CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Method for Anomaly Detection using Data Requirements

A practical implementation for field test data

Master's Thesis in Computer science and engineering

(Chitra Rajasekar, Rachana Ramakrishna)

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2026

Master's Thesis 2026

Method for Anomaly Detection using Data Requirements

A practical implementation for field test data

Chitra Rajasekar, Rachana Ramakrishna

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2026

Method for Anomaly Detection using Data Requirements
Chitra Rajasekar, Rachana Ramakrishna

Supervisors: Farnaz Fotrousi, Yi Peng, Department of Computer Science and Engineering
Industrial Supervisors: Christian Beckers, Erik Jansson, Volvo Penta
Examiner: Jennifer Horkoff, Department of Computer Science and Engineering

Master's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2026

Chitra Rajasekar, Rachana Ramakrishna
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Anomaly detection is an important problem across a wide range of industries, where the ability to identify rare and unusual patterns within large volumes of data can have significant operational, financial and security implications. This study presents an understanding of the problem statements and challenges involved in existing data pipelines for managing and analyzing sensor data to identify engine anomalies. To gain a better understanding of these challenges, interviews were conducted within the case company.

Industrial engines generate large volumes of field test data through embedded software and engine sensors. Embedded software processes this data efficiently to control devices, monitor performance, and ensure that engines operate reliably under resource constraints. Effective monitoring relies on sensor data, such as temperature, pressure, speed, and fuel consumption, to track engine performance in real time. These data are required to detect faults at an early stage, optimize efficiency, and ensure safe operation. Therefore, an effective domain driven anomaly detection method is developed that relies on data requirement specifications to ensure that the designed model is based on domain knowledge.

This thesis focuses on implementing a method for domain driven anomaly detection using data requirements derived from multiple sources, including interviews, technical documentation such as Database CAN files, and field test datasets collected from the case company Volvo Penta. The study aims to bridge the gap between raw sensor data and the definition of meaningful data requirements by distinguishing relevant signal anomalies from data error outliers. Furthermore, a domain driven anomaly detection method is designed as an artifact using the defined data requirements to validate anomaly thresholds and address these challenges.

In addition, an analysis of the detected anomalies, while considering both temporal context and multi-signal correlations, demonstrates that the proposed approach can accurately identify and classify outliers as either relevant anomalies or data error outliers. This approach improves the reliability of anomaly detection by reducing false interpretations and ensuring that anomalies are evaluated using well defined validation criteria. Consequently, the derived data requirements support the analysis and validation of the proposed anomaly detection method.

Keywords: Anomaly detection, Isolation forest algorithm, sensor datasets, unsupervised machine learning, data requirements, domain knowledge and Database CAN files

Acknowledgements

This thesis would have not been possible without the support of many people. We would like to express our sincere gratitude to our university supervisors, Farnaz Fotrousi and Yi Peng for their continuous support, guidance and encouragement. We are also deeply grateful to our industrial supervisors at Volvo Penta, Christian Beckers, Erik Jansson and the whole team for their insightful feedback throughout our thesis. Their practical insights and encouragement played a crucial work in guiding our thesis. We would also extend our appreciation to our examiner, Jennifer Horkoff, for her constructive feedback to enhance the quality of the study. Lastly, we would like to thank our partners, family and friends for their encouragement and patience throughout this journey.

Chitra Rajasekar, Rachana Ramakrishna, Gothenburg, July 2026

Contents

List of Figures **xi**

List of Tables **xiii**

1 Introduction **1**
1.1 Problem Description 1
1.2 Purpose of the Study 2
1.3 Significance of the Study 3
1.4 Research Questions 3
1.5 Thesis Outline 4

2 Background **5**
2.1 Data Requirements 5
2.1.1 Understanding data requirements and their sources 5
2.1.2 Data gathering process for data requirements 6
2.1.3 Challenges associated with data requirements 7
2.2 Anomalies 7
2.2.1 Types of anomalies 7
2.2.2 Types of anomaly detection technique 8
2.2.3 Anomaly detection problems and challenges 8

3 Related Work **11**
3.1 Statistical and Machine Learning Approaches for Anomaly Detection 11
3.2 Hybrid Approaches for Anomaly Detection 12
3.3 Domain Driven Anomaly Detection 13

4 Methods **15**
4.1 Problem Investigation 16
4.1.1 Interviews 16
4.1.2 Document Analysis 20
4.1.3 Data Requirements Derivation 22
4.2 Design Implementation 23
4.3 Evaluation 23
4.3.1 Expert Reviews 23
4.3.2 Quantitative Model Evaluation 25

5 Artifact **29**

5.1 Artifact: Method for Domain Driven Anomaly detection (DDA Method) 29
5.1.1 Input for DDA Method 30
5.1.2 Design of DDA method 31

6 Results **35**
6.1 Defined Data Requirements for Anomaly detection 35
6.1.1 Challenges in managing sensor data through interviews 35
6.1.2 Derived data requirements 38
6.2 Applied example of a domain driven anomaly detection method . . . 41
6.2.1 Root cause for the identified anomalies 42
6.3 Effectiveness of the DDA Method 46
6.3.1 Analysis on the DDA Method 46
6.3.2 Analysis on the Identified Anomalies 50

7 Discussion **57**
7.1 Threats to Validity, Limitations and Future Work 58
7.2 Ethical Considerations 59

8 Conclusion **61**

Bibliography **I**

A Appendix **I**
A.1 Data Collection Interview Guide Sample I
A.2 Results of Thematic Analysis III
A.3 Code for data preparation - step 1 VI
A.4 Load the extracted data into data frames - step 2 VI
A.5 Predefined threshold check - step 3 VII
A.6 Configuration of IF model - step 4 VII
A.7 Code for using auto-calculated contamination to validate threshold value for Signal_1 - step 5 VIII
A.8 Code for retrain the final model to classifies each data points - step 6 IX
A.9 Evaluation Interview Guide Sample X

List of Figures

2.1 Data flow from Engine sensors data to Databricks 6

4.1 Overview of process involved in each phase of research study 15
4.2 Thematic diagram example for qualitative analysis study for Practitioners challenges (Naeem and Ozuem, 2022) 20
4.3 Thematic diagram example for qualitative analysis study for document analysis (Naeem and Ozuem, 2022) 22

5.1 Method for domain driven Anomaly detection (Artifact) 29
5.2 DDA Method Activity Diagram 30

6.1 Yes branch time series graph for Signal_1 with known threshold . . 42
6.2 No branch time series graph for Signal_3 with unknown threshold . 43
6.3 Time context analysis for Feb 3rd to Feb 6th dates 43
6.4 Time context analysis for Feb 5th day 2026 44
6.5 Multi signal correlation for Signal_1 with Signal_5 and Signal_6 . . 45
6.6 Anomalies confirmed during engine startup period for Signal_1 . . . 51
6.7 Anomalies confirmed during engine shutdown period for Signal_1 . . 52
6.8 Anomalies non-confirmed during engine startup period for Signal_1 . 53
6.9 Anomalies non-confirmed during engine shutdown period for Signal_1 53
6.10 Anomalies confirmed during engine startup period for Signal_3 . . . 55

List of Tables

4.1 Datasets used for quantitative evaluation and their corresponding threshold methods 27

5.1 Characteristics required for domain-driven anomaly detection 33

6.1 Data Requirements derived for anomaly detection 41
6.2 Category based on the interview questions 46
6.3 Summary of domain expert perceptions comparing the traditional method and the DDA Method 50
6.4 Quantitative evaluation of the identified anomalies using expert confirmation 51

1

Introduction

1.1 Problem Description

The increased use of modern engine systems has generated massive volumes of sensor data and has made these embedded systems highly software intensive rather than hardware based [18]. The development of these embedded systems relies heavily on software engineering practices to manage the complexity and size of the software used [18]. The integration of advanced development methods has created new challenges in ensuring reliable anomaly detection from sensor data, making it an important area of research within the broader field of software engineering.

Real world engine systems depend on the suitability and reliability of the data being used. Incomplete, inconsistent, or unreliable data can lead to undetected anomalies or generate unnecessary false alarms. Furthermore, data management remains a challenge, as most Machine Learning (ML) systems rely on well defined and trustworthy data [22]. It has been observed that poorly structured data requirements highlight the need for explicit data specifications, as outcomes are highly dependent on how collected data is defined and interpreted [24]. The case company Volvo Penta supplies engines for marine and industrial applications, where a key challenge is distinguishing meaningful anomalies from false alarms caused by sensor errors or other data related issues.

In Requirements Engineering (RE), domain knowledge has traditionally formed the foundation for requirements elicitation. It is an important asset for interpreting data, defining constraints, and ensuring that requirements accurately reflect real world operating conditions [1]. In addition, engine data alone is often insufficient for identifying meaningful anomalies. Since signal behavior can be highly context dependent, domain knowledge becomes essential for understanding normal operating conditions and defining relevant characteristics of the data [1]. This study therefore emphasizes the role of domain expertise in supporting clearer requirement definitions, particularly in domains that rely heavily on engine sensor data.

Understanding the specific data requirements necessary to characterize system behavior must be guided by domain experts. Such requirements help clarify what characteristics and constraints are needed for the system to distinguish between normal and abnormal behavior [29]. However, there is limited guidance on how

domain knowledge can be systematically translated into data requirements that support anomaly detection. Therefore, this thesis focuses on designing a method for anomaly detection in engine datasets by deriving domain knowledge and transforming it into data requirements that describe expected system behavior. The study investigates the application of Requirements Engineering to derive these data requirements from multiple sources, including interviews, document analysis, and data loggers, and incorporates them into a framework for anomaly detection.

Furthermore, not all anomalies necessarily indicate poor or invalid data. Anomalies are deviations that may arise due to errors, inconsistencies, unexpected events, or shifts in data distributions [24]. For this reason, defining data requirements is essential to ensure that collected and processed data is interpreted within the appropriate domain context. An example of an anomaly in an engine dataset is an engine temperature signal that normally operates between 0 and 120 degrees Celsius but suddenly rises to 125 degrees Celsius for a short period before returning to its normal range. Such deviations may indicate a sensor malfunction, an unusual operating condition, or another event that requires further investigation depending on the surrounding context.

Based on these observations, the objective of this study is to design and evaluate a domain driven method for anomaly detection in engine datasets, where data requirements derived from domain knowledge are used as input to guide the detection of anomalies. Addressing this challenge is increasingly important for Requirements Engineers and Analytics System Architects who seek to develop data driven solutions for anomaly detection. By establishing domain aligned data requirements as a foundation for the proposed method, the study aims to support more reliable identification and interpretation of anomalies in engine datasets.

1.2 Purpose of the Study

The purpose of this study is to design and evaluate a method for anomaly detection in engine datasets based on data requirements derived from domain knowledge. An artifact is designed and implemented as a process oriented framework that demonstrates how domain knowledge and data requirements can support anomaly detection in sensor and engine datasets. The study also focuses on identifying important characteristics of the data, such as sudden spikes, extreme deviations from the normal range, data error outliers, and in some cases clustering at specific time intervals, that are needed to recognize those patterns as anomalies.

The Design Science Research (DSR) methodology is used as part of designing the software engineering process. The research questions are formulated to examine the challenges faced by practitioners and the data requirements that are necessary for anomaly detection. Finally, to evaluate the proposed method, different evaluation methods are used in an industrial setting together with sensor data to ensure that the artifact effectively supports anomaly detection. By examining what data characteristics are essential and how domain knowledge can be translated into data

requirements, the results are expected to support Requirements Engineers, Data Engineers, and Analytics System Architects in defining and managing data requirements for a more systematic and domain informed anomaly detection process in engine datasets.

1.3 Significance of the Study

In general, the study focuses on addressing a key research gap in anomaly detection by designing a domain driven method for anomaly detection in engine datasets. The proposed method utilizes data requirements elicited from domain knowledge as a foundation for identifying and interpreting anomalies in sensor and engine data. Data requirements in the context of this thesis are defined as the specifications needed to support the development of the anomaly detection method, including domain constraints such as threshold values and other domain expert defined conditions. From a practical point of view, the domain driven anomaly detection method helps practitioners systematically identify and interpret anomalies in sensor and engine data by incorporating domain specific knowledge into the detection process. From a research perspective, the study contributes to the integration of Requirements Engineering and anomaly detection by incorporating domain related data requirements into the design of the artifact. This also paves the way for future research on domain driven anomaly detection methods and the use of data requirements in data intensive systems.

1.4 Research Questions

RQ1: What data requirements are needed to support anomaly detection across systems?

RQ1.1: What challenges do data practitioners face in managing and analyzing sensor data to identify anomalies in systems?

RQ1.2: What data requirements for anomaly detection can be identified through document analysis?

RQ2: What characteristics and functionalities are required for a domain driven anomaly detection method to detect anomalies and provide contextual insights?

RQ3: How effective is the domain driven anomaly detection method in identifying anomalies based on data requirements?

RQ3.1: How do domain experts perceive the domain driven anomaly detection method with the traditional method ?

RQ3.2: Does incorporating data requirements confirm the quality of the anomalies identified by domain driven anomaly detection method?

1.5 Thesis Outline

The thesis report is organized into several chapters starting with the Introduction chapter. Each chapter focuses on different aspects of the research.

- Chapter 2: Background - This chapter introduces data requirements and their role in sensor-based datasets. It then discusses different types of anomalies and anomaly detection in sensor datasets. Finally, it highlights the challenges in defining data requirements and the need for a structured approach to support anomaly detection and relevant data characteristics.

- Chapter 3: Related Work - This chapter reviews existing research on anomaly detection methods. It identifies the research gaps in existing statistical, machine learning based and domain driven approaches.

- Chapter 4: Methods - This chapter describes the Design Science Research (DSR) methodology used in the study. It explains each research phase, including data collection and data analysis methods used to address the research questions.

- Chapter 5: Results - This chapter presents the results of all research questions collected through different research methods. It includes the identification of key data requirements derived from interviews and document analysis. It also presents the implementation of the domain driven anomaly detection method. Finally, it reports the evaluation results of the proposed method using domain expert feedback and quantitative analysis.

- Chapter 6: Discussion - This chapter discusses the results and provides suggestions for future work. It also addresses the limitations of the study and threats to validity.

- Chapter 7: Conclusion - This chapter provides the conclusion and a summary of the overall thesis.

2

Background

The goal of this chapter is to describe the background of the key concepts and methods used throughout the thesis. This chapter introduces the core concepts related to data requirements and anomaly detection in the context of Requirements Engineering (RE). It explains how multiple sources, such as interviews, technical files and documentation are used to derive data requirements for identifying anomalies in the data. In addition, it introduces the methods and techniques that support the implementation of the proposed anomaly detection approach.

2.1 Data Requirements

In modern machine learning, data has become an important asset. The quality of data needs to meet specific requirements to ensure the functional requirements of a ML system. Similarly, data requirements specify that necessary data conditions are needed, how data are collected and used [32]. This is especially important in the automotive and embedded domains, where a large amount of data is generated from engine sensors during field tests. Engineers must know what data are needed for testing, debugging, validation and analytics. Furthermore, structured specification templates must be followed when deriving data requirements [7]. Hence, understanding the data requirements reduces the time spent on unwanted data and improves system reliability [14]. Thus, RE process focuses on deriving data requirements by collecting information of data sources and domain knowledge through the interviews [19].

2.1.1 Understanding data requirements and their sources

In the case of company of field test data, data requirements are derived to ensure that the information collected is essential for data pipeline, evaluation, and decision making. In this study, a set of system functional and data requirements was identified to ensure reliable data analysis of engine signals. These requirements were derived through a combination of stakeholder interviews and document analysis, including Database CAN (Controlled Area Network) files, real values of signals with their position, size, format, scaling and limits. The data requirements defined criteria for engine signal selection, threshold values and conditions necessary for outlier detection. Thus, they provide an accurate analysis of engine behavior during the

normal operation of the engine.

2.1.2 Data gathering process for data requirements

The data is collected from the embedded software system of the case company, where sensors signals are recorded during field test operations and stored for data analysis.

- **Engine Sensors and Data Logger** : In automotive embedded systems, the engine operation involves several processes, including air intake, fuel injection, combustion and exhausts, all of which must be continuously monitored and regulated to ensure smooth and reliable operation. To achieve this, these engines are equipped with numerous sensors that record key parameters such as temperature, pressure, speed and airflow. These signal values were sent to the data logger during field test operation, as shown in **Figure** 2.1. This data logger automatically records and stores the sensor data with timestamps under different conditions in the system

- **Technical files and Documentation** : The technical documentation defines data elements, frequency, and threshold values. These sources are important for data requirements and transmitted signal messages are stored in .dbc files which are referenced to derive data requirements [19].

- **Data collection** : Data collected from interviews with Data engineers, Data analytics, Data scientists, Data specialists, and Field test engineers provided a broader perspective on data generation [13]. They provided information on engine expectations, field conditions, and some example scenarios for sensor values 4.1.

- **Databricks** : Databricks is a cloud based tool used for data analytics platform built on Apache spark that enables large scale data processing, analysis and machine learning models to find patterns and make decision from the datasets. It provides collaborative notebooks, scalable computing and tools for data engineering, data science and data analytics. These tool is used to clean, process and analyze large volume of datasets to detect maximum threshold values, handle missing values and evaluate data quality evaluate data quality using data pipelines.

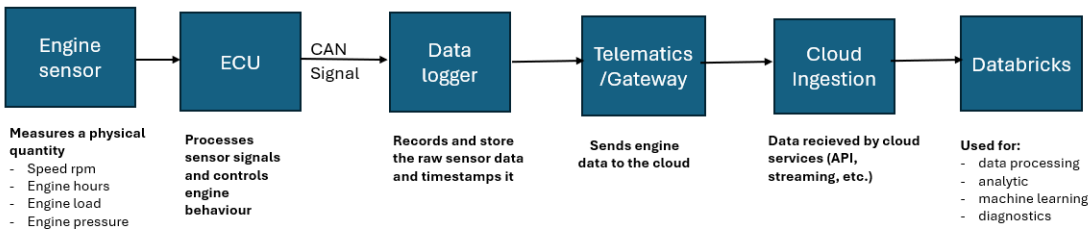

Figure 2.1: Data flow from Engine sensors data to Databricks

2.1.3 Challenges associated with data requirements

In the context of case company, the key challenges associated with data requirements are collecting data from multiple sources and defining data requirements. Also, field test data contain a large volume of information and are hard to analyze due to data with multiple signals from different engines. Furthermore, linking all requirements back to sources such as interviews, database CAN files, mapping, and prioritization would make be more challenging to obtaining data requirements more challenging. Additionally, deriving data requirements requires studying field test data and understanding what data are useful, what is missing, what exceeds the maximum value range, and the unrealistic behaviors of engine resets. Hence, converting these findings into formal requirements is useful in detecting anomalies. The results obtained are used to validate the accuracy of the anomaly[8].

