Informatics

Der Umgang mit fehlenden Werten – Theorie vs. Praxis

Eine Fallstudie basierend auf Open-Source- und Industriedaten

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Cordula Eggerth
Matrikelnummer 00750881

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass. Dipl.-Ing. Dr.techn. Dietmar Winkler
Mitwirkung: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Stefan Biffl

Wien, 5. Oktober 2022

Cordula Eggerth	Dietmar Winkler

Informatics

Handling Data Completeness using Statistical Experiments: A Case Study on Open-Source and Industry Data

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Software Engineering & Internet Computing

by

Cordula Eggerth
Registration Number 00750881

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass. Dipl.-Ing. Dr.techn. Dietmar Winkler
Assistance: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Stefan Biffl

Vienna, 5th October, 2022

Cordula Eggerth Dietmar Winkler

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Cordula Eggerth

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 5. Oktober 2022

Cordula Eggerth

Kurzfassung

Der Grad der Datenvollständigkeit ist eine der Hauptdimensionen von Datenqualität. Wenn daher Datenqualität als Ganzes behandelt werden soll, kann die Performance in Form der gewählten Kennzahlen durch die Betrachtung verschiedener Arten die Datenqualität zu steigern und mit fehlenden Werten umzugehen im positiven Sinne beeinflusst werden. Da eine gewisse Datenqualität die Voraussetzung für die erfolgreiche Umsetzung von industriellen Anwendungen beispielsweise in CPPS-Engineering ist, können durch das Feststellen der vielversprechendsten Methode im Umgang mit fehlenden Werten wesentlich bessere Ergebnisse erzielt werden.

Demnach strebt diese Masterarbeit an, Einblicke sowohl von der theoretische Seite in Form von wissenschaftlichen Publikationen als auch von praktischer Seite, d.h. Personen, die im beruflichen Alltag mit unvollständigen Datensätzen konfrontiert sind, zu vergleichen. Das Ziel des Literature Review und der Onlineumfrage unter Praktikern war es, Informationen zu den jeweiligen Präferenzen im Umgang mit unvollständigen Daten zu sammeln.

Die darauffolgenden statistischen Experimente wurden im Rahmen von jeweils einer Open-Source-Fallstudie und einer Industriefallstudie, der Sensordaten eines Produktionsprozesses zugrunde liegen, durchgeführt. Im Zuge dessen wurden sechs Methoden zur Handhabung fehlender Werte in Kombination mit vier Anteilen fehlender Daten anhand von Ausführungszeit der Algorithmen, direkter Bewertung sowie indirekter Bewertung, letztere in Bezug auf RMSE und MAE, evaluiert. Hierbei zeigte sich, das jene Methoden mit der jeweils besten und schlechtesten Performance auch über die Fallstudien hinweg sehr ähnlich waren, aber wesentlich von der gewählten Kennzahl abhängen.

Abstract

Data completeness is one of the main data quality dimensions. Thus, to tackle data quality as a whole, considering different ways to enhance data quality in incomplete datasets can illustrate the performance impact on the chosen metrics. As sufficiently high data quality is a prerequisite for industrial applications for instance in CPPS engineering, determining the most promising method to handle missing values can essentially contribute to obtain better results.

Therefore, this thesis aimed to bring together insights from both the theoretical side of research publications and the missing handling methods suggested in comparison to the perspective of practitioners in jobs confronted with incomplete data. Both the literature review of research publications and the online survey among practitioners as to handling missing data were intended to obtain information on the respective preferences.

The subsequent statistical experiments were carried out in line with an open-source case study (in the natural resources sector) and an industry case study (using sensor data from a manufacturing process) to evaluate six missing value handling methods in combination with four different missing data rates in the underlying datasets according to the execution time of the imputation methods, direct evaluation as well as indirect evaluation, the latter with respect to RMSE and MAE. It turned out that the best and worst performing methods are similar across the case studies, but depend on the target performance metric.

Contents

CHAPTER 1

Introduction

Software applications in line with industrial production systems have increasingly entailed data collection, pre-processing, and analysis activities, notably in connection with Industry 4.0 and Cyber-Physical Production Systems (CPPS) [7]. The data stems from different sources such as manual measurement by humans or automated measurement by robots, machines, and sensors [7]. Data-driven activities are considered critical factors within decision-making, and can essentially determine the success or failure of the final outcome of a process in terms of quality, as decisions are based on the data in the CPPS [8] [7]. Missing values caused by communication network issues, sensor failures, and human errors, or outlier values for example due to erroneous measurement devices constitute a challenge that requires suitable measures to inspect and maintain data quality. Such issues can occur during or even after the data collection phase, as issues in processing, transferring, or aggregation activities might occur. In addition to this, “soft sensors“ like Fortuna et al. (2007) coined them, could be used to back up hardware sensor data for double-checks with regard to measurements, as well as for data simulation in cases where the measurements are not complete [25]. With regard to the latter, actual physical systems are depicted in analysis models, i.e. a synthetic emulation of the underlying system, so that simulation experiments can be done, and be later on projected back to the physical system [8]. According to Fortuna et al. (2007), soft sensors are embedded within the area of virtual instrumentation for industrial process control, and represent a means to “estimate system variables or product quality with the help of mathematical models“ [25].

Data quality turns out to be a decisive factor for the process outcome from several angles. Hence, the interest in and the necessity of analyzing data in a variety of sizes and shapes has been growing continuously over the last years [15] [14]. Generating insights in a data-oriented way is seen as key to add value today, which can manifest itself in accelerated decision-making, business advantages and unprecedented research insights [15] [85]. Although capturing data has gradually been facilitated and the means of capturing data have become more available, merely the large volume of data does not automatically lead to valuable data analysis results as such [15]. Indeed, inspecting the data and checking its quality at an early stage can save a significant amount of resources (e.g. time, money, and computing power), and helps prevent data analysis issues. Corrales et al. stated the general problem of "Garbage In - Garbage Out" that frequently occurs concerning data quality, which means that low quality input leads to low quality output that goes hand in hand with inefficient learning opportunities based on the data [27] [15]. To put it differently, the mere abundance does not ensure the quality of the data and subsequent analyses. Even given the availability of a large volume of data, it could be useless regarding its form or content for the purpose of the currently aspired data analysis.

In context with this, data quality should be seen in a way that the data needs to fulfil certain requirements and to fit the respective purpose of the later analysis phase(s) [15] [85]. Data quality cannot be defined in a straight forward way by one clear definition, but rather has a number of facets that need to be considered [45] [4]. Amongst others, data quality refers to "the capability of data to satisfy stated and implied needs when used under specified conditions“ (similar to what the related International Organization for Standardization (ISO) standard proposes), or in a nutshell, as the "fitness for use“ [45] [85]. This indicates that for being considered "fit“, or rather suitable, the data needs to meet certain characteristics, such as "performance efficiency, reliability, maintainability, or compatibility“ [7]. As already mentioned before, there is no single performance measure, which characterizes data quality. It depends on the purpose, the quality dimension, and the domain of application, which metric can best reflect the state of data quality [14] [4] [46]. Before actually stepping into the analysis phase, the suitability in terms of data quality should be questioned and inspected since poor data quality at the beginning of the analysis process might have adverse consequences later on [14] [46]. Low data quality could be the cause for unreliable outcomes. Quality issues can still make or break the success of the analysis process with regard to data in general [14] [46].

Several data quality dimensions have been so far identified by various researchers (without representing an exhaustive list) ranging from timeliness, completeness, and consistency to understandability, interpretability, and accessibility [45] [85] [46]. According to ISO/EC guidelines, the inherent (or "intrinsic“ as Juddoo et al. (2018) as well as Juneja and Das (2019) put it in [45] [46]) data quality characteristics refer to "the data itself, in particular to data domain values and possible restrictions“ and comprise the following (non-exhaustive list of) components [8]:[1]

- *Accuracy* (i.e. how close the value under concern comes to the actual real value of a feature)
- *Completeness* (i.e. how many and which values are available, i.e. not null, for a specific feature, or overall in the dataset)

[1] Note: Source refers to the content of all bullet points.

- *Credibility* (i.e. how authentic and realistic the values appear to be)
- *Currentness* (i.e. how up to date the values are with regard to recently measured values)

Moreover, the system-dependent (or “contextual“ as Juddoo et al. (2018) as well as Juneja and Das (2019) put it in [45] [46]) data quality characteristics do not refer to the data itself, but rather to the system around them, which includes also external factors such as devices or sensors, and infrastructure, and comprise the following [7]:[2]

- *Availability* (i.e. how easily the data values are available for authorized human and/or machine users)
- *Portability* (i.e. how easily the data values can be moved to another system or process while not losing their information content)
- *Recoverability* (i.e. how easily the data values can be regained once they have been deleted or lost)

According to the ISO/EC guidelines, some of the data quality characteristics even refer to both inherent and system-dependent characteristics, namely accessibility, confidentiality, performance efficiency, compliance, traceability, precision, and understandability [7]. Juneja and Das (2019) as well as Firmani et al. (2016) further mentioned format and semantics conformity of data, redundancy, and uniqueness as data quality dimensions [46] [24]. The data quality characteristics also have a substantial influence on practice settings, as over 70% of data activities carried out in data warehousing and data science are ascribed to checking, improving, and analyzing data quality [4]. The way how each one of these quality dimensions is tackled in the data preparation process has a considerable impact on the results that can be obtained. Cichy and Rass (2019) pointed out in their literature review on the use of data quality dimension in data quality management frameworks that the dimension of “completeness“ is one of the most widely used dimension [14]. At this point “completeness“ refers to whether the “depth, breadth, and scope“ of the data is appropriate, and how missing or “null“ values in the underlying data are tackled [14].

Within the data analysis process, the underlying dataset is first examined with regard to data quality dimensions. Imperfect data collection situation in practice due to network, sensor, database availability issues, faulty entries, or format problems frequently leads to some attribute values being missing in the dataset [55]. From this arises the question whether to delete the observations affected by attributes with missing values or to tackle the incompleteness of the dataset. According to Lin and Tsai (2019), if less then 10% to 15% (i.e. a rather minor amount) of data of an attribute is missing, dropping the affected

[2] Note: Source refers to the content of all bullet points.

observations does not have a fiercely detrimental impact on the subsequent data analysis [55]. In case the share of missing data is larger than that, it is advisable to carefully evaluate strategies of dealing with the missing data, and what impact these strategies might have on the outcome [55]. Additionally, it has to be mentioned that even in settings with a very small percentage of missing values, the missing values might be exactly those observations that would have a large impact on the data analysis results [55]. When comparing these strategies, it turns out that in practice missing value imputation using either statistical or machine learning based methods is chosen in most cases, with the use of the latter ones having gained pace over the last decade [55]. So far missing value imputation methods have been most widespread with regard to the medical and image classification domain [55]. Therefore, there is a need to explore the value added of using missing value imputation methods, identified via the use of several performance metrics, in line with industrial data in the pre-processing phase in a systematic way in order to improve the subsequent data analysis results.

In view of the above, this thesis project will focus on analyzing as well as enhancing the overall data quality, and notably the completeness dimension of data from the manufacturing domain. In industrial applications, in line with CPPS engineering, high quality of the input is a crucial precondition for achieving high quality of the output data [7] [37]. Within the data quality dimensions, data incompleteness due to sensor errors, environmental conditions, machine failure, human perception errors and the like can diminish input data quality and lead to missing data points [7] [4] [37]. No guarantee can be given that data within CPPS processes will be complete at any time with certainty. Complete data would be the ideal case. However, in reality the sources of perturbance to completeness are manifold. Therefore, this thesis aims to firstly investigate what dimensions data has, with a focus on the dimension of data completeness and how it is embedded in the overall notion of data quality.

Incomplete data can in turn hamper the performance of machine learning algorithms in subsequent workflows of smart manufacturing. White et al. conducted a study on imputation of missing data entries based on manufacturing data in 2018, in which they pointed out that over 70% of examined manufacturers who participated in a state survey used some imputation method for at least one input variable in the manufacturing process [90]. Based on an analysis of manufacturing companies from the United States comprising data from around the year 2002 to the 2010s, as much as 20% to 40% of variable entries were reported missing, and had to be imputed (in a manufacturing setting) to make the data complete, according to White et al. [90]. The main imputation method used by the manufacturers surveyed was mean imputation, which is one of the simplest and most basic imputation methods available [90]. For these reasons, this project aims to explore and evaluate methods for making incomplete data complete and to handle missing data entries with regard to direct comparison with a datasets that were made pseudo-incomplete by deleting values, as well as by indirect measurement by the means of subsequently applied supervised machine learning techniques and their respective influence on performance, to gain insights into the experience with regard to data completeness from practitioners, and to develop a “virtual sensor“ (cf. “soft sensor“) with the help of statistical experiments, which evaluates a series of data imputation techniques, both statistical and machine learning based, based on specific datasets to make incomplete data complete again in order to examine whether the performance of subsequent machine learning methods, more precisely regression methods, can be influenced and improved in this way. The statistical experiments are carried out in line with an open-source as well as an industry case study. The so-called virtual sensor tries to create well-fitting replacements for the missing data entries using a particular imputation technique to avoid losing valuable data samples as a whole, and to improve input data quality for subsequent data analysis steps in the course of manufacturing workflows and processes. **Therefore, the research goal of this thesis is to get insights into how practitioners handle data completeness, and to apply as well as evaluate missing value imputation methods in line with the open-source use case, and the industry use case.**

Figure 1.1 below depicts the flow diagram of the thesis phases. The thesis starts with the chapter 2 **Related Work**, in which previous related research is summarized, and the research gap is described. This chapter is intended to explain why the thesis is relevant, and how it can fill the research gap. In line with the **Research Questions & Methods** phase in chapter 3, an overview on the research approach is given, and it is stated how the methods are embedded according to design science. The Research Questions (RQ) are stated, and the respective solution approach is described in greater detail. The third phase is composed of three sub-phases, which comprise the chapter 4 **Literature Review** on techniques to handle data completeness and how they can be compared, 5 **Survey** on data completeness which is conducted among practitioners in industry and academia, and 6 **Statistical Experiments**, which comprise an open-source case study and an industry case study where data imputation techniques are applied, compared, and finally evaluated. The three sub-phases planned to be sequential, as they deliver insights for the respective next sub-phase, but somewhat overlap. Then the results and insights from the three sub-phases are considered with reference to the research questions raised previously and the related work in the chapter 7 **Discussion & Limitations**. Furthermore, the limitations of the research undertaking to be borne in mind are stated. In the end, the chapter 8 **Conclusion & Future Work** present the final thoughts, and what research extensions on this topic should be pursued in the future in this area.

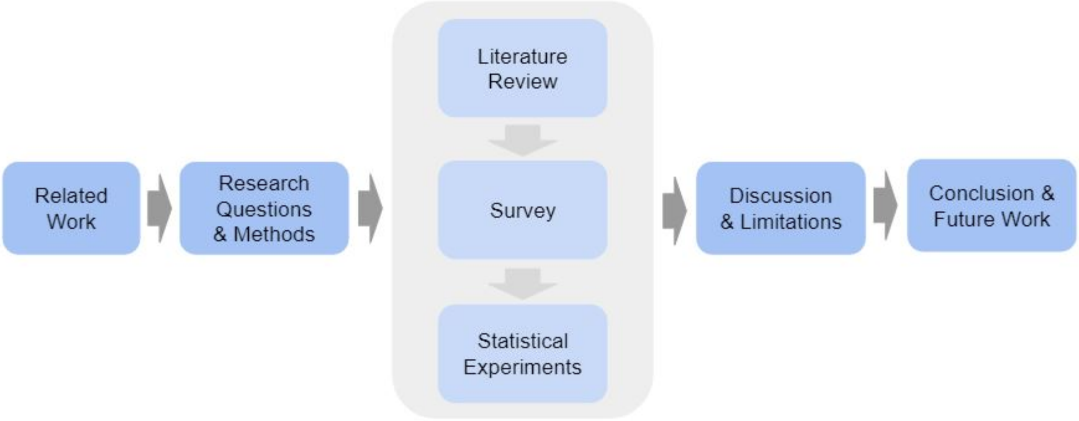


Figure 1.1: Sequence diagram of the thesis

The chapter **Related Work** covers literature and research projects that have previously been done in the field of data quality with a focus on completeness. General publications as well as specifically papers covering the industrial domain are screened, and analyzed. At this step, the objective is to give an overview of what kind of research has been carried out so far in the area under concern. Based on these insights, the research gap emerges, and is described. Within this chapter, an explanation is given why the thesis is relevant to further evolve the field, and how it can contribute to filling the identified research gap.

The chapter **Research Questions & Methods**, comprises both a research overview as well as a section on research questions and the respective methods. The research overview firstly introduces the research approach chosen for this thesis. It details and visualizes the flow of the research project's components, and how they are connected. It indicates how the selected methods fit into the design science framework to proceed from the research problem to the solution. The section on research questions describes which topics the research questions cover, and what their motivation with respect to the overall research goal is. The first research question (RQ1) deals with the dimensions of data quality, and explores notably data completeness among the dimensions. The second research question (RQ2) determines the methods and models used in research and in the industry to inspect, evaluate, and improve the handling of missing values within the data science lifecycle. Finally, the third research question (RQ3) focuses on the handling of incomplete data to make them complete again, and to find out how the method of making them complete impacts the data quality in terms of subsequent application of supervised machine learning techniques based on selected datasets. The RQ3 consists of two sub-questions, namely RQ3a and RQ3b, out of which the first one covers the open-source data case study, and the second one comprises the industry data case study. In the end, the results from both sub-questions are brought together by comparing the performance outcomes, and metrics measured during the case studies.

The **Literature Review** is intended to elaborate on RQ1 in order to shed light on the dimensions of data quality, notably on data completeness. Moreover, it covers the techniques available to impute missing data values. For this reason, a keyword definition and search strings are prepared, which are subsequently executed in the selected search engines and platforms. The sources deemed relevant are extracted, and analyzed to gain an overview in relation to the topic, which can later on serve as an input in line with the creation of survey questions to answer the second research question.

The **Survey** comprises the development of a questionnaire on the handling of data completeness and the use of imputation methods regarding missing values in datasets among practitioners and academics that, and aims to answer RQ2 essentially. The survey is drafted based on the results of the literature review, and the related work. It is carried out in the form of an online survey with a planned duration of several minutes. At this step, the goal is to find out how data completeness is perceived by practitioners, and how they handle it in their work environment. The survey results are collected and analyzed systematically.

The following chapter **Statistical Experiments** uses the input from the related literature, the literature review, and the online survey, to define the experimentation framework for the open-source and the industry case studies. The framework definition comprises the selection of data imputation techniques, of the seed values, the number of experiment runs, subsequent supervised machine learning methods, and the choice of performance metrics, baseline as well as statistical tests to evaluate the results. In addition to these characteristics being specified, the dataset for the open-source case study and the industry case study are presented and illustrated in a descriptive way. Then the experiments are carried out using both datasets and the performance metrics are calculated. The experiment results are finally subject to statistical tests to compare the strategies against the baseline. The results are depicted both numerically as well as visually, and are compared across both case studies.

The chapter **Discussion & Limitations** aims to answer the research questions stated above while considering the related work and the state of the art. It presents the results of the literature review, the online survey, and the statistical experiments comprising the two case studies. On top of this, the discussion also points out the limitations that have to be considered in connection with the answers to the corresponding research questions that might have occurred due to assumptions, limited research consideration, survey scope, or the selected methods in the experiments phase.

The chapter **Conclusion & Future Work** finally closes the thesis by bringing together the answers to the research questions, and the results from the discussion and limitations. It illustrates what can be concluded from the research carried out in this thesis, and what research still needs to be done in the future to further advance the research area.

