ESTABLISHING ANOMALY DETECTION APPROACHES FOR TIME SERIES DATA FROM AUTOMOTIVE ORGANISATIONS

A Thesis in the Field of Business and Management

for the Degree of Doctor in Business and Management

Advisor: Prof. Dr. Alberto Ferraris

Author: Deniel Tichomirov

University of Turin

June 2024

Acknowledgements

I would like to express my deepest gratitude to all those who have supported me throughout the course of my dissertation journey. This work would not have been possible without the guidance, encouragement, and assistance of many individuals.

First and foremost, I am profoundly grateful to my advisors, Prof. Dr. Alberto Ferraris and Prof. Dr. Axel Lamprecht, for their invaluable guidance, constant encouragement, and unwavering support throughout this research. Their expertise, patience, and constructive feedback have been instrumental in shaping this dissertation.

I am particularly grateful to Mercedes-Benz for providing me with the resources, facilities, and an academic environment necessary for conducting this research. My heartfelt thanks go to Roland Gottfried, Patrick Sayer, Markus Boebel, Paolo Pagani, Ezra Tampubolon and Lukas Weber for their logistical and technical support.

Special appreciation goes to my friends, Daniel Albrecht, Jonas Krieger, Dennis Schauermann and Stefan Whittington for their camaraderie, moral support, and for making this journey enjoyable and memorable. Their encouragement and intellectual discussions have been a great source of inspiration.

I am deeply indebted to my family for their endless love, patience, and understanding. My parents, Julia Grinberg and Vladislav Tihomirov have been my pillars of strength, providing me with unwavering support and encouragement. I also would like to mention every relative in my family, Marcel Tihomirov, Asia Grinberg, Gregor Grinberg, Natalia Tihomirov, Igor Lipniak, Christina Roos, Martina Roos, Peter Henn, Elfriede Wallrapp and Wolfgang Wallrapp and thank them for their relentless support.

To my partner and best friend, Antonia Roos, thank you for your patience, love, and continuous support throughout this challenging journey. No matter how hard it was, you always got my back.

To everyone who has contributed to this dissertation in one way or another, I extend my heartfelt thanks and appreciation.

Abstract

The automotive industry is currently experiencing a significant access of data due to the increasing connectivity of vehicles. This data, primarily generated by sensors and other on-board systems, holds the potential to revolutionise engine development, manufacturing, and maintenance processes. However, harnessing this Big Data presents both technical and organisational challenges. This dissertation explores the application of Artificial Intelligence (AI) in compressing, processing, and analyzing large automotive datasets. By leveraging advanced AI algorithms, valuable insights can be extracted from sensor data, providing information on engine performance, durability, and potential issues. Techniques such as anomaly detection, data compression, and time series analysis are particularly useful in identifying patterns and trends that manual analysis might miss. Despite the potential benefits, integrating AI capabilities into the automotive engine development workflow poses significant challenges. The complexity and volume of data, along with the technical expertise required, are major barriers. Additionally, organisational resistance to change and concerns about the reliability and transparency of AI-driven decisions can impede the adoption of these technologies. To address these challenges, this dissertation proposes a comprehensive approach that includes the technical implementation of AI solutions and necessary organisational changes to facilitate acceptance and adoption. This involves providing extensive training on AI and data analytics, demonstrating the value of AI through pilot projects, and effectively communicating the benefits to stakeholders. By overcoming these obstacles, the automotive industry can fully exploit the potential of Big Data and AI, leading to more efficient and reliable engine development processes. This research aims to establish a deep understanding of engine measurement data, categorize it, and develop AI systems to optimize and analyze this data. The goal is to enhance operational use cases and improve organisational acceptance of AI technologies within automotive operations. Key areas of focus include the integration of an automated MLOps pipeline, time series analysis from endurance testing for engine development, and the integration of Anomaly Detection methods to streamline the AI implementation process.

Keywords: anomaly detection, Artificial Intelligence, automotive, Big Data, clustering, data compression, driveability, endurance testing, engine development, hesitations, measurement data, MLOps, pipeline, taxonomy, time series data.

Index

List of Figures

List of Tables

List of Abbreviations

ADLS	Azure Data Lake Storage
AHC	Agglomerative Hierarchical Clustering
AI	Artificial Intelligence
ASAM	Association for Standardisation of Automation and Measuring Systems
CAN	Controller Area Network
CD	Continuous Development
CI	Continuous Integration
CT	Continuous Testing
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DevOps	Development Operations
DL	Deep Learning
DSR	Design Science Research
DTW	Dynamic Time Warping
ECU	Electronic Control Unit
EU	European Union
iO	in order
IQR	Interquartile Range
IT	Information Technology
JSON	JavaScript Object Notation
KDD	Knowledge Discovery in Databases
KPI	Key Performance Indicator
LAN	Local Area Network
LSTM	Long-Short Term Memory
MDF	Measurement Data Format
MIL	Malfunction Indicator Light
ML	Machine Learning
MLFlow	Machine Learning Flow
MLOps	Machine Learning Operations
MSE	Mean Squared Error
niO	not in order
OBD	On-Board Diagnosis
ODS	Open Data Services

OEMs	Original Equipment Manufacturers
PDF	Portable Document Format
PIC	Power Iteration Clustering
PoC	Proof of Concept
PU	Perceived Usefulness
PEOU	Perceived Ease of Use
RNN	Recurrent Neural Networks
RPM	Revolutions Per Minute
SDK	Software Development Kit
TAM	Technology Acceptance Model
TS	Time Series
UDF	User-Defined Function
VEI	Volcanic Eruption Index
WARP	Worldwide Analytics Reporting Platform

1 Introduction

The automotive industry is facing massive amounts of data, driven by the increasing connectivity of vehicles. This data, generated by sensors and other on-board systems, holds immense potential for improving engine development, manufacturing, and maintenance processes. However, effectively harnessing this "big data" presents significant technical and organisational challenges.

1.1 Motivation and Problem Statement

One key area where AI can enable meaningful insights is in the compression, processing, and analysis of large automotive datasets (Luckow et al., 2015). Sensor data from engines can provide valuable information about engine performance, durability, and potential issues (Luckow et al., 2015), (Liang et al., 2019). By applying advanced AI algorithms to this data, engineers can identify patterns, anomalies, and trends that would be difficult to detect manually. This can lead to more efficient engine designs, faster troubleshooting, and proactive maintenance solutions.

Recent advancements in Machine Learning (Bughin et al., 2017) have enabled the development of sophisticated models capable of extracting insights from the massive measurement data generated in the automotive industry. In the automotive context, Machine Learning techniques can be leveraged to extract valuable insights from sensor data generated during engine development and testing (Nghiem et al., 2023). Techniques such as Anomaly Detection and time series analysis can be leveraged to uncover hidden relationships and make accurate predictions about engine behavior (Luckow et al., 2015), (Liang et al., 2019).

However, the successful integration of these AI-powered capabilities into the automotive engine development workflow comes with challenges. The volume and complexity of the data, as well as the technical expertise required to effectively utilize advanced AI algorithms and models, can be significant barriers to adoption. Automotive engineers and data scientists must have the necessary skills to pre-process the data, select appropriate AI techniques, and interpret the insights generated by these systems. Additionally, organisational resistance to change and concerns about the reliability, transparency, and explainability of AI-driven decision-making can hinder the acceptance and deployment of these transformative technologies within automotive operations. Overcoming these technical and organisational obstacles is crucial for the industry to unlock the full potential of Big Data and AI in driving more efficient and reliable engine development processes (Sircar et al., 2021), (Aldoseri et al., 2023), (Luckow et al., 2015).

1.2 Objectives of the Work

To address these challenges, the theory must adopt a comprehensive approach that involves not only the technical implementation of AI-based solutions, but also the changes necessary to foster acceptance and adoption of AI software in organisations. This may involve providing extensive qualitative research on AI and data analytics, demonstrating the business value of AI-driven insights through spot tests and clear communication of the benefits to affected stakeholders. By overcoming these challenges, science and practice in further consideration can unlock the full potential of Big Data and AI, leading to more efficient and profound utilisation of Anomaly Detection in different scientific fields.

Therefore, the establishment of a deep understanding which peculiarities from time series data occur, what these mean and how they can be categorized, how they can be handled with AI systems and how this information can lead to a fast upscaling of the projects based on operational use cases, is needed. This will enable the optimization and compression of time series data, the detection of anomalies in that data and the assessment of the AI system's impact on the organisational acceptance within the operations (Chidambaram et al., 2022), (Sircar et al., 2021), (Arena et al., 2021), (Zhu and Zhang, 2021).

Chapter 2 provides the theoretical foundation for the Technology Acceptance Model (TAM), which posits that individuals' decisions on how and when to use a new technology are influenced by various factors. Specifically, the perceived usefulness and perceived ease of use of the technology serve as the starting point for anomaly detection in time series data, as these factors shape users' impressions and adoption of the software. The chapter also discusses the integration of anomaly detection into the broader field of AI, establishing the connection between the theoretical model and its practical implementation. Chapter 3 presents the design-oriented research approach, a common method for simultaneously addressing practical problems and advancing theoretical knowledge. The chapter explains the 6 principles that form the foundation of design-oriented research and the development of individual artifacts. Furthermore, it explores the application of these principles to the practical problem of anomaly detection in time series data from endurance testing in engine development, focusing on longitudinal acceleration events in vehicles. Chapter 4 introduces the published articles included in the work. The first article aims to enrich the understanding of the term "anomaly" and related concepts, addressing the confusion and mixing of these terms in both literature and practice. It proposes a taxonomy that can be used in scientific classification of anomalies and for analyzing time series data in the automotive industry. The second article then explains the core algorithm for unsupervised learning of unmarked time series data, emphasizing data preparation and the identification of essential features. The artifact serves as a method for reducing large amounts of time series data for engine development, with the potential for broader adaptation to similar time series data. Finally, the third article defines and implements a framework for the algorithm, focusing on making it adaptable and scalable for efficient deployment in production environments. This framework involves the adoption of the MLOps approach, a modification of the DevOps approach, and the algorithm is further modified for supervised learning of labeled time series data. Chapter 5 concludes the work by discussing the theoretical and practical relevance, potential business impacts, and an outlook for subsequent research.

Because of this urgent demand for robust and flexible AI solutions, the thesis elaborates on the theoretical approach of Anomaly Detection in time series data and takes endurance testing measurement data from an automotive manufacturer to emphasize the business need for automotive organisations (Braei and Wagner, 2020).

2 Theoretical Background

2.1 Technology Acceptance Model (TAM)

The Technology Acceptance Model (TAM) is a widely used theoretical framework in the field of information systems research that aims to explain and predict user acceptance of new technologies (Erba et al., 2019), (Chakraborty et al., 2023), (Oliveira et al., 2021), (Capogrosso et al., 2023). The model assumes that an individual's intention to use a technology is determined by two key factors: perceived usefulness (PU) and perceived ease of use (PEOU).

PU refers to the degree to which a person believes that using a particular system or technology will enhance their job performance and improve their productivity. Numerous studies have shown that PU has a strong impact on attitudes towards the use of IT systems. When users believe that a technology is useful, they are more likely to accept and regularly use it. For example, Venkatesh and Davis (2000) found that PU has a significant influence on the use of ERP systems. Examining the acceptance of medical technology, Holden and Karsh (2010) found that when healthcare personnel recognize the usefulness of electronic health records and telemedicine services, they are more inclined to integrate them into their daily practice.

Studies have shown that the user-friendliness of software and systems in companies also has a significant impact on their acceptance and use, the PEOU. For example, the user-friendliness of online shopping platforms and digital payment systems plays a crucial role in user acceptance. User-friendliness can also influence PU. If a system is user-friendly, it can lead users to perceive it as more useful because they experience less effort and frustration during use. Increased PU can positively influence usage intention and actual usage behavior (Davis, 1993), (Morris & Dillon, 1997).

All in all, PU refers to the degree to which a person believes that using a particular system or technology will enhance their job performance, while PEOU refers to the degree to which a person believes that using a particular system or technology will be free of effort (Fernández-Robin et al., 2019), (Sargolzaei, 2017). The more useful the technology is perceived to be, the higher the likelihood that it will be accepted and used. These two factors are theorized to influence an individual's attitude towards using the technology, which in turn influences their intention to use it (Taherdoost, 2019), (Sargolzaei, 2017).

2.2 Anomaly Detection

The field of Artificial Intelligence (AI) has gone through a transformative journey, with each step unlocking new possibilities and redefining the understanding of intelligent systems. At the core of this progression lies the fundamental discipline of Machine Learning, which has evolved from simple linear models to the powerful and versatile deep neural networks (Zhang et al., 2023), (Tan and Lim, 2018).

Neural Networks, inspired by the human brain, have been instrumental in driving the growth of Deep Learning, a subfield of Machine Learning that stands out at extracting complex hierarchical patterns from data (Stuchi et al., 2020). Deep Learning architectures, with their multi-layered structure, have revolutionized domains such as computer vision, speech recognition, and natural language processing, enabling machines to perceive and understand the world in more detailed ways (Zhang et al., 2023), (Haensch et al., 2022).

Recent progress in parallel computing, optimization techniques, and the availability of large datasets have further pushed the field of Deep Learning, making it a dominant force in Artificial

Intelligence (Haensch et al., 2022), (Ruff et al., 2021). One of the key factors within this landscape is the application of Deep Learning to the domain of Anomaly Detection, where the goal is to identify patterns that deviate from the norm within high-dimensional datasets (Ruff et al., 2021).

Anomaly Detection has become increasingly important in a wide range of applications, from predictive maintenance in industrial settings to fraud detection in finance, and even to the analysis of time series data in the automotive industry (Braei and Wagner, 2020). The inherent challenge in Anomaly Detection lies in the rarity of anomalous events, making it difficult to obtain representative training data (Choi et al., 2021).

Traditional statistical and Machine Learning approaches have had limited success in tackling this challenge, as they often rely on the assumption of a known data distribution or require extensive feature engineering (Braei and Wagner, 2020).

Machine Learning techniques, such as one-class classification and clustering-based methods, have shown promise in identifying anomalies in structured data (Nassif et al., 2021). However, as the complexity of data increases, the limitations of these classical approaches become more obvious (Ruff et al., 2021). Deep Learning, with its ability to learn rich, hierarchical representations from raw data, has emerged as a powerful tool for Anomaly Detection, particularly in the realm of Time Series data, where it has demonstrated remarkable performance (Zhang et al., 2019), (Ruff et al., 2021). Researchers have explored various Deep Learning architectures, such as Autoencoders, Generative Adversarial Networks, and Recurrent Neural Networks, to tackle the challenges of Anomaly Detection in multivariate time series data (Jeong et al., 2023). These approaches consider the temporal characteristics of the data and successfully adapt deep neural networks to the field of time series analysis.

As industries become increasingly automated and connected, the need for robust and scalable Anomaly Detection solutions has become paramount (Braei and Wagner, 2020). Deep Learning-based methods have shown great promise in this area, offering a pathway to unlock the full potential of Artificial Intelligence in addressing complex real-world challenges (Choi et al., 2021).

2.3 Reference to Current Work

The flexibility of the Technology Acceptance Model is evident in its application across a diverse range of domains, including agriculture, construction, urban planning, voting, dieting, family planning, donating blood, women's occupational orientations, breast cancer examination, choice of transport mode, turnover, using birth control pills, education, consumer's purchase behaviors, and computer usage (Sargolzaei, 2017). This highlights the model's potential to provide valuable insights into the user acceptance of emerging technologies, such as Anomaly Detection in time series data within the context of automotive engine development.

By applying the Technology Acceptance Model to the use of Anomaly Detection techniques in automotive engine development, researchers can gain a better understanding of the factors that influence the acceptance and adoption of these technologies by engineers and technicians. This knowledge can inform the design, evaluation, and implementation of such technologies, finally facilitating their successful integration into the engine development process (Taherdoost, 2019), (Sargolzaei, 2017), (Sengaji and Radiansyah, 2022), (Liu, 2009), (Fernández-Robin et al., 2019).

Through expert interviews, the features, and characteristics of the first AI-based prototype are examined. By applying the theoretical model, requirements for the implementation of the Machine Learning model detecting potential anomalies in time series data from endurance testing of vehicle engines within an automated pipeline are formulated.

The further method is involving the future users in the development process through iterative design and prototyping. This allows to receive early feedback from users. They can also test further prototypes and provide their opinions on their usefulness. This feedback can then be used to improve and adjust the product. This is enabled by agile methods as Scrum and Kanban to gain the users' continuous feedback. Through short development cycles and regular review meetings, the team ensures that the Machine Learning model within the automated pipeline is continuously improved and adapted to the users' needs.

In this case, the Technology Acceptance Model is merely used to apply a systematic theoretical model to the current research project. The factors perceived usefulness and perceived ease of use can subsequently be examined in further research through empirical investigations.

3 Methodology of Design Science Research

3.1 Design Science Research (DSR)

Design Science is a research approach that focuses on the creation and evaluation of IT artifacts such as software or AI solutions. The research questions and the developed artifacts should address real, relevant problems. The results should be applicable and useful in practice. This approach combines technological innovation with scientific methodology to develop useful and effective solutions. The focus is on creating innovative solutions that go beyond existing approaches and create new possibilities. The research should also generate new knowledge that extends beyond the specific solution and includes general principles. The Design Science approach, on the other hand, provides a systematic and rigorous framework for the development and evaluation of IT artifacts. By combining practical relevance with scientific rigor, this approach helps create effective solutions for real-world problems while simultaneously expanding theoretical knowledge (Ahmed et al., 2020), (Hevner et al., 2004), (Peffers et al., 2020), (Oesterle et al., 2010).

The Design Science Research is characterized by the development and evaluation of artifacts, which can take the form of constructs, models, methods, or instantiations (Cross, 1993).

According to this methodology, the artifacts developed should follow to the stated key principles:

- Abstraction: The artifact must be designed in a flexible and adaptable manner, enabling it to be applicable and extensible to a broader class of problems beyond the specific use case. This allows the artifact to be reused and applied to a wider range of contexts, enhancing its overall utility (Oesterle et al., 2010).
- Originality: The artifact must exhibit a novel and innovative approach that goes beyond existing solutions and advances the state-of-the-art. The artifact should introduce new ideas, techniques, or capabilities that push the boundaries of current knowledge and practice (Oesterle et al., 2010).
- Justification: The development of the artifact must be grounded in and supported by both practical real-world applications as well as relevant scholarly research and theories. This ensures that the artifact is based on a solid foundation of empirical evidence and theoretical understanding (Oesterle et al., 2010).
- Utility: The artifact must be able to generate tangible value and benefits for the target stakeholders and users, addressing their specific needs and pain points. The artifact should demonstrate its ability to effectively solve the problem it is designed for (Oesterle et al., 2010).
- Relevance: The problem being addressed by the artifact must be clearly justified as being relevant and significant to the practical domain and/or the academic research community. The artifact's development should be driven by the importance and impact of the problem it aims to solve (Oesterle et al., 2010).
- Rigor: The methods and processes employed in the design, development, and evaluation of the artifact must adhere to scientific principles and methodologies, ensuring a systematic and rigorous approach. This includes the use of appropriate research methods, data collection techniques, and analytical procedures to ensure the validity and reliability of the artifact (Oesterle et al., 2010).

Furthermore, the Design Science Research process typically involves four key activities: problem identification and motivation, defining solution objectives, design and development, and evaluation.

The analysis phase describes the problem statement, formulates the research questions, and determines the type of artifact to be designed. To this end, problem-solving approaches in science and, if available, in practice are analyzed. Based on this, a research plan and relevant research methods are defined. In the analysis phase, in-depth interviews with experts are particularly conducted (Oesterle et al., 2010).

In the design phase, the artifact to be developed must be clearly defined and its objectives justified. The development process should utilize recognized methodologies and tools, such as the construction of prototypes, to ensure a systematic and rigorous approach (Oesterle et al., 2010).

During the evaluation phase, the created artifacts are verified against the initially defined goals and requirements, and an assessment is carried out using appropriate evaluation methods. This can be done through the application and testing of the prototype, as well as through review by subject matter experts (Oesterle et al., 2010).

The dissemination of the research results to the relevant stakeholders must be explicitly planned, for example, through the implementation and deployment of the artifacts in companies or the presentation of the findings at academic conferences. This will contribute to the scientific discourse and enable the broader adoption and impact of the developed solutions (Oesterle et al., 2010).

A typical current example of a design-oriented research problem would be the use of AI technologies to automate and improve processes. There are two main challenges: the development of functional and useful IT artifacts requires interdisciplinary knowledge and intensive collaboration between research and practice and evaluating the artifacts in various contexts and ensuring the generalizability of the results presents another challenge (Hevner et al., 2004).

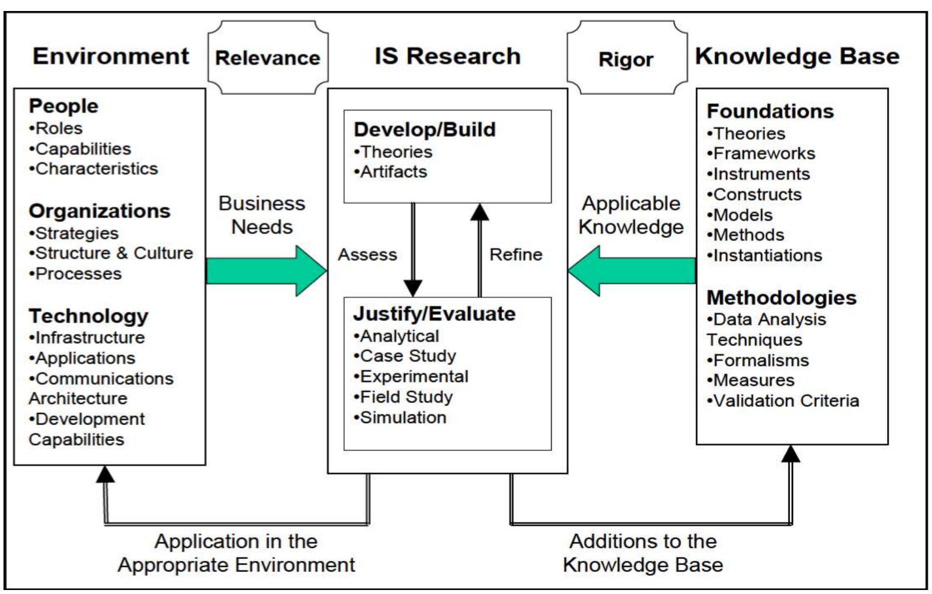


Figure 1. Research framework Design Science (Hevner et al., 2004)

The requirements for the artifact are defined by people within the context of organisational structures and business processes, taking the technological infrastructure into account. The developed artifact should be implementable in this environment. Together, these business needs define the research problem. The knowledge base provides scientifically established methods, models, hardware, and software techniques that are relevant to the research project (Hevner et al., 2004). Moreover, the approach has been extended to six phases for the development of information systems: the demonstration phase which shows the deployment of the artifact in a real or simulated environment to demonstrate its functionality and the communication phase which documents and publishes the research results (Peffers et al., 2020). Additionally, linearity was replaced by a comprehensive iterative approach that allows for regressions into previous phases.

Meanwhile, the Design Science approach is also being used in research projects in the field of Machine Learning (Duque et al., 2024), (Muntean and Militaru, 2022), (Podder and Bub, 2023), (Mitchell, 2023). Muntean and Militaru (2022) propose four central activities: Feature Selection, Clustering, Classification, and the application of Evaluation Metrics. These activities are not viewed in isolation but are part of an iterative development process spanning multiple phases. This iterative approach ensures that each phase is carefully refined and aligned with one another to optimize the final machine learning artifact (ML artifact). This iterative process is repeated until the final ML artifact is developed, meeting the project's requirements and specifications. This systematic and iterative approach ensures that the product is not only technically sound but also practically relevant and applicable.

Therefore, this thesis adapts the application of Design Science Research to the current research problem where a MLOps pipeline with an Anomaly Detection algorithm for time series data in the engine development process of automotive manufacturers is implemented (Amorim et al., 2021), (Strode and Chard, 2014), (Hlongwane and Grobbelaar, 2022).

Considering PEOU in software development is crucial for the acceptance of software. The DevOps approach, with its principles and practices, can significantly contribute to improving the user-friendliness of software and IT systems. DevOps, a combination of "Development" and "Operations," promotes close collaboration and integration between developers and IT operations teams. This close collaboration, continuous integration and deployment, rapid response to user feedback, automated testing, and continuous monitoring are all factors that can help make systems more user-friendly, stable, and adaptable. By implementing DevOps practices, organisations can ensure that they deliver not only functional but also user-friendly systems. DevOps promotes the continuous collection and analysis of user feedback. This feedback can be directly incorporated into the development process, allowing developers to address usability issues more quickly (Vuppalapati et al., 2020).

For Machine Learning systems, the Machine Learning Operations (MLOps) approach can significantly enhance user-friendliness. MLOps, which adapts DevOps principles to machine learning systems, aims to optimize the development, deployment, and maintenance of machine learning models (Kreuzberger et al., 2022).

The MLOps approach was specifically used in the developed anomaly detection system to enhance user-friendliness through automated workflows, rapid updates, scalability, and monitoring (Kreuzberger et al., 2022).

3.2 Reference to Current Work

In the context of the research project, the problem to be addressed is the need for an efficient and scalable approach to implementing Anomaly Detection in the engine development process of automotive manufacturers. The objective is to develop a MLOps pipeline that can effectively handle time series data and provide accurate Anomaly Detection capabilities (Hlongwane and Grobbelaar, 2022), (Cross, 1993).