2.2 Anomalies

Anomalies refer to the data pattern or observation that deviates from expected or normal system behavior [4]. In engine field testing, large amounts of data are collected from sensors to measure temperature, speed, pressure, and fuel consumption. During normal operations, these values follow regular patterns. Under certain conditions, when a sensor reading or group of readings deviates from this pattern, it leads to anomaly. Such anomalies may indicate engine faults, sensor failures, delay in timestamp or human errors.

2.2.1 Types of anomalies

Anomalies are classified into three forms that are studied in the literature, such as point anomalies, contextual anomalies and collective anomalies.

- **Point anomalies** : When a single sensor value deviates from the normal pattern considered, such as a sudden increase or decrease in the temperature value. This is the simplest form of anomalies and is most observed by the researcher [4].

- **Contextual anomalies** : If a point or data instance is anomalous in a specific context, it is considered as an anomaly, but not otherwise, it is referred to as contextual anomaly which is also known as conditional anomaly [4]. For example, high engine speed may be considered normal if it has gradually increased over time, where as a sudden spike from low engine speed to high could be considered an anomaly [4].

- **Collective anomalies** : If a sequence or collection of points is abnormal with respect to the rest of the data points [4]. For example, when the engine is experience heavy loads, high temperatures of various kinds are expected. A very low temperature in this scenario could be considered an outlier [4].

2.2.2 Types of anomaly detection technique

Based on the literature study [4], the choice of anomaly detection technique depends on the availability and type of data labels, particularly labels indicating anomalies. Methods are categorized based on whether the data is labeled or unlabeled. This distinction is crucial for selecting appropriate machine learning approaches for anomaly detection. Based on the type of data available, anomaly detection techniques can be divided into three different types.

- **Supervised learning** : When applying supervised mode, the training dataset uses labeled instances for an algorithm to predict model for normal vs anomaly classes. The major challenge would be obtaining accurate labels to represent the anomaly It is normally hard to include all types of anomalies which is needed for algorithm to perform [4].

- **Semi supervised learning** : When applying semi supervised mode training datasets use labeled instances for only the normal class. This lets the model to learn only what are normal points. Such techniques are not commonly used, it is harder to cover every possible anomalous that can occur in the data points [4].

- **Unsupervised learning** : In the unsupervised mode, no labeled instances are required. The model analyzes and clusters unlabeled data without human involvement or predefined tags. The algorithm discovers hidden patterns and natural gaps in the raw dataset. This technique is the most widely applicable [4].

In conclusion, the choice of a learning technique depends on the nature of the available data. In this thesis, an unsupervised learning approach was used because the dataset is unlabeled. Unsupervised learning analyzes the data to identify observations that differ significantly from the overall data distribution. These observations can then be isolated and investigated as potential anomalies.

2.2.3 Anomaly detection problems and challenges

Several studies in the literature discussed key challenges in anomaly detection, particularly related to data quality and data processing pipelines. Sensor data collected from multiple sources often contain noise, inconsistencies, and incomplete information, which directly impact the reliability of anomaly detection systems [3]. In addition, issues introduced during data ingestion and integration further degrade data quality, making it difficult to ensure consistency across datasets [11]. Since machine learning anomaly detection methods depend on the data, these challenges significantly affect model performance.

Some common data related problems in anomaly detection include:

- **Duplicate data**: Repeated records can bias the dataset, leading to misleading patterns and increased computational overhead [3].

- **Data Error Outlier values**: Extreme values may result from sensor errors or unexpected system behavior, making it difficult to distinguish between true anomalies and noise without domain knowledge [11].

- **Data latency**: Temporal delays and recurring latency patterns can influence anomaly detection, particularly in time-series data where anomalies may occur during specific operational conditions [21].

Furthermore, distinguishing true anomalies from normal signal variations remains a significant challenge, particularly in time-series engine data where signal spikes may occur due to operational transitions rather than actual faults. Without a reference threshold grounded in system specifications, models risk producing false positives or missing critical anomalies.

This chapter has discussed the role of data requirements and anomaly detection techniques in the context of field test datasets. It highlights how multiple data sources and collection methods contribute to understanding system behavior while also introducing challenges related to inconsistencies and anomalies. To address these issues, the next chapter presents the research methodology discusses approaches for anomaly detection .

3

Related Work

Detecting anomalies using data requirements derived from the domain knowledge of experts has emerged as an important research area, as existing studies have primarily focused on statistical, machine learning, and hybrid approaches for anomaly detection [4]. This challenge becomes particularly evident in sensor based datasets, where the interpretation of anomalies frequently depends on domain expertise and application specific constraints. Therefore, this chapter reviews existing anomaly detection approaches highlighting their strengths, limitations, and the research gaps that motivate the contribution of this thesis.

3.1 Statistical and Machine Learning Approaches for Anomaly Detection

Existing studies explore anomaly detection approaches from both statistical and machine learning perspectives [2]. These approaches are used to identify outliers and abnormal patterns in datasets, which often indicate unusual or unexpected behavior. Together this helps in understanding the strengths and limitations of different anomaly detection techniques. Existing research also highlights that outliers and anomalies are among the most recurring problems in datasets. However, the techniques used to identify them do not incorporate explicit domain-specific reasoning.

Existing studies have proposed various statistical and machine learning methods for anomaly detection [2]. These methods are used to identify outliers and abnormal patterns in datasets that may indicate unusual or unexpected behavior. Although they have proven effective in many applications, their performance depends on the characteristics of the data and they often lack domain-specific contextual understanding.

Statistical anomaly detection techniques are among the most widely used approaches for identifying abnormal behavior in datasets. Statistical instances occurring in low-probability regions are typically classified as anomalies because they deviate from the statistical properties and probability distributions of the data [5]. Statistical methods can be categorized into parametric and non-parametric techniques, where parametric methods assume knowledge of the underlying data distribution while non-parametric methods do not [5]. The Interquartile Range (IQR) method is one of

the most widely used non-parametric approaches for outlier detection [28]. However, traditional statistical methods often rely on measures such as the mean and standard deviation and are generally more suitable for uni variate data [5].

To address the challenges of large and complex datasets, machine learning methods have gained increasing attention. Different approaches, including supervised, unsupervised, and semi-supervised methods, can be applied depending on the structure of the dataset. In particular, unsupervised anomaly detection methods are commonly used to identify unusual patterns in time-series and high-dimensional data [10]. The Isolation Forest algorithm is a widely adopted unsupervised machine learning technique that detects anomalies using random decision trees and identifies observations with shorter path lengths as anomalous [20]. Its effectiveness in handling large-scale and high-dimensional datasets has made it a popular choice for anomaly detection.

Despite the effectiveness of both statistical and machine learning methods, several limitations remain. Traditional statistical techniques, including IQR-based methods, may struggle with high-dimensional datasets, while machine learning approaches primarily focus on algorithmic performance and detection accuracy [5]. Furthermore, existing anomaly detection methods such as [12, 27] largely rely on different algorithmic combinations and data-driven mechanisms without incorporating explicit domain knowledge from experts. As a result, detected anomalies may not always align with domain-specific expectations and requirements. Therefore, there is a need for well-structured domain-driven anomaly detection methods that integrate expert knowledge and derive domain requirements to provide more interpretable, adaptable, and context-aware anomaly detection solutions.

3.2 Hybrid Approaches for Anomaly Detection

Recent studies have explored hybrid and multi stage approaches to improve anomaly detection performance in complex datasets. Ghrib et al. proposed a hybrid anomaly detection approach for time series data that combines multiple detection techniques to leverage their complementary strengths and improve detection robustness [12]. Similarly, Sperl and Chung introduced a two-step anomaly detection framework in which anomaly detection is performed through sequential stages, allowing simple methods to initially identify potential anomalies before applying more advanced techniques for further analysis [27]. The authors argue that a single detection phase is often insufficient for practical time series datasets and that a multi-stage approach can improve anomaly identification accuracy.

Despite their advantages, hybrid and two-step approaches introduce challenges such as increased system complexity, integration difficulties, and the potential propagation of errors between detection stages. More importantly, these approaches primarily focus on improving detection performance through algorithmic combinations and data-driven processing. They provide limited support for incorporating explicit domain knowledge and expert derived requirements into the anomaly detection process. Therefore, there remains a need for domain driven anomaly detection methods

that systematically integrate expert knowledge and domain specific requirements to provide more interpretable, adaptable, and context aware anomaly detection solutions.

3.3 Domain Driven Anomaly Detection

Domain driven anomaly detection extends traditional anomaly detection by incorporating domain knowledge and contextual information into the detection process. These approaches are often categorized as knowledge based anomaly detection where expert knowledge, business rules and constraints are used to identify anomalies that cannot be solely detected through statistical and machine learning techniques. However [16] also identifies several challenges, including the lack of structured frameworks and steps for effectively translating domain expertise into operational anomaly detection systems, as well as limitations in scalability and practical adoption.

Although existing anomaly detection approaches have made significant progress, they still exhibit limitations such as computational complexity, sensitivity to parameter tuning, and dependence on feature selection. Furthermore, these approaches are predominantly driven by statistical and algorithmic mechanisms rather than domain specific reasoning. Previous studies [5] acknowledge that anomaly detection is inherently domain dependent. However, they do not provide systematic methods for incorporating domain knowledge within anomaly detection frameworks. Consequently, the contextual interpretation of anomalies remains limited, as most existing methods focus on statistical deviations or learned data patterns rather than domain specific requirements.

Therefore, there remains a need for domain driven anomaly detection approaches that systematically derive requirements from domain experts and transform them into practical anomaly detection mechanisms. To address this gap, this thesis proposes a structured domain driven anomaly detection method that integrates expert knowledge and domain specific requirements to identify context aware anomalies in a more interpretable and adaptable manner.

4

Methods

This thesis will follow Design Science Research (DSR) which adheres to the the engineering phases of Wieringa [30] and Knauss guidelines [17]. From two guideline studies, we have adopted three phases including problem investigation, design implementation and evaluation. Finally the results are directly evaluated for more accurate analysis.

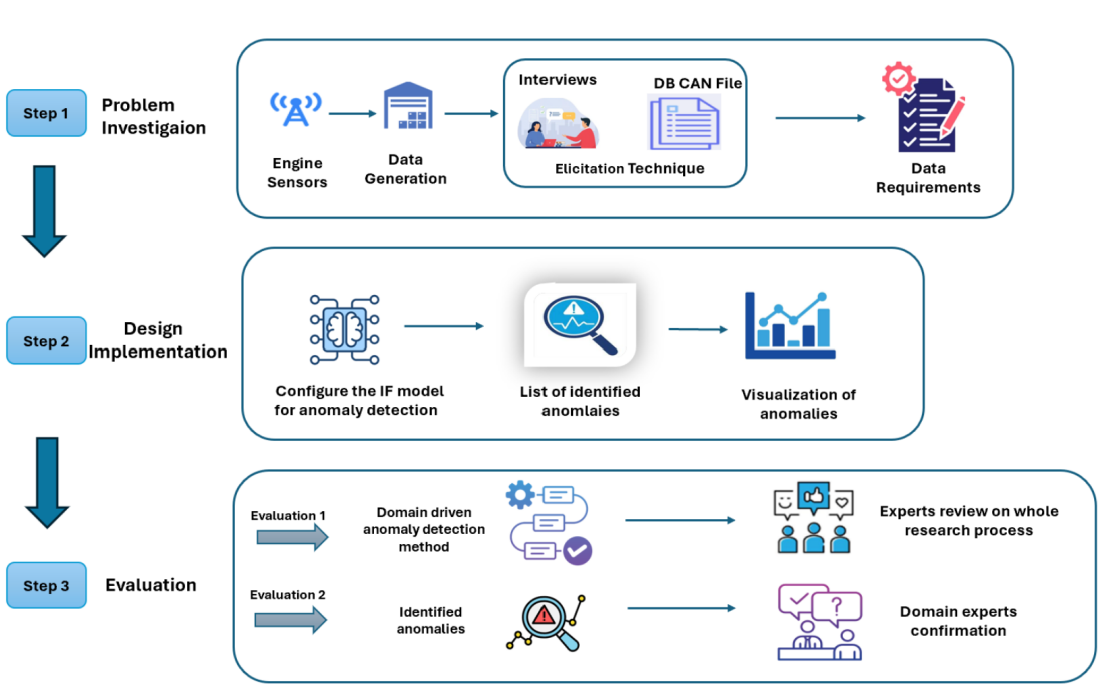

Figure 4.1: Overview of process involved in each phase of research study

The problem investigation phase identified requirements through interviews, document analysis, thematic analysis and domain knowledge. Based on this, a solution was developed and implemented as domain driven method for anomaly detection using machine learning technique. The method is evaluated through expert reviews and quantitative measures Thus, it converts engine domain knowledge to data requirements for anomaly detection. This research methodology is appropriate because it focuses on an innovative artifact.

The goal of this thesis is guided by the Research questions (RQs) mentioned in Section 1.4. To address each research question, the thesis focuses on both qualitative and quantitative analysis [6][2]. The RQ1 identifies challenges faced by data practitioners in understanding the engine datasets and deriving the requirements to design the method for identifying anomalies [7]. The RQ2 focuses on the implementation of the domain driven anomaly detection method. The RQ3 focuses on the evaluation of the domain driven anomaly detection methods by performing both qualitative analysis through domain experts and quantitative analysis through evaluation metrics. All defined methods and procedures of this study are explicitly linked to each of the Research questions (RQs) mentioned above. Figure 4.1 presents a high level overview of the research steps involved in each phase of our research study.

4.1 Problem Investigation

In the first phase, we focused on identifying the main problems and causes of anomaly detection challenges in engine datasets. We collaborated with the Data science and Engineering team of the case company who is focused at data driven actionable insights. For each data source (interview, documentation, etc.), problem investigation was carried out through two phases: data collection and thematic analysis.

4.1.1 Interviews

Stakeholders including domain experts, data engineers, data scientists, data analysts, and field test engineers, were identified and invited to participate in the interviews as part of the data collection process. The participants were selected based on their direct involvement and experience with field test data and data pipelines within the case company. A purposive sampling approach was used, where individuals were intentionally chosen due to their relevant expertise and knowledge of the problem domain. A total of eight professionals took part in the interviews, contributing both technical and operational insights related to field test data and data pipelines.

Data Collection

During this interview process, the researcher asked a series of questions about the subject of the case study. Each interview lasted approximately 45 to 60 minutes and was conducted in a conversational manner to encourage the interviewees. Before the actual interviews, a pilot interview was conducted to test the clarity, relevance of the questions and major adjustments were made accordingly. Ethical considerations were carefully followed by informing the participants about the purpose of the study and maintaining confidentiality of their responses. Consent was approved from all participants before conducting interviews. The use of interviews follows a qualitative case study approach suggested by [26].

An interview guide categorized with various questions and sections are formulated.

- **Orientation :** The first section, majorly focuses on the background of the

interviewee and general overview about the datasets.

- **Information Gathering :** The next section involves specific questions and what challenges practitioners face when managing the datasets and analyzing sensor data to identify anomalies in systems (RQ1.1).

- **Closing :** The final section majorly focuses on questions about the field test dataset that is involved in identifying documents and data patterns for anomaly detection (RQ1.2).

The data for this study were collected through semi-structured and structured interviews which allowed the researcher to gather both consistent and detailed information from the participants. In a semi-structured interview, questions are planned, but they are not necessarily asked to question in the same order as listed and allow the researcher to better understand the perspective of the interviewees. And in many ways a structured interview is similar to the question based survey [26]. The structured questions ensured that all participants were asked similar pattern questions, which helped to maintain consistency in the data and allowed the participant to explain more in detail as well as recorded the responses.

A structured interview guide was developed to ensure the data collection process to align with research objectives. The interview guide is included in Appendix A.1.

The interview questions focused on identifying practitioner challenges and deriving data requirements for anomaly detection. Sample list of questions included are:

- Can you explain briefly your role and experience with engine datasets ?

- What current challenges are encountered when we deal with anomalies in datasets ?

- Are there any specific conditions under which temperature sensor values deviate from the expected range ?

- Are there any tools, rules, or techniques you commonly use for data validation for field test datasets ?

- Did you come across some anomalies that occur only under specific conditions ? (e.g., load, temperature, environment)

- Are there documents or examples you recommend reviewing ?

- If available, how do you think they are helpful in understanding the existing system and operational guidelines ?

These questions allow a broad range of responses and answers from the interviewee to the subject [25]. Most of the interview participants were open and willing to share their opinions. In some cases, researcher asked additional questions to better understand their answers. The overall interview process worked well and provided valuable

information that helped the researcher better understand the research problems.

Therefore, this specific phase will focus on challenges involved in the data flow in real engines, how the data is transmitted and stored. This supports **RQ1.1** for investigating the problems faced by the case company in identifying anomalies.

Thematic Analysis

The qualitative interview data in this study were analyzed using thematic analysis following the step by step approach described by Wagas Neem [23].

- First step, the interview recordings were transcribed and carefully read several times to become familiar with the data.
- **Important statements** and **quotations** were identified during this stage.
- Next, recurring words were selected as **keywords** as highlighted in yellow and used to create **codes** that represent meaningful parts of the data. These **codes** were then grouped together to form **sub themes** that describe common pattern in the interviews.
- Finally, derived **themes** are mapped with respective all the **RQ1's** as shown in the Chapter 6 to understand the main concepts related to the research problem and insights support the data requirements needed for anomaly detection methods.

These analysis were carried out in a step by step process as described in the example using Thematic analysis for anomaly detection is shown in **Figure** 4.2 and derived data requirements.

Step 1 : Familiarization with the Data

Important statements and repeated ideas were noted from the interviews. Statements such as "Most common fault pattern observed is the maximum threshold exceeding the limit", "We just don't remove the bad data, we flag it", "New engine signals threshold were unknown", "We have found some null-values", "I don't want duplicate values for each timestamp", and "It's easier to fix something if you know the context" were identified as key points related to anomaly detection and data quality.

Step 2 : Identifying Keywords

From the interview data, keywords such as: "maximum threshold", "flag data", "unknown threshold", "null values", "duplicates", and "context" were identified.

Step 3 : Coding the data

Keywords were grouped into meaningful codes that describe the main issue.

- **"maximum threshold" → Data anomaly**
- **"flag data" → Outlier classification**
- **"unknown threshold" → Threshold uncertainty**
- **"null values" → Missing data issue**
- **"duplicates" → Data duplication issue**
- **"context" → Lack of contextual information**

Step 4 : Developing Sub-themes

Similar codes were grouped together to form broader sub-themes.