CHAPTER 2

Related Work

This chapter covers in the corresponding sub-sections related previous research that constitutes the foundation for this thesis. The part 2.1 gives an overview on the challenges that occur with regard to data quality, notably in a CPPS context. The sub-section 2.2 explores the notion of data quality, and presents its components, the data quality dimensions, as well as the different groupings of dimensions. The sub-section 2.3 focuses on the data completeness dimension, and elaborates on how missing values have been treated in academic literature, according to previous research.

2.1 Data Quality Challenges in CPPS

In line with Industry 4.0, cyber-physical production systems, which make use of artificial intelligence (Artificial Intelligence (AI)), data-driven activities, and automated machines as well as processes, are increasingly deployed [8] [92]. Cyber-physical systems, to which also CPPS belong, usually comprise a combination of both physical elements, and a "cyber twin" that emulates the physical part closely [5]. Data is the connecting link that brings together the physical as well as the digital world at this point, as the digital part represents the physical one and provides opportunities for data analysis [72]. When it comes to Internet of Things, a number of cyber-physical systems are connected in terms of communication to transfer, process, and exchange data [5]. In Industry 4.0, the devices and machines are enabled to carry out tasks without any intervention by human workers. CPPS combine AI and IT with physical elements to become "smart devices" , which then interact in the production process and beyond that with signals from external parties [69].

CPPS bring together advances from computer science, machine learning, IT, networking, and manufacturing technology, as they connect the physical world (e.g. devices, machines) with the digital world, i.e. the "cyber" part, while being continually monitored in terms of data collection and analysis [62] [8]. The cyber part, i.e. IT-enabled processes and applications, is connected with the physical layer via an "interaction interface" that exchanges information and data between both of them, and makes use of communication networks such as the internet [69]. According to Plakhotnikov (2021), cyber-physical systems are "systems that integrate equipment, sensors, actuators, computing resources, and information systems throughout the entire value chain, usually extending beyond a single enterprise or business" [69]. In doing so, machines, parts, data, and process steps are uniquely identified, and enable higher automation in communication and collaboration within manufacturing, which finally aims to seize demands from the clients faster to customize the products [79] [8], while increasing effectiveness, flexibility, and quality of the whole manufacturing process [62] [92].

Among the main features that characterize CPPS is firstly *intelligence*, i.e. being "smart", in a way that the participating devices and machines are capable of exchanging data autonomously with other devices and machines and their external environment, as to Monostori (2016) [62]. Secondly, *connectedness* characterizes CPPS, which means that devices and machines can collaborate with humans and other devices or machines via communication networks (e.g. the internet or virtual networks) [62]. The third main feature is the *responsiveness* of CPPS to signals from devices, humans, and the surroundings [62]. Even though new opportunities (such as real-time tracking, data collection and automated machines) arise in such an environment, it also entails downsides such as problems with data quality that is a crucial element for success, but might hamper the progress [79] [92]. In the ideal case, one would expect robust data-driven processes, real-time monitoring, analytics, transparent data collection, and correct modelling or self-aware flagging for maintenance from CPPS [62] [3]. As CPPS applications rely on data-driven processes, which in turn require (i.e. obtain and create) large quantities of data, this can also constitute a weak point in the production chain as the ideal case is usually not perfectly reflected in reality. For this reason, Williams (2020) suggests that in this context one should be vigilant to detect possible data quality issues that could lead to bias, noise, and decreased accuracy with regard to the analysis outcomes [92].

Therefore, manufacturing companies are well advised to integrate data quality assurance processes into their CPPS, and to determine the data quality dimensions relevant to them to capture data quality from various angles [92]. Plakhotnikov (2021) also stated that data quality represents one of the main challenges in Industry 4.0 and CPPS [69]. Atat (2018) proclaimed that after the data collection phase in CPPS, that the data should undergo a data cleaning before the data flows into the analysis phase in order to spot quality issues as early as possible [5]. Sensors and further devices can generate and receive large amounts of data, but the data itself might be corrupted due erroneous signals, non-standardized data which leads to inconsistent formats, faulty sensor parts, network errors or outages, lacking quality control relating to open data, and wrong calibration [69].

This can lead to outliers (i.e. values that appear very unlikely given the usual range of values being registered by the devices), noise, or missing values. Lee (2017) stated that data in industrial settings appears to be usually "more structured, more correlated, and more orderly in time, and more ready for analytics“ as such because such data is captured by tailored devices that are suitable for the context due to more controlled processes, and reduced human intervention [53]. Still, the connectedness feature of CPPS that was mentioned above, introduces risks to data quality, as the manufacturing company might not exert control over all devices, sensors, and humans involved along the production chain, and supporting processes [62] [53].

In the manufacturing domain, the “3 B's“ appear to be particularly characteristic of the data collected and generated, which comprises “Below-surface, Broken, and Bad quality“ [53]. “Below-Surface“ means that industrial data is not just the data, but involves also the physical elements behind the data, which requires domain knowledge for the interpretation and processing of data [53]. In industrial data analytics, the data quality dimension of data completeness tends to be particularly crucial because just large volume of data or tailored analytics methods do not lead to the desired outcomes in case too much data entries are missing for whatever reason, be it network issues, incompatible data formats, sensor or human errors [53] [72]. This is considered as the second “B“, which means “broken“, and refers to the fact that incomplete data might hamper the successful operations of industrial processes and data analytics [53].

For this reason, tailored pre-processing is recommended. Thirdly, “Bad quality“ points out that the volume of data is not sufficient to obtain precise analysis results if the quality of data is very low [53] [54] [72]. Just like data completeness, Lee (2017), Nguyen (2022), and You (2021) highlight the data quality dimension data accuracy as another critical dimension for industrial data analytics, as in industrial processes data usually “possess clear physical meanings“ [97], so that inaccurate data can substantially impair the production results, or even lead to dangerous decisions such as safety and security issues [53] [97] [63]. Such data quality problems can be due to external as well as internal causes. Among the external causes are for instance synchronization problems of clocks of the participating devices, which could lead to deviations and therefore non-available values at the time of measurement, furthermore disturbance or even attacks to the time signals [97] [54]. Among the internal causes, the number of minimum necessary transmitted values per time unit might not be reached, or aging devices could miss out on values due to weather conditions, overloading of the device, or chemical processes [97] [63]. Moreover, data can also get lost during transfer due to replay attacks[1] or man-in-the-middle attacks[2] to name just a few, that lead to delay or corrupted data [97] [8]. In the processing phase, attackers could gain access from a remote location, and command-control the system in a way that it is overloaded or otherwise forced to drop data. All those issues can lead to valuable data entries being lost, and make the data incomplete.

In conformance with the research projects mentioned above, and notably Lee's (2017)

[1] In a *replay attack*, the attacker steals the authentication information of the system through eavesdropping on the network, and then illegally “repeats valid data transfers“. [97]

[2] In a *man-in-the-middle-attack*, the attacker obtains access to the communication channel, then opens a new clandestine communication channel, and starts communicating with the system. [97]

and Atat’s (2018) previous work [53] [5], this thesis also acknowledges the influence of the data completeness dimension on the overall data quality in ‚CPPS and Industry 4.0. Therefore, this thesis takes up the topic of handling the data completeness dimension from various angles, and continues the research with regard to enhancing data quality from the data completeness viewpoint.

2.2 Data Quality and its Dimensions

On the one hand, this thesis draws upon previous works concerning data quality dimensions, with a focus on data completeness dimension. On the other hand, previous research on methods to deal with incomplete data, or missing values in the context of data quality and notably in the industrial or manufacturing sector, form the basis for this thesis. The pieces of research cited in this part are available via and were retrieved from Scopus.

Workflows involving data processing and analysis have become an integral part of most research and company projects. When it comes to data quality in such settings, Heinrich et al. (2018) stated that it cannot be evaluated based just on one metric, but it is composed from several dimensions, which can vary in their importance from case to case [38]. The notion “dimension“ for the aspects of data quality was chosen in order to convey the meaning of making data quality evaluable or measurable in a quantitative way [43] [18]. According to Heinrich et al. (2018), more than 80% of the participants to a survey said that “poor data quality“ led to poor business results at their company [38]. More than 65% of the participants in the survey claimed that they had witnessed cases in which poor data quality negatively influenced their business within the last year, and in excess of 80% of chief executives consider data quality very crucial for their company’s progress [38]. Furthermore, another survey showed that around 60% of the participants were not confident in the data quality and Data Quality (DQ) management approach at their company [40].

So far, there is no official unified definition of “data quality“, but many attempts have been made to elaborate on the characteristics and dimensions that data quality should possess [59] [10]. Data quality describes the degree how well the data sources available to the person or organization analyzing it matches with the real actual data [38]. By comprising multiple dimensions, it can consider quality from various angles, which has been systematically explored from the 1950s onwards mostly related to general “quality issues“ in a production context, but subject to substantially wider and fast growing research interest since the 1990s, and led amongst others to the formulation of ISO standards related to data quality in 2011 and 2014, which referred to data quality as the “degree how much data characteristics correspond to what was needed“ [42], meaning that the data conforms to the stipulated requirements [59] [56] [48] [81] [10] [40]. The ISO proposed a data quality model that comprises two groups of data quality dimensions, namely the *inherent*, and the *system-dependent* dimensions. The inherent dimensions, along which one can evaluate data quality, include [42][3]:

[3] Note: Source refers to the content of all bullet points.

- *Accuracy*: how close the value under concern comes to the actual real value of a feature
- *Completeness*: how many and which values are available, i.e. not null, for a specific feature, or overall in the dataset
- *Consistency*: "the degree to which there are no conflicting values in the data and its features" [42])
- *Credibility*: how authentic and realistic the values appear to be
- *Currentness*: how up to date the values are with regard to recently measured values; also referred to as *timeliness* or *currency*
- *Accessibility*: how easily a person or organization that intends to use the data can actually access it [10]
- *Compliance*: the degree of agreement with regulatory and / or topic-related voluntary standards (e.g. regarding format, content)
- *Efficiency*: how efficiently the data captures the underlying information [10]
- *Confidentiality*: how the data respects confidentiality of persons and organizations, and does not leak information unnecessarily

Among the system-dependent data quality dimensions, which consider the hardware and / or software system where the data is stored and managed, are the characteristics of *availability*, *portability*, *recoverability*, *understandability*, *precision* as well as *traceability*, according to the ISO [42] [48]. Jayawardene (2013) also called the inherent dimensions *declarative* as they focus on the data itself and further describe its characteristics irrespective of any consumers of the data [43]. Based on this, inherent characteristics can be tackled in the system itself without necessarily involving the user [43]. As for the system-dependent dimensions, they depend on the interaction with the users and his / her judgement of this usage experience, according to Jayawardene (2013), and occur when the user performs some data operations [43].

In more recent times, the dimension of *trust* (in the data and / or its source(s)) was added to set of existing data quality dimensions, notably in the context of increasingly used open data which is freely available, sensor measurements, big data, and social networks data [48] [18]. Overall, the data quality dimensions can refer to structured, semi-structured as well as unstructured data, as quality matters for any kind of data, no matter what it shape is [38] [48] [68]. Another approach, proposed by the Data Management Association Work Group, is total data quality management, which considers data quality to be composed of the dimensions *accuracy*, *completeness*, *timeliness* (also referred to as *currency*), *consistency*, *validity*, and *uniqueness*, which slightly deviates from the ISO dimensions [64]. Further dimensions mentioned in the literature in connection with data quality in general are *authority*, *objectivity*, *coverage of audience*, *popularity*, *information-to-noise-ratio*, *auditability*, *adequacy*, and *cohesiveness* [10] [81] [29]. Ge (2019) presented the dimensions *consistency* (considering duplicate data as the main issue), *accuracy* (considering data outliers), *completeness* (considering missing values), *credibility* (considering low trust values and sources), *accessibility* (considering semantic and measurement issues), *timeliness* (considering delayed or wrongly timed values), and *interpretability* (considering anonymized data values) as most influential with a focus on the industrial domain [29].

Günther et al. (2019) see the DQ dimensions from several perspectives, namely firstly the *user perspective*, which refers to the "expectations and intended use" from the user's viewpoint [34]. Secondly, Günther et al. (2019) mention the *data perspective* that involves the choice of DQ dimensions in conformance with the objectives of the underlying project, and later on the evaluation of the DQ dimensions [34]. The third perspective is the *real-world perspective*, which embeds the project into a specific domain, and then defines the DQ dimensions relevant for it [34]. In line with their literature survey, Günther et al. (2019) also elaborated on the main groups of DQ dimensions, which included *accuracy*, *timeliness*, *completeness*, *relevance*, and *consistency* [34].But there is still potential for research on measuring the respective DQ dimensions, as research was in many aspects too generic and not taking individual project features into account in terms of weighing the relevant DQ dimensions for a specific project, Günther et al. (2019) claimed [34]. Furthermore, the interdependencies between the DQ dimensions should be examined more thoroughly in research [34]. Somewhat similar to this distinction, Zhang (2021) divided the DQ dimensions into the "practical perspective" on the one hand, which details the user's view on evaluating the respective data characteristics, and the "system perspective" that refers rather to the infrastructure, corporate or software system around the data on the other hand [99].

In the context of data quality evaluation and management used for Internet of Things (IoT) and industrial applications, Liu et al. (2020) discerned that this is an application area which notably "relies heavily on the quality of the data collected by devices", and provided a first systematic literature review on the topic, based on around 45 research studies from 1999 to 2018 to explore how DQ dimensions, issues, and metrics are related [56]. According to previous research, sensor readings in IoT and industrial applications are not fully reliable in practice which leads to common data quality problems such as inconsistent data, diverging formats, missing values, duplicates, or erroneous measurements [99]. From Liu et al.'s (2020) systematic literature review, it emerged that the most frequently mentioned data quality dimensions with respect to the field of IoT and industrial applications are accuracy, completeness, and timeliness, in this specified order [56]. According to Liu et al. (2020), the by far most frequently pointed out issue in data preparation and analysis was "missing data", followed by "measurement errors", "outliers", and "noise" [56]. Zhang (2021) named missing data as the second most widespread data issues in the industrial domain, just after outliers [99]. As causes of missing data, Zhang (2021) mentioned network issues, instable internet connection, power problems at the device, and environmental factors [99]. Data is considered “accurate“ once the data available to the analyst is corresponding to the true value in reality [56]. The accuracy could be hampered by “measurement errors“, which involves for instance faulty sensor choice or placing, thus leading to bad data quality (e.g. outliers, null values, noise) [56]. As for timeliness, IoT and industrial applications require that the data is updated at the “time of observation for an object“, and not after a latency period, which means that the data should be “fresh“, as Liu et al. put it [56]. If timeliness is not respected, missing values as well as out of date values due to delay periods could be introduced to the dataset [56] [81]. Liu et al. (2020) and Song (2020) stated that the DQ dimension of completeness was mentioned under the synonymous notions of “availability“ or “missing data“ by other researchers [56] [81]. As causes of incomplete datasets for example sensor failures or network communication problems were mentioned [81] [56] [9].

Jayawardene (2013) conducted a study among industry professionals and academics to gain insights on their view of various common data quality dimensions based on around 15 publications from an academic as well as an industry background, which comprises more than 120 dimensions [43]. The goal of Jayawardene (2013) was to find out how the participants perceived DQ dimensions, i.e. whether they perceived each one rather as inherent, system-dependent, both of them, or even none of them, and to cluster the main dimensions [43]. The main clusters determined by the participants finally comprised the dimensions of *completeness*, *accuracy*, *validity*, *usability*, *reliability*, *credibility*, *timeliness*, *consistency*, and *accessibility* [43]. For instance the *completeness* dimension was perceived as part of the inherent or declarative characteristics group by the majority of the study participants, as the null values or missing values can already be determined in the data itself, and are not connected to user interactions [43]. Batini (2009) denoted *accuracy*, *timeliness*, *completeness*, and *consistency* as the “basic set of data quality dimensions“, which have prevailed in the literature [6]. Although the basic dimensions appear to be similar, their meaning and scope differ substantially across research and practical projects [6]. Yang (2017) described the problem of missingness in line with data integration and collection from two sides, namely the “direct incompleteness“ meaning that a specific value is not available, and the “indirect incompleteness“, which means that a value is null during loading, updating, or merging operations [94]. Missing entries were cited among the most frequently occurring data quality issues alongside “format incompatibility, conflicting entries, multi-resource integration, multi-table files, and multi-meaning attributes“ [94]. In their research, Zhang (2021) compared DQ dimension models based on 21 papers, and five ISO standards in line with a literature survey, based on which they concluded that the accuracy, completeness, and timeliness were they most widely used dimensions for assessing data quality [99]. Cai (2015), Heinrich (2018), and Jayawardene (2013) mentioned that after selecting the relevant data quality dimensions for the project under concern, the dimensions should be evaluated according to the selected metrics to assess the data quality both in a qualitative and quantitative way [10] [38] [43]. Depending on the outcome of the assessment, the measures might be necessary to improve the data quality and to re-evaluate according to the dimensions, which is recommended in the preparation phase rather than in the analysis phase itself [10] [38] [43].

2.3 Handling the Data Completeness Dimension by Treating Missing Values

The most time-intensive phase in the data science process is the preparation and preprocessing phase, in which the data undergoes cleaning steps, and amongst others the treatment of missing values [13] [73]. Data completeness can be improved by using methods to deal with missing values in the underlying data. Lee (2020) and Chuo (2022) concluded from their studies that data incompleteness in one or more rows (i.e. observations) or columns (i.e. features) has remained among the main challenges for industrial data analytics next to imprecise, biased and insufficient data so far, which requires further practical as well as academic attention in terms of methods to deal with the challenge [52] [13]. In case more than 50% of the values of a feature are missing, whole columns tend to be dropped, as completing them with whatever technique is labelled as "unreliable" [52]. According to a systematic study by the U.S. Census of Manufacturers from 2002 to 2007 across more than 200000 production plants, over 70% of the participating organizations indicated that they had used imputation methods for one or more variables in their datasets [90]. Relating to variables containing incomplete data, the share of missing values ranged from around 20% to 40% [90]. Given the share of missing values, dropping incomplete observations as a whole or dropping incomplete features would considerably shrink the datasets, and therefore impair statistical operations carried out later on with the data [90]. Mean imputation was the technique that was most frequently chosen, based on the assumption that it would decrease the variance of variables where the completion took place [90]. Although the mean imputation could already improve the situation over the incomplete state, further statistical and machine learning based techniques were recommended to be applied in order to aim for even higher performance advances [90] [52] [73].

Considering performance of techniques to make data complete, there are two ways how it can be measured, firstly the direct evaluation, and secondly the indirect evaluation (see e.g. Nugroho (2019) [65]). The *direct evaluation* means that the analysis is started from a complete dataset, and is artificially made incomplete, i.e. pseudo-incomplete. Then the pseudo-incomplete dataset is made complete again using imputation methods, and the share of correctly imputed values is compared with the corresponding values from the complete dataset for each imputation method. The method with the highest percentage of correct imputations is the one with the best performance. The *indirect evaluation* means that the analysis is started based on either a complete or an incomplete dataset. Subsequently, imputation methods are used to make the dataset complete again. Afterwards, classification (in case the target variable is a categorical variable) or regression methods (in case the target variable is a numeric (continuous) variable) are applied to the dataset, and their respective performance is measured. The performance metrics of the regression or classification methods are compared as metrics to determine which imputation method finally performed best. Performance metrics for classification cases are for example accuracy, precision, recall, F1 score, specificity, receiver operating characteristic (ROC)[4], or Area under the ROC (Receiver Operating Characteristic) curve (AUC) [52] [31]. Performance metrics for regression cases are for instance R^2 or R-squared[5], Mean Squared Error (MSE)[6], Root Mean Square Error (RMSE)[7], the Mean Absolute Error (MAE)[8], the Root Mean Absolute Error (RMAE)[9], similarity measures, correlation measures, or a modified version thereof [52] [31] [36].

