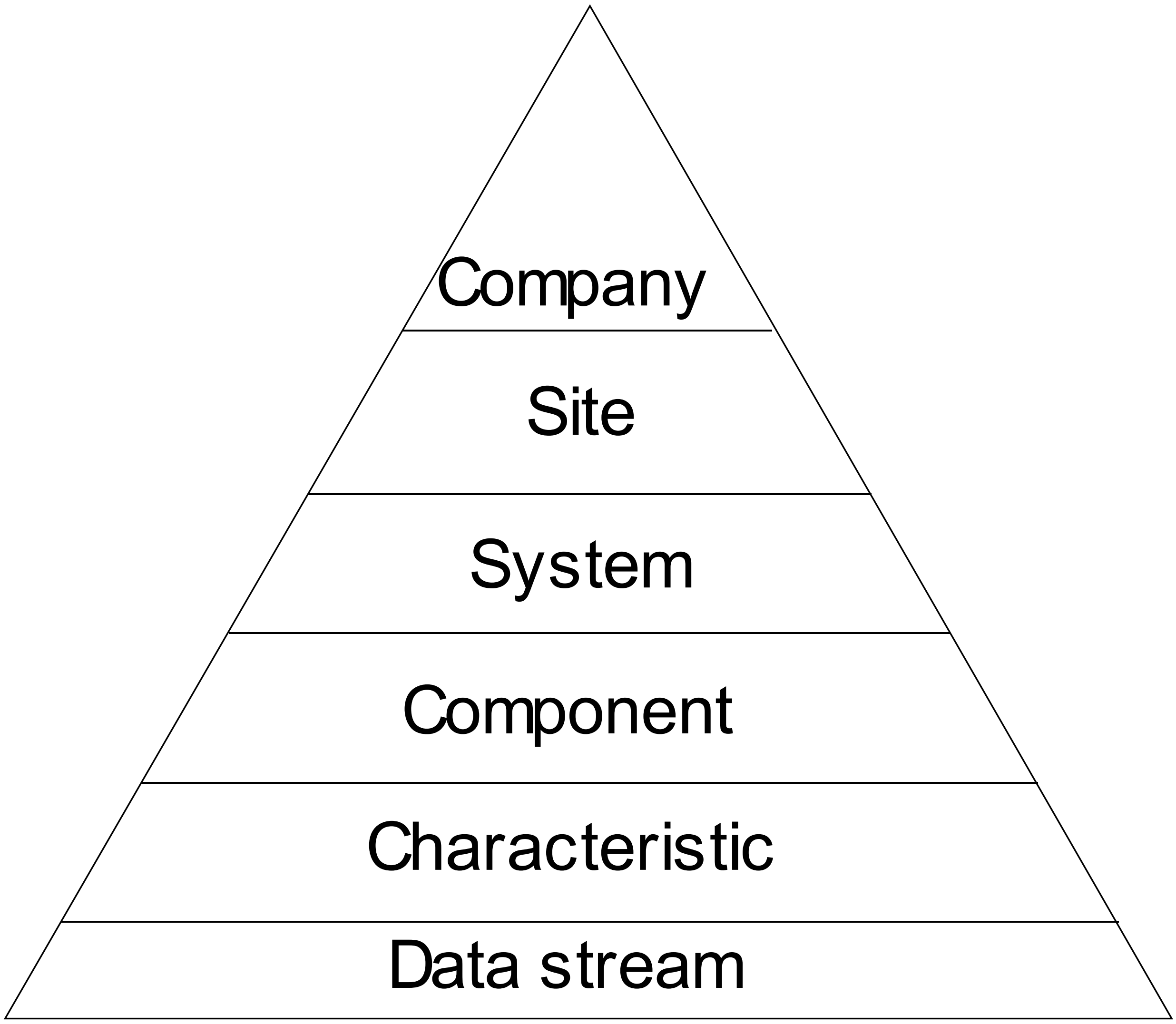

Figure 2: Generic data layout of condition monitoring setup

In the generic data layout, every row can consist of multiple instances linked to the row above it e.g. each site has multiple systems, which in return have multiple characteristics.

A detailed explanation of the rows in Figure 2 are provided below.

- Company: This is the top level of an organisation e.g. Mining company A.
- Site: A facility which belongs to the company e.g. Mining operation located in South Africa.

- System: The grouping of similar sections / equipment on a site e.g. the refrigeration system responsible for providing cooling to the site.
- Component: A piece of equipment in the system e.g. a fan.
- Characteristic: The available measurement types e.g. vibration.
- Data stream: The individual measurements from equipment e.g. vibration measurement of a specific bearing.

The generic layout described above was compared to the structure of different companies. This ensured that the layout was representative of most companies. Some of the company structures used for comparison include:

- Mining companies,
- Bakeries,
- Restaurants, and
- Manufacturing companies.

2.2.2) Measurements

A general data flow describing how data is captured and transferred to a database in the cloud environment is displayed in Figure 3.

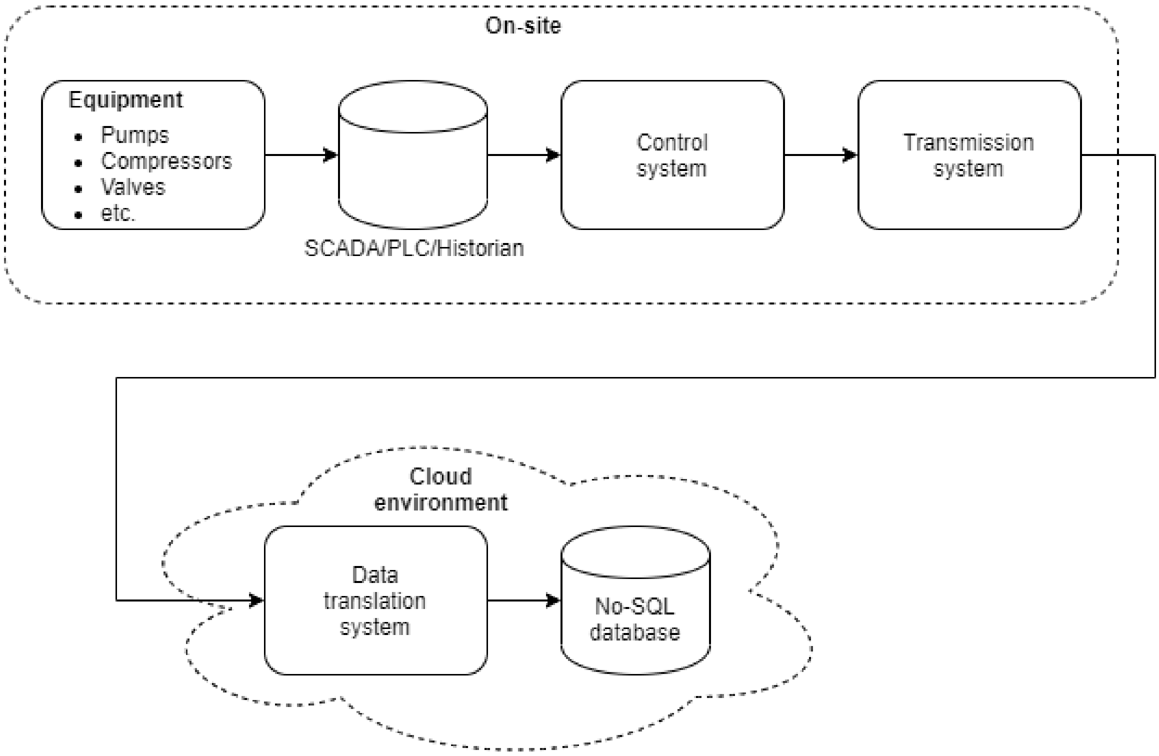

Figure 3: Image describing data flow from point of capture to database

This data flow identifies two distinct parts, namely on-site and the cloud environment. On-site describes the environment where the physical equipment is located which is monitored. A SCADA[3] system is used to capture the measurements for each component. The data resolution can differ between two minutes and daily samples depending on the component measured and sensors available. The SCADA sends the measurements to the on-site automated control system. This data is then sent to a transmission system which sends it to the cloud environment. In the cloud environment, the data is received by a translation system which translates the data into a usable format and stores it in a No-SQL[4] database from where it will be used.

[3] SCADA – Supervisory Control And Data Acquisition
[4] No-SQL – Non-relational database

There are four common measurements that are collected from equipment. These four measurement types are linked to each component and their integrity will be calculated by the system by putting them into context.

- Running status: a measurement which indicates whether a piece of equipment is running or not. It is represented by a Boolean value, meaning it simply shows if something is “on” (1) or “off” (0).
- Power consumption: the amount of power used by a piece of equipment. This can be used to help determine whether equipment is running or not in the event that the running status is unreliable or unavailable. It is also used to give context to the other measurements. This is measured by either current (A[5]) or power (kW[6]), depending on where the component has sensors available.
- Temperature: the current temperature of a piece of equipment at a specific measurement point. This is used to determine safe operating conditions for the equipment and will also be used to give context to the other measurements. For this study, temperature will be measured in ^{o}C.
- Vibration: the amount of vibration experienced by a piece of equipment. When equipment is running, vibration will be generated. This is used to determine safe operating conditions for equipment and will also be used to give context to the other measurements. For this study, vibration will be measured in mm/s[7].

[5] Ampere – measurement of current
[6] kW – unit of power (k = 10^3)
[7] mm/s – the amount of millimetres equipment moves in one second

2.3) Methodology

In order to address the objectives identified in Section 1.7), the diagram in Figure 4 was created and will be followed.

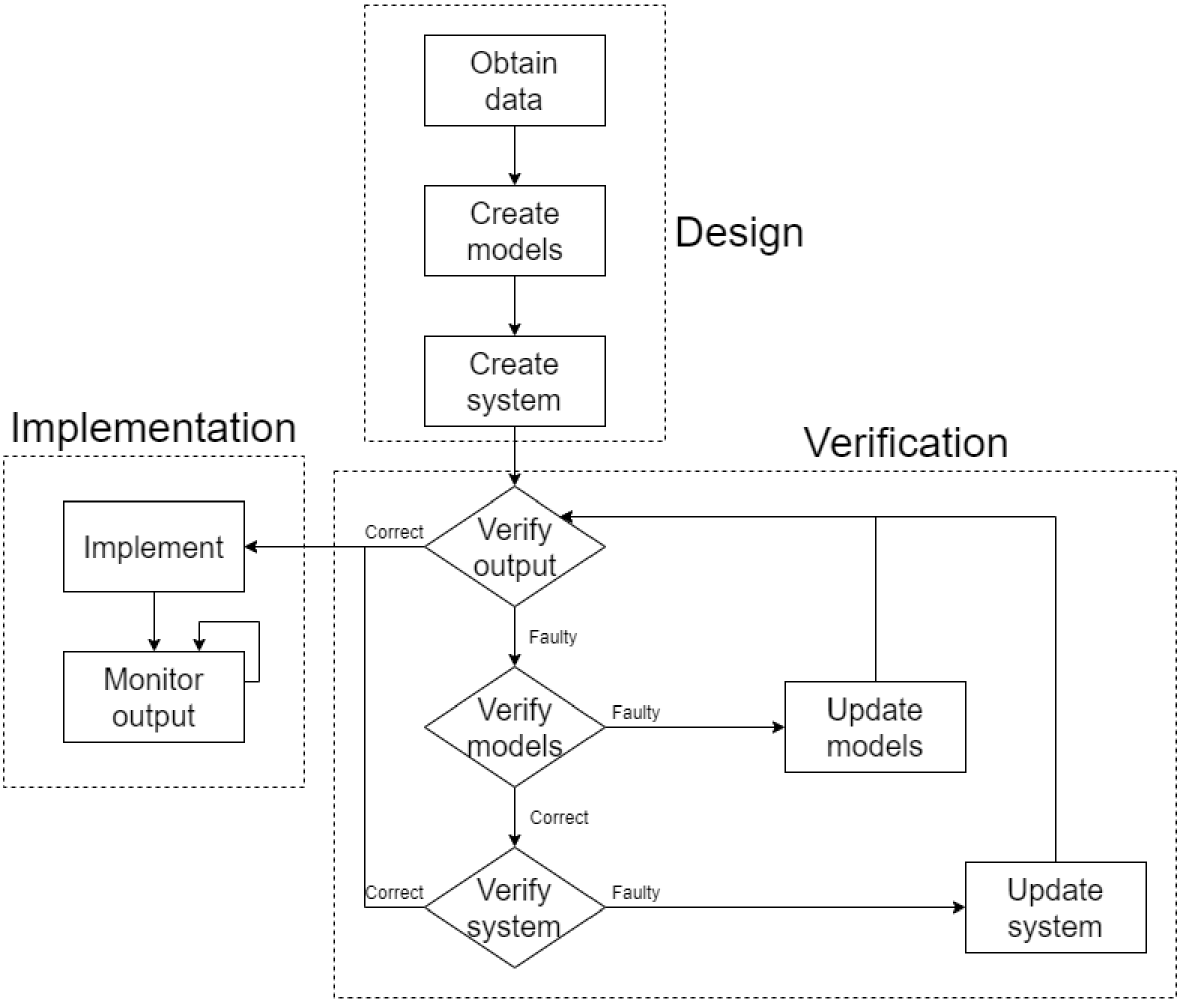

Figure 4: Image describing the methodology to be used

The above methodology identifies three main sections, namely design, verification, and implementation. The design section entails obtaining CM data to be classified as either high or low integrity data points, the creation of the models and behavioural limits which will be used to mimic an actual component, and the creation of the system which will classify the integrity of the obtained data by using the models and behavioural limits.

The verification section involves calibrating the system and models to accurately classify the integrity of data points. This is done by using test data to ensure that the results of the system are as expected.

The implementation section will see the system implemented on a case study to classify the integrity of actual CM data. The goal of implementing the system on a case study is to identify low integrity data points and their causes. Design and verification are discussed in Sections 2.4) and 2.5) respectively, while the implementation is discussed in Chapter 3).

2.4) Design

In this section, the design block in the methodology will be discussed. For this study, focus will be placed on the following four components, namely refrigeration systems, fans, pumps, and compressors as they are commonly found in the mining industry. These components are often monitored to ensure they operate effectively and inside their operational limits.

2.4.1) Obtain data

As discussed in 2.2.2), data is saved in a No-SQL database in the cloud environment. These measurements are saved as BSON[8] documents in the No-SQL database. These BSON documents are divided into two categories, namely tags and value documents. Tag documents contain all relevant data of the measuring point, whilst value documents contain a measurement for the measuring point at a specific timestamp. An example of a tag document is displayed in Figure 5.

```
{
   _id: ObjectId("123456789012345678901234"),
   Name: "Measuring point 1",
   Created: ISODATE("2019-01-01T00:00:00")
}
```

Figure 5: BSON document representation of a tag document

A tag document consists of three fields, which are discussed in Table 6.

[8] Binary JSON file – File type used in No-SQL databases such as MongoDB to store data

Table 6: Breakdown of tag document fields

Tag document field	Explanation
_id	An auto-generated field which keeps track of the entries in the database. This is the primary identifier of the entry.
Name	Describes the measuring point for which value documents are recorded.
Created	Timestamp of when the measuring point was added

The abovementioned table describes all of the fields found in a tag document which is stored in the No-SQL database. An example of a value document is displayed in Figure 6.

```
{
    _id: ObjectId("98765432109876543210043210"),
    TagId: ObjectId("12345678901234567890 1234"),
    Timestamp: ISODATE("2019-01-01T00:30:00"),
    Value: 52.3,
    DataQualityAnalysisResults:
    {
        DI: True
    }
}
```

Figure 6: BSON representation of a value document

A value document consists of five fields, as described in Table 7.

Table 7: Breakdown of value document fields

Value document field	Explanation
_id	An auto-generated field which keeps track of the entries in the database. This is the primary identifier of the entry.
TagId	Reference to the _id field of a tag document for which this value was recorded.
Timestamp	Timestamp of when the measurement was recorded.
Value	Recorded measurement value.
DataQualityAnalysisResults	Dictionary that keeps track of different data quality calculations

The abovementioned table describes each field found in a value document which is stored in the No-SQL database.

Each tag document can be linked to multiple value documents, but not the other way around. As seen in Figure 6, value documents contain a field to indicate the data integrity of the data point, indicated by DI. This field can have a value of either True or False, indicating whether the data point is of high or low integrity.

A currently implemented CM system is displayed in Figure 7.

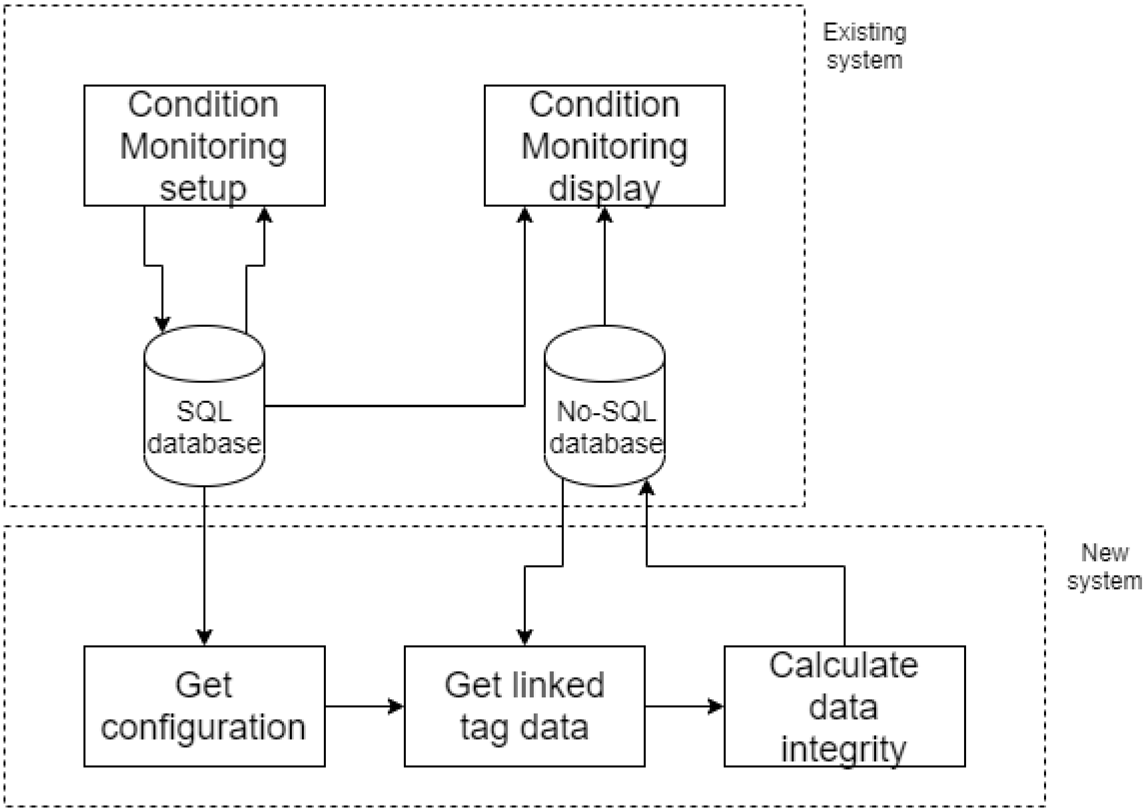

Figure 7: Adapted existing CM system implementing new data integrity system

Figure 7 shows an existing CM system which can be modified to include a data integrity calculation system. The existing system consists of a GUI[9] CM setup website which interacts with a MySQL[10] database to store configuration data for the CM display website. The CM display website uses the configuration data from the MySQL database to retrieve the relevant measurement data from a No-SQL[11] database. This data is used to display the condition of monitored equipment by means of a coloured grid and graphs. The new system works in a similar way to the CM display website as it also retrieves the configuration data from the MySQL database. The configuration data is used to retrieve the relevant measurement data from the No-SQL database. The measurement data is used to calculate the integrity of each data point and is written back to the No-SQL database.

The system makes use of two different database types as each one is more suited for a different role. A MySQL, or relational, database is ideal for giving structure to data. This makes it easier to keep track of how data relates to other data, thus it is used to save the configuration

[9] Graphical User Interface
[10] Relational database
[11] Non-relation database

settings. No-SQL, or non-relational, databases are used to store massive amounts of data. Normally, No-SQL databases have no structure and simply store raw data.

For data to be put into context, configuration data of a component is required along with the measurement data. The configuration data is used to determine which data stream represents which characteristic. The new system will allow the specification of data stream context. This will be done through a user interface as described later in 3.1).

In order to add the new system to the existing one, some adaptions need to be made. The setup menu and relational database tables will need to be altered in order to link a data stream as a specific characteristic to a model.

The updated configuration data can now be used to get the correct data stream from the No-SQL database, link it to a model, and calculate the data integrity of each data point. The result of the data integrity analysis is then saved in the data integrity field of the value document to be used at a later stage.

2.4.2) Models

The models represent each component being monitored and are used to give context to individual data streams.

All of the components that will be modelled follow some similarities in their operation. This is due to various factors, such as physical limits, laws of nature, and operating environment, to name a few. The similarities are represented as equations and are specific to this study. These similarities are described below.

- Power consumption: All of the components used require energy to be operational, resulting in equation 1

  $$S_{on} = E_{cons} > 0 \tag{1}$$

  where S_{on} is the running state of the component and E_{cons} is the power consumption of the component, measured in either kWh or A depending on which is available. Simply speaking, if the component is consuming energy, it is running.

- Temperature increase during operation: If a component is in operation, its temperature will increase due to electrical energy being converted to thermal energy. However, the temperature will never exceed a certain operating limit (specific to each component).

These limits are specified by the specification sheets of the component and the operator in charge of its operation. These limits should be implemented into the components' control system to ensure safe operating conditions. These two statements are used to create equation 2

$$T_{HComp} = t_{On} \times k_{HComp} + T_{Amb} \leq T_{CUL} \qquad (2)$$

where T_{HComp} is the current temperature of the component whilst in operation, t_{On} is the current total time that the component was "on", k_{HComp} is a temperature gain constant for the specific component, T_{Amb} is the natural temperature due to the ambient temperature of the component when not operating, and T_{CUL} is the upper operating temperature limit of the component above which the control system switches off the component.

- Temperature decrease after operation: After a component was in operation, its temperature will gradually decrease over time as the heat is transferred into the environment. This temperature can never go below a certain threshold due to the environmental factors applied to it. These two elements are combined to create equation 3

$$T_{CComp} = t_{Off} \times k_{CComp} + T_{Amb} \geq T_{CLL} \qquad (3)$$

where T_{CComp} is the current temperature of the component after operation has ended, t_{Off} is the current total time that the component has been switched off after operation, k_{CComp} is a temperature loss constant for the specific component, T_{Amb} is the natural temperature due to the ambient temperature of the component when not operating, and T_{CLL} is the lower temperature limit of the component due to the environment in which it is located.

- Positive values: Power consumption and run status cannot be less than 0.

- Whilst in operation, components experience vibration: Due to rotating / moving parts, components will experience vibrational forces. These vibrations can be detrimental to a components' working condition. Vibrational forces in surrounding areas can also influence the component. These two statements are combined to form equation 4:

$$V_{On} = V_{Env} + V_{Comp} \leq V_{CL} \qquad (4)$$

where V_{On} is the current total vibration experienced by the component, V_{Env} is the vibration experienced by the component due to other sources in the area, V_{Comp} is the vibration generated by the component itself, and V_{CL} is the upper limit of vibration experienced before the component is switched off by the control system.

For all of the components, the upper and lower limits defined in equations 1 – 4 are specific to the component or control system. These limits are defined per component in the component configuration interface, as discussed in 2.4.1).

Figure 8 describes the basic inputs of each model, namely the four main measurements discussed in Section 2.2.2).