- **Data anomaly → Data Anomaly**
- **Outlier classification → Classification of Outliers**
- **Threshold uncertainty → Anomaly Detection**
- **Missing data issue → Valid Dataset**
- **Data duplication issue → Valid Dataset**
- **Lack of contextual information → Context Analysis**

Step 5: Generating Themes

Sub-themes were further grouped into main themes representing key findings.

- **Data Anomaly → Practitioners' Challenges**
- **Classification of Outliers → Practitioners' Challenges**
- **Anomaly Detection Challenges → Practitioners' Challenges**
- **Valid Dataset → Practitioners' Challenges**
- **Context Analysis → Practitioners' Challenges**

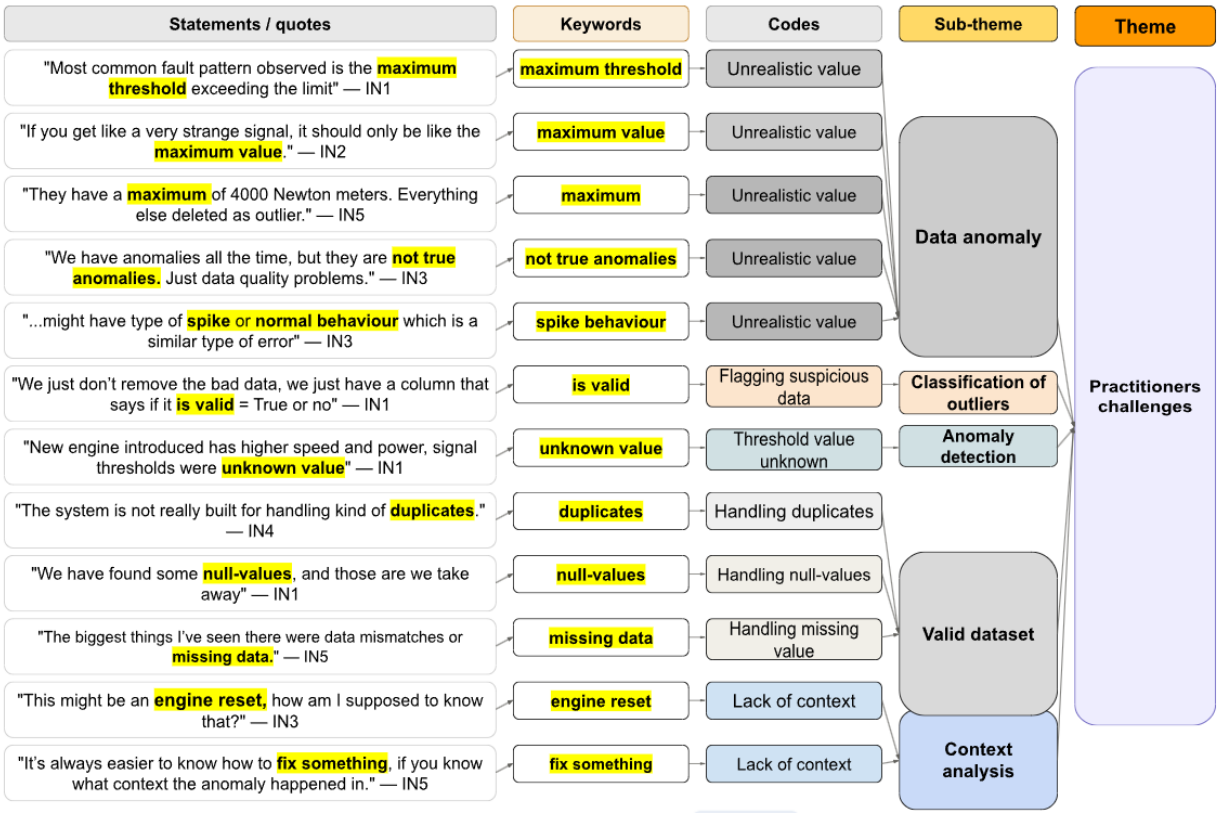

Figure 4.2: Thematic diagram example for qualitative analysis study for Practitioners challenges (Naeem and Ozuem, 2022)

Finally, the themes were analyzed and interpreted to understand the overall research problem and answer the research question. The theme Practitioners' Challenges highlights issues such as data anomaly and classification of outliers, anomaly detection, valid datasets and conext analysis. These findings helped identify key data requirements and challenges in managing field test data. The same procedure was applied to all remaining statements and quotations to maintain consistency in the thematic analysis. In addition, Documentation and Data scope and Ground Truth Validation also emerged as final themes in the analysis. The Documentation and Data scope theme includes sub-themes such as Document Analysis and Scope of datasets Document Analysis, which highlight the importance of proper documentation. Similarly, Ground Truth Validation theme focuses on evaluation, emphasizing the accuracy of the anomalies. These themes further support the understanding of the challenges in distinguishing true anomalies from data faults, and contribute to defining the necessary data requirements for the study. Detailed mappings of keywords, codes, sub themes, and themes are included in the (Appendix A.2) for additional information.

4.1.2 Document Analysis

In particular, the documentation addressing **RQ1.2** contains the Database CAN (DBC) file for different engines, such as combustion engines and diesel vehicles,

etc. Depending on the specific engine for the logger, accurate DBC files need to be selected based on the minimum and maximum values for specific signal names. These DBC files define numerous signals representing various sensors and parameters.

Data Collection

In addition to the interviews, document analysis was conducted to understand the data structure, management practices, and validation procedures in field test data. This study examined the internal documentation and reference files, including DBC files, which are used to define the engine sensors and their parameters. During the interviews, the field testing team requested to refer to DBC files and apply these as reference files. This combined approach ensured that both the formal documentation and practical usage were captured, providing a comprehensive view of handling document processes.

Thematic Analysis

Thematic analysis was applied to both the interview data and the collected documents, including reference files such as DBC files. The process followed a structured approach to ensure consistency across different data sources. First, the interview transcripts and document contents were reviewed to become familiar with the data.

- Important statements, definitions, and references (e.g., signal descriptions in DBC files) were identified.
- Next, relevant keywords were extracted from both the interviews and documents, such as “DBC file”, “multiple DBC files”, “dbc values”, “restricted”, “updated rarely”, "most commonly", "most important".
- These keywords were then assigned codes representing key concepts such as “DBC file management,” “reference files,” and “ prioritizing signals”.
- These codes were further grouped into sub-themes, including Document Analysis and Scope of Datasets, which capture issues related to documentation quality and dataset selection.
- Finally, these sub-themes were organized into a broader theme, Documents and Data Scope, highlighting the importance of well-defined documentation and controlled dataset scope in supporting consistent data interpretation and reliable anomaly detection.

This combined analysis helped link practical insights from interviews with formal definitions from documents. The overall thematic analysis process applied in this study is illustrated in **Figure** 4.3, which shows the step-by-step flow from interviews to final themes.

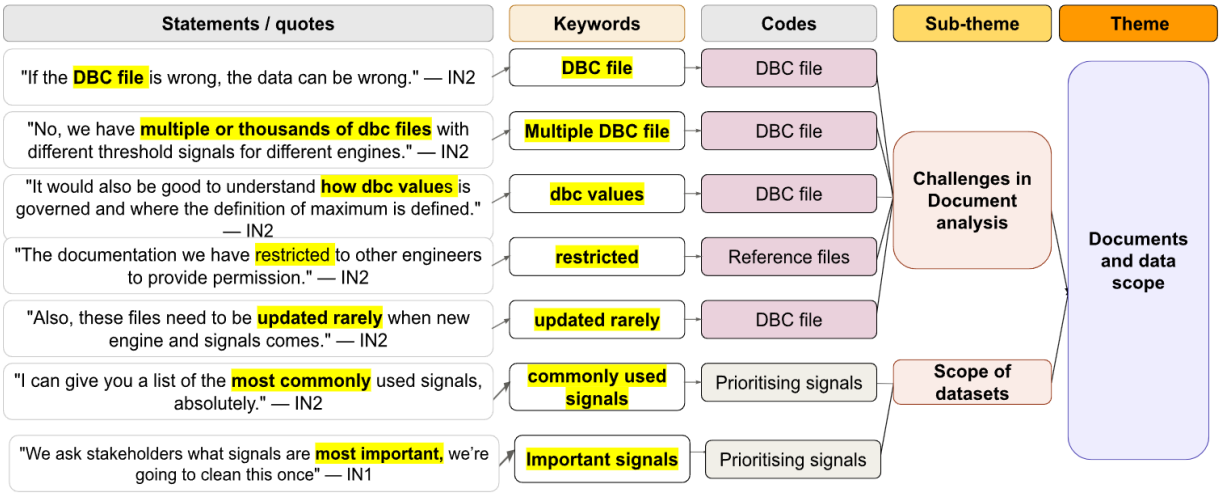

Figure 4.3: Thematic diagram example for qualitative analysis study for document analysis (Naeem and Ozuem, 2022)

The DBC file contains predefined maximum threshold values from the signal definitions. These threshold values were then applied as domain-specific for anomaly detection.

Below illustrates the syntax for the specific engine sensor information to read data from the DBC file for threshold values for each signal. For example, Signal_1 of VP_ENG have a valid threshold range from 0 to 500, as specified in the DBC file.

Syntax : SG_VP_ENG_Signal_1_Speed : 16|16@1+ (0.125,0) [0|500] "rpm" ECU2

Abbreviation :

- **SG_VPEngine_Signal_1_Speed → sensor name**
- **16|16 → start bit | length**
- **@1 → little-endian (@0 = big-endian)**
- **+ → unsigned (- = signed)**
- **(0.125,0) → scale, offset**
- **[0|500] → min/max**
- **"rpm" → unit**
- **ECU2 → receiver**

This supports **RQ1.2** for analyzing requirement documents which provides information for deriving data requirements.

4.1.3 Data Requirements Derivation

The derivation of data requirements in this study was guided by the multi-layered framework proposed in the literature [7]. The sample template from the frame-

work was adapted to align with the specific needs of this study. Once the data requirements were derived, data requirement validation was conducted through a qualitative expert review with domain experts from the case company. During this process, the derived data requirements were presented to the domain experts in a structured table format and discussed in a narrative and iterative manner. Each data requirement was assessed based on its relevance, completeness and correctness in relation to the operational context. Feedback was collected regarding some ambiguous definitions and any inconsistencies with real world context. This validation ensured that the data requirements reflects and supports the method for domain driven anomaly detection.

4.2 Design Implementation

This design implementation focuses on addressing **RQ2** which includes an artifact, i.e, a method for domain driven Anomaly detection as shown in **Figure** 5.1. The knowledge gathered during the problem investigation phase becomes the foundational requirement to implement and classify the anomalies and normal behavior in the dataset. A domain driven anomaly detection process will be implemented using unsupervised machine learning techniques to identify anomalies detailed in **Chapter** 5 [28][20][15].

4.3 Evaluation

The evaluation focuses on assessing the effectiveness of the proposed domain driven anomaly detection method in identifying anomalies based on data requirements, thereby addressing **RQ3**. First, the evaluation investigates how domain experts from the case company perceive the proposed domain driven anomaly detection method in comparison with the traditional anomaly detection method. Second, the anomalies identified by the domain driven anomaly detection method are evaluated using quantitative performance measures to assess the quality of the detected anomalies and determine the extent to which they are confirmed as true anomalies.

4.3.1 Expert Reviews

To perform a qualitative evaluation of the proposed domain driven anomaly detection method, expert reviews were conducted to assess how domain experts perceive the method and its effectiveness in identifying anomalies based on data requirements. The evaluation also examined the usability of the proposed method in comparison with the traditional anomaly detection approach. A total of four interviews were conducted with domain experts from the case company, representing various roles, including data engineers, field test engineers, and data analysts involved in the field test data pipelines.

Interview preparation

Each interview lasted between 45 and 60 minutes and was conducted using a semi-structured format. An interview guide consisting of predefined questions was prepared beforehand to ensure consistency across the interviews while allowing flexibility for follow up discussions. Prior to each interview, the participants were informed about the ethical considerations of the study, including the confidentiality of their responses. The interview guide was structured into the following sections and adapted based on the qualitative case study approach proposed by Baškarad [26].

- **Orientation**: The first section provided a high level overview of the proposed domain driven anomaly detection method, including its overall workflow and the role of data requirements in supporting anomaly identification.
- **Information Gathering**: The second section focused on gathering the participants perceptions of the proposed domain driven anomaly detection method, including its usability, applicability, and effectiveness in comparison with the traditional anomaly detection method.
- **Closing**: The final section invited participants to provide suggestions for improving the proposed method and discuss potential directions for future work.

The interview guide was structured to ensure that all aspects of the proposed domain driven anomaly detection method were systematically evaluated, as presented in Appendix A.9. The questions were designed to gather domain experts perceptions of the proposed method and its suitability for practical adoption in comparison with the traditional anomaly detection approach. Some of the questions included in the interview guide are as follows:

- How useful do you find the proposed domain driven anomaly detection method for identifying anomalies based on data requirements?
- How does the proposed domain driven anomaly detection method compare with the traditional anomaly detection method in terms of usability and applicability?
- Is the current model configuration of the proposed domain driven anomaly detection method suitable, or do you recommend any modifications?
- Can the proposed domain driven anomaly detection method be integrated into the existing data pipeline at the case company?
- What improvements or future enhancements would you suggest for the proposed domain driven anomaly detection method?

The interview questions were designed to gather detailed feedback on the usability, applicability, and perceived effectiveness of the proposed domain driven anomaly detection method. A semi-structured interview format with open ended questions was adopted to provide flexibility for follow up discussions and allow participants to elaborate on their experiences and opinions. This approach facilitated an in depth

qualitative evaluation of the method based on the interview categories presented in **Table** 6.2.

Narrative Analysis

Narrative analysis was applied to the interview data following the structured approach proposed by Wong and Breheny [31]. This approach was selected because the objective of the qualitative evaluation was to understand how each domain expert perceived the proposed domain driven anomaly detection method in comparison with the traditional anomaly detection method. Unlike thematic analysis, which focuses on identifying common themes across participants, narrative analysis preserves each participant's reasoning, experiences, and explanations as a coherent account. This was considered more appropriate for the study since the evaluation involved a small number of domain experts whose individual perspectives and practical experiences were valuable in assessing the proposed method.

The interview data were analyzed using the following steps:

- First, the interview transcripts were reviewed multiple times and organized according to the interview questions to gain familiarity with the collected data.
- Next, each participant's responses were interpreted as a coherent narrative, preserving the context and sequence of their explanations rather than fragmenting the data into independent themes.
- The individual narratives were then compared to identify similarities and differences in the experts perceptions regarding the usability, applicability, and effectiveness of the proposed domain driven anomaly detection method compared with the traditional anomaly detection method.
- Finally, the narratives were interpreted collectively to develop an overall understanding of the strengths, limitations, and potential improvements of the proposed method based on the perspectives of the domain experts.

The narrative analysis supports **RQ3.1** by providing an in depth understanding of how domain experts perceive the proposed domain driven anomaly detection method, including its usability, applicability, and suitability for integration into existing industrial workflows when compared with the traditional anomaly detection approach.

4.3.2 Quantitative Model Evaluation

This section presents the quantitative evaluation of the anomalies identified by the proposed domain driven anomaly detection method. The evaluation aims to assess the correctness of the detected anomalies by comparing them against the domain confirmed anomalies. To achieve this, quantitative performance metrics were used to measure how accurately the proposed method identifies true anomalies and dis-

tinguishes them from normal observations.

Evaluation Setup

The quantitative evaluation focused on assessing the quality of the anomalies identified by the proposed domain driven anomaly detection method. The identified anomalies were independently reviewed and confirmed by two domain experts from the case company to determine whether they represented true anomalies within the application domain. The evaluation was based on the experts assessment of each detected anomaly, which served as the reference for determining the validity of the identified anomalies.

The following prerequisites were required to perform the quantitative evaluation:

- **Detected anomalies**: The list of anomalies identified by the proposed domain driven anomaly detection method.
- **Expert validation**: Independent confirmation of each identified anomaly by two domain experts from the case company.

Evaluation Metric

The quantitative evaluation was based on the *validation rate*, which measures the proportion of anomalies identified by the proposed domain driven anomaly detection method that were confirmed as true anomalies by the domain experts. The validation rate was selected because the objective of the evaluation is to assess the quality of the detected anomalies through expert confirmation rather than traditional classification metrics.

The validation rate was calculated as:

$$\text{Validation Rate} = \frac{\text{Number of anomalies confirmed by domain experts}}{\text{Total number of anomalies identified by the proposed method}}$$

A higher validation rate indicates that a larger proportion of the anomalies identified by the proposed method were confirmed as true by the domain experts, demonstrating the effectiveness of the method in identifying meaningful anomalies based on data requirements.

Steps for domain expert confirmation

To quantitatively evaluate the proposed domain driven anomaly detection method, the anomalies identified by the method were confirmed by two domain experts from the case company. The process was performed by visually inspecting the detected anomalies within the corresponding time series graphs to determine whether each anomaly represented a true anomalous event based on domain knowledge. The following steps were performed.

- The proposed domain driven anomaly detection method was executed on the selected datasets to identify the list of detected anomalies.
- Each detected anomaly was visualized using its corresponding time series graph, highlighting the anomalous data point together with the surrounding sensor measurements to provide sufficient context.
- The time series graphs were independently reviewed by two domain experts, who determined whether each detected anomaly represented a true anomaly based on their domain expertise and understanding of the sensor behavior.
- The anomalies confirmed by the domain experts were used to calculate the validation rate, representing the proportion of detected anomalies that were considered valid.

Table 4.1 presents the datasets used for the quantitative evaluation of the proposed domain driven anomaly detection method. For each dataset, the table specifies the threshold method used for anomaly detection, either a predefined domain threshold or an automatically calculated threshold. The anomalies detected from these datasets were subsequently reviewed and validated by the domain experts to compute the validation rate. This evaluation supports **RQ3.2** by assessing the quality of the anomalies identified by the proposed domain driven anomaly detection method through expert confirmation.

S.No.	**Dataset name**	**Threshold method used**
1	Signal_1	Predefined threshold
2	Signal_3	Auto calculated threshold

Table 4.1: Datasets used for quantitative evaluation and their corresponding threshold methods

5

Artifact

This chapter presents the design and implementation of an artifact for anomaly detection using the Isolation Forest (IF) algorithm. The artifact takes raw engine sensor data and derived data requirements as inputs to identify abnormal patterns in the datasets. As a result, it classifies the detected observations, distinguishing relevant anomalies from data error outliers. In addition, contextual analysis, including time based pattern analysis and multi-signal correlation was performed to explain the causes of the detected anomalies.