In the analysis phase, the goal is to understand the level of knowledge that future users of the system have regarding the concept of anomalies. Both in science and practice, the different manifestations of the anomaly concept are often confused, leading to misunderstandings in the implementation of Anomaly Detection use cases. Users mix up terminology and use it incorrectly, resulting in algorithms not being implemented according to requirements (Carreno et al., 2020), (Tichomirov et al., 2023). Therefore, an anomaly taxonomy will be developed, aiming to clarify the understanding of different anomaly manifestations (Carreno et al., 2020), (Tichomirov et al., 2023). This taxonomy will be established through a literature review in 4.1.

Furthermore, in preparation for the design and development phase, functional and non-functional requirements will be gathered through expert interviews. The functional requirements aim to find technical synergies between the Machine Learning algorithm and vehicle and engine-specific functionalities. This relates to the perceived usefulness of the Technology Acceptance Model. The higher the perceived usefulness by users, the more likely they are to see the benefit of the developed AI system. The non-functional requirements focus on the user interface and the overall usability of the AI system. Here, a connection is drawn to the perceived ease of use of the Technology Acceptance Model. The higher the perceived ease of use by users, the more likely the AI system will be used in everyday life, as the barrier to understanding how to use it is low or nonexistent (Taherdoost, 2019), (Sargolzaei, 2017).

The design and development phase of the research involves the creation of the MLOps pipeline and its Machine Learning model for Anomaly Detection, including the selection and integration of appropriate tools and technologies for data ingestion, model training, deployment, and monitoring. The pipeline would be designed to seamlessly handle the complexities of time series data and enable the integration of an Anomaly Detection algorithm (Hlongwane and Grobbelaar, 2022).

On one hand, the design and development phase include the implementation of two artifacts: the Machine Learning model for Anomaly Detection in the time series data of endurance testing, and the MLOps pipeline for scaling the use case, versioning the models, and reporting the technical results produced by the algorithm. Both open-source and proprietary modules are used for implementing the Machine Learning model. A proprietary software development kit (SDK) designed to analyze vehicle measurements from time series data was used for processing the time series data. All other modules for implementing the Machine Learning model for Anomaly Detection were realized using open-source modules such as NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, and others. The time series data and algorithms were processed on an internal company cloud instance to enable distributed computing, thus adhering to the concept of Big Data, as detailed in 4.2. The MLOps pipeline was also set up under these circumstances on the cloud instance. The MLOps pipeline provides the framework for the Anomaly Detection model and uses appropriate modules and cloud resources. An MLFlow logging module supports the versioning of the developed Machine Learning models. The basic setup of the MLOps pipeline is described in 4.3.

On the other hand, the design and development phase thrive on active engagement with future users to qualify the algorithm and the pipeline according to data or vehicle-specific requirements. This takes place within agile sprints on a weekly basis, where data, engine, and project experts come together to accompany the development of the artifacts.

The demonstration phase involves the implementation of the MLOps pipeline in a real-world automotive engine development context, showcasing its capabilities in processing time series data and detecting anomalies. An investigation of anomalies in the new diesel engines with Euro 7 emission standards was conducted. It was crucial to see whether the Machine Learning model, based on the measurement data of older diesel engines with Euro 6 emission standards, could detect new anomalies. The resulting report should then provide a list of potential anomalies for detailed analysis by a panel of experts.

The evaluation phase assesses the performance, scalability, and overall effectiveness of the MLOps pipeline in the target application. Metrics such as accuracy and the validation loss (MSE) are used to evaluate the Anomaly Detection algorithm, while measures of pipeline efficiency, robustness, and ease of deployment are considered. Additionally, a manual was developed to simplify the use of the MLOps pipeline, which is also intended to increase the perceived ease of use according to the Technology Acceptance Model.

The communication phase involves the dissemination of the research findings, including the design, development, and evaluation of the MLOps pipeline, to the broader academic and industry communities. Both contributions within the organisation were created and published, and workshops on the use of the MLOps pipeline were conducted within a Big Data community in research and development. Additionally, the developed artifacts were published beyond the engine department, allowing for evaluation with experts from other fields (see **Fehler! Verweisquelle konnte nicht gefunden werden.**). Furthermore, the artifacts were published in the form of scientific publications to contribute to the scientific community. Here, the focus was on the scientific insights gained from the developed artifacts, as seen in 5.2.

4 Artifacts of the Design Science Research

As described, the artifacts were published in the form of scientific publications. In the following chapter, they will therefore be explained both technically and scientifically in a well-founded manner. Everything that was described in 3.2 will be discussed here in detail once again.

4.1 Article 1: Anomaly Taxonomy

4.1.1 Introduction

The beginning of the digital transformation did not only consider abnormal behavior. Even in the years before, developers and engineers were concerned with the search for errors and anomalies in measurement data that is stored by the sensors to control units of various storage media. It should be noted that over time, the degree of interconnectivity in vehicles has increased between the various Electronic Control Units (ECUs), sensors, and actuators in modern vehicles (Burkacky et al., 2019), (Jablonski, 2020). Thus, the product life cycle of vehicles requires monitoring of these systems to prevent hardware and software faults that may turn into anomalous behaviour (Jablonski, 2020). In addition, competition among Original Equipment Manufacturers (OEMs) continues to drive the networking of in-vehicle systems to improve the quality of driving for customers and ensure their satisfaction.

The paper emphasizes anomalies that occur in the development phase of Mercedes-Benz Cars engines among endurance run time series data. Research on several cases where Anomaly Detection is applied in a company related context, can be found in Syed and Guttag (2011), Alexandrov et al. (2020) and Oentaryo et al. (2014).

First, the paper stresses peculiar behaviour of physical quantities from different vehicle sensors, which may occur as anomalies. This means that sensors might measure unexpected values compared to the measures, which are expected due to the construction of the vehicle and its engine. Second, the paper assesses a taxonomical approach to classify unexpected events in time series data properly. The taxonomy should then be used to achieve a learning effect from unknown peculiar events, which might emerge as anomalies, novelties, or rarities. Carreno et al. (2020), Avizienis et al. (2004), and Dou et al. (2019) have published preparatory work that outlines the points used as the base for the paper's research.

An anomaly defines an unusual, deviant behaviour that does not conform to expected norms in the literature. Although anomalies are rare, they represent an important phenomenon both in theory and in practice. For this reason, the detection of anomalies attracts a lot of attention (Abdelrahman and Keikhosrokiani, 2020), (Liu and Nielsen, 2016), (Chahla et al., 2019). Anomalies arise in many areas and numerous applications require the recognition and detection of abnormal behaviour in data. Intruders (Ribeiro et al., 2016), (Pimentil et al., 2020), (Luca et al., 2016), (Phua et al., 2010), (Yeung and Ding, 2003) in the security industry, road accidents (Theofilatos et al., 2016) in traffic data, volcanic eruptions (Dzierma and Wehrmann, 2010) in geology, corporate bankruptcies (Fan et al., 2017) in business, or stimuli hitherto unknown in neuroscience (Kafkas and Montaldi, 2018) are events that take place as abnormal behaviours. These terms are referred to as rare event, anomaly, novelty, and outlier (Carreno et al., 2020).

Various authors researched the designations for different irregularities in recent years from the examples given. However, the terminology has changed and evolved so that different terms represent similar problems over time. Additionally, there has been confusion between the respective terms due to the different application areas (Carreno et al., 2020). In engine development, these terms also need to be specified precisely to better understand the anomalies' characteristics that have occurred.

In the engine development at Mercedes-Benz, the use of the measured data from the endurance run is crucial. The business unit responsible for the endurance run takes sensor measurement data and embeds the generated data into an IT infrastructure so that the different development groups working with this data can handle their use cases. Among other things, the endurance run serves as a data supplier for the employees in engine development, enabling them to build technically error-free engines that meet the needs of the customers. Due to the large data volumes in the order of several petabytes, the engine experts find it difficult to process the time series data without tools, web applications, and conformable technical aids. Using firmly defined training data that represents anomalies, data models can be developed to ensure error-free engine development. The company follows the Big Data Analytics approach to exploit the full potential of the existing measurement data.

Therefore, we motivate the topic around Anomaly Detection within measurement data and perform the example of hesitations. In chapter three, we incorporate anomalies from the endurance run measurement data, especially time series data, in the field of Big Data Analytics by presenting a taxonomical structure. The background to this is that both theory and practice have a mixture of different manifestations of anomalies (Carreno et al., 2020). With the help of a systematic classification for time series data in the automotive environment, we can classify events in the future. On the one hand, we can achieve a learning effect that will help people understand the meaning of anomalous events and differentiate them from one another. On the other hand, we can use the systematic classification to adopt the right combination of different Big Data ML technologies according to their contextual needs and specific applications' requirements, such as the usage of supervised, unsupervised, or reinforcement learning. Both effects follow the paradigm of Big Data ML to discover knowledge and make intelligent decisions (Oussous et al., 2018). We apply the taxonomy to the hesitation use case in chapter four to establish a foundation for further Anomaly Detection approaches of this event by using Big Data ML methods. In this case, the taxonomical approach gives insights about the ML model regarding the hesitation event. Finally, we provide the results and a further outlook.

To delve into the methodology of the paper in more detail, we first analyzed the actual state of the business unit by conducting interviews with experts from the engine development field. We found that this area is divided into three different trades: development, series car care, and endurance run testing. Each of these three areas uses its own data source to search for anomalies in the time series data. The series car care team uses data that reproduces a complaint in the customer's vehicle. As soon as the developers reproduce the fault, they make assumptions about the root cause. This means that there is only a small data set for a specific manifestation of a complaint, which is not sufficient to train a model. During development, the team uses data that is also measured on test drives for individual cases. The data basis here is larger, but it does not cover the complete driving dynamics of a vehicle. However, with endurance run data, we have both a large database and a distinct variety of engine characteristics, route profiles, time periods that cover the driving dynamics of the vehicles. In addition, engine developers use most of this data to develop driving dynamics for new engines based on the curves of individual measurement channels. Therefore, we decided in favor of the endurance run database.

Since all vehicle components are highly interconnected and there are many interdependencies between the components, we also questioned which anomalies have certain characteristics. Various components, such as a gear change, insufficient fuel injection or an uneven road, can cause bucking in a vehicle. Hesitations can have similar characteristics. These characteristics often lead to problems in understanding and classifying these events in engine development. Therefore, the research question of the paper revolves around how to prevent misleading information in the field of Anomaly Detection from time series data to improve engine development.

To circumvent misleading information flows, we decided to use a grounded-theory approach. We developed a taxonomy that classifies individual events into a systematic context based on predefined characteristics. This should help to better understand individual characteristics of events in the time series data of engine development. In this paper, we elaborate on this general theory to apply it to an application case, the hesitation.

The main contributions of our paper are,

1. The elaboration of a taxonomy for anomalies from time series data for the use of AI-based data models.

2. The taxonomical classification of hesitations from measurement data of the endurance run for engine development.

4.1.2 Motivation

As analytics techniques become more complex, particularly for smart sensors and edge devices in industrial systems, Big Data analytics and AI enable the analysis of large amounts of data (Hof, 2013). This permits various tasks such as resource management, prediction of performance metrics, and the detection of anomalies and faults (Shah et al., 2019). In this case, engine development considers the perspective of anomalies related to the vehicles' driveability. Due to the sophisticated degree of interconnectivity in the vehicles, numerous and relevant aspects constantly increase (Burkacky et al., 2019), (Jablonski, 2020). Hence, OEMs aggregate solutions for the engine experts by improving the user experience and creating significant and rapid reporting to encourage their comprehension (Shams et al., 2022), (Bikakis et al., 2018).

Engineers conceive the peculiarities for anomalies in time series data at first to detect anomalies during the development phase of engines. In this case, they employ endurance run measurement data from Mercedes-Benz that underlays the time domain. This data is selected for our research because it is used directly for the development of new engines. They measure the data based on different test tracks and areas of application, engine setups, engine outputs, drive variants, and test subjects. By covering these different cases, engineers can simulate the driving behavior of the customers. This approach considers all kinds of driving behavior for the development of new engines. From time to time, there may be different indicators that engineers cannot measure directly, such as the driving behavior of the driver during the endurance run testing. An individual driving style or possibly insufficient experience can lead to conspicuous behavior of the vehicle. However, engineers should not incorporate this behavior into the new engines. Therefore, a preliminary investigation using an Anomaly Detection model is unavoidable. Engineers carry out the collection of endurance run time series data via the built-in sensors and actuators in the vehicle, which are connected to the ECU of the vehicle. The ECU and the associated measurement system store the data in the form of time series, from which the waveforms of the approximately 10,000 measurement channels generate. Each individual measuring channel quantifies data at sampling frequencies of one to 200 Hertz, depending on the requirements of the business unit and the specific application of a use case. This produces large volumes of data, which is aggravating for humans to recognize coherences between several measurement channels with the naked eye at once.

To define anomalies from a business context of the engine development, engineers abstract conventional powertrain systems, such as diesel and gasoline engines, as well as novel technologies like the electric engine or engines powered by the fuel cell. Engineers align endurance run time series data from vehicles within diesel powertrains since electric engines are known to have a short acceleration delay due to the lack of clutch and shift times and low moments of inertia. The maximum available torque produced by electric engines is almost entirely available during acceleration from standstill (Bharadwaj, 2021), (Jones, 2022). Since Shams and Solima (2019) demonstrated that Big Data Analytics has inexhaustible potential, (Toonders, 2014), (Andrade, 2022), (Bhageshpur, 2019), engineers establish a precise delineation for anomalies in time series data using hesitations as an example to prevent mixing-up and misunderstanding terms from a business context's view. Shams and Solima (2019) also stated that authors have limited understanding in the context of Big Data. Thus, it is challenging to exploit the full potential of it. At this point, engineers must build a basis to use Big Data methods properly.

The term "hesitation" defines an acceleration delay of vehicles in the following research. Both in development and in the customer environment, a delay in the acceleration of a vehicle describes hesitation. Mercedes-Benz specifically uses this term, but it could have various names in different companies and industries. Shams et al. (2022) also state that a lack of common terms in data management makes it harder for users to benefit from the analyses.

Hesitation has two different types. On the one hand, delays when vehicles accelerate from a standstill can be defined as hesitations. On the other hand, when vehicles accelerate when they are already in motion and have picked up speed, hesitations occur. In the history of anomalous events at Mercedes-Benz, there have been isolated cases of hesitations, among others. Customers felt a drop in acceleration when they depressed the accelerator pedal while exiting a traffic circle, which is not expected, especially for vehicles with high-performance powertrains. Normally, vehicles also respond to drivers' accelerator pedal requests in a relatively short time, depending on the drive mode set. A fast acceleration in sports mode and a lower acceleration in economy or comfort mode can then be expected. Although the event causes discomfort to customers, it does not fall under a warranty and goodwill case because of its not-safety-related characteristic but a comfort-related one. Nevertheless, the engine experts at Mercedes-Benz driveability classified this behavior as an anomaly that should not be neglected during the development phase. Accordingly, identifying such events already during the development of new engines is the goal to prevent them in the customer environment. The paper's focus lays on the generic classification of anomalies in time series data by a taxonomical approach. Comparing former papers from Syed and Guttag (2011), Alexandrov et al. (2020), and Oentaryo et al. (2014), the classification of hesitations is deduced from a generic classification of anomalies in time series data by a taxonomical approach.

4.1.3 Taxonomical Approach Defining Anomalies in Time Series

We aim to classify anomalies from time series data content-wise in a uniform classification scheme, a taxonomy, since different interpretations of anomalies exist. We investigate the classification of anomalies from time series data in the automotive environment of passenger cars. Furthermore, we consider anomalies from time series data as symptoms of subsequent faults and damages to distinguish between current terms and their meanings. In his article, Carreno et al. (2020) emphasized the inevitability of standardizing different terms.

We use the top-down principle to perform the classification, deducing the specific case for the categorization of hesitations in time series data from the general theory of anomalies. To achieve this, we combine characteristics from IT, statistics, and mechanics to establish a reference to anomalies within the automotive context.

The research points are structured as follows in this section:

4.1.3.1 Exploration of Entities,

4.1.3.2 Placement in Systemic Context,

4.1.3.3 Placement in Environment of Abnormal Events,

4.1.3.4 Placement in Types of Anomalies,

4.1.3.5 Cardinality of Anomalies,

4.1.3.6 Dynamics of Anomalies.

4.1.3.1 Exploration of Entities

First, the paper considers to which types of input variables anomalies can be assigned. It starts from two data sources that are used among OEMs in the development. These are divided into customer data and measurement data. Since the target of the engine development phase is development data, customer data is neglected.

Measurement data are collections of measurements of physical or calculated quantities by sensors, which are recorded over a closed period. They are called time series. Time series are isolated data points that are recorded at regular points in time. The vehicle's ECUs generate measurement files with the help of predefined experiment files. These experiments define parameters like the sampling frequency, used software versions, relevant signal channels and more. The user can load the experiment files with the help of a local area network interface (LAN) into the vehicle from a computer.

To get a feel for the measured information, let us look at a scenario where a thousand channels of vehicles log measurement data at a sampling frequency of 100 hertz. Hertz is a unit named by Heinrich Rudolf Hertz, which describes how much data points per second are measured. This means that 100 data points per second are logged with one measurement channel. Conversely, this means recording 100,000 data points per second for all channels. If one data point occupies a storage volume of one byte, this results in a total of 100,000 bytes, 1,000 kilobytes, or one megabyte per second. Extrapolated to 24 hours, the data storage volume reaches a total of 86,400 megabytes or 86.4 gigabytes for a single vehicle. Typically, measurement files are recorded for entire fleets of vehicles. The number of vehicle fleets can vary and corresponds to a multiple of 86.4 gigabytes of measured data. Therefore, it is mandatory to reduce the amount of data to certain measurement channels due to the high storage and computation costs.

Time series data also has other properties according to their type. Two of these characteristics describe signal-based and the message-based in time series data. Signal-based time series data continuously provides signals, such as sound signals, based on vibrations caused by the engine. The engine development operates with sensor signals that the in-built ECUs deposit in their memory storage. Message-based time series data transmit independent units that contain complete pieces of information, such as data packets in a vehicle network operating with CAN interface. In short, signal-based time series data is continuous and message-based time series data is discrete in case of labouring with endurance run data. The manifestations of signal-based and message-based time series data are time series data in which the statistical properties, such as mean and variance, vary over time. These are called non-stationary time series data (Nason, 2006). Stationary time series data remain constant in mean and variance over time (Nason, 2006).

4.1.3.2 Placement in Systemic Context

Anomalies occur as brief events in time series in the development phase. The more difficult it is to spot these moments with the naked eye, still less to perceive them while driving a vehicle. Exemplary in this case are highly scaled IT systems, such as cloud platforms, where it is increasingly difficult to monitor relevant metrics or abnormal system behavior. Extensive systems with a high number of linked components, as is the case of a vehicle, run the risk of being out of order. If a fault occurs in one component, it can affect other components of the IT system and cause the entire system to fail (Oppenheimer et al., 2003). As Avizienis et al. (2004) and Dou et al. (2019) have already shown, the events are differentiated into three types concerning the anomalies' categorization.

- Fault: A fault is the root-cause of a failure. Normally, a system should be designed in a way that the fault remains inactive and does not cause an error. However, when it becomes active, it contributes to forcing the entire system into errors and following failures.
- Error: An error is the event that is different from the normal behavior of the system and represents abnormal behavior. Unlike a failure, an error is difficult to observe.
- Failure: A failure defines an event that does not deliver the actual purpose of the system to the user. The user can observe this event, unlike an error.

4.1.3.3 Placement in Environment of Abnormal Events

The determination of the anomaly types is based on the classification in the systemic context, which can be found in different papers. Since these terms are clearly delineated in parts of the existing literature, comprehension problems arise when dealing with Anomaly Detection problems (Carreno et al., 2020). Therefore, the anomalies are distinguished into the following four terms.

- Rare event: In many articles, rare events are referred to irregularities that depend on time t and need to be captured quickly (Theofilatos et al., 2016), (Dzierma et al., 2010), (Heard et al., 2010). Theofilatos et al. (2016) showed an example in which they made a study of traffic accidents on toll roads. In this study, accidents in different sections of the traveled roads could be detected by sensors in the ground and traffic cameras, which were then transferred to a model. The data is used as one-hour intervals and then labeled by experts. If the model gets a new hour interval, it detects new accident events (Theofilatos et al., 2016). In the study of Dzierma et al. (2010), a geomorphological approach is mentioned that deals with the prediction of volcanic eruptions in certain time periods. The metric used is the volcanic eruption index (VEI) for two new volcanoes. The goal to solve both problems is to predict the occurrence of rare events in a limited period of time (Carreno et al., 2020). Accordingly, time series are used for the detection of rare events, which is considered a characteristic from the supervised learning point of view (Hamilton, 1994). Thus, the goal is to use new time series as rare or normal for a pre-trained model, which is known as supervised time series classification approach in Machine Learning (ML) (Esling and Agon, 2012). Since rare event detection applies to time series and labeled data – normal (N) and rare (R) – measured data usually have unbalanced distributions. This means that many more normal events are found in the measurement data than rare events: $P(R) << P(N)$ (Mitchell, 1997).

Thus, the detection of rare events considers a highly unbalanced supervised time series classification (Köknar-Tezel and Latecki, 2011), (Cao et al., 2011).

- Anomaly: Most irregularities labeled as anomaly are independent of time, unlike the rare event. Nevertheless, these data are also labeled into two categories, normal (N) and anomaly (A) (Carreno et al., 2020). As an example, Miri Rostami and Ahmadzadeh attempted to detect breast cancer in patients. The data collected consisted of many patients suffering from breast cancer. Patients who suffered from breast cancer were considered anomalous. Thus, the data classifies both categories, normal and anomaly (Miri Rostami and Ahmadzadeh, 2018). In a second example, Fiore et al. detected credit card transactions that classify legal or illegal capital movements using a neural network (Fiore et al., 2017). An unbalanced distribution between normal and anomalous events is also found in datasets containing anomalies (Chandola et al., 2009). Therefore, the detection of an anomaly refers to distinctly unbalanced supervised classification. Again, it means that a larger number of normal compared to anomalous events can be found $P(A) << P(N)$ (Carreno et al., 2020).
- Novelty: In the literature, the term novelty uses the context of anomalies as an event that has only one class. Einarsdottir et al. (2016) conducted a study on food monitoring. Here, so-called food envelopes are surveyed, which contain food and may repeatedly contain foreign objects. To ensure that customers do not eat these foreign bodies in the food, a classifier uses labeled dataset of X-ray images for model training showing food without foreign bodies. When an x-ray of food with foreign bodies is used, the model predicts that it is a novelty. In the best case, the model predicts that the image is normal if no foreign bodies are detected. In the case of a novelty, two different scenarios can be considered. On the one hand, the static novelty detection examines a binary supervised classification problem whose dataset consists of a single class. The model decides on events that isolate the normal behavior. In case a new event is detected, the model classifies it as novelty or normal. The model must be able to make a correct classification for the event (Kafkas and Montaldi, 2018), (Einarsdottir et al., 2016). On the other hand, dynamic novelty detection examines the literature, in which future, evolving, novel classes are detected (Masund et al., 2013). It is a supervised classification problem where the number of labels for the class is unknown. In other words, the probability distribution that the classifier predicts can change dynamically. The classifier must adapt to these changes accordingly and place a new class in the already existing classes or store it as a new class (Masund et al., 2013), (Zhu et al., 2018).
- Outlier: The term outlier examines the terms rare event, anomaly, and novelty in the literature (Carreno et al., 2020). However, the term outlier is used in connection with unsupervised learning methods, i.e., with unlabeled data sets. Moreover, the term is often associated with noise, which represents spurious or inconsistent behavior (Aggarwal, 2015), (Aggarwal, 2017). Consequently, the detection of outliers serves as a preparatory measure for classification as rare event, anomaly, or novelty (Teng et al., 1990), (Rousseeuw et al., 2011). Examples for this are human errors that occur when retrieving data that appears uncontrollable and manifests as an irregularity (Barai and Lopamudra, 2017).

4.1.3.4 Placement in Types of Anomalies

The reviewed anomaly types can be distinguished by considering different application areas. Furthermore, anomalies represent their anomaly type. Thereby, a certain anomaly type can be assigned to each anomaly type, which is illustrated in the following chapters. Basically, anomalies are divided into three groups (Abdelrahman and Keikhosrokiani, 2020), (Chahla et al., 2019), (Pimentil et al., 2020), (Djordevic, 2018), which are mentioned for the sake of completeness.

- Point anomaly: Point anomalies are independent, stand-alone data instances that describe a random irregularity or deviation. Furthermore, this anomaly may not necessarily be interpretable compared to the normal behavior of a data set. In connector assembly processes, some of the components may not conform to the standard in terms of their dimensions, and thus may be outside the specified tolerance limit. These connectors are identified as point anomalies (Chahla et al., 2019), (Fahim and Sillitti, 2019). In the following, point anomalies are shown in figure 2 as O1 and O2.
- Collective anomaly: Collective anomalies are collections of individual data points. The collection of the data points may exhibit unusual characteristics with respect to an entire data set (Chahla et al., 2019). For the assembly process example, collective anomalies occur when a sequence of assembled components have dimensions that do not conform to the standard (Chahla et al., 2019), (Pimentil et al., 2020). In figure 2, these anomalies are illustrated as data points O3.