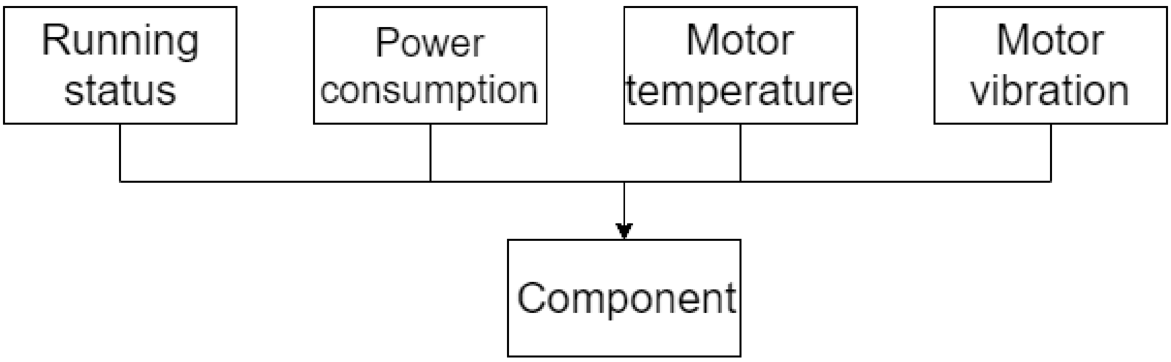

Figure 8: Schematic of the four inputs each model has

These models are generic, so they can be used for all components. To calibrate the models, the limits and constants should be updated according to the specific component which is being modelled [44]. These updated values can be verified by running a sample data set through the system and inspecting the results.

2.4.3) System

As discussed in 2.4.1), an existing system can be adapted to use the new system. The CM display in Figure 7 uses configuration data saved in a relational database to obtain the linked data from a No-SQL database. This data is used to generate graphs and tables to give both an overview and detailed view of the different components according to the data layout described in Figure 2. The new system can be implemented to verify the integrity of the data before the data is displayed in the dashboard. The system uses the same configuration saved in the SQL database to get the corresponding data for each component. This data is then linked to a model which is used by the system to calculate the integrity of the data. The results of the analyses are written back into the No-SQL database so it can be used in the display dashboard.

The system flow is displayed in Figure 9.

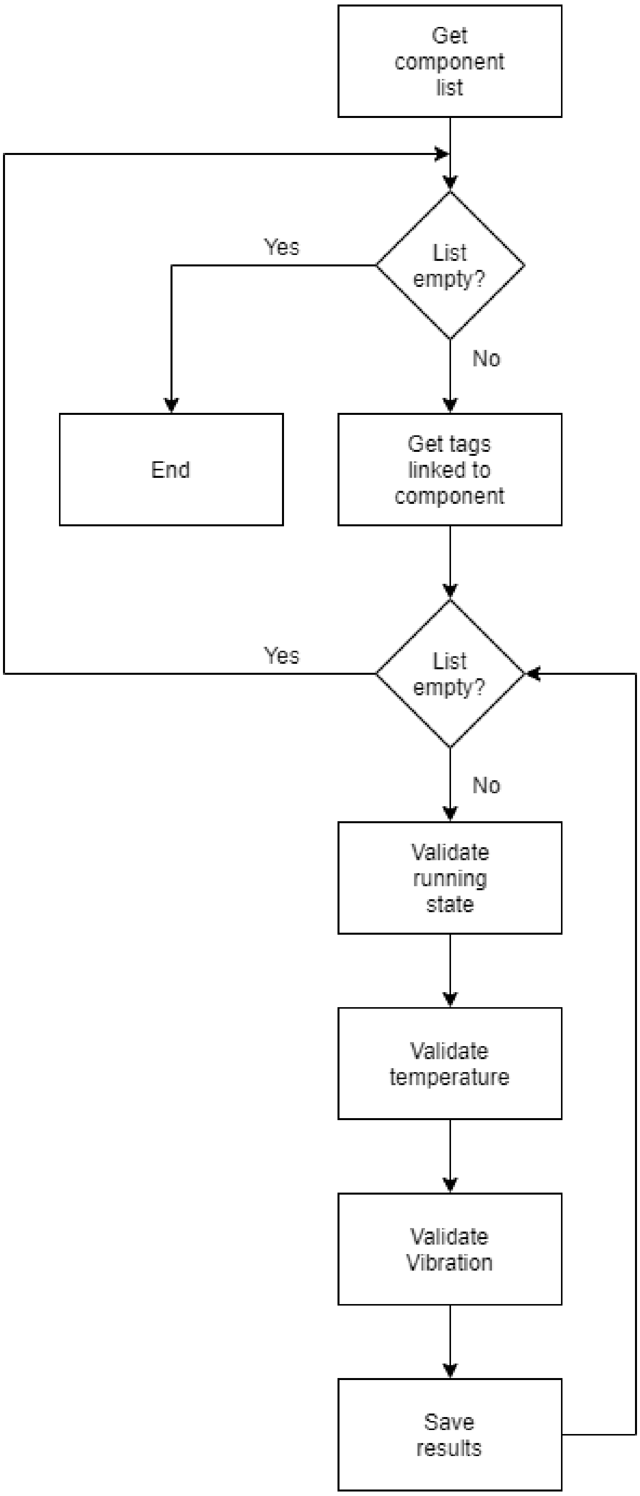

Figure 9: Image describing system flow to validate data integrity

The system described above will use the relational database to get a list of components. If there are no components configured, the system will end processing. For each component listed, the linked characteristics will be retrieved. If no characteristics are linked to the component, the system will continue to the next component. Once a component with linked characteristics is found, the four characteristics’ integrity will be evaluated. This continues until components have been analysed.

The running state for the component at the given time will be determined along with the integrity of the running status and power consumption values. The running state will then be used to evaluate the integrity of the temperature and vibration measurements.

After all the characteristics have been evaluated, the results will be saved in the No-SQL database so they can be used in the CM dashboard at a later stage. The result of the analysis for a characteristic, further referred to as the data integrity score, can be described as a Boolean output, namely each data point will be flagged as true when the data point has high integrity and false when it has low integrity.

Running state describes whether a component is in use. Each component has a running status value which indicates whether the component is running (1) or not (0). A power consumption value is also available for the component which indicates how much energy the component uses at a given point in time. Using the combination of these two values and equation 1, Table 8 was created where F stands for false and T stands for true.

The truth table uses two inputs and returns three outputs. The two inputs are the run status value and power consumption values at a given point in time. The three outputs are written in the form XYZ, where X is the running state, Y is the running value correctness, and Z is the power consumption value correctness. Each output is described below:

- Running state: This indicates whether the component was operational at the given point in time.
- Running value correctness: This indicates whether the value for the component running status at the given point in time seems correct or not.
- Power consumption value correctness: This indicates whether the value for the power consumption at a given point in time seems correct or not within operational limits.

Table 8: Truth table to classify run state, run status, and power consumption

	Run status value				
Power consumption value		null	< 0	0	> 0
	null	FFF	TFF	FTF	TTF
	< 0	TFF	TFF	TFF	TTF
	0	FFT	FFT	FTT	FFT
	> 0	TFT	TFT	TFT	TTT

It is important to note that the following assumptions were made when creating the truth table:

- Power consumption is always correct: If a component uses energy, the component is "on". This means that power consumption is regarded as more reliable than the running status value. The truth table changes when the running status value is seen as more reliable.
- Power consumption is calibrated: If a component uses no energy, the reading is 0. This is important due to the previous point. If the run status value was seen as more reliable, the run status values should be calibrated.
- Running status transitions are "on": When a component is busy starting up or shutting down, the running status value can be between zero and one. The system treats these stages as if the component is running.
- Negative is impossible: Neither the running status nor the power consumption can be negative. These are seen as impossible values and will always be flagged as false. However, this assumes that the sensor equipment measurements are reversed and treats the component as if it was in a running state.

The method used to validate the temperature measurement at a given point in time for a component is described in Figure 10.

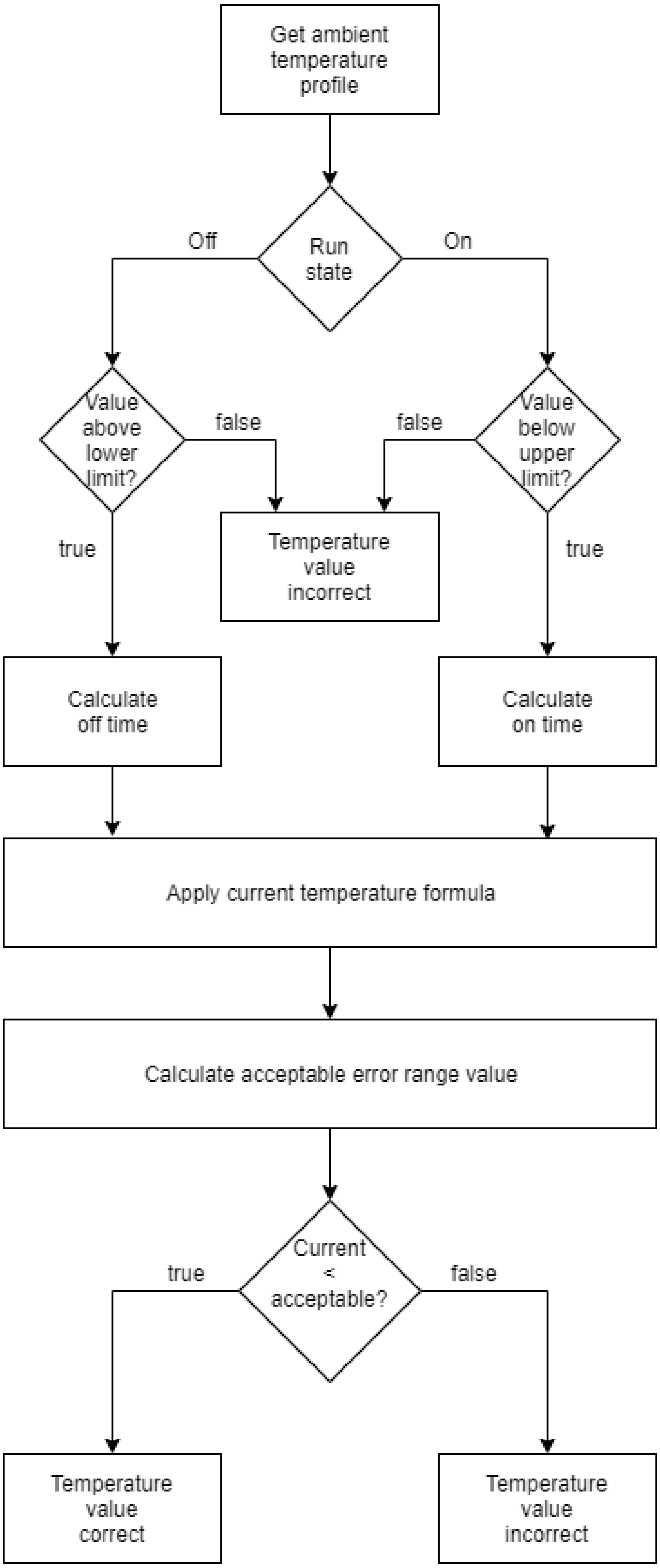

Figure 10: Image describing process of validating temperature values

The method follows a similar approach for a component in both the “on” and “off” state. An ambient temperature profile (Appendix A) is retrieved from the database to be used in calculations. The system then checks whether the component was “on” or “off” at the given time. This is used to determine which calculations need to be done.

If the temperature exceeds the applicable limit (equations 2 and 3), the temperature is inaccurate. If the value is within the limits, the total time the component was in the current operation state is calculated. Equations 2 or 3 (whichever is applicable for the current point) is used to calculate an “ideal” current temperature.

An acceptable error value range is calculated from this “ideal” value by making use of the following equation

$$T_I - (T_I \times E_C) \leq T_{IR} \leq T_I + (T_I \times E_C) \qquad (5)$$

where T_I is the ideal temperature calculated from Equations 2 or 3 (as discussed above), E_C is an acceptable error percentage for the measured value to deviate from the calculated ideal value and T_{IR} is the ideal temperature range. This range is then compared to the current temperature value. If the current temperature falls within this range, the value seems accurate, otherwise it is inaccurate.

The method used to validate the vibration measurement at a given point in time for a component is described in Figure 11.

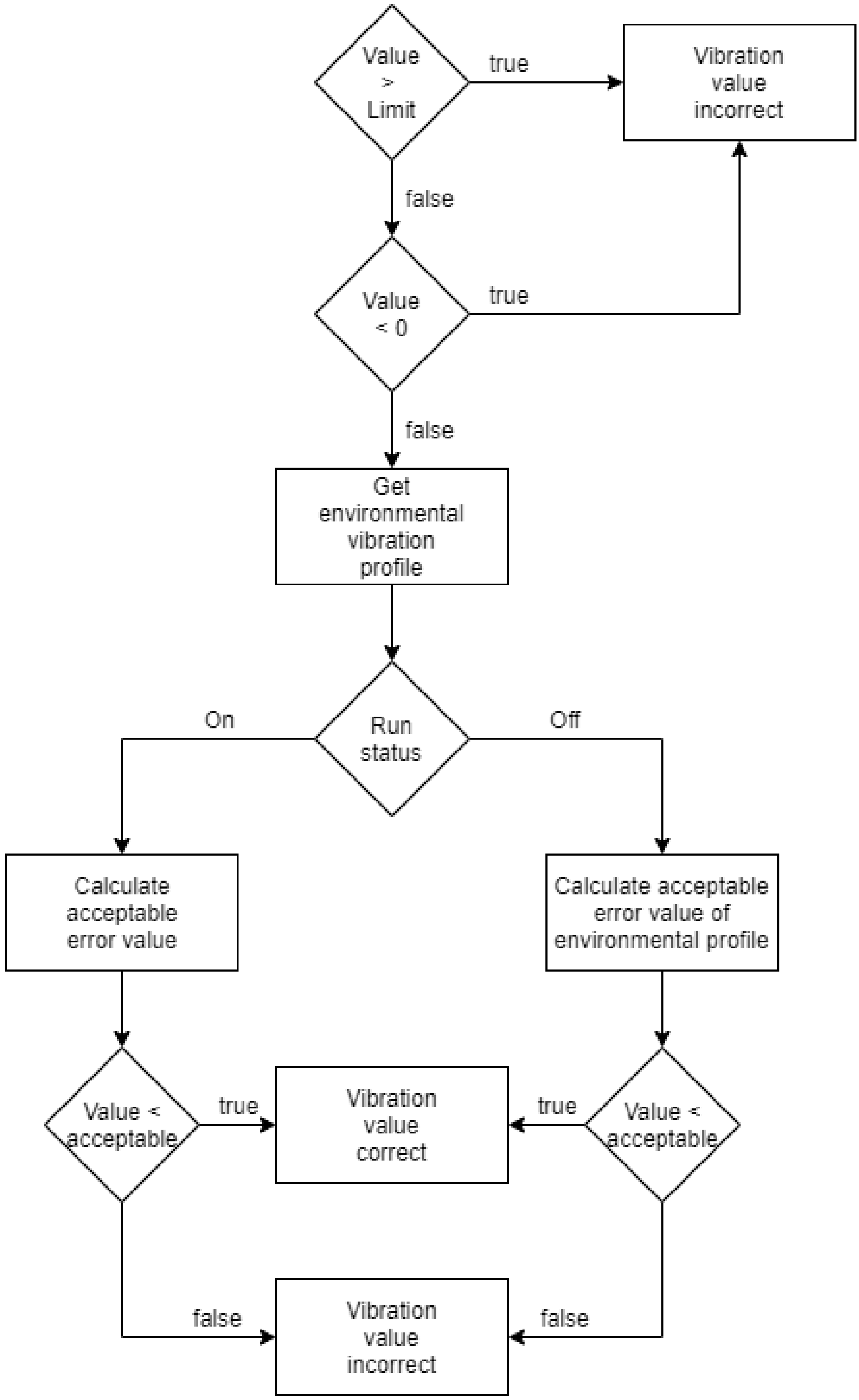

Figure 11: Image describing process of validating vibration values

The method, similar to the temperature validation method, follows a similar approach for a component in the “on” and “off” state. The current measurement is compared to the vibration limit (equation 4) and zero (vibration measurements are calibrated to be positive values.) If the value exceeds either of the two limits, the value seems inaccurate.

Next, an environmental vibration profile (Appendix B) is retrieved from the database. The running status of the component is then examined. When the component is off, only the environmental vibration is considered when calculating an acceptable value range using an error constant. The ideal vibration value range is calculated using

$$V_{Env} - (V_{Env} \times E_C) \leq V_{IR} \leq V_{Env} + (V_{Env} \times E_C) \tag{6}$$

where V_{Env} is the environmental vibration data point for the current time as retrieved from Appendix B, E_C is an acceptable error percentage for the measured value to deviate from the calculated upper limit value and V_{IR} is the ideal vibration range. When the component is in the “on” state, the ideal vibration value range is calculated using

$$V_{On} - (V_{On} \times E_C) \leq V_{IR} \leq V_{On} + (V_{On} \times E_C) \tag{7}$$

where V_{On} is the current total vibration experienced by the component as discussed in Equation 4, E_C is an acceptable error percentage for the measured value to deviate from the calculated “on” value and V_{IR} is the ideal vibration range.

The current vibration measurement is then compared to the acceptable value range. If the measurement falls within the range, the value seems accurate, otherwise it seems inaccurate.

The result of the analysis is saved per data point in its value document so it can be used at a later stage.

2.5) Verification

To verify that the system addresses not only the study objects, but also does so reliably, the system was tested using sample data (Appendix C). The sample data was run through the system and the following results were obtained.

This test case made use of the following constants for the first iteration:

- $k_{HComp} = 2$ (Component temperature gain constant, refer to equation 2),
- $T_{CUL} = 65$ (Component upper temperature limit, refer to equation 2),
- $k_{CComp} = -2$ (Component temperature loss constant, refer to equation 3),
- $T_{CLL} = 17$ (Component lower temperature limit, refer to equation 3),
- $V_{CL} = 4$ (Component vibration limit, refer to equation 4),
- $E_C = 0.1$ (Allowable error constant for measurements).

2.5.1) Clean data set

The clean data set (indicated by green values in Table 12) was first put through the system to verify whether the system incorrectly flags correct data as faulty. Using the original constants listed above, the system obtained the results displayed in Figure 12.

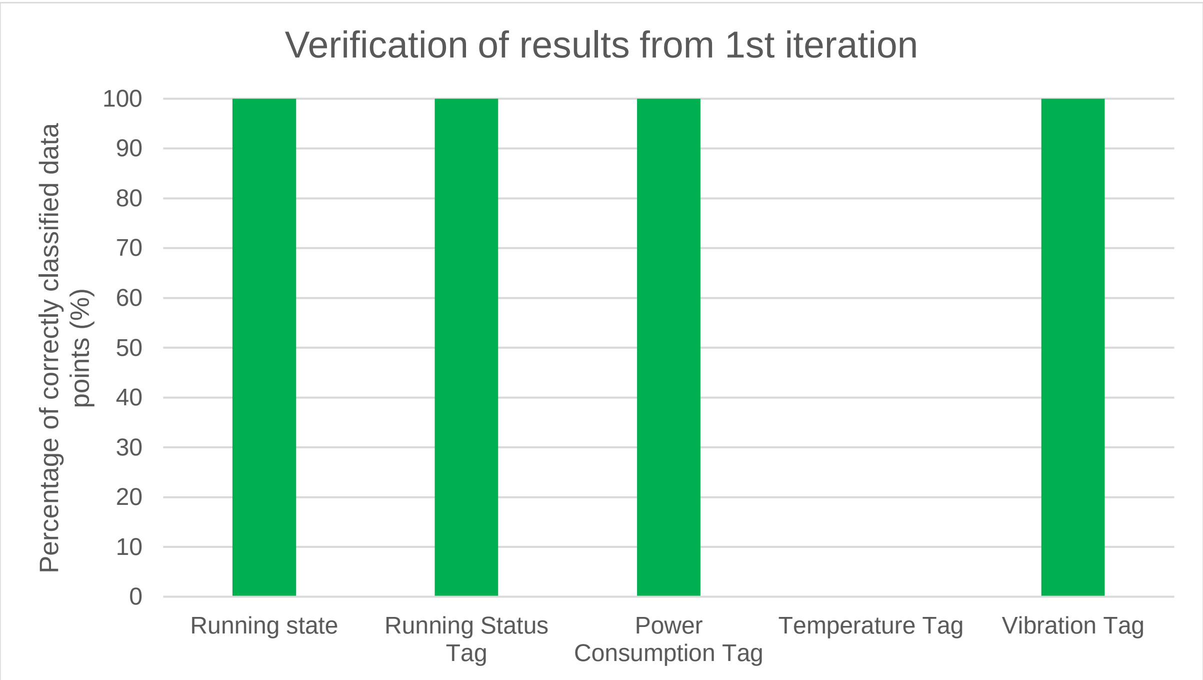

Figure 12: Bar chart displaying 1st iteration system accuracy with clean data set

The system was capable of correctly identifying 80% of the values from the test data set. Upon investigation, it was seen that the error constant E_C made the system misidentify temperature values by as little as 0.8 ^{o}C. The constant was updated to 0.2 to ensure the system correctly identifies the temperatures in the test set. The increase from 0.1 to 0.2 resulted in an increased allowed temperature range, widening the allowable error to 13 ^{o}C.

Upon further investigation into sample data, it was found that the gain and loss constants for temperature were also incorrect. The gain and loss constants were recalculated to 5 ^{o}C and -5 ^{o}C respectively.

After the constants were updated, the sample data set was run through the system again. Using the newly calculated constants, the system obtained the results displayed in Figure 13.

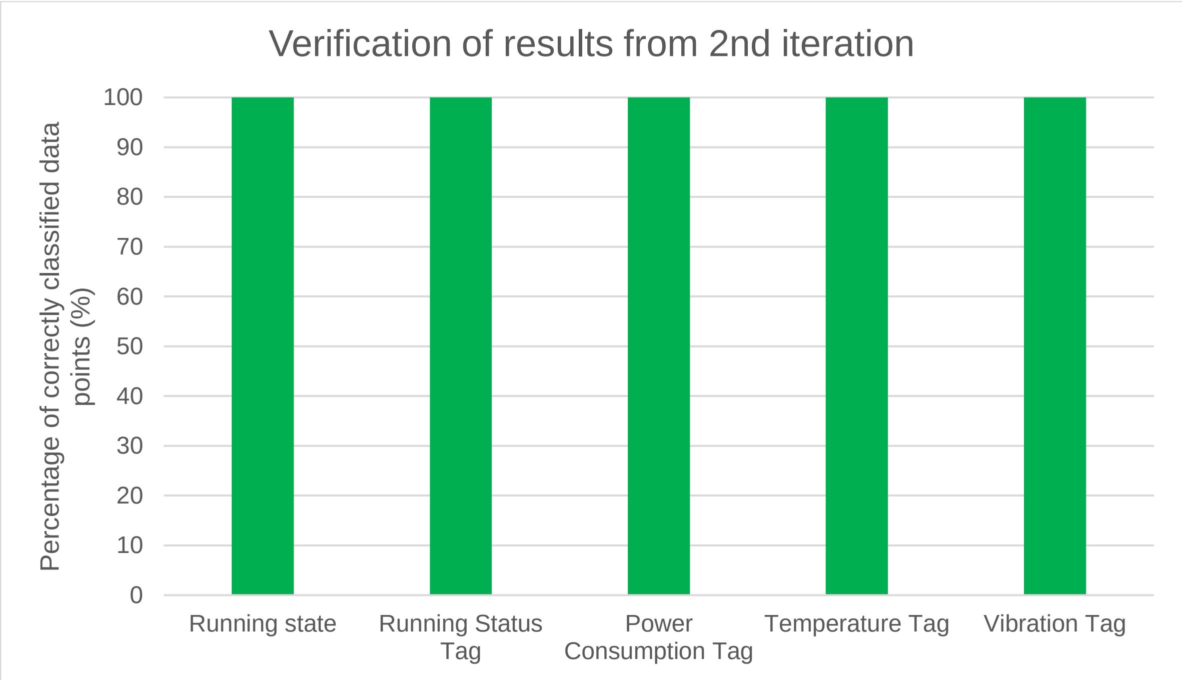

Figure 13: Bar chart displaying 2nd iteration system accuracy with clean data set

The results of the 2nd iteration of system verification were more favourable. The wider allowable error range ensured that the temperatures were correctly identified, along with the rest of the measurements.