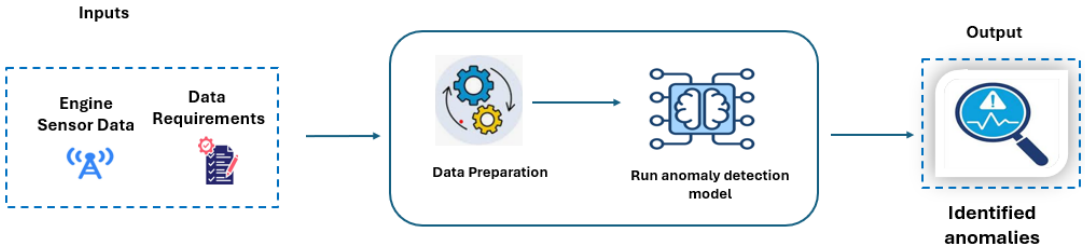

Figure 5.1: Method for domain driven Anomaly detection (Artifact)

5.1 Artifact: Method for Domain Driven Anomaly detection (DDA Method)

Based on the design and implementation, an artifact (DDA Method) is designed to automatically detect anomalies using the Isolation Forest (IF) algorithm. The purpose of this artifact is to demonstrate a structured, end-to-end approach for identifying unusual or irregular patterns within datasets. The artifact uses an unsupervised machine learning technique that can be applied to large volumes of sensor data without requiring labeled data. Instead, the model learns the structure of normal data and classifies data error outliers as anomalies. The IF algorithm was selected because of its computational efficiency, its ability to handle large datasets, and its capability to isolate and partition data points using a tree-based structure [9]. Moreover, the IF algorithm is flexible and can be applied to a wide range of data types, including time-series, continuous, and categorical data [9]. Thus, the artifact was designed to detect anomalies in unlabeled sensor data while minimizing false

positives. Thus, the Isolation Forest algorithm was adopted to provide adaptability and accurate anomaly detection without relying on fixed thresholds.

The artifact is structured into two stages, as shown in Figure 5.1. In the first step, engine sensor data and the data requirements derived from the results of **RQ1** are provided as inputs to the anomaly detection process. These data requirements were validated by the case company. The next step involves data preparation followed by the execution of the configured anomaly detection model using the IF algorithm to identify anomalies. Finally, the detected anomalies are produced as the output of the artifact. Figure 5.2 illustrates the detailed activity diagram of the proposed artifact, showing the workflow of the anomaly detection process. It expands upon the high level workflow presented in Figure 5.1 .

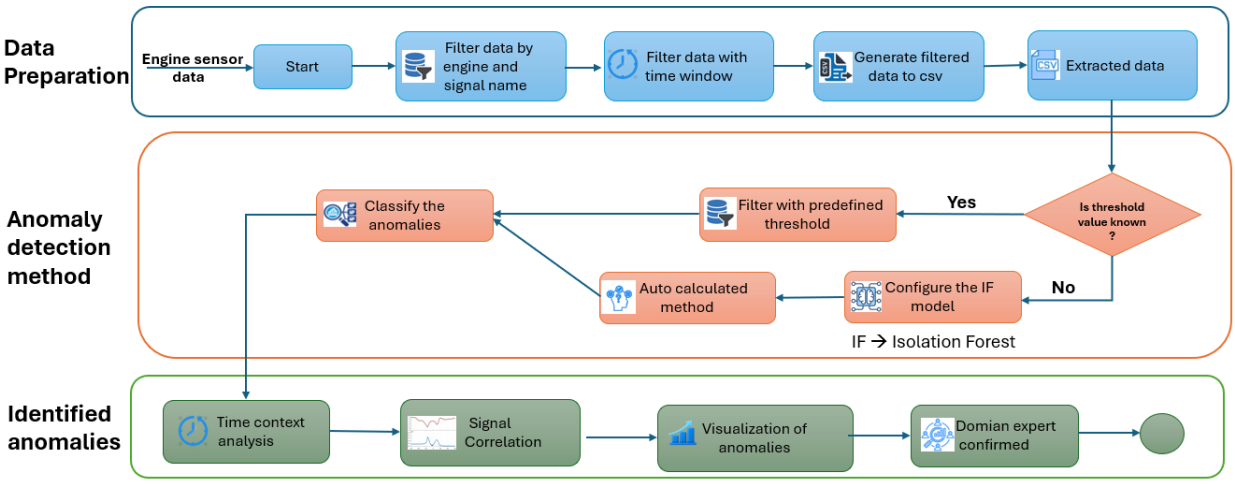

Figure 5.2: DDA Method Activity Diagram

5.1.1 Input for DDA Method

The input for this method is an engine sensor dataset. This involves by selecting relevant information, such as the engine name and signal name and filtering data within a specific time window based on scope of the research. This step is essential to ensure that the subsequent analysis and anomaly detection are accurate and reliable. The complete list of derived data requirements (DREQs) is presented in **Table** 6.1.

1. Data Preparation :

The data preparation process consists of several steps, as shown in **Figure** 5.2:

- Filter the data by a specific engine name (VP_ENG) and signal name (Signal_1) within a specified time window.
- Extract the selected engine signal data using the specified start and end dates for the dataset.

- Finally, generate the filtered dataset as a CSV file, which can be exported in the (.CSV) file file format, as shown in the code implementation in Appendix A.3.

- The same process can be applied to extract data for the remaining signals.

The extracted CSV file is loaded into a pandas DataFrame using pd.read_csv with the specified file path. Duplicate records are then removed based on the timestamp, as defined in data requirements(DREQ20).

5.1.2 Design of DDA method

This method is designed to detect anomalies in sensor datasets using two different approaches:

(1) Applying domain based data requirements (DREQ 1 to 7 and DREQ 14) with predefined signal threshold values to validate detected anomalies.

(2) Applying the Isolation Forest (IF) algorithm to automatically calculate the contamination parameter (DREQ 15) and validate the defined thresholds.

The remaining data requirements (DREQ 8 to 13) primarily support data understanding and the establishment of ground truth for the anomaly detection method. These requirements are guided by key considerations, including what data are needed, where the data originate, and who has access to them.

2. Anomaly Detection Methods :

The prepared dataset is loaded from a CSV file into the pandas library in Python, as shown in the code implementation in Appendix A.4. The CSV file is read into a pandas DataFrame using pd.read_csv(). The required numerical values are then converted into a NumPy array to prepare the data for the machine learning process. Finally, the relevant features, such as timestamp and measured_value, are selected and prepared for further processing before running the anomaly detection model.

Among the different approaches discussed in Section 3.1, the Isolation Forest (IF) algorithm was selected based on the literature study. It is applied to learn the normal structure of the data and identify deviations from it. The IF algorithm works by constructing random decision trees, known as isolation trees. At each node of a tree, a feature is selected, and a split value is chosen between the minimum and maximum values of that feature. This partitioning process continues recursively until each data point is completely isolated in its own leaf node. Normal data points, which tend to cluster together, require more splits and therefore have longer path lengths. The anomaly score for each data point is computed based on the average path length across all trees in the ensemble. The shorter the path length, the higher the anomaly score and the more likely the data point is to be a data error outlier [9].

The anomaly detection method has several steps as shown in the activity diagram **Figure** 5.2.

- **Check for Threshold Condition:** When the threshold value for a signal is known (Yes branch), the corresponding data requirements can be directly applied to validate the signal (see Appendix A.5). For example, for Signal 1, the defined valid range (DREQ 2) is used to check whether the observed values fall within the acceptable limits. If a signal value exceeds the maximum threshold or falls below the minimum threshold, it is identified as an outlier and flagged as a data error value. This process ensures that only valid and reliable data are considered for further analysis and supports accurate anomaly detection.

- **Configure the Isolation Forest (IF) model :**

  In the case company, there are thousands of signals across multiple engine types, making it challenging to identify and define threshold values from different DBC files for each signal. This process requires significant effort, as each signal may have its own specifications and reference limits. When threshold values are not available, a data driven approach is applied using the Isolation Forest (IF) algorithm, implemented through the Python machine learning library scikit-learn. Initially, the dataset is sorted by timestamp and the index is reset to maintain the correct sequence of the data. The timestamp and measured_value columns are then selected as the input features for detecting abnormal pressure spikes. By applying the IF algorithm to these features, abnormal patterns can be effectively identified based on deviations in the data distribution.

  Parameter configuration : The IF model is configured with hyperparameter **contamination set to "auto"** and **number of estimators is 100** as a default parameters (Appendix A.6) for configuring IF model. The contamination controls the cutoff threshold to decide which points are normal or anomaly. In this way, the model does not rely on a fixed or user defined percentage of anomalies. Instead, it automatically estimates a practical and efficient threshold based on the data distribution. This allows the algorithm to initially assess which points appear more isolated without forcing a strict cutoff. At the same time **random_state = 42**, ensures that the model produces consistent and reproducible results each time it runs by fixing the randomness in how trees are built.

  The IF model produces 44% of the anomalies and IF score ranges from [-0.2082, 0.0795]. The IF_label shows (-1) are anomaly, IF_label shows (1) are normal, and IF_score is continuous score (lower range = more anomalous). In this case, 44% of the data points (more suspicious score) are flagged as anomalies, which does not imply that all of them are true anomalies.

  Therefore, validation relies on retraining a machine learning model to automatically estimate the contamination rate based on highly suspicious anomaly scores generated by the IF algorithm. The model considers the lower 5% of anomaly scores as an initial subset to capture extreme deviations in the data.

Within this subset, the largest jump between consecutive scores is identified (see Appendix A.7), which serves as an adaptive determining the contamination level. Additionally, differences between consecutive data points are analyzed to detect significant gaps, which indicate unusual patterns in data behavior.

The IF algorithm approach enables adaptive anomaly detection based on the actual data distribution rather than relying on fixed thresholds. The outlier classification is defined according to (DREQ 17), where anomalies are categorized into relevant outliers and data error outliers. Relevant outliers (DREQ17.1) represent data points that still reflect normal engine behavior for a specific signal, while data error outliers (DREQ17.2) correspond to incorrect measurements that do not fit engine behavior, such as unrealistic high or low values caused by various factors.

Based on the identified patterns (DREQ19), the model classifies each data point as either "Relevant" or "Data Error Outlier" as described in **Table** 5.1. To enhance the interpretation of detected anomalies, time-context analysis and multi-signal correlation are performed and visualized using graphs (DREQ18).

Table 5.1: Characteristics required for domain-driven anomaly detection

S.No.	Terminology	Description
1	Relevant outliers	Relevant outliers shall be defined that does fit engine behavior for a specific engine signal. (Within threshold range).
2	Data error outliers	Data error outliers shall represent incorrect measurements that does not fit engine behavior like unrealistic high due to various factors for each specific signal. (Above threshold range)
3	True anomalies	Identified anomalies that are confirmed by domain experts are considered true anomalies.

- **Time context analysis :** The identified anomalies are extracted particularly by hourly patterns from the timestamp. This enables the identification of recurring clusters of spikes grouped at specific time interval. The output of detected anomalies includes aggregating anomaly counts over time windows.

- **Multiple signal correlation analysis :** The identified anomalies are correlated with other signals such as engine speed (signal_5), start and stop request signals (Signal_6) are applied. Thus, not only detects anomalies but explains them by linking pressure spikes to real engine behavior events such as engine starts and failure.

The results are visualized using graphs to better understand the detected patterns

and identify consistent behavior, as discussed in Chapter 6.2.1. The identified anomalies are further investigated to determine their root causes. This design enables not only the detection of anomalies but also the analysis of their temporal distribution. This examines the patterns or data points which spread over time to identify repeated behavior at specific hours or intervals. Thus, supports root cause analysis by revealing sensor glitches, periodic patterns, and engine operational behavior. Finally, the identified anomalies are validated through domain expert review to ensure their correctness and reliability.

Overall, the artifact proposes an anomaly detection method, and provides a structured list of anomaly instances.

6

Results

This chapter presents the results of the study based on the analysis of the collected data. The results for RQ1 focus on identifying challenges faced by practitioners through interviews, thematic analysis, documentation and finally derive the data requirements. RQ2 presents the characteristics and functionality for the method for anomaly detection of derived data requirements by using the defined data requirements. Finally, RQ3 focuses on evaluating the effectiveness of the proposed DDA Method through domain expert perception of the overall anomaly detection process and expert validation of the identified anomalies.

6.1 Defined Data Requirements for Anomaly detection

The results for RQ1 focus on identifying key data requirements based on insights from interviews and document analysis. The findings highlight the main challenges faced by practitioners, importance of documentation and the need for method to detect the anomalies.

6.1.1 Challenges in managing sensor data through interviews

A total of eight interviews were conducted with professionals working in data related roles, including data engineers, data scientists, data analysts and field test engineers. The participants had experience in handling sensor data, data pipelines, and field test systems. This diverse background helped in capturing a broad range of practical challenges and insights related to finding unusual pattern occurring in data. A pilot interview was conducted to evaluate the clarity and relevance of the interview questions. The initial questions were found to be generally clear, but some were simplified to improve understanding and encourage detailed responses. Minor adjustments were made to the structure and flow of the questions based on the feedback. Overall, the pilot study helped ensure that the final interviews were effective in capturing meaningful insights.

Results of Thematic Analysis

The interviews were analyzed based on thematic analysis which resulted in three main themes and eight sub themes that answers the **RQ1**. The main themes identified include Practitioners' Challenges, Documentation and Data scope and a Ground Truth Validation. These themes reflect the key issues, practices, and requirements needed to support anomaly detection across systems. The findings are summarized below.

Theme 1: Practitioners' Challenges

This theme captures the practical difficulties faced by data practitioners when analyzing sensor data of field test data. It includes challenges related to unrealistic value, unknown threshold value, anomaly detection and root cause analysis.

Here are the detailed explanation of sub themes and examples:

- **Difficulty in Identifying Data Anomalies:** Participants reported challenges in identifying true anomalies due to the presence of unrealistic signal values, especially those exceeding maximum thresholds. For example, one participant stated, "Most common fault pattern observed is the maximum threshold exceeding the limit," while another mentioned, "We have anomalies all the time, but they are not true anomalies."
- **Challenges in Classification of Outliers:** Participants highlighted difficulties in distinguishing between valid behavior and incorrect data. Instead of removing data, they preferred flagging suspicious values. This is reflected in statements such as, "We just don't remove the bad data, we just have a column that says if it is valid = True or no," and "Flag certain rows as suspicious, not remove them."
- **Limitations in Anomaly Detection:** Participants emphasized that anomaly detection is difficult due to unclear or unknown threshold values, especially for new engine systems. For instance, one participant stated, "New engine introduced has higher speed and runs at high power, all that signals threshold were unknown," while another noted, "No, We have multiple or thousands of dbc files with different thresholds for different engines."
- **Ensuring a Valid Dataset:** Maintaining a valid dataset was identified as a key challenge due to missing values and duplicate records. Participants mentioned, "We have found some null-values, and those we take away," and "I don't want duplicate values for each timestamp." Additionally, issues with system limitations were highlighted, such as "The system is not really built for handling duplicates."
- **Lack of Context for Data Interpretation:** Participants reported that lack of contextual information makes it difficult to interpret anomalies correctly. For example, one participant stated, "This might be an engine reset, and then I'm like, how am I supposed to know that?" Another emphasized, "It's always easier to fix something if you know the context in which the anomaly happened."

These findings illustrate that practitioners face both operational and technical challenges that impact the reliability and usability of field test data, thereby providing clear answer for RQ1.1 by identifying the difficulties involved in managing and analyzing sensor data for anomaly detection.

Theme 2: Documentation and Data scope

This theme focuses on the role of documentation and reference materials, such as DBC files, in defining dataset scope and supporting data quality assessment. Clear and well maintained documentation enables consistent interpretation, validation, and processing of data.

Here are the detailed explanation of sub themes and examples:

- **Challenges in Document Analysis:** Participants highlighted several challenges related to the accuracy and maintenance of DBC files used as reference documentation. Inaccurate definitions were found to directly impact data reliability, as reflected in the statement, "If the DBC file is wrong, the data can be wrong." In addition, participants noted that these files are not frequently updated, which creates issues when new engines or signals are introduced. This is supported by the observation that DBC files are "updated rarely when new engine and signals comes." Furthermore, a lack of clarity regarding how threshold values are defined and governed was identified as a concern, as one participant stated, "It would also be good to understand how DBC values is governed and where the definition of maximum is defined." Together, these issues highlight limitations in documentation quality and its impact on data interpretation.
- **Need for Signal Prioritization and Data Scope:** Participants indicated that working with large datasets requires focusing only on important signals. This is reflected in statements such as, "We ask stakeholders what signals are most important," and "I can give you a list of the most commonly used signals." Additionally, the complexity of multiple data sources was highlighted, "No, We have multiple or thousands of dbc files with different threshold signals for different engines."

The analysis highlights that well-maintained documents and reference files are critical for ensuring consistency and reliability in data handling. These findings directly answer RQ1.2 by showing how document analysis, particularly of DBC files, helps in identifying key data requirements such as valid thresholds and signal definitions are necessary for effective anomaly detection.

Theme 3: Ground Truth Validation

This theme explains how ground truth validation helps confirm if detected anomalies are correct, using domain expert reviews to support accurate evaluation.

Here are the detailed explanation of sub themes and examples:

- **Difficulties in Evaluation of the anomalies :** Participants highlighted the importance of validating anomalies using reliable reference points to ensure accurate interpretation of results. A key challenge identified was the lack of clear ground truth for confirming whether detected anomalies are valid. For example, one participant stated, "The ground truth would be like take the manually identified anomalies with domain experts," emphasizing the need for expert vali-

dation. Additionally, participants pointed out the importance of evaluating data based on quality expectations such as accuracy and consistency, as reflected in the statement, "It's always based on the three expectations which are the data is correct, the data is complete and the timeliness of the data." These insights highlight that effective evaluation requires both domain expertise and defined accuracy criteria to validate anomalies reliably.

Ground truth validation helps to detect anomalies, confirm the accuracy of anomalies, and supports quantitative evaluation methods for field test data, which answers **RQ3.2**.

The relationship between statements, keywords, codes, sub-themes, and themes derived from the interviews is illustrated in the Appendix A.2. This study addresses all the research questions **RQ1** and **RQ3.2**.

6.1.2 Derived data requirements

As defined in **Section** 1.3, a total of **twenty data requirements** were identified in relation to RQ1. The data requirements were structured using an adaptive template comprising a unique data requirement ID, requirement type, and a description of each requirement. Table 6.1 presents the complete list of derived data requirements. The requirement type column was used to distinguish between functional requirements and constraints. These requirements were derived through semi-structured interviews with practitioners, thematic analysis of the collected data, and document analysis. Together, these requirements provided the foundation for identifying anomalies in the proposed anomaly detection method.