Lee (2020) investigated whether "higher prediction accuracy supports better decision-making" [52]. In the case of regression, Lee's (2020) research suggests that higher accuracy is beneficial to the quality of decision-making [52]. For classification cases, the misclassification errors of the confusion matrix have to be taken into account, namely type I error, i.e. false positive, and type II error, i.e. false negative, that can have an impact on the "decision risk" [52]. Statistical and machine learning methods should ideally keep both errors low, but there is a trade-off between them. So, regarding decision risk it has to be considered which one to prioritize. For example in a manufacturing process, type I error means that a produced item is marked as damaged even though it is not damaged, thus a false alarm, while type II error means that a produced item is marked as not damaged even though it is in reality damaged, thus a missed damaged product which could be dangerous. The former situation would be bad because time is lost because the product subject to the false alarm needs to be inspected again, but the latter situation is at least as bad, if not much worse, as the damaged part is not detected, and is built into a final product that is shipped to the customer, and could cause harm there [52]. Thus, the choice of performance metric finally depends essentially on what the manufacturer would like to achieve. Accuracy takes into account the overall correctly classified parts, precision handles type I error, while recall handles type II error, and F1 score takes both precision and recall into account [75].

The code execution to execution time (or "total computation time") throughout the missing value replacement process also represents a viable performance metric to compare missing value imputation methods, which is influenced by the scope of the dataset and the share of missing entries [36]. In a limited share of publications, k-fold cross-validation (CV) was used in order to make the performance metrics more robust across several runs [36] [52].

Nugroho (2019) confirmed based literature study that missing data can considerably influence the prediction results, and that the method of imputing missing data can have an impact on the performance too [65]. This was explored by following a data science process model that started with the raw data collection, preparation and formatting, then

[4] ROC: Curve where the "true positive rate is on the y-axis and false positive rate on the x-axis, by varying thresholds, which describes the trade-off between hit rate and false alarm rate, i.e. the relationship between sensitivity and specificity"[36].

[5] R^2: "proportion of variance between estimated values and actual values, calculated using the sum of squared error" [31].

[6] MSE: "mean of the squared difference between actual values and the predicted value" [31].

[7] RMSE: root of MSE.

[8] MAE: average of the "absolute difference between the actual values and the predicted values" [31].

[9] RMAE: root of MAE.

training a model, and finally predicting using the model, and evaluating the performance [65]. Depending on the situation, the missing data mechanism might be one of the following mechanisms, on which researchers uniformly agree [65] [93] [41][10]:

- *Missing Completely at Random (MCAR):* Data that is missing completely at random does not exhibit a relations or dependencies between the missing data entries. Thus, no missing value depends on other values in the dataset or from the further missing data:

  $$\mathbb{P}(\text{missing value} \mid \text{full dataset}) = \mathbb{P}(\text{missing value})$$

- *Missing at Random (MAR):* Data that is missing at random depends on some of the values from the underlying dataset:

  $$\mathbb{P}(\text{missing value} \mid \text{full dataset}) = \mathbb{P}(\text{missing value} \mid \text{observed dataset})$$

- *Missing Not at Random (MNAR):* Data that is missing not at random means that the "probability of a missing value is directly dependent on the missing value itself", thus missing and observed data are not correlated [65]:

  $$\mathbb{P}(\text{missing value} \mid \text{full dataset}) \neq \mathbb{P}(\text{missing value} \mid \text{observed dataset})$$

Nugroho (2019) presented in the study a non-exhaustive summary of methods to deal with missing values, as illustrated in Figure 2.1, which partitions the methods into "statistical" and "machine learning based" ones, with single imputation comprising for example regression, single hot deck, single cold deck, unconditional mean, and stochastic imputation. [65]. The methods differ in computing power necessary, so for example case deletion, maximum likelihood, mean, and regression imputation methods require low computing resources, while multiple imputation and machine learning based methods usually require more resources [65]. As for the imputation methods, research has dedicated to developing new methods and/or new combinations of methods, followed by application or comparison studies based on existing imputation methods, according to Nugroho's (2019) literature review [65]. The most rare case was a combination of the two types of research [65].

A slightly different categorization of missing value treatment methods was suggested by Xu (2015) (see Figure 2.2), who also pointed out both row-wise and column-wise case deletion methods as a basic form of dealing with missing values [93]. However, case deletion face the downside that a large amount of data is dropped, and might entail high bias in case of MNAR data, which might altogether sacrifice valuable information [93] [41]. In mean imputation, the mean across all observations or across some condition is imputed

[10] Note: Source refers to the content of all bullet points.

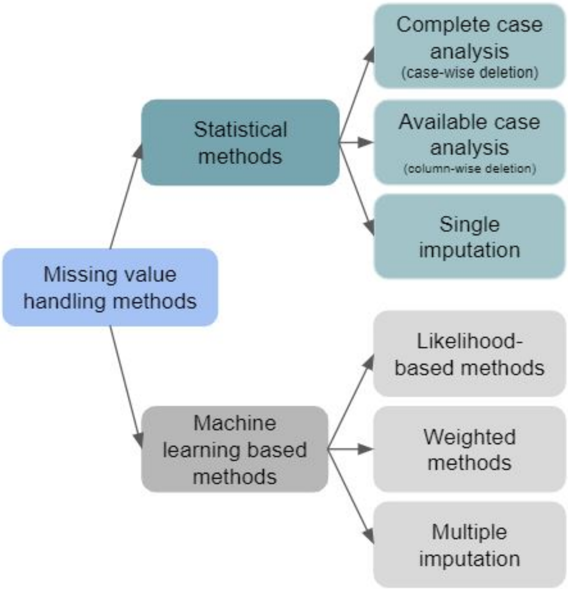

Missing value handling methods
Statistical methods
Complete case analysis (case-wise deletion)
Available case analysis (column-wise deletion)
Single imputation
Machine learning based methods
Likelihood-based methods
Weighted methods
Multiple imputation

Figure 2.1: Methods for missing value treatment by Nugroho (2019) [65]

instead of all missing values [93]. In hot-deck imputation, the missing value is replaced by a value from a complete case that is similar to the one with the missing value [93]. The regression replacement is done by considering the variable for which a value is missing as a regression target variable, and then performing a regression using the remaining variables in the data to predict it [93]. The interpolation methods require the dataset to be MAR, and the correlation to be low among the features, as it "fits a polynomial based on neighboring points and predicts the missing entries" [93]. The maximum likelihood based method requires a distribution assumption for the target variable, which serves to estimate the parameters in order to infer the entries where data was missing [93]. The expectation-maximization (EM) method iterates through expectation and maximization steps to finally arrive at a parameter estimate [93]. Multiple imputation methods assume at least MAR mechanism for the missing values, and "obtain a distribution via Monte Carlo simulation for missing data elements, which is computationally more intense compared to the aforementioned approaches [93]. A literature survey carried out by Young (2011) showed that the multiple imputation technique is more robust than the statistical methods [98]. Unsupervised machine learning based methods use clustering approaches, and supervised machine learning based methods use classification or regression (e.g. decision trees, random forests) to impute the missing values.

In line with a systematic mapping, Young (2011) compiled an overview of missing value treatment methods, which is illustrated by Figure 2.3 [98]. The first category are the "simple methods", which comprise list-wise and pair-wise deletion, are suitable for both categorical and numeric data, and just delete incomplete observations or features, which entails the downside that valuable data gets lost [98]. The *"pre-replacement non-*

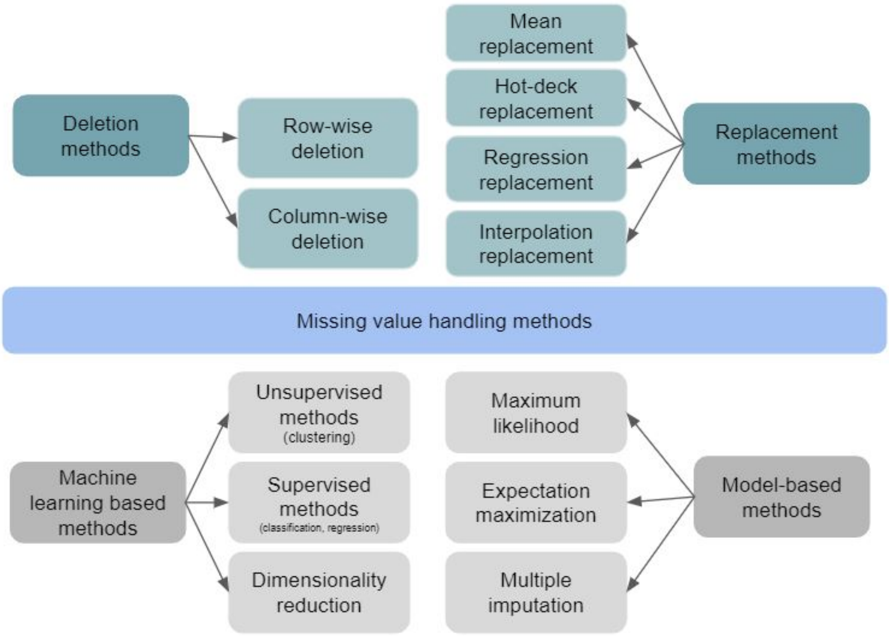


Figure 2.2: Methods for missing value treatment by Xu (2015) [93]

statistical methods“ do not rely on statistical models, but rather on similar observations, and are suitable for both categorical and numeric entries [98]. The *“pre-replacement statistical methods“* comprise mean-mode replacement techniques (for both numeric and categorical entries), linear regression (for numeric values) and logistics regression (for categorical entries) methods [98]. Furthermore, Young (2011) distinguished among the machine learning based methods between *“pre-replacement machine learning missing value methodologies“* such as replacement under same variation, nearest neighbor, k-nearest neighbor (kNN), EM, multi-layered perceptron (MLP), support vector machine (SVM), further neural networks types, which are mainly iterative methods making use of random number generation, and thus lead to higher computational effort [98]. Compared to the aforementioned methods which impute only one single entry, multiple imputation methods replace several missing values, and are recommended for settings with more than 5%-15% of all datapoints because they appear to introduce less bias into the dataset [98]. According to the papers examined in the survey, multiple imputation appears to be a appropriate approach to deal with missing values in case of up to around 25% of values missing in a dataset [98]. In the category *“embedded machine learning missing value methodologies“*, Young (2011) mention decision and regression trees, and its variation C4.5, robust association rules, and several artificial neural network (ANN), which are able to “handle missing data internally“ [98].

Hasan (2021) considered only statistics based and machine learning based imputation methods in their literature review (see Figure 2.4) [36]. The main categories were split

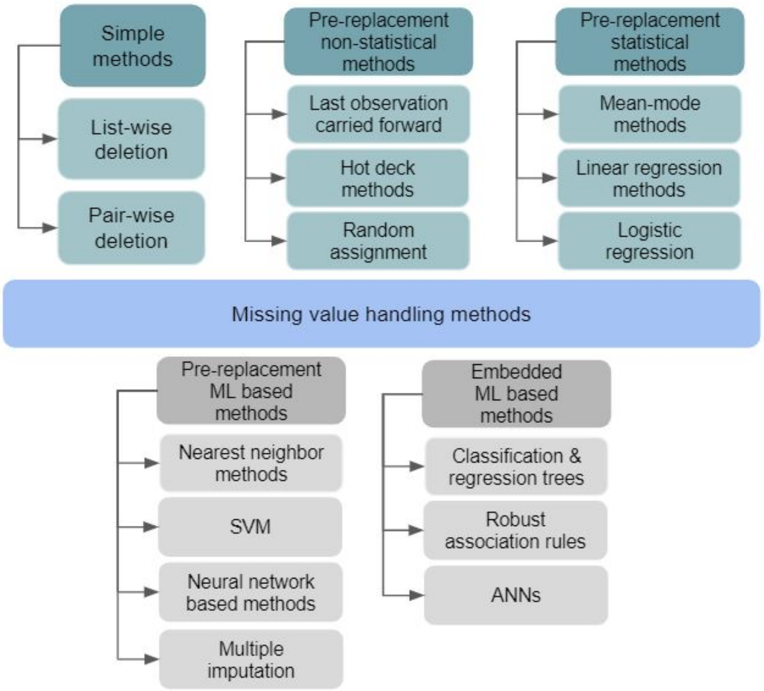

Simple methods
List-wise deletion
Pair-wise deletion
Pre-replacement non-statistical methods
Last observation carried forward
Hot deck methods
Random assignment
Pre-replacement statistical methods
Mean-mode methods
Linear regression methods
Logistic regression
Missing value handling methods
Pre-replacement ML based methods
Nearest neighbor methods
SVM
Neural network based methods
Multiple imputation
Embedded ML based methods
Classification & regression trees
Robust association rules
ANNs

Figure 2.3: Methods for missing value treatment by Young (2011) [98]

into sub-categories, which contain somewhat similar imputation methods such as tree-based methods, kernel-based, or clustering methods, while mentioning more fine-grained methods [36].

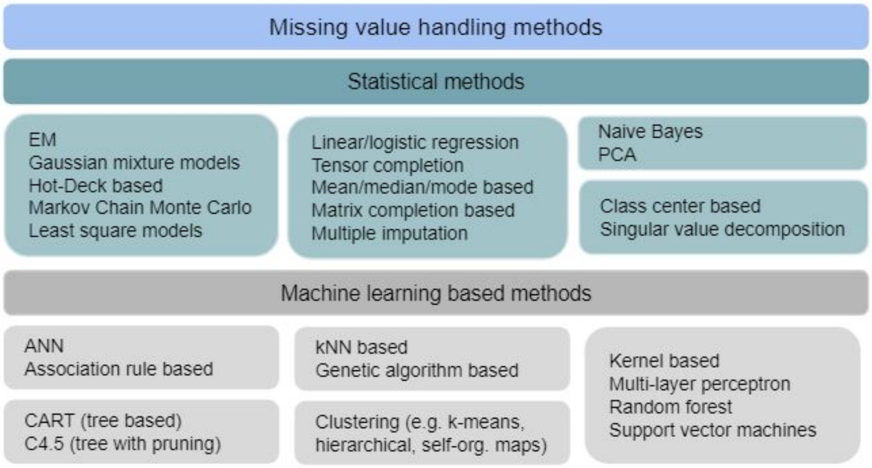


Figure 2.4: Methods for missing value treatment by Hasan (2021) [36]

Gond (2021) carried out a follow-up survey on the machine learning based approaches for missing value imputation with a focus on structured datasets, in which they ascertained that depending the choice missing value imputation method can have an influence on the outcome of learning algorithms performance metrics such as accuracy [31]. Mean and regression imputation are still the most frequently used techniques among practitioners, according to Gond's literature survey [31]. Again, it emerged that multiple imputation has the advantage of higher accuracy, but the drawback of higher computational effort. Machine learning based approaches comprise clustering approaches (such as k-means), ANN, association rule methods, SVM, self-organizing maps, classification and regression trees (CART) (or its adaptation C4.5), EM, recurrent neural networks (RNN), Naive Bayes, deep learning (referring to more complex neural networks), and kNN [31].

Young (2011) concluded that missing value treatment methods can substantially influence the predictive power of a dataset by decreasing variance or introducing bias [98]. Still, the No Free Lunch (NFL) theorem holds saying that there is no particular method that is the best method on each dataset and with regard to each problem, as one cannot generalize across all kinds of "attribute correlations, data distributions, amounts of missing values, sample sizes" [98], and outliers, normalization, the classifier type, computational effort, or complexity of implementation [36]. Gond(2021) stated as well that the performance of some methods can be higher based on selected datasets and combinations of variables, but not in general across all datasets, types, and situations [31]. Young (2011), later on also flagged by Nugroho (2019), also summarized from the literature survey that more studies regarding the "imputation efficiencies when dealing with large amounts of missing values" [98] should be subject to further research [98] [65] [23]. Gond (2021) confirmed those findings and suggestions for further research in their literature survey on machine learning based methods once more [31]. They also suggested that in case of around 50% or more of data entries missing, the performance differences between the algorithms will be more substantial [31]. Considering the use of missing value treatment in industry practice, Young (2011) stated based on the literature survey that the majority of practitioners did use either mean-mode, or some of the simpler single imputation techniques, and rarely more advanced statistical or machine learning based techniques, notably due to the availability of easily usable software, execution time (notably for some machine learning based methods), and the "time as well as statistical knowledge" necessary for implementing and comparing outcomes [98] [26].

Hasan (2021) conducted a literature review and analysis on how "missing value imputation affects the performance of machine learning" based on 191 papers published from 2010 to 2021 [36]. From this study focusing on machine learning based missing value imputation methods emerged that the topic has gained in importance over time [36]. The number of publications stayed at a quite stable level from 2010 to 2017, and then gradually more research had been contributed up to 2021, notably since 2018, as illustrated in Figure 2.5 [36]. For 2021, it has to be said that the literature review covered only publications until August, thus mid-year [36].

As already mentioned before, statistical experiments can be carried out with regard to data completeness. Next to the *direct evaluation* in case a complete dataset is available,

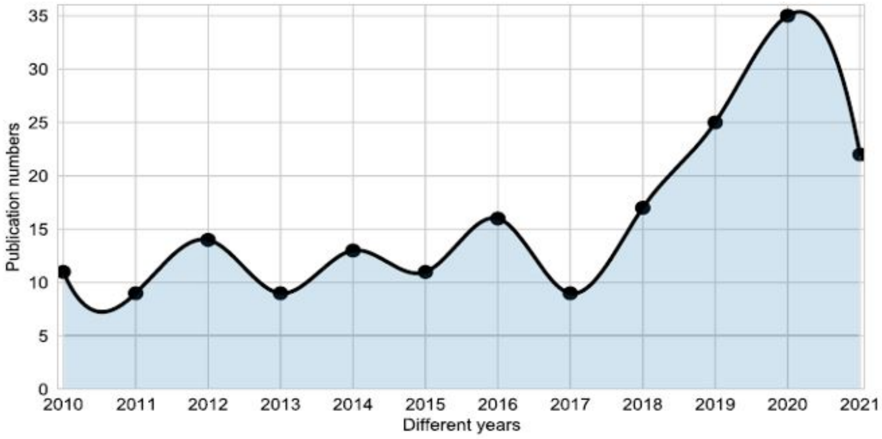


Figure 2.5: Development of the number of publications on machine learning (ML) based imputation methods from 2010 to 2021 by Hasan (2021) [36]

and from which several entries are deleted, then replaced again by estimations, and calculated what the share of correctly replaced entries is (which is notably used for categorical features), Hasan's (2021) literature review showed that the most comparable experiment set-up considering existing research is the one, whose experiment design is outlined in Figure 2.6 (adapted from Hasan (2021)) and covers both direct as well as indirect evaluation, which starts from a complete dataset (step 1), and subsequently makes it artificially incomplete via simulated deletion of entries according to the chosen random mechanism (step 2 and 3) [36]. Then missing value imputation (MVI) methods are applied to the incomplete dataset in order to replace the missing values by estimated values (step 4), which leads finally to a complete dataset again (step 5) [36]. Based on the complete dataset(s) that had undergone missing values imputation using one or more methods, and the original complete dataset (as ground truth), classification and/or regression methods are applied, and their performance metrics are evaluated, and finally compared (step 6), i.e. indirect evaluation, or the original complete dataset is compared to the imputed pseudo-complete dataset (step 6), i.e. direct evaluation [36]. Alamoodi (2021), Syafie (2018) and Dhungana (2021) suggested a similar experiment design [2] [84] [16].