2.5.2) Erroneous data set

The data set was updated to include the errors indicated in Appendix C as indicated in red. 11 erroneous days were added and are described in Table 9.

Table 9: Erroneous data set with introduced error and expected outcome

Run status	Power consumption	Temperature	Vibration	Expected result
		Exceeds upper limit		Faulty temperature
Off				Faulty run status
			On upper limit	Not flagged
	Low power consumption			Not flagged
			Negative vibration	Faulty vibration
Off				Faulty run status
	Very high value		High vibration	Not flagged
Off	High value	Exceeds upper limit	On lower limit	Flagged run status, temperature, vibration
Off				Flagged run status
Off				Flagged run status
		Below lower limit		Flagged temperature

The first iteration of verifying the system on the error-induced data set delivered the results displayed in Figure 14.

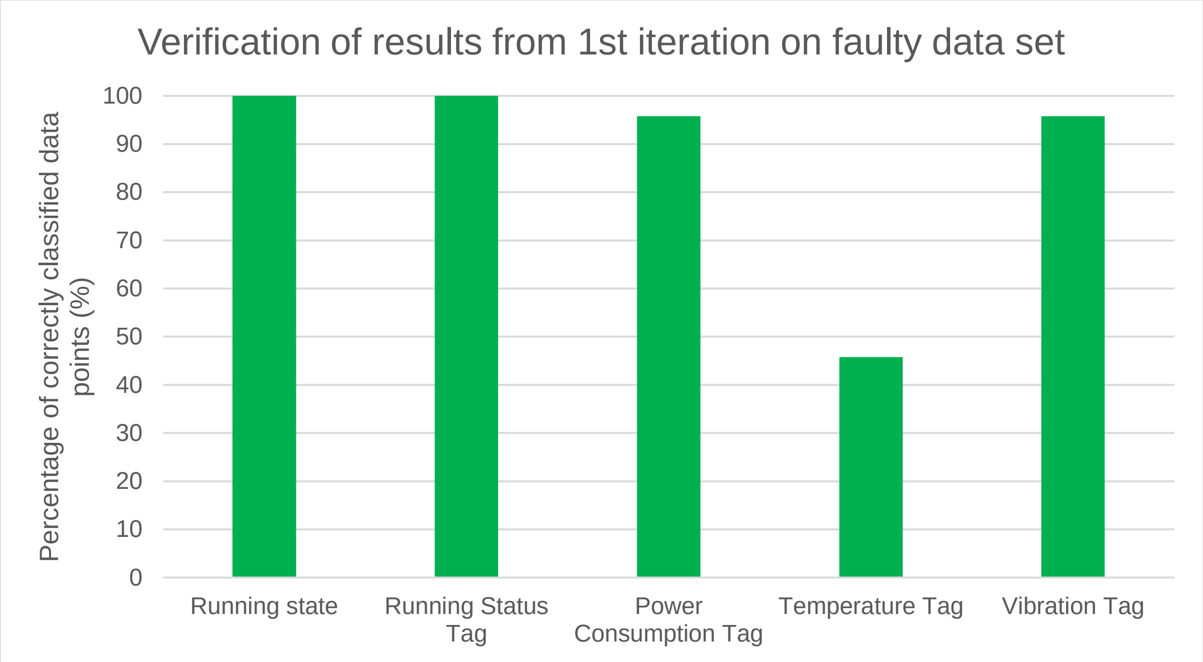

Figure 14: Bar chart showing 1st iteration system accuracy for erroneous data set

As can be seen in the abovementioned bar chart, the results from the test were somewhat successful with 4 out of 5 sections having a success rate of over 95%. The temperature measurements, however, had a very low success rate. This was due to a small oversight in the system which would calculate the state time (time the component has been in the current state of operation) by only looking at past values, rather than past values and data integrity score. The system was updated to remove this flaw and a second iteration was run on the faulty data set.

The second iteration results are displayed in Figure 15.

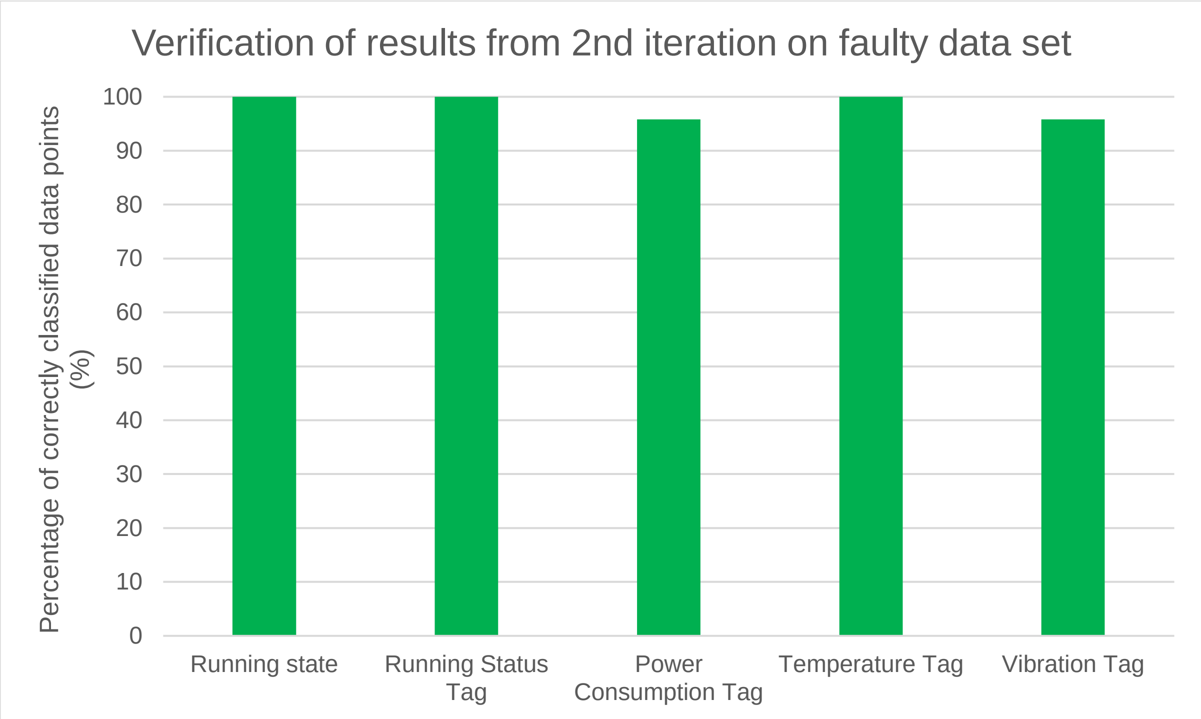

Figure 15: Bar chart showing 2nd iteration system accuracy for erroneous data set

By adding the check to verify that the last state change point was verified by the system, the accuracy for flagging temperatures increased from 45.8% to 100%. This results in an overall accuracy of 98.32% when using an allowable error range of 0.2.

In both the 1st and 2nd iterations, some shortcomings of the system were identified. These shortcomings are as follow:

- Power consumption values are not verified using a power consumption profile. Introducing a profile in a similar manner to temperature and vibration could further increase the accuracy of the system. However, a generic profile will be difficult to create as different components have different energy requirements, thus a component-specific profile will need to be created.
- Vibration measurements struggle to get flagged as faulty when they are within the operating limits. A more accurate method of verifying vibration measurements is something that can be considered for future work.

CM is a continuous process of monitoring equipment, as stated in 1.4). Thus, full erroneous days were excluded from verification as it is unlikely that a full day of faulty data should occur under regular monitoring.

2.6) Summary

The system requirements and details regarding the data layout and how data measurements were taken was discussed in Section 2.2).

Section 2.4) discussed the design of the solution. This included how the current system worked and how it should be adapted to include the new system. The methodology was discussed along with the model design. The system was designed, and it was explained how the models are used in the system and how each characteristic will be evaluated.

The system was verified with a test case in Section 2.5). Some shortcomings of the system were identified. Overall, the system did well in testing and was able to correctly classify 98.32% of measurements when using an allowable error range of 20%.

Chapter 3
Results

3.1) Introduction

The system and models were implemented on a case study in order to validate whether the proposed solution addresses the identified need. The system was implemented at an industrial company, hereafter referred to as the company, located in South Africa. The company provides various services to their clients, including monitoring the condition of their equipment. Their existing system was adapted, as described in Figure 7, and the data integrity system was added.

The configuration menu used for the CM dashboard was modified to configure the context of the data streams to a component, along with the relational database table which stores the data. The configuration menu can be seen in Figure 16. In the menu, the Company, or client, is selected from the drop-down list labelled “Group”. Once the Company is selected, the existing company structure is loaded. From this structure, the user navigates down the structure until they select the desired Component to configure, as indicated in red.

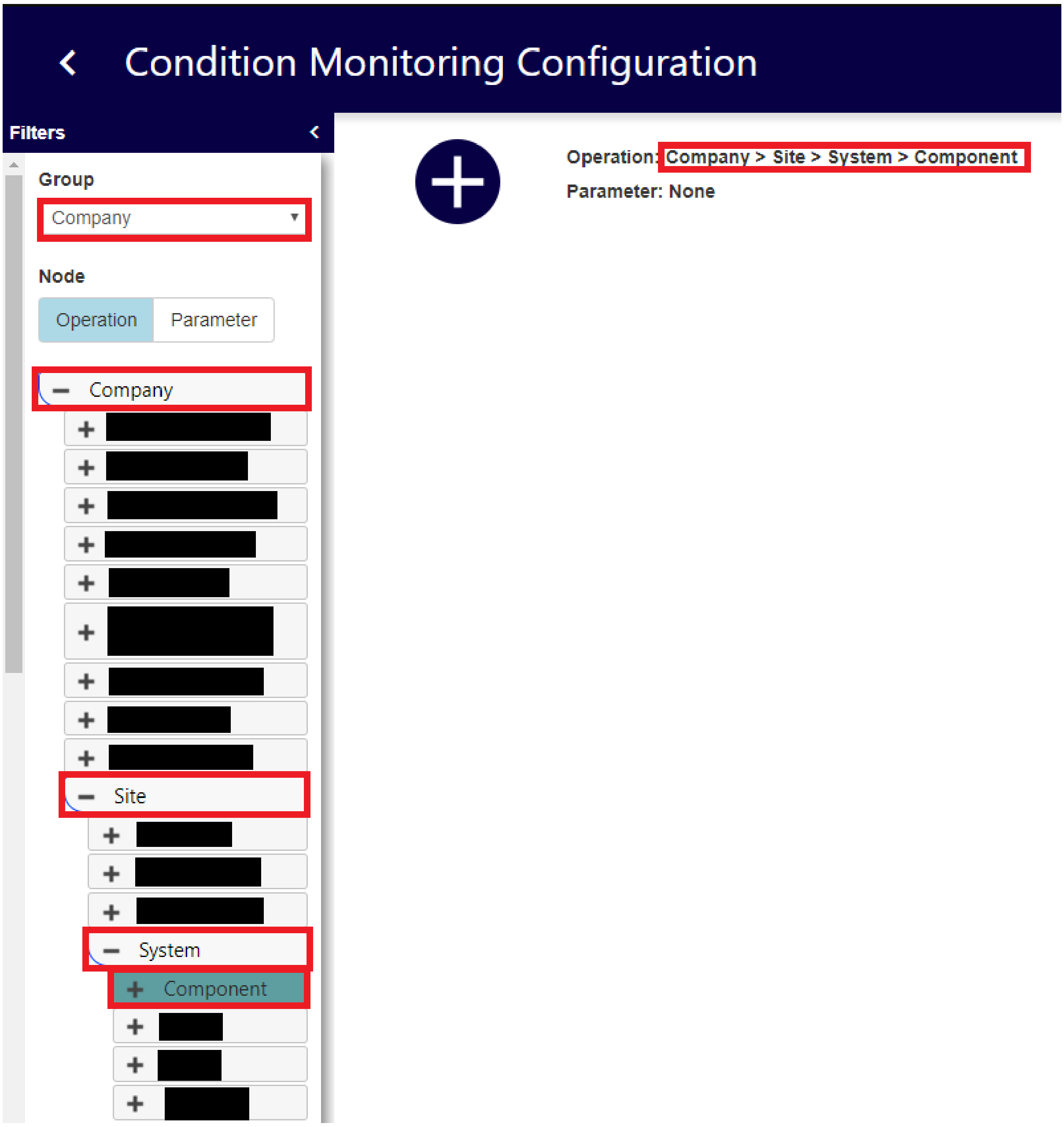

Figure 16: Screenshot of configuration menu in which a component is selected

After the component has been selected, the characteristic setup can be completed, as displayed in Figure 17 and Figure 18.

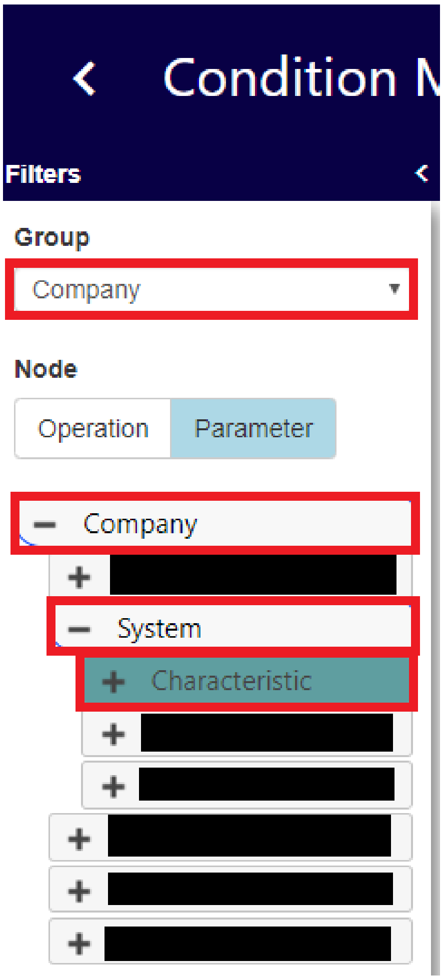

Figure 17: Screenshot of configuration menu for selected component

Figure 17 shows how the desired characteristic can be selected for which the data streams should be defined. Once a characteristic is selected, the user has the option to link data streams it.

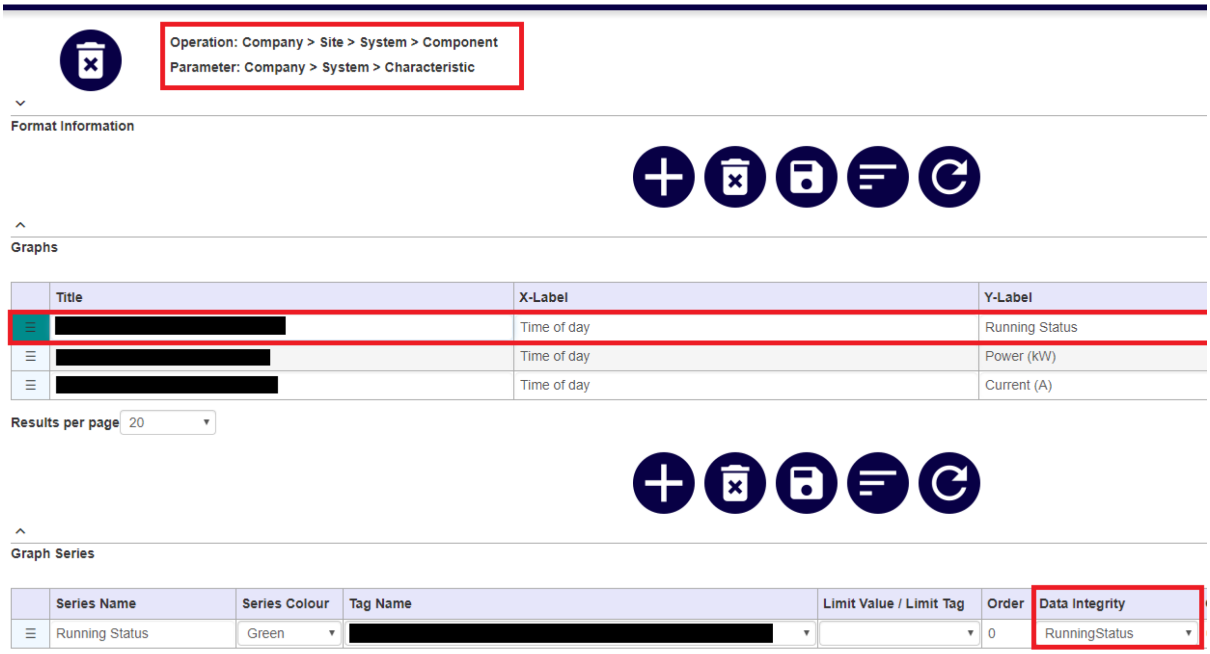

Figure 18: Screenshot of data stream linking and definition

Figure 18 displays how data streams are linked to the selected characteristic. Each characteristic is configured individually. A table (the top table) is used to link graphing attributes for the CM display website, discussed in 2.4.1). A tag, or data stream, is linked in a second table (bottom table) to each of the attributes. An additional field, named Data Integrity, was added to the graph series table. In this field, the data stream can be set as a running status, power consumption, temperature, or vibration input to the component model. The system uses the characteristic setup per component to calculate the data integrity of the data streams.

The non-relational database which stores the data already has the capability of storing data quality-related results. This functionality is used to store the data integrity results.

3.2) Criteria for implementation

The company uses single source data for their CM system. This data is displayed on their dashboard without any data quality or integrity checks. The system does indicate when a value is close to or exceeds a set limit; however, these indications can be misleading. The system does not take quality or integrity into consideration, which may result in falsely flagging data points. Two weeks' worth of data for each component is needed to calibrate the system.

3.3) Case study

The developed system was implemented for one of the company's clients, namely a deep-level mining company hereafter referred to as the client. The company actively monitors 18 of this client's sites, each of which follow the generic data layout described by Figure 2.

For the case study, the six sites which most frequently experience data quality and integrity issues were identified. These sites were identified by the person who was in charge of the CM platform for the company and inspected the data daily. From these six sites, focus was placed on the eight components which are most plagued by low quality data, as identified by the person mentioned above. Each of these components are from a different system. The components chosen are listed in Table 10.

Table 10: Case study component list with common data issue

Component	Site	System	Common data issue
Component A	Site A	Pump	Impossible vibration measurements
Component B	Site B	Fan	High vibration measurements
Component C	Site B	Refrigeration	High temperature measurements
Component D	Site C	Compressor	Impossible temperature measurements
Component E	Site D	Compressor	Impossible temperature measurements
Component F	Site E	Fan	Impossible temperature measurements
Component G	Site E	Refrigeration	High vibration measurements
Component H	Site F	Pump	Incorrect power consumption

From the abovementioned table, it can be seen that two of each component type will be used in the case study. Each component has a common data issue identified manually by analysing the raw data. Common data issues differ per component and similar components experience different issues, with the exception of the two compressors.

Identifying low data integrity points in the clients' data will add additional functionality to the companies' systems. This functionality can be offered to the company's clients as an extra service, which in turn could improve the efficiency of the client.

A years' worth of data for each component was put through the system to analyse the data integrity. After the analysis, the results were inspected manually to ensure that the data points were correctly flagged. Each component is discussed individually, followed by an overview for the entire case study.

Three types of graphs can be displayed for each analysed component. The first graph displays the total number of data points flagged correctly vs incorrectly per data stream. This graph contains the total number of data points for each data stream, the calculated running state, and the total of the previous data points. Each data point for each data stream was either correctly or incorrectly classified by the system. The green bar shows how many data points were correctly classified as either high or low integrity points, while the red bar indicates how many points were incorrectly classified.

The second graph type is a sample of the data streams in which a phenomenon occurred. These graphs differ depending on the type of anomaly which took place.

The third graph displays the number of data points classified as low integrity points and how many of the points were classified correctly. The graph focuses on the four data streams linked to the component and gives an overview of the total. The blue bar indicates how many of the data points were flagged as being low integrity points, while the green bar indicates how many points were correctly classified.

3.3.1) Component A

Component A is a pump located on Site A. The measurements for this component have a tendency to contain impossible vibration measurements. These impossible vibration measurements are measurements that surpass the “on” state upper vibration limit set for the component when the component is in an “off” state. In these situations, the vibration measurements should be equal or close to the environmental vibration.

The statistics for component A’s data integrity is summarised in Figure 19.

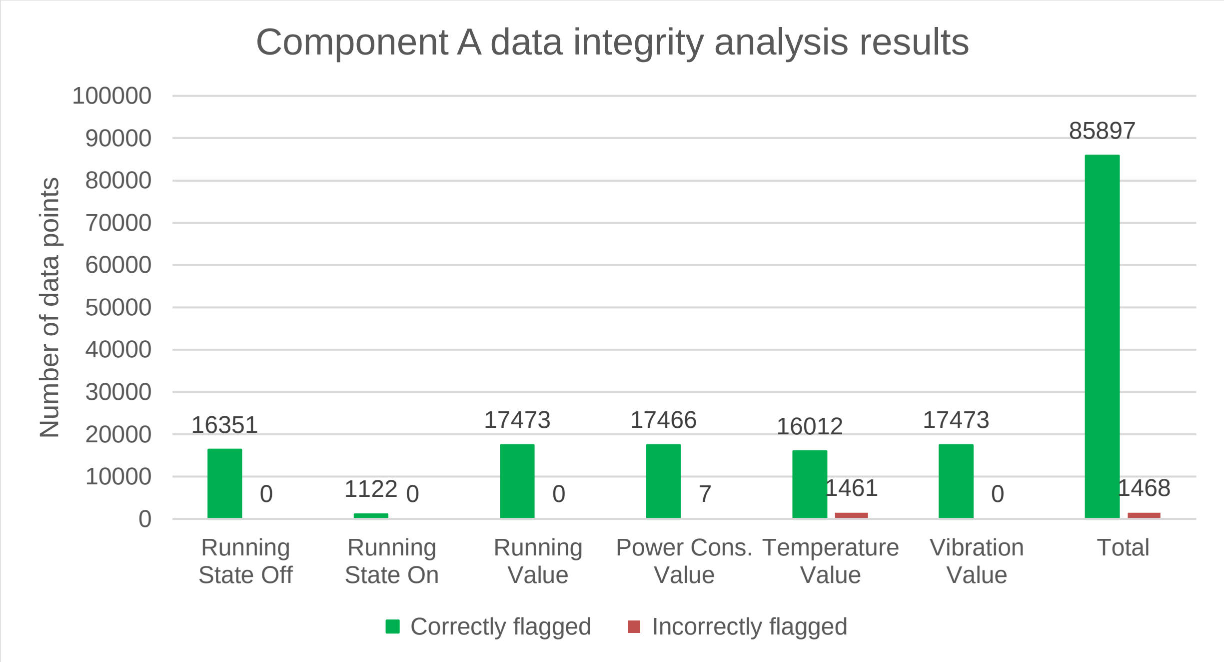

Figure 19: Bar chart displaying analysis results for component A

The abovementioned chart displays the number of points calculated correctly by the system versus the number of points calculated as incorrect for component A. From the chart, a few important things can be seen regarding component A. Firstly, component A was “off” for almost 94% of the year. This reduces the number of low integrity points as all data streams should have near-perfect, calibrated values.

Secondly, some data points were flagged falsely. The temperature data stream had the largest number of erroneous flagged points by the system. The system incorrectly flagged 1 461 (~8.5%) temperature measurements. This was largely due to the ambient temperature being higher than the calibrated values. Although the calculated ambient temperature profile in Appendix A resembles the ambient temperature at the component, it does not take weather and environmental changes into consideration. This means that choosing the period used for calibration can greatly affect the accuracy of the system.

If the temperature measurements are not correctly flagged, temperatures exceeding the set limits could result in components overheating or breaking down. This could lead to a loss in production due to component downtime during maintenance, which will have a monetary impact on the client. Recurring instances will ultimately lead the client to lose faith in the system and/or the company which is responsible for CM of the components.