The data requirements are divided into two main categories: constraints (DREQ1 - DREQ13) and functional requirements (DREQ14 - DREQ20). These also include both general constraints and signal specific domain constraints defined in accordance with the DBC file. DREQ1 specifies a general constraint limiting the scope of the requirements to the VP_ENG engine, ensuring that only relevant signals are included in the analysis, while other signals require separate threshold definitions. DREQ2 - DREQ7 define domain specific constraints for individual engine signals, where each signal is associated with a valid numerical range as specified in the DBC file. DREQ8 - DREQ13 primarily focus on data validation, data processing, and data governance aspects of the anomaly detection method.

The functional requirements (DREQ14 - DREQ20) define the core logic of the anomaly detection method, including anomaly identification and outlier classification. DREQ14 and DREQ15 focus on detecting anomalies with and without predefined thresholds. DREQ16 and DREQ20 describe the characteristics of a valid dataset record, including completeness and duplication checks. DREQ17 - DREQ19 define the outlier classification logic, incorporating temporal context and multi-signal correlation to improve detection accuracy.

Summary of RQ1: A list of data requirements were derived by collecting data through interviews and document analysis to understand the challenges faced by data practitioners. The collected data was analyzed using thematic analysis. Based on these derived data requirements the implementation of the domain driven anomaly detection process will be followed.

Data Req ID	Req type	Data Req desc
DREQ1	Constraint	The requirements defined must be specific to the VP_ENG engine scope and other relevant signals require separate threshold values as specified in the DBC file.
DREQ2	Domain Constraint	The Signal_1 for the VP_ENG engine shall have a valid range of 0 to 500, as specified in the DBC file.
DREQ3	Domain Constraint	The Signal_2 for the VP_ENG engine shall have a valid range of 0 to 499, as specified in the DBC file.
DREQ4	Domain Constraint	The Signal_3 signal for the VP_ENG engine shall have a valid range of -10 to 210, as specified in the DBC file.
DREQ5	Domain Constraint	The Signal_4 signal for the VP_ENG engine shall have a valid range of -100 to 1000, as specified in the DBC file.
DREQ6	Domain Constraint	The Signal_5 signal for the VP_ENG engine shall have a valid range of 0 to 5000, as specified in the DBC file.
DREQ7	Domain Constraint	The Signal_6 signal for the VP_ENG engine shall have a valid range of 0 to 5, as specified in the DBC file.
DREQ8	Domain Constraint	The method for anomaly detection shall use engine signals such as pressure, temperature and RPM, along with their corresponding threshold values and signal definitions from the DBC file.
DREQ9	Domain Constraint	The threshold values shall always be validated against correct DBC definitions and engineering specifications with field test engineers on engine operating limits.
DREQ10	Domain Constraint	The system shall collect engine data from the CAN bus network and process the received data using the DBC file corresponding to the specific engine.
DREQ11	Domain Constraint	Ground truth and accuracy of anomalies shall be validated with domain experts through manual identification of anomalies.
DREQ12	Domain Constraint	DBC files shall be updated whenever engine specifications or signal definitions change.
DREQ13	Domain Constraint	The access of DBC files shall be restricted to authorized engineers to protect data integrity and prevent unauthorized modifications to signal definitions or thresholds.

DREQ14	Functional	The method for anomaly detection shall detect anomalies based on predefined threshold limits.
DREQ15	Functional	The method for anomaly detection shall detect anomalies without a predefined threshold limit by integrating engine domain knowledge.
DREQ16	Functional	A valid record in the dataset shall have the required column values and atleast one non-null or non-NAN value among the signal values columns (StringValue, IntValue, ScalarValue, RawValue, and VectorValue). Records where all these signal values are null or NAN values shall be considered invalid.
DREQ16.1	Functional	A valid record in the dataset with multiple signals shall have non-null values to validate whether a data point is an outlier or not.
DREQ17	Functional	The outlier classification shall be categorized into relevant outliers and data error outliers.
DREQ17.1	Functional	The relevant outliers shall be defined that does fit engine behavior for each specific signal.
DREQ17.2	Functional	The data error outlier shall be represent incorrect measurements that does not fit engine behavior like unrealistic high and low values due to various factors for each specific signal.
DREQ18	Functional	To validate whether a data point is an outlier or not, context in terms of time and multi signal correlation shall be required.
DREQ19	Functional	The data error outlier shall be classified and flagged for each specific signal.
DREQ20	Functional	The structured dataset shall be checked for duplicate records based on the timestamp column defined.

Table 6.1: Data Requirements derived for anomaly detection

6.2 Applied example of a domain driven anomaly detection method

To answer RQ2, we apply our proposed artifact to detect anomalies based on the findings from RQ1. The DDA Method is applied to Signal_1 using predefined thresholds and to Signal_3 without predefined thresholds.

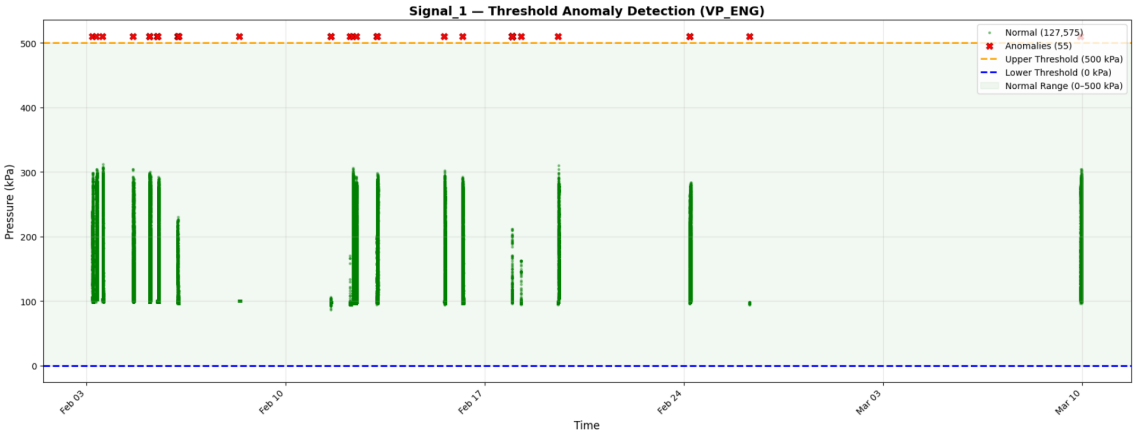

Figure 6.1: Yes branch time series graph for Signal_1 with known threshold

The detected anomalies exhibit distinct characteristics, such as sudden spikes, extreme deviations from the normal operating range, and in some cases clustering at specific time intervals indicating common anomaly patterns. The Signal_1 sensor readings collected from February to March 2026 have a predefined threshold data requirement (DREQ 2) of 0 to 500 kPa (kiloPascal). Most of the detected outliers represent fluctuations within normal operating conditions, while 55 data points exceed the defined threshold, as shown in Figure 6.1. These high pressure events occurring at approximately 510 kPa require further investigation.

The IF model was applied to Signal_3. The anomaly detection model first analyzes the data to learn normal behavior. It then compares each data point with the learned pattern and assigns an anomaly score using a gap detection technique to automatically calculate the estimated contamination parameter, thereby improving accuracy. The normal operating threshold ranges from 20°C to 72°C. The IF model detected anomalies on March 2 at approximately 10:00, with values between 63°C and 69°C, as shown in Figure 6.2. These observations were flagged because the IF model identified unusual patterns in the feature space, which consists of the hour, day of the week, minute, and the measured value.

As a result, the model classified 132,595 readings as normal (label = 1) and flagged 22 readings as anomalies (label = -1). All 22 anomalous readings occurred within the normal operating temperature range of 72°C but were clearly isolated from the surrounding observations and successfully distinguished from normal operational temperature variations. These detected anomalies were subsequently discussed with domain experts to confirm whether they represented true anomalies.

6.2.1 Root cause for the identified anomalies

We investigated all the identified anomalies using the DDA Method. During the period from February 3 to February 6, we analyzed the temporal behavior and contextual evolution of Signal_1 to identify anomaly patterns and clustering. By filtering the dataset to this specific time window and analyzing it on a daily basis,

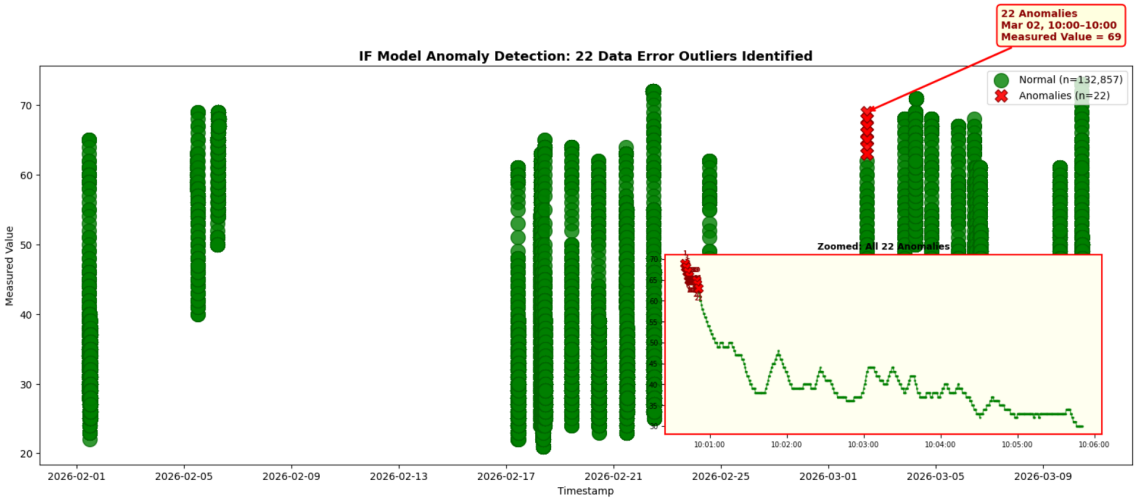

Figure 6.2: No branch time series graph for Signal_3 with unknown threshold

the clustering of high pressure spikes provided contextual insights into when and how anomalies occurred within operational cycles. This demonstrates that the DDA Method supports anomaly identification and contextual interpretation based on time windows and multi signal correlation.

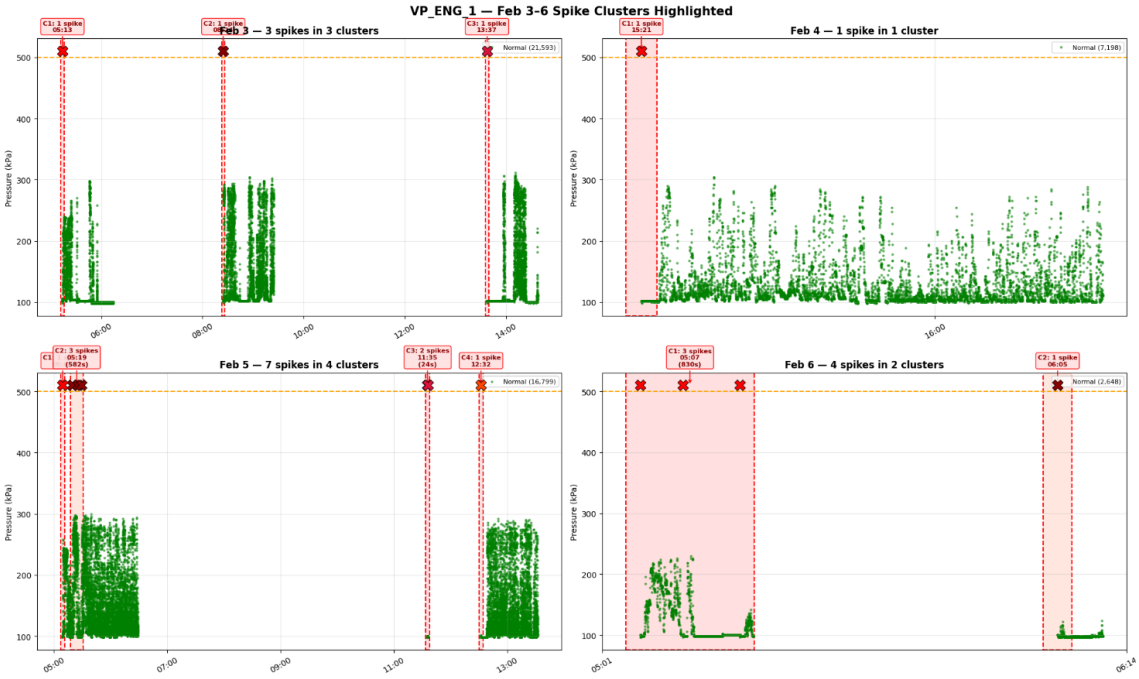

Figure 6.3: Time context analysis for Feb 3rd to Feb 6th dates

- **Time context Analysis:** From the Figure 6.3, a trend is observed with most spikes occurring during early morning. These spikes tend to occur in the 05:00–05:30 AM, often as short bursts. Feb 5th and 6th shows longer duration clusters (582s, 830s) during those mornings. The green normal readings surrounding the spikes show the pressure was otherwise stable at 96–104 kPa, confirming these are true transient anomalies rather than sensor drift. The

pressure jumps from 100 kPa to 510 kPa and then immediately returns to 100 kPa. These are sudden, short-duration spikes that may occur when the driver switches the engine power on and off rapidly or due to a wiring issue. They last only a few seconds, after which the sensor readings return to normal.

What's more interesting some dates especially, Feb 5th and Feb 6th from Figure 6.3, these spikes don't happen once, they just last longer and occurs in groups. This suggests that something in the engine may be starting to go wrong during those periods and these repeated spikes point to problems that should be looked into more closely as described by the domain experts.

From the Figure 6.4, three anomalies are confirmed by the domain experts as transitional e.g., a failed or restart attempt are at 510 kPa. Most readings are normal between 100 to 280 kPa and only spiked at 510 kPa. Looking at the green line, before the anomaly period, pressure is relatively quiet at 100 - 170 kPa. During 05:10–05:29, the signal becomes much nosier. This seems to be when engine is actively being cranked or in started status. The three spikes to 510 kPa occur within this noisy cranking period at 05:19, 05:25, and 05:28. Afterward, the signal returns to normal. This indicates that the anomalies are not random but are associated with specific operational conditions during engine start-up attempts.

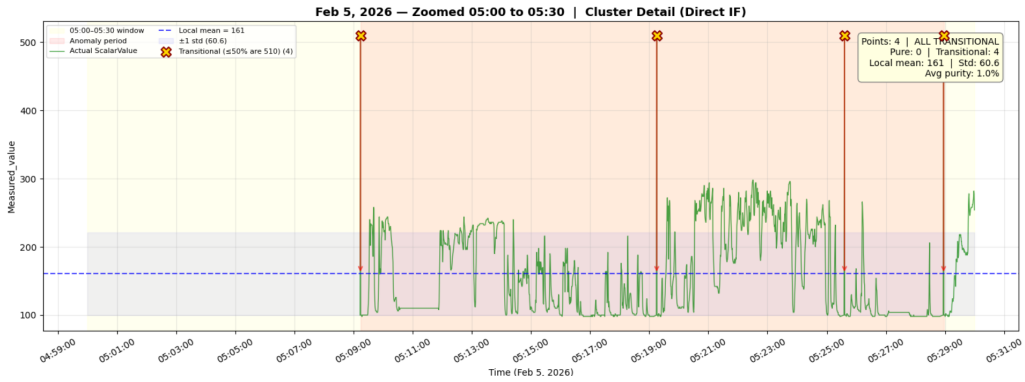

Figure 6.4: Time context analysis for Feb 5th day 2026

- **Multi Signal Correlation:** The multi-signal correlation analysis provides deeper insight into the root cause of the detected pressure anomalies by linking Signal_1 with other correlated signals, such as engine speed (Signal_5) and the start request signal (Signal_6). At each instance of the 510 kPa spike, the engine speed drops to 0 RPM, while the start request signal simultaneously transitions to OFF, indicating the exact moment when a start attempt is terminated, as shown in **Figure** 6.5 and this behavior was described by the domain experts. When viewed alongside the earlier time-context analysis, these spikes consistently occur during the unstable cranking phase and coincide with engine start attempts. Thus, the observed correlation is explainable, as these events are associated with failed engine starts, particularly during early morning operation.

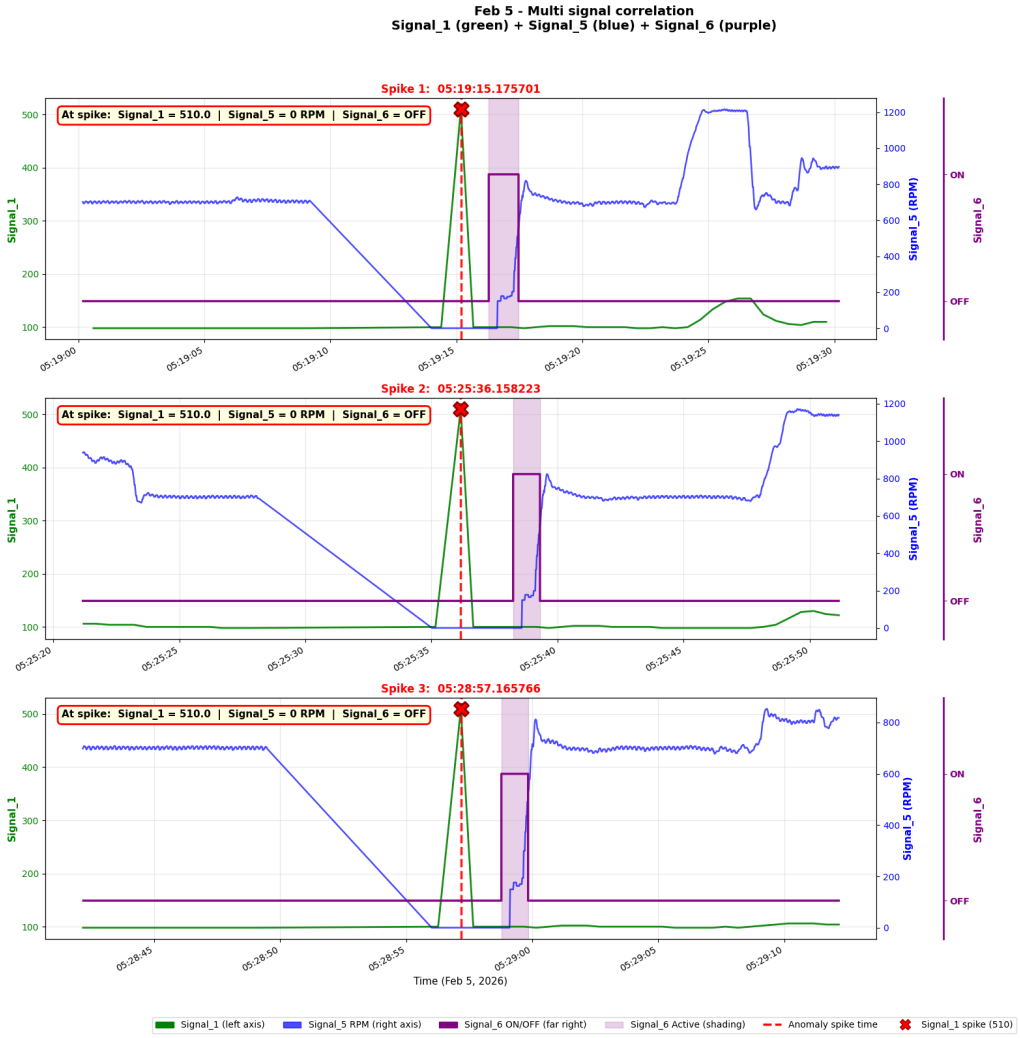

Figure 6.5: Multi signal correlation for Signal_1 with Signal_5 and Signal_6

Overall, the engine operates stably under normal conditions, but pressure spikes consistently occur during specific operating conditions, indicating a repeatable and explainable cause. This analysis approach can also be extended to other signals to validate similar patterns and further investigate the underlying causes. By applying time based clustering and multi-signal correlation across additional signals, the relationships associated with the detected pressure spikes can be further confirmed.