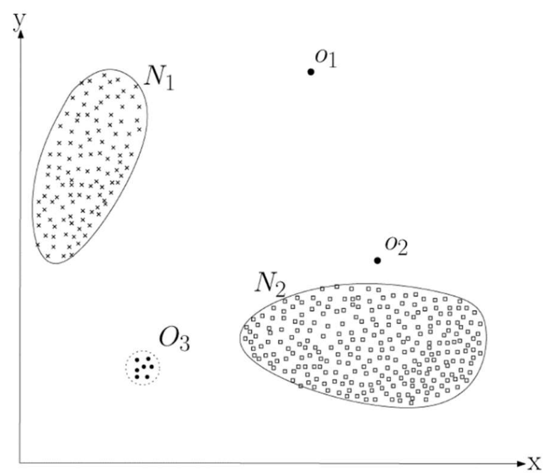


Figure 2. Samples of anomalies in a two-dimensional data set. (reffering to Chandola et al., 2016)

- Contextual anomaly: Contextual anomalies are data instances that highlight anomalies in a particular context (Chahla et al., 2019). These types of anomalies are common in time series data (Abdelrahman and Keikhosrokiani, 2020), (Chandola, et al., 2009), (Djordevic, 2018). Basically, contextual anomalies can be determined by the combination of contextual and behavioral features. Contextual features often use time and space, while behavioral features depend on the domain being analyzed (Djordevic, 2018), (Chandola, et al., 2009). The anomalous behavior is determined based on the values for the behavioral attributes in a given context. A data instance may be a contextual anomaly in a given context, but an identical data instance (in terms of the behavioral attributes) might be considered normal in a different context. This property is key to identifying context and behavior attributes for a contextual Anomaly Detection method (Song et al., 2007). In the assembly process, many components with high and low deviations are detected that do not conform to the norm after an inspection. One reason could be a high load on the assembly machine, as the machine assembles the individual components and wears out the parts with long working time and high load. In Figure 2, the data points in N1 are contextual anomalies.

4.1.3.5 Cardinality of Anomalies

Basically, anomalies can consist of one or more variables. The terms used for time series with such attributes come from statistics that are called univariate, bivariate and multivariate. Accordingly, a univariate time series is a time series that describes an isolated measured variable and applies to novelties (Masud et al., 2013), (Zhu et al., 2018), because it consists of a single class. This case does not necessarily have to occur, since an anomaly can have one class.

Multivariate time series consist of m individual time series, each of which represents an ordered sequence of n real values (Pan et al., 2015). Thus, a bivariate simultaneously corresponds to a multivariate time series. Mathematically, time series represent sequences of numerical values.

$$\boldsymbol{TS_1} = [\boldsymbol{x_1^1}, \boldsymbol{x_2^1}, \dots, \boldsymbol{x_n^1}] \tag{1}$$

$$\boldsymbol{TS_2} = [\boldsymbol{x_1^2}, \boldsymbol{x_2^2}, \dots, \boldsymbol{x_n^2}] \tag{2}$$

$$\dots$$

$$\boldsymbol{TS_m} = [\boldsymbol{x_1^m}, \boldsymbol{x_2^m}, \dots, \boldsymbol{x_n^m}] \tag{3}$$

The dimension of a time series is its cardinality at a point in time t. If the dimension of a time series is one, we call them univariate. For larger dimensions, time series are called multivariate (Jablonski, 2020).

4.1.3.6 Dynamics of Anomalies

Since the detection of anomalies from time series data of the endurance run is targeted in the automotive environment, another level is added to the taxonomy. The mechanical attributes of vehicles characterize moving and non-moving bodies in terms of their longitudinal direction.

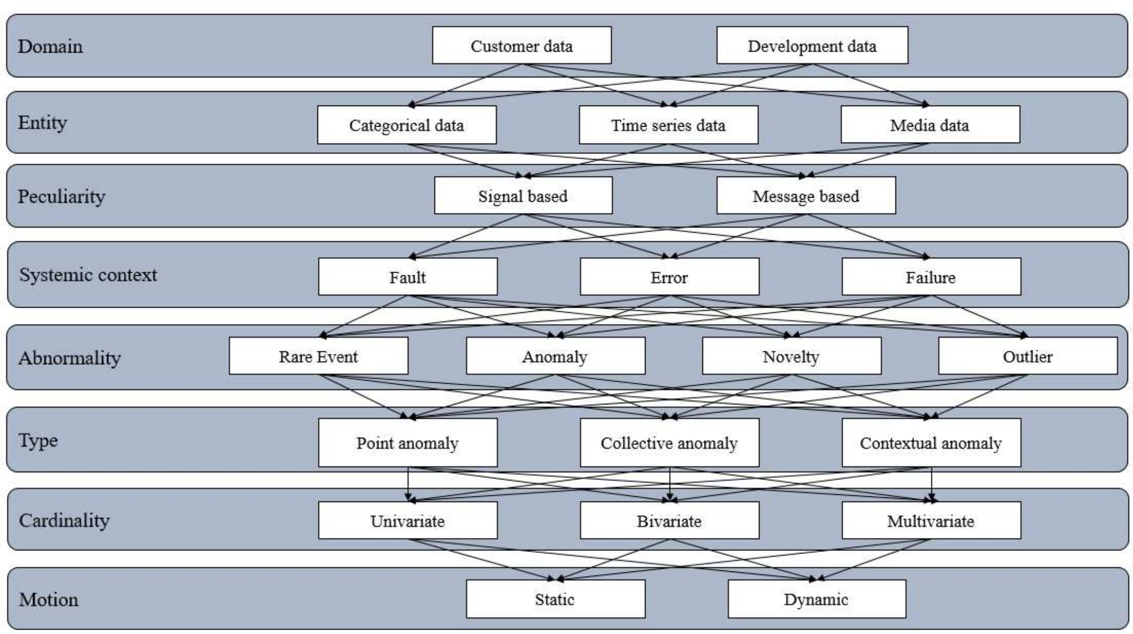


Figure 3. Structure for anomalies occurring in time series of endurance run data.

Discussing moving bodies, the vehicles' dynamics examine a foundation of mechanics. Dynamics describe the study of motion of bodies and the action of forces and resulting motion that are applied to vehicles (Attenborough and Postema, 2008). Although it is misleading to use terms as ‘dynamic’ and ‘static’ differently as described in literature, both terms characterize accelerating vehicles (dynamic) and accelerating vehicles from a standstill (static). These attributes employ to any anomalies in time series data from the endurance run. A detailed explanation with examples follows in chapter four.

With the collected information, the taxonomy establishes the description, classification and definition of anomalies from time series data in the automotive environment (see figure 3). In the following, the paper applies the developed attributes to hesitations. First, the term hesitation has to be defined explicitly due to applying the taxonomical structure to the event.

4.1.4 Taxonomical Categorisation of Hesitations

Based on the developed taxonomy to the hesitations in the time series of the endurance measurement data. The paper proceeds similarly to chapter three and assigns hesitations to a respective attribute of the taxonomy. In this way, it handles alternative expressions of the hesitations and other anomalies in the engine development in the future.

4.1.4.1 Exploration of Entities in Engine Development

A collection of data measured and monitored over a period of time is called a time series. These time series are data points, each ordered by a point in time. The hesitation refers to an event in the time series data that is measured with the help of metrology installed in the vehicles and processed by the ECUs.

4.1.4.2 Placement in Systematic Context of Engine Development

The paper uses the terms fault, error, and failure to describe the semantics of an error chain in which hesitations occur. Avizienis et al. (2004) refer to the progression of the error chain as an error propagation that illustrates the physical relationship between each error chain step chronologically with the failure in vehicles as a starting point and the fault in vehicles as the ending point as shown in figure 4. Since mechanical components are installed in the engine, such as a turbocharger in diesel vehicles, the material of the turbocharger can corrode due to aging, causing leaks to form. Such leaks can cause insufficient boost pressure to build up in the turbocharger, resulting in a reduction for fuel injected into the engine's cylinders. When the vehicle accelerates, the engine receives an insufficient amount of fuel injection, resulting in less combustion of the fuel and thus less mechanical work transferred to the chassis. In a driver, this misbehavior manifests itself in the engine as acceleration deceleration, which corresponds to hesitation. Accordingly, the hesitation describes a symptom of the defective turbocharger (fault), i.e. as an error. Through built-in monitoring systems of the engine, this information can be transmitted to the combination instrument in the driver's cockpit and activate the malfunction indicator light (MIL), which is considered a warning indicator for the driver and signals damage in the engine. The activation of the MIL is thus referred to because of the preceding damage, i.e. failure.

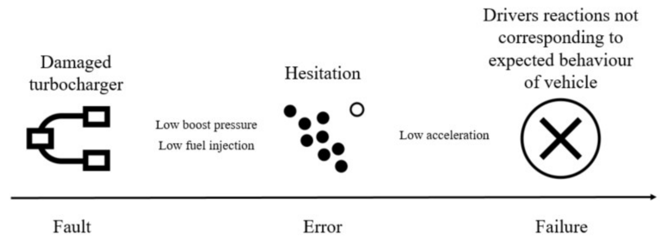


Figure 4. Error propagation for a single form of hesitations. This event should not occur under normal circumstances and is used to illustrate the error propagation.

Since sensors monitor each of the physical mentioned quantities, the current coherences isolate the cause of the fault. The corresponding hesitation events must be detected by ML methods at first.

4.1.4.3 Placement of Hesitations as Abnormality

In order to implement a detection for hesitations, the exact definition of this event is required. The identification of the time intervals with occurred hesitations was treated in previous feasibility studies. These time periods are not necessarily the hesitation anomalies that have to be detected. The collection of the detected events were tied to mathematical constraints, such as the rising edge of the accelerator pedal value as the starting point and the local maximum longitudinal acceleration of a vehicle as the ending point. In these results, there were both normal, expected hesitations, which are plausible due to the physical structure of the engine and anomalous, unexpected hesitations, which do not correspond to the application structure of the engine. To identify the differences between these events, unsupervised learning employs clustering methods to form groups that separate the different curve shapes of the hesitations (Aggarwal, 2015), (Aggarwal, 2017). Up to this point, an outlier detection is performed, which is preparatory for the classification as a rare event, anomaly or novelty (Teng et al., 1990), (Rousseeuw et al., 2011). The detected events are classified according to expected and unexpected events by evaluating first cluster results with engine experts. Collections of implausible and conspicuous curve shapes are revealed due to the usage as training and test data for a supervised learning approach.

Since hesitations are to be detected over the period of an endurance run test cycle of two years, there are missing dependencies concerning the time dimension (Carreno et al., 2020), indicating classification as an anomaly. For rare events, the results are available in the short term (Theofilatos et al., 2016), (Dzierma et al., 2010), (Heard et al., 2010), which has application in real-time systems and online detections. In this case, the hesitations are detected offline after endurance run test cycles are completed. Hence, the hesitations are classified as anomalies. Novelties, on the other hand, have only one class in most cases, and they are used in conjunction with binary problems. Moreover, novelties are used when the normal cases and not the novel ones i.e., the anomalous cases are isolated (Kafkas and Montaldi, 2018), (Einarsdottir et al., 2016).

4.1.4.4 Placement in Types of Hesitations

According to Abdelrahman and Keikhosrokiani (2020) and Chandola et al. (2009), the input variables are time series where the type of hesitations describes contextual anomalies. Point and collective anomalies are excluded because of two reasons. On the one hand, hesitations do not appear as individual data points in time series data (Chahla et al., 2019), (Fahim and Sillitti, 2019). On the other hand, collective anomalies describe the collection of individual data points and still do not have a contextual relation to further measured variables (Chahla et al., 2019), (Pimentil et al., 2020), which is not the case for hesitations, as we will see in the next section.

4.1.4.5 Cardinality of Hesitations

The original dataset of features may be redundant and too large to manage. Therefore, a first step in Anomaly Detection is to select a subset of features or construct a new and reduced dataset of features to facilitate learning of the ML model and improve generalization and interpretability. Considering a fault isolation of hesitations, which should help in the engine development to detect early triggers for hesitations and to eliminate the resulting faults, a limitation to the dataset is performed by reducing the number of features. The signal of the longitudinal acceleration of the vehicle will not be sufficient due to high abstraction. Since the dimension of a time series represents its cardinality at each time point and hesitation represents a time series with diverse features, we define it as a multivariate time series (Jablonski, 2020).

4.1.4.6 Dynamics of Hesitations

For a further classification of hesitations, the anomaly is also examined from a mechanical point of view. In engineering mechanics, dynamics define the study of motions of solid bodies (Attenborough and Postema, 2008). The hesitation as an anomaly occurs both in vehicles accelerating from a standstill and in vehicles already rolling that have picked up pace. Hesitations refer to the static motion that shows the progression of accelerating vehicles from a standstill.

4.1.4.7 Illustration of Taxonomy for Hesitations

Taking everything into account, hesitations can be classified with the help of figure 3, so that the taxonomy for hesitation anomalies can be established (see figure 5).

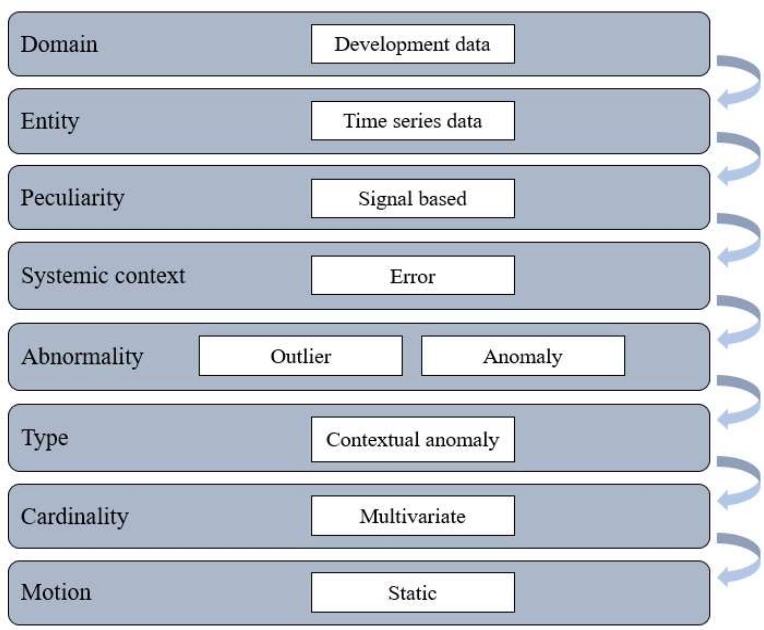


Figure 5. Hesitation categorized in taxonomy table for anomalies occurring in time series of endurance run data.

The taxonomy enables to classify various anomalies that occur in the time series data of the automotive environment and derive suitable ML methods that are tailored to the detection of the events.

4.1.5 Discussion

We combined our research in the Anomaly Detection field and the practical experience collected at Mercedes-Benz to produce this article. To identify and rectify faults in engine development before they occur in the customer environment, we need to formulate clear definitions of anomalies and develop high-performance tools that can process large amounts of data. These tools will allow deeper insights that contribute to the understanding of highly networked vehicles. Therefore, a systemic classification of hesitations in the context of anomaly terms is beneficial for the engine development of OEMs from both a theoretical and practical standpoint.

4.1.5.1 Theoretical Value

First, we consider different anomaly terms with the example of hesitations in endurance run data as such. There are no scientific results for this engine-related anomaly yet. Based on Carreno's (2016) research, there is a misleading understanding in the literature regarding the usage of distinct Anomaly Detection methods. Hence, we derive the classification from the endurance run time series data in detail. Avizienis et al. (2004), Dou et al. (2019), Theofilatos et al. (2016), Dzierma et al. (2010), Heard et al. (2010), Abdelrahman and Keikhosrokiani (2020), Chahla et al. (2019), and Aggarwal (2015) all highlight occasional characteristics of anomalies, making it relevant to incorporate these attributes.

Furthermore, the taxonomy allows us to derive different driveability relevant events that are similar to hesitations from endurance run time series data. By giving a systemic structure to identify anomalies from time series data in the automotive context, future papers can approach a precise selection of ML methods to detect anomalies, as noted by Carreno et al. (2016), Syed and Guttag (2011), Alexandrov et al. (2020), and Oentaryo et al. (2014).

4.1.5.2 Practical Applications

This approach can be implemented independently of the drive type, which will benefit EU-wide from 2035 onwards in the course of the transformation towards electro mobility and away from conventional gasoline and diesel engines (Cornet et al., 2021), (Mazur et al., 2021). Furthermore, specific ML models for different events in time series data of the endurance run can be applied to any drive type.

If new and previously unknown events are found in the measurement data during the development of new engines, the taxonomy can be used to classify them. By exploring the taxonomy features for the undetected events in more detail, the events can be assigned to the components they can be traced back to, the experts who have the expertise on this event or component, and to which trade the event should be assigned. This would enable the company to match and resolve these events more quickly if they are routed to an appropriate area.

The long-term benefits of time and cost savings demonstrate the economic efficiency of this approach for the company. The exact amount of savings is difficult to estimate as several factors play into the respective processes of the development phase of vehicles and engines. However, according to the Rule Of Ten, the later anomalies result in defects in later stages of development, and the higher the costs to fix the defect will be by a factor of 10 (Bamberg, 2018).

4.1.5.3 Limitations of the Study

Since the know-how in companies is usually not bundled, but each employee has his or her own store of knowledge, it has been challenging to develop a structure for the taxonomy that is as generally valid as possible. For this purpose, both expert interviews and document analyses of existing processes had to be carried out, which are only touched upon in the literature in the example of engine development. Accordingly, this taxonomy is applicable to time series data of the endurance run and this is independent of the drive type. But it should be further investigated whether the taxonomy is applicable to several OEMs.

4.1.5.4 Future Research Areas

Due to the limitations of the study, it should also be examined to see if the taxonomy is applicable to several companies in the automotive sector, as individual processes may differ within companies and industries.

Moreover, a deeper elaboration of different ML models needs to be benchmarked and the challenges of contextual time series data need to be addressed in terms of their unbalanced occurrence frequency between normal and anomalous events (Carreno et al., 2016). In addition, working with large data volumes is crucial to extend representative results (Cardoso Silva et al., 2020) by the help of cloud services and resources (Jin et al., 2015), (Kumar et al., 2018) and open-source frameworks for distributed computation. Monitoring these resources is also essential, as the storage and computational costs of processing large amounts of data can increase rapidly.

Mäkinen et al. (2021) highlights a further task, recommending the use of Machine Learning Operations (MLOps) as a developmental process based on time series data in companies for operationalization purposes. By using this approach, ML models can be implemented in a production environment to ensure an automated analysis process for developers and engineers.

4.1.6 Conclusion

The taxonomical structure outlined in the paper allows developers and engineers to identify essential characteristics of time series data, particularly hesitations in endurance run data, leading to precise and error-free analyses for the detection of current anomalies. This enables companies to incorporate customer complaints from the development phase of engines, resulting in reduced future costs from customer vehicle repairs and endurance test drives. The taxonomy also technically classifies various anomalies that occur in engine development, building a sustainable understanding of such events in the company context. By using the subsequently developed analysis based on measurement data, ML projects can be implemented to detect anomalous events early on, improving vehicle comfort. However, further research is needed to examine the applicability of the taxonomy to different companies in the automotive sector and to address challenges related to working with large data volumes and monitoring resources. Additionally, MLOps is recommended as a developmental process for operationalizing Machine Learning models based on time series data in companies.

4.2 Article 2: DTW Clustering

4.2.1 Introduction

Organisations worldwide leverage massive amounts of data to gain value in everyday development processes. Particularly, data volumes released by internal data providers are intended for use in development processes. In automotive engineering, these data are essential for identifying faults in vehicle components. The high degree of networking and complexity of individual components and systems in vehicles present significant challenges (Dai and Gao, 2013). Advanced techniques are thus required for efficient data provision and Anomaly Detection, especially given the rapid growth of datasets (Ikhlaq and Kechadi, 2023). Automotive manufacturers increasingly rely on advanced data analysis methods to monitor the performance of various systems and components (Johanson et al., 2014). As data initially exist in a raw state, they must be optimally compressed using Big Data methods. Big Data systems assist organisations in efficiently collecting, storing, and analyzing large datasets, providing insights into vehicle behavior and emerging trends in engines (Johanson et al., 2014). Through complex analytics and Machine Learning, companies can gain useful insights to enhance decision-making, develop customer-centric strategies, and personalize experiences. Hence, robust Big Data platforms are crucial for managing the information flood and optimizing customer-centric activities (Johanson et al., 2014), (Demirbaga and Aujla, 2024), (Ullah et al., 2024). Specific data mining methods, such as classifications, regressions, clustering, and Anomaly Detection, are employed to explore data (Hu et al., 2014). For instance, a combination of Anomaly Detection and clustering algorithms is used to detect acceleration events. Such analyses yield new insights that can aid in the development of new engines, ensuring improved performance and reliability. One of the main goals in this area is compressing extensive datasets to extract essential information effectively (L'Heureux et al., 2017). By aggregating large datasets into manageable subsets, engineers can focus their attention on specific areas of engines, optimizing analysis time and resource allocation in engine development (L'Heureux et al., 2017). This research focuses on aggregating unlabeled time series measurement data obtained from the endurance testing of Mercedes-Benz vehicles, particularly identifying potentially anomalous decelerations commonly known as hesitations (Tichomirov et al., 2023). According to Xu et al. (2018) Anomaly Detection has been a significant topic in numerous research studies and scientific papers on Machine Learning (ML) and its application in the industry. These transient anomalies pose a significant challenge in unlabeled measurement datasets, requiring innovative approaches for their identification and characterization (Xu et al., 2017). Typically, measurement data in this domain are time-dependent and are determined using time series analysis methods, targeting the detection of irregular accelerations in vehicles based on the longitudinal acceleration curve. Dynamic Time Warping offers a common approach to comparing curve shapes from time series data using a distance measure (Li et al., 2008), (Neamtu et al., 2018), (Salvador and Chan, 2007). Based on this distance measure, different curve shapes can be divided into clusters. The aim is to prepare the clusters of longitudinal acceleration analytically and finally evaluate them with vehicle experts. Such insights are crucial for better understanding vehicle behavior and optimizing performance under various driving conditions (Azadani and Boukerche, 2022). By comprehensively investigating transient anomalies, this article aims to enhance understanding of vehicle behavior and facilitate the development of more robust and reliable engine systems. From this situation arise the following research questions addressed in this article:

RQ1: To what extent can large raw data sets from endurance testing be reduced to aggregate significant information for engine development experts?

RQ2: How can anomalies in unlabeled endurance testing measurement data be identified using data mining methods leading to Anomaly Detection in engine development?

4.2.2 Scientific Methodology

The research project is addressed using the design-oriented approach, resulting in an artifact that aggregates conspicuous curve shapes from time series data of endurance testing. The artifact is developed following the Knowledge Discovery in Databases (KDD) process (Fayyad et al., 1996). The KDD process includes the selection, preprocessing, transformation, analysis, evaluation, and interpretation of patterns (Basu and Meckesheimer, 2007) from endurance testing measurement data to extract useful knowledge. The objective is to identify unrecognized patterns and correlations to support the decision-making of vehicle experts in their assessment of engines. The developed artifact is then evaluated in an expert panel comprising engine development, data, and project management areas, which provided the basis for the artifact's requirements. Finally, a conclusion regarding the artifact is drawn from the experts.

4.2.3 Contextualizing Hesitations from Endurance Testing

To establish context for the use case, the terms hesitation and endurance testing are explained. Endurance testing, known as the Worldwide Analytics Reporting Platform (WARP), is a project that creates a platform for recording, managing, providing, and distributed evaluation of data from worldwide vehicle endurance testing. The engine development departments at Mercedes-Benz are the customers of the WARP project. The expert departments are tasked with ensuring the long-term quality of products. They test a vehicle's life under extreme conditions, whether in icy Finland or Dubai's desert. Endurance tests provide measurement data (time series data) for vehicle development. To manage the data, WARP was introduced as a Big Data platform. From the tests, specific events such as jerks, idle dips, battery charging curves, and others can be analyzed based on channels attached to the vehicle. If the expert departments identify these events, they can consider them in new engines and adjust behavior. In our case, we observe decelerations of endurance test vehicles since vehicle dynamics play a crucial role for the customer in these vehicles, and longitudinal acceleration improves the driving experience. The behaviors of a vehicle are examined under the term longitudinal dynamics. Longitudinal dynamics can be divided into two evaluation metrics. Firstly, the classic vehicle performance is assessed under steady-state full load tests, including maximum speed measurements, elasticity measurements, and measurements between 0 and 100 km/h. Secondly, the longitudinal agility of vehicles is assessed under non-steady-state full load or partial load tests. Longitudinal agility typically includes customer-specific driving behaviors such as traffic light starts, gas pedal agility under full load, starting dosability, and gas pedal agility under partial load. The combination of these areas ultimately yields an index capturing the customer-experienced maximum performance. This article focuses on traffic light starts under full load or start agility.

	Start up	**In motion**
Full throttle	Traffic light start up	Gas burst agility
Partial throttle	Start up dosing	Partial throttle agility

Table 1. Scope of investigation for longitudinal agility

In vehicles with automatic transmission, accelerations from standstill are measured. Unlike acceleration under partial load, measurements are taken only on straight, free stretches, not during vehicle parking, where the driver modulates their acceleration desire via the throttle pedal (Simon and Ferreira, 2022). As longitudinal agility is defined as customer-centric driving, we assume an average customer performing traffic light starts in Comfort mode. Furthermore, we differentiate measurements based on the vehicle's start-stop system. This criterion must be considered in the subsequent evaluation of longitudinal acceleration signals, as the start-stop mechanism introduces additional delay.