The analysis results of the flagged data points are displayed in Figure 20.

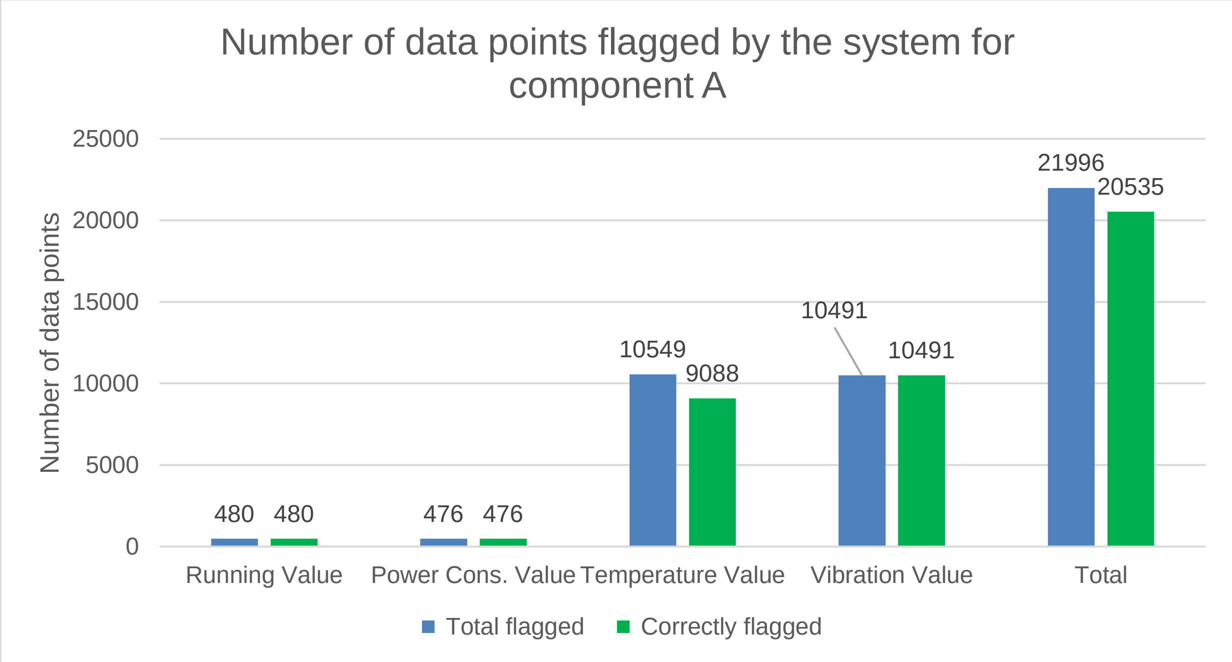

Figure 20: Bar chart displaying flagged data points for component A

The abovementioned figure displays the total number of points that were flagged as low integrity points versus the number of points that were verified to be correctly flagged as low integrity points. As mentioned earlier, there is a high number of temperature points that were falsely flagged. Component A had a tendency to have impossible vibration points. These points exceed the set upper vibration limit to a large extent. Upon further investigation it was discovered that the impossible values occur at times when the system is in the off state, as seen in Figure 21.

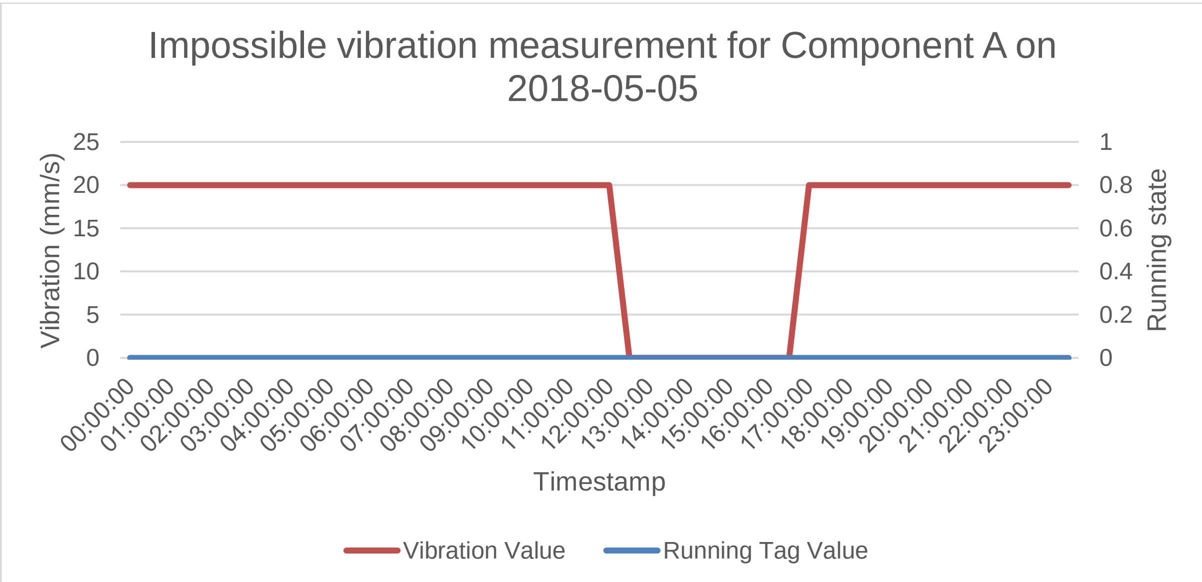

Figure 21: Data streams for component A displaying impossible vibration

Using the contextual analysis, it was discovered that the system received zero values for the running -, power consumption -, and temperature values. The most likely cause for this anomaly is that the measurement equipment had disconnected from the component along with the vibration sensor having a static, floating value (in this case 20 mm/s.)

3.3.2) Component B

Component B is a fan located on Site B. The component has a tendency to contain high vibration measurements. These high vibration measurements are often caused by spikes in the data. The statistics for component B's data integrity is summarised in Figure 22.

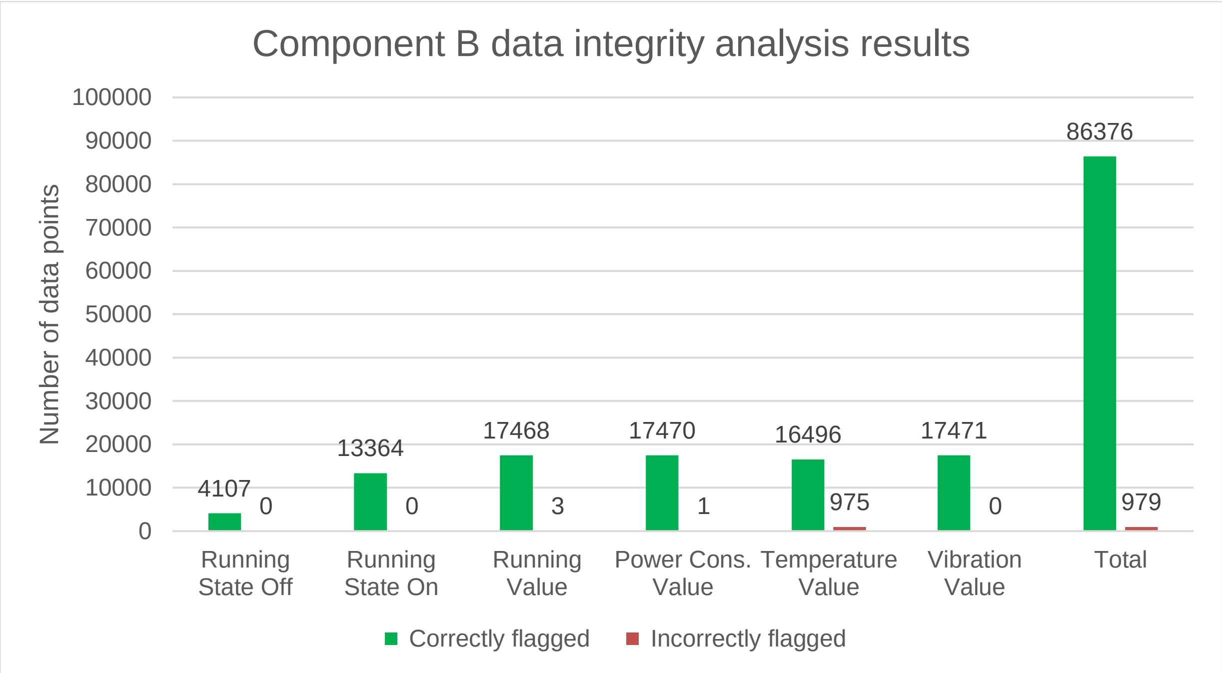

Figure 22: Bar chart displaying analysis results for component B

The abovementioned chart displays the number of points calculated correctly by the system versus the number of points calculated as incorrect for component B. Unlike component A, component B was "on" for the majority of the testing period (~76.5%). This typically means that there is more room for error as there are more external factors influencing the component and, indirectly, the measurements.

The system accuracy for component B was approximately 98.8%, with the temperature measurements again being the greatest problem. Similar to component A, the ambient temperature that exceeded the ambient temperature profile was a problem.

Besides these issues, another observation was made. Due to the component being in the "on" state much more than the previous component, there were a lot of transition states. These transition states are the periods where the component goes from the "on" state to the "off" state and vice versa. The system falsely flagged most of the data points in these states. The system uses equation 2 and 3 to determine the values in these states, which is not accurate enough as they assume a linear temperature increase, which is not the case.

The analysis results of the flagged data points are displayed in Figure 23.

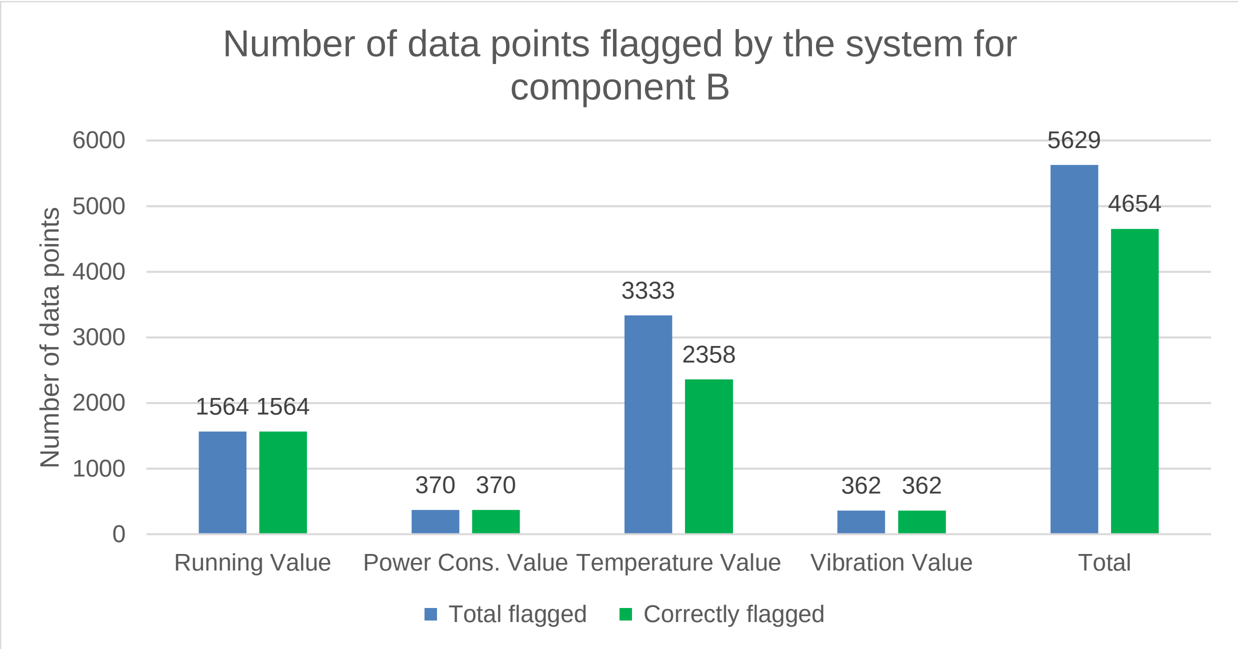

Figure 23: Bar chart displaying flagged data points for component B

The abovementioned chart displays the total points flagged as low integrity points versus the number of verified low integrity points for component B. As stated in Table 10, component B had a lot of high vibration measurements. As seen in Figure 23Figure 28, all these high value points were successfully identified by the system. The only issue encountered by the system was the temperature in the transition states, as mentioned earlier and illustrated in Figure 24.

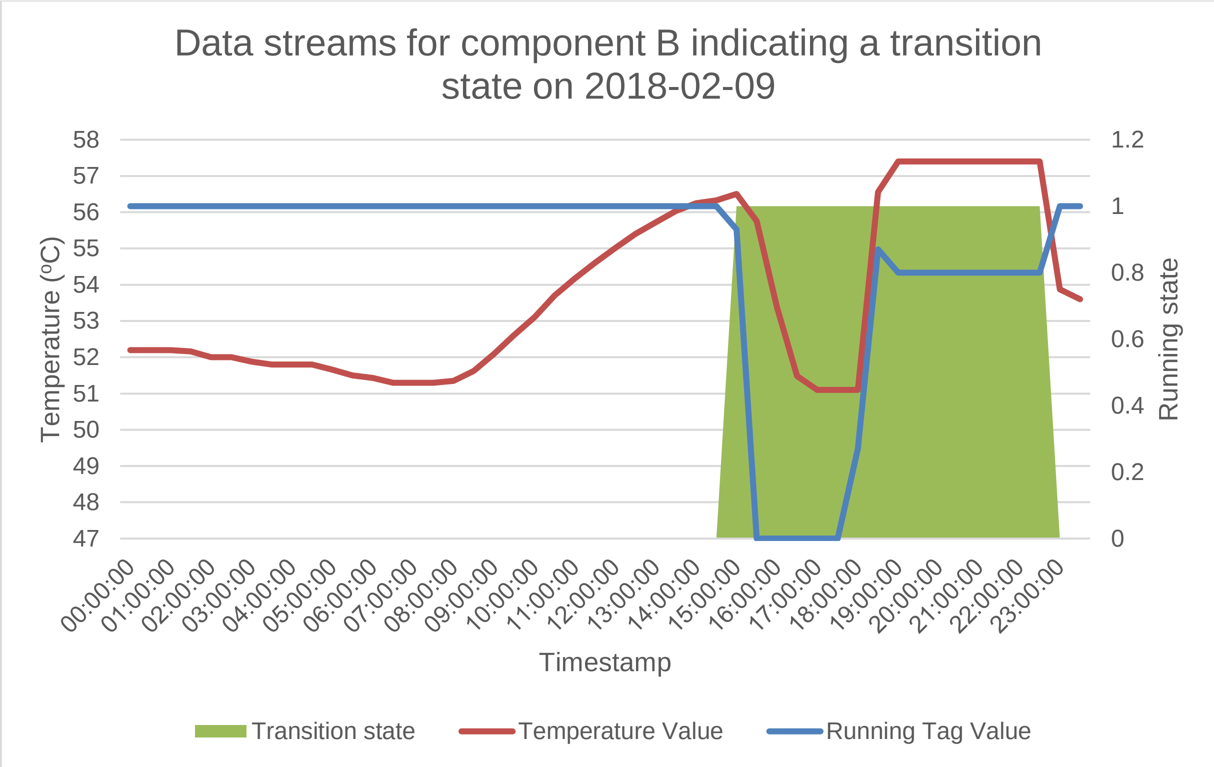

Figure 24: Data streams for component B indicating a transition area

In the abovementioned figure, the temperature of the component is plotted against the component running status value. The running state value is measured in a two-minute resolution, summed, and the average is calculated per half hour. For the time period of 15:00 to 23:00, the component was constantly switched between the “on” and “off” state, resulting in the decimal running state values. The green area indicates this time period in which transition states occurred. As seen from the graph, the temperature values follow a similar pattern to the running value, with the exception of the limit exceeding the value of 57.3 ºC where the limit is set to 55 ºC.

Although these high values were correctly flagged as low integrity points, the rest of the points in the transition states were incorrectly flagged as low integrity points. This was due to the models not being calibrated accurately enough as well as the system classifying transition states as being in the “on” state.

3.3.3) Component C

Component C is a fridge plant located on Site B. The component has a tendency to contain high temperature measurements.

The statistics for component C's data integrity is summarised in Figure 25.

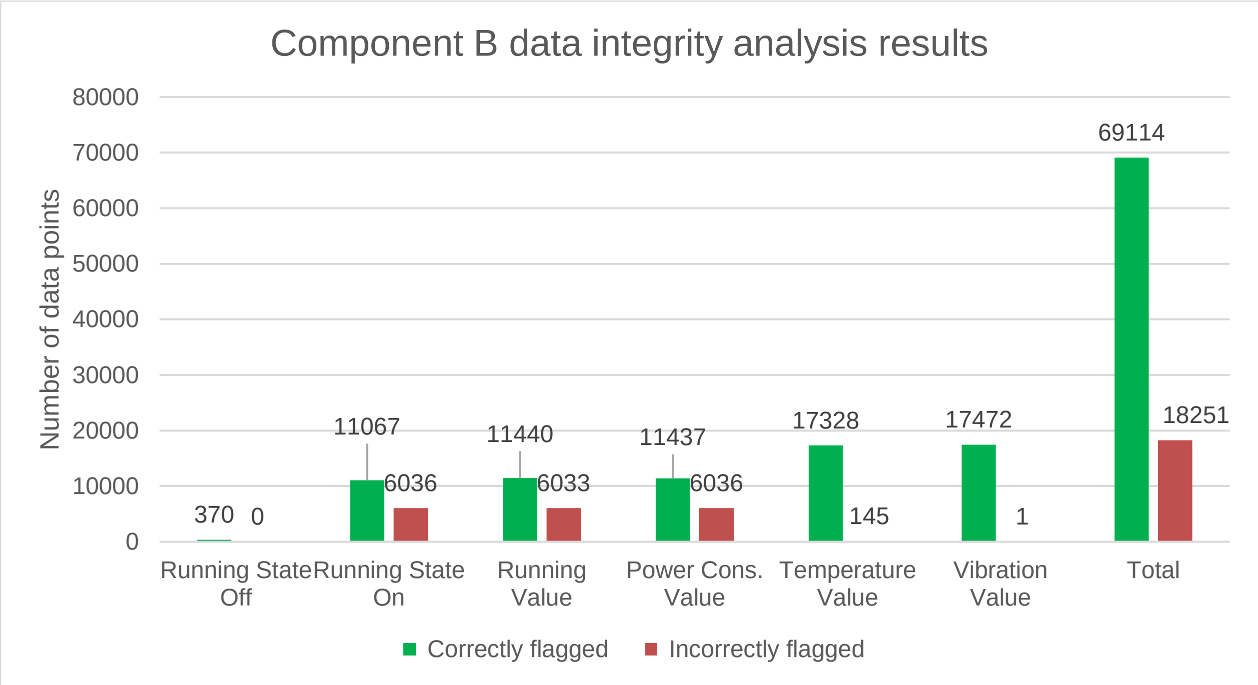

Figure 25: Bar chart displaying analysis results for component C

The abovementioned chart displays the results for component C. During the analysis, 18 251 data points (~20%) were falsely flagged as being high integrity points. The data streams which were affected were the running value, power consumption values, and as a result of these two, the temperature value. Upon investigation, it was discovered that the power consumption sensor was not calibrated, resulting in its "zero-value" being 0.61, as displayed in Figure 26.

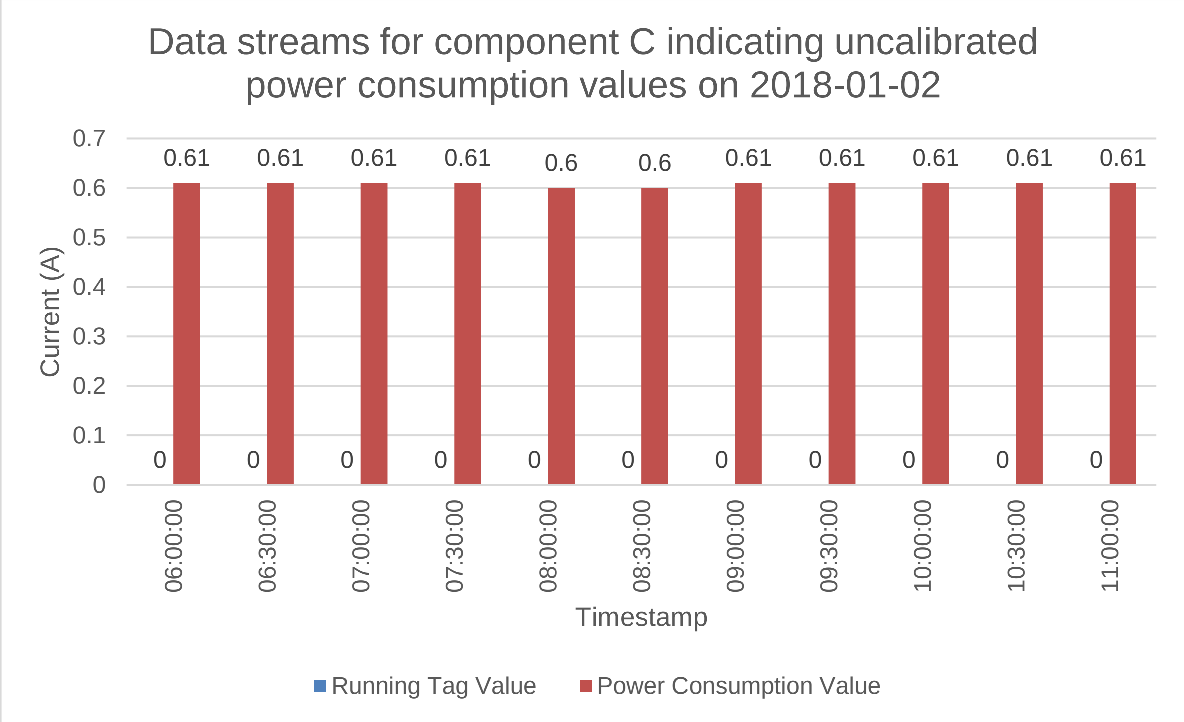

Figure 26: Bar chart illustrating uncalibrated current values for component C

This violates the truth table assumptions, resulting in the running state being incorrectly calculated as "on". This flags the running status as low integrity, along with the temperature. In the currently implemented CM platform, these instances are missed as the platform does not do calculations when the running value is zero. This results in none of the measurements being flagged when they are faulty. This uncalibrated power consumption identifies an improvement to the system which was not identified during verification.

The analysis results of the flagged data points are displayed in Figure 27.