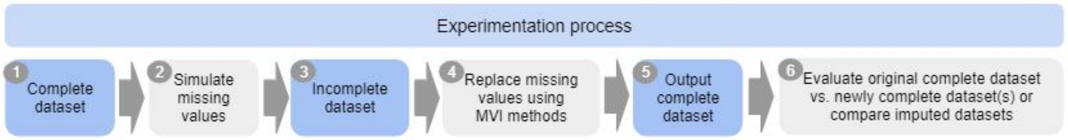

Experimentation process
1 Complete dataset
2 Simulate missing values
3 Incomplete dataset
4 Replace missing values using MVI methods
5 Output complete dataset
6 Evaluate original complete dataset vs. newly complete dataset(s) or compare imputed datasets

Figure 2.6: Data completeness experiment design adapted from Hasan (2021) [36]

In the literature review, Hasan (2021) also revealed that from 2010 to 2021, direct strategies have been largely used, and that most research studies either concentrated on the direct or the indirect evaluation, but did not apply both [36]. As for the statistical compared to the machine learning based imputation methods, Hasan's (2021) literature review exposed that around 60% to 70% of the studies concentrated on statistical methods [36]. Among the remaining studies that used machine learning based imputation methods, the most frequently used method by far was kNN, followed by random forest (RF), SVM, Bayesian principal component analysis (BPCA), and decision tree (DT) (see Figure 2.7) [36]. In case of classification methods, accuracy (used in 40% of the studies), specificity (14%), and sensitivity (12%) appeared to be the most popular performance evaluation metrics, according to Hasan's (2021) literature review [36]. However, it has to be mentioned at this point that accuracy does not take into consideration possible class imbalances, i.e. whether some categories are much more frequent than other [36] [52]. Failing to consider it might contribute to higher accuracy, but develop to the detriment of precision and recall [36]. This specific issue is for example accounted for in the AUC, and ROC, which are, however, have been used merely in around 9% of the research publications as performance evaluation metrics (from 2010 to 2021) [36].

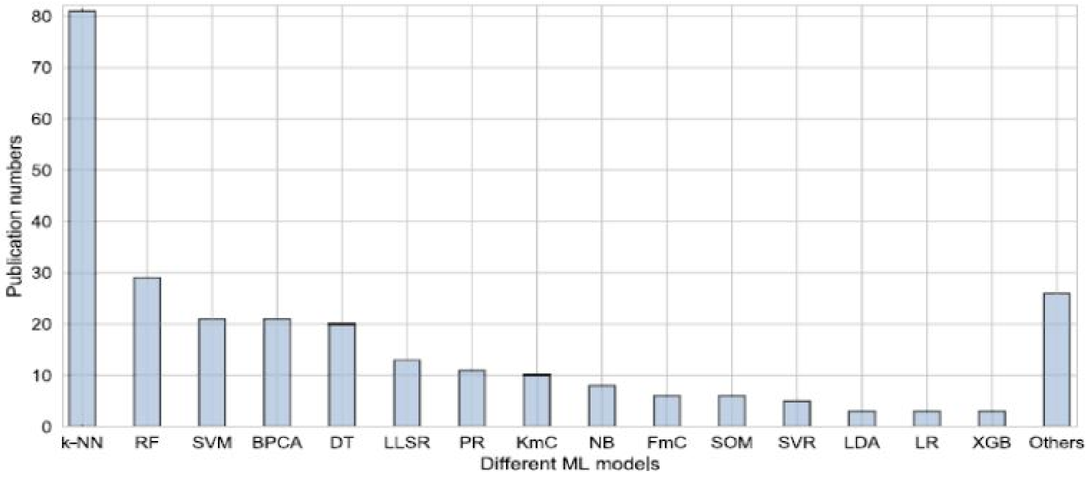


Figure 2.7: Use of machine learning based methods from 2010 to 2021 according to literature review from Hasan (2021) [36]

According to the literature review by Hasan (2021) that analyzed 191 publications, the direct evaluation of missing value imputation methods has been much more frequently used across the last ten years, except for the year 2012, which is also illustrated by Figure 2.8 [36]. Starting from 2018, the use of direct performance evaluation strategies reinforced even more [36]. One further aspect that Hasan (2021) pointed out is the question of when to carry out standardization in the context of distance based methods such as kNN, as the time of standardization might influence the performance metrics [36]. Although standardizing is suggested to occur prior to missing value replacement, there are also publications that propose it the other way around because the replacement might otherwise impact the center as well as scale of the data [36] [30] [2]. So, this aspect remains inconclusive with regard to a concrete recommendation or observations from previous research in the field [36] [30] [11]. On the whole, Hasan (2021) recommend to apply both the direct and the indirect evaluation, and in addition to this measure the performance in terms of execution time of the method(s), which has still been a gap in the experiment design of most research publications on the topic of missing value imputation so far which only concentrate on a very narrow evaluation setting, but would shed much light on how different MVI methods compare regarding their performance measured from various angles [36].

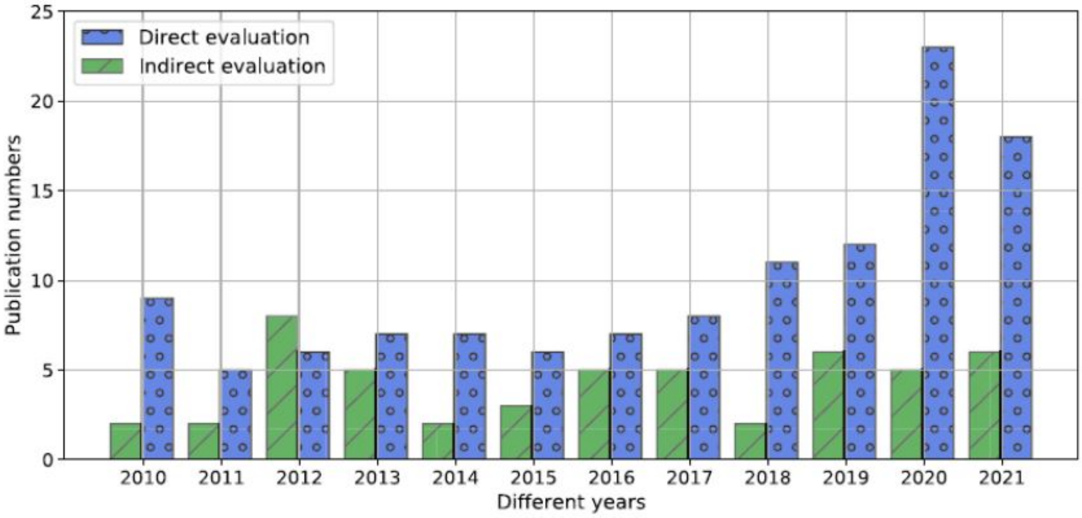


Figure 2.8: Use of direct vs. indirect performance evaluation from 2010 to 2021 according to literature review from Hasan (2021) [36]

An example for examining one particular method is a study from Latief (2020), who evaluated the performance of the missing value imputation method XGBoost directly in line with classification based on a medical dataset for several percentages of missing values (ranging from 5% to 20%), and the original complete dataset as a baseline [51]. Similar studies have been carried out by Gashi (2021), Sadhu (2020), Muhammadasraf (2021), Sowmya (2021), Wang (2020), Zheng (2016), and Sundararajan (2019) [28] [77] [35] [82] [87] [60] [83]. The kNN and the mean imputation were used as MVI methods, and the performance was evaluated in the form of confusion matrix metrics (i.e. accuracy, specificity, sensitivity, precision, recall) including cross-validation of the results [51]. Rusdah (2020) conducted a similar experiment with the XGBoost classifier but evaluated only the accuracy metric [76]. In contrast to this, for example Alamoodi (2021) carried out experiments using very high missing rates of 75% and more, and also mentioned that research studies concentrate usually focus on one of three ranges of missing rates, namely below 30%, 30% to 50%, or over 50% [2]. Dhungana (2021) carried out experiments including seven imputation methods from diverse categories, and using missing rates from 5% to 45%, evaluated based on the metrics RMSE and execution time [16]. Alamoodi (2021) additionally stated that most studies concentrated on simulating missing data, i.e.

starting from an initially complete dataset and making it deliberately incomplete by the means of some deletion strategy, and usually focus on only one dataset [2].

From the above, it can be seen that research activities in the area of missing value imputation had long been of stable interest, and have gained pace since around 2018 [36]. In practical settings, mean-mode imputation, and simple statistical regression techniques appear to be largely used, according to research from Young (2011), Gond(2021), Hasan (2021), Feldman (2018), and Alamoodi (2021), amongst others [98] [31] [36] [23] [2]. These claims about practical settings have been made based on loose observations in line with some publications. As a complement to these observations, this thesis aims to fill the research gap by conducting an online survey among practitioners in both industry and academic settings in order to find out more about how they deal with data completeness, how they are affected by it (e.g. which share of missing values, i.e. missing percentages, they face), and which methods they choose or rather avoid with regard to tackling missing values. The literature review prior to the online survey is intended to gain more insights into current practice, as well as how and in which context the respective missing value imputation methods are applied. Moreover, the methods used in the manufacturing domain will be explored. After the literature review and online survey, statistical experiments will be carried out in order to apply some of the techniques inspected, and to compare their performance based on classification and/or regression metrics both for direct and indirect evaluation using structured datasets and the MCAR mechanism. In addition to this, execution time will be captured. This approach is chosen to fill the research gap, as most studies look only at either direct or indirect evaluation as such, and many do not consider execution time [36]. So, this experimental approach is intended to gain multi-faceted insights into the performance of selected missing value imputation methods in line with two case studies. The experiment design and its phases is chosen as presented by Hasan (2021) (see Figure 2.6) and Dhungana (2021), which is a widely used research design for statistical experiments, and implemented as well by Latief (2020), Gashi (2021), Sadhu (2020), Muhammadasraf (2021), Sowmya (2021), Wang (2020), Zheng (2016), and Sundararajan (2019) [28] [77] [35] [82] [87] [60] [83] [51] [16]. Therefore, the statistical experiments aim to evaluate various missing value imputation methods in the direct and indirect evaluation setting, and with varying missing value rates.

CHAPTER 3

Research Questions and Methods

This chapter covers the research questions and methods. In the section 3.1, an overview on the research design is given. In the section 3.2, the research questions are laid out, the motivation for them is presented, and the solution approach in terms of research methods is shortly outlined for each research question.

3.1 Research Overview

When considering the roadmap for this thesis, the phases and methods are embedded in the design science framework as a paradigm for problem-solving. Design science aims to “produce and apply knowledge to create effective artifacts in context“ so that they achieve some objectives (i.e. requirements) set by the stakeholders [70]. According to Hevner (2004), “knowledge and understanding of a problem domain and its solution are achieved in the building and application of the designed artifact“ [39], or as Wieringa (2014) put it, design science refers to an iterative process of “designing an artifact that improves something for stakeholders and empirically investigating the performance of it in a context“ [91]. Venable (2017) described design science in a way that it “creates and evaluates artifacts to solve human problems“, possibly with iterations through some of the stages [86]. As illustrated by Figure 3.1, design science in information systems (IS) research develops and / or builds artifacts, then evaluates them in the form of analytical procedures, case studies, experiments, field studies, or simulations in an iterative manner [39]. Artifacts in the context of design science can be “for example methods, techniques, or algorithms used in software and information systems“ [91] [86]. There are feedback loops which can lead to the refinement of the artifact and again evaluation processes [39]. The relevance of the outcome needs to be stated in order to make sure that the goals of the stakeholders can be achieved, as the solution is being applied to the *environment* [39]. The *knowledge base* comprising the foundations (i.e. theories, frameworks, methods, models, and the like) as well as the methodologies (i.e. measures, formalisms, data analysis techniques) provides the “knowledge applicable“ to the artifact creation and respective evaluation [39]. Likewise, outcomes from the IS research project can also be relevant for the knowledge basis, and might be added to it for further research [39].

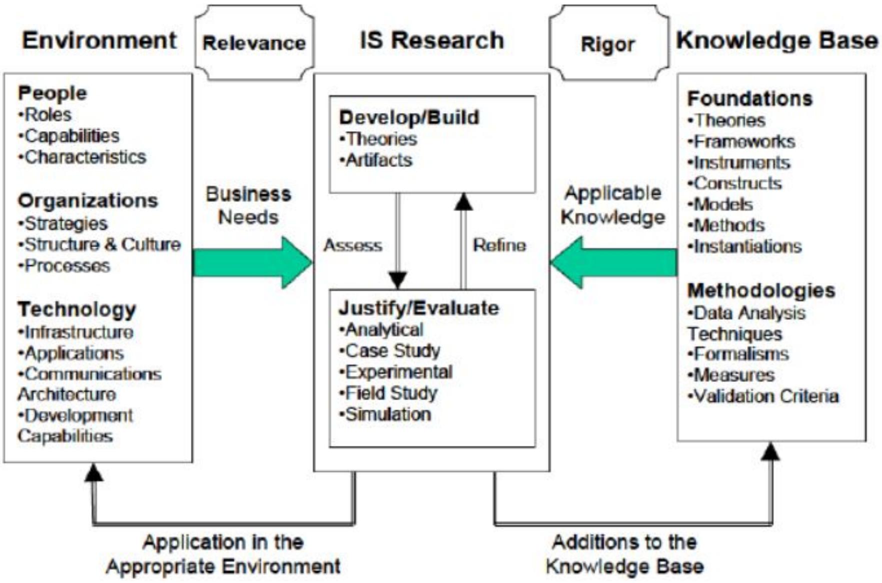


Figure 3.1: Design Science Framework for Information Systems Research by Hevner (2004) [39]

In this thesis, the roadmap is laid out in the IDEF0 diagram illustrated by Figure 3.2, which models the steps throughout the research phases including the timely sequence and the methods used. When considering the design science framework by Hevner (2004), the artifacts created are the results of the literature review, the online survey, and the algorithms involved in the statistical experiments [39]. They are assessed in line with both the open-source data case study, and the industry cases study, according to performance metrics, and using statistical baseline and tests, and applied to the chosen environment [39] [91]. The evaluation phase inspects how well the “artifacts have been transferred to the actual problem context“, notably in relation to the objectives set for the stakeholders [91]. The relevance of the problem for the *environment* and the proposed sequence of phases to close the research gap has been discussed in the chapter Related Work, which inspected the state of the art, and identified the research gap, and how this thesis can contribute to tackle it. With respect to the *knowledge base*, the thesis draws upon the theories, and constructs elaborated in the Literature Review, and the data analysis methods and statistical simulation or tests suitable, and performance metrics (i.e. evaluation criteria) for the Statistical Experiments. The research methods selected for the respective research phases are presented in the IDEF0 diagram.

The thesis starts from the research goal to enhance data quality by applying algorithms to tackle the data completeness dimension. The first research question (RQ1) is being answered by the means of the related work and the literature review in order to seize relevant data quality characteristics with a focus on data completeness. The literature review contributes to answering the second research question (RQ2), which intends to find out which methods and models are used in an academic as well as industry setting to handle missing values. This is complemented by the results from the online survey, which adds insights from practitioners. The information gathered from the related work, the literature review and the online survey sequence serve as a basis for the selection of the missing value imputation techniques to analyze and evaluate for research question 3 (RQ3). In line with an open-source case study and an industry case study, statistical experiments are carried out in order to compare several missing value imputation algorithms according to various performance metrics, so that conclusion regarding their suitability for the chosen datasets can be drawn. Overall, the whole flow of research phases relies on knowledge from existing quality standards, such as ISO standards on data quality, and from algorithms, statistics, as well as empirical software engineering (EMSE) best practices regarding the online survey and experimentation.

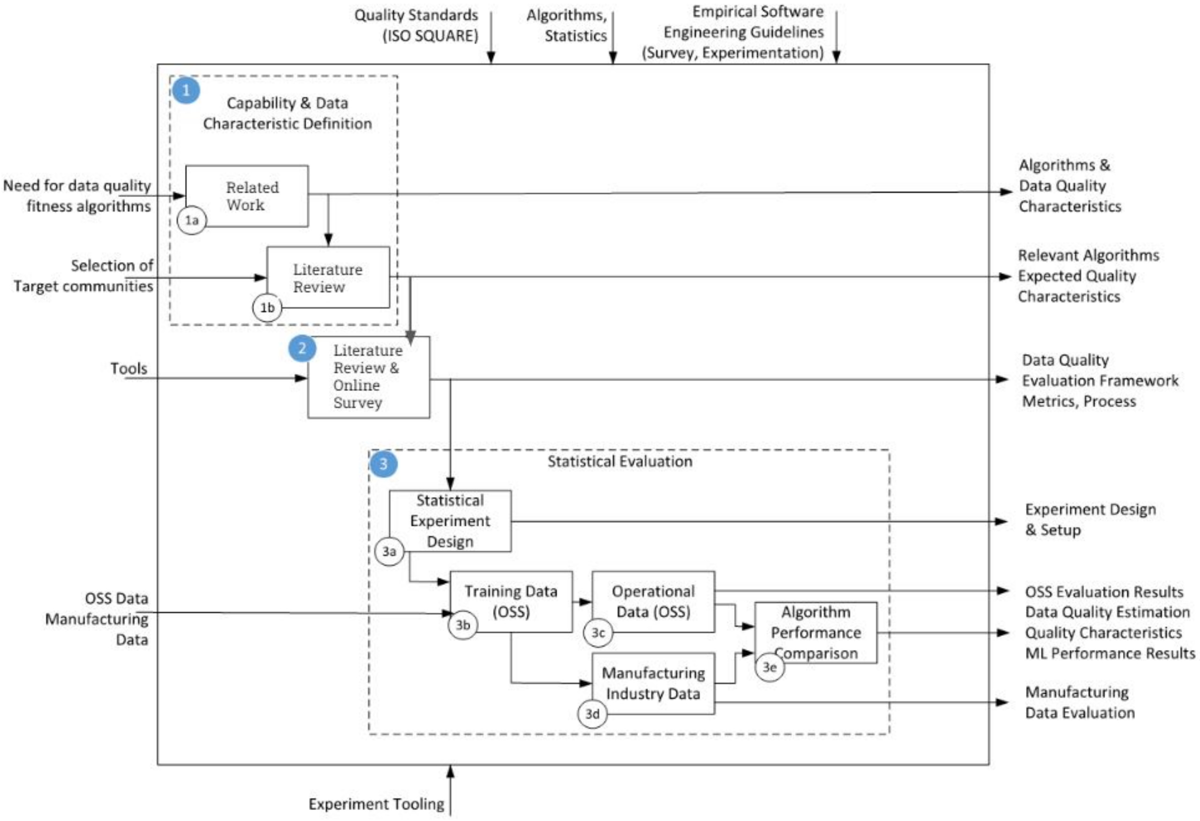

Quality Standards (ISO SQUARE)
Algorithms, Statistics
Empirical Software Engineering Guidelines (Survey, Experimentation)
1
Capability & Data Characteristic Definition
Need for data quality fitness algorithms
Related Work
1a
Selection of Target communities
Literature Review
1b
2
Literature Review & Online Survey
Tools
3
Statistical Evaluation
Statistical Experiment Design
3a
Training Data (OSS)
3b
Operational Data (OSS)
3c
Algorithm Performance Comparison
3e
Manufacturing Industry Data
3d
OSS Data Manufacturing Data
Algorithms & Data Quality Characteristics
Relevant Algorithms Expected Quality Characteristics
Data Quality Evaluation Framework Metrics, Process
Experiment Design & Setup
OSS Evaluation Results
Data Quality Estimation
Quality Characteristics
ML Performance Results
Manufacturing Data Evaluation
Experiment Tooling

Figure 3.2: Roadmap regarding research questions and solution approaches

3.2 Research Questions and Chosen Methods

This sub-chapter presents the three research questions of the thesis. The third research question comprises two sub-questions, named 3a and 3b, which are explained separately. At the beginning of each research question, the motivation for it is described. Then the research question is stated, and subsequently the solution approach in terms of methods used to answer the respective research question is outlined.