Hesitations are defined as the rising edge of the throttle pedal value up to the maximum longitudinal acceleration of the vehicle. A distinction is made between a static event, where the vehicle accelerates from standstill, and a dynamic event, where the vehicle is already in motion. Additionally, hesitations pertain to the context of the time domain, defined as multivariate events fundamentally dependent on several physical quantities, such as the throttle pedal position. Since the measurement data from endurance testing are unlabeled, we initially consider hesitations from the perspective of outlier detection (unsupervised learning) (Tichomirov et al., 2017), (Teng et al., 1990), (Rousseeuw and Hubert, 2011), (Basu and Meckesheimer, 2007). In further research endeavors, these outliers can be labeled and classified with the help of expert knowledge. Depending on the application, additional measurement parameters are examined, which can be included as features in an exploratory approach.

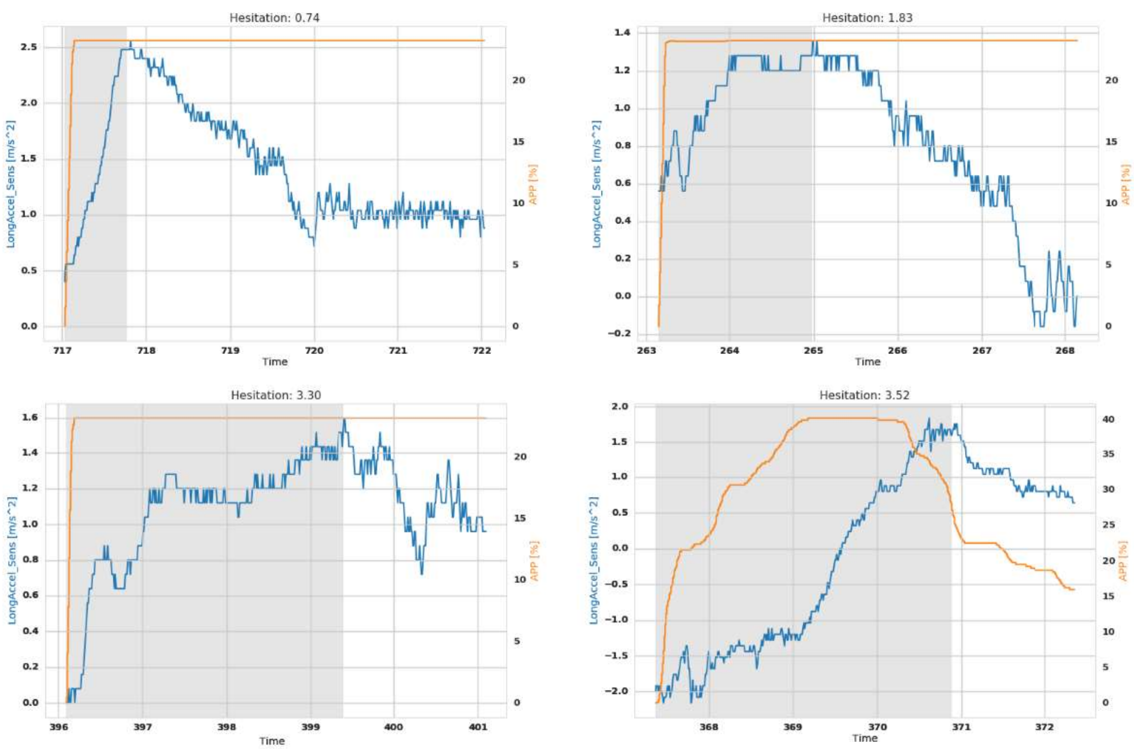


Figure 6. Hesitation with longitudinal acceleration in [m/s²] (blue) and pedal position in [%] (orange) as a time series marked by the deceleration interval in [s] (grey).

The longitudinal acceleration deceleration of a vehicle with a six-cylinder diesel engine is graphically depicted. Generally, it can be stated that an engine with a higher number of cylinders has higher power and therefore accelerates faster than vehicles with a smaller power level. It is to be expected that the deceleration interval from the actuation of the throttle pedal to the maximum longitudinal acceleration is faster, resulting in a shorter hesitation event. However, we cannot yet differentiate between expected and unexpected hesitations. We merely observe that the upper acceleration durations are shorter than the lower ones. The purpose of the graph is to provide an initial understanding of hesitation events. The exact method of aggregating hesitations from the endurance testing data will be explained in the following chapter.

4.2.4 Data Provisioning and Processing

With the theoretical insight into the time series data or measurement data from Mercedes-Benz engine development, we will now discuss how the utilized data were provided. For this purpose, six different approaches were evaluated on how to best access the data for further analysis.

4.2.4.1 Data Science Applications in Engine Development

To reliably provide the measurement data, existing web applications implemented in an internal Mercedes-Benz cloud environment were utilized. This cloud is an extension of the Microsoft Azure cloud and serves use cases and applications in the context of engine development. Essentially, the cloud is used as a private Azure instance. The web applications are housed in a separate Data Science Toolkit and only required minor modifications. They provide a robust approach for loading, checking, and ingesting the endurance testing measurement data.

The basic workflow in the Data Science Toolkit, which includes the individual web applications, consists of three steps. In the first step, a dataset related to the use case is created, which lands in the Blob Storage of Azure Data Lake Storage. This dataset, filled with raw data in Measurement Data Format (MDF), a common format for vehicle measurement data, is restricted by the vehicle series and the endurance testing date. Once the dataset is populated, a measurement data quality application can check the raw data against four criteria:

- Consistent sampling frequency of individual measurement channels: A check is performed on the sampling frequency with which the measurements were recorded across all MDF data. Sampling frequencies of 1, 10, 50, 100, or 200 Hertz are expected, depending on the granularity required for the use case. For the hesitation use case, we use 10 Hertz measurements, as this significantly reduces the data volume, and a granularity of 10 data points per second is sufficient for evaluating hesitation events.
- Highlighting of data gaps: The MDF data is checked to be continuous. If gaps are detected, it is possible to discard the measurement file or to handle the empty cells in the later dataframe by enriching them with the average value.
- Checking for plausibility of individual measurement channels: A plausibility limit is defined for individual measurement channels. If the value of the selected measurement channel exceeds this limit, the data point is not plausible from a physical perspective and must be discarded in the later dataframe.
- Checking for missing measurement channels: Individual measurement channel names are noted in a search bar for which the application should search. If the names are not found, it may be a corrupt measurement file. Extreme caution is required, as the names must be entered exactly into the search. Otherwise, the measurement channel cannot be found due to user error.

Once the measurement data in the dataset has been checked, the data ingest takes place. A data ingest converts the raw data into a big-data-native format and transforms the raw data into a specific schema used for Big Data analysis purposes. In this case, the big-data-native Parquet format and an ASAM ODS packed schema are used. Thus, the data are suitable for analysis on a Big Data platform such as Mercedes-Benz's private cloud environment.

The data can then be analyzed for specific use cases. For this purpose, a web application is used that performs clustering based on the DTW distance measure. The functionality of the application will be explained in Chapter 5.

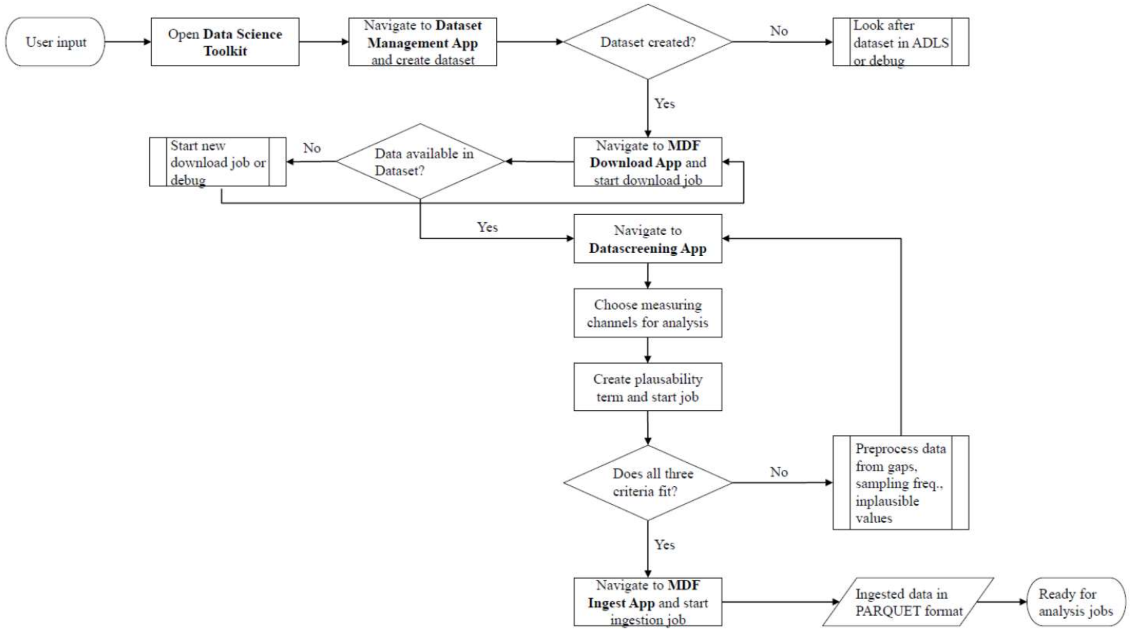


Figure 7. Process for data provision in engine development for measurement data from endurance testing.

Further approaches were examined to provide the measurement data of the endurance test. However, the already described approach has proven to be effective and user-friendly. Additionally, a proprietary library for analyzing measurement data in MDF format was utilized. This library is not available as open-source software. A positive aspect is that this software is also designed for MDF data processing and operates with the same ASAM ODS standard. Using the library also saves time. Consequently, other approaches were not pursued for this reason, among others.

4.2.4.2 Feature Selection

To further refine hesitations, relationships between the dependent variable of longitudinal acceleration (LongAccel_Sens) and various independent variables from the endurance testing measurement data were explored. Initially, brief interviews were conducted with five engine experts, during which the most important measurement channels related to longitudinal acceleration were identified based on experience. Five channels were directly linked to longitudinal acceleration: throttle pedal position (APP), current gear (Gear_State), current vehicle speed (Veh_Spd), current engine speed (Eng_Spd), and current engine RPM (EngRPM). These channels are considered standard metrics by experts and are used in the analysis of measurement data. To validate these metrics, the Spearman correlation coefficient for each measurement channel with the dependent variable, longitudinal acceleration, was calculated. The Spearman correlation was chosen because the Pearson correlation is not robust to outliers. Spearman correlation calculates the coefficient based on the ranks of individual data points, making it suitable for this analysis. Additionally, the mechanisms in engines do not always exhibit linear relationships between components. Unlike Pearson correlation, Spearman correlation is robust to non-linear relationships (de Winter et al., 2016). The result yielded a correlation matrix that relates all mentioned metrics to each other. The correlation coefficient can range from -1 to 1. A value of 1 indicates a positive correlation between the compared metrics, while a value of -1 indicates a negative correlation. The closer the value is to 0, the weaker the correlation between the metrics (de Winter et al., 2016). The focus here is solely on the magnitude of the correlation coefficient.

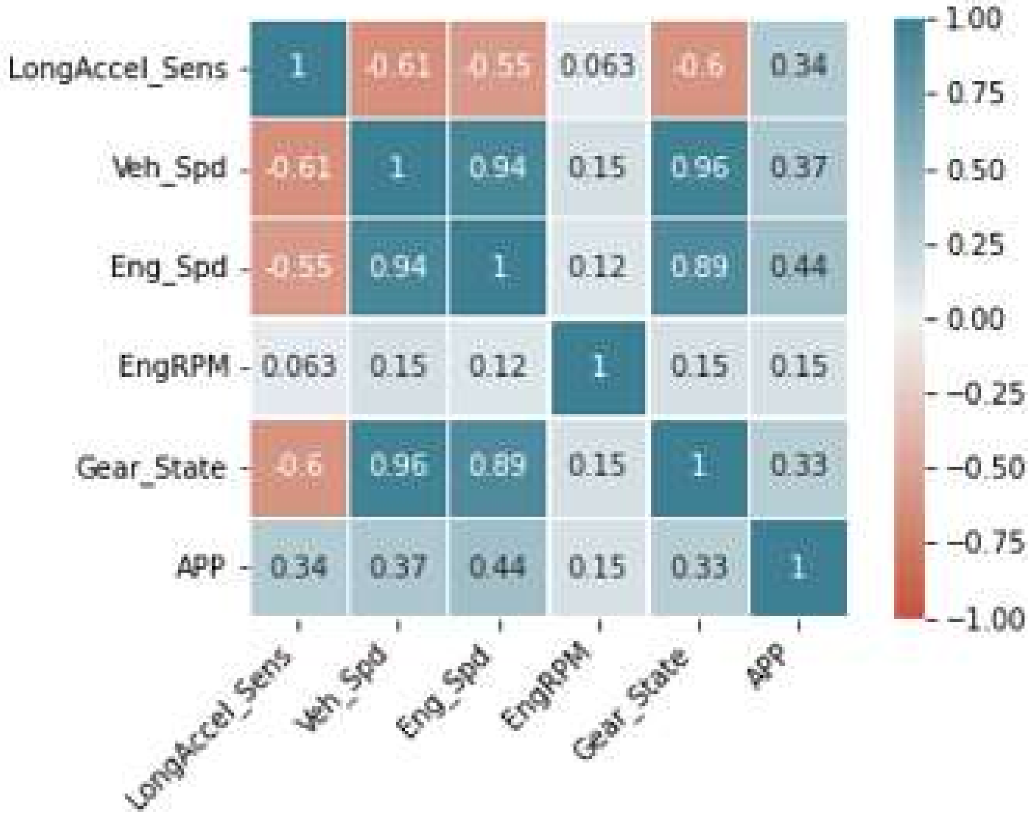


Figure 8. Spearman correlation matrix for the longitudinal acceleration LongAccel_Sens compared to all standard channels.

From these metrics, features were refined in further weekly sprints with the assistance of engine experts. For the hesitation use case, the features listed in Table 3 were considered in consultation with the engine experts as users. The engine speed was not further considered because, on the one hand, it does not strongly correlate with longitudinal acceleration, and on the other hand, according to experts, it does not exhibit direct dependencies on longitudinal acceleration based on experience. A threshold of 0.5 or -0.5 was selected as the lower limit for a "good" correlation (de Winter et al., 2016).

Name	Description	Unit
app_diff	Difference in accelerator pedal position from maximum to minimum	%
app_max	Maximum accelerator pedal position	%
app_t_max	Maximum rise time of the accelerator pedal position	%
app_t_max_half	Half of the maximum rise time of the accelerator pedal position	%
cor_app_v_longAcc	Correlation between accelerator pedal position and longitudinal acceleration	-
dtw_app_v_longAcc	Distance between accelerator pedal position and longitudinal acceleration due to DTW	-
length	Length of a hesitation event	s
hesitation	Detected time from the first increase in the accelerator pedal position to the maximum longitudinal acceleration within a 5-second window	s
max_gear	Maximum gear reached	-
min_gear	Minimum gear reached	-
rpm_avg	Average motor speed	1/min
rpm_max	Maximum motor speed	1/min
rpm_min	Minimum motor speed	1/min
downshifts	Number of switching operations down	-
upshifts	Number of switching operations high	-
vel_avg	Average value of the vehicle speed	km/h
vel_max	Maximum value of the vehicle speed	km/h
vel_min	Minimum value of the vehicle speed	km/h

Table 2. Characteristic features from the engine experts' experience reports

4.2.5 Methodological Approach

In the literature, we find methods that focus on curve trajectories of time series, such as Dynamic Time Warping (DTW) (Li et al., 2008), (Neamtu et al., 2018), which could be suitable for the application of hesitation detection. Additionally, methods like Recurrent Neural Networks (RNN), Autoencoders, and Long-Short Term Memory (LSTM) models are used for Anomaly Detection in time series data (Jablonski, 2020), (Al Jallad et al., 2019), (Roelofs et al., 2021). Since we initially opt for an unsupervised learning approach to reduce the volume of data from the durability testing (Tichomirov et al., 2017), (Teng et al., 1990), (Rousseeuw and Hubert, 2011), the DTW algorithm, which compares curve trajectories using a distance measure, is utilized. Similar distance measures are known in linear algebra as Euclidean distances. However, the Euclidean distance, the cross product of two vectors, does not suffice for comparing curve shapes based on the distance of directly opposite points (Mengyao, 2020), (Keogh et al., 2020). Hence, the DTW distance measure was developed, which compares the individual points of the vectors independently of their order but based on the closest distance from a point of vector A to vector B (Li et al., 2008), (Neamtu et al., 2018). Recurrent neural networks, autoencoders, or LSTM models are used for supervised learning approaches where data is already labeled, aiming for classification. In our case, we process the data without labels and explore initial findings from connected hesitation sequences based on clustering with the engine development experts.

4.2.5.1 Dynamic Time Warping

In theory, DTW compares the similarity between two time series by finding the optimal mapping between the time points of the two series. It works as follows: First, we create a matrix where the rows represent the time points of the first time series and the columns represent the time points of the second time series. Then, we calculate the cost for each possible mapping between the time points and select the path with the lowest total cost. This path shows us how the two time series can be aligned to maximize their similarity (Li et al., 2008), (Neamtu et al., 2018), (Salvador and Chan, 2007).

- First, we define the cost for mapping between two time points, which can be, for example, the squared difference of the values at the time points.
- Then, we create a matrix M where M[i][j] contains the cost for mapping time point i in the first time series to time point j in the second time series.
- We fill the matrix M recursively, starting from M[0][0] to M[n][m], where n is the length of the first time series and m is the length of the second time series.
- Finally, we find the path with the lowest total cost through dynamic programming and calculate the similarities between the two time series.

Essentially, DTW searches for the best match between the two time series, even if they are of different lengths or distorted (Li et al., 2008), (Neamtu et al., 2018), (Salvador and Chan, 2007).

4.2.5.2 Power Iteration Clustering for Hesitations

In our case, this means that we initially use datasets from various vehicle series to obtain a representative result. Since we need to analyze the behavior of potential hesitations first, we use clustering methods (Salvador and Chan, 2007) to detect outliers. With the help of clustering, we explore different curve trajectories of longitudinal accelerations. The algorithm provides a set of clusters, each containing similar longitudinal acceleration curves. In some clusters, we expect normal longitudinal acceleration curve trajectories. In other clusters, we should identify noticeable curve trajectories corresponding to the sought outliers. With expert assessment, we can label the clusters showing irregular longitudinal acceleration curve trajectories as anomalies. This ensures that we detect events relevant to engine development experts.

Power Iteration Clustering (PIC) is an algorithmic approach for grouping time series data based on the use of Dynamic Time Warping (DTW).

- Similarity Matrix: First, a similarity matrix is created for all pairs of time series in the dataset. The similarity matrix S is computed using DTW to quantify the similarities between the time series. The elements Sij of matrix S represent the similarity between time series i and j.

- Normalization: The similarity matrix S is then normalized to ensure values between 0 and 1. This is done by dividing each element of the similarity matrix by the maximum similarity value in the matrix.
- Affinity Matrix: Based on the normalized similarity matrix S, an affinity matrix A is created, indicating the strength of the connection between the time series. A common approach is to define the affinity matrix as A = 1 – S to represent the distance between the time series.
- Power Iteration: Power Iteration Clustering (PIC) uses the power iteration method to find the principal components of the affinity matrix A. This involves iteratively applying multiplication of the affinity matrix with a vector and normalizing the result until convergence is achieved. The dominant eigenvectors of the affinity matrix represent the cluster structures in the time series data.
- Cluster Assignment: Once the dominant eigenvectors are found, the time series are grouped into clusters based on their similarities to the eigenvectors. This is done by assigning each time series to the cluster whose associated eigenvector has the highest similarity.

Overall, PIC combines the dynamic adjustment of similarities using DTW with the analysis of dominant structures in the data through power iteration to enable effective and robust grouping of time series (Weizhong et al., 2013), (Lin and Cohen, 2010).

Clustering algorithms use parameters that control the configuration of the algorithm. One of the parameters is the cluster number k (Mengyao, 2020). To find a suitable value for the parameter k, the Elbow method is used.

In the analysis, the question arises as to how many clusters k should ideally be used. To answer this question, we apply the Elbow method, which helps determine the appropriate number of clusters (Lin and Cohen, 2010), (Mengyao, 2020). We use weighted standard deviation as a measure of the consistency of curve trajectories within the clusters. Thus, we compare each individual hesitation event with the centroid of the cluster. The centroid here represents the median of the entire cluster. To determine the spread of different hesitation events in the cluster compared to the cluster median, the weighted standard deviation is used. It serves as a measure of the dispersion of the values of the hesitation events around their mean. However, the cluster centroid is used as the median. The weighted standard deviation allows us to assess the compactness of the clusters. Higher compactness indicates that the curve trajectories of hesitations within a cluster are more similar.

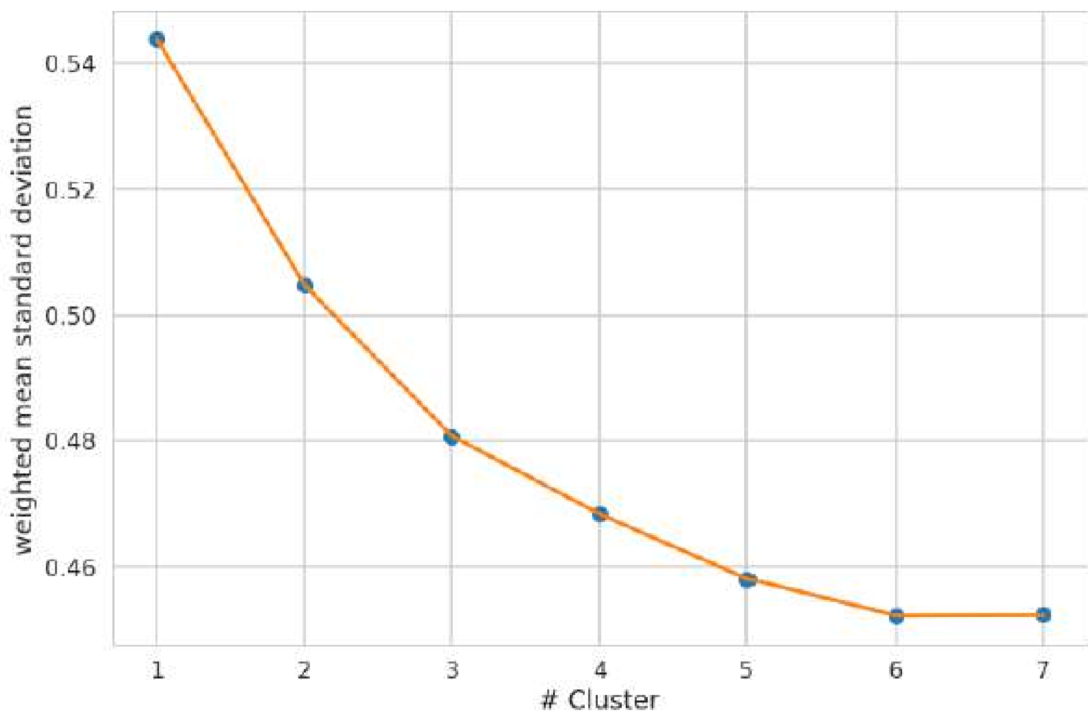


Figure 9. Elbow method for identifying a suitable number of clusters.

Based on the analyses, it is observed that beyond five or six clusters, no significant improvement in the weighted standard deviation is discernible. Therefore, it is decided to use a cluster number of k = 6.

4.2.6 Results

The DTW and PIC methods were integrated into a web application. The resulting artifact is named the DTW Clustering Application. The following results of the analysis stem from three vehicles with a six-cylinder diesel engine. These training and test data were used due to the project's scope. Analyzing other propulsion types, such as hybrid or gasoline, is also feasible. However, additional features that relate to the battery are necessary, especially for hybrid vehicles. Therefore, only diesel engines were considered for this purpose.