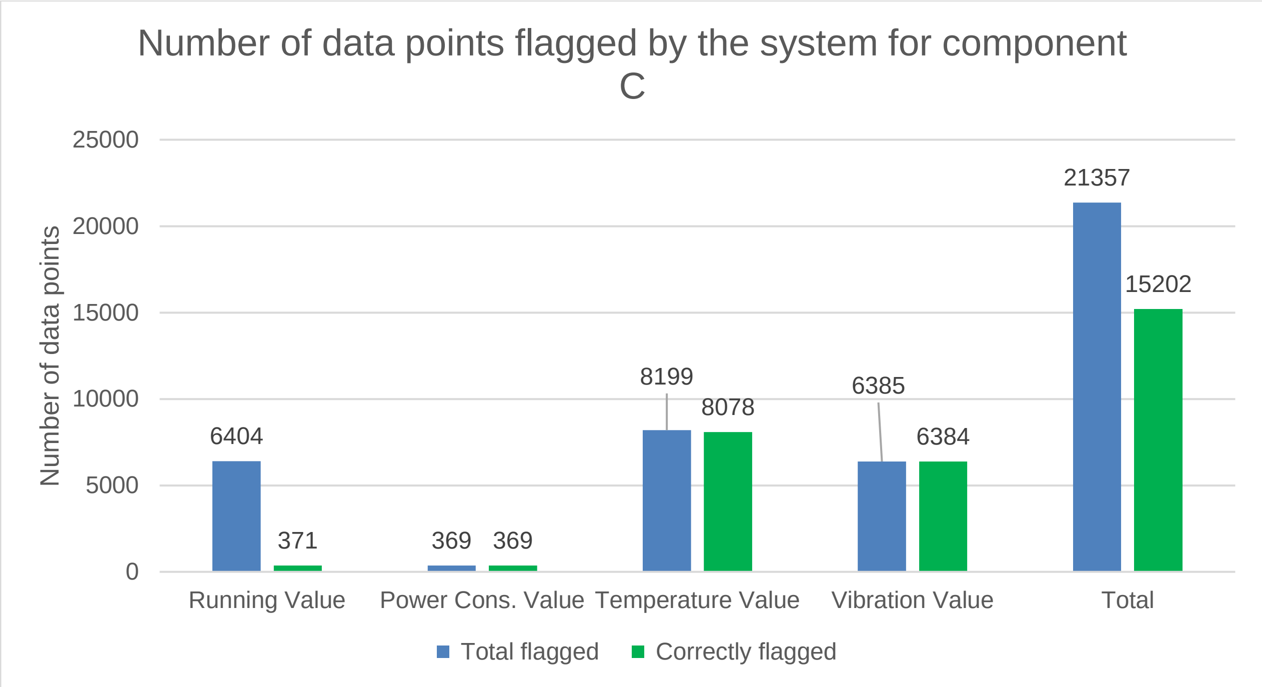

Figure 27: Bar chart displaying flagged data points for component C

The bar chart displays how the system is negatively impacted when data points fall outside of the design spec, indicated by the low number of correctly flagged running values. This suggests that the system should be expanded to verify the assumptions on which it is built before implementing them.

3.3.4) Component D

Component D is a compressor located on Site C. The component has a tendency to contain impossible temperature data points. These impossibilities occur when the temperature exceeds the set upper limit of 75 ^{o}C.

The statistics for component D's data integrity are summarised in Figure 28.

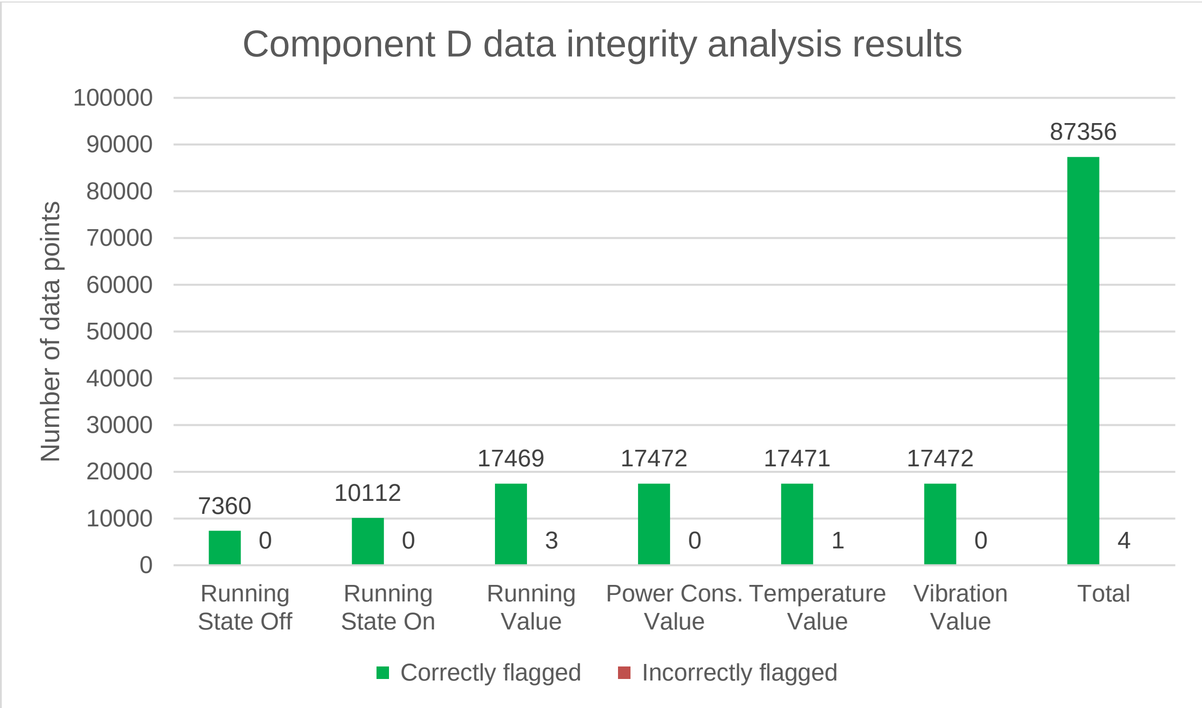

Figure 28: Bar chart displaying analysis results for component D

From the bar chart it can be seen that the system was able to correctly identify approximately 99.9% of low integrity data points. These low integrity points include the impossible temperature measurements which are commonly found for this component. The impossible temperature measurements seemed to mostly occur during the transition stage when the component is shut off, as displayed in Figure 29.

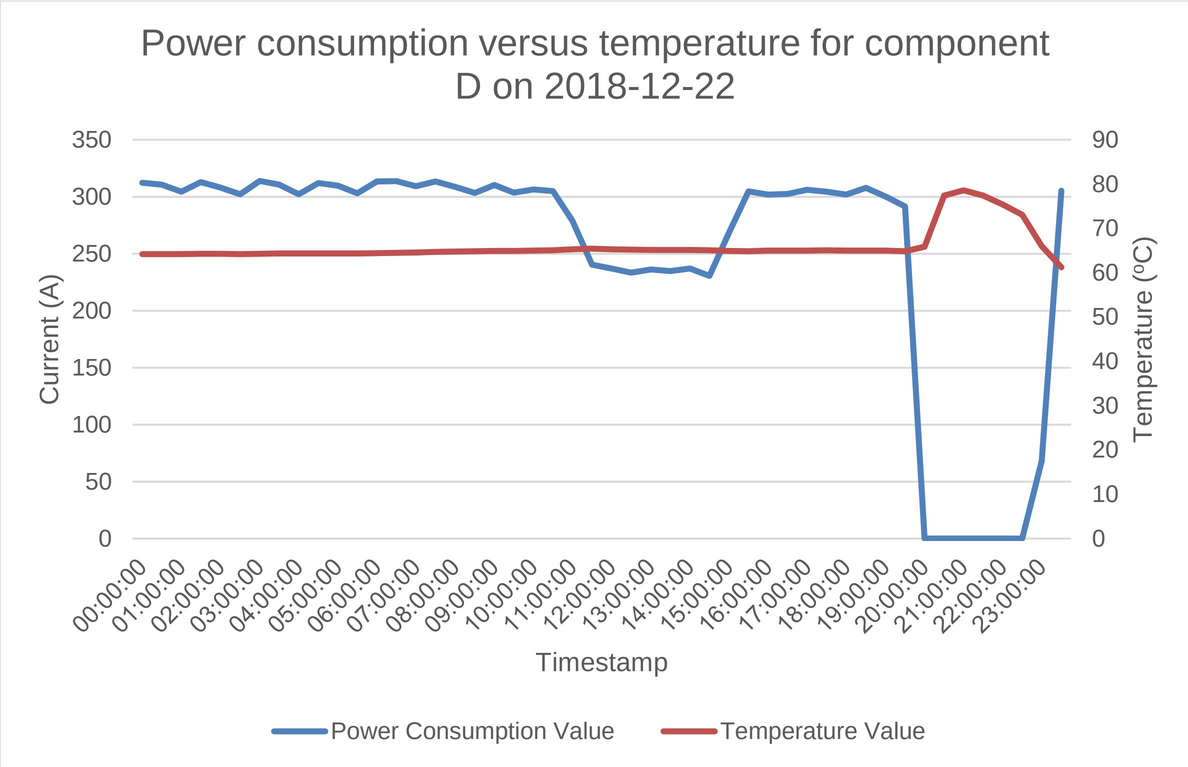

Figure 29: Line chart of impossible component D temperature in off state

In the abovementioned figure, it can be seen how the temperature for component D exceeded 80 °C, whilst the component was shut off.

This value was not simply flagged as an impossibility due to the component being in the "off" state, but also as it exceeded the maximum upper temperature limit of 75 °C. The exact cause of the temperature measurement is unclear, as the temperature data stream includes various low data integrity points.

Normally, the temperature of a component might increase slightly in the first few minutes after it is shut down due to the rotational energy being converted to heat energy. However, this temperature declines rapidly afterwards, which was not the case here. The analysis results of the flagged data points are displayed in Figure 30.

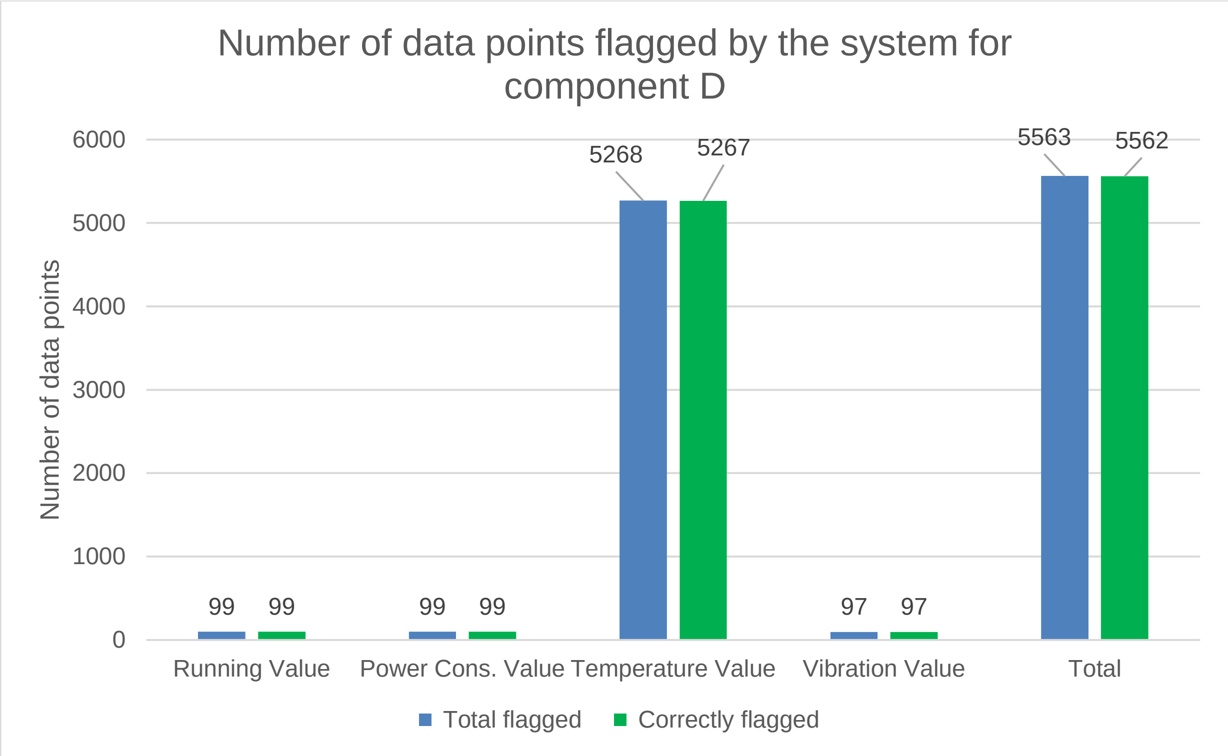

Figure 30: Bar chart displaying flagged data points for component D

The abovementioned bar chart displays how the component was plagued by low integrity temperature measurements, which contribute to roughly 95% of the flagged low integrity data points. The majority of the low data integrity points are a result of data loss, whilst the rest are data points that exceed either the upper or lower limit values.

3.3.5) Component E

Component E is a compressor located on Site D. The component has a tendency to contain impossible temperature data points. These impossibilities occur when the temperature exceeds the set upper limit.

The statistics for component E's data integrity are summarised in Figure 31.

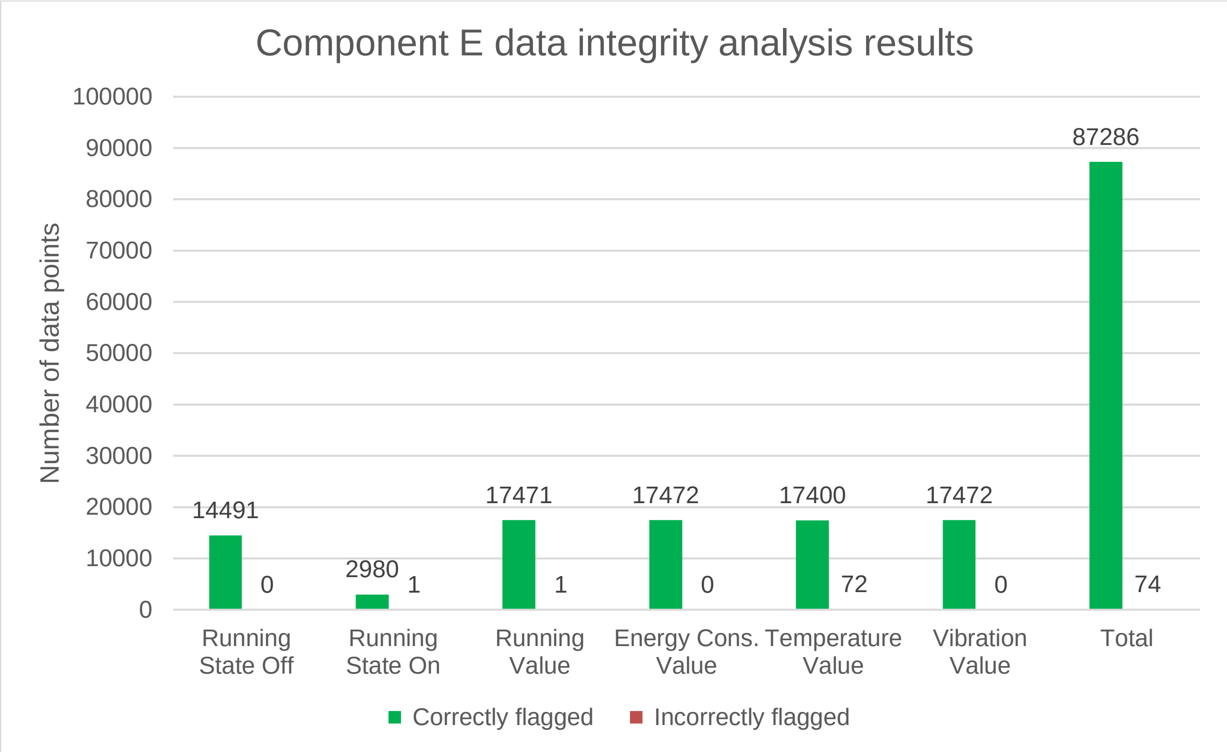

Figure 31: Bar chart displaying analysis results for component E

The abovementioned figure illustrates that the system did not experience difficulties whilst analysing the data of component E. Similar to other components, the majority of the faulty flagged data points were data points located in transition states where the component was either switched “on” or “off”. The impossible temperature values frequently experienced by component E were identified and flagged as low integrity points. Figure 32 shows some of these impossible points.

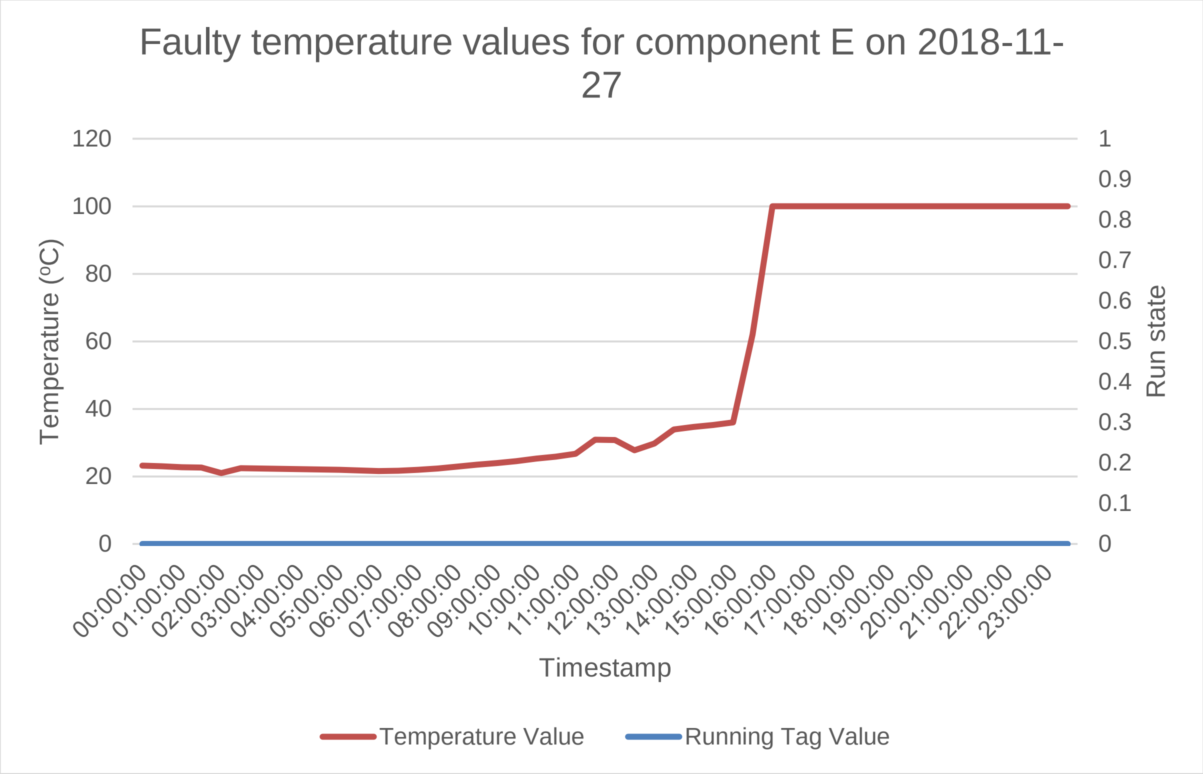

Figure 32: Line chart of impossible component E temperature during off state

The figure shows impossible values measured for component E. These values are not only flagged as impossible due to them exceeding the upper temperature limit, but also because they occur during the “off” state of the component. The most likely explanation for these values is that the measuring equipment was disconnected from the component, resulting in a static floating value of 100 ºC. Floating values are caused when a sensor is not calibrated correctly, resulting in a random value being seen as the zero point.

One of the limitations of the currently implemented CM system is that it doesn’t do an analysis when the component is in the “off” state. In this case, the high temperature values would not have been flagged as faulty. This increases the time it takes for personnel to identify that something is wrong with the component or its measuring equipment. This could lead to the clients losing faith in the current system, which would negatively impact the company responsible for the monitoring of equipment. The analysis results of the flagged data points for component E are displayed in Figure 33.

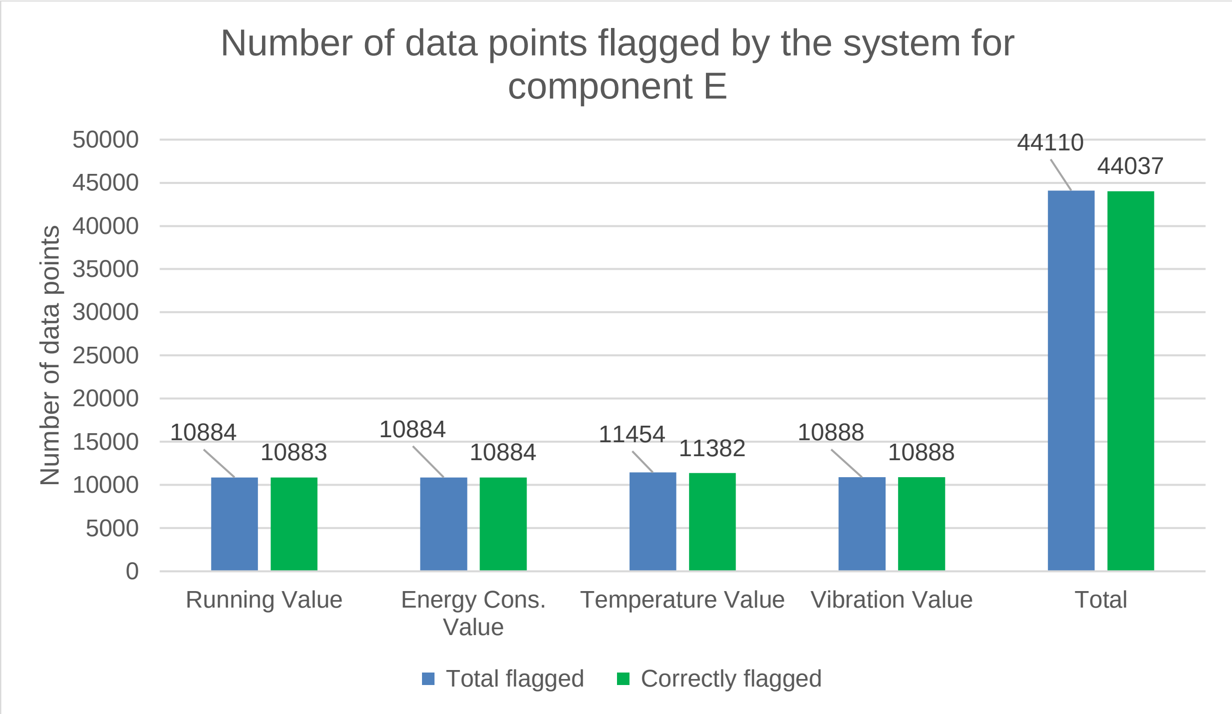

Figure 33: Bar chart displaying flagged data points for component E

Component E had multiple low integrity data points, as illustrated by the abovementioned figure. The distribution of faulty data is spread evenly between the four data streams. These low integrity points were caused by data loss, most likely due to a network error encountered when the data was sent from the measuring equipment to the translation system, as described in Figure 3. Approximately ~63% of the measurements for component E were lost. This raises a serious concern as the component was only effectively monitored for a third of the year.

3.3.6) Component F

Component F is a fan located on Site E. The component has a tendency to contain impossible temperature data points. These impossibilities occur when the temperature exceeds the set upper limit.

The statistics for component F's data integrity are summarised in Figure 34.

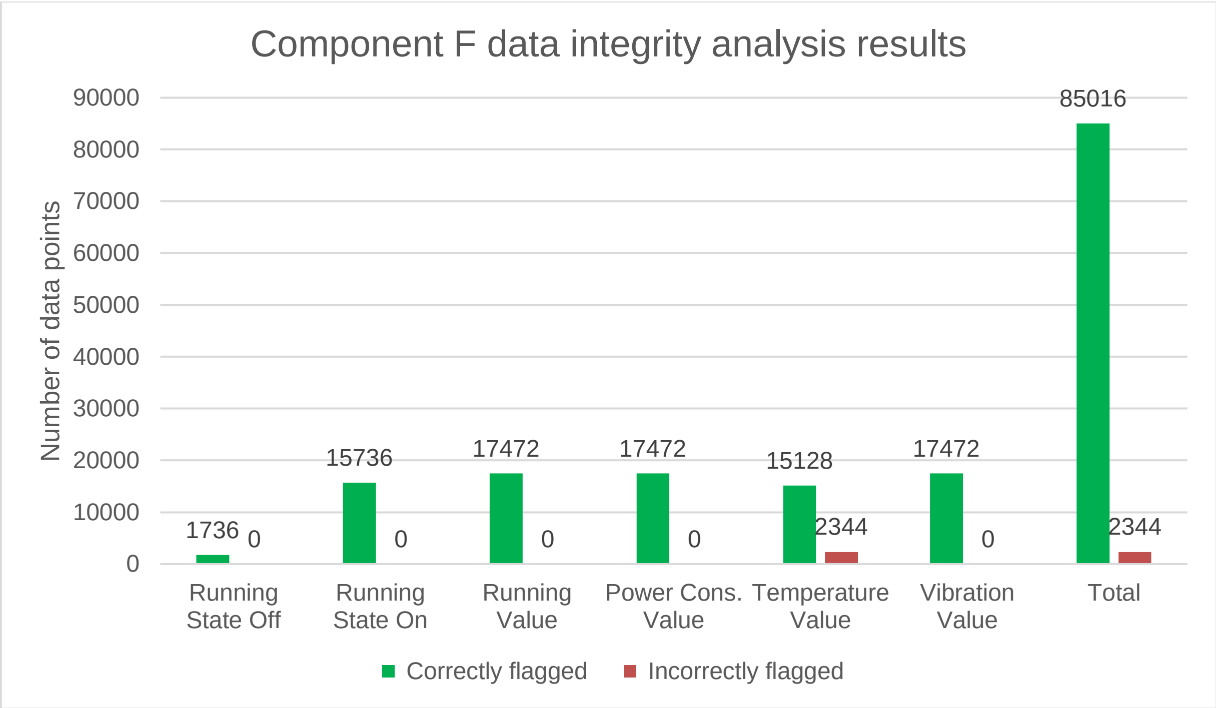

Figure 34: Bar chart displaying analysis results for component F

The abovementioned figure displays how the system struggled with the classification of temperature measurements for component F. The system was able to correctly classify ~ 86.5% of the temperatures, with the remainder of the temperatures falling in transition states. The common occurrences of impossible temperature values were flagged as low integrity points, with an example of impossible temperatures recorded for component F displayed in Figure 35.