3.2.1 RQ1: Data quality dimensions and relation to completeness

The first research question (RQ1) is motivated by the fact that no single, uniform definition of data quality exists [45] [4] [85]. Instead, it is rather multi-faceted, and comprises a variety of dimensions (see for example Juddoo (2018) [45], Arbesser (2016) [4], Taleb (2016) [85]). Among these quality dimensions is the dimension of data completeness. Therefore, the thesis intends to shed light at what dimensions exist in current research, what different approaches are available in the literature to formulate and group data quality dimensions, and how the dimension of data completeness fits into the dimensions in terms of importance, frequency of usage, and content scope.

Therefore, the first research question is formulated as follows:

RQ1: ***Which dimensions does data quality have, and what is the relation of it to data completeness?***

Therefore, RQ1 aims to find out what the current state of data quality research with a focus on data completeness is. This is relevant in order to extract knowledge about current trends as well as shortcomings in the context handling data quality. How the RQ1 is embedded in the research and solution approach can also be seen from the flow diagram in Figure 3.2, in which it corresponds to part 1 (consisting of 1a and 1b) that elaborate on the capability and data characteristic definition. From a methodological point of view, the first research question can be answered by drawing upon the literature review and existing related literature, which provide thorough insights into the data quality dimensions models by various researchers, and how the data completeness is positioned among the quality dimensions.

3.2.2 RQ2: Techniques to handle data completeness

The second research question (RQ2) was motivated by the objective of illustrating what kinds of models and methods are available for handling missing values in datasets, i.e. how data completeness can be handled, both in an academic, as well as in a practical setting. It tries to explore what methods have been chosen in the literature, and what kinds of approaches can be distinguished. The thesis aims to find out how practitioners in the industry approach the topic of data completeness, and how relevant the techniques and approaches suggested by academics are to them. Likewise, RQ2 aims to present the current state of research, and how the various approaches can be evaluated and compared. It aims to find out how the ideas from research are handled in practice and explores in how far those methods are used in the industry setting, while also involving the expertise from industry professionals.

The second research question is phrased in the following way:

RQ2: ***What methods and models are used in research and in the industry to inspect, evaluate, and improve the handling of missing values throughout the lifecycle?***

When considering the solution approach for this research question, it will be answered in two ways. How the RQ2 is embedded in the research and solution approach can also be seen from the flow diagram in Figure 3.2, in which it refers to part 2 that helps to select the relevant tools, and obtain insights for the definition of the evaluation framework, performance metrics, and the processes necessary for the implementation of the experiments. Firstly, the literature review collects and summarized insights from research publications in the context of data completeness and missing values imputation methods, while considering as well their use in the industrial domain. The detailed results of this literature review are available in the chapter 4. Secondly, the online survey gathers insights on how data completeness is handled in practice, meaning how practitioners in the industry see the topic, and what kinds of techniques they apply to handle missing values. The combination of these methods, i.e. the literature review, and the online survey will also constitute the foundation for setting up the experiment design for the statistical subsequent experiments.

3.2.3 RQ3: Open-source and industry use cases on missing value imputation

The third research question (RQ3) refers the techniques how missing values can be imputed to make an incomplete dataset complete again. It notably intends to find out how suitable a selection of techniques are with regard to the chosen performance metrics in both the direct and indirect evaluation setting, while considering execution time too. As the research question 3 comprises two use cases, namely open-source data and industry data use case (the latter referring to CPPS), it is divided into section RQ3a and RQ3b, which refer to the respective use cases.

The overall third research question is formulated as follows:

RQ3: ***Which methods of handling incomplete data are most suitable with a view to enhancing data quality based on the selected data sets?***

The results from both RQ1 and RQ2 are used to plan the experiment design for the statistical experiments. In line with them, a comparison between the outcome of statistical experiments will be drawn based on open-source as well as industry datasets in the form of a case study for each setting. This approach is aimed to shed more light at concrete data cases, and to compare to the insights that were previously generated in line with RQ1 and RQ2. To address RQ3 fully, two sub-research questions that focus on individual domains, i.e., open-source data and industry data from CPPS engineering environments, are stated. The detailed experiment design, and the results of the statistical experiments are available in the chapter 6. How the RQ3 and its sub-questions are embedded in the research and solution approach can also be seen from the flow diagram in Figure 3.2, which is illustrated as part 3 with its components *Statistical Experiment Design* (3a), *Training Data (OSS)* (3b), *Operational Data (OSS)* (3c), *Manufacturing Industry Data (3d)*, and *Algorithm Performance Comparison (3e)*, and is explained in greater detail in chapter 6.

As for the research question 3a (RQ3a), it is based on an exemplary open-source dataset that could also be used in practice, and to which missing value imputation methods (in conformance with the literature review and the online survey) are applied. The objective is at this point to evaluate how the different methods of handling missing values may exert an influence on the overall data quality across subsequent workflows such as classification and / or regression steps according to a set of performance metrics.

The research question 3a is phrased in the following way:

RQ3a: ***Which methods of handling incomplete data are most suitable with a view to enhancing data quality based on the selected open-source dataset?***

From a methodological point of view, the research question 3a is answered by the means of statistical experiments according to the in respective research commonly used experiment design in the context of missing value imputation (see for example the literature review of 191 papers by Hasan (2021) [36]. The experiments explore whether tackling missing values can help increase data quality. Also, this approach examines the hypothesis that there is no specific method that is superior in all cases and with regard to all performance criteria, but rather specific application cases require tailored handling.

The research question 3b (RQ3b) involves the application of selected missing value imputation methods to an industry dataset, which is representative for the CPPS domain. Just like the RQ3a, it aims to evaluate how the different methods of handling missing values may exert an influence on the overall data quality across subsequent workflows like regression steps according to a set of performance metrics.

The research question 3b is phrased in the following way:

RQ3b: ***Which methods of handling incomplete data are most suitable with a view to enhancing data quality based on manufacturing data in an industrial context? In what way differ or correspond the results based on the open-source and industrial datasets?***

Considering methods, the RQ3b comprises the implementation of statistical experiments to evaluate the performance of several missing value imputation methods according to selected performance metrics in order to determine data quality subsequent to the imputations. It considers the industry case study referring to virtual commissioning, which leads to a “virtual sensor“ being created. This means that a series of data features is generated during the engineering and commissioning process. The information regarding the features is available at the beginning, i.e. in the set-up phase, but not in the operating phase of the process as it is assumed to would be too time-consuming to measure them continuously or the machine is unable to do so. Therefore, the missing values need to be replaced by more appropriate values to enhance data completeness. The question raised here is how well the missing data entries can be replaced by suitable values in order to increase overall data quality during the operating phase.

CHAPTER 4

Literature Review

The literature review aims to explore what methods and models are used (in research) to deal with missing values and data completeness thought the data science lifecycle. This method was chosen as it is suited to answer RQ2, which was previously indicated in the IDEF0 diagram in the chapter 3 (see Figure 3.2).

Concerning the methodological basis for this literature review, its structure and phases are used in a form adapted from Kitchenham (2009), who published a framework on how to set up and systematically conduct a systematic literature review (SLR) [49]. Its main goal is to gather state-of-the-art literature sources, and to extract insights from the main relevant publications among them [49]. Figure 4.1 depicts the phases for the systematic literature framework published by Kitchenham (2009). It starts with the preparation phase, which comprises the definition of search keywords, and generation of search strings [49]. Then the search engines are selected and the search strings are run in the selected search engines [49]. For the initial analysis, the search results from the relevant sources are collected, and are analyzed with regard to status, ID, year, publication type, title, and publication media [49]. Afterwards, the publications flagged for analysis are extracted, and screened again for further criteria such as publication objective, evaluation type, topic category, or number of citations [49]. The selection process from the search string to the final selection of publications is summarized in section 4.2 . The content of the finally selected publications is inspected and discussed in greater depth based on their full-text version [49].

4.1 Literature Review Preparation

The section “SLR Preparation“ firstly comprises the definition of search keywords. It portrays the formation of search strings submitted to selected search engines of electronic libraries and databases. In addition to this, it gives an overview on the relevant sources regarding online resources.

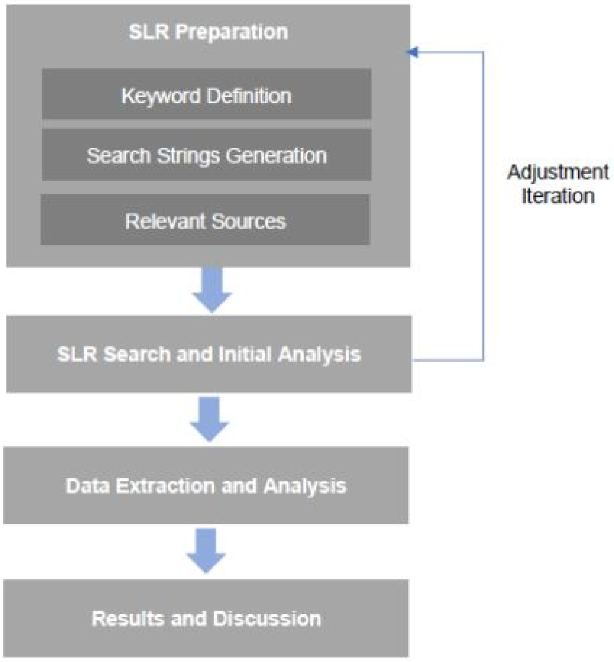


Figure 4.1: SLR framework adapted from Kitchenham (2009) [49]

4.1.1 Keyword Definition and Search Strings

Several initial key word combinations, as stated below, were run in the selected search engines. Finally, the query which is shown last in the list (which is highlighted in bold font) was chosen for proceeding further after an iterative adaptation of search queries, as it comprised a suitable number of hits in the databases. List of executed search queries, with the final chosen query written in bold as the last list item:

- *(missing value OR imputation technique OR imputation method OR data completeness OR missing value imputation) AND (manufacturing OR industrial)*
- *(missing value OR missing values OR missing data imputation) AND (cyber physical OR IoT or internet of things OR industry 4.0)*
- *(missing data imputation OR missing value imputation) AND (manufacturing OR industrial OR industry OR cyber-physical OR cyber physical OR IoT or internet of things OR industry 4.0)*
- *(missing data imputation OR missing value imputation) AND (manufacturing OR industrial OR industry OR cyber-physical OR cyber physical OR IoT or internet of things OR industry 4.0) AND NOT (time series)*

- ***(missing value OR missing values OR missing data imputation OR imputation technique OR imputation method OR data completeness OR missing value imputation) AND (manufacturing OR industrial OR industry OR cyber-physical OR cyber physical OR IoT or internet of things OR industry 4.0) AND NOT (time series)***

4.1.2 Relevant Sources

The search string highlighted above was executed in a selection of relevant publication databases of the corresponding academic digital libraries from the field of computer science. The following resources were considered:

- IEEE Xplore: `http://ieeexplore.ieee.org`
- ACM Digital Library: `https://dl.acm.org/`
- Web of Science: `https://www.webofscience.com/wos/history`
- Scopus: `https://www.scopus.com/search/form.uri?zone=TopNavBar&origin=searchadvanced&display=basic#basic`

In relation to the literature sources, the relevance of publications for this thesis was limited to scientific journals, conference proceedings and papers, white papers (only from official institutions and universities), university publications, and scientific books, as those were considered relevant and suitable for further analysis. The literature review comprises only resources that are available in electronic form in digital libraries. Firstly, this definition of scope was intended to ensure a stable scientific level of the literature. Secondly, the goal of the literature review was to gain insights on how missing value imputation is handled in academia, so this is the most relevant area.

4.2 Literature Search and Initial Analysis

As illustrated just above, several keywords and search strings were tried in the online databases of IEEE Xplore, ACM Digital Library, Web of Science, and Scopus. The exploration was narrowed down to the final search string *(missing value OR missing values OR missing data imputation OR imputation technique OR imputation method OR data completeness OR missing value imputation) AND (manufacturing OR industrial OR industry OR cyber-physical OR cyber physical OR IoT or internet of things OR industry 4.0) AND NOT (time series)*. The following table summarizes the number of search results obtained per database for the chosen search string:

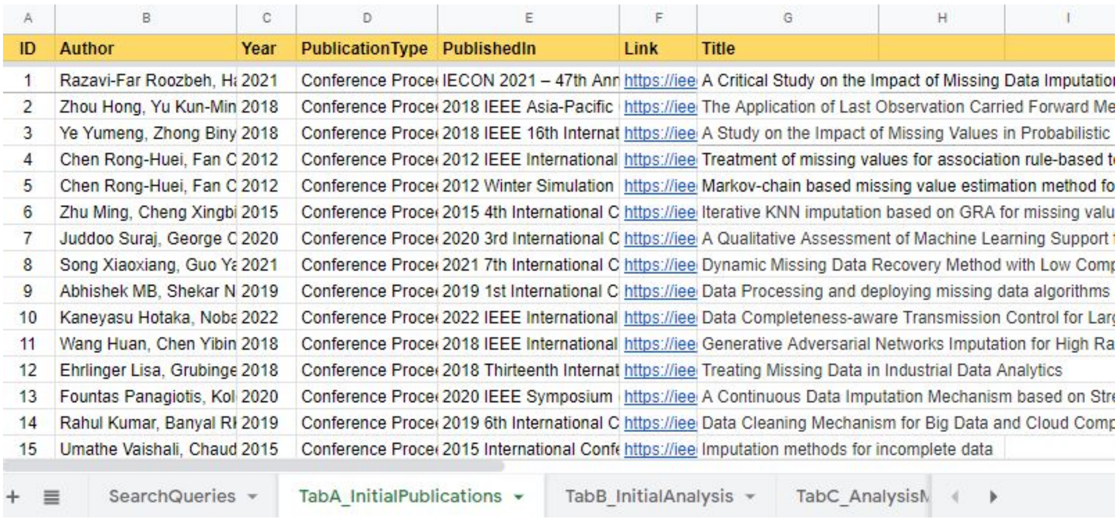
Given the number of search results and the quality of the first hits, the initial list of publications (see Appendix 8 for details) was extracted from the online databases of IEEE Xplore and Web of Science taken together, and was filtered for duplicates. The

Table 4.1: Number of hits per online library database

Digital library	Nr. of hits
ACM Digital Library	26
IEEE	80
Scopus	663
Web of Science	263

initial publications were assigned an ID for analysis, and furthermore the author(s), publication year, publication type, publication medium, and the title were captured in an Excel sheet. This procedure resulted in an initial publications list that comprised around 141 publications. Publications that occurred in the search results, but were obviously not suitable (e.g. regarding the research area) were already discarded prior the inclusion into the initial publications list. The list was saved in the Excel file `LiteratureReview.xlsx` in `TabA_Initial_Publications`.

Figure 4.2 depicts an excerpt from the list of initial publications, which includes the given ID, the author, publication type, publication medium, link for retrieval, and the title of the publication. The following plots give an overview on the time period in which the pieces of research were published, and the publication type.



ID	Author	Year	PublicationType	PublishedIn	Link	Title
1	Razavi-Far Roozbeh, H	2021	Conference Proce	IECON 2021 – 47th Ann	https://iee	A Critical Study on the Impact of Missing Data Imputatio
2	Zhou Hong, Yu Kun-Min	2018	Conference Proce	2018 IEEE Asia-Pacific	https://iee	The Application of Last Observation Carried Forward Me
3	Ye Yumeng, Zhong Biny	2018	Conference Proce	2018 IEEE 16th Internat	https://iee	A Study on the Impact of Missing Values in Probabilistic
4	Chen Rong-Huei, Fan C	2012	Conference Proce	2012 IEEE International	https://iee	Treatment of missing values for association rule-based t
5	Chen Rong-Huei, Fan C	2012	Conference Proce	2012 Winter Simulation	https://iee	Markov-chain based missing value estimation method fo
6	Zhu Ming, Cheng Xingbi	2015	Conference Proce	2015 4th International C	https://iee	Iterative KNN imputation based on GRA for missing valu
7	Juddoo Suraj, George C	2020	Conference Proce	2020 3rd International C	https://iee	A Qualitative Assessment of Machine Learning Support
8	Song Xiaoxiang, Guo Y	2021	Conference Proce	2021 7th International C	https://iee	Dynamic Missing Data Recovery Method with Low Comp
9	Abhishek MB, Shekar N	2019	Conference Proce	2019 1st International C	https://iee	Data Processing and deploying missing data algorithms
10	Kaneyasu Hotaka, Noba	2022	Conference Proce	2022 IEEE International	https://iee	Data Completeness-aware Transmission Control for Lar
11	Wang Huan, Chen Yibin	2018	Conference Proce	2018 IEEE International	https://iee	Generative Adversarial Networks Imputation for High Ra
12	Ehrlinger Lisa, Grubinge	2018	Conference Proce	2018 Thirteenth Internat	https://iee	Treating Missing Data in Industrial Data Analytics
13	Fountas Panagiotis, Kol	2020	Conference Proce	2020 IEEE Symposium	https://iee	A Continuous Data Imputation Mechanism based on Str
14	Rahul Kumar, Banyal R	2019	Conference Proce	2019 6th International C	https://iee	Data Cleaning Mechanism for Big Data and Cloud Comp
15	Umathe Vaishali, Chaud	2015	Conference Proce	2015 International Conf	https://iee	Imputation methods for incomplete data

Figure 4.2: List of initial publications

In this literature review, a time period ranging from 1999 to 2022 was taken into account, which is reflected in Figure 4.3. From Figure 4.3 that shows the list of initial publications grouped by year, it can be seen that the largest part of pieces of research under concern has been published later than 2016.

The initially retrieved publications are categorized as books, conference proceedings, and

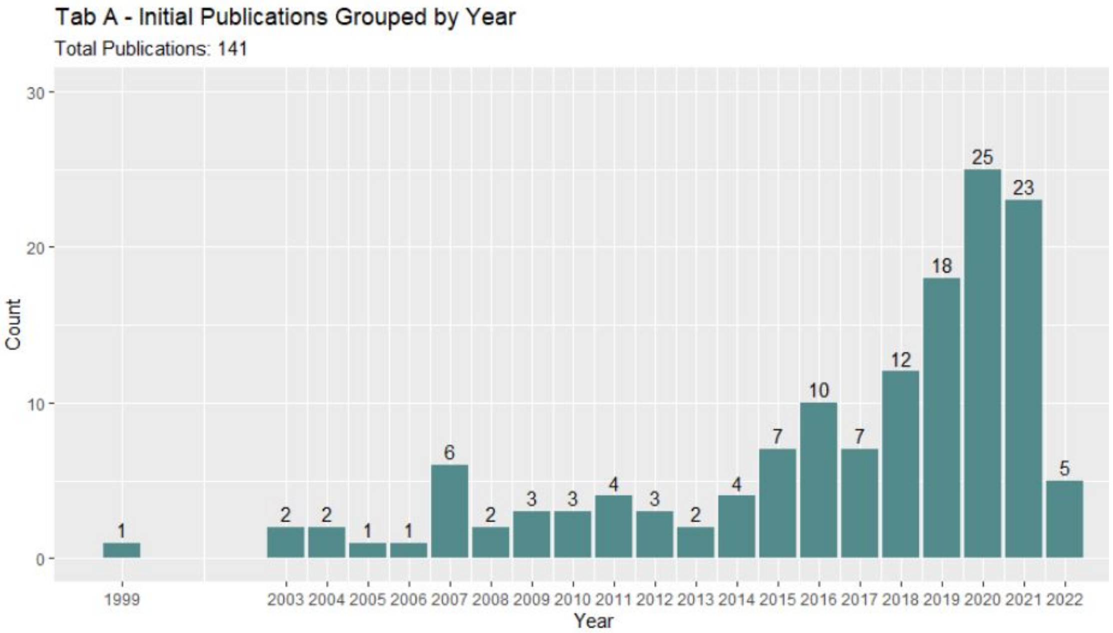


Figure 4.3: Initial Publications Grouped by Year

journals. The Figure 4.4 illustrates that most publications initially sourced are conference proceedings or journal articles, split around equally, and a very small part is sourced from books.