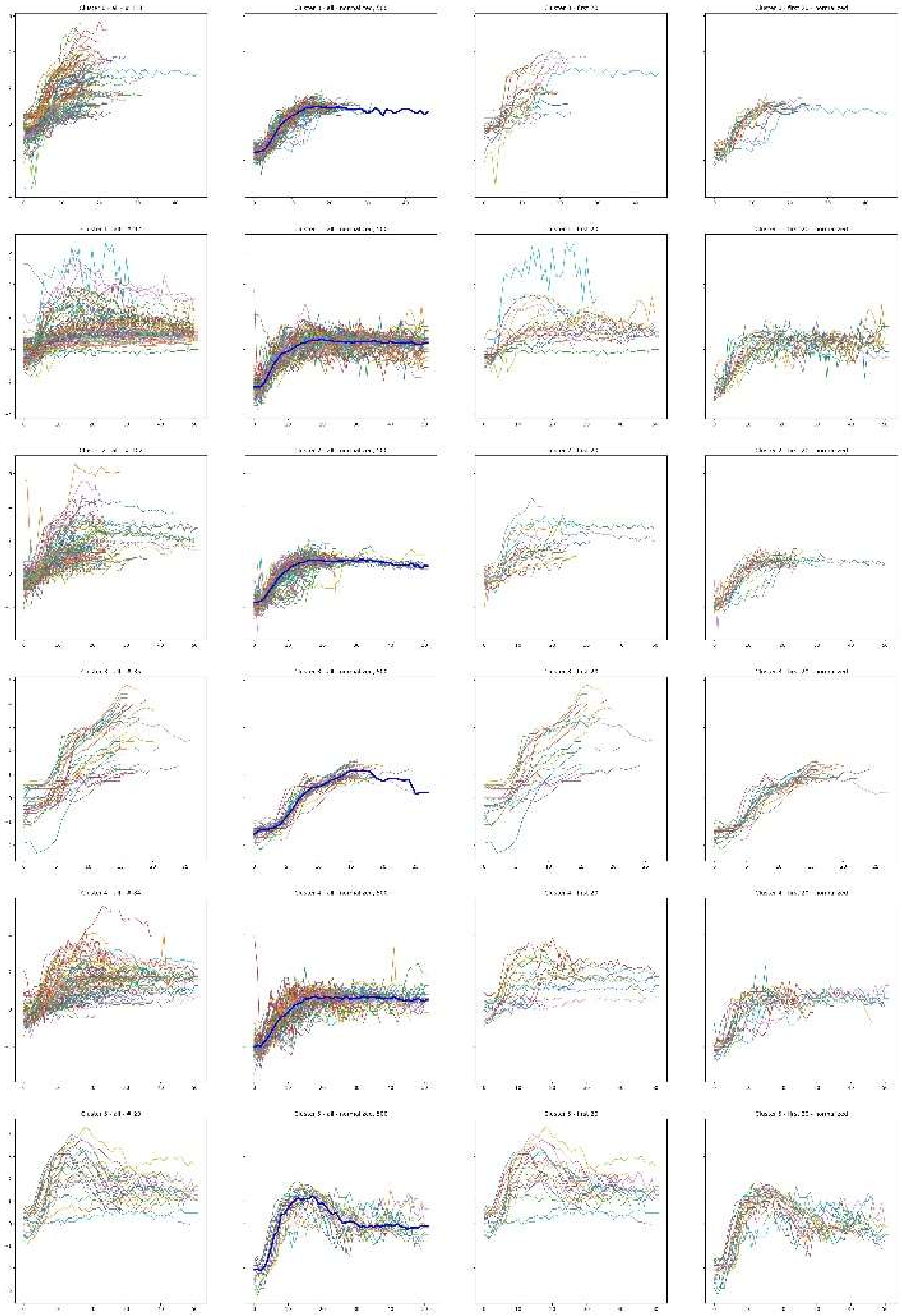


Figure 10. Six clusters with all events, all normalised events with median, the first twenty events and the first twenty events normalised for two six-cylinder vehicles. Vertical axis for longitudinal acceleration LongAccel_Sens in [m/s²] and horizontal axis for longitudinal acceleration LongAccel_Sens in [m/s²] and horizontal axis for time t in [10^{-1} s].

In figure 10, the six different clusters are depicted. The clusters are presented vertically. Horizontally, from left to right, the clusters are shown with all events, with all normalized events with centroid (median for each time point of the sequence), the first twenty events, and the first twenty events normalized. Additionally, the centroids of the clusters have been plotted to facilitate comparison between clusters.

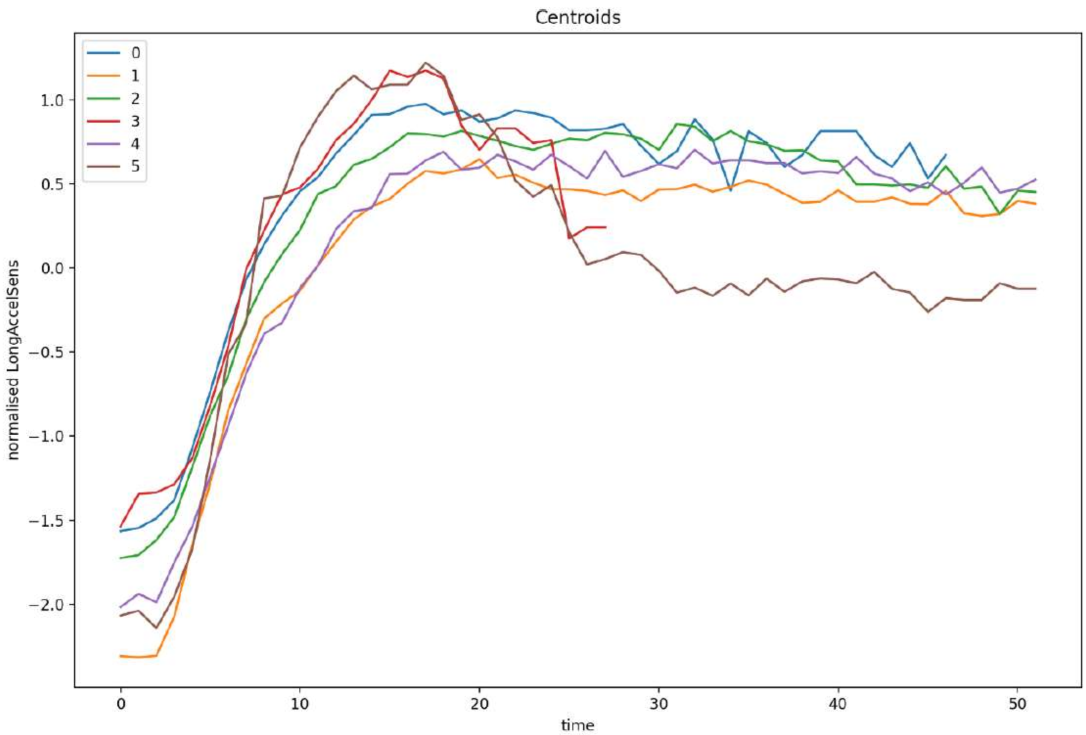


Figure 11. Centroids of all longitudinal acceleration clusters for two six-cylinder vehicles. Vertical axis for longitudinal acceleration LongAccel_Sens in [m/s²] and horizontal axis for time t in [10^{-1} s].

At first glance, the clusters, except for Cluster 0, 2, and 3, do not appear remarkable. The curve profiles correspond to the expected starting behavior of the vehicles. Conversely, Clusters 0 and 3 exhibit events with a maximum duration of 3.25 seconds, although the events are supposed to be detected within a predefined 5-second window. Cluster 2 contains events that are 5 seconds long, but they are only few. Therefore, these events would need to be extracted separately, which would entail additional effort. This step can be expanded upon in the development of a new version. The histogram of the "length" feature confirms that the hesitation events in these two clusters range between 1.4 and 3.25 seconds.

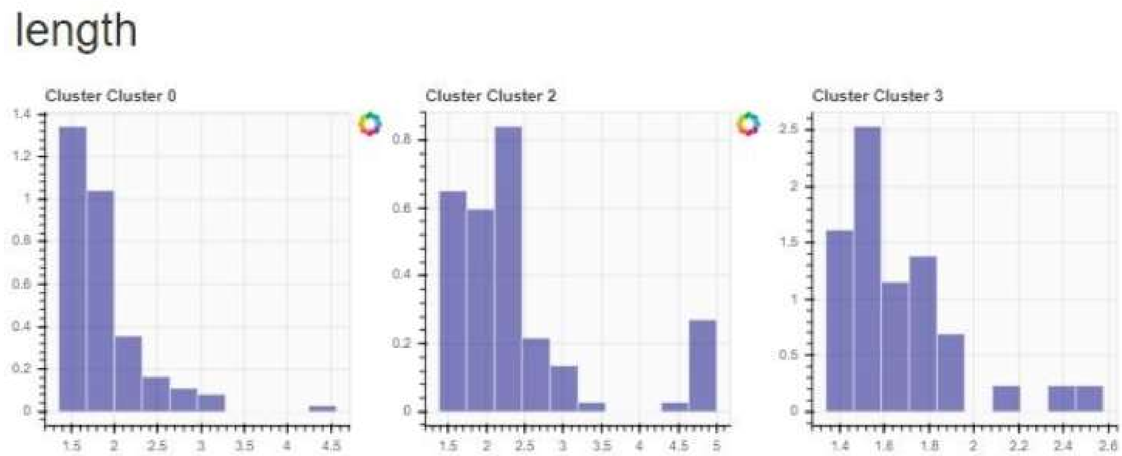


Figure 12. Histogram of the “length” feature for clusters 0, 2 and 3 for two six-cylinder vehicles. Vertical axis describes the normalised frequency and horizontal axis for time t in [s].

It is assumed that Clusters 0, 2, and 3 contain corrupted curve profiles. These shortened curve profiles may stem from faulty measurements and should not be used for further model training. Upon closer examination of the features "max_gear" and "upshifts," it is observed that only Cluster 5 exhibits hesitation events measured from standstill or during vehicle acceleration (see figure 13). As indicated in Table 1, events sought are those belonging to the "traffic light start" category, meaning they demonstrate acceleration under full load. Conversely, this implies that the vehicle executes one or at most two upshifts within the measured 5-second window.

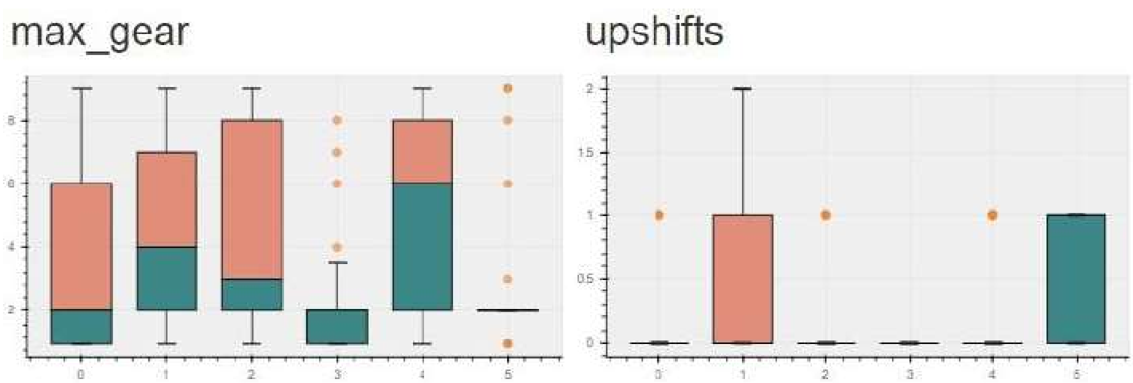


Figure 13. Boxplots of the characteristics “max_gear” and “upshifts” for two six-cylinder vehicles. The vertical axis describes the normalised frequency and the horizontal axis the cluster number.

For all clusters, hesitation events are identified where the gear reaches up to 9 (maximum gear level). Only Cluster 5 for "max_gear" consists, according to the interquartile range (IQR) of the box plot, of 50% of data reaching a maximum of Gear 2. As described above, these are the sought-after events. This behavior is confirmed with the box plot for "upshifts." Cluster 5 exhibits events that have a single upshift, which can be assumed for a traffic light start within a 5-second window for the test samples of the selected vehicles. Now, it could be that these events occur within a speed interval that cannot be attributed to a traffic light start. It is assumed that vehicle speeds above 50 km/h are considered. The feature "vel_max" indicates the maximum vehicle speed. The events are distributed in the interval of 0 to 50 km/h, which appears plausible for the speed during a traffic light start.

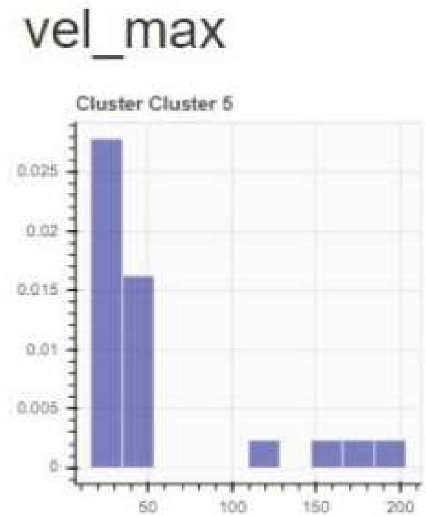


Figure 14. Boxplot of the "vel_max" feature for two six-cylinder vehicles. The vertical axis describes the normalised frequency and the horizontal axis the vehicle speed in [km/h].

Unfortunately, Cluster 5 consists of 23 hesitation events. For further model training, the number of events from Cluster 5 is insufficient. Additional vehicles need to be examined, and data collected to ensure a representative set of events for subsequent detection model training.

For this purpose, further development of the artifact is necessary, introducing an additional process step allowing the separation of required events from the clusters and generating a new dataset for model training.

4.2.7 Discussion

The paper presents an approach to processing large raw data sets from engine development vehicle endurance tests. Additionally, preparatory steps are outlined on how patterns in unlabeled data can be identified using Machine Learning methods for Anomaly Detection.

4.2.7.1 Addressing Research Questions

Firstly, large raw data sets from endurance testing are reduced to aggregate significant information for engine development specialists. To achieve this, Dynamic Time Warping (DTW) is applied as a distance measure to compare individual hesitation events based on their curve shapes, regardless of stretching or compression on the time axis t. Secondly, features are used to examine data using data mining methods to identify patterns (Hu et al., 2014). In particular, hesitation events are grouped into different clusters based on unsupervised learning. These clusters are then evaluated by engine experts, allowing the individual events to be used as a new dataset for Anomaly Detection model training.

The paper addresses two fundamental research questions in the context of engine development based on data from endurance testing. Firstly, it explores how large raw data sets can be reduced to aggregate meaningful information for domain experts. For this purpose, DTW is used as a distance measure to compare individual hesitation events based on their curve shape, independent of stretching or compression on the time axis. Secondly, it investigates how anomalies in unlabeled measurement data can be identified to prepare for Anomaly Detection in engine development. The insights gained from the analyses can provide clusters for further model training. The assessment of normal or abnormal clusters must always be done in consultation with domain experts in engine development, who contribute domain knowledge.

The research initially focuses on providing and preprocessing raw data from endurance testing. Existing web applications are used in a private cloud environment to load, verify, and transform the data into a suitable format. A measurement data quality application checks the data for consistency of sampling frequency, the presence of data gaps, the plausibility of measured values, and the presence of specific measurement channels. Subsequently, the data is converted into a big-data-native format and prepared for further analysis.

In the next step, data is analyzed using data mining methods to identify anomalies and irregularities. A clustering algorithm based on the DTW distance measure is used. By analyzing these clusters, potential anomalies and irregularities in the measurement data can be identified.

Overall, the paper offers a structured approach to reducing large raw data sets and identifying conspicuous clusters in unlabeled measurement data from endurance testing. By applying these methods, domain experts can gain valuable insights and make informed decisions in the engine development process.

4.2.7.2 Limitations of Research

Regarding Power Iteration Clustering (PIC), achieving convergence refers to the process where the iterative application of the power matrix to a starting vector leads to a stable state. In this stable state, the eigenvectors of the affinity matrix do not change significantly between successive iterations, indicating that they accurately capture the dominant structures in the data. Specifically, this means that the results of power iteration no longer change significantly when the algorithm is run for further iterations. This is an indicator that the algorithm is converging to a solution, and its execution can be terminated.

If convergence is not achieved, this can affect the accuracy and reliability of the results of the clustering algorithm. It may be necessary to review and, if necessary, adjust the algorithm to promote convergence and ensure that the results are appropriate. Failure to achieve convergence can be due to unstable data structures, non-optimal algorithmic parameters, and numerical instabilities.

In our case, in some analyses, the limits were reached due to the large amount of data, preventing the analyses from being completed. Both the amount of aggregated data from the similarity matrix S, resulting from DTW with $O(N^2)$ complexity, and unstable structures due to high dimensionality in the measurement data, could be causative factors for this effect. This behavior should be examined in further research.

4.2.7.3 Further Research Directions

It has been found that the aggregated cluster data can provide the basis for further development of a supervised learning approach. The data can be defined as a new dataset with non-anomalous, labeled data and then used for model training for full-fledged Anomaly Detection. Additionally, the presented methods should be transitioned into an automated process to offer engine development specialists further analysis time and improved manageability with the results.

4.2.7.4 Data Availability

Due to legal restrictions, the original data from Mercedes-Benz AG cannot be made publicly accessible in any form.

4.3 Article 3: MLOps Pipeline

4.3.1 Introduction

In today's industry, the rapid professionalization of Proof of Concepts (PoCs) and pilots is crucial to adequately respond to sudden problem cases. Due to the increasing complexity of connected vehicle components, the automotive industry is compelled to rely more heavily on intelligent systems such as Machine Learning (ML) and Deep Learning (DL) (Cline et al., 2017). These methods help in efficiently analyzing large data sets and detecting unknown anomalies. However, using these algorithms poses a challenge because they are complex and their practical application for users in engine development is not trivial. It is essential to make intelligent systems productive quickly and work towards scalability to meet growing demands. By employing Machine Learning Operations (MLOps), a methodology for the efficient deployment and maintenance of Machine Learning models, anomaly cases can be brought into production more swiftly. This enables timely responses to potential problems and continuous improvement of system reliability and performance. Hence, early detection and resolution of unusual events in engine development play a crucial role (Betz, J. et al., 2019; Bekar, T. E., Nyqvist, P. and Skoogh, A., 2020).

To qualify new engines, extensive data from endurance tests are needed. Endurance testing of vehicle fleets under various driving conditions generates a wealth of time series measurement data (Bock, F., Siegl, S. and German, R., 2016). Time series measurement data are of particular interest because they represent continuous measurements over time, revealing dynamic behavioral patterns of vehicles. To efficiently analyze these extensive data sets, a semi-automated MLOps pipeline is required to support experienced vehicle experts in detecting anomalous events. Such a pipeline should compress the data to a manageable level while extracting essential information for Anomaly Detection (Tichomirov et al., 2024). The information will then be used in the engine development departments.

To this end, the approach for detecting acceleration delays in the measurement data was developed as the first artifact of a design-oriented research project, intended for integration into the MLOps pipeline. The existing unsupervised learning approach will be expanded (Tichomirov et al., 2024) by labeling anomalous events in the measurement data for subsequent use in a supervised learning approach. Initially, a data set containing the raw data from the endurance tests in the Measurement Data Format (MDF) is created. After undergoing a quality check for completeness, sampling frequency, and plausibility, the data set is converted into the Big-Data-native Parquet format of the ASAM ODS standard. The user then defines parameters describing the sought event and generates a new data set. Additionally, other measurement channels can be included as feature sizes for the analysis to identify potentially anomalous behavior in the measurement data more easily. Based on the Dynamic Time Warping (DTW) algorithm, the events generated by the user-defined function (UDF) are compared using distance measures from event to event (Keogh, E. and Pazzani, J. M., 2001; Rakthanmanon, T. et al., 2012). The comparison is performed using two clustering algorithms: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Agglomerative Hierarchical Clustering (AHC) (Ester, M. et al., 1996; Nazari, Z. et al., 2015). Finally, the results are evaluated by a group of engine development experts. Anomalous events and clusters are labeled as in order (iO) or not in order (niO). The normal data are then used to train an autoencoder model that will later identify potential anomalies in test data.

Another crucial element of the MLOps pipeline is the continuous monitoring and evaluation of ML models. This ensures that the models remain reliable even after deployment in production and can be updated or adjusted as needed. A robust monitoring system can help detect model deviations early and take appropriate measures to optimize model performance (Breck, E. et al., 2017). Models can also be evaluated using MLFlow based on model Key Performance Indicators (KPIs), ensuring that the most suitable model is used to create an anomaly report.

The following research questions arise from the presented workflow for this article, aiming to determine how an MLOps pipeline can be implemented in engine development and what insights can be gained from optimizing the process:

RQ1: How can analysis models for time series data in a prototype phase be rapidly scaled to extract crucial information from measurement data in engine development?

RQ2: To what extent can the operationalization of complex Machine Learning models be optimized to achieve the highest possible quality of Anomaly Detection results in time series data?

4.3.2 Motivation

In engine development, processing time series data is crucial for optimizing engine performance and reliability. MLOps can play a central role, as it manages the entire lifecycle of Machine Learning models, enabling the handling of dynamic data across different industries. According to a survey by Mäkinen et al. (2021), up to 40% of data scientists from various sectors work with both developing Machine Learning models and building the associated infrastructure, with a significant portion of about 28% of their work focusing on time series data. This underscores the importance of MLOps in this field.

Time series data, such as sensor data from engines, require continuous monitoring and analysis. Through MLOps, models that process and evaluate these data can be regularly updated and retrained to ensure the accuracy and relevance of predictions. This is particularly important as operating conditions and performance characteristics of engines constantly change. An example of this is the implementation of new combustion engines based on the Euro 7 emission standard. The European Green Deal roadmap includes a proposal for stricter air pollutant emissions for combustion engine vehicles by 2021 (European Commission, 2020). Therefore, it is essential to quickly respond to adjustments for combustion engines. The initial implementation and subsequent extension due to changing requirements of Machine Learning models can support the work of engine experts by automating the processing of large measurement data and the creation of reports. Consequently, organisations have models in production that are increasingly business critical. They continually consider how to scale and continuously develop these models while maintaining the quality of production models (Mäkinen et al., 2021).

Another aspect is the maturity of the Machine Learning models with which engine experts work. Organisations that operate pipeline-centric workflows are characterized by frequent retraining and deployment of models. In engine development, where engines are constantly optimized to provide precise predictions based on measurement data, MLOps is therefore indispensable. It enables seamless integration and continuous improvement of Machine Learning models, ultimately leading to higher efficiency and reliability of engines (Mäkinen et al., 2021), (John et al., 2021).

Further challenges to be addressed include predictive analyses and failure prediction in engine development, supported by the application of time series analyses. According to Breck et al.

(2017), Kreuzberger et al. (2022), and Sculley et al. (2015), MLOps practices enable these challenges to be met by providing a robust infrastructure for continuous development, testing, and deployment of Machine Learning models.

Building on this, Tichomirov et al. (2024) developed a clustering model for time series data from vehicle endurance tests, which requires infrastructural expansion. The previously mentioned methods for data provision and analysis can be integrated into an MLOps pipeline, allowing for Anomaly Detection of transient events in production.

4.3.3 Structure of the MLOps Pipeline

With the successful introduction of Development Operations (DevOps), companies aim for continuous processes in the development of software systems. To harmonize the development and operation of such systems, MLOps extends the principles of DevOps to ML systems (Mäkinen et al., 2021), (Vuppalapati et al., 2020).

MLOps is a practice in the ML field that applies DevOps principles to ML systems to unify the development and operation of ML systems (Zhou et al., 2020). From the perspective of Continuous Integration (CI), additional test procedures are introduced alongside classical unit and integration tests, such as data and model validations. From the perspective of Continuous Development (CD), processed datasets and trained models are automatically and continuously delivered by data scientists to ML system engineers. From the perspective of Continuous Testing (CT), the arrival of new data and the degradation of model performance necessitate triggering model retraining, thereby improving model performance (Breck et al., 2017).

The current structure of the MLOps pipeline is implemented on an internal corporate cloud environment based on Microsoft Azure. The pipeline is used for detecting anomalies in time series data. As a use case, longitudinal acceleration events of vehicles, referred to as hesitations, are considered (Tichomirov et al., 2023). This includes the MDF Download App, which downloads the required raw data, the Data Screening App to check data quality and completeness, and the MDF Ingest App, which reads the measurement data and converts them into the Big-Data-native Parquet format to prepare them for the next stage of the pipeline. These modules are part of a data science toolkit for engine development and are specifically designed to process time series data from endurance tests.

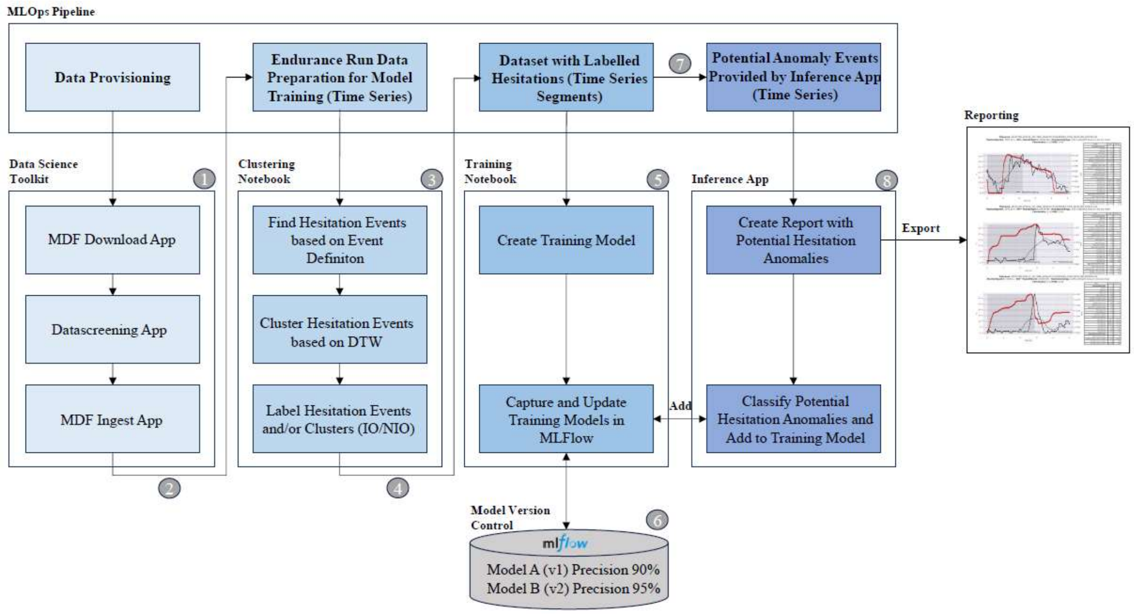


Figure 15. MLOps pipeline in the engine development for hesitation detection.