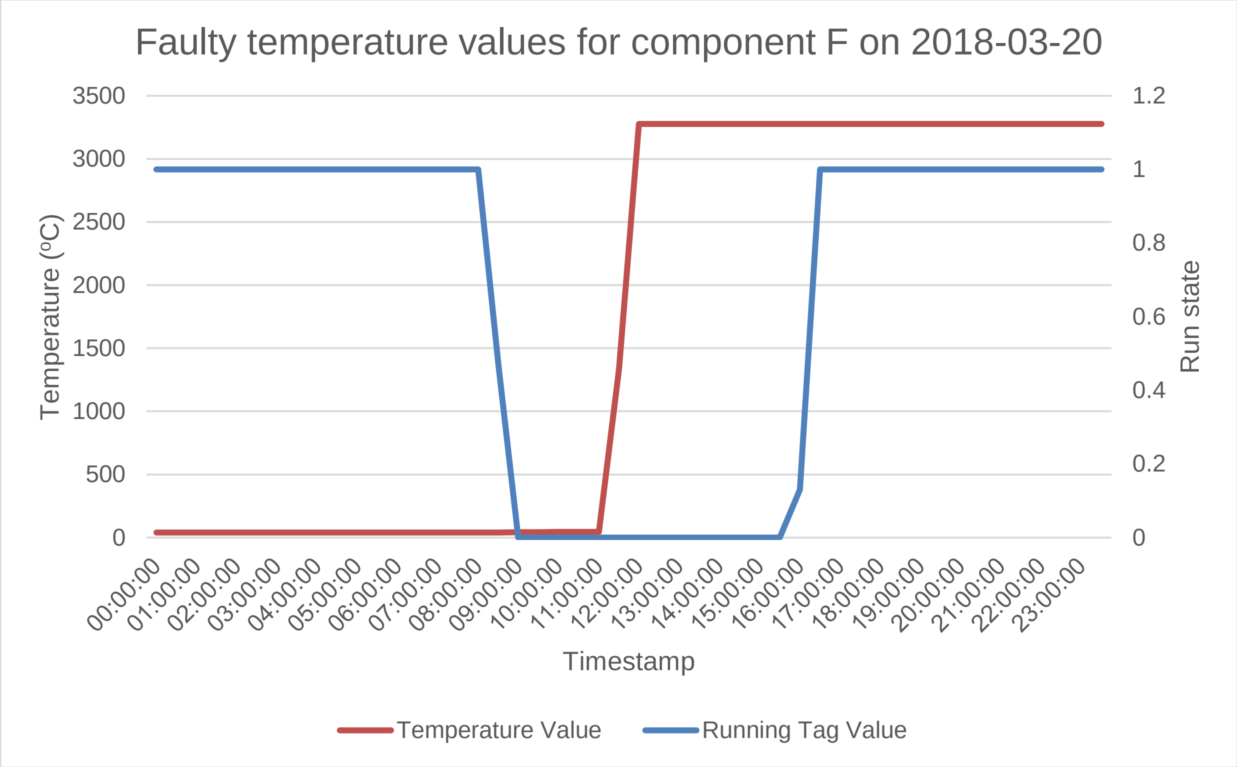

Figure 35: Line chart illustrating impossible temperature for component F

As seen in the abovementioned figure, component F experienced some faulty readings for the temperature characteristic. Multiple instances of impossible values were recorded for component F, with the majority being values that exceed the upper temperature limit. In Figure 35, however, a more severe instance was recorded with the temperature exceeding not only the upper limit, but also the operational limit of the component itself. This was most likely due to the measurement equipment disconnecting from the component and reading a static floating value into the system.

Floating values are not identified by the current system; thus they are only seen as values that exceed the set limits. This results in the actual problem, in this case a defective sensor, not being replaced or calibrated, which extends the lifetime of the problem. These types of incomplete classifications of the current system also negatively influence the maintenance schedule as the component will be unnecessarily serviced more frequently, wasting money. The analysis results of the flagged data points for component F are displayed in Figure 36.

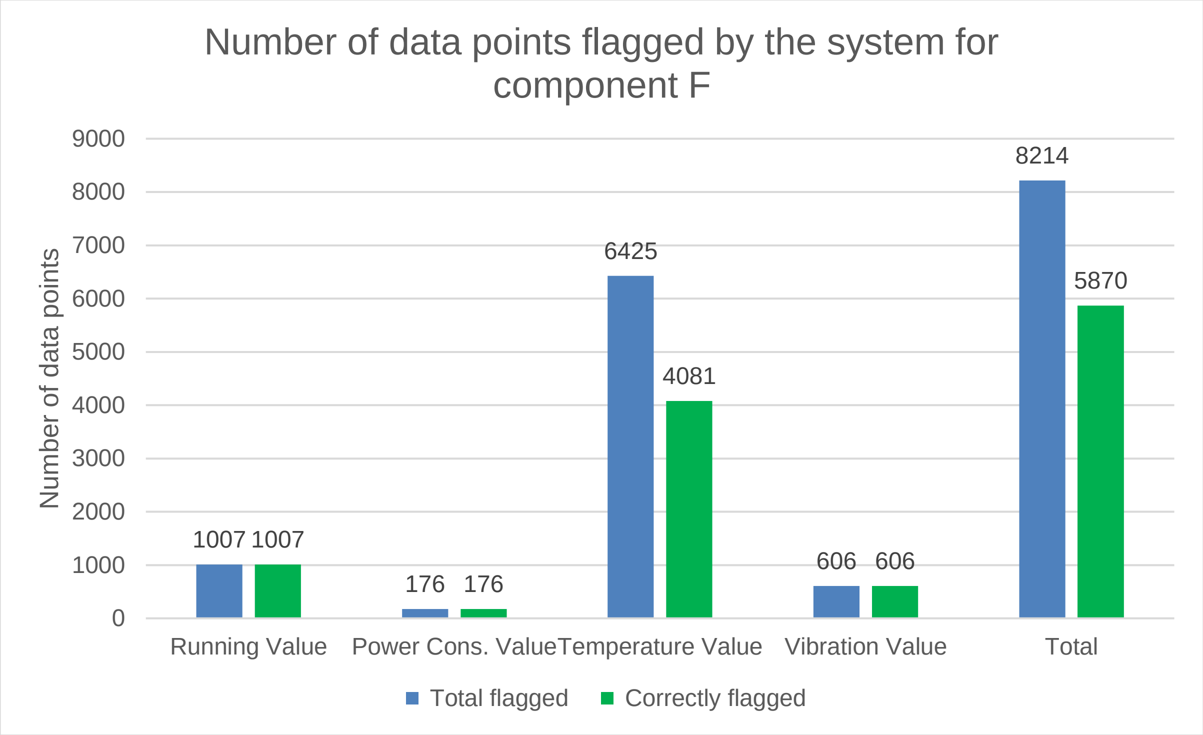

Figure 36: Bar chart displaying flagged data points for component F

The system struggled classifying the temperatures for component F, with only ~ 63% of the points being correctly flagged as low integrity points. This was due to the component constantly switching between the "on" and "off" state, creating multiple transition states. This can be addressed by taking transition states into consideration when calculating the data integrity.

3.3.7) Component G

Component G is a fridge plant located on Site E. The component has a tendency to contain high vibration data points. The statistics for component G's data integrity are summarised in Figure 37.

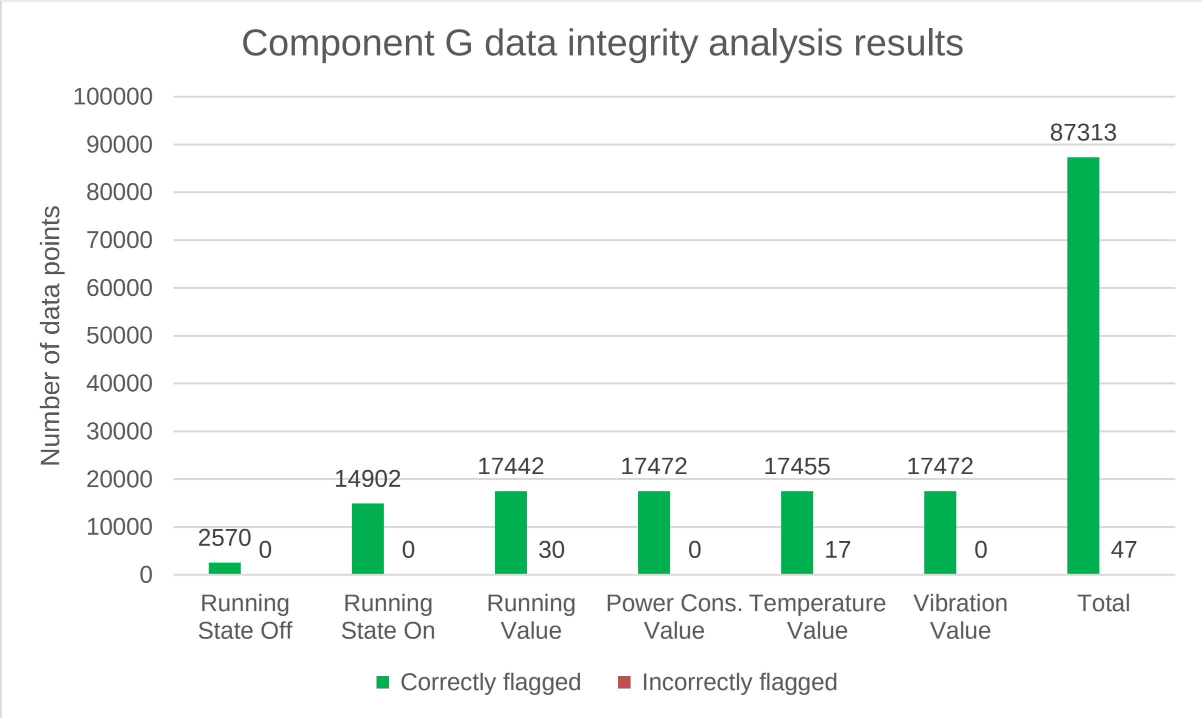

Figure 37: Bar chart displaying analysis results for component G

The abovementioned figure illustrates that the system was able to classify low integrity points for component G without much complication. The system did incorrectly flag the running status values in very specific conditions. When the running value had data loss, and the power consumption had a valid value, the data loss was treated as if it was a valid measurement. This anomaly occurred due to a translation error made by the system in which the absent data for the running status value was filled by a positive value. According to the system truth table, the data point was classified correctly, however, the data point itself was incorrect. This caused the system to misclassify some of the data loss for the running value data stream. The analysis results of the flagged data points for component G are displayed in Figure 38.

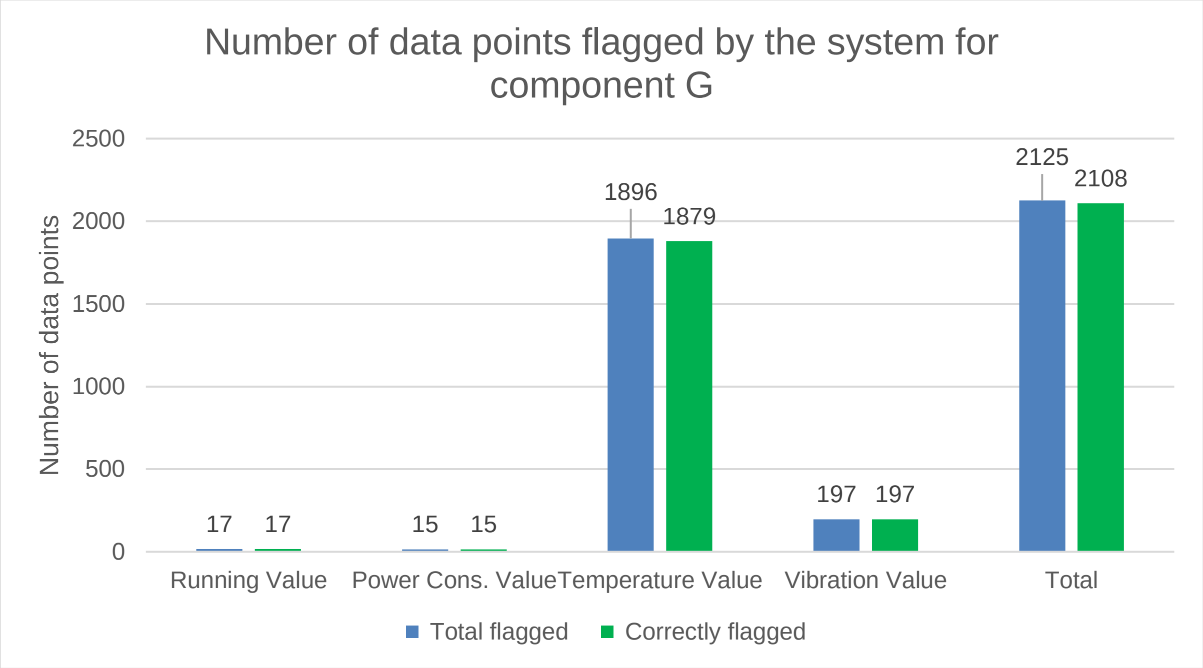

Figure 38: Bar chart displaying flagged data points for component G

As displayed by the abovementioned figure, the system was able to accurately classify data points in terms of their integrity. Similar to the analysis of other components, the system again struggled to classify the temperature measurements correctly during transition states.

3.3.8) Component H

Component H is a pump located on Site F. The component has questionable power consumption data points as they seem much higher than what is expected for the component.

The statistics for component H's data integrity are summarised in Figure 39.

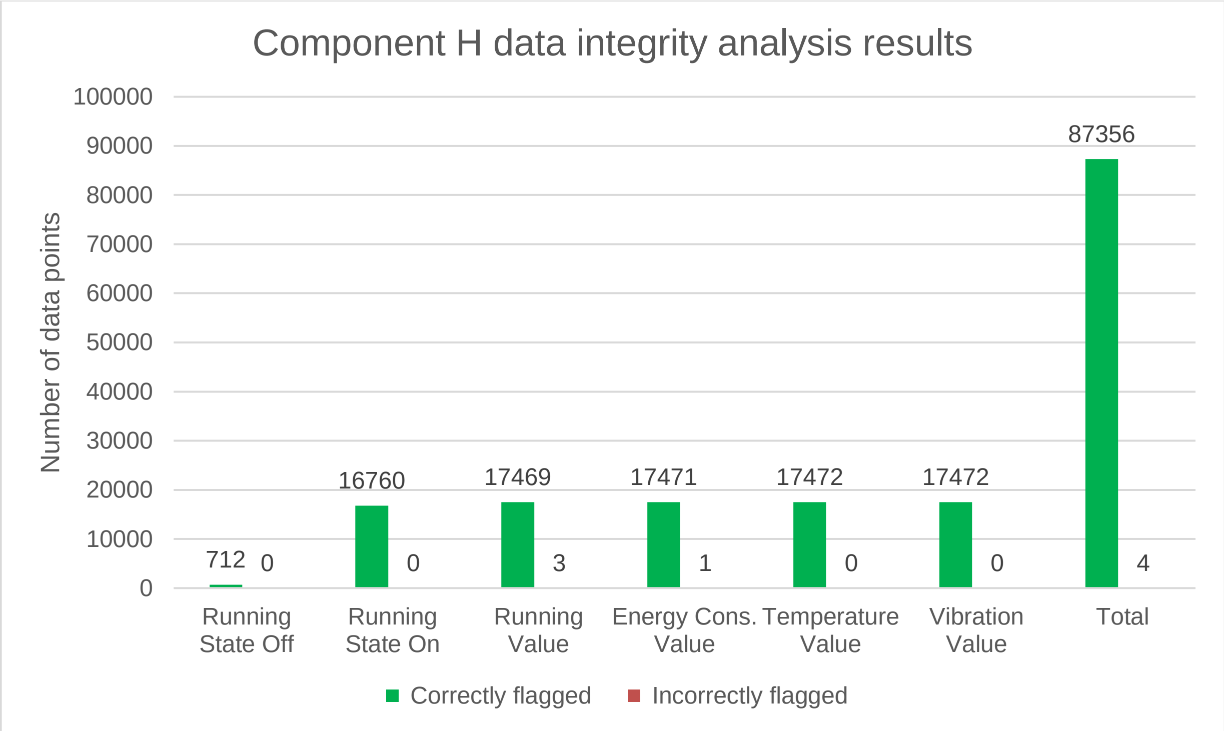

Figure 39: Bar chart displaying analysis results for component H

In the abovementioned figure, it can be seen that the system was able to flag the measurements for component H. The analysis yielded an extremely high accuracy of classification, which was unexpected as some shortcomings of the system had been identified by the time component H was analysed. Upon further investigation into the results, it was discovered that the system had correctly classified the data points according to the component configuration.

It was also discovered that the client had configured the component incorrectly. This resulted in the wrong characteristics being flagged for having low integrity points. Table 8 was created upon some assumptions, with the most important being that the power consumption values are always correct. This assumption allows the system to then verify whether the running status value is correct and determine whether the component is in the "on" or "off" state. The running status and power consumption for component H for a random day are compared in Figure 40.

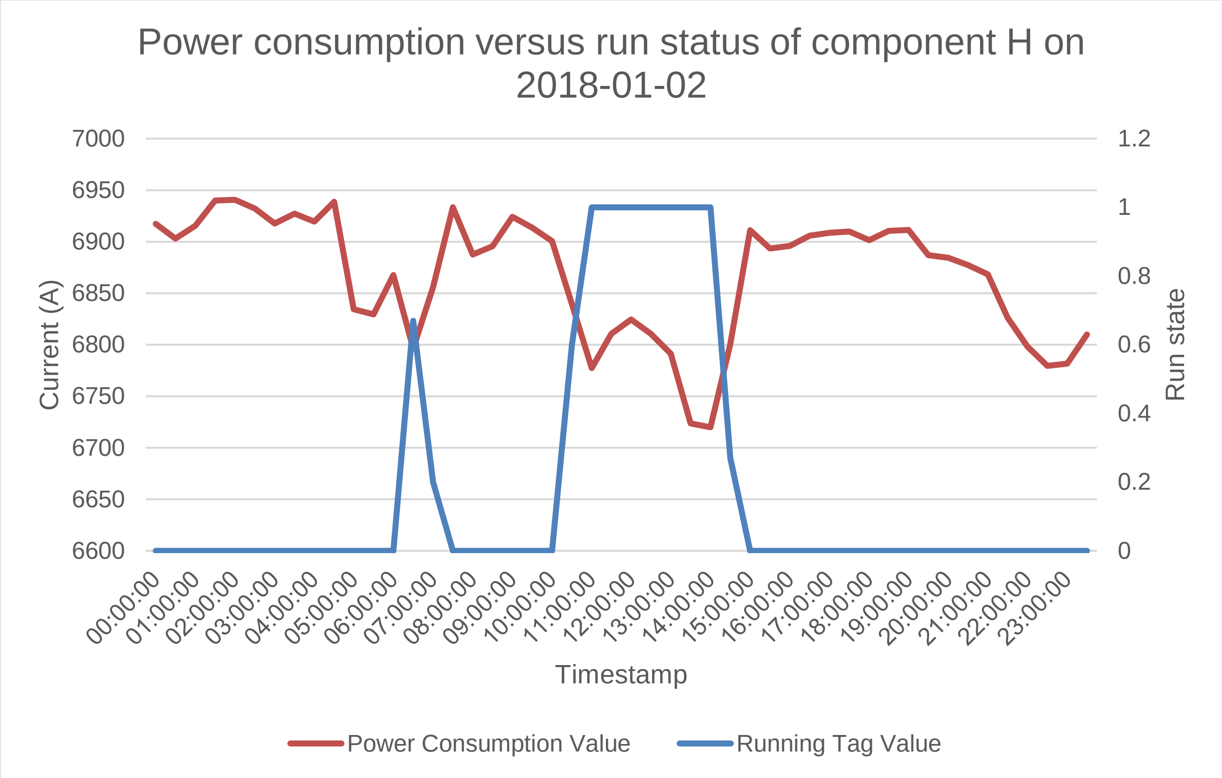

Figure 40: Line chart comparing power consumption to run status of component H

The above figure shows that for the duration of the day, the component consumes power. It also shows that the running status is only “on” for two sections of the day. With the assumption that the power consumption is correct, the running status values were classified as incorrect and the component was seen as running for the entire day. The temperature and vibration measurements are compared to the power consumption in Figure 41.

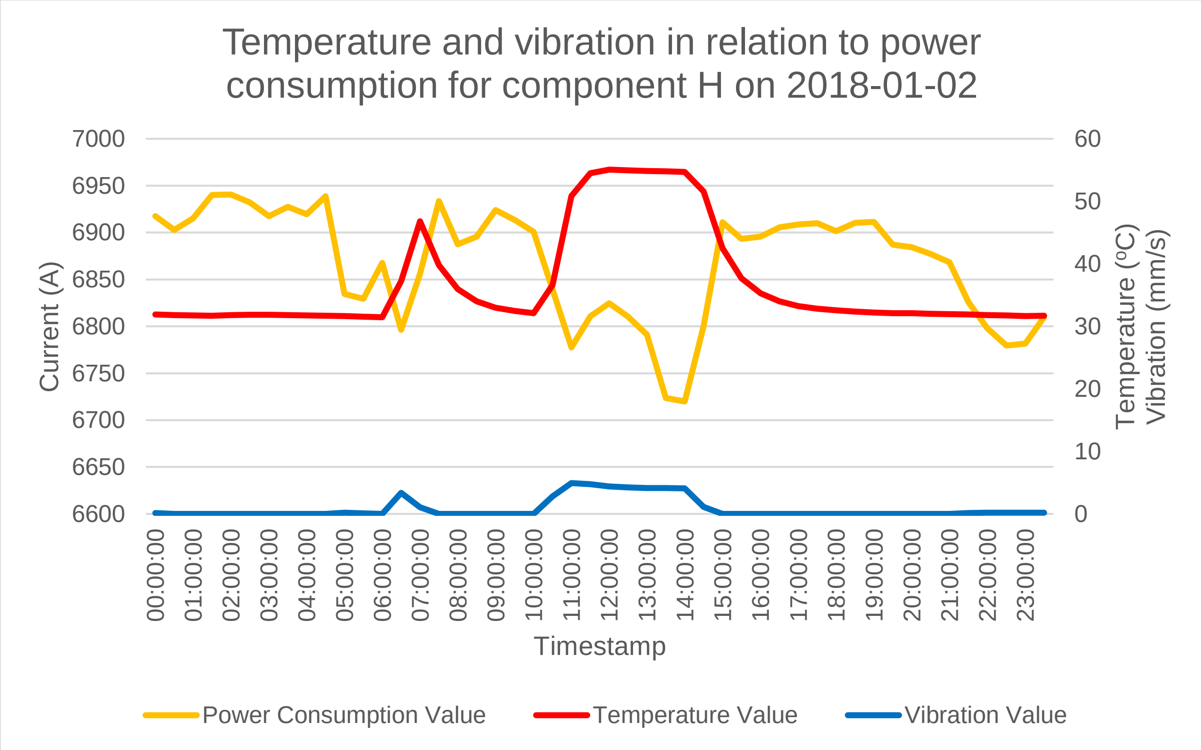

Figure 41: Line chart of temperature, vibration and power cons for component H

The abovementioned figure displays a correlation between the temperature and vibration measurements. This correlation is, however, not shared with the power consumption measurements. As a result, the temperature measurements are flagged as low integrity points as the component is seen to be permanently “on”. The temperature and vibration measurements are compared to the running status measurements in Figure 42.