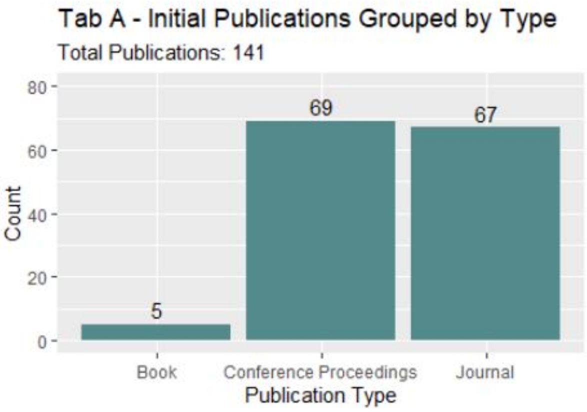


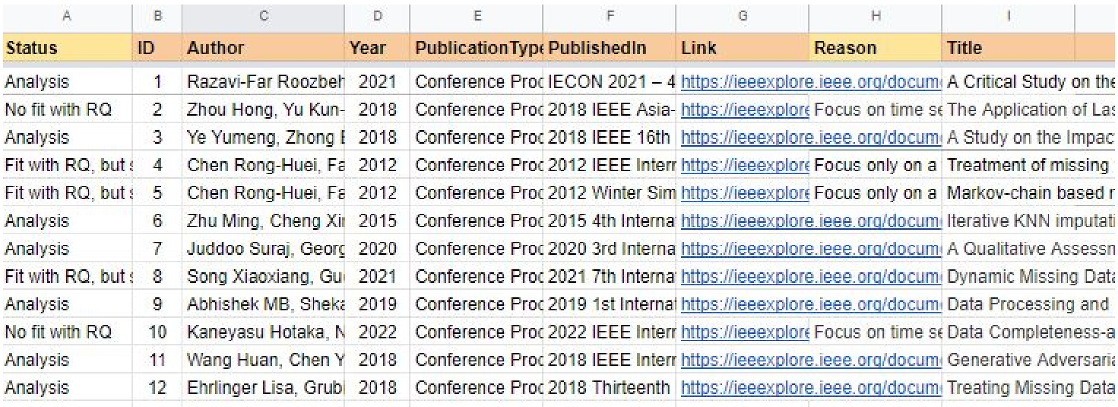
Figure 4.4: Initial Publications Grouped by Type

After the initial sourcing of publications and filtering for duplicate papers, the list of initial publications was reviewed with regard to detecting items that are not considered a good fit with the research question they should answer. Therefore, the list of initial publications was inspected in relation to content fit based on their title and abstract. In case the paper appeared suitable for contributing to answer the underlying research question, as stated above, the publication was included in the analysis matrix, i.e. the narrowed list of publications to be inspected in greater depth. For the analysis matrix, a number of publications around 20 was chosen as it was suggested by Kitchenham (2009) to include a narrow version of the initial literature list for more detailed analysis [49]. The tab `TabB_InitialAnalysis` comprises the information from the initial publications list extended with the status of inclusion or exclusion into the final publications list for content analysis. The column *Reason* is also added to the tab in order to provide more details to the status evaluation in case this appears necessary. Every publication from the initial list (see Appendix 8 for details) was assigned one status out of the status options indicated in table 4.2.

Status	**Description**
Access problem	The publication cannot be freely accessed (within the university access limits) or full-text is not available
No fit with RQ	The publication would fit at least one of the research questions, but its scope is limited to a very specific sub-branch of the topic, which does not offer the right level of granularity for this literature review
Analysis	The publication has a good fit with at least one of the research questions and offers a suitable granularity for the literature review so that it can be considered for the literature review

Table 4.2: Status options for publication evaluation

The figure 4.5 shows an excerpt of the list of initial publications (see tab `TabB_InitialAnalysis` in the Excel file), where status, and if necessary, reason was added.



A	B	C	D	E	F	G	H	I
Status	**ID**	**Author**	**Year**	**PublicationTyp**	**PublishedIn**	**Link**	**Reason**	**Title**
Analysis	1	Razavi-Far Roozbeh	2021	Conference Proc	IECON 2021 – 4	https://ieeexplore.ieee.org/docum		A Critical Study on the
No fit with RQ	2	Zhou Hong, Yu Kun-	2018	Conference Proc	2018 IEEE Asia-	https://ieeexplore	Focus on time se	The Application of Las
Analysis	3	Ye Yumeng, Zhong I	2018	Conference Proc	2018 IEEE 16th	https://ieeexplore.ieee.org/docum		A Study on the Impac
Fit with RQ, but s	4	Chen Rong-Huei, Fa	2012	Conference Proc	2012 IEEE Interr	https://ieeexplore	Focus only on a	Treatment of missing
Fit with RQ, but s	5	Chen Rong-Huei, Fa	2012	Conference Proc	2012 Winter Sim	https://ieeexplore	Focus only on a	Markov-chain based r
Analysis	6	Zhu Ming, Cheng Xi	2015	Conference Proc	2015 4th Interna	https://ieeexplore.ieee.org/docum		Iterative KNN imputati
Analysis	7	Juddoo Suraj, Georg	2020	Conference Proc	2020 3rd Interna	https://ieeexplore.ieee.org/docum		A Qualitative Assessn
Fit with RQ, but s	8	Song Xiaoxiang, Gu	2021	Conference Proc	2021 7th Interna	https://ieeexplore.ieee.org/docum		Dynamic Missing Data
Analysis	9	Abhishek MB, Sheka	2019	Conference Proc	2019 1st Internat	https://ieeexplore.ieee.org/docum		Data Processing and
No fit with RQ	10	Kaneyasu Hotaka, N	2022	Conference Proc	2022 IEEE Interr	https://ieeexplore	Focus on time se	Data Completeness-a
Analysis	11	Wang Huan, Chen Y	2018	Conference Proc	2018 IEEE Interr	https://ieeexplore.ieee.org/docum		Generative Adversaria
Analysis	12	Ehrlinger Lisa, Grub	2018	Conference Proc	2018 Thirteenth	https://ieeexplore.ieee.org/docum		Treating Missing Data

Figure 4.5: Excerpt of initial analysis sheet

The initial analysis resulted in a total of 27 publications selected from the initial publications list, which were flagged for closer analysis. The detailed list of selected publications is available in the Appendix 8. The Figure 4.6 illustrates the distribution of the chosen publications over the years of their publishing. It can be seen from Table 4.6 that most of the publications in the final selection stem from 2018 or more recent years.

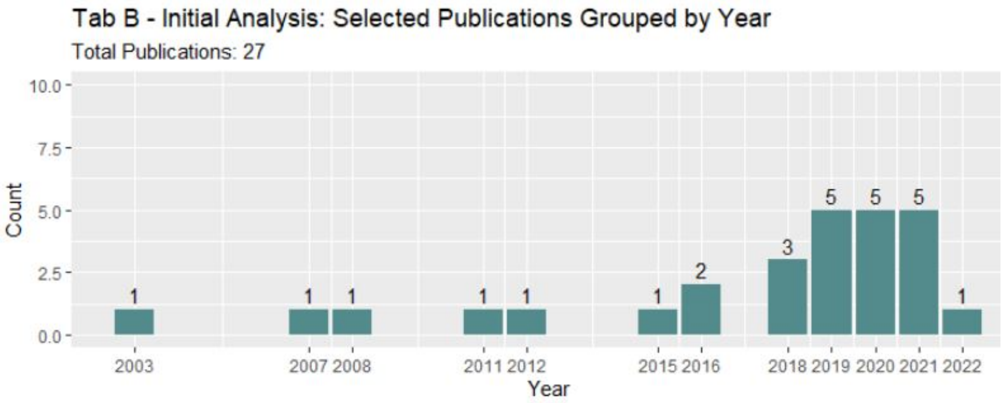


Figure 4.6: Publications selected in initial analysis by year

As a summary of the screening and selection flow, the Figure 4.7 illustrates the process of screening steps from the initial search string, along the search results in the publication databases, the removal criteria, and finally the resulting 27 publications that were considered for a more in-depth analysis.

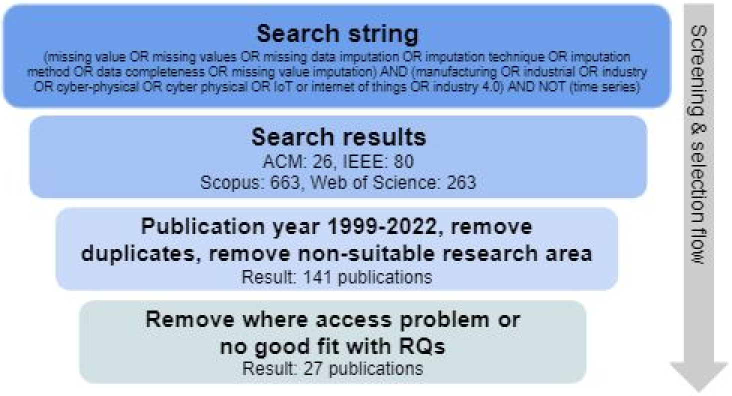
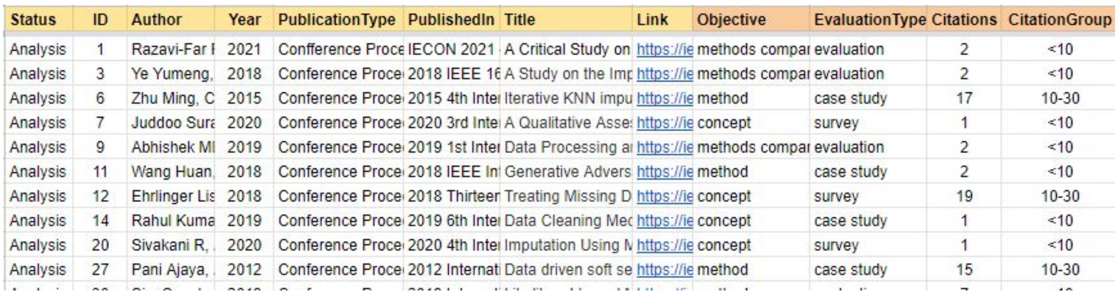

Search string
(missing value OR missing values OR missing data imputation OR imputation technique OR imputation method OR data completeness OR missing value imputation) AND (manufacturing OR industrial OR industry OR cyber-physical OR cyber physical OR IoT or internet of things OR industry 4.0) AND NOT (time series)
Search results
ACM: 26, IEEE: 80
Scopus: 663, Web of Science: 263
Publication year 1999-2022, remove duplicates, remove non-suitable research area
Result: 141 publications
Remove where access problem or no good fit with RQs
Result: 27 publications
Screening & selection flow

Figure 4.7: Publication screening and selection flow

4.3 Data Extraction and Analysis

During the data extraction and analysis stage, the selected publications were inspected more thoroughly. The columns *objective*, *evaluation type*, and *citation group* were added to the final publications in the tab `TabC_AnalysisMatrix`. *Objective* refers to the main goal of the publication, which could be one of “method“, “methods comparison“, and “concept“. *Evaluation type* refers to the steps needed to get to the intended goal, such as case study, evaluation, or survey. *Citation group* describes the number of times the respective publication had been cited until the time of this thesis, which was queried from Google Scholar, and Scopus. The Figure 4.8 presents an excerpt of the analysis matrix.



Status	ID	Author	Year	PublicationType	PublishedIn	Title	Link	Objective	EvaluationType	Citations	CitationGroup
Analysis	1	Razavi-Far F	2021	Confference Proce	IECON 2021 -	A Critical Study on	https://ie	methods compar	evaluation	2	<10
Analysis	3	Ye Yumeng,	2018	Conference Proce	2018 IEEE 16	A Study on the Imp	https://ie	methods compar	evaluation	2	<10
Analysis	6	Zhu Ming, C	2015	Conference Proce	2015 4th Inte	Iterative KNN impu	https://ie	method	case study	17	10-30
Analysis	7	Juddoo Sura	2020	Conference Proce	2020 3rd Inte	A Qualitative Asse	https://ie	concept	survey	1	<10
Analysis	9	Abhishek Ml	2019	Conference Proce	2019 1st Inte	Data Processing ar	https://ie	methods compar	evaluation	2	<10
Analysis	11	Wang Huan,	2018	Conference Proce	2018 IEEE In	Generative Advers	https://ie	method	case study	2	<10
Analysis	12	Ehrlinger Lis	2018	Conference Proce	2018 Thirteen	Treating Missing D	https://ie	concept	survey	19	10-30
Analysis	14	Rahul Kuma	2019	Conference Proce	2019 6th Inte	Data Cleaning Mec	https://ie	concept	case study	1	<10
Analysis	20	Sivakani R, .	2020	Conference Proce	2020 4th Inte	Imputation Using M	https://ie	concept	survey	1	<10
Analysis	27	Pani Ajaya, .	2012	Conference Proce	2012 Internati	Data driven soft se	https://ie	method	case study	15	10-30

Figure 4.8: Excerpt of the analysis matrix

4.4 Results and Discussion

The following sections give on the one hand an overview on the final selection of publications (see Appendix 8 for details). On the other hand, the content of the selected publications is explored. They provide insights into the chosen publications, which led to the final result of the analysis matrix, and they intend to answer the related research question.

4.4.1 Final Selection of Publications

When it comes to the main objective of the publications, it turned out that most of them aim to achieve a concept or methods comparison, and a smaller amount of them presents a (new or adapted) method, as it can be seen from Figure 4.9.

The evaluation type of the publications was considered, which showed that most papers were classified in the category of evaluation or case study, with a smaller part of them (i.e. six occurrences) being of survey type (see 4.10).

The number of citations was queried from Google Scholar and Scopus, and is summarized in Figure 4.11. Most of the pieces of research were located in the citation groups up to 30 citations. However, it has to be taken into account that the majority of the selected papers is very recent, so they naturally have been available for a narrower time frame to be cited.

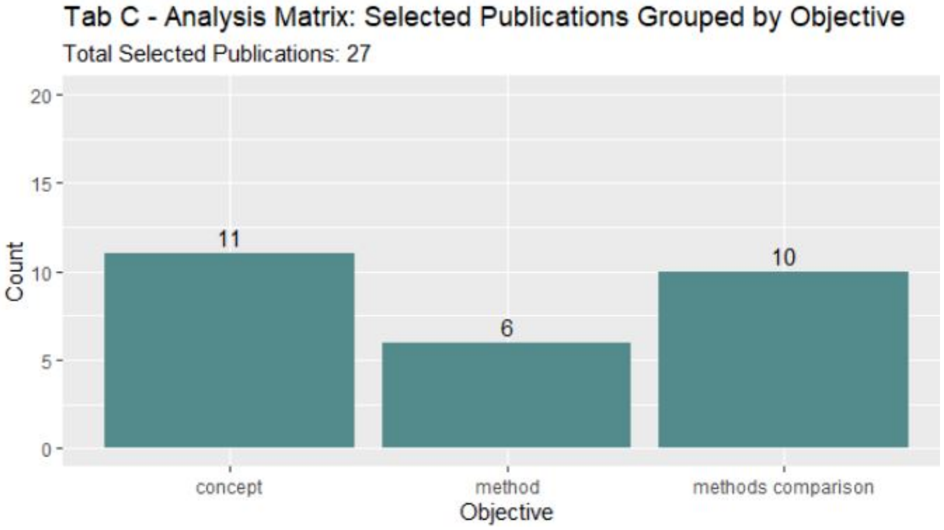


Figure 4.9: Selected publications by objective

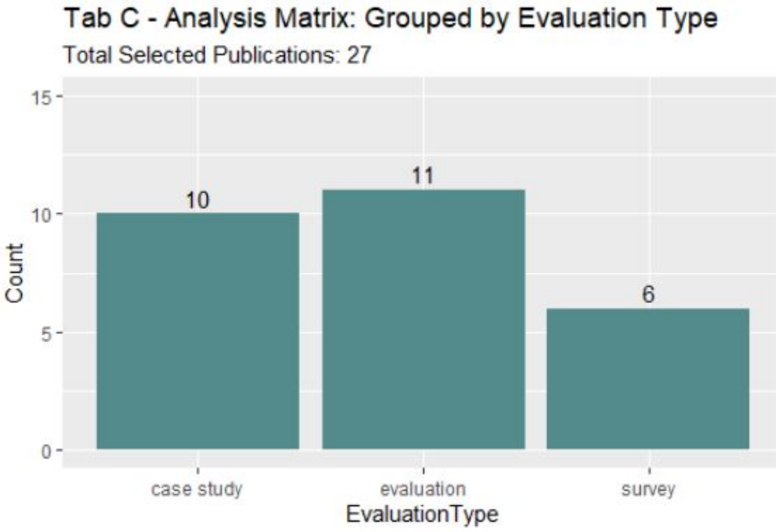


Figure 4.10: Selected publications by evaluation type

4.4.2 Content Analysis of Selected Publications

Next to data accuracy issues, data completeness issues are the most prevalent data quality dimensions to deal with in the area of data analytics [44] [45] [57] [11]. At least an estimated 10% of data, which is still seen as a relatively moderate missing rate, are missing in practical settings, which requires some action with regard to this topic, be it leaving out the missing entries completely or imputing them [71] [77] [80]. Whatever option is selected, it can have a decisive impact on further data analysis processes. As Juddoo (2020) and Rahul (2019) put it, machine learning and statistical techniques

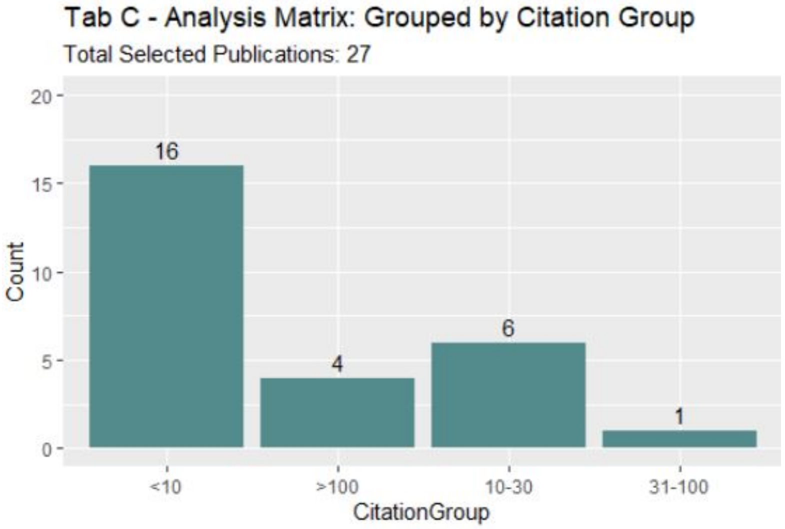


Figure 4.11: Selected publications by citation group

are paramount to counter those issues [44] [71]. In line with their research, Juddoo (2020) proclaimed that "there is no unique machine learning algorithm most suitable to deal with" the aforementioned data quality dimensions, event though they observed regression-based techniques to be tendentially superior [44] [11].