The next step involves preparing the formatted time series for model training. The processed time series data are aggregated through a clustering notebook using Spark jobs. The aggregated data now consist of the time series segments initially defined in a parameter cell within the notebook. As described in Tichomirov et al. (2024), the time series segments are compared using Dynamic Time Warping (DTW) to group them through DBSCAN or AHC clustering.

Subsequently, the results are evaluated by a working group of engine and data experts. Based on this information, both entire clusters and individual time series segments can be labeled as in order (iO) or not in order (niO). For the case of hesitations, where unknown events need to be detected, time segments and clusters with expected curve patterns are sought, which will then be used as training data for the autoencoder model. The autoencoder model is a neural network trained to reconstruct its input data by first encoding it into a low-dimensional representation (latent space) and then decoding it back. The reconstruction of time series data occurs by compressing the data through the encoding layers and attempting to restore them through the decoding layers (Hinton and Salakhutdinov, 2006). Anomalies are identified by measuring the difference between the original input data and the reconstructed data. High reconstruction errors indicate anomalies, as the model cannot effectively learn rare patterns (Ruff et al., 2021).

Model training is conducted using another training notebook that creates a model to predict hesitations. By selecting the normal time segments (iO), the model recognizes hesitation events after training that exhibit abnormal behavior deviating from the norm of the training data, as fundamentally described by Bock et al. (2016). Training a model based on niO events would lead to non-representative results due to the insufficient number of abnormal events (Bock et al., 2016). If the model now identifies anomalies in the test data of new engines, these events can be applied by engine experts, accordingly, ensuring they do not recur in a new series.

Model management is supported by MLFlow, which documents the various model versions and their performance. Models demonstrating high quality and thus a low Mean Squared Error (MSE) can be activated for an inference web application in MLFlow. The web application generates reports and integrates the models. This application creates reports highlighting potential hesitation anomalies. These reports are forwarded to experts to validate the results and further optimize the models.

The pipeline thus demonstrates a structured approach for detecting and classifying anomalies in time series data. By integrating data science tools, clustering techniques, and model management, a continuous improvement process is ensured, which aims to enhance both the precision and efficiency of Anomaly Detection.

4.3.4 Results

The endurance test measurement data from four vehicles are first downloaded and stored in a dataset. This dataset is stored in the Blob Storage of Azure Data Lake Storage (ADLS) on an internal corporate instance. The data are checked for gaps, plausible data points of the independent variable, longitudinal acceleration, and plausible sampling frequencies using the Data Screening App. Similar quality controls are performed as described by Kirchen et al. (2017). After quality control, the data are converted into the Parquet format to perform distributed computing operations on the instance using Spark jobs (Will et al., 2021). The raw endurance test data do not exhibit any data gaps, were continuously sampled at a frequency of 10 Hz, and fall within an interval between -5 and 6 m/s², which corresponds to a coherent acceleration range. This step, as explained in Tichomirov et al. (2024), will be used for the analysis of hesitations within the MLOps pipeline.

4.3.4.1 Hesitation Analysis Parameters

Once all the data are available, the analysis parameters for detecting hesitation events based on the accelerator pedal position and longitudinal acceleration are added.

Name	Value	Unit
PED_GRAD_MIN	30.0	% per s
PED_GRAD_DURATION_MIN	0.0	s
PED_LOW_MAX	0.0	%
PED_HIGH_MIN	20.0	%
PED_LOW_DURATION_MIN	0.2	s
PED_HIGH_DURATION_MIN	0.2	s
PED_LOW_HIGH_SHIFT_MAX	1.0	s
TIL_FIXED_LEN	5.0	s
TIL_FIXED_SHORT_LEN	3.0	s
TIL_FIXED_SHORT_MIN_POINTS	3.0	-
FREQ_REF	10.0	Hz

Table 3. Hesitation event parameters for analysis.

The defined analysis parameters are briefly explained to understand figure 16 and to better comprehend the constraints in searching for hesitation events. Figure 16 does not show real signals but schematically illustrates the areas required for the definition of hesitations.

- N_EVENTS_MAX: Maximum number of hesitations to be detected.

- PED_GRAD_MIN: Minimum value for the gradient of the pedal position at the start of the hesitation.
- PED_GRAD_DURATION_MIN: Minimum duration for the gradient of the pedal position at the start of the hesitation.
- PED_LOW_MAX: Maximum value of the pedal position just before the start of the hesitation.
- PED_HIGH_MIN: Minimum value of the pedal position after the steepest gradient within the hesitation.
- PED_LOW_DURATION_MIN: Minimum duration of the pedal position before the start of the hesitation.
- PED_HIGH_DURATION_MIN: Minimum duration for the pedal position after the steepest gradient.
- PED_LOW_HIGH_SHIFT_MAX: Minimum duration between the interval where the pedal position is less than PED_LOW_MAX.
- TIL_FIXED_LEN: Fixed length of the interval displayed in the report.
- TIL_FIXED_SHORT_LEN: Fixed length of the time interval used to train the autoencoder model.
- TIL_FIXED_SHORT_MIN_POINTS: Minimum number of data points required for the training intervals.
- FREQ_REF: Frequency to which the intervals for the autoencoder model are standardized.

Compared to Tichomirov et al. (2024), the definition of acceleration events is extended using the described analysis parameters. This article uses the rising edge of the accelerator pedal position as the initial trigger point and the maximum longitudinal acceleration within a 5-second window. For the current application, the definition is expanded. The first 0.2 seconds of the pedal signal are ignored, assuming the pedal signal's dead time. After 0.2 seconds, the pedal signal responds and rises.

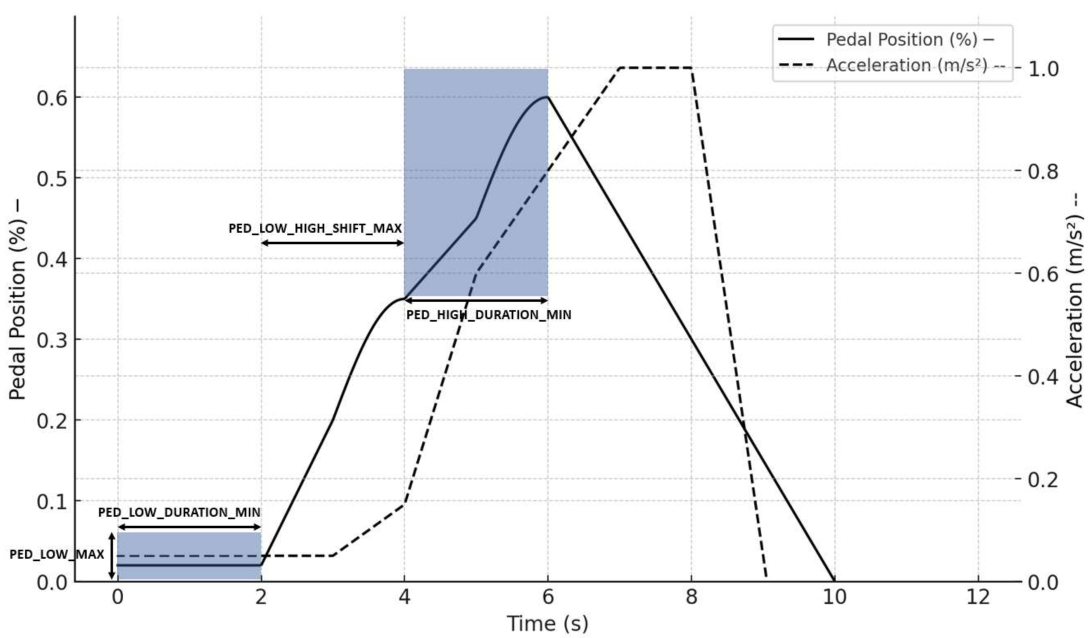


Figure 16. Illustrated hesitation event parameters in graph.

Additionally, the following signals are used as parameters relevant for generating subsequent features:

- CHAN_PED: Accelerator pedal position in %.
- CHAN_ACC: Longitudinal acceleration of the vehicle in m/s².
- CHAN_ODO: Odometer reading in km.
- CHAN_SPD: Vehicle speed in km/h.
- CHAN_RPM: Engine speed in r/min.
- CHAN_GR: Current gear.
- CHAN_ADDITIONAL: Array to which multiple signal names can be added if additional sizes need to be included.
- ENDURANCE_PROFILE: Endurance profile On-Board Diagnosis (OBD), marked as emissions-relevant testing.
- SOFTWARE: Software version of the engine control unit
- CHAN_MODE: Driving mode the vehicle is in at the time of a hesitation.

The mentioned metadata enhance the comparability of the analyzed vehicles. The endurance test data were selected based on these three parameters, although the driving modes can change during testing. At this point, it was decided not to include a feature store in the MLOps pipeline, as the number of features for feature management is still too low (Schlegel, M. and Sattler, K., 2022).

Using user-defined functions, all hesitation events were detected from the endurance test measurement data of four vehicles. For the aggregated data, a Dataframe with 39 columns was created. The columns consist of both the hesitation time series segments and the time series segments of the listed signals and their features. Additionally, the time series segments were enriched with the metadata. The Dataframe is saved as a Pickle file on the cloud instance for reuse and further investigation of the features.

4.3.4.2 Hesitation Events

For control purposes, a cell in the notebook displays the first 10 hesitation events to verify the curve patterns and validate them with the help of engine experts.

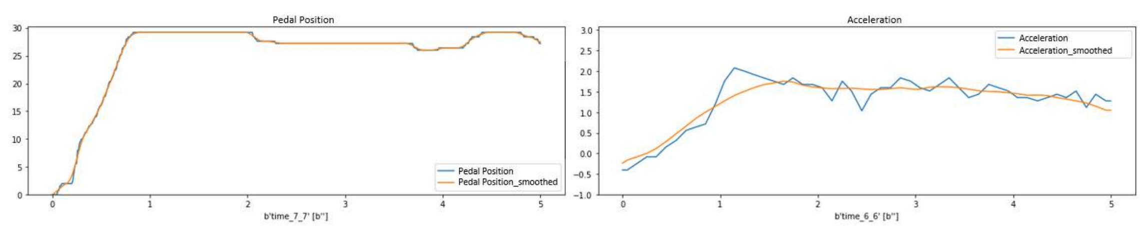


Figure 17. Hesitation event shown by Pedal Position, Pedal Position_smoothed, Acceleration and Acceleration_smoothed.

The signals for the accelerator pedal position and the longitudinal acceleration are displayed at 10 Hz as blue curves, while the smoothed signals are shown as orange curves.

4.3.4.3 Clustering

Next, one of the two clustering algorithms is chosen for the analysis of the hesitation events. DBSCAN's strength lies in detecting dense areas and outliers, which is less intuitive with time series data using DTW. The distance calculated by DTW is not directly comparable to a fixed radius in a spatial context. It is an accumulation of distances along the optimal path through the time series, making the interpretation of ϵ, the radius configured for DBSCAN, more complex

(Ester, M. et al., 1996). AHC creates a hierarchical structure (dendrogram) that represents the relationships between data points at different clustering levels. This hierarchy is particularly useful for visualizing and interpreting the complex relationships in time series data (Nazari, Z. et al., 2015). In this case, AHC is used exemplarily, as the basis of DTW distances results in stronger clustering. Outliers, which DBSCAN targets, can be identified with the help of engine experts who can differentiate the function and curve patterns for the longitudinal acceleration of vehicles. Additionally, adjusting a clustering parameter, such as with AHC, is initially more user-friendly compared to the radius ϵ in DBSCAN, especially for engine experts who may not be familiar with the clustering algorithm's intricacies. At this point, it is again relevant to form an interdisciplinary working group to evaluate the results.

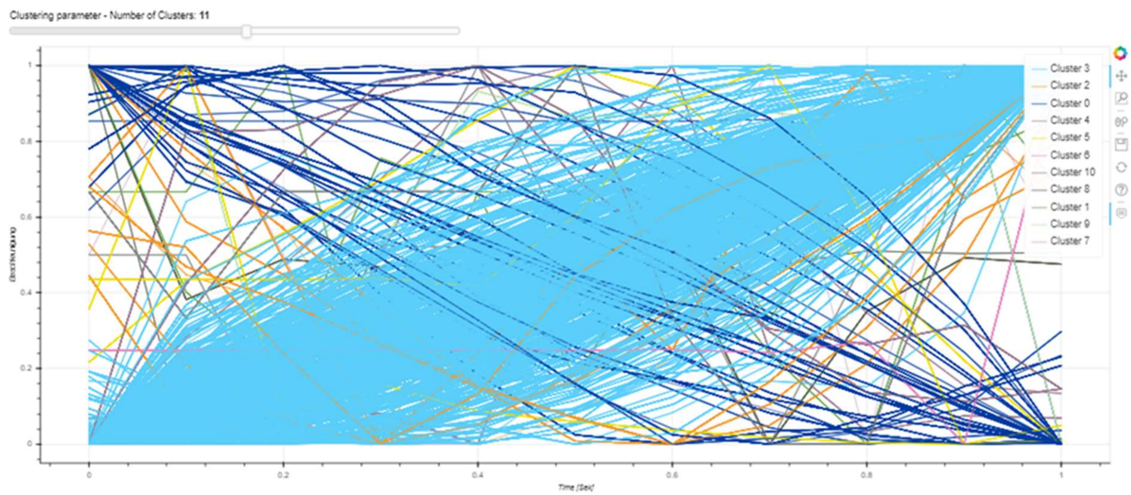


Figure 18. Hesitation events clustered by AHC with scrolling bar for adjustment of number of clusters parameter.

Adjusting via the control element assists in selecting the clusters. With a cluster count of k = 11, clearly identifiable clusters have formed, which are now examined separately. Particularly notable are Cluster 3 (light blue) and Cluster 0 (dark blue), which contain many hesitation events. The goal is to identify clusters and individual events to be used for training the autoencoder model for supervised learning. As previously mentioned, the autoencoder seeks normal or expected curve patterns of individual events and clusters, as the model is trained on these data. Consequently, the model can distinguish between normal and anomalous events (Hinton and Salakhutdinov, 2006).

Additionally, figure 19 shows that the clusters of hesitation events display a time interval of only one second. This indicates that it is merely a display and not the real data, which have been shortened to one second. The length of the time intervals for the hesitation events is determined by the parameter TIL_FIXED_SHORT_LEN, which is currently set at 3 seconds. Thus, the autoencoder model is trained with time series segments of 3 seconds. For the generated report, the time interval of potential anomalies is controlled by the parameter TIL_FIXED_LEN, which is set to 5 seconds. Since the reconstructed events are linked via the metadata filename of the Dataframe, mapping is performed when creating the report. This ensures that the real and not the calculated hesitation events are displayed in the report.

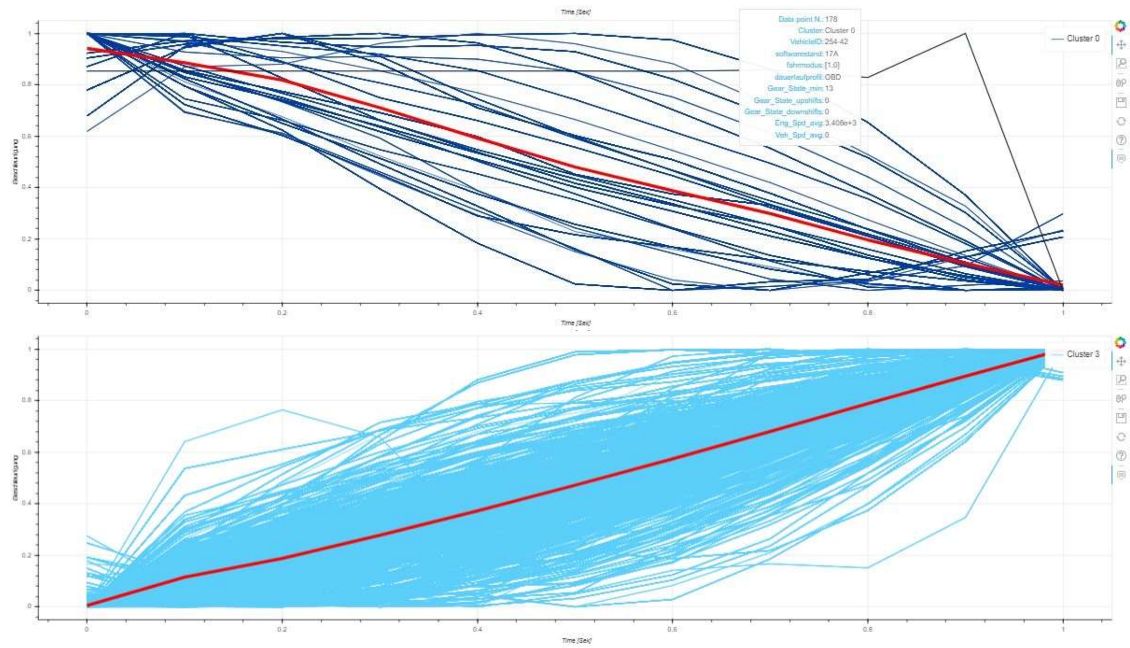


Figure 19. Hesitation events clustered by AHC with normal Cluster 0 and Cluster 3.

If individual events need to be removed from a cluster, the characteristics can be viewed via hover, providing engine experts with insights into normal or anomalous behavior. In Cluster 0, a hesitation event deviates from the cluster norm. Examining the curve, the longitudinal acceleration remains constant at 0.83 m/s² before a brief increase and then a decline. This indicates the vehicle is idling. This behavior is evident from the average engine speed (Eng_Spd_avg) and the average vehicle speed (Veh_Spd_avg). The engine speed is 3408 r/min, but the vehicle speed is 0, indicating that the vehicle is idling while the accelerator pedal is pressed, causing the engine speed to increase. This hesitation event can be excluded from the training data.

Other clusters have a small number of up to 8 hesitation events with abnormal curve patterns, where the vehicle is also idling, and only the accelerator pedal is pressed, which occurs during idle tests to check for idle drops. Additionally, the Gear_State_min attribute shows that the vehicle is in park mode (P). To accurately represent the curve patterns of Clusters 0 and 3 during model training, the remaining clusters are ignored for the training data. However, this poses a risk of excluding events detected by the autoencoder as potential anomalies, even if the curve pattern is plausible, such as in the idle test.

It is important to note that the displayed clusters represent only one second of the hesitation time series segments. Therefore, the clustering module should include an additional analysis parameter that determines the length of the aggregated training data. At this stage, a one-second length was chosen because the MLOps pipeline is still under development, and excessive data volumes would further strain the cloud instance resources, leading to longer analysis times. This is solely about computational power, as storage capacity is not a reason to shorten the aggregated training data. The following table shows the individual files where the aggregated data are stored.

Name	**Description**	**Memory**
dataset.json	JSON object of dataset	579 B

dist_matrix.pkl.npy	Pickle file with distance matrix of DTW analysis	200 MB
res_hesitation_train.pkl	Pickle file with training data for Autoencoder	36 MB
events_params.json	JSON object with hesitation analysis parameters	823 B
events.pkl	Pickle file with all hesitation events from raw data	36 MB
function_find_hesitations.pkl	Pickle file of UDV function to find hesitations	1.9 KB
plot_clusters.html	HTML file with plotted clusters of hesitations	10.9 MB
plot_io_nio.html	HTML file with labeled clusters and single events	10.8 MB

Table 4. Aggregated data divided in files for version management on ADLS.

The total storage used for the files amounts to approximately 293 megabytes (MB). In the next step, the clusters can be written into arrays based on their cluster number, marking the time series segments as in order (iO) or not in order (niO). The same applies to individual normal or anomalous hesitation events, which are written into either an iO or niO array. The training data for the autoencoder are saved in separate files for reuse and versioning, as shown in Table 2.

4.3.4.4 Model Training

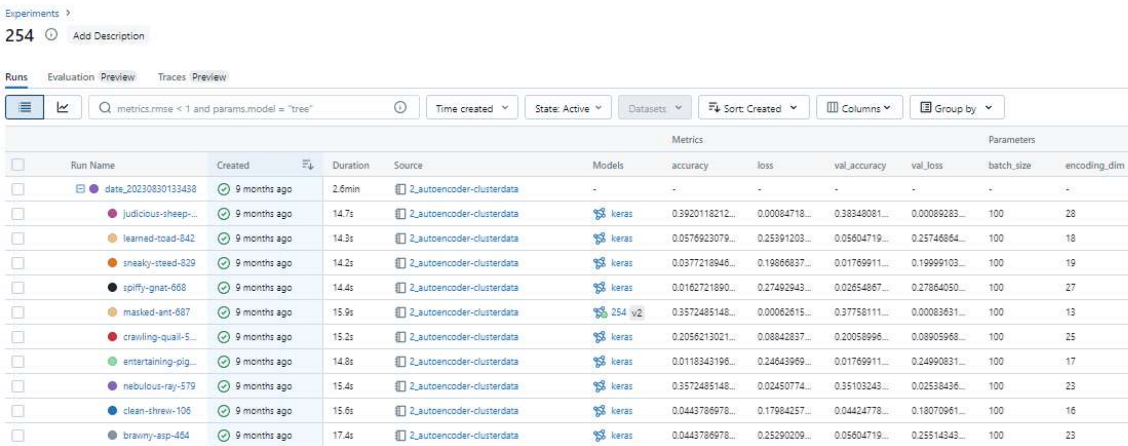
A control panel in the training notebook is used to select the required dataset. Subsequently, the analysis parameters and the training data from the Pickle file are loaded. Another module in this notebook allows for defining additional features and preparing them for model training. The new features are added to the JSON object events_params.json under the channel object. This process also involves a form of feature versioning, though not yet based on a feature store. The features are listed in the final report for each potential hesitation anomaly.

The notebook includes instructions for creating these new features. Generally, the features are chosen from the signals already extracted in the clustering notebook. If a feature from a new signal is needed, it must first be defined in the clustering notebook and extracted from the raw data. The creation of a new feature can be determined by the mapping name of the signal, for example, CHAN_PED. Using the max() function, a start and end point (t_start and t_end), and data cleaning, CHAN_PED can be constrained. The specific feature is called APP_max_0_2_mean and consists of CHAN_PED, within an interval from 0 to 2 seconds, cleaned by the mean parameter. The mean parameter replaces any data gaps or Not a Number (NaN) values with the average value of all CHAN_PED values.

In total, 9855 hesitation events were labeled as iO out of a maximum of 10,000 events and are now used for model training. Parallel to this, logging is initiated based on MLFlow to manage the respective training models.

Next, the training-test ratio of the hesitation events is set to 0.2. The initialized autoencoder class stores both the input and output data for training and testing the model.

After 10 iterations of model training, the best models are determined based on the lowest validation loss. The number of iterations can be increased, but for initial validation, it is set to MAX_EVALS = 10. If the model does not meet the performance metrics, the number of iterations can be increased. The notebook selects the best model based on the validation loss metric, which corresponds to the reconstruction error (MSE), and activates it in the inference app for report generation.

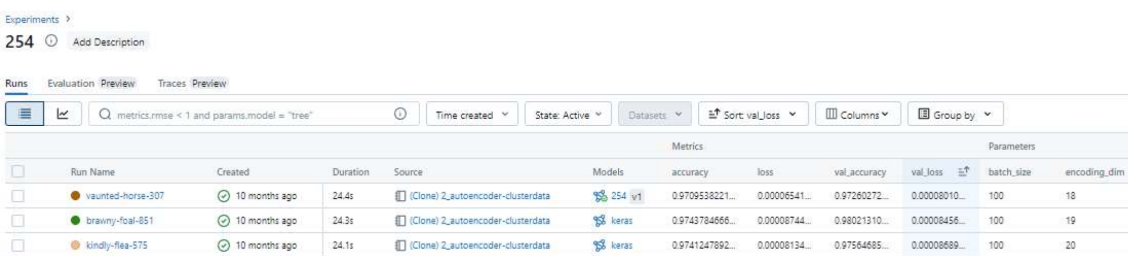


Experiments >
254 Add Description
Runs Evaluation Preview Traces Preview
metrics.rmse < 1 and params.model = "tree" | Time created | State: Active | Datasets | Sort: Created | Columns | Group by

Run Name	Created	Duration	Source	Models	accuracy	loss	val_accuracy	val_loss	batch_size	encoding_dim
date_20230830133438	9 months ago	2.6min	2_autoencoder-clusterdata	-	-	-	-	-	-	-
judicious-sheep-...	9 months ago	14.7s	2_autoencoder-clusterdata	keras	0.3920118212...	0.00084718...	0.38348081...	0.00089283...	100	28
learned-toad-842	9 months ago	14.3s	2_autoencoder-clusterdata	keras	0.0576923079...	0.25391203...	0.05604719...	0.25746864...	100	18
sneaky-steed-829	9 months ago	14.2s	2_autoencoder-clusterdata	keras	0.0377218946...	0.19866837...	0.01769911...	0.19999103...	100	19
spiffy-gnat-668	9 months ago	14.4s	2_autoencoder-clusterdata	keras	0.0162721890...	0.27492943...	0.02654867...	0.27864050...	100	27
masked-ant-687	9 months ago	15.9s	2_autoencoder-clusterdata	254 v2	0.3572485148...	0.00062615...	0.37758111...	0.00083631...	100	13
crawling-quail-5...	9 months ago	15.2s	2_autoencoder-clusterdata	keras	0.2056213021...	0.08842837...	0.20058996...	0.08905968...	100	25
entertaining-pig...	9 months ago	14.8s	2_autoencoder-clusterdata	keras	0.0118343196...	0.24643969...	0.01769911...	0.24990831...	100	17
nebulous-ray-579	9 months ago	15.4s	2_autoencoder-clusterdata	keras	0.3572485148...	0.02450774...	0.35103243...	0.02538436...	100	23
clean-shrew-106	9 months ago	15.6s	2_autoencoder-clusterdata	keras	0.0443786978...	0.17984257...	0.04424778...	0.18070961...	100	16
brawny-asp-464	9 months ago	17.4s	2_autoencoder-clusterdata	keras	0.0443786978...	0.25290209...	0.05604719...	0.25514343...	100	23

Figure 20. Listed models with performance metrics for 10 iterations.