Summary of RQ2: The method for DDA is implemented using the defined data requirements and the engines sensor data as inputs. The prepared dataset is then configured using applied predefined threshold and Isolation Forest algorithm to detect the anomalies. To give context to the identified anomalies, time context analysis and multi-signal correlation analysis is performed for in depth analysis. An example (Signal_1 and Signal_3) was applied for the implemented artifact and was discussed with domain experts to confirm the true anomalies.

6.3 Effectiveness of the DDA Method

The answer **RQ3**, we present the evaluation of the proposed DDA Method in identifying anomalies based on data requirements. The evaluation consists of both qualitative and quantitative assessments to determine the effectiveness of the proposed method. First, the findings from the expert reviews are presented to examine how domain experts perceive the proposed DDA Method in comparison with the traditional anomaly detection approach. Next, the quantitative evaluation results are presented by validating the anomalies identified by the proposed DDA Method through domain expert confirmation and computing the validation rate.

6.3.1 Analysis on the DDA Method

To address **RQ3.1**, four domain experts from the case company, representing the roles of field test engineer, data scientist, and data engineer, evaluated the proposed DDA Method. The interviews focused on understanding the experts perceptions of the proposed method, its applicability in industrial practice, and its comparison with the traditional anomaly detection approach. The findings are presented according to the interview categories summarized in **Table** 6.2.

S.No.	Category	Category description
1	Data Preparation	The section covers the required inputs for the proposed DDA Method, including the derived data requirements, DBC file, and sensor data used for anomaly identification.
2	DDA Method	This section focuses on experts perceptions of the DDA method, including its usability, applicability, effectiveness and comparison with the traditional anomaly detection approach.
3	Improvements and Suggestions	The final section addresses questions based on experts suggestions for improving the proposed DDA method and its adoption within the case company.

Table 6.2: Category based on the interview questions

Results of Narrative Analysis

The results of the narrative analysis are presented according to the interview categories described in **Table** 6.2. For each category, the perspectives of the domain experts are compared to identify common viewpoints as well as differences in their experiences with the proposed DDA Method. The findings provide insights into the perceived usability, applicability, effectiveness, and potential improvements of the DDA Method in comparison with the traditional anomaly detection approach. The results are summarized in the following sections.

Category 1: Data Preparation

This category presents the domain experts perspectives on the data preparation stage of the proposed DDA Method. The discussion focused on the use of derived data requirements, DBC files, and the data extraction process as inputs to the DDA Method. The experts also reflected on how these practices compare with the existing workflow, where no structured approach for anomaly identification based on data requirements currently exists. Sample narratives from the interviews are presented below.

Interviewee 1: *"For many signals, data requirements catch quite a lot of anomalies, but might miss some difficult anomalies. Relying on the DBC file has not been done before, so it is a new scope as it is done by field test engineers. Also, converting to a CSV file in terms of computation might be slow, but it is only a slight concern and the current approach is still fine."*

Interviewee 2: *"I think introducing more data requirements into our existing processing pipeline would make a big difference. We just need the proper DBC file for it. If the anomaly detection process performs well, then I would say there is absolutely nothing wrong with the approach."*

Interviewee 3: *"It solves part of the problem and is beneficial to some extent. The thresholds should work across engines. We don't currently use DBC files, but it is good to have a limited scope using CSV files. If this is implemented as a continuous process, we should use tables instead."*

Interviewee 4: *"I think it looks great. The data requirements seem appropriate for commonly used signals. No, we are definitely not using DBC files today, but this is the direction we want to move towards."*

The narratives indicate that all four domain experts considered the data preparation stage of the DDA Method to be practical and applicable within the industrial context. They agreed that incorporating derived data requirements provides a structured foundation for anomaly identification, while the use of DBC files introduces information that is not currently utilized in the existing workflow. In contrast to the traditional approach, where anomaly detection largely depends on manual investigation and expert knowledge, the DDA Method provides a systematic process for preparing data prior to anomaly detection.

Although the experts viewed the overall approach positively, they also suggested several improvements. Interviewee 1 highlighted the computational overhead of CSV based processing, while Interviewee 3 recommended replacing CSV files with database tables if the method is deployed as part of a continuous production pipeline. The experts also emphasized that additional data requirements could be incorporated over time to extend the applicability of the DDA Method to a broader range of engine signals. Overall, the experts perceived the data preparation stage as a practical enhancement that supports the implementation of the DDA Method and complements the current industrial workflow.

Category 2: DDA Method

This category presents the domain experts perceptions of the proposed DDA Method. The discussion focused on the overall model configuration, the comparison with the traditional anomaly detection approach, the use of multi signal correlation during anomaly analysis, and the challenges associated with evaluating the identified anomalies. Sample narratives from the interviews are presented below.

Interviewee 1: *"In the model configuration methods, future work on other relevant anomalies could have been considered.Also, evaluating it overall is difficult as labeling data is difficult as we might not know it is a true anomaly or not and precision on a small scale is still okay."*
Interviewee 2: *"I think it sounds very reasonable and it sounds like a great composition on the model configuration. Also, I think it was very interesting to have like these, finding these correlations between anomalies and having an in depth analysis. No, I have no idea on evaluation metrics but I am like impressed by this."*
Interviewee 3: *"It's good to compare the model methods, it's part of the experimentation process. Detection is a very useful tool to identify potential errors. It could also be very useful for identify anomalies using those analysis."*
Interviewee 4: *"I think so multi signal correlation works well, because it's very good to see if the engines build is running."*

Overall, the narratives indicate that the domain experts had a positive perception of the proposed DDA Method. Interviewees 2, 3, and 4 agreed that the method provides a practical approach for anomaly detection. Interviewee 2 described the model configuration as a great composition and appreciated the use of multi signal correlation for enabling a more in depth analysis of anomalies. Similarly, Interviewee 4 agreed that multi signal correlation works well as it provides better insight into engine behavior. Interviewee 3 also supported the approach and recognized anomaly detection as a useful tool for identifying potential errors. In contrast, Interviewee 1 did not disagree with the proposed method but highlighted limitations and future improvements. The evaluation of anomaly detection methods remains challenging due to the lack of labeled anomaly datasets. Although the experts differed in their perspectives on the evaluation process. Therefore, conducted interviewees agreed that the DDA Method is a promising enhancement over the current method by providing a more systematic approach to anomaly identification, while suggesting further improvements.

Category 3: Improvements and Suggestions

The final category focused on the overall feedback and suggestions provided by the domain experts regarding the proposed DDA Method. The discussion included reflections on the overall workflow, opportunities to reduce manual effort, and recommendations for future improvements. Sample narratives from the interviews are presented below.

Interviewee 1: *"Interviews and data requirements derivation was overall good and*

the key part was converting the interview data to requirements. The workflows are clear and easy to follow. Issues we face with that large volumes of data."
Interviewee 2: *"Just like that great work, like it's very fun to see it and its clear and then the journey that you've made is an amazing progress in just a few months. Maybe we have to take that as an input and we need to automate for our machine learning models in the future. Yes, make it less tedious and like manual work."*
Interviewee 3: *"I think it's important to set a very clear scope from the beginning and I think you guys followed that and narrowed it down very specifically."*
Interviewee 4: *"It would definitely be great to have this automatically, just to save time. I think you kept it at a great level."*

Overall, the final section expressed a positive perception of the proposed DDA Method from all four interviewees and agreed that the overall workflow is well structured and practical for industrial use. Interviewee 1 appreciated the systematic process of translating interview findings into data requirements while highlighting scalability as a future challenge when processing large volumes of data. Interviewee 2 and Interviewee 4 both suggested increasing the level of automation to reduce manual effort and facilitate integration into future machine learning workflows. Interviewee 3 emphasized that maintaining a well defined scope contributed to the clarity and feasibility of the proposed method. Collectively, the experts agreed that the DDA Method provides a solid foundation for anomaly detection and identified automation, scalability, and broader industrial integration as the primary directions for future work.

Summary of all interviews

Overall, the interview findings indicate that the domain experts perceived the proposed DDA Method as a practical enhancement over the current anomaly identification workflow. The method introduces a more structured approach to data preparation and anomaly detection while highlighting opportunities for future improvements such as automation and scalability. **Table** 6.3 summarizes the key differences between the traditional approach and the proposed DDA Method based on the expert interviews.

S.No.	Categories	Traditional Method	DDA Method
1	Data Preparation	Relies primarily on manual expertise and existing data processing pipelines. DBC files and derived data requirements are not systematically used for anomaly identification.	Introduces derived data requirements together with DBC files to provide structured inputs for anomaly identification. The approach was considered practical, although experts suggested replacing CSV files with database tables for future deployment.
2	DDA Method	Anomalies are mainly investigated manually based on domain expert knowledge, with limited support for systematic analysis or multi signal correlation.	Provides a structured workflow for anomaly identification using data requirements, supported by time context and multi signal correlation analysis. Experts considered the methods practical and valuable for supporting anomaly investigation.
3	Improvements and Suggestions	Current workflow requires considerable manual effort and lacks automation for anomaly identification.	Experts recommended automating the DDA Method and improving scalability to support deployment within existing industrial pipelines.

Table 6.3: Summary of domain expert perceptions comparing the traditional method and the DDA Method

6.3.2 Analysis on the Identified Anomalies

This section presents the results for **RQ3.2** by evaluating the quality of the anomalies identified by the proposed DDA Method through domain expert confirmation. The evaluation assesses the proportion of detected anomalies that were confirmed as true anomalies by two domain experts, thereby determining the effectiveness of the proposed method in identifying meaningful anomalies based on data requirements.

As shown in **Table** 6.4, two datasets were used for the quantitative evaluation, with each dataset producing a list of identified anomalies that were subsequently reviewed by the domain experts. The results for each dataset are presented and discussed in the following sections.

S.No.	Dataset name	Threshold Method	Detected Anomalies	Expert Confirmed Anomalies	Validation Rate
1	Signal_1	Predefined threshold	55	27	49.1%
2	Signal_3	Auto calculated threshold	22	22	100%

Table 6.4: Quantitative evaluation of the identified anomalies using expert confirmation

Evaluation of Signal_1

Signal_1 was evaluated using the predefined domain threshold method to assess the quality of the anomalies identified by the proposed DDA Method. A total of 55 anomalies were detected, of which 27 were confirmed as true anomalies by two domain experts, resulting in a validation rate of 49.1%. To better understand these validation results, representative time series graphs are presented for both confirmed and non-confirmed anomalies. This analysis provides insight into the strengths and limitations of the DDA Method when applied using predefined domain thresholds.

Confirmed Anomalies

The identified anomalies were grouped into different time windows and presented as clusters of related anomalous events during the domain expert review. This enabled the domain experts to examine each anomaly within its operational context rather than as an isolated data point. The confirmed anomalies exhibited distinct deviations from the expected signal behavior and exceeded the predefined threshold. Representative examples of these confirmed anomaly clusters are presented in **Figures** 6.6 and 6.7 to illustrate the anomalous patterns identified by the proposed DDA Method and the rationale behind the experts confirmation.

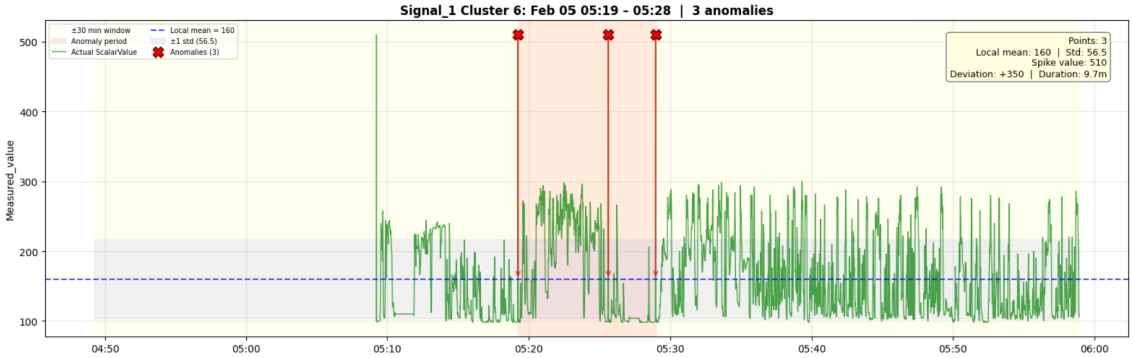

Figure 6.6: Anomalies confirmed during engine startup period for Signal_1

Figure 6.6 presents the first example of a confirmed anomaly identified in Signal_1. The detected anomalies occur during the engine startup period, where three consec-

utive measurements exceed the predefined threshold value. Unlike the surrounding measurements, which remain within the expected operating range, these observations form a continuous abnormal pattern over a short time window rather than isolated spikes. After reviewing the time series graph together with its surrounding context, both domain experts confirmed these observations as true anomalies because they represented a sustained deviation from the expected signal behavior during engine start up.

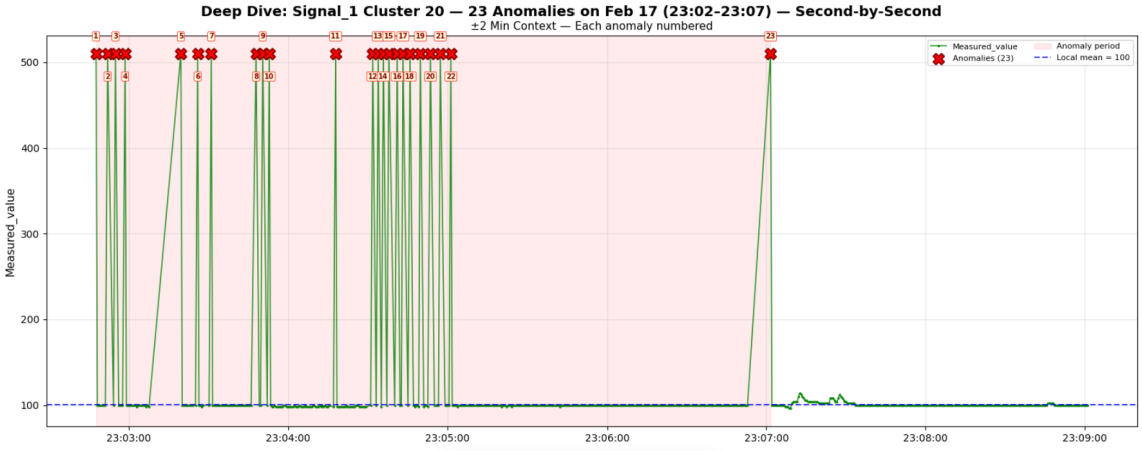

Figure 6.7: Anomalies confirmed during engine shutdown period for Signal_1

Figure 6.7, shows another example of confirmed anomalies identified during the engine shutdown period. In this case, a cluster of 23 anomalies is observed within a short time interval, where repeated peaks consistently exceed the predefined threshold. Instead of occurring as individual outliers, the anomalies appear as a sequence of repeated abnormal measurements, indicating persistent anomalous behavior. Based on the temporal pattern and the repeated occurrence of these deviations, both domain experts confirmed the entire cluster as true anomalies. This demonstrates that presenting anomalies within their time window context enables domain experts to distinguish genuine anomalous behavior from isolated signal fluctuations.

Overall, the confirmed anomalies in Signal_1 demonstrate that the proposed DDA Method was able to identify abnormal signal patterns. In both examples, the anomalies exceeded the predefined threshold and appeared as clustered deviations rather than isolated measurements and was interpreted as *data error outliers*. However, the validation rate of 49.1% indicates that, although the predefined threshold identified several genuine anomalies, it also detected observations that the domain experts considered to represent normal operating behavior. This highlights the importance of incorporating domain expertise to distinguish true anomalies from threshold based detections.

Non-confirmed Anomalies

The remaining anomalies identified in Signal_1 that were not confirmed by the domain experts were further analyzed to understand the reasons behind their rejection. Representative examples of these non-confirmed anomalies are presented in **Figures**

6.8 and 6.9.

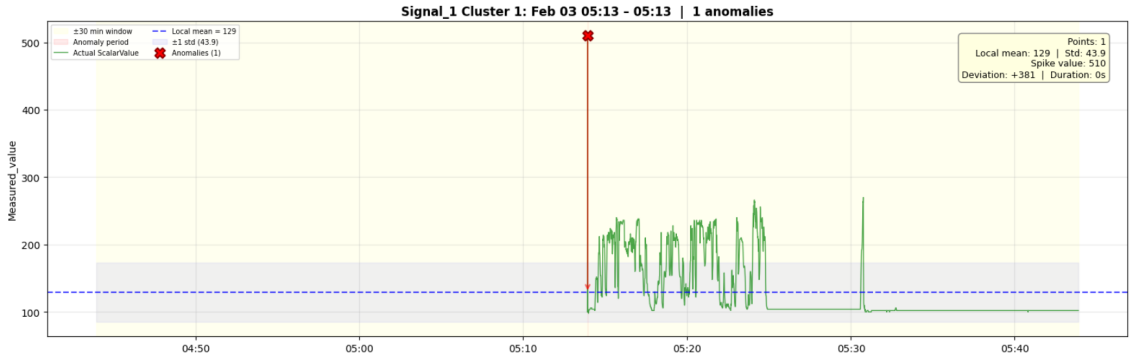

Figure 6.8: Anomalies non-confirmed during engine startup period for Signal_1

Figure 6.8 shows an example where the detected observations exceeded the predefined threshold but were not confirmed as true anomalies by the domain experts. Although the DDA Method classified these observations as anomalous, the experts identified them as acceptable operating behavior because they occurred during the engine start up period. When examined within the surrounding time window, the signal followed a pattern that was consistent with the normal operation of the engine rather than indicating an abnormal event.