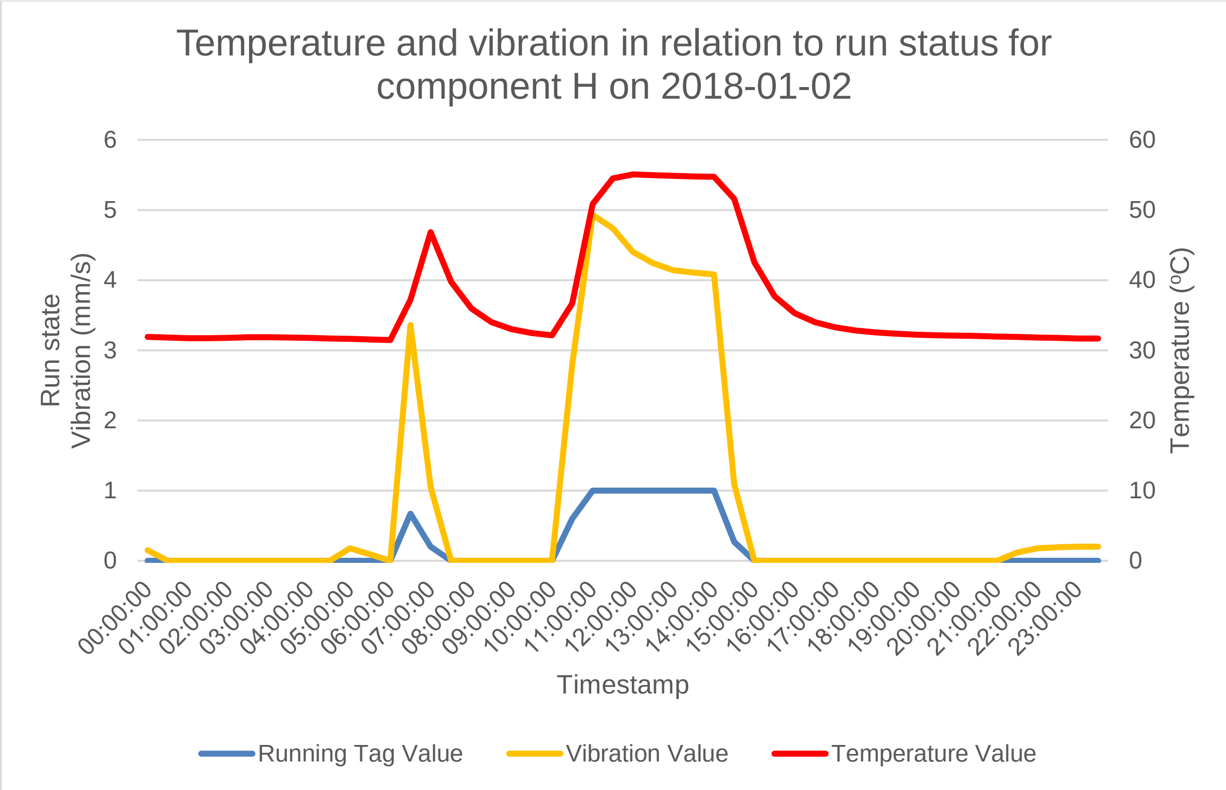

Figure 42: Line chart of temperature, vibration and run status for component H

The abovementioned figure displays the same correlation between the temperature and vibration measurements discussed earlier. However, this correlation is shared by the running status measurements, creating a more realistic representation of the component conditions for the day. After reviewing multiple days, it was confirmed that the power consumption tag was incorrectly linked to the component as the temperature, vibration, and running status measurements correlate for the entire dataset. This exposes a shortcoming in the system, as it does not verify the assumptions made for Table 8 before applying it.

The analysis results of the flagged data points for component H are displayed in Figure 43.

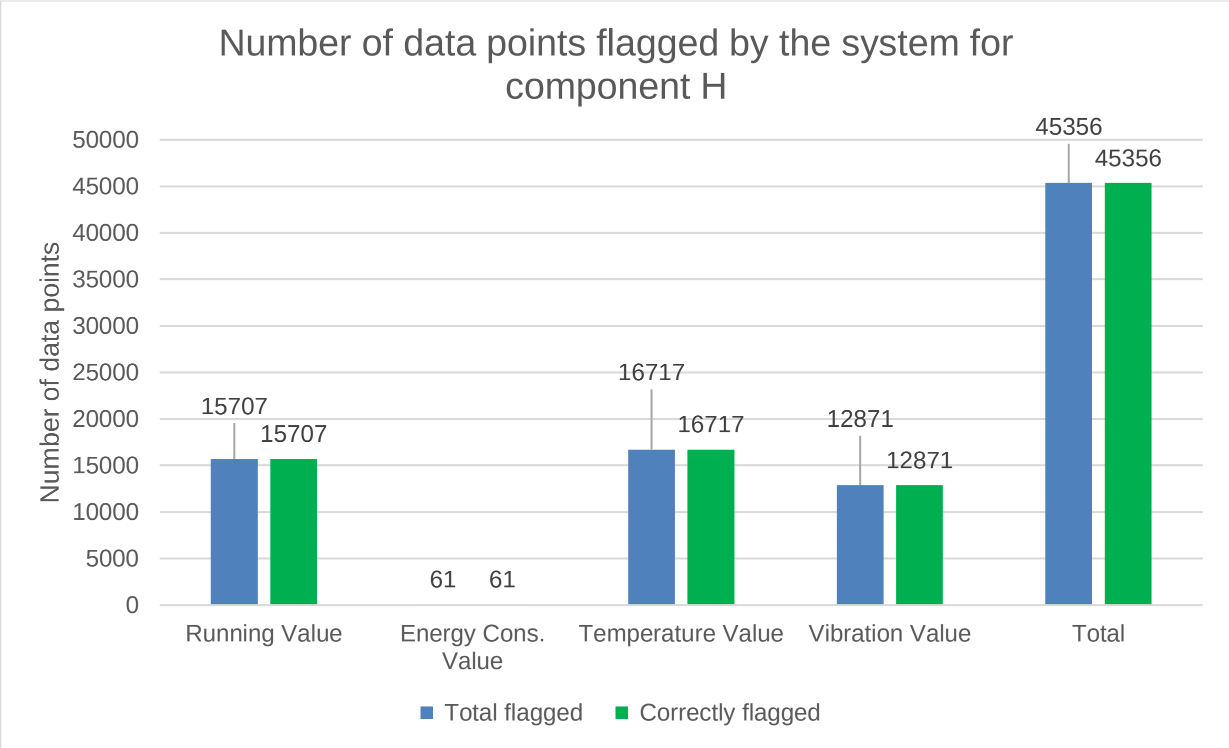

Figure 43: Bar chart displaying flagged data points for component H

As discussed earlier, the results in the abovementioned figure shows that the system was able to accurately classify the measurements based on the component configuration. However, due to the incorrect component configuration, the results should be closer to the opposite with the power consumption measurements being flagged as low integrity points while the running status value, temperature, and vibration measurements should have fewer flagged low integrity points.

3.3.9) Overview

The system was able to correctly classify low integrity data points with an average accuracy of ~93.86% for 698 880 data points. All eight of the case study components contained low integrity data points, but to a varying degree. This is displayed in Figure 44.

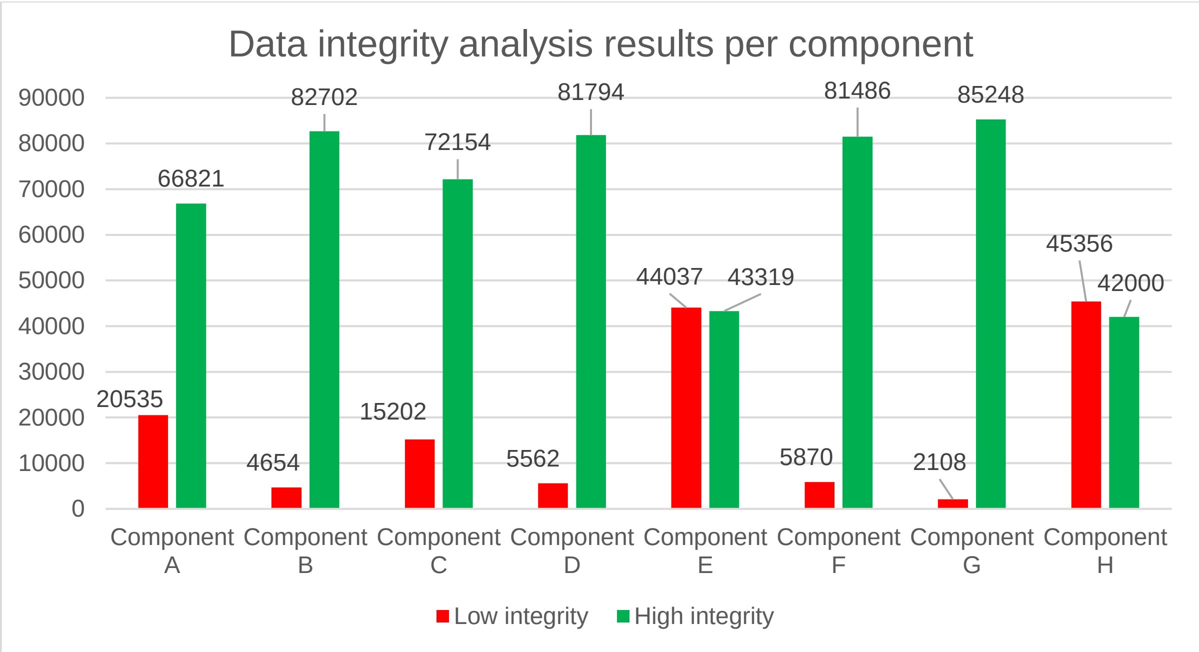

Figure 44: Bar chart displaying component level summary of characteristic results

The abovementioned graph shows the summary of low integrity versus high integrity data points identified per component. From the eight components, two components (E and H) suffer from extremely high levels of low integrity data points. Both of these components registered more faulty measurements than trustworthy measurements for an entire year. This is worrisome as it could imply that the maintenance schedules for and operating conditions of these components are not ideal.

If the maintenance schedules are not optimised, it could have negative monetary implications for the client. If unnecessary maintenance is scheduled for the components, unnecessary downtime and production losses could be the end result. On the other hand, if too little maintenance is scheduled, the components could break down unexpectedly.

If the components are operating in non-ideal conditions, they could deteriorate faster than planned for. This could result in early decommissioning or replacement of equipment, which has a negative monetary impact on the client.

The data integrity analysis results were summarised per component type to gain a better understanding of which components tend to experience measurement issues. This is displayed in Figure 45.

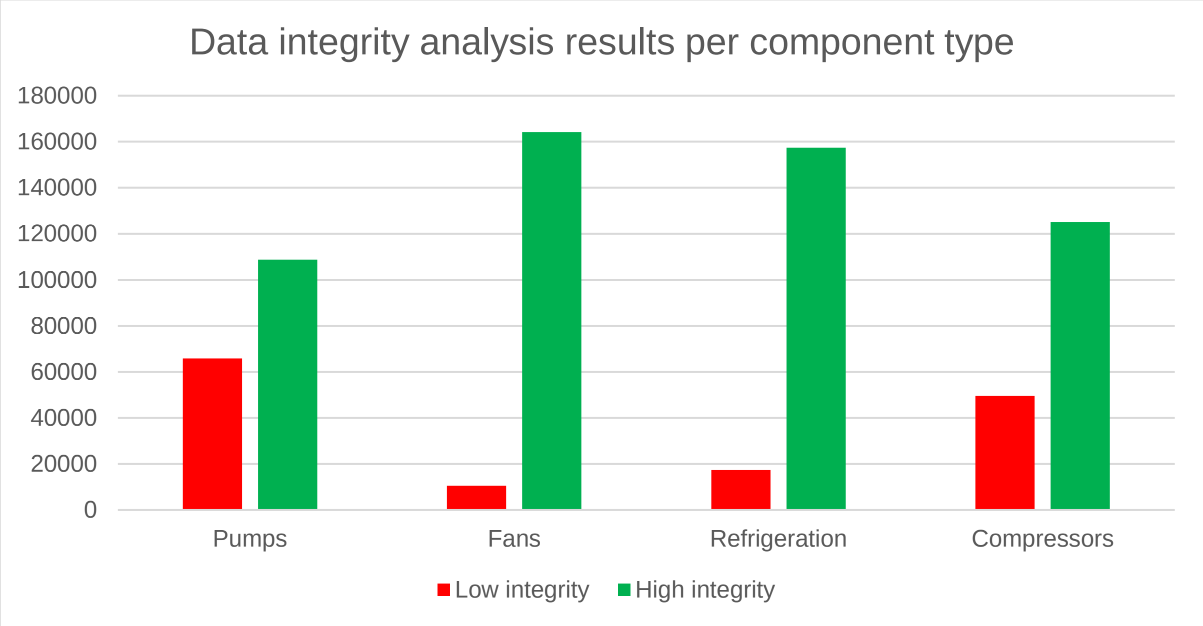

Figure 45: Bar chart displaying integrity summary per component type

As seen in the abovementioned figure, high and low integrity data points were categorised per component type. From this figure, it can be seen that all component types experience low integrity data points. However, it can also be seen that pumps and compressors have a higher tendency to experience low integrity data points. This is presumably due to the operating conditions in which these component types are located. This can be further investigated by exploring the low integrity data point results per component type, as seen in Figure 46.

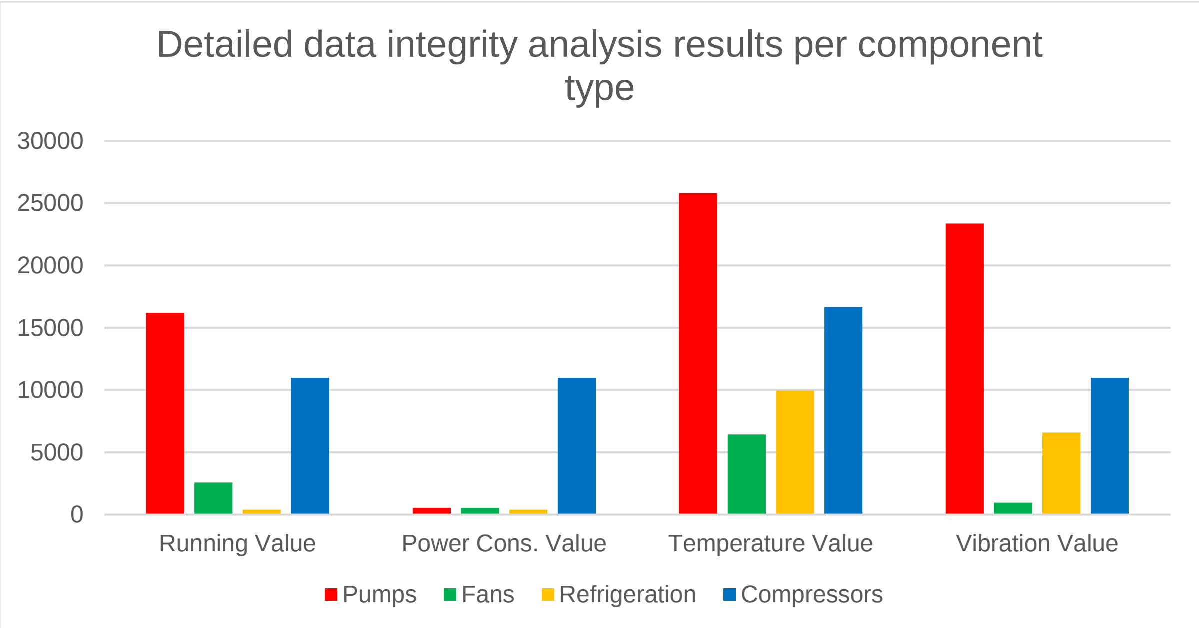

Figure 46: Bar chart showing low integrity breakdown per component type

The abovementioned figure displays the breakdown of low integrity data points flagged by the system for each component type. It can be seen that compressors have data problems for all analysed characteristics. This could imply that the component is located in areas where measuring equipment either:

- malfunctions,
- are not repaired and calibrated regularly, or
- the networks over which the measurements are sent are of low quality and that these measurements are corrupted during transmission.

Pumps experience a significant amount data problems when it comes to running value, temperature and vibration measurements. This could be due to the conditions in which the pumps are operated. A likely cause for these specific issues could be that pumps vibrate constantly during operation, resulting in measurement equipment disconnecting. Sudden bursts of vibration caused by constantly stopping and starting pumps could also contribute to the disconnection of measuring equipment.

Refrigeration has low integrity points mostly for temperature and vibration. As the refrigeration plants are used to cool down products, such as hot water, fluctuating temperature measurements are expected. These fluctuations could make it easier for faulty readings to be overlooked.

Fans, similar to pumps, experience mostly low integrity data readings for running value, temperature and vibration. This might be caused by the way fans work. Once a fan is switched off, it does not stop immediately. These moving parts cause vibration that leads to increased temperatures.

The system flagged low integrity data point for all of the measured characteristics, as displayed in Figure 47.

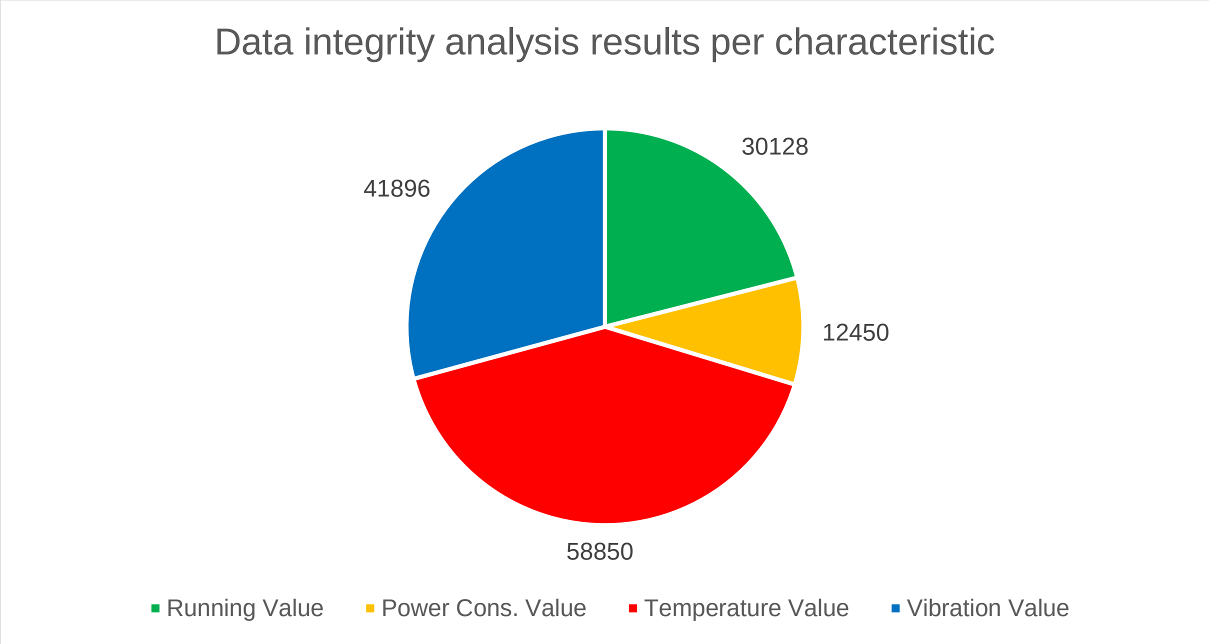

Figure 47: Pie chart displaying characteristic low integrity point summary

The abovementioned graph displays the breakdown of low integrity data points for the entire case study. As seen, the majority of the low integrity points fall into one of three characteristics, namely temperature, vibration or running value.

If a running value was flagged as low integrity, it was almost guaranteed that either the vibration and/or temperature would also be faulty. The running value is important for the existing CM system as calculations are only done when the component is in the "on" state. With the inclusion of the data integrity analysis system, the existing system would be able to give more accurate monitoring results.

The system struggled to classify temperature measurements in transition states as it works on a Boolean state principle i.e. the system only considers a component as "on" or "off". Despite this, the system was able to identify close to 60 000 low integrity temperature measurements.

During verification, the error constant (E_C) had to be doubled in order to obtain the desired classification results of temperature measurements. After analysing all case study component results, it was discovered that this could have been avoided if the calculated ambient temperature profile in Appendix A was scaled towards the data set. This would calibrate the system further by ensuring that the temperature profile correlated better to the environmental conditions experienced by each component. As a result, the error constant could be decreased to more accurately classify low integrity points. However, if the error constant were to be decreased too much, the system would falsely flag data points when natural environmental events were to occur.

The system also incorrectly classifies data points when one or more of the core assumptions used to create Table 8 is violated. The system can be made more accurate by calibrating it more accurately for each component.

The system only makes use of contextual data and does not consider that a data stream can be erroneous. There were instances were data streams had hanging data that were within the error range of the profile data, resulting in them being flagged as high integrity points. In reality, these values should be flagged to prompt the client to inspect their measuring equipment.

3.4) Additional benefits identified

Two main additional benefits were identified during the analysis of components A, E, F, and H:

- Faulty measuring equipment identification: Indicated by impossible values, faulty or disconnected measuring equipment was identified after reviewing the results of the system. This identification can be used to determine a CBM schedule for the measuring equipment, which in return would ensure high quality data for the CM platform. This could reduce the number of faulty and / or missed equipment issues.
- Component configuration insurance: As explained in 3.3.8), human error introduced a faulty configuration into the system. This can result in inefficient CBM schedules as the equipment can be misdiagnosed by the current platform. After analysing the results of the system analysis, this incorrect configuration was identified and rectified.

3.5) Summary

In this chapter, the system was applied to a case study to validate that the problem identified in Section 1.7) was addressed.

Criteria used to decide on a case study was discussed in Section 3.2).

Section 3.3) describes the case study and which components were chosen to analyse. The method for calibrating the system for each component was discussed before the results for each component analysis were discussed in Section 3.3.1) through Section 3.3.8), with an overview of the entire case study in Section 3.3.9).

In Section 3.4) some additional benefits were identified which were provided by the system.

Chapter 4
Conclusion

4.1) Discussion

Chapter 1) consisted of a literature review in which shortcomings in the current literature were identified. This was used to create a need for the study, namely validating the integrity of single source CM data, along with study objectives.

Chapter 2) described how the study objectives would be addressed. A methodology was developed to adapt an existing system to address the identified problem. A new system would be created and combined with the existing system to expand the capabilities of the current system.

Chapter 2.4) discussed the design of a system and its' models to address the shortcomings identified. The newly designed system would make use of contextual data streams, linked to models, to determine the data integrity of each stream. The results would be saved along with the original data to be used for analysis at a later stage.

Chapter 2.5) discussed how the solution was verified to ensure that it would address the shortcomings in literature. One weeks' error-free test data was used to calibrate the system. After two iterations, the system gave a high success rate at classifying low integrity data points. One weeks' erroneous test data was then sent through the system and, after two iterations, the system was capable of accurately classifying data points as being of either high or low integrity.