Since the performance of the accuracy and validation accuracy was relatively low after 10 iterations, another attempt was conducted with 100 iterations for comparison, as shown in figure 21. The performance for both metrics exceeds 97%, and the validation loss is reduced by an order of magnitude.



Experiments >
254 Add Description
Runs Evaluation Preview Traces Preview
metrics.rmse < 1 and params.model = "tree" | Time created | State: Active | Datasets | Sort: val_loss | Columns | Group by

Run Name	Created	Duration	Source	Models	accuracy	loss	val_accuracy	val_loss	batch_size	encoding_dim
vaunted-horse-307	10 months ago	24.4s	(Clone) 2_autoencoder-clusterdata	254 v1	0.9709538221...	0.00006541...	0.97260272...	0.00008010...	100	18
brawny-foal-851	10 months ago	24.3s	(Clone) 2_autoencoder-clusterdata	keras	0.9743784666...	0.00008744...	0.98021310...	0.00008456...	100	19
kindly-flea-575	10 months ago	24.1s	(Clone) 2_autoencoder-clusterdata	keras	0.9741247892...	0.00008134...	0.97564685...	0.00008689...	100	20

Figure 21. Listed models with performance metrics for 100 iterations.

The results of testing the autoencoder model are displayed. In figure 22, it is clear how the good data are almost perfectly reconstructed. Each curve can be easily matched to its reconstructed counterpart, which is not the case in figure 23.

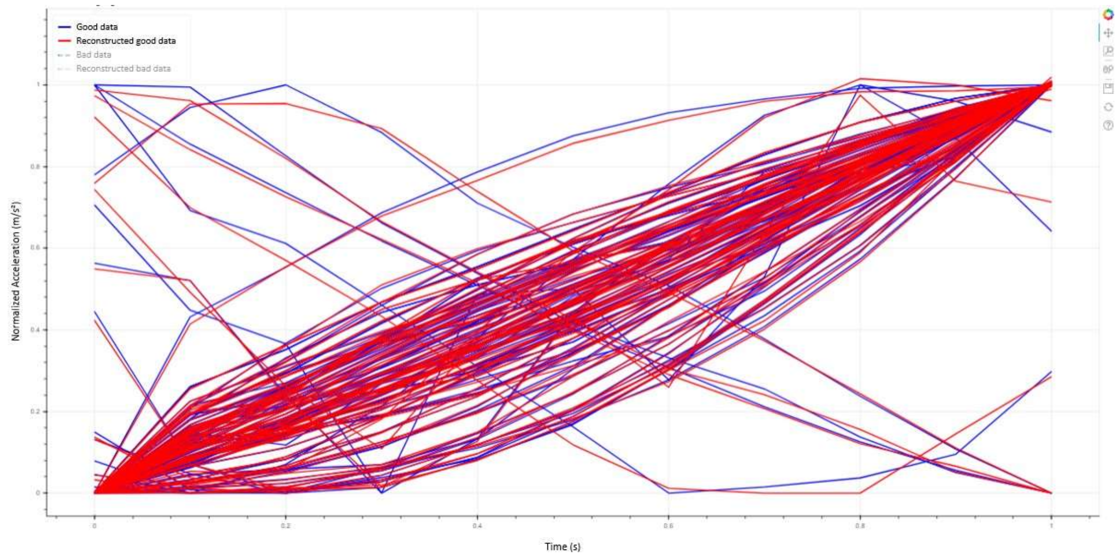


Figure 22. Normalized good, reconstructed hesitation events after testing Autoencoder model.

The erroneous reconstructions cannot be matched to the input data because the reconstruction error for these curves must have been high. The poorly reconstructed data can be loaded in the next step through the inference app and individually displayed in a report.

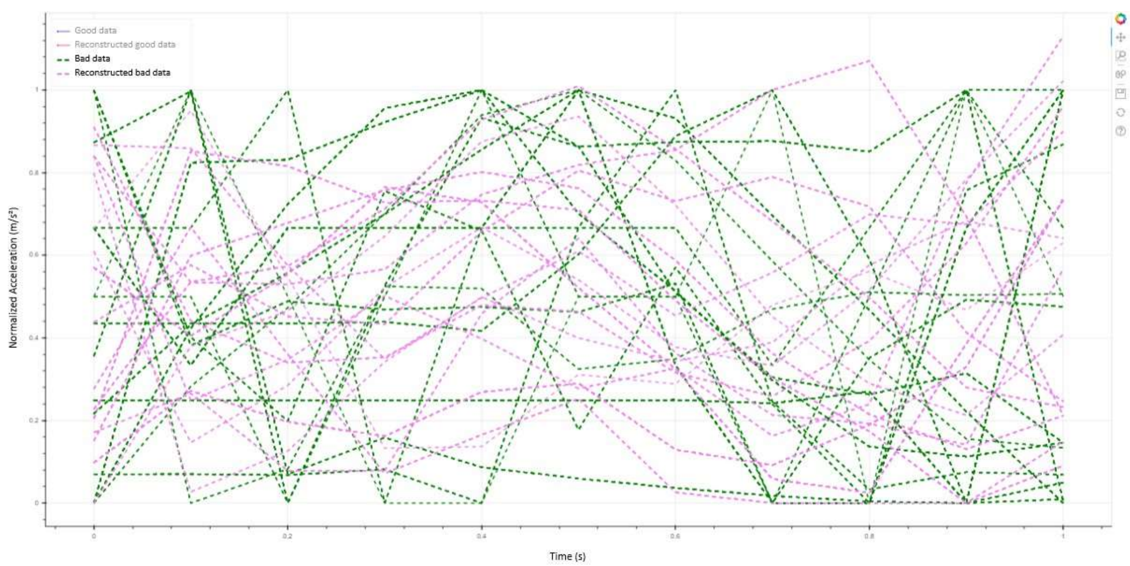
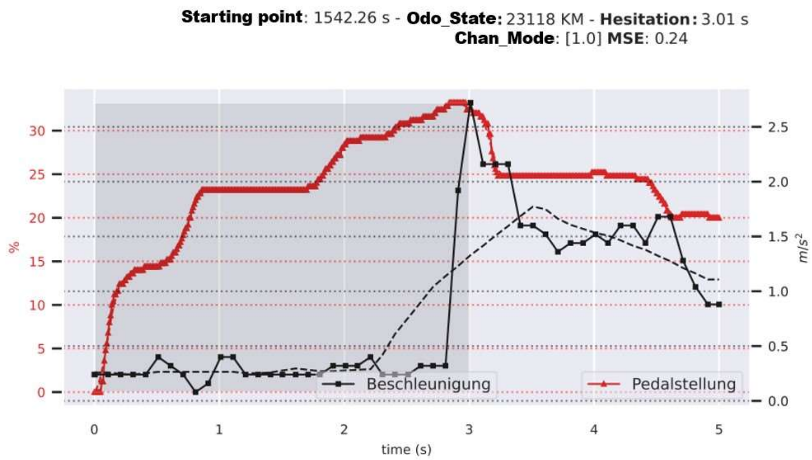


Figure 23. Normalized bad, reconstructed hesitation events after testing Autoencoder model.

4.3.4.5 Inference App

The report is generated in Portable Document Format (PDF). The dataset used for this can be produced through the Data Provisioning module (see figure 15). Clustering and model training are not necessary, as the evaluated model is already activated in the application. The dataset is created and selected in the inference app via a dropdown menu. The analysis starts and generates a web job, the results of which are documented in the report.

The first four pages contain model information, analysis parameters for detecting hesitation events, the vehicles examined, standard features, as well as user-defined features. Additionally, the MSE, the odometer reading at the time of a hesitation, and the user-defined features are compared using a pair plot. This provides users with the means for root cause analysis if the information from the other features is insufficient.



	Value	Unit
Standard Feature		
APP_diff	33.11	%
APP_max	33.2	%
APP_t_max	3.51	s
APP_t_max_half	2.71	s
cor_APP_v_LongAccel_Sens	0.29	-
dtw_APP_v_LongAccel_Sens	157.64	-
LongAccel_Sens_max	1.77	m/s^2
LongAccel_Sens_t_max	3.51	s
delay_APP_v_LongAccel_Sens	0.9	s
Gear_State_min	0.0	-
Gear_State_max	1.0	-
Gear_State_upshifts	1	-
Gear_State_downshifts	0	-
Eng_Spd_min	555.0	1/min
Eng_Spd_max	1315.0	1/min
Eng_Spd_avg	1072.89	1/min
Veh_Spd_min	0.0	m/s
Veh_Spd_max	9.19	m/s
Veh_Spd_avg	1.84	m/s
Self Defined Features		
APP__max_0_2_mean	28.0	%
LongAccel_Sens__avg_0_2_mean	0.26	m/s2
Veh_Spd__min_2_3_mean	0.0	m/s
Eng_Trq__max_0_4_mean	120.5	Nm

Figure 24. Extract of report with noticeable hesitation event.

In Figure 24, a hesitation event is depicted that can be evaluated as a potential anomaly. Comparing the MSE of this hesitation with that of the others, it is noticeable that the MSE is slightly higher here. It is evident that the vehicle's acceleration remains slightly above 0 and constant for almost 3 seconds, despite the accelerator pedal position exceeding 30%. The vehicle speed in the time interval from 2 to 3 seconds (Veh_Spd_min_2_3_mean) is 0, indicating that the vehicle is stationary. However, there is a gear shift, although it is not specified which gear the vehicle is in currently. It can be assumed that either a special test was started during the trial, measuring the engine at idle and altering a map to keep the vehicle stationary, or the handbrake was applied during acceleration. Given that the vehicle under test has an electric handbrake, further investigation is needed to understand why it releases only after approximately 3 seconds.

This report allows further potential hesitation anomalies to be discussed and addressed within a working group. The result thus offers a small number of hesitation events based on predefined features that can be dealt with by a group of experts. This saves time in aggregating and analyzing the measurement data from the endurance tests. If the listed hesitation events or existing features are insufficient, a deeper root cause analysis of the original measurement can be conducted at any time via the filename, by evaluating additional signals. These signals can then be introduced as new features into the autoencoder model, providing a quickly scalable method for modifying Anomaly Detection models for time series data from endurance tests.

4.3.5 Discussion

Fundamentally, the current state of the MLOps pipeline for detecting potential anomalies in endurance test measurement data (time series data) is a valuable tool for engine experts, allowing for data provision from measurement data, model training, monitoring of model metrics, and version control. It should be noted that the entire pipeline can be utilized more efficiently if individual modules of the pipeline can be distributed to different working groups, such as Business Engineering, Data Engineering, Data Science, and Engine Development. Assigning tasks to the respective working groups promotes the transition of models from prototype to production phase, as each working group focuses on their area of expertise. Therefore, the use of a multi-faceted MLOps pipeline should be interdisciplinary and allow for cross-functional collaboration between the task areas (Mäkinen et al., 2021), (John et al., 2021).

Extensions such as a feature store, data repository, code repository, metadata store, model registry, and feedback loops enhance automation, versioning, explainability, and traceability (Zhou et al., 2020), (John et al., 2021). In the current project, the MLOps pipeline has a relatively low maturity level, as it is the first version for implementing Anomaly Detection from engine development measurement data. For this reason, a feature and metadata store were omitted. The data repository is in a Blob Storage of the company's private cloud. Rules for creating datasets related to naming conventions and data duplication for the hesitation use case have been established for the repository. The code repository uses the accessible service of Github, which is only accessible to company employees. The model registry is managed via MLFlow. Further optimization scopes for the mentioned modules are necessary, depending on the Machine Learning model itself. If the model is enriched with additional information such as metadata, parameters, or model metrics, a more differentiated management of data, source code, and Machine Learning models can be implemented. The initial results of the MLOps pipeline in the form of data provision and reporting were positively received by the working groups. The focus is on the handy and quick reduction of large measurement data sets from the endurance test and the generation of the clear report that shows potential anomalies in the measurement data. However, it is necessary to examine in a further research project the measurable benefit the MLOps pipeline provides for engine development.

4.3.6 Conclusion

In conclusion, the application of MLOps in engine development represents a significant advancement. By combining automated data pipelines with advanced ML algorithms, large volumes of time series measurement data can be efficiently compressed and analyzed. Additionally, the provision of new deployments of Machine Learning models enables rapid Anomaly Detection, providing valuable insights to vehicle experts. Engine development focuses on its core competency, the development of new engines. At the same time, engine experts can quickly process and evaluate large data sets, implement minor and major model changes in production agilely, and manage different models based on model metrics. The implementation of an MLOps pipeline thus significantly contributes to increasing efficiency and reliability in engine development.

5 Conclusion and Outlook

5.1 Summary of Findings

The thesis developed a systemic classification of anomalies, particularly focusing on hesitations in endurance testing time series data of engines. This classification addresses the misleading understanding of distinct Anomaly Detection methods in the literature and offers a detailed taxonomy derived from time series data. The taxonomy allows for the classification of previously unknown events in measurement data, enabling quicker identification and resolution of anomalies. Additionally, the thesis describes the implementation of ML models to detect anomalies in engine data, demonstrating the practical application of these models regardless of the propulsion system. Additionally, the thesis demonstrates that an automated pipeline for machine learning models can be integrated into engine development to process measurement data quickly and provide significant insights. Not only the technical implementation is addressed, but also the potential impact on the work culture of cross-functional groups.

The implementation of an MLOps pipeline in engine development represents a significant advancement, allowing for efficient compression and analysis of large volumes of time series measurement data. The thesis contributes to both theoretical knowledge and practical applications by providing a structured approach to Anomaly Detection, potentially leading to significant efficiency and reliability improvements in engine development. Further research is needed to explore the full potential and applicability of the developed methods and tools concerning their perceived usefulness and perceived ease of use.

5.2 Contribution to Theory and Practice

The thesis provides a new taxonomy for classifying engine-related anomalies, offering a structured method for future research in Anomaly Detection. This taxonomy addressed gaps in existing literature regarding the correct usage of Anomaly Detection methods and operates as an extension of Carreno et al. (2020) especially for time series data in engine development. It offers a detailed classification for endurance testing data of vehicle fleets. By using the current taxonomy, the contributors in engine development and other cross-functional groups can get a correct understanding of peculiarities that appear in the measurement data since the mixing of different anomaly terms is often not clarified.

The work also enhances the understanding of anomalies in highly networked vehicles, contributing to the theoretical framework of Anomaly Detection in automotive contexts. Here, the DTW method was used to align hesitation events from the measurement data. Based on the DTW distance metric, clustering algorithms such as PIC, DBSCAN, and AHC can be used to obtain results giving meaningful insights to engineers in engine development. For this reason, a generic approach to identifying similarities in the curve progressions of existing hesitations was developed. This approach can be applied theoretically, but particularly in practice, to various events in the measurement data and different types of propulsion systems.

The practical application of the ML models and the MLOps pipeline demonstrated their utility in real-world engine development, improving the efficiency of Anomaly Detection. Active engagement with engine experts and iterative development cycles ensured that the tools meet practical needs, resulting in precise and error-free analyses. Early detection of anomalies during the development phase can significantly reduce future costs related to customer complaints and vehicle repairs (Bamberg, 2018).

5.3 Managerial Implications

As the automotive industry continues to evolve, the need for efficient and streamlined MLOps has become increasingly apparent. Automotive engine development, in particular, is a domain where time series data plays a crucial role, and the successful adaptation of MLOps can have significant managerial implications.

One of the key managerial considerations is the need for a strong collaborative culture between cross-functional teams, including data scientists, Machine Learning engineers, front-end engineers, and domain experts (Symeonidis et al., 2022). The traditional isolated approach to Machine Learning development often leads to compatibility issues and technical challenges, which can hinder the successful deployment of ML models in production (Symeonidis et al., 2022). To address this, a huge software company like Google has often adopted a "swarm" approach, where cross-functional teams work together on specific use cases, fostering a more integrated and agile development process (Kreuzberger et al., 2022), (Hewage and Meedeniya, 2022), (Pichai, 2021), (Bayucca, 2024).

Another important aspect is the maturity level of the MLOps system (Symeonidis et al., 2022). A low-maturity system that relies on classical Machine Learning techniques requires a high level of coordination and communication between teams, which can be challenging to maintain (Symeonidis et al., 2022). Successful MLOps implementation in automotive engine development context would require the establishment of robust and efficient pipelines with strong compatibility, ensuring seamless transitions between the various stages of the ML lifecycle (Symeonidis et al., 2022), (Kreuzberger et al., 2022). Additionally, the ability to manage traceability and automate processes within the MLOps system is crucial for automotive engine development, (Hewage and Meedeniya, 2022). This can help streamline the development process, reduce manual efforts, and better estimate the resources required for successful deployment (Hewage and Meedeniya, 2022).

Another key contribution was provided, revealing that an even stronger, more iterative and collaborative processing of task packages or project milestones remains important for cross-functional use cases. In the current research project, the individual tasks were handled through weekly sprints, which had high complexity and therefore needed frequent discussions with all stakeholders of the working group. This continuous dialogue and iterative approach was necessary to ensure precision in requirement formulation, timely identification of potential problems, and the development of alternative solutions, as is already common practice in agile methodologies (Schön et al., 2017). Maintaining this level of cross-functional coordination and adaptability was crucial for the successful implementation of the MLOps pipeline within the automotive engine development context.

Active and continuous exchange between users in engine development and topics surrounding data processing enabled overcoming obstacles related to complex Machine Learning and engine systems. As mentioned previously, this led to improvements in PU and PEOU, thereby increasing the acceptance of Machine Learning systems to the engine domain experts and build-up engine domain knowhow for all data and ML experts. If extended to a broader context, this suggests that the digital transformation within an organisation can be promoted to some degree. The utilization of learning platforms such as Udemy to train employees on emerging areas like electrification and data analysis supports this observation. However, to make a more qualified statement about the impact on digital transformation, a more rigorous scientific investigation would be required.

Overall, the managerial implications of adapting MLOps for automotive engine development go beyond just the technical aspects. It requires a holistic approach that considers the organisational culture, team dynamics, and process maturity, in order to effectively harness the power of Machine Learning and drive innovation in the automotive industry.

5.4 Outlook

Further research is needed to comprehensively evaluate the perceived usefulness and ease of use of the MLOps pipeline in the context of automotive engine development. While the current project results have been consistently satisfactory, based on feedback from the project team and the direct use of the pipeline for developing new engines to meet the Euro 7 emission standard, a more in-depth and longitudinal scientific investigation is required to thoroughly assess the long-term maturity and widespread adoption of the MLOps pipeline after an extended period of usage. This could be implemented through a comprehensive survey of MLOps pipeline users across a diverse range of engine development projects and teams. Additionally, since automotive engine development inherently involves complex error diagnosis, analysis, and correction processes, a more advanced technical implementation to systematically identify the root causes of observed errors should be undertaken. Preliminary work has been done to include additional pair plots for mean squared error (MSE) and different engine performance measures in the Anomaly Detection reporting module, which can serve as a solid foundation for deeper and more nuanced analysis of the underlying causes of the observed errors. This could lead to the development of more effective and streamlined error correction strategies, thereby enhancing overall engine optimization and performance.

An initial prototype MLOps pipeline has been developed, which requires thorough testing and evaluation from both technical and organisational perspectives. The technical testing should involve users in the engine development process over an extended period to assess the pipeline's performance, robustness, and scalability. Additionally, it is crucial to observe the organisational impact of the MLOps pipeline, such as its effect on the work culture and the implementation of cross-functional working methods across different departments. Key questions to be explored include the integration of new processes and tools into existing workflows, the level of collaboration and knowledge sharing among teams, and the overall acceptance and adoption of the MLOps approach by employees. This comprehensive evaluation, encompassing both technical and organisational aspects, will help ensure the long-term success and sustainability of the MLOps implementation in the automotive engine development context.

References

Abdelrahman, O., & Keikhosrokiani, P. (2020). Assembly Line Anomaly Detection and Root Cause Analysis Using Machine Learning. IEEE Access, 8, 189661-189672. https://doi.org/10.1109/ACCESS.2020.3029826

Aggarwal, C. C. (2015). Outlier analysis. In Data Mining (pp. 237–263). Springer.

Aggarwal, C. C. (2017). Outlier analysis. Springer, Berlin.

Ahmed, C. M., R., G. R. M., & Mathur, A. P. (2020, October 6). Challenges in Machine Learning based approaches for Real-Time Anomaly Detection in Industrial Control Systems. https://doi.org/10.1145/3384941.3409588

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Türkmen, A. C., & Wang, Y. (2022). GluonTS: probabilistic and neural time series modeling in Python. Journal of Machine Learning Research, 21(1), Article 116.

Al Jallad, K., Aljnidi, M., & Desouki, M. S. (2019). Big Data analysis and distributed Deep Learning for next-generation intrusion detection system optimization. Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0248-6

Aldoseri, A., Al-Khalifa, K. N., & Hamouda, A. (2023, June 13). Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges. Multidisciplinary Digital Publishing Institute, 13(12), 7082-7082. https://doi.org/10.3390/app13127082

Amorim, A. C., Silva, M. M. D., Pereira, R., & Gonçalves, M. R. (2021, November 1). Using agile methodologies for adopting COBIT. Elsevier BV, 101, 101496-101496. https://doi.org/10.1016/j.is.2020.101496

Andrade, F. (2022). Is Data The New Oil of the 21st Century or Just an Overrated Asset? Towards Data Science. https://towardsdatascience.com/ (Accessed 01 November 2022).

Arena, F., Collotta, M., Luca, L., Ruggieri, M., & Termine, F. G. (2021, December 31). Predictive Maintenance in the Automotive Sector: A Literature Review. Multidisciplinary Digital Publishing Institute, 27(1), 2-2. https://doi.org/10.3390/mca27010002

Attenborough, K., & Postema, M. (2008). A pocket-sized introduction to dynamics. University of Hull. https://doi.org/10.5281/zenodo.7504154

Avizienis, A., Laprie, J.-C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 1(1), 11–33.

Bamberg, C. (2018). Vermeidung von Verschwendung als Grundprinzip bei der Stationsoptimierung. In C. Bamberg, N. Kasper, M. Korff, & R. Herbold (Eds.), Moderne Stationsorganisation im Krankenhaus (pp. 75-105). Springer. https://doi.org/10.1007/978-3-662-57536-9_5

Barai, A., & Lopamudra, D. (2017). Outlier detection and removal algorithm in K-means and hierarchical clustering. World Journal of Computer Application Technology, 5(2), 24–29.

Bayucca, R. (2024). Google Organisational Structure: Everything You Need To Know. Correctdigital.com. https://correctdigital.com/google-organisational-structure/

Bekar, T. E., Nyqvist, P., & Skoogh, A. (2020). An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study. SAGE Publishing, 12(5), 168781402091920-168781402091920. https://doi.org/10.1177/1687814020919207

Betz, J., et al. (2019). Autonomous Driving—A Crash Explained in Detail. Multidisciplinary Digital Publishing Institute, 9(23), 5126-5126. https://doi.org/10.3390/app9235126

Bhageshpur, P. (2019). Data Is The New Oil – And That's A Good Thing. Forbes Technology Council. https://www.forbes.com/ (Accessed 01 November 2022).

Bharadwaj, P. (2021). Why Do Electric Cars Accelerate Faster Than Internal Combustion Cars? DriveSpark. https://www.drivespark.com/ (Accessed 05 November 2022).

Bikakis, N., Papastefanatos, G., & Papaemmanouil, O. (2018). Big Data exploration, visualization and analytics. Big Data Research, 18, 100123. https://doi.org/10.1016/j.bdr.2019.100123

Bock, F., Siegl, S., & German, R. (2016). Mathematical Test Effort Estimation for Dependability Assessment of Sensor-Based Driver Assistance Systems. https://doi.org/10.1109/seaa.2016.49

Braei, M., & Wagner, S. (2020, January 1). Anomaly Detection in Univariate Time series: A Survey on the State-of-the-Art. Cornell University. https://doi.org/10.48550/arxiv.2004.00433

Breck, E., et al. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. https://doi.org/10.1109/bigdata.2017.8258038

Bughin, J., Hazan, E., Ramaswamy, S., Chui, M., Allas, T., Dahlström, P., Henke, N., & Trench, M. (2017, June 1). Artificial Intelligence: the next digital frontier?. https://apo.org.au/node/210501

Burkacky, O., Deichmann, J., & Stein, J. P. (2019). Mapping the automotive software-and-electronics landscape through 2030. McKinsey & Company. https://www.mckinsey.com/ (Accessed on 05 Jul 2022).