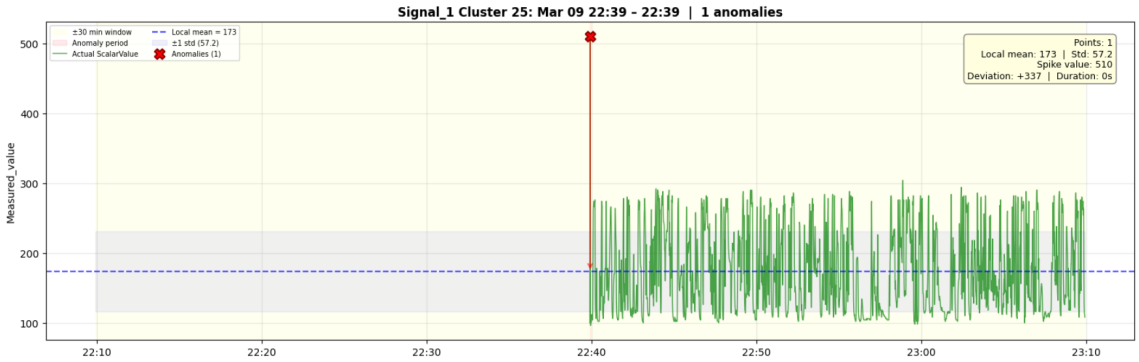

Figure 6.9: Anomalies non-confirmed during engine shutdown period for Signal_1

Similarly, **Figure** 6.9 presents another example of detected anomalies that were not confirmed by the domain experts. While these observations also exceeded the predefined threshold, the experts observed that similar signal patterns repeatedly occurred during engine shutdown periods. Rather than representing genuine anomalous events, these repeated threshold exceedance formed part of the normal operational characteristics of the engine.

Across the non-confirmed examples, the domain experts observed a recurring daily spike pattern in which threshold exceedance were concentrated during the early morning and late night periods, corresponding to engine startup and shutdown operations. Since these patterns occurred consistently at similar times of the day, they

were considered expected operational behavior rather than true anomalies. Consequently, although the predefined threshold successfully detected deviations above the threshold value, domain expert confirmation demonstrated that not all threshold exceedance represented meaningful anomalous events. These findings explain the validation rate of 49.1% for Signal_1 and highlight the importance of incorporating domain expertise to distinguish genuine data error outliers from acceptable operational behavior.

Evaluation of Signal_3

Signal_3 was evaluated using the auto calculated threshold method to assess the quality of the anomalies identified by the proposed DDA Method. A total of 22 anomalies were detected, all of which were confirmed as true anomalies by two domain experts, resulting in a validation rate of 100%. To further illustrate these validation results, representative time series graphs of the confirmed anomalies are presented and discussed based on the domain experts observations. This analysis demonstrates the effectiveness of the DDA Method in identifying meaningful anomalies using an automatically calculated threshold.

Confirmed Anomalies

Similar to Signal_1, the identified anomalies in Signal_3 were grouped into different time windows and reviewed by the domain experts using the corresponding time series graphs. This allowed the experts to evaluate the detected anomalies within their operational context before confirming them as true anomalies. Representative examples of these confirmed anomalies are presented in **Figure** 6.10. Unlike Signal_1, all detected anomalies in Signal_3 were confirmed by the domain experts, resulting in a validation rate of 100%.

Figure 6.10 presents a representative example of the confirmed anomalies identified in Signal_3. In this example, the detected anomalies occur below the predefined threshold value, where the maximum measured value is approximately 210. Consequently, these observations would not have been detected using the predefined threshold approach, which identified no anomalies for this time window. In contrast, the DDA Method with the auto calculated threshold identified 22 anomalies, beginning at a value of approximately 69 and gradually decreasing to around 63 over a continuous time interval.

Rather than appearing as isolated spikes, the detected anomalies form a consistent pattern of measurements that remain outside the expected operating behavior for the corresponding time window. The domain experts confirmed these observations as true anomalies because they represented *relevant outliers* that were not captured by the predefined threshold approach. This demonstrates the capability of the auto-calculated threshold to identify meaningful anomalous behavior beyond fixed threshold values. Although all detected anomalies were confirmed, further investigation is required to determine the underlying cause of this abnormal behavior.

Overall, the results for Signal_3 demonstrate that the proposed DDA Method

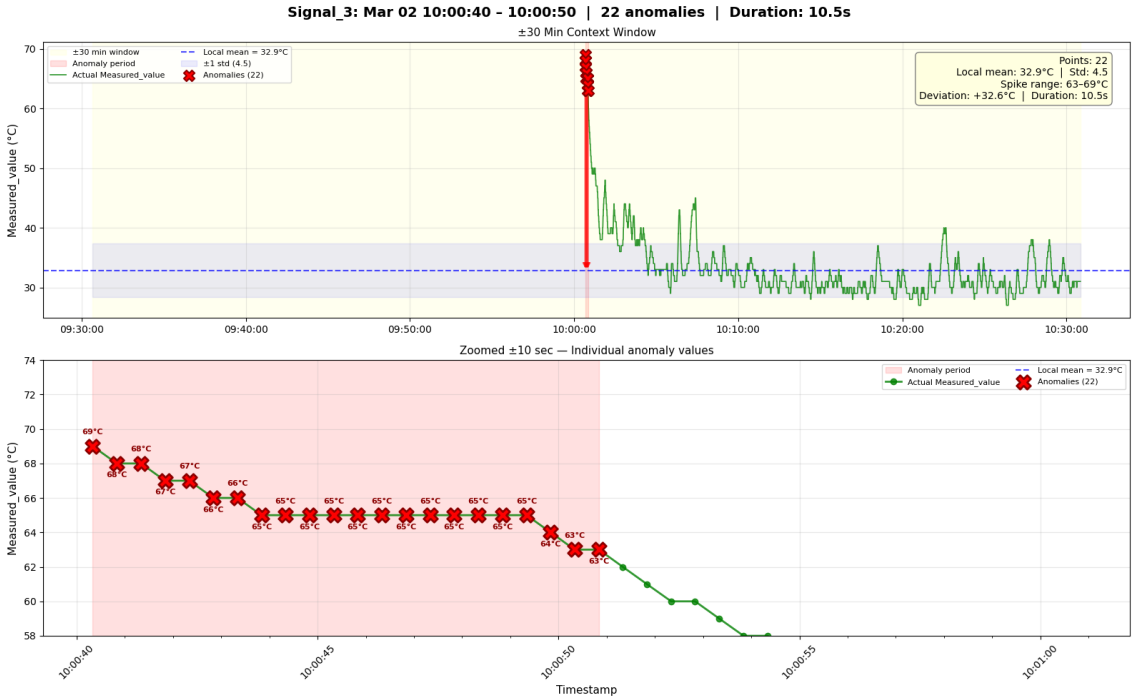

Figure 6.10: Anomalies confirmed during engine startup period for Signal_3

effectively identified meaningful anomalous behavior that was subsequently validated by the domain experts. Although the auto calculated threshold achieved a validation rate of 100% for Signal_3, this result should not be interpreted as indicating that the method consistently achieves perfect performance. Rather, the validation rate reflects the effectiveness of the method for the specific dataset and time period evaluated in this study. Further evaluation using additional datasets and operating conditions is required to assess the generalizability and robustness of the method.

Summary of RQ3: The effectiveness of the proposed DDA Method was evaluated using both qualitative and quantitative approaches. Qualitatively, the perceptions of the domain experts towards the DDA Method were analyzed in comparison with the traditional approach, where the narrative analysis indicated that the proposed DDA method provides a more structured and systematic process for identifying anomalies by incorporating domain knowledge into the anomaly detection workflow. Quantitatively, the quality of the identified anomalies was assessed using the validation rate, which demonstrated that the DDA Method identified meaningful anomalies across both datasets, including data error outliers and relevant outliers detected using predefined and auto calculated thresholds. Overall, the findings indicate that the proposed DDA Method is effective in identifying anomalies based on data requirements while emphasizing the importance of domain expert confirmation to verify the relevance of the detected anomalies.

7

Discussion

The results of this study provide useful insights into the impact of the proposed DDA Method. The findings from RQ1 show that the derived data requirements are highly valuable for anomaly detection. Our proposed DDA Method utilizes these data requirements to address an area that has not been explored in previous work by Khan et al. [16]. The derived data requirements also have practical value for the case company, as previous studies have not focused on systematically deriving data requirements for identifying and detecting anomalies. Furthermore, interviews with domain experts and document analysis highlighted the importance of the derived data requirements. The document analysis approach is still relatively new within the case company, and there may still be some uncertainty regarding the threshold values defined in the data requirements, as identified during the expert interviews.

The findings from RQ2 show how the derived data requirements directly support the DDA Method by reducing false positives, which was also identified as a major research gap by Khan et al. [16]. The results indicate that characteristics such as domain specific threshold values and contextual data requirements are important for accurately detecting anomalies and providing meaningful contextual insights. One challenge is that, although both approaches support anomaly detection, the approach based on automatically calculated thresholds must account for a wide range of datasets and cannot rely on a single set of data points. In contrast, the approach using predefined thresholds was able to handle the dataset as a whole because the threshold values were already known. Another challenge was that the evaluation focused on a single feature, which may limit the method's applicability when a new dataset is introduced without manually defined threshold values.

The findings from RQ3 also helped in understanding domain experts perspectives on the proposed DDA Method through interviews and quantitative evaluation. This addresses a gap identified in previous research by Chandola et al. [5] and Khan et al. [16], where limited guidance is provided on evaluating the quality of identified anomalies. Labeling the data and evaluating the anomaly detection methods were challenging, as previous studies have not extensively discussed or evaluated these activities using expert-defined data requirements.

The study also highlights the positive impact of the derived data requirements in identifying anomalies by transforming collected data into meaningful information.

The potential of combining domain expertise with Machine Learning (ML) models builds upon previous research by Khan et al. [16] and Chandola et al. [5], while also aligning with current trends in embedded software engineering [2].

7.1 Threats to Validity, Limitations and Future Work

This section outlines the limitations of the study and the potential threats to the validity of the findings. These limitations and threats should be considered in future work to further strengthen and enhance the study.

Threats to validity

Internal validity: During data collection, domain experts might misinterpret certain issues or may not accurately recall specific problems, which could introduce expert bias. Additionally, using a small set of engine datasets might not fully capture the diversity of the data requirements and could make the DDA Method appear more accurate than it actually is. This may limit the generalizability of the findings to the engine datasets explored and evaluated in this study and may not represent the wide range of scenarios encountered in real world systems.

External validity: The findings of this study are based on a specific engine and may not be applicable to other engine domains or industries. Therefore, the derived data requirements may not be applicable to other engine types or real world systems. Expanding the dataset to include a more diverse set of engine datasets would improve the generalizability of the findings and support the applicability of the proposed DDA Method across different engine domains.

Limitations and Future work

The thesis has several limitations arising from both the nature of the dataset and the practical constraints of the study. The primary limitation is the dataset used to evaluate the DDA Method. The dataset was specific to a single engine and did not cover a wider range of field test data from different engines. Future work should focus on including a more diverse range of datasets, which would support the derivation and evaluation of data requirements across different engine types. Based on the findings of this study, future work should investigate the proposed Data Requirement DREQ16, which involves validating that each record contains at least one valid signal value and ensuring that records with multiple signals contain non-null values before anomaly detection is performed.

In addition, the auto-calculated threshold approach in the DDA Method was restricted to a limited set of data points within the dataset. This reduced the reliability of the results, as these data points might not fully represent the overall dataset. Furthermore, the DDA Method used only a single feature to train the Isolation Forest model. In addition, both threshold approaches considered only data error

outliers, while the deeper analysis focused exclusively on these outliers. Therefore, future work should consider the entire dataset, include additional features for training the model, and perform a more comprehensive analysis of the relevant outliers detected by the DDA Method.

Another limitation was the evaluation of the DDA Method, which was challenging due to the absence of labeled datasets. This made it difficult to verify whether the identified anomalies represented true anomalous events. More appropriate evaluation metrics should be explored, and the datasets should be prepared in a way that better supports the evaluation of the detected anomalies. In addition, the dataset contained limited variation in the types of anomalies. While clear and significant outliers were relatively easy to detect, contextual and correlational outliers are often more difficult to identify and may have a greater impact. The absence of such anomalies limited the evaluation of the proposed DDA Method under more complex and realistic conditions. These limitations suggest that the evaluation process requires further refinement to provide a more comprehensive assessment of the anomaly detection methods. Additional evaluation methods beyond the validation rate could further strengthen the assessment.

The current implementation also has limitations, as each cell in the Databricks notebook is executed sequentially and requires significant user involvement. Therefore, future work should focus on extending the proposed approach by automatically deriving threshold values from the DBC file based on the defined data requirements. Furthermore, the Isolation Forest anomaly detection process could be integrated into the existing data cleaning pipelines at the case company, representing a practical extension of the proposed methodology. In the first stage, raw data could be ingested into the pipeline through Databricks. Once ingested, the data could be stored in Delta Lake tables, which provide reliable transactional storage and data versioning. The data preparation process could also be automated within the Databricks workspace using PySpark or pandas, enabling the preparation logic to scale efficiently across large volumes of data in a distributed environment. Once the dataset is prepared, the Isolation Forest model can be trained using machine learning techniques. Using the derived data requirements together with automatically calculated contamination parameter settings, the trained model could then be deployed as a batch inference job within the existing pipelines, where it could be scheduled to run at predefined intervals depending on the use case. Finally, the anomaly detection results could be presented to data analysts and engineers for review and further investigation.

7.2 Ethical Considerations

This study involves engine sensor datasets that may contain sensitive information. These datasets were handled in accordance with the confidentiality requirements of the case company. Sensitive information was not disclosed, and the datasets were presented in a way that protected confidential information by using alias names where appropriate. This study also involved interviews with domain experts, which

were conducted in accordance with established research ethics principles, including informed consent, confidentiality, and responsible data handling. The domain experts were informed about the purpose of the study before the interviews and their consent was obtained prior to participation.

AI based tools were used only to assist with improving the grammar, sentence structure, and overall clarity of the report. They were not used to generate the research methodology, results, analysis, or conclusions. All technical content, interpretations and final decisions remain solely the responsibility of the authors.

8

Conclusion

In conclusion, this thesis focused on deriving data requirements from multiple sources and applying them to field test datasets to support DDA Method. The derived data requirements identified relevant signal behavior, defined valid threshold ranges based on DBC files, and established a structured foundation for implementing and evaluating the proposed DDA Method. The developed DDA Method was able to identify anomalies effectively. The evaluation results, including feedback from domain experts, indicated that the proposed method was understandable, applicable, and valuable for supporting anomaly detection in the case company.

The quantitative evaluation demonstrated that the incorporation of data requirements enabled the method to identify anomalies when actual anomalous events were present in the dataset. However, the study also identified several limitations, including the need for more diverse datasets, additional evaluation metrics, and broader feature selection to improve the applicability of the proposed DDA Method in more complex real world scenarios.

A

Appendix

A.1 Data Collection Interview Guide Sample

Introduction:

The main focus of the interview is on identifying the problem and causes for improving data quality in engine datasets. The conducted interviews are part of the data collection process of the problem investigation phase to learn more related to the field test engines datasets and gather related information. It focuses on three different types of data collection to understand the domain knowledge better. The collected information has both qualitative and quantitative data and then analyzed for deriving the data requirements. With your consent, **this interview is recorded via audio** and will be kept confidential and used only for research and data analysis purposes.

Interview duration: 45-60 minutes per interviewee

General questions for All Roles:

1. How do you **describe your work** in simple words ?
2. What is your current **role and responsibility** related to the field test datasets ?
3. How long were you working with a case company and previously you had worked in the similar area?
4. Can you briefly **describe** the field test system and the type of data it generates ?
5. How is **data captured during field tests** ? What **challenges** do you face in collecting reliable and consistent data from the field?
6. How are these field test data currently **collected, stored, and used** ?
7. What **data quality issues** do you commonly face in current times ? (e.g., missing data, noise, delays)
8. Did you come across **some anomalies** that occur only under specific conditions ? (e.g., load, temperature, environment)
9. Which **signals or data features** in the field test datasets are mostly used from your experience ?
10. Are there any tools, rules, or techniques you commonly use for **data validation** for field test datasets ?
11. Can you give us a few examples of how we can **identify faults** in the data values ?
12. Are there **documents or examples** you recommend reviewing ? If available, how do you think they are helpful in understanding the existing system and operational guidelines ?
13. Can the **faults** in the field test datasets be **categorized** into certain patterns of events ?
14. Finally, to close the interview, what **advice** would you give to researchers dealing with **data quality** issues in field test datasets ?

A.2 Results of Thematic Analysis

Theme 1 — Practitioners challenges

Statement / quote	Keyword	Code	Sub-theme	DREQ	Theme
"Most common fault pattern observed is the maximum threshold exceeding the limit" — IN1	*maximum threshold*	*Unrealistic value*	**Data anomaly**	*DREQ-11.2*	**Practitioners challenges**
"If you get a very strange signal, it should only be like the maximum value." — IN2	*maximum value*	*Unrealistic value*	**Data anomaly**	*DREQ-11.2*	
"They have a maximum of 4000 Newton meters. Everything else will be deleted as an outlier." — IN5	*maximum*	*Unrealistic value*	**Data anomaly**	*DREQ-11.2*	
"We have anomalies all the time, but they are not true anomalies. They are just data quality problems" — IN3	*not true anomalies*	*Unrealistic value*	**Data anomaly**	*DREQ-11.2*	
"...might have type of spike or normal behaviour which is a similar type of error" — IN3	*spike behaviour*	*Unrealistic value*	**Data anomaly**	*DREQ-11*	
"We just don't remove the bad data, we just have a column that says if it is valid = True or no" — IN1	*is valid*	*Flagging suspicious data*	**Classification of outliers**	*DREQ-13*	
"flag certain rows of suspicious like by data, not remove them, but just have a flag so you can remove suspiciously bad data"	*suspiciously bad data*	*Flagging suspicious data*	**Classification of outliers**	*DREQ-13*	
"New engine introduced has higher speed and runs at high power, all that signals threshold were unknown value" — IN1	*unknown value*	*Threshold value unknown*	**Anomaly detection**	*DREQ-9*	
"The system is not really built for handling kind of duplicates." — IN4	*duplicates*	*Handling duplicates*	**Valid dataset**	*DREQ-14*	
"We have found some null-values, and those are we take away" — IN1	*null-values*	*Handling null-values*	**Valid dataset**	*DREQ-10 (future work)*	
"The biggest things I've seen there were data mismatches or missing data." — IN5	*missing data*	*Handling missing value*	**Valid dataset**	*DREQ-10 (future work)*	
"This might be an engine reset, and then I'm like, how am I supposed to know that?" — IN3	*engine reset*	*Lack of context*	**Context analysis**	*DREQ-12*	
"It's always easier to know how to fix something, if you know what context the anomaly happened in." — IN5	*fix something*	*Lack of context*	**Context analysis**	*DREQ-12*	

Sub-themes: data anomaly, classification of outliers, anomaly detection, valid dataset, context analysis, scope of datasets. Drawn from interviews IN1, IN2, IN3, IN4, and IN5.