Chapter 3) discussed how the solution was implemented on a case study. The criteria for implementation and details regarding the chosen case study were explained. Screenshots were supplied to demonstrate how components were set up so the developed system could perform analysis on the data streams.

Chapter 3.3) discussed the results of the case study in which the system validated the integrity of data points for eight different components, spanning across six different sites. In total, 698 880 data points were analysed, with roughly 93.86% of the points being correctly classified as being either high or low integrity points. It was concluded that the case study had a large problem with data loss and faulty measurement equipment.

In Chapter 3.4), additional benefits of the system were identified. These benefits were found to help a company reduce human error in their system and ultimately provide more efficient services to their clients.

The objectives identified in Section 1.7) address the need for this study. Whether they were addressed and where are summarised in Table 11.

Table 11: Verification of met study objectives

Study objective	Section	Addressed
Develop a system	2.4.3)	✓
Create models	2.4.2)	✓
Link data streams	2.4.2)	✓
Calibrate models	2.5.1)	✓
Determine data integrity	2.5.2)	✓
Flag low integrity data points		✓

4.2) Recommendation for future work

Some shortcomings were identified whilst applying the system on a case study. These shortcomings and their potential solutions are discussed below.

4.2.1) Include data quality

As stated in Section 1.7), a similar study, [31], was conducted which calculates the data quality per data stream. Including these results into the new system could result in more accurate classifications of the data integrity. This would address some of the shortcomings of the system identified in Section 3.3.9) and can be used to verify the assumptions made for Table 8.

4.2.2) Verify power consumption

As stated in Section 2.4.2), some assumptions were made to create Table 8. Ultimately it was decided that power consumption would be more reliable to work with than the running status value. However, as shown in Section 3.3), this is not always the case. To improve the system, the running state of the component needs to be calculated more accurately. Verifying the power consumption before assuming that it is correct will reduce the amount of erroneous run state calculations. This can be done in two ways:

- Include temperature in running state calculation: The temperature value can be used in combination with the running status value to verify whether the power consumption seems trustworthy. Afterwards, the running state can be calculated using the existing truth table. The truth table can also be expanded to include the temperature value, calculating the running state using all three values.

- Verify data quality of power consumption values: As mentioned in Section 4.2.1), the data quality results can be used to verify the power consumption value. If the quality is acceptable, the existing truth table can be used to calculate the running state. If not, a truth table prioritising the running status value can be used to calculate whether the component is running.

4.2.3) Incorporate results into current system

The system currently saves the result of the data integrity calculation into the No-SQL database. The existing CM platform can be modified to display the data integrity analysis result, informing the user of low integrity data. This can help reduce the amount of falsely-flagged instances as well as help the company provide a more efficient service to their clients. This could also motivate the clients to ensure their measuring equipment is maintained and accurate.

4.2.4) Calibrated profile data

As described throughout Section 3.3), the profile data calculated does not accurately represent the component being analysed. To make this profile more accurate, one or several of the following can be done:

- Seasonal profiles: The profiles used for ambient temperature and environmental vibration can be split into several profiles. Having a profile calculated using only data from a specific season should make the profile more accurate for those seasons.
- Component type profiles: Rather than using the data from the four different components to create a single profile, a profile can be calculated for each component type. This should increase the accuracy of that profile when applied to the specific component type.
- Component-specific profiles: To ensure that a profile is calibrated, it can be created from historical data for the specific component. This will ensure that environmental changes and factors are minimised and increase the accuracy of the profile when applied to new data for the component.

4.2.5) Transition state calculations

Additional equations can be derived and implemented to calculate the current temperature of a component in the transition states. These transition states occur when a component goes from the "off" state to the "on" state and vice versa. This will reduce the amount of faultily flagged temperature data points.

4.2.6) Notification system for results

The company uses a reporting system to inform their clients of the state of operations. This reporting system is also used within the company to keep track of project performance, equipment conditions, and deadlines. These reports include the data measured and summarised to different levels within the company e.g. to component level, site level, etc. Including the data integrity results in these reports could help the company motivate measuring equipment maintenance and replacement. This could also give the client more confidence in the reports and the decisions made on the data.

An instant notification system, such as a SMS system, could help to reduce the time it takes to check on faulty measuring equipment. If a notification was received when a faulty value, such as an impossible measurement, started appearing, an investigation could be launched. This could reduce the amount of low integrity data being captured into the system, helping the company become more efficient.

4.3) Closure

The system implemented to address the study objectives and fill the gap identified in literature proved to be useful. Some limitations to the system were identified and future work was recommended to address them. Some additional benefits of the system were identified which adds more value to it. These benefits could help reduce the impact of human error in CBM platforms.

5) References

[1] A. Zaslavsky, C. Perera, and D. Georgakopoulos, ‘Sensing as a Service and Big Data’, in *Proceedings of the International Conference on Advances in Cloud Computing (ACC)*, 2012, pp. 21–29.

[2] M. Baily and J. Manyika, ‘Is manufacturing “cool” again?’, 2013. [Online]. Available: https://www.mckinsey.com/mgi/overview/in-the-news/is-manufacturing-cool-again. [Accessed: 05-Nov-2019].

[3] S. Yin and O. Kaynak, ‘Big Data for Modern Industry: Challenges and Trends’, in *Proceedings of the IEEE*, 2015, vol. 103, no. 2, pp. 143–146.

[4] B. Esmaeilian, S. Behdad, and B. Wang, ‘The evolution and future of manufacturing: A review’, *J. Manuf. Syst.*, vol. 39, pp. 79–100, 2016.

[5] C. McCaffrey, J. Flak, I. Marttila, N. Pesonen, and P. Pursula, ‘Development of a reader device for fully passive wireless sensors’, in *2017 IEEE Sensors*, 2017, pp. 1–3.

[6] A. McAfee and E. Brynjolfsson, ‘Big Data : The Management Revolution’, *Harvard Business Review*, 2012.

[7] M. Madhikermi, A. Buda, B. Dave, and K. Främling, ‘Key data quality pitfalls for condition based maintenance’, in *2017 2nd International Conference on System Reliability and Safety, ICSRS 2017*, 2018, pp. 474–480.

[8] C. Adrian, R. Abdullah, R. Atan, and Y. Y. Jusoh, ‘Expert Review on Big Data Analytics Implementation Model in Data-driven Decision-Making’, in *2018 4th International Conference on Information Retrieval and Knowledge Management (CAMP)*, 2018, pp. 1–5.

[9] O. Kwon, N. Lee, and B. Shin, ‘Data quality management, data usage experience and acquisition intention of big data analytics’, *Int. J. Inf. Manage.*, vol. 34, no. 3, pp. 387–394, 2014.

[10] V. Grover, R. H. L. Chiang, T. P. Liang, and D. Zhang, ‘Creating Strategic Business Value from Big Data Analytics: A Research Framework’, *J. Manag. Inf. Syst.*, vol. 35, no. 2, pp. 388–423, 2018.

[11] R. Singh and K. Singh, ‘A Descriptive Classification of Causes of Data Quality Problems

in Data Warehousing', *IJCSI Int. J. Comput. Sci. Issues*, vol. 7, no. 3, pp. 41–50, 2010.

[12] R. M. C. Ratnayake and M. Kusumawardhani, 'Reliability analysis of condition monitoring data on aging plants: A case study from topside static mechanical systems', in *IEEE International Conference on Industrial Engineering and Engineering Management*, 2014, pp. 1401–1405.

[13] A. Haug, F. Zachariassen, and D. van Liempd, 'The costs of poor data quality', *J. Ind. Eng. Manag.*, vol. 4, no. 2, pp. 168–193, 2011.

[14] A. Haug and J. Stentoft, 'Barriers to master data quality', *J. Enterp. Inf. Manag.*, vol. 24, no. 3, pp. 288–303, 2011.

[15] S. van Jaarsveld, 'Developing an integrated information system to assess the operational condition of deep level mine equipment', North-West University, 2018.

[16] J. A. Deysel, M. Kleingeld, and C. J. R. Kriel, 'DSM strategies to reduce electricity costs on platinum mines', in *Proceedings of the Conference on the Industrial and Commercial Use of Energy, ICUE*, 2015, pp. 89–96.

[17] C. J. R. Kriel, J. H. Marais, and M. Kleingeld, 'Modernising underground compressed air DSM projects to reduce operating costs', in *Proceedings of the Conference on the Industrial and Commercial Use of Energy, ICUE*, 2014, pp. 1–6.

[18] W. Conradie, M. Kleingeld, and F. G. Jansen Van Rensburg, 'Development of a model to predict cost savings of reconfigured mine water reticulation systems', in *Proceedings of the Conference on the Industrial and Commercial Use of Energy, ICUE*, 2019, pp. 1–6.

[19] J. L. Buys, M. Kleingeld, and C. Cilliers, 'Optimising the refrigeration and cooling system of a platinum mine', in *Proceedings of the Conference on the Industrial and Commercial Use of Energy, ICUE*, 2015, pp. 36–43.

[20] B. Pascoe, H. J. Groenewald, and M. Kleingeld, 'Improving mine compressed air network efficiency through demand and supply control', in *2017 International Conference on the Industrial and Commercial Use of Energy (ICUE)*, 2017, pp. 1–5.

[21] H. G. Brand, M. Kleingeld, P. Mare, and A. J. Schutte, 'The effect of surface Bulk Air Cooler water flow on deep-level mine temperatures', in *Proceedings of the Conference on the Industrial and Commercial Use of Energy, ICUE*, 2017, pp. 1–6.

[22] S. van Jaarsveld, S. W. van Heerden, and J. van Rensburg, 'Development of a Condition

Monitoring Information System for deep level mines', in *SAIIE28 Proceedings*, 2017.

[23] N. Solomakhina, T. Hubauer, S. Lamparter, M. Roshchin, and S. Grimm, 'Extending statistical data quality improvement with explicit domain models', in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 720–725.

[24] J. Huang, 'From Big Data to Knowledge: Issues of Provenance, Trust, and Scientific Computing Integrity', in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 2197–2205.

[25] A. G. S. Gous, W. Booysen, and W. Hamer, 'Data quality evaluation for measurement and verification processes', in *Industrial & Commercial use of Energy (ICUE)*, 2016, pp. 9–15.

[26] J. Y. Song, Y. P. Guo, S. J. Zhang, and J. D. Hao, 'Data Integrity Assessment Method Based on Data Dependence', in *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, 2018, pp. 1–9.

[27] 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)*, 2014, pp. 1–5.

[28] Wayne W. Eckerson, *Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data*, 1st ed. Chatsworth: 101communications LLC, 2006.

[29] N. Laranjeiro, S. N. Soydemir, and J. Bernardino, 'A Survey on Data Quality: Classifying Poor Data', in *2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2016, pp. 179–188.

[30] Q. Fu and J. M. Easton, 'Understanding data quality: Ensuring data quality by design in the rail industry', in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 3792–3799.

[31] A. Goosen, 'A system to quantify industrial data quality', North-West University, 2018.

[32] W. Hamer, 'A practical approach to quantify RSA Section 12L EE tax incentives for large industry', North-West University, 2016.

[33] T. Hongxun *et al.*, 'Data quality assessment for on-line monitoring and measuring system of power quality based on big data and data provenance theory', in *2018 3rd IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2018,

pp. 248–252.

[34] M. A. Hayes and M. A. M. Capretz, ‘Contextual Anomaly Detection in Big Sensor Data’, in *2014 IEEE International Congress on Big Data*, 2014, pp. 64–71.

[35] D. J. Hill, B. S. Minsker, and E. Amir, ‘Real-time Bayesian anomaly detection in streaming environmental data’, *Water Resour. Res.*, vol. 45, no. 4, 2009.

[36] A. K. S. Jardine, D. Lin, and D. Banjevic, ‘A review on machinery diagnostics and prognostics implementing condition-based maintenance’, *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, 2006.

[37] F. De Carlo and M. A. Arleo, ‘Maintenance cost optimization in condition based maintenance: A case study for critical facilities’, *Int. J. Eng. Technol.*, vol. 5, no. 5, pp. 4296–4302, 2013.

[38] S. Jong-Ho and J. Hong-Bae, ‘On condition based maintenance policy’, *J. Comput. Des. Eng.*, vol. 2, no. 2, pp. 119–127, 2015.

[39] B. Chindondondo, L. Nyanga, A. Van der Merwe, T. Mupinga, and S. Mhlanga, ‘Development of a Condition Based Maintenance System for a sugar producing company’, in *SAIIE26 Proceedings*, 2014, pp. 1–14.

[40] M. Bengtsson, ‘On condition based maintenance and its implementation in industrial settings’, *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2007.

[41] F. Besnard, ‘On maintenance optimization for offshore wind farms’, Chalmers University of Technology, 2013.

[42] C. Romesis and Y. Li, ‘Condition-Based Maintenance for Gas Turbines Plants’, pp. 1–19.

[43] S. Telford, M. Mazhar, and I. Howard, ‘Condition Based Maintenance (CBM) in the Oil and Gas Industry: An Overview of Methods and Techniques’, in *International conference on Industrial Engineering and Operations Management*, 2011, pp. 1152–1159.

[44] A. Romanowski, ‘Big Data-Driven Contextual Processing Methods for Electrical Capacitance Tomography’, *IEEE Trans. Ind. Informatics*, vol. 15, no. 3, pp. 1609–1618, 2019.

[45] K. Kolomvatsos, C. Anagnostopoulos, and S. Hadjiefthymiades, ‘Data Fusion and Type-2

Fuzzy Inference in Contextual Data Stream Monitoring', *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 8, pp. 1839–1853, 2017.

[46] H. Decker, 'Quantifying the quality of stored data by measuring their integrity', in *2nd International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*, 2009, pp. 796–801.

[47] S. Hiremath and S. Kunte, 'A novel data auditing approach to achieve data privacy and data integrity in cloud computing', in *International Conference on Electrical, Electronics, Communication Computer Technologies and Optimization Techniques (ICEECCOT)*, 2018, pp. 306–310.

[48] S. Tan, L. Tan, X. Li, and Y. Jia, 'An efficient method for checking the integrity of data in the cloud', *China Commun.*, vol. 11, no. 9, pp. 68–81, 2014.

6.1) Appendix A

An ambient temperature profile is a representation of what the temperature of a component will be when not in operation due to environmental effects. A year's worth of data for four components were used. Half-hourly data was obtained and the average value for each timestamp over the year for all four components was obtained. The four components used were an industrial fan, an industrial pump, an industrial compressor, and an industrial refrigeration unit. The ambient temperature data set of the combining profile is displayed in Figure 48.

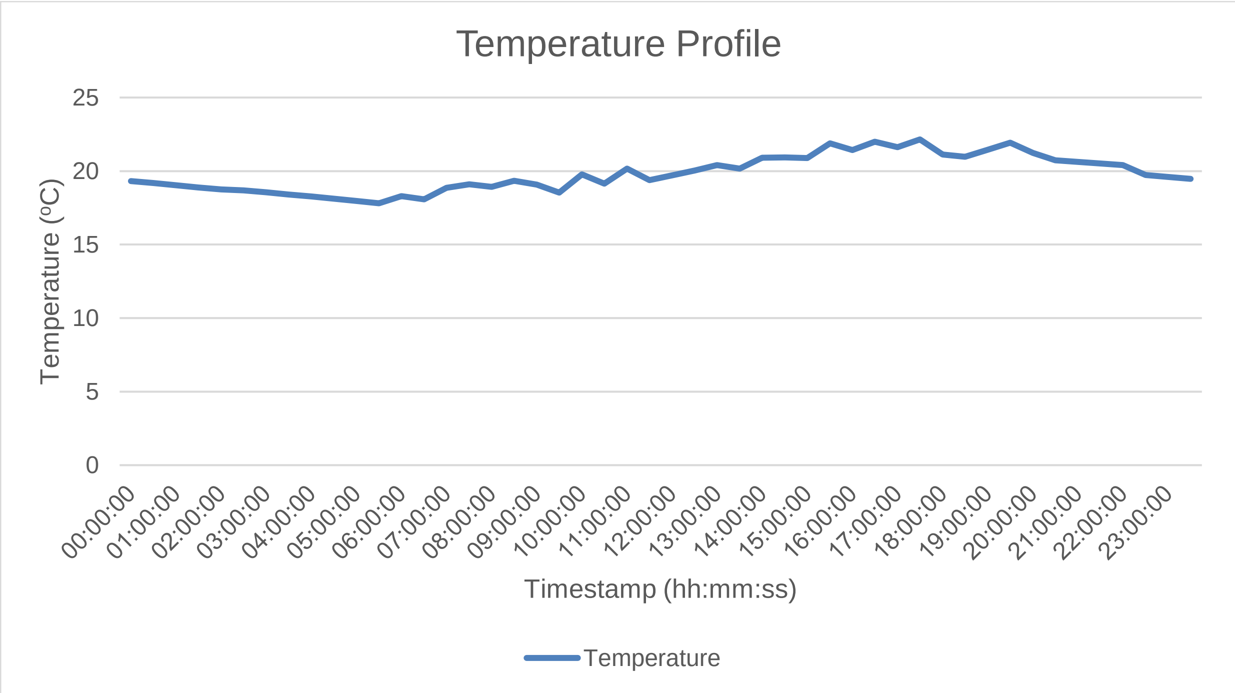

Figure 48: Ambient temperature profile data set

6.2) Appendix B

An environmental vibration profile is a representation of what the vibration experienced by a component will be when not in operation due to environmental effects. A year's worth of data for four components were used. Half-hourly data was obtained and the average value for each timestamp over the year for all four components were obtained. The four components used were an industrial fan, an industrial pump, an industrial compressor, and an industrial refrigeration unit. The environmental vibration data set of the combining profile is displayed in Figure 49.

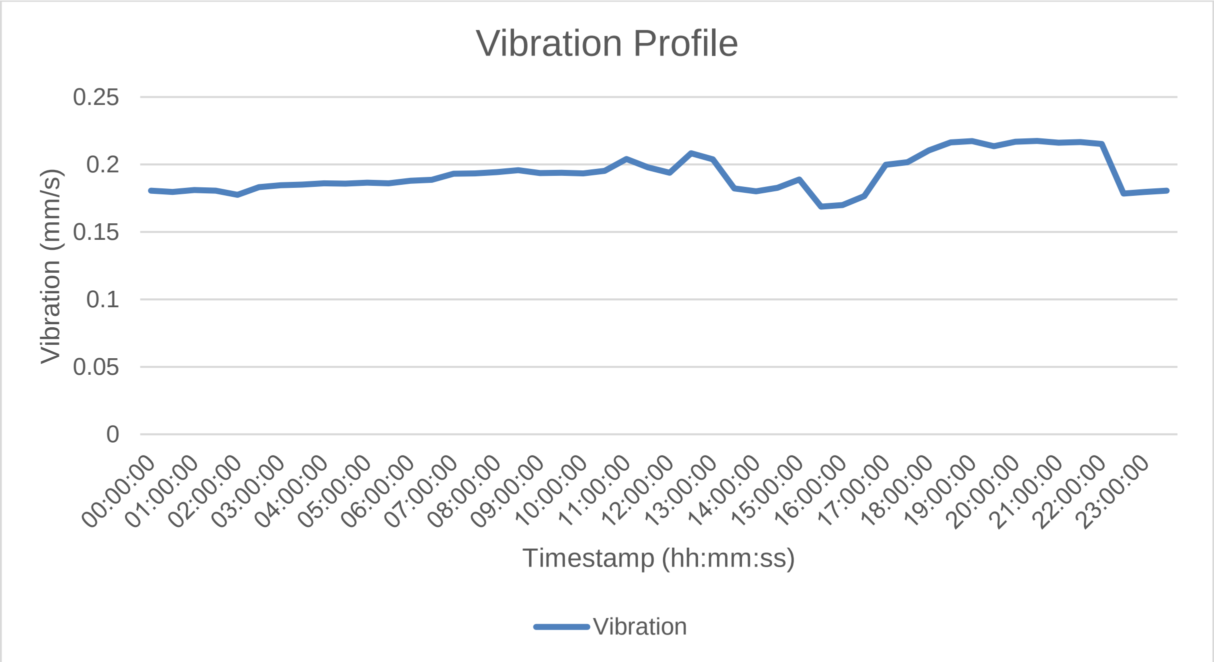

Figure 49: Environmental vibration profile data set

6.3) Appendix C

In order to test the system, a test case of a single component with all four characteristics linked was created. The data for the test case was fabricated manually for an entire day with no induced errors. This was used to ensure the system did not give any incorrect answers for clean data. Subsequently, errors were manually introduced to verify that the system could successfully identify them. The test case data set is listed in Table 12. Cells that have two values represent the values that were changed from being correct to faulty. The value in green is the original value whilst the red value is the erroneous value.

Table 12: Test data set used for verification with (red) and without (green) errors

Date	Time	Running status	Power consumption (A)	Temperature (^{o}C)	Vibration (mm/s)
2018-01-01	00:00:00	1	402.74	56.58	0.96
2018-01-01	00:30:00	1	403.24	56.3	0.96
2018-01-01	01:00:00	1	403.65	56.1 90	0.97
2018-01-01	01:30:00	1	402.75	55.84	0.96
2018-01-01	02:00:00	1	401.72	55.63	0.96
2018-01-01	02:30:00	1	401.64	55.35	0.95
2018-01-01	03:00:00	1	401.22	55.2	0.96
2018-01-01	03:30:00	1 0	401	54.9	0.96
2018-01-01	04:00:00	1	401.01	54.65	0.92 4
2018-01-01	04:30:00	1	402.33	54.39	0.96
2018-01-01	05:00:00	1	401	54.01	0.95
2018-01-01	05:30:00	1	402.88	53.75	0.96
2018-01-01	06:00:00	1	401.32	53.5	0.96
2018-01-01	06:30:00	1	398.45 180	53.3	0.96
2018-01-01	07:00:00	1	399.71	53.26	0.96
2018-01-01	07:30:00	1	401.29	53.32	0.96
2018-01-01	08:00:00	1	401.22	53.64	0.93 -0.4
2018-01-01	08:30:00	1	401.03	53.87	0.93
2018-01-01	09:00:00	1 0	402.15	54.08	0.92
2018-01-01	09:30:00	1	401.2	54.37	0.93
2018-01-01	10:00:00	1	401.18	54.62	0.95
2018-01-01	10:30:00	1	401.24	55	0.96

2018-01-01	11:00:00	1		399.01		55.37		0.96	
2018-01-01	11:30:00	1		399.42		55.73		0.96	
2018-01-01	12:00:00	1		400.51		56.21		0.96	
2018-01-01	12:30:00	1		399.26	8000	56.58		0.96	2.5
2018-01-01	13:00:00	1		396.74		57.03		0.93	
2018-01-01	13:30:00	1		396		57.44		0.93	
2018-01-01	14:00:00	1		399.02		57.89		0.93	
2018-01-01	14:30:00	1		397.72		58.31		0.92	
2018-01-01	15:00:00	1		396.3		58.65		0.93	
2018-01-01	15:30:00	1		395.9		58.81		0.91	
2018-01-01	16:00:00	1		396.65		58.95		0.9	
2018-01-01	16:30:00	1		396.04		59.05		0.92	
2018-01-01	17:00:00	1	0	395.85	450	59.2	78	0.92	0
2018-01-01	17:30:00	1	0	395.15		59.2		0.92	
2018-01-01	18:00:00	1	0	394.41		59.22		0.91	
2018-01-01	18:30:00	1		395.12		59.17		0.92	
2018-01-01	19:00:00	1		396.91		58.96		0.91	
2018-01-01	19:30:00	1		398.66		58.72		0.93	
2018-01-01	20:00:00	1		401.53		58.35		0.96	
2018-01-01	20:30:00	1		400.42		57.93		0.97	
2018-01-01	21:00:00	1		402.09		57.49		0.96	
2018-01-01	21:30:00	1		401.38		57.06	18	0.97	
2018-01-01	22:00:00	1		400.45		56.59		0.93	
2018-01-01	22:30:00	1		398.65		56.22		0.96	
2018-01-01	23:00:00	1		399.76		55.83		0.97	
2018-01-01	23:30:00	1		399.33		55.49		0.96	