Therefore, different algorithm approaches have to be weighed against each other in terms of drawbacks and benefits in every new situation, depending on the characteristics of the dataset, such as size, variable types, and industry [44]. For this reason, this literature review aims to present examples of how different supervised algorithms were applied to various structured datasets, and how they performed in the respective situations. The limitations of the algorithms used can play a role for the efficiency and performance impact [44] [77] [80] [32] [78]. Given those facts, it can even happen that the missing value imputation algorithms introduce new bias and errors to the dataset, which highlights the need to actually examine the application of various missing value imputation techniques to the dataset(s) under concern [44].

According to a literature review from Young (2011), practitioners are supposed to use simple methods such as selective dropping of entire rows or columns, or replacement of the missing values by one fixed value (e.g. median, mean, mode), mostly due to their convenience in terms of time and effort needed to implement [98]. However, a concrete questionnaire has not been carried out yet among practitioners to find out about that aspect [98] [36] [77] [78]. The overall techniques available to deal with missing values can be divided into "disregarded methods, pre-replacement methods, and embedded methods" [98]. The *disregarded methods* (or to put differently removal methods) focus on dropping instances (i.e. row-wise, or also called list-wise deletion) or features with missing values as a whole [98]. At this point, with low missing rates such an approach might still work, but in case of higher missing rates, a considerable amount of valuable data is left out from the analysis [98] [22]. *Pre-replacement methods* put a statistically calculated characteristic (e.g. the mean, median, mode) at the place of the missing entry [98]. For example last observation carried forward (LOCF), which takes the previous existing sample's values as a replacement for missing values, hot- and cold-deck imputation, where the replacement value is chosen from a set of most recently seen similar observations in the currently inspected dataset in case of hot-deck or from another dataset "not being currently processed" [98] in cold-deck, as well as random replacement belong to the non-statistical pre-replacement methods, as not statistical calculation or modelling is involved in those techniques [22]. Although they can under some conditions be useful, they might also not take into account dependencies between features, and introduce large variance into the dataset[98]. On the other hand, mean / mode / median imputation (or cluster-based variations thereof), and (multiple) linear and logistic regression form part of the statistical pre-replacement methods because they require the application of a concrete statistical formula to calculate the replacement [98]. Just like above, it is computationally effective, but could lead to understating the actual standard deviation in the dataset as the replacements are chosen to be around the central statistical distribution characteristics [98]. Moreover, the pre-replacement machine learning based methods comprise for instance expectation-maximization, kNN, self-organizing maps, SVM, and neural networks approaches, which make use of "random number assignments and iteration", but require higher computational effort [98]. According to previous research, machine learning based methods have exhibited good performance results across a variety of datasets, even when compared to embedded methods, so they should be included in any comparison of methods for a new application scenario [98]. As the above single imputation is not so well suited to a proportion of more 10% of missing values in the dataset, multiple imputation (e.g. multiple imputation by chained equations (MICE), predictive mean matching, ANN, regression trees) is recommended as it would not overstate performance metrics while understating the variance [98] [67] [96]. In multiple imputation, every missing entry "is replaced by a set of values drawn from an implicit model derived from all available values", which generates in fact several pseudo-complete datasets (used according to different likelihood values), and does not require too much higher computational effort and makes not restrictive normality assumptions [98]. Multiple imputation have achieved good performance and better emulate the actual variability of the data even in higher missing rate scenarios, according to previous research [98] [22]. *Embedded methods* use a statistical or machine learning technique to estimate the missing values, and "handle missing data internally" [98]. They include for instance classification and regression trees such as C4.5 or CandRT, ANN variations, and association rules [98]. Trees can handle the missing values among their nodes, while association rules determine patterns how they can go via relationships of constituent elements to estimations of missing values. What has to be taken into account when choosing the imputation method for a specific underlying dataset, is that the methods come with benefits and drawbacks such as inadvertent decrease of variance in the dataset, lost observations, or unwanted high computational effort [98]. It is therefore recommended that methods from several categories are explored on the dataset under concern in order to find the method(s) most suitable to the situation, and to clearly reflect on the limitation that they might entail while considering the circumstances (e.g size of the dataset, domain, number of features, data types) [98].

In a CPPS and Industry 4.0 setting, businesses make use of sensors for data collection from their manufacturing machines and devices in order to connect the physical with the digital world, and to provide analysis options for the generated manufacturing data, notably for the so called “key process input variables“ which have a substantial influence on the production output [11]. For example faulty sensors, erroneous machines, noise, sensor changes, or environmental variables form part of the reasons why data in the industrial domain might be incomplete [11] [22]. Besides, in manufacturing data preprocessing activities tend to be very extensive, meaning that they consume “around 80% of the project analysis time“ [11]. Although missing data is one of the main problems, more focus has been placed on outlier and anomaly detection by research so far [11]. Among the models well suited to industrial data, kNN, SVM, mean imputation, and simpler statistical methods for miss rate percentages lower than approximately 30 to 40% have been mentioned, while over 40% the features are usually purged as such [11] [89] [22]. In case of class imbalance, synthetic minority oversampling technique (SMOTE) can be added to re-balance the class labels in the dataset [11]. As an approach to the missingness analysis, Carbery (2022) suggest to visualize the missing values by feature and in the bigger picture compared to the full dataset and run correlation analysis to see dependencies between variables, then perform imputation, and finally evaluate the imputed values to find out about the quality of the imputation [11]. Figure 4.12 shows a missingness map across all variables of a dataset, where black indicates that the data is missing, and grey indicates that the data is available [11]. CART analysis, which goes through the data via tree-based structures, can likewise help to identify variables that have an impact on missingness, i.e. to explore which variables can help forecast the missing rates [11].

For the missing value imputation, Carbery (2022) applied complete case analysis, bag method, kNN, LOCF, mean as well as median imputation, random forests and a variation of the latter called *RFextra* in their respective implementations from R packages such as *caret*, *imputeT*, and *missRanger* [11]. In line with the experimental set-up, the dataset was split into 75% training data, and 25% test data, and the modeling was carried out using 5-fold cross validation and SMOTE to balance classes so that the majority class is not over-emphasized [11]. Carbery (2022) chose “accuracy, balanced accuracy, sensitivity, and specificity“ as performance metrics [11]. The figure 4.13 illustrates a typical summary of missing value imputation results for an indirect evaluation example, where the performance of the missingness method is determined by applying an XGBoost classifier to the pseudo-complete dataset after imputation, and measuring the selected performance metrics [11]. In this particular case, it turned out that the complete case analysis performed best in case of accuracy as a major performance criterion and without class balancing, while LOCF constitutes the most performant missingness method in case balanced accuracy and SMOTE are considered [11]. From the above, it can be seen

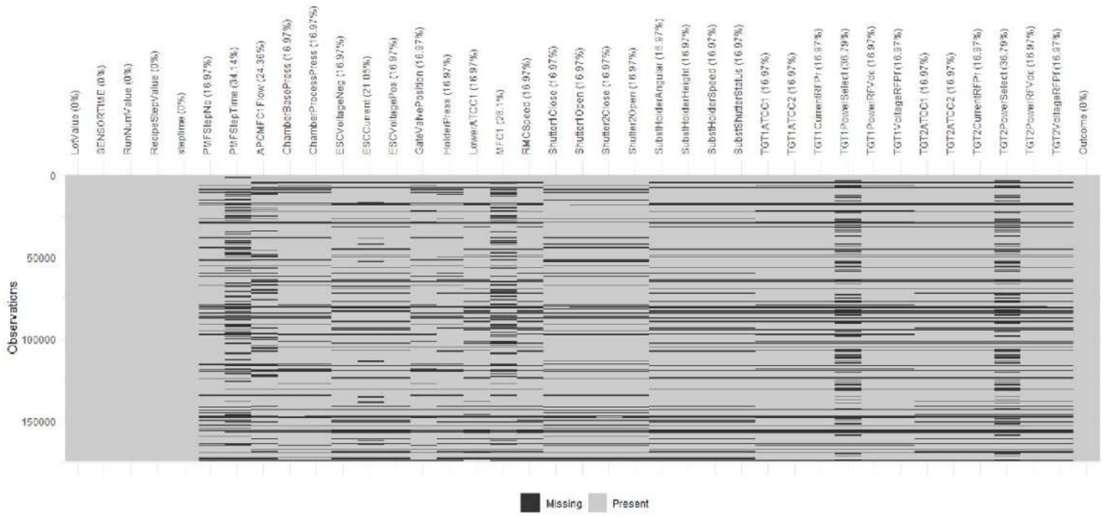
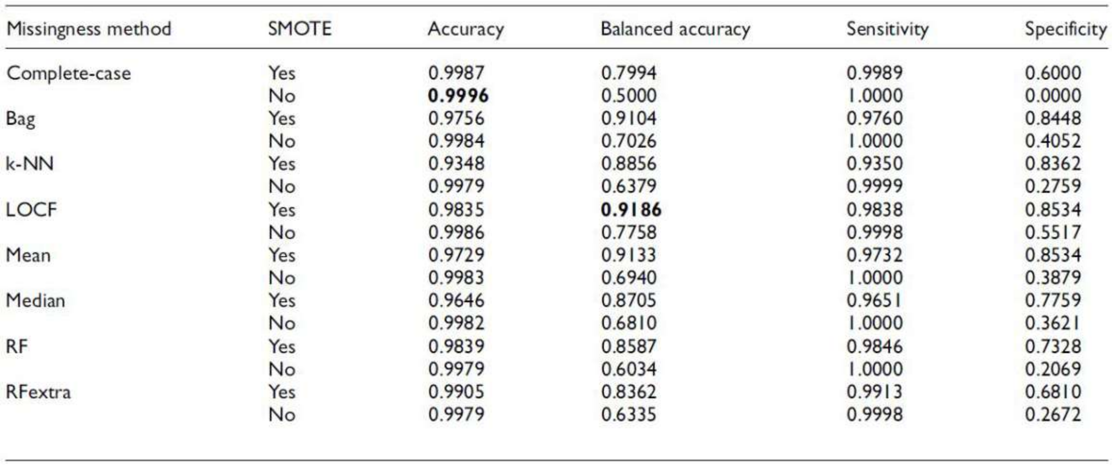


Figure 4.12: Missingness map [11]

that running a number of models is more beneficial to judge the effectiveness in a given situation than just looking at one single model result.



Missingness method	SMOTE	Accuracy	Balanced accuracy	Sensitivity	Specificity
Complete-case	Yes	0.9987	0.7994	0.9989	0.6000
	No	**0.9996**	0.5000	1.0000	0.0000
Bag	Yes	0.9756	0.9104	0.9760	0.8448
	No	0.9984	0.7026	1.0000	0.4052
k-NN	Yes	0.9348	0.8856	0.9350	0.8362
	No	0.9979	0.6379	0.9999	0.2759
LOCF	Yes	0.9835	**0.9186**	0.9838	0.8534
	No	0.9986	0.7758	0.9998	0.5517
Mean	Yes	0.9729	0.9133	0.9732	0.8534
	No	0.9983	0.6940	1.0000	0.3879
Median	Yes	0.9646	0.8705	0.9651	0.7759
	No	0.9982	0.6810	1.0000	0.3621
RF	Yes	0.9839	0.8587	0.9846	0.7328
	No	0.9979	0.6034	1.0000	0.2069
RFextra	Yes	0.9905	0.8362	0.9913	0.6810
	No	0.9979	0.6335	0.9998	0.2672

Figure 4.13: Experiment results: performance metrics using XGBoost classifier [11]

In the case of Juddoo (2020), several approaches were scrutinized, namely Bayesian regression, kNN, singular value decomposition (SVD), principal component analysis (PCA), and low rank matrix completion. As an experimental design approach, the concept mentioned in figure 2.6 (adapted from Hasan (2021)) was used, which is a well-founded method to compare missing value imputation methods [36] [44]. The experiments done (based on transportation data from the Melbourne intelligent transportation system) used different versions of missing rates of around up to 56%, which led in the end to the conclusion that least squares variants (e.g. local least squares) performed better (based on RMSE) than other methods in situations with low missing rates, and more complex imputations were better in scenarios with higher missing rates [44]. Random hot-deck imputation was named in the literature as an even more performant way, but was not explicitly inspected by Juddoo (2020). The missing value imputation process was run in different tools, including RapidMiner and Python, and it turned out that Python appeared to be better suited for this task, so Juddoo (2020) recommend their approach as Python offers missing value imputation facilitators such as *impyute*[1] or *scikit-learn impute*[2] [44]. Juddoo (2020) specified their approach as follows: [44][3]

- The feature containing missing values is named `y`.
- Then the data is divided into a part containing missing values (`X_test`), and without missing values `X_train`.
- Split the target variable (y_train) from the `X_train` data.
- Apply a missing value imputation algorithm, and predict `y_pred`.
- Append this result to the `X_test` data as a new column
- Merge the datasets.

Abhishek (2021) dedicated more specifically to missing value imputation algorithms considering cyber-physical applications, more precisely real-time data analysis in the context of water storage tanks using intelligent sensing and networking [1]. As one of the main limitations to predicting water storage levels, Abhishek (2019) mentioned missing values incurred during real-time monitoring comprising sensors and automated devices [1]. For replacing the missing values, mean imputation, kNN, expectation-maximization, and matrix completion were used so that the different methods were available for later comparison of fit to the dataset [1]. In *mean imputation*, all missing values are replaced by the mean value of the respective feature. In *kNN*, which is an online method, the nearest k entries measured, according to the chosen distance measure, to the observation under concern are considered for replacing the missing value by their average [1]. Based on prior experience, kNN achieves good imputation performance even in case of larger missing rates [1]. In *EM* imputation, the algorithm iterates through expectation stage (where it "ascribes esteems"), and maximization stage (where it "checks whether the esteem is in all likelihood") [1].

The authors chose several performance metrics including RMSE, MAE, and mean absolute percentage error (MAPE), and assumed that the data is missing (completely) at random [1]. Various missing rates ranging from 0% to 60% were considered, and the experiment

[1] https://impyute.readthedocs.io/en/master/
[2] https://scikit-learn.org/stable/modules/impute.html
[3] Note: Source refers to all bullet points in this list.

was set up approximately in accordance with the layout presented in figure 2.6 (adapted from Hasan (2021)), where the dataset was made pseudo-incomplete, and the missing values were imputed using different techniques [36] [1]. The concrete experimental design was set up as to include the following steps [1]:[4]

- Do preliminary cleaning.
- Make a copy of the original complete dataset.
- Delete some of the values in the copy, i.e. make them missing (up to 60%) randomly as well as to some determined non-random procedure
- Apply missing value imputation techniques to the test data
- “Compare original training data with imputed test data“
- Calculate and compare performance metrics (here: RMSE, MAE, MAPE)

Ehrlinger (2018) analyzed the treatment of missing values in the area of Industry 4.0 analytics, i.e. using a case study from a Voestalpine steel manufacturing plant, while applying different imputation techniques to replace missing values, which led her to the conclusion that transaction and production data is very much available, but the problem is in many cases the data quality [19]. It turned out that most research related to missing values does not refer to industrial analytics, but rather to social science analytics concentrating on survey data [19]. But notably in the industrial domain it occurs that a large number of observations lack complete data for all features due to faulty sensors or network communication, which makes dropping whole observations or features nearly impossible, even though the deletion of observations is the most widely used technique in practice due to its simplicity [19] [95]. However, it is suspected that in practice professionals dealing with incomplete data do not take it very serious, and furthermore find it hard to gain an overview across the usefulness of the various possible imputation methods [19]. For the analysis, Ehrlinger (2018) used a dataset, which comprised around 200000 observations of 560 features, composed of mostly temperature measures recorded during the steel milling and corresponding water cooling process [19]. Furthermore, among the challenges notably prevalent in the industrial domain, the size of data (tending towards or even big data), automated data collection by the means of sensors (where contacting non-respondents is not possible like in social sciences), non-normally distributed data (which needs to be reflected in the statistical significance test selection), and mainly numeric entries [19]. Zhu (2015) adapted the kNN imputation into an iterative version that it achieves higher performance in the area of trash pickup logistics if accuracy and speed to converge are considered [101]. As an experiment design, the one from figure 2.6 is largely used as well, with the addition of a reporting dashboard to visualize the results [36] [19]. To achieve the pseudo-incomplete data, on

[4] Note: Source refers to all bullet points in this list.

the one hand random deletion took place, and on the other hand, some data entries were used according to a pattern [19]. If the data is MCAR or free from patterns in the missingness, this could also be tested for using the Little's test which is somewhat a “chi-square test for missing values“ and indicates no pattern in case the p-value is not significant and the null hypothesis is not rejected [19]. If the p-value is significant, the missingness mechanism is likely MAR or MNAR [19]. According to Ehrlinger (2018), multiple imputation is the most widely used imputation technique, so they also applied this technique for their case study, which used Markov Chain Monte Carlo algorithm for joint modeling, Ward's method for clustering, correlation evaluation, and for numeric data linear regression (implemented in the programming language SAS) [19]. The performance was later evaluated based on MSE for all methods mentioned above, and compared to the actual values, i.e. the ground truth [19]. As a conclusion, the clustering and joint modeling approaches came in worst in terms of performance metrics, while partial correlation and linear regression led to the best results [19].

The challenging nature of treating missing values was also confirmed by Liu Yuehua (2020), who explore the impact of missing values imputation in industrial IoT sensor data in order to enhance reliability of monitoring processes as sensor measurements cannot easily be recovered as such once lost [57]. Their study was based on datasets from Australian manufacturing plants and focused notably on large availability gaps in sensor data (capturing amongst others “temperature, speed, vibration, and pressure“ features containing around two million measurements) generated from IoT devices using Loess approaches combined with time series methods such as seasonal trend decomposition for imputation and Bayesian multiple imputation in the multivariate case [57]. For the univariate case, interpolation (using splines and polynomials) as well as linear and logistic regression, and LOCF were applied [57]. As an addition to the existing methods, iteration was able to enhance the performance in terms of RMSE over other the state of the art (when compared to the real values) [57]. The experiments were implemented using the R packages *Zoo* (e.g. `na.locf`, `na.aggregate`, `na.approx` for interpolation, `na.spline`), and *ImputeT* (e.g. `na.seadec` for seasonal decomposition).

High quality data analytics applications require high quality input data, which also includes completeness. In case of incomplete data, most missing value imputation methods weigh each feature equally, and do not take into account their respective involvement in “downstream“ activities [95]. Yao (2021) pointed out that performance might be impacted by how missing values are treated, and considered this finding in their more refined method called fine-tuned generative adversarial network (GAN) (or in short FIGAN) model, which was tailored especially to industrial soft sensor applications [95]. For this purpose, a “pseudo-labelled soft sensor carries out data imputation and label prediction interactively“ that also takes into account possibly missing values in the sensor labels, whose effectiveness was tested in two case studies in the context of steel and penicillin manufacturing [95]. The FIGAN model is able to tackle “incomplete data directly in the training process“, then learns about the dataset's distribution, and performs multiple imputation [95]. Even during the imputation process, the FIGAN model considers the soft sensor performance changes in terms of the significance of downstream features (via a feedback loop), and offers a customized way of imputing dynamically, i.e. the "trained soft sensor is employed to forecast the labels for the unlabelled imputed data" [95]. So, the components of the FIGAN play together in a way that the generator creates the data according to a specific initial distribution, followed by the discriminator which "distinguishes between what values are observed and imputed, then the hint generator which collects information regarding the necessary mask matrix for the discriminator, and finally the soft sensor (e.g. trained with XGBoost) which helps to reconstruct the unavailable labels" [95]. As for the evaluation, again RMSE was used as a performance metric in addition to R^2 [95]. In the experimental set-up, which corresponds to the one flagged above (see figure 2.6), the new FIGAN model (with XGBoost for the final soft sensor) was compared to further GAN, random forest (more precisely *MissForest*), and mean imputation [95]. In line with the experiments, FIGAN model was able to perform better than the other techniques for some attribute combinations, but not for all, which suggests that it cannot be seen as generally superior [95]. When examining different percentages of missing data, it turned out that the FIGAN model achieved good results up to 50% missing values, but more than that led to a significant decline in performance [95].