Capogrosso, L., Cunico, F., Cheng, D., Fummi, F., & Cristani, M. (2023, January 1). A Machine Learning-oriented Survey on Tiny Machine Learning. Cornell University. https://doi.org/10.48550/arxiv.2309.11932

Cardoso Silva, L., Rezende Zagatti, F., Silva Sette, B., Nildaimon dos Santos Silva, L., Lucredio, D., Furtado Silva, D., & de Medeiros Caseli, H. (2020). Benchmarking Machine Learning Solutions in Production. 19th IEEE International Conference on Machine Learning and Applications (ICMLA). https://doi.org/10.1109/icmla51294.2020.00104

Carreno, A., Inza, I., & Lozano, J. A. (2020). Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. Artificial Intelligence Review, 53(3575-3594). https://doi.org/10.1007/s10462-019-09771-y

Chahla, C., Snoussi, H., Merghem, L., & Esseghir, M. (2019). A novel approach for Anomaly Detection in power consumption data. Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, 483-490. https://doi.org/10.5220/0007361704830490

Chakraborty, N., Hasan, A., Liu, S., Ji, T., Liang, W., McPherson, D. L., & Driggs-Campbell, K. (2023, January 1). Structural Attention-Based Recurrent Variational Autoencoder for Highway Vehicle Anomaly Detection. Cornell University. https://doi.org/10.48550/arxiv.2301.03634

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: a survey, association for computing machinery. ACM Computing Surveys, 41(3), 15:1–15:58. https://doi.org/10.1145/1541880.1541882

Chidambaram, S., Maheswaran, Y., Patel, K., Sounderajah, V., Hashimoto, D. A., Seastedt, K. P., McGregor, A. H., Markar, S., & Darzi, A. (2022, September 13). Using Artificial Intelligence-Enhanced Sensing and Wearable Technology in Sports Medicine and Performance Optimisation. Multidisciplinary Digital Publishing Institute, 22(18), 6920-6920. https://doi.org/10.3390/s22186920

Choi, K., Yi, J., Park, C., & Yoon, S. (2021, January 1). Deep Learning for Anomaly Detection in Time series Data: Review, Analysis, and Guidelines. Institute of Electrical and Electronics Engineers, 9, 120043-120065. https://doi.org/10.1109/access.2021.3107975

Cornet, A., Conzade, J., Schaufuss, P., Schenk, S., Tschiesner, A., Hensley, R., Hertzke, P., Heuss, R., von Laufenberg, K., & Möller, T. (2021). Why the automotive future is electric. McKinsey & Company. https://www.mckinsey.com/ (Accessed 17 February 2023).

Cross, N. (1993, January 1). A History of Design Methodology. Springer Nature, 15-27. https://doi.org/10.1007/978-94-015-8220-9_2

Cui, M. (2020). Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. Geoscience and Remote Sensing, 3, 9-16. https://doi.org/10.23977/geors.2020.030102

Dai, X., & Gao, Z. (2013). From Model, Signal to Knowledge: A Data-Driven Perspective of Fault Detection and Diagnosis. IEEE Transactions on Industrial Informatics, 9(4), 2226-2238. https://doi.org/10.1109/TII.2013.2243743

D'Alconzo, A., Drago, I., Morichetta, A., Mellia, M., & Casas, P. (2019). A Survey on Big Data for Network Traffic Monitoring and Analysis. IEEE Transactions on Network and Service Management, 16(3), 800-813. https://doi.org/10.1109/TNSM.2019.2933358

Davis, F. D. (1993, March 1). User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. Academic Press, 38(3), 475-487. https://doi.org/10.1006/imms.1993.1022

Demirbaga, U., & Aujla, G. S. (2024). RootPath: Root Cause and Critical Path Analysis to Ensure Sustainable and Resilient Consumer-Centric Big Data Processing Under Fault Scenarios. IEEE Transactions on Consumer Electronics, 70(1), 1493-1500. https://doi.org/10.1109/TCE.2023.3329545

Dou, S., Yang, K., & Poor, H. V. (2019). PC^2A: Predicting Collective Contextual Anomalies via LSTM with Deep Generative Model. IEEE Internet of Things Journal.

Duque, J., et al. (2024). Data Science with Data Mining and Machine Learning: A design science research approach.

Dzierma, Y., & Wehrmann, H. (2010). Eruption time series statistically examined: probabilities of future eruptions at Villarrica and Llaima Volcanoes, Southern Volcanic Zone, Chile. Journal of Volcanology and Geothermal Research, 193(1–2), 82–92.

Einarsdóttir, H., Emerson, M. J., Clemmensen, L. H., Scherer, K., Willer, K., Bech, M., Larsen, R., Ersbøll, B. K., & Pfeiffer, F. (2016). Novelty detection of foreign objects in food using multi-modal x-ray imaging. Food Control, 67, 39–47.

Erba, A., Taormina, R., Galelli, S., Pogliani, M., Carminati, M., Zanero, S., & Tippenhauer, N. O. (2019, July 17). Real-time Evasion Attacks with Physical Constraints on Deep Learning-based Anomaly Detectors in Industrial Control Systems. Cornell University. https://arxiv.org/abs/1907.07487v2

Esling, P., & Agon, C. (2012). Time series data mining. ACM Computing Surveys, 45(1), 1–34.

European Commission. (2020). European vehicle emissions standards – Euro 7 for cars, vans, lorries and buses. Have your say. https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/12313-European-vehicle-emissions-standards-Euro-7-for-cars-vans-lorries-and-buses_en

Fan, S., Liu, G., & Chen, Z. (2017). Anomaly Detection methods for bankruptcy prediction. 4th International Conference on Systems and Informatics (ICSAI), pp 1456–1460.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurasamy, R. (1996). Advances in Knowledge Discovery and Data Mining. Cambridge, London: MIT press.

Fernández-Robin, C., Yáñez, D., & McCoy, S. (2019, April 24). Intention to Use WhatsApp. IntechOpen. https://doi.org/10.5772/intechopen.81999

Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2017). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences, 479, 448–455.

GluonTS: Probabilistic and neural time series modeling in Python. (2022). Journal of Machine Learning Research, 21(1), Article 116.

Haensch, W., Raghunathan, A., Roy, K., Chakrabarti, B., Phatak, C., Wang, C., & Guha, S. (2022, January 1). A Co-design view of Compute in-Memory with Non-Volatile Elements for Neural Networks. Cornell University. https://doi.org/10.48550/arxiv.2206.08735

Hamilton, J. D. (1994). Time series analysis. Princeton University Press.

Heard, N. A., Weston, D. J., Platanioti, K., & Hand, D. J. (2010). Bayesian Anomaly Detection methods for social networks. Annals of Applied Statistics, 4(2), 645–662.

Hevner, A., Brocke, J., & Maedche, A. (2018). Roles of Digital Innovation in Design Science Research. MIS Quarterly, 28(1), 75-105.

Hewage, N. T., & Meedeniya, D. (2022, January 1). Machine Learning Operations: A Survey on MLOps Tool Support. Cornell University. https://doi.org/10.48550/arxiv.2202.10169

Himeur, Y., Elnour, M., Fadli, F., et al. (2023). AI-Big Data analytics for building automation and management systems: a survey, actual challenges and future perspectives. Artificial Intelligence Review, 56, 4929–5021. https://doi.org/10.1007/s10462-022-10286-2

Hinton, E. G., & Salakhutdinov, R. (2006). Reducing the Dimensionality of Data with Neural Networks. American Association for the Advancement of Science, 313(5786), 504-507. https://doi.org/10.1126/science.1127647

Hlongwane, S., & Grobbelaar, S. S. (2022, June 2). A Practical Framework for Value Creation in Health Information Systems From an Ecosystem Perspective: Evaluated in the South African Context. Frontiers Media, 13. https://doi.org/10.3389/fpsyg.2022.637883

Hof, R. D. (2013). Deep Learning. MIT Technology Review. https://www.technologyreview.com/s/513696/deep-learning/ (Accessed 10 August 2022).

Holden, R. J., & Karsh, B. T. (2010). The technology acceptance model: its past and its future in health care. Journal of biomedical informatics, 43(1), 159–172. https://doi.org/10.1016/j.jbi.2009.07.002

Ikhlaq, U., & Kechadi, T. (2023). Harnessing Big Data in Agriculture by Addressing Heterogeneity in Large-Scale Data Mining Techniques and Limitations. Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, 1439-1445. https://doi.org/10.1109/CSCE60160.2023.00239

Jeong, Y., Yang, E. Y., Ryu, J. H., Park, I., & Kang, M. (2023, January 1). AnomalyBERT: Self-Supervised Transformer for Time Series Anomaly Detection using Data Degradation Scheme. Cornell University. https://doi.org/10.48550/arxiv.2305.04468

Jin, X., Wah, B., Cheng, X., & Wang, Y. (2015). Significance and challenges of Big Data research. Big Data Research, 2(2), 59-64.

John, M. M., Olsson, H. H., & Bosch, J. (2021). Towards MLOps: A Framework and Maturity Model. In 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Palermo, Italy, 2021, pp. 1-8. https://doi.org/10.1109/SEAA53835.2021.00050

Jones, P. (2022). Electric Cars Feel Faster – Here's Why (Explained). Motor and Wheels. https://motorandwheels.com/why-do-electric-cars-feel-faster-explained/ (Accessed 05 November 2022).

Jablonski, G. (2020). VGG Based Unsupervised Anomaly Detection in Multivariate Time Series. In A. Bartoszewicz, J. Kabziński, & J. Kacprzyk (Eds.), Advanced, Contemporary Control. Advances in Intelligent Systems and Computing, 1196. Springer, Cham. https://doi.org/10.1007/978-3-030-50936-1_107

Kafkas, A., & Montaldi, D. (2018). How do memory systems detect and respond to novelty? Neuroscience Letters, 680, 60–68.

Keogh, E., & Pazzani, J. M. (2001). Derivative Dynamic Time Warping. https://doi.org/10.1137/1.9781611972719.1

Kirchen, I., et al. (2017). Metrics for the evaluation of data quality of signal data in industrial processes. https://doi.org/10.1109/indin.2017.8104878

Köknar-Tezel, S., & Latecki, L. J. (2011). Improving SVM classification on imbalanced time series data sets with ghost points. Knowledge and Information Systems, 28(1), 1–23.

Kreuzberger, D., Kühl, N., & Hirschl, S. (2022). Machine Learning Operations (MLOps): Overview, Definition, and Architecture. Cornell University. https://doi.org/10.48550/arxiv.2205.02302

Kumar, P., Lokesh, S., Varatharajan, R., Babu, G. C., & Parthasarathy, P. (2018). Cloud and IoT based disease prediction and diagnosis system for healthcare using Fuzzy neural classifier. Future Generation Computer Systems, 86, 527-534.

L'Heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. M. (2017). Machine Learning With Big Data: Challenges and Approaches. IEEE Access, 5, 7776-7797. https://doi.org/10.1109/ACCESS.2017.2696365

Li, G., Wang, Y., Li, M., & Wu, Z. (2008). Similarity Match in Time Series Streams under Dynamic Time Warping Distance. IEEE International Conference on Computer Science and Software Engineering. https://doi.org/10.1109/CSSE.2008.1117

Liang, L., Hao, Y., & Li, G. Y. (2019, February 1). Toward Intelligent Vehicular Networks: A Machine Learning Framework. Institute of Electrical and Electronics Engineers, 6(1), 124-135. https://doi.org/10.1109/jiot.2018.2872122

Lin, F., & Cohen, W. (2010). Power Iteration Clustering. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, June 21-24, 2010 (pp. 655-662).

Liu, L. (2009, December 1). Technology Acceptance Model: A replicated test using TETRAD. Wiley-Blackwell, 24(12), 1230-1242. https://doi.org/10.1002/int.20382

Liu, X., & Nielsen, P. S. (2020). Regression-based online Anomaly Detection for smart grid data. arXiv:1606.05781. https://arxiv.org/abs/1606.05781

Luckow, A., Kennedy, K., Manhardt, F., Djerekarov, E., Vorster, B., & Apon, A. (2015, October 1). Automotive Big Data: Applications, workloads and infrastructures. https://doi.org/10.1109/bigdata.2015.7363874

Mäkinen, S., Skogström, H., Laaksonen, E., & Mikkonen, T. (2021). Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? In 2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN), Madrid, Spain, 2021, pp. 109-112. https://doi.org/10.1109/WAIN52551.2021.00024

Mazur, M., Seyfferth, J., & Hörning, J. (2021). Accelerating the transformation to e-mobility. PwC Central and Eastern Europe. https://www.pwc.com/c1/en (Accessed on 17 February 2023).

Mazur, M., Seyfferth, J., & Hörning, J. (2021). Accelerating the transformation to e-mobility. PwC Central and Eastern Europe. https://www.pwc.com/c1/en (Accessed on 17 February 2023).

Mengyao, C. (2020). Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. Geoscience and Remote Sensing, 3, 9-16. https://doi.org/10.23977/geors.2020.030102

Mihaela, M., & Florin Daniel, M. (2022). Design Science Research Framework for Performance Analysis Using Machine Learning Techniques. Electronics, 11(16), 2504. https://doi.org/10.3390/electronics11162504

Mitchell, T. M. (1997). Machine Learning. McGraw-Hill, New York.

Miri Rostami, S., & Ahmadzadeh, M. (2018). Extracting predictor variables to construct breast cancer survivability model with class imbalance problem. Shahrood University of Technology, 6(2), 263–276.

Mitchell, C. (2023). Identity Theft Prediction Model using Historical Data and Supervised Machine Learning: Design Science Research Study.

Morris, M. G., & Dillon, A. (1997, January 1). How user perceptions influence software use. IEEE Computer Society, 14(4), 58-65. https://doi.org/10.1109/52.595956

Nasr Azadani, M., & Boukerche, A. (2024). Driving Behavior Analysis Guidelines for Intelligent Transportation Systems. IEEE Transactions on Intelligent Transportation Systems, 23(7), 6027-6045. https://doi.org/10.1109/TITS.2021.3076140

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. (2021, January 1). Machine Learning for Anomaly Detection: A Systematic Review. Institute of Electrical and Electronics Engineers, 9, 78658-78700. https://doi.org/10.1109/access.2021.3083060

Nason, G. P. (2006). Stationary and non-stationary time series. In Statistics in volcanology (p. 60).

Nazari, Z., Kang, D., Asharif, M. R., Sung, Y., & Ogawa, S. (2015). A new hierarchical clustering algorithm. In 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, 2015, pp. 148-152. https://doi.org/10.1109/ICIIBMS.2015.7439517

Neamtu, R., Ahsan, R., Rundensteiner, E. A., Sarkozy, G., Keogh, E., & Nguyen, C. (2018). Generalized Dynamic Time Warping: Unleashing the Warping Power Hidden in Point-Wise Distances. In IEEE 34th International Conference on Data Engineering (ICDE), https://doi.org/10.1109/ICDE.2018.00054

Nghiem, T. X., Drgoňa, J., Jones, C. N., Nagy, Z., Schwan, R., Dey, B., Chakrabarty, A., Cairano, S. D., Paulson, J. A., Carron, A., Zeilinger, M. N., Cortez, W. S., & Vrabie, D. (2023, May 31). Physics-Informed Machine Learning for Modeling and Control of Dynamical Systems. https://doi.org/10.23919/acc55779.2023.10155901

O'Donovan, P., Leahy, K., Bruton, K., & O'Sullivan, D. T. J. (2015). An industrial Big Data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. Journal of Big Data, 2(1), 25.

Oliveira, N., Sousa, N., Oliveira, J., & Praça, I. (2021, December 15). Anomaly Detection in Cyber-Physical Systems: Reconstruction of a Prediction Error Feature Space. https://doi.org/10.1109/sin54109.2021.9699339

O'Donovan, P., Leahy, K., Bruton, K., & O'Sullivan, D. T. J. (2015). An industrial Big Data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. Journal of Big Data, 2(1), 25.

Oesterle, H., Becker, J., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A., & Sinz, E. (2010). Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. http://www.alexandria.unisg.ch/Publikationen/71074. 62. https://doi.org/10.1007/BF03372838

Oentaryo, R., Lim, E.-P., Finegold, M., Lo, D., Zhu, F., Phua, C., Cheu, E.-Y., Yap, G.-E., Sim, K., Nguyen, M. N., Perera, K., Neupane, B., Faisal, M., Aung, Z., Woon, W. L., Chen, W., Patel, D., & Berrar, D. (2014). Detecting click fraud in online advertising: a data mining approach. Journal of Machine Learning Research, 15(1), 99–140.

Oppenheimer, D., Ganapathi, A., & Patterson, D. A. (2003). Why do Internet services fail, and what can be done about it?. USENIX Symposium on Internet Technologies and Systems, 67, Seattle, WA.

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. Journal of King Saud University – Computer and Information Sciences, 30(4), 431 – 448.

Pan, D., Liu, D., Zhou, J., & Zhang, G. (2015). Anomaly Detection for satellite power subsystem with associated rules based on Kernel Principal Component Analysis. Microelectronics Reliability, 55(2015), 2082 – 2086.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2020). Design science research process: A model for producing and presenting information systems research. arXiv preprint arXiv:2006.02763.

Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A Comprehensive Survey of Data Mining-based Fraud Detection Research. arXiv:1009.6119v1. https://arxiv.org/abs/1009.6119.

Pichai, S. (2021). A hybrid approach to work. Blog.google. https://blog.google/inside-google/life-at-google/hybrid-approach-work/

Pimentil, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2020). A review of novelty detection. Signal Processing, 99, 215-249. https://doi.org/10.1016/j.sigpro.2013.12.026.

Ribeiro, R. P., Pereira, P., & Gama, J. (2016). Sequential anomalies: a study in the Railway Industry. Machine Learning, 105, 127–153. https://doi.org/10.1007/s10994-016-5584-6.

Roelofs, C. M. A., Lutz, M.-A., Faulstich, S., & Vogt, S. (2021). Autoencoder-based anomaly root cause analysis for wind turbines. Energy and AI, 4, 100065. https://doi.org/10.1016/j.egyai.2021.100065

Rousseeuw, P. J., Leroy, A. M., Wiley, J., Sons, W. I. S., & Hubert, M. (2011). Robust statistics for outlier detection. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1(1), 73–79.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., & Müller, K. (2021, May 1). A Unifying Review of Deep and Shallow Anomaly Detection. Institute of Electrical and Electronics Engineers, 109(5), 756-795. https://doi.org/10.1109/jproc.2021.3052449

Samsung Newsroom. (2015). Eight Major States to Semiconductor Fabrication, Part 1: Creating the Wafer. Samsung Newsroom. https://news.samsung.com/global/ (Accessed 10 August 2022).

Sargolzaei, S. (2017, May 24). Developing Technology Acceptance Models for Decision Making in Urban Management. MedCrave Group, 2(6). https://doi.org/10.15406/mojce.2017.02.00050

Schön, E., Winter, D., Cuaresma, M J E., & Thomaschewski, J. (2017, January 1). Key Challenges in Agile Requirements Engineering. Springer Science+Business Media, 37-51. https://doi.org/10.1007/978-3-319-57633-6_3

Sculley, D., et al. (2015). Hidden technical debt in Machine Learning systems. NIPS Proceedings, 2503-2511. https://papers.nips.cc/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf

Shah, S. Y., Dang, X.-H., & Zerfos, P. (2019). Root Cause Detection using Dynamic Dependency Graphs from Time Series Data. IEEE International Conference on Big Data (Big Data), pp. 1998 – 2003.

Shams, S. M. R., & Solima, L. (2019). Big Data management: implications of dynamic capabilities and data incubator. Management Decision. https://doi.org/10.1108/MD-07-2018-0846

Shams, S. M. R., Vrontis, D., & Christofi, M. (2022). Stakeholder Causal Scope Analysis–Centered Big Data Management for Sustainable Tourism. Journal of Hospitality & Tourism Research, 46(5), 972–978. https://doi.org/10.1177/1096348020978547

Simon, D., & Ferreira, D. (2022). Längsdynamik. Internal Source.

Sircar, A., Yadav, K., Rayavarapu, K., Bist, N., & Oza, H. (2021, December 1). Application of Machine Learning and Artificial Intelligence in oil and gas industry. Elsevier BV, 6(4), 379-391. https://doi.org/10.1016/j.ptlrs.2021.05.009

Song, X., Wu, M., Jermaine, C., & Ranka, S. (2007). Conditional Anomaly Detection. IEEE Transactions on Knowledge and Data Engineering, 19(5), 631–645.

Strode, D. E., & Chard, S. M. (2014, December 1). A proposal for using design science in small-scale postgraduate research projects in information technology. https://doi.org/10.1109/tale.2014.7062633

Stuchi, J. A., Boccato, L., & Attux, R. (2020, January 1). Frequency learning for image classification. Cornell University. https://doi.org/10.48550/arxiv.2006.15476

Syed, Z., & Guttag, J. (2011). Unsupervised Similarity-Based Risk Stratification for Cardiovascular Events Using Long-Term Time series Data. Journal of Machine Learning Research, 12, 999–1024.

Symeonidis, G., Nerantzis, E., Kazakis, A., & Papakostas, G A. (2022, January 26). MLOps - Definitions, Tools and Challenges. https://doi.org/10.1109/ccwc54503.2022.9720902

Taherdoost, H. (2019, January 8). Importance of Technology Acceptance Assessment for Successful Implementation and Development of New Technologies. Global Journal of Engineering Sciences, 1(3). https://doi.org/10.33552/gjes.2019.01.000511

Tan, K., & Lim, B. P. (2018, January 1). The Artificial Intelligence renaissance: Deep Learning and the road to human-Level machine intelligence. Cambridge University Press, 7(1). https://doi.org/10.1017/atsip.2018.6

Teng, H., Chen, K., & Lu, S. (1990). Adaptive real-time Anomaly Detection using inductively generated sequential patterns. In Proceedings, 1990 IEEE computer society symposium on research in security and privacy (pp. 278–284). IEEE.

Tenbusch, S., & Arnold, D. (2018). WARP (Worldwide Analytics and Reporting Platform) stellt sich vor. Internal Source.

Theofilatos, A., Yannis, G., Kopelias, P., & Papadimitriou, F. (2016). Predicting Road Accidents: A Rare-events Modeling Approach. Transportation Research Procedia, 14, 3399-3405. https://doi.org/10.1016/j.trpro.2016.05.293

Tichomirov, D., Ferraris, A. & Lamprecht, A. (2023). The science of smooth driving: a taxonomy for anomaly detection in endurance run time series data, International Journal for Big Data Management, Vol. 3, No. 1, pp. 66-85. https://doi.org/10.1504/IJBDM.2023.133459

Tichomirov, D., Ferraris, A. & Lamprecht, A. (2024). Acceleration Events in Engine Development: Preparing an Anomaly Detection Approach for Vast Time Series Data, IEEE Transactions on Consumer Electronics (to be published)

Toonders, J. (2014). Data is the new oil of digital economy. Wired. https://www.wired.com/insights/2014/07/data-new-oil-digital-economy/ (Accessed 01 November 2022).

Venkatesh, V. & Davis, F. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. Management Science. 46. 186-204. 10.1287/mnsc.46.2.186.11926.

Vuppalapati, C., Ilapakurti, A., Chillara, K., Kedari, S., & Mamidi, V. (2020, December). Automating Tiny ML Intelligent Sensors DevOPS Using Microsoft Azure. In 2020 IEEE International Conference on Big Data (Big Data), pp. 2375-2384. IEEE.

Weizhong, Y., Brahmakshatriya, U., Xue, Y., Gilder, M., & Wise, B. (2013). p-PIC: Parallel power iteration clustering for Big Data. Journal of Parallel and Distributed Computing, 73(3), 352-359. https://doi.org/10.1016/j.jpdc.2012.06.009

Will, J., et al. (2021). Training Data Reduction for Performance Models of Data Analytics Jobs in the Cloud. https://doi.org/10.1109/bigdata52589.2021.9671742

X. Xu, H., Liu, L., & Li, M. (2018). A comparison of outlier detection techniques for high-dimensional data. International Journal of Computational Intelligence Systems, 11(1), 652-662. https://doi.org/10.2991/ijcis.11.1.50

Yeung, D.-Y., & Ding, Y. (2003). Host-based intrusion detection using dynamic and static behavioral models. Pattern Recognition, 36(1), 229-243. https://doi.org/10.1016/S0031-3203(02)00026-2

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., & Chawla, N. V. (2019, July 17). A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. Association for the Advancement of Artificial Intelligence, 33(01), 1409-1416. https://doi.org/10.1609/aaai.v33i01.33011409

Zhang, G., Luo, L., Zhang, L., & Liu, Z. (2023, January 18). Research Progress of Respiratory Disease and Idiopathic Pulmonary Fibrosis Based on Artificial Intelligence. Multidisciplinary Digital Publishing Institute, 13(3), 357-357. https://doi.org/10.3390/diagnostics13030357

Zhou, Y., Yu, Y., & Ding, B. (2020). Towards MLOps: A Case Study of ML Pipeline Platform. In 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 2020, pp. 494-500. https://doi.org/10.1109/ICAICE51518.2020.00102

Zhu, Y., & Zhang, K. (2021, November 1). The Application Status and Development Prospects of Artificial Intelligence and Industrial Robots in the Manufacturing Industry. IOP Publishing, 2066(1), 012067-012067. https://doi.org/10.1088/1742-6596/2066/1/012067

Zhu, Y., Ting, K. M., & Zhou, Z. H. (2018). Multi-label learning with emerging new labels. IEEE Transactions on Knowledge and Data Engineering, 30, 1901–1914.