Theme 2 — Documents and data scope

Statement / quote	Keyword	Code	Sub-theme	DREQ	Theme
"If the DBC file is wrong, the data can be wrong." — IN2	*DBC file*	*DBC file*	**Challenges in Document analysis**	*DREQ-15*	**Documents and Data scope**
"No, we have multiple or thousands of dbc files with different threshold signals for different engines." — IN2	*multiple*	*DBC file*	**Anomaly detection**	*DREQ-9*	
"It would also be good to understand how dbc values is governed and where the definition of maximum is defined." — IN2	*dbc values*	*DBC file*	**Challenges in Document analysis**	*DREQ-15*	
"The documentation we have restricted to other engineers to provide permission." — IN2	*restricted*	*Reference files*	**Challenges in Document analysis**	*DREQ-20*	
"Also, these files need to be updated rarely when new engine and signals comes." — IN2	*updated rarely*	*DBC file*	**Challenges in Document analysis**	*DREQ-19*	
"I can give you a list of the most commonly used signals, absolutely." — IN2	*commonly used signals*	*Prioritising signals*	**Scope of datasets**	*DREQ-1 to 7*	
"So how we approach it is that we ask stakeholders what signals are most important, we're going to clean this once" — IN1	*important signals*	*Prioritising signals*	**Scope of datasets**	*DREQ-1 to 7*	

Sub-themes: document analysis, scope of datasets. All quotes from interviews IN1 and IN2 that discussed DBC file governance in detail.

Theme 3 — Ground truth validation

Statement / quote	Keyword	Code	Sub-theme	DREQ	Theme
"The ground truth would be like take the manually identify anomalies with domain experts" — IN4	*domain experts*	*Reference point validation*	**Difficulties in Evaluation of the anomalies**	*DREQ-18*	**Ground truth validation**
"That you cannot measure timeliness of a data, of course or like the consistency, but may be consistency could be measured, accuracy of anomaly could be measured"- IN7	*consistency and accuracy*	*Reference point validation*	**Difficulties in Evaluation of the anomalies**	*DREQ-18*	

Sub-theme: evaluation quotes from IN4 and IN7 are coded under theme ground truth validation.

A.3 Code for data preparation - step 1

```
import os
from pyspark.sql.functions import col

# Define feb_mar_all
feb_mar_month = spark.table("product_table.raw_datasets").filter(
    (col("Engine_Name") == "VP_ENG") &
    (col("Signal") == "Signal_1") &
    (col("timestamp") >= "2026-02-06 05:00:00") &
    (col("timestamp") <= "2026-03-09 23:00:00")
)

# Select columns and export
export_feb_mar = feb_mar_month.select(
    "timestamp", "Engine_Name", "Signal").orderBy("timestamp")

export_feb_pdf = export_feb_mar_month.toPandas()

# Save to workspace
output_path_feb = "file_path.csv"
export_feb_pdf.to_csv(output_path_feb, index=False)

print(f"CSV exported successfully!")
print(f"Path: {output_path_feb}")
print(f"Records: {len(export_feb_pdf):,}")
print(f"File size: {os.path.getsize(output_path_feb_month) /
↪  (1024*1024):.2f} MB")
print(f"Time range: {export_feb_pdf['timestamp'].min()} to
↪  {export_feb_pdf['timestamp'].max()}")
print(f"Columns: {list(export_feb_pdf.columns)}")
```

A.4 Load the extracted data into data frames - step 2

```
import pandas as pd
import numpy as np
csv_path = "file_path.CSV"
df = pd.read_csv(csv_path, parse_dates=['timestamp'])
print(f"Loaded {len(df)} rows")
print(f"Timestamp range: {df['timestamp'].min()} to
↪  {df['timestamp'].max()}")
```

```
display(df.head())
```

A.5 Predefined threshold check - step 3

```
import matplotlib.pyplot as plt
import matplotlib.dates as mdates

df = df.sort_values('timestamp').reset_index(drop=True)

# === Remove duplicate records based on timestamp ===

df =
↪  df.drop_duplicates(subset=['timestamp']).reset_index(drop=True)

# === Threshold-Based Anomaly Detection: 0-500 kPa ===
THRESHOLD_MIN = 0
THRESHOLD_MAX = 500

df['is_anomaly'] = (df['Measured_Value'] < THRESHOLD_MIN) |
↪  (df[''Measured_Value'] > THRESHOLD_MAX)

anomalies = df[df['is_anomaly']]
normal = df[~df['is_anomaly']]

print(f"Threshold range: [{THRESHOLD_MIN}, {THRESHOLD_MAX}] kPa")
```

A.6 Configuration of IF model - step 4

```
from sklearn.ensemble import IsolationForest

df = df.sort_values('timestamp').reset_index(drop=True)

df =
↪  df.drop_duplicates(subset=['timestamp']).reset_index(drop=True)

#  Features: Measured value +  timestamp features
pdf['hour'] = pdf['timestamp'].dt.hour
pdf['day_of_week'] = pdf['timestamp'].dt.dayofweek
pdf['minute'] = pdf['timestamp'].dt.minute

feature_cols = ['Measure_Value', 'hour', 'day_of_week', 'minute']

X = pdf[feature_cols].values
```

```
n = len(X)

# Isolation Forest model configuration
if_model = IsolationForest(
    n_estimators=100,
    contamination='auto',
    random_state=42
)
if_model.fit(X)
scores = if_model.decision_function(X)

df['if_label'] = if_model.fit_predict(df[feature_cols])
df['if_score'] = if_model.decision_function(df[feature_cols])

candidates = df[df['if_label'] == -1]
print(f"Features: {feature_cols}")
print(f"IF candidates: {len(candidates)}/{len(df)}
↪  ({100*len(candidates)/len(df):.2f}%)")
print(f"IF score range: [{df['if_score'].min():.4f},
↪  {df['if_score'].max():.4f}]")
display(candidates[['Measured_value','if_score']].describe())
```

A.7 Code for using auto-calculated contamination to validate threshold value for Signal_1 - step 5

```
model_scan.fit(X)
scores = model_scan.decision_function(X)
print(f'Scores shape: {scores.shape}')
print(f'Score range: [{scores.min():.6f}, {scores.max():.6f}]')
print(f'Negative scores (potential anomalies): {(scores <
↪  0).sum()}')
print(f'Positive scores (likely normal): {(scores >= 0).sum()}')

sorted_scores = np.sort(scores)

# Search only the bottom 5% of scores (training atleast 10 and
↪  atmost 200 data points)
search_n = int(np.clip(n * 0.05, 10, 200))
bottom_scores = sorted_scores[:search_n]

# Calculate differences between consecutive scores to find jumps
gaps = np.diff(bottom_scores)
```

```
# A gap is "significant" if it's more than 2x the standard
↪  deviation of all gaps
gap_found = len(gaps) > 0 and gaps.max() > np.std(gaps) * 2

if gap_found:
    # Use the position of the largest gap as the anomaly/normal
    ↪  boundary
    gap_pos = np.argmax(gaps)
    score_threshold = bottom_scores[gap_pos]
    n_anomalies = int((scores <= score_threshold).sum())
    auto_contamination = max(n_anomalies / n, 1 / n)
    detection_method = "gap detection"
else:
    # No significant gap found → data has no clear anomalies
    auto_contamination = None
    n_anomalies = 0
    detection_method = "no clear gap found - signal looks clean"

# --- Results ---
print(f"Detection method  : {detection_method}")
print(f"Search window     : bottom {search_n} scores (5% of
↪  {n:,})")
print(f"Max gap size      : {gaps.max():.6f}")
print(f"Gap threshold (2): {np.std(gaps) * 2:.6f}")
if gap_found:
    print(f"Gap position      : index {gap_pos}")
    print(f"Score threshold   : {score_threshold:.6f}")
    print(f"Anomalies found   : {n_anomalies}")
    print(f"Auto contamination: {auto_contamination:.6f}
    ↪  ({auto_contamination*100:.4f}%)")
else:
    print(f"No anomalies detected - all readings appear normal")
```

A.8 Code for retrain the final model to classifies each data points - step 6

```
# ---------- TRAIN FINAL MODEL and CLASSIFY ----------
if auto_contamination is not None:
    model = IsolationForest(contamination=auto_contamination,
    ↪  random_state=42)
    model.fit(X)
    df['anomaly_label'] = model.predict(X)
    df['anomaly_score'] = model.decision_function(X)
```

```
    df['status'] = df['anomaly_label'].map({1: 'NORMAL', -1: 'Data
    ↪  Error Outliers'})
else:
    df['anomaly_label'] = 1
    df['anomaly_score'] = 0.0
    df['status'] = 'NORMAL'

# ---------- SUMMARY ----------
normal_data = df.loc[df['anomaly_label'] == 1, 'ScalarValue']
extreme_data = df.loc[df['anomaly_label'] == -1, 'ScalarValue']

print(f"Normal  range  : {normal_data.min():.2f} →
↪  {normal_data.max():.2f} ({len(normal_data):,} readings)")
if len(extreme_data):
    print(f"Extreme range  : {extreme_data.min():.2f} →
    ↪  {extreme_data.max():.2f} ({len(extreme_data):,} readings)")
    print(f"\n--- ALL EXTREME ANOMALIES ---")
    display(df.loc[df['status'] == 'Data Error Outliers',
    ↪  ['timestamp', 'ScalarValue', 'anomaly_score', 'status']])
else:
    print("No extreme anomalies detected.")
```

A.9 Evaluation Interview Guide Sample

Introduction:

The main focus of this interview is to evaluate the usefulness and applicability of the DDA Method in comparison with the traditional approach to anomaly detection in engine datasets. This interview is conducted as part of the evaluation phase to understand how the proposed DDA Method supports data analysis and engineering workflows within the case company. It aims to gather insights into its effectiveness in identifying and classifying anomalies. The collected responses will include both qualitative feedback and practical observations, which will be analyzed to assess the impact of the proposed data requirements and methodology. With your consent, this interview will be recorded, and all responses will be kept confidential and used solely for research and analysis purposes.

Interview duration: 45-60 minutes per interviewee

Data Preparation

1. How do you perceive the **overall data preparation process** of the proposed DDA Method, including the derived data requirements, DBC files, and sensor data extraction?
2. Do you think the **derived data requirements** support the identification of anomalies and are practical to implement within the current data pipeline?
3. What are your views on using **DBC files** to derive threshold values as part of the DDA Method?
4. Do you consider the **current data extraction approach** (e.g., CSV files) appropriate, or would you recommend alternative approaches for practical deployment?
5. Do you have any suggestions **for improving the data preparation stage** of the proposed DDA Method?

DDA Method

1. How do you perceive the **overall DDA Method in comparison with the traditional approach** currently used for anomaly detection?
2. Do you think the **model configuration** used in the DDA Method is appropriate, or would you recommend any improvements?

3. What are your views on the approach used to **validate the identified anomalies**? Do you suggest any improvements?
4. Do you think incorporating **time context and multi-signal correlation** improves the effectiveness of anomaly detection?

Improvements and Suggestions

1. Do the proposed methods **reduce manual effort** in data validation and cleaning within the current pipeline?
2. Do the defined requirements **align with the domain knowledge** of the engine data?
3. What **challenges** do you foresee in applying these data requirements in the future?
4. What additional data requirements would you suggest?
5. Do you have any **overall suggestions for improving** the research process or recommendations on how it could have been conducted differently?

Bibliography

[1] Marina Araújo et al. "Domain Knowledge in Requirements Engineering: A Systematic Mapping Study". In: *arXiv preprint arXiv:2506.20754* (2025). DOI: 10.48550/arXiv.2506.20754.

[2] Ane Blázquez-García et al. "A review on outlier/anomaly detection in time series data". In: *ACM computing surveys (CSUR)* 54.3 (2021), pp. 1–33.

[3] Michael Franklin Bosu and Stephen G MacDonell. "Data quality in empirical software engineering: a targeted review". In: *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. 2013, pp. 171–176.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Computing Surveys* 41.3 (July 2009), pp. 1–58. DOI: 10.1145/1541880.1541882.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[6] Daniela Cruzes and Tore Dybå. "Recommended Steps for Thematic Synthesis in Software Engineering". In: Oct. 2011, pp. 275–284. DOI: 10.1109/ESEM.2011.36.

[7] Sangeeta Dey and Seok-Won Lee. "A Multi-layered Collaborative Framework for Evidence-driven Data Requirements Engineering for Machine Learning-based Safety-critical Systems". In: June 2023, pp. 1404–1413. DOI: 10.1145/3555776.3577647.

[8] Widad Elouataoui, Saida El Mendili, and Youssef Gahi. "Quality Anomaly Detection Using Predictive Techniques: An Extensive Big Data Quality Framework for Reliable Data Analysis". In: *IEEE Access* 11 (2023), pp. 103306–103318. DOI: 10.1109/ACCESS.2023.3317354.

[9] Abubakar Mastour Adam Fadul. "Anomaly Detection Based on Isolation Forest and Local Outlier Factor". Submitted in partial fulfillment of a structured masters degree at AIMS South Africa. MA thesis. African Institute for Mathematical Sciences (AIMS), South Africa, May 2023.

[10] Abubakar Mastour Adam Fadul. "Anomaly detection based on isolation Forest and local outlier factor". In: *Africa University* (2023).

[11] Harald Foidl et al. “Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers”. In: *Journal of Systems and Software* 207 (2024), p. 111855.

[12] Zeineb Ghrib, Rakia Jaziri, and Rim Romdhane. “Hybrid approach for Anomaly Detection in Time Series Data”. In: July 2020, pp. 1–7. DOI: 10.1109/IJCNN48605.2020.9207013.

[13] K. M. Habibullah et al. “Requirements Engineering for Automotive Perception Systems: An Interview Study”. In: *arXiv preprint arXiv:2302.12155* (Feb. 2023). URL: https://arxiv.org/abs/2302.12155.

[14] Hans-Martin Heyn et al. “Requirement Engineering Challenges for AI-intense Systems Development”. In: *Department of Computer Science and Engineering, Chalmers University of Gothenburg, Sweden / Veoneer Sweden AB* (2026). Corresponding author: Hans-Martin.Heyn@gu.se.

[15] Hun Kang and Kyoungok Kim. “Robust isolation forest using soft sparse random projection and valley emphasis method”. In: *International Journal of Machine Learning and Cybernetics* 17.3 (2026), p. 122.

[16] Abdul Qadir Khan et al. “Knowledge-based anomaly detection: Survey, challenges, and future directions”. In: *Engineering Applications of Artificial Intelligence* 136 (2024), p. 108996. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2024.108996. URL: https://www.sciencedirect.com/science/article/pii/S0952197624011540.

[17] Eric Knauss. “Constructive Master’s Thesis Work in Industry: Guidelines for Applying Design Science Research”. In: *arXiv preprint arXiv:2012.04966* (2020). DOI: 10.48550/arXiv.2012.04966.

[18] Peter Liggesmeyer and Mario Trapp. “Trends in embedded software engineering”. In: *IEEE software* 26.3 (2009), pp. 19–25.

[19] Sachiko Lim, Aron Henriksson, and Jelena Zdravkovic. “Data-Driven Requirements Elicitation: A Systematic Literature Review”. In: *Received: 25 March 2020 / Accepted: 2 December 2020 / Published online: 4 January 2021* (2021). https://www.diva-portal.org/smash/get/diva2:1625117/FULLTEXT01.pdf.

[20] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest”. In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[21] Yang Liu et al. “Temporal Logical Attention Network for Log-Based Anomaly Detection in Distributed Systems”. In: *Sensors* 24.24 (2024), p. 7949. DOI: 10.3390/s24247949.

[22] Lucy Ellen Lwakatare et al. “A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation”. In: *Product-Focused Software Process Improvement*. Springer, 2019. DOI: 10.1007/978-3-030-19034-7_14.

[23] Muhammad Naeem et al. “A Step-by-Step Process of Thematic Analysis to Develop a Conceptual Model in Qualitative Research”. In: *Qualitative Market Research: An International Journal* 26.1 (2023), pp. 123–146. DOI: 10.1108/QMR-02-2022-0033.

[24] Aiswarya Raj et al. “Modelling Data Pipelines”. In: *Proceedings of the 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2020, pp. 13–20. DOI: 10.1109/SEAA51224.2020.00014.

[25] Colin Robson. *Real World Research: A Resource for Social Scientists and Practitioner-Researchers*. 2nd ed. Oxford, UK: Wiley-Blackwell, 2002.

[26] Baškarada Sasa. “Qualitative Case Study Guidelines”. In: *The Qualitative Report* 19.40 (2014), pp. 1–18. DOI: 10.46743/2160-3715/2014.1008. URL: https://doi.org/10.46743/2160-3715/2014.1008.

[27] Ryan E Sperl and Soon M Chung. “Two-step anomaly detection for time series data”. In: *2019 International Conference on Data and Software Engineering (ICoDSE)*. IEEE. 2019, pp. 1–5.

[28] Ramnath Takiar. “The takiar z-test–a better opinion than the T-test for mean comparision among small samples”. In: *Bulletin of Mathematics and Statistics Research* 12.2 (2024), pp. 1–15.

[29] Andreas Vogelsang and Markus Borg. “Requirements Engineering for Machine Learning: Perspectives from Data Scientists”. In: *arXiv preprint arXiv:1908.04674* (2019). DOI: 10.48550/arXiv.1908.04674.

[30] Roel Wieringa. *Design Science Methodology for Information Systems and Software Engineering*. Springer, 2014. DOI: 10.1007/978-3-662-43839-8.

[31] Gemma Wong and Mary Breheny. “Narrative analysis in health psychology: A guide for analysis”. In: *Health Psychology and Behavioral Medicine* 6.1 (2018), pp. 245–261.

[32] Asma Yamani et al. “DRGM: Data Requirement Goal Modeling for ML-Based Systems”. In: *Proceedings of the International Conference on Software and System Processes / Requirements Engineering-related venue*. Author emails: g201906630, g201906430, g201907430, malak.baslyman, jhassine@kfupm.edu.sa. Dhahran, Saudi Arabia, 2024.