Another approach in the area of intelligent manufacturing was proposed by Wang (2019), who used data from polymer production monitoring processes [89]. The issues at this point was as in the previously presented cases that the collected monitoring data was incomplete due to sensor and networking problems [89]. But in contrast to the GAN approach, an ordinary least squares (OLS) based autoencoder (i.e. a network with orthogonal hidden neurons which are selected using ordinary least squares estimation) was developed to generate replacements for missing values via the decoder part. The method was evaluated based on the MSE (in a cross-validated form) for various missing rates ranging from 2% to 25% via random removal in a dataset comprising around 10000 observations, and exhibited good performance compared to multiple imputation approaches for missing rates above 5%, but much worse than kNN and random forest for missing rates lower than that [89]. The experimental set-up was aimed to compare the OLS-based autoencoder approach with other imputation methods such as kNN, random forest, and several variants of GAN, the implementation was done in Matlab, and repeated 100 times [89]. The advantage of the autoencoder in this case was that it captured the highest variance of the original data to use it for predictions in an unsupervised way, and that it was able to keep the computational effort low by bringing in OLS evaluation for evaluating the contributions of the neurons [89]. For instance kNN is easily understandable, but choosing the distance function and number of neighbors takes time and with larger datasets computation power [89]. Random forests were also evaluated as the tree-based approaches do not require strict assumptions and can be applied in most scenarios [89]. Still, the larger the number of trees, the higher the computational effort for the method [89]. As also presented by Yao (2021) [95], GAN offer an extension to the multiple imputation approaches, but via a generative strategy by considering the "probability distributions over observed data via an adversarial process", meaning that the generative model seizes the probability distributions while the discriminative model “estimates the probability that a given sample datapoint comes from the real data rather than the generator“ [89] [88]. However, finding the actual distribution characteristics could also constitute a barrier [89] [88]. This excursion should highlight that although here the focus is placed on supervised techniques, there are new unsupervised techniques, which might be even more favorable under certain circumstances.

In the area of condition-based e-maintenance of industrial centrifugal compressors, Loukopoulos (2017) conducted a research study in line with that a number of imputation methods were applied to tackle the issue of incomplete data [58]. The experimental design was chosen in accordance with the common layout described in detail by Hasan (2021) [36] [58]. Applying a number of univariate as well as multiple imputation missing value imputation methods to a dataset with missing values in one feature showed that the latter group led to higher accuracy in the direct evaluation case irrespective of the percentage of missing values or the location of them in the dataset at hand [58]. The underlying dataset contained for instance information of the machine’s operational status, previous failure count, and further operations measurements regarding the process and mechanics – overall 25 variables and 474 observations – which is required to detect the need for maintenance activities as early as possible [58]. For the purpose of dealing with lost data entries mainly due to sensor failures and network connectivity issues, a set of missing values such as SVD, kNN, Bayesian PCA were applied to industrial centrifugal compressor data for the first time, according to the authors with the goal to determine the most appropriate one with regard to the performance metrics of accuracy and execution time [58]. From this experiment, Bayesian PCA and EM came out as the most performant considering accuracy in the case of 1-5% of missing data, but the former worst in terms of computational effort in the direct evaluation experiment setting [58]. On the other hand, kNN appeared to achieve the most robust results across varying missing rates and missing data locations in the dataset [58]. Other related works proclaimed that in similar domains kNN was able to achieve the best accuracy across missing rates of 1, 5, 15, 25, and 50%, and outperformed notably SVD and mean imputation [58]. Overall, mean and median replacement from the ad-hoc methods category, linear and cubic spline from the interpolation method category, autoregressive model from the time series category, self-organizing maps and kNN from the online method category, Bayesian PCA from the probability based category, and expectation-maximization algorithm from the statistical category were implemented and evaluated (based on normalized RMSE) in Matlab except for the EM one which was implemented in R [58]. In contrast to the other papers, Loukopoulos (2016) also explored the influence of missing data, if it occurs only in one variable, and not spread across the whole dataset [58].

Condition-related monitoring data, comprising around 100000 observations and 25 features, has been preprocessed as well by McMahon (2020), but in another area, namely railway asset management comprising infrastructure and rolling stock assets [61]. This includes for example performance, management, financial, or operations features [61]. The missingness mechanism is usually MAR or MCAR in this area, according to McMahon

(2020) [61]. For their study, data from various railway sites were collected, which were made pseudo-complete using EM, deep learning approaches (i.e. from more simplistic multi-layer perceptron to long short-term memory), and some time series models considering trends and seasonality, later on evaluated regarding accuracy [61]. For evaluation purposes, the authors chose RMSE as a metric, implemented the experiments in Matlab, and likewise did they use an experiment design set-up similar to Hasan (2021) [36] and Loukopoulos (2017) [58] [61].

What is more, several studies including Feldman (2018) tried to quantify by the means of simulation experiments the influence of incomplete data on subsequent decision-making quality via machine learning methods (such as linear discriminant analysis (LDA)), into which the incomplete or pseudo-complete data is fed [23]. Feldman (2018) inspected the application case of (binary) LDA classification after making an incomplete dataset complete [23]. Their research finally aimed to connect "data quality, model quality, and decision quality", meaning that the better the quality of the underlying data, then the better the model based on it can be fine-tuned, and the better the decision quality of subsequently applied supervised machine learning algorithms, and vice versa [23]. The results of this research led the researcher to proclaim that "with classifiers trained based on an incomplete dataset, the likelihood of assigning the wrong class to a test instance is higher" under the assumption of MCAR missingness pattern [23]. In the simulation experiments which were carried out in Matlab, kNN, likelihood-based, and Bayesian probabilistic models were used to impute the missing data, and evaluated based on "accuracy, precision, sensitivity, and / or specificity" [23]. A similar experiment was carried out by Razavifar (2021) who studied the "impact of missing data imputation for classifying intrusions in cyber-physical water systems", and used accuracy as well as F1 measure (i.e. a combination of precision and recall) for evaluation [74].

Feldman's (2018) research indeed confirmed the claims made by Cartwright (2003), who analyzed the contribution of kNN, hot-deck as well as sample mean imputation in the industrial software engineering data context, where missing values tend to naturally occur [23] [12]. To generate more robust replacement values, multiple imputation was used as well, which generated several full datasets that were later on brought together considering the uncertainty and probabilities [12]. Cartwright (2003) observed that the aforementioned imputation techniques substantially ameliorated the performance of subsequent machine learning activities in a situation of around 15% to 20% missing data entries out of the whole dataset, even though kNN and mean imputation were rather simplistic methods [12]. Likewise, Farhangfar (2008) led a research project that explored the influence of imputation by the means of hot-deck, single imputation, multiple imputation algorithms on the results of six classification methods (i.e. Ripper, C4.5, kNN, SVM with radial basis function (RBF) kernel, Naive Bayes)), which showed that given the underlying datasets an improvement over the situation without imputation could be achieved, and for missing rates ranging from 5% to 50% [21]. The experiment design used by Farhangfar (2008) is illustrated in figure 4.14 [21]. In line with the experiments, the data is split into training and test data at the beginning (see 4.14). The training set is then made pseudo-incomplete by deleting randomly some of the values (see 4.14). Then imputation methods are applied to make the dataset complete again, and arrive at the imputed dataset, on which classification methods are applied (see 4.14). Similarly, the test dataset that was initially set aside is now used to apply classification methods 4.14. Later on, performance in terms of accuracy is evaluated for the originally complete test dataset without imputation, and the imputed dataset. Overall, it can be said from the experiments that imputation contributes to increase classification accuracy for the methods in most scenarios, except mean imputation which proved only useful in the case of 50% missing rate [21]. According to Farhangfar (2008), “statistically significant improvements were achieve for the kNN, SVM with RBF kernel, and RIPPER classifiers, while C4.5 and Naive Bayes were missing data resistant“, and the change in missing rate does not influence the performance directly proportionately [21]. Nevertheless, there was no single best method that prevailed in terms of performance across all datasets analyzed [21].

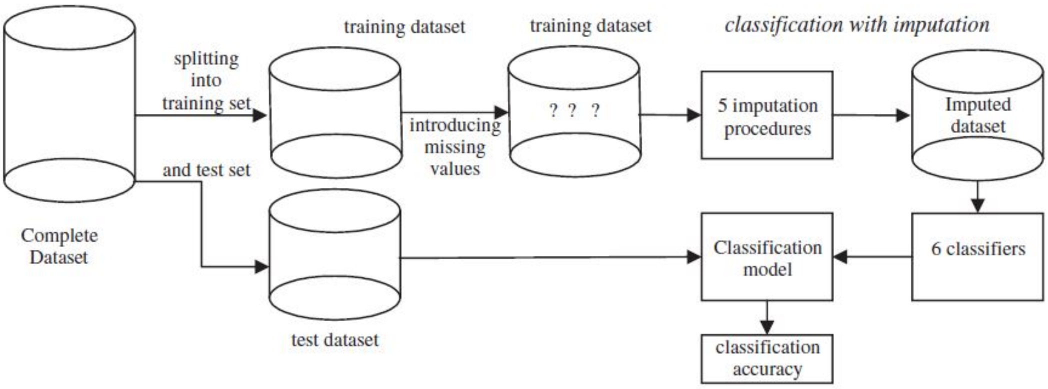


Figure 4.14: Experiment design by Farhangfar (2008) [21]

In this indirect evaluation case, step-wise regression, which selected the features having the highest coefficients in each step, was carried out to make predictions based on the pseudo-complete dataset, and evaluated based on the adjusted R^2 criterion relevant for regression that sums up the squared residuals [12]. The higher the adjusted R^2 criterion, the better the model’s goodness of fit [12]. As a last stage in the experiment, a statistical significance test, more precisely the non-parametric Paired Samples Wilcoxon Signed Rank test, was done in order to test whether the difference between the residuals was significant or not [12]. Compared to the non-imputed incomplete dataset, by using kNN, multiple imputation, and sample mean imputation the R^2 criterion could be increased from around 0.22 to 0.97 [12].

When it comes to IoT sensor networks, missing values tend to occur quite frequently for a number of reasons as stated above. Okafor (2021) examined the implications of such data issues due to missing sensor values for the on-site sensor calibration [66]. To get an understanding of how sensor calibration can be impacted by missing data, “variational autoencoders, neural networks with random weights, MICE, kNN, and random forest“ (using *missForest* implementation) were chosen as imputation techniques [66]. The dataset contained environmental sensor measurements across a range of particle and gas perception sensors from urban monitoring sites in Colorado (USA) used for air quality and traffic monitoring with usually around 20% missing rate [66]. This was followed by an indirect evaluation of the imputed dataset using supervised algorithms for sensor calibration, namely multiple linear regression, random forest, decision tree, and XGBoost algorithms, with grid search to determine the best parameters respectively [66]. Several missing rate scenarios were analyzed ranging from 5% to 70% of missing values, and RMSE, MAE, and R^2 were chosen for performance comparison [66]. The experiments were carried out in Python (version 3) in combination with Jupyter Notebook [66]. Erhan (2021) also examined a dataset from the area of environmental intelligent sensing with regard to the effectiveness of kNN and *missForest* as imputation methods which resulted a decreased execution time [20]. The plot that compares the experiment results is shown in figure 4.15, which depicts the chosen imputation methods' (variational autoencoder (VAE), neural network with random weights (NNRW), MICE, missForest and kNN) RMSE result per missing rate [66]. At this point, it has to be mentioned that a lower RMSE value is associated with a better performance. It can be seen from the figure 4.15 that higher missing rates lead to a higher error, i.e. worse performance [66]. As a result, the variational encoder arrived at the best position and achieved the highest performance, which represented an improvement over using the non-imputed incomplete dataset with a better sensor calibration than before [66].

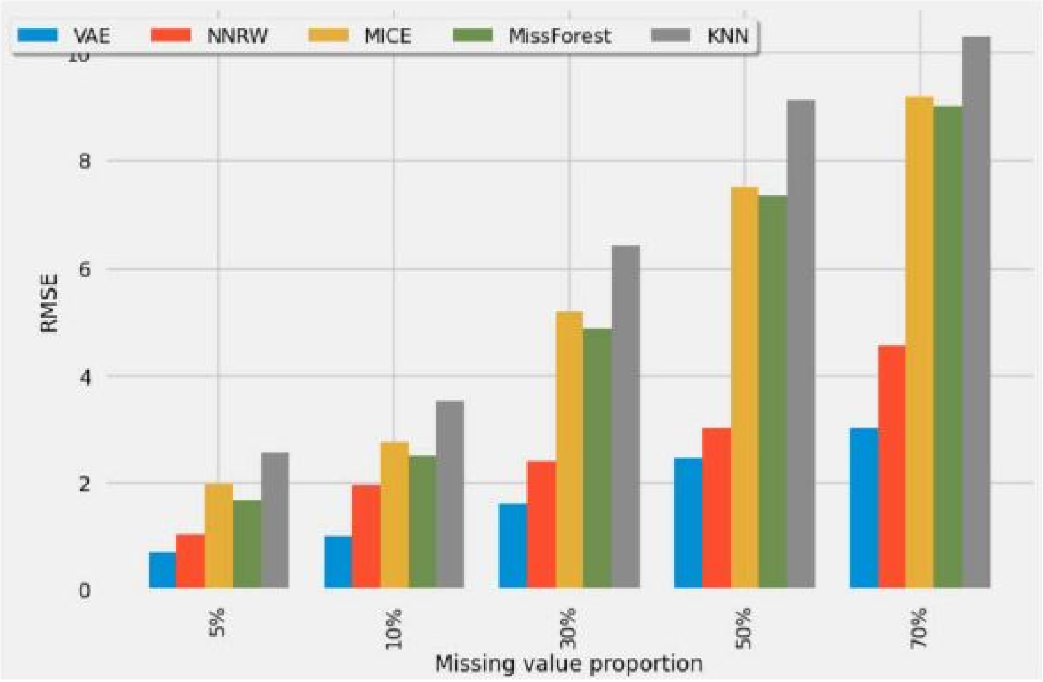


Figure 4.15: Experiment results: RMSE performance for different missing rates [66]

CHAPTER 5

Survey

This chapter refers to the online survey. It contains on the one hand the section Survey Design 5.1, which explains how the survey was set up, prepared and conducted. The second section 5.2 is dedicated to illustrate the survey results.

5.1 Survey Design

The online survey was design comprised several phases as it is presented in Figure 5.1 (adapted from [47], [100], [33]). As shown in the first phase, the survey was chosen as a method to answer the second research question (RQ2), i.e. what methods and models are being used in research and in the industry to inspect, evaluate, and improve the handling of missing values throughout the data analysis in practice. The focus was placed on structured datasets. It was conducted in the form of an online survey as the potential participant pool for this topic is well available via e-mail, online chat, discussion forums, and the like. The target group was specified as people working part- or full-time in jobs related to data analysis ranging from management and research to operations, software engineering, statistics, and data science roles across all domains.

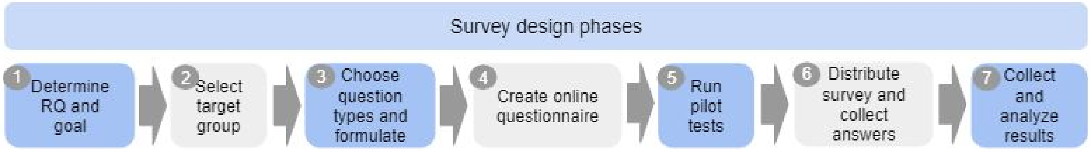

Survey design phases
1 Determine RQ and goal
2 Select target group
3 Choose question types and formulate
4 Create online questionnaire
5 Run pilot tests
6 Distribute survey and collect answers
7 Collect and analyze results

Figure 5.1: Survey design phases (adapted from [47], [100], [33])

Thirdly, the question types were chosen and the questions were formulated accordingly. The questionnaire comprised seven sections, each of which contained several questions. In the beginning, information on the *demographic background* was collected, which included the level of completed education, the main educational background, the main business area or domain of working of the participant, the current role as well as the experience in it. Furthermore, information about the size of the organization, programming languages used at work, age group, and gender were collected. As a question type single choice questions, i.e. of closed type were used, and an "other" option was provided to enter any information not captured by the given answer options. The next section referred to the *handling of incomplete datasets / missing values in practice*, which asked the participants to judge how they would handle datasets containing various percentages of missing data entries. They were asked to indicate the overall approach of their organization towards handling incomplete datasets, and its relative importance. All questions in this section were closed, single-choice questions. In the section on the *knowledge of methods to handle incomplete datasets*, the participants were asked to indicate their practical experience with a series of methods for missing data handling (such as mean / mode / median imputation, hot-deck imputation, replacement by constant, kNN, SVM) as well as their preference for a specific method, in case they had one. The questions this section were again of closed nature. Subsequently, the section on *software tools and choice of methods* dealt with what software packages and libraries the participants used in practice to tackle incomplete datasets in the form of a single choice grid across the methods, and one further open question where the participants could enter information on any additional, not yet mentioned package or library including the related programming language. The section on the *choice of methods and learning* referred to factors that might influence the decision on what method or model to choose for handling missing data, and what kind of background the participant had in his / her learning experience and education until then with regard to this topic. The former questions were single-choice questions. Finally, participants were offered the opportunity to share any remaining thoughts or experiences on handling incomplete datasets in practice. In the last section, the participants had the opportunity to indicate whether they would like to obtain the results of this survey, and further information on their organization and contact details. The full version of the online questionnaire can be found in the Appendix 8 for a more detailed view on the questions.

In the fourth phase, the online questionnaire was created using Google Forms as a tool. It included a prior information and consent form regarding data processing, followed by a description of what "dealing with incomplete datasets" means so that a shared understanding of the topic is ensured. After this, the previously mentioned questionnaire was placed.

The fifth phase comprised pilot tests which were run with five test participants that did in the end not participate in the actual online survey. They merely tested the understandability of the questions formulated as well as the suitability of the question types on different devices (e.g. mobile vs. desktop display). The duration was estimated to be around ten minutes. Feedback from the pilot tests was integrated in the questionnaire.

Subsequently, the survey was made available online via link sharing and e-mail. The participants were invited via personalized e-mail or online chat apps and forums, and were asked further distribute the online survey invitation to contacts suitable for the potential participant description to obtain somewhat a snowballing effect. The online survey was active from 30 May 2022 to 07 July 2022, i.e. over a period of 39 days.

Finally, the survey answers were collected from the Google Forms online platform, and were exported as a CSV file, which was later on analyzed in R. The results of the survey are presented in the upcoming chapter 5.2 *Survey Results*.

5.2 Survey Results

The questionnaire was fully completed by 330 participants within a time period from 30 May 2022 to 07 July 2022. All survey responses were complete and could be processed for the survey results. For single-choice questions, pie charts were used for illustration as the numbers add up to 100%, while for multiple choice questions other forms of representation were chosen. With regard to their highest completed education, most participants have completed a bachelor degree (52%) as it can be seen in Figure 5.2. Further 28% completed studies at master degree level, 11% achieved high school certificate, 8% a PhD or doctoral degree, and 1% completed an apprenticeship (see 5.2). The related absolute numbers are provided in table 5.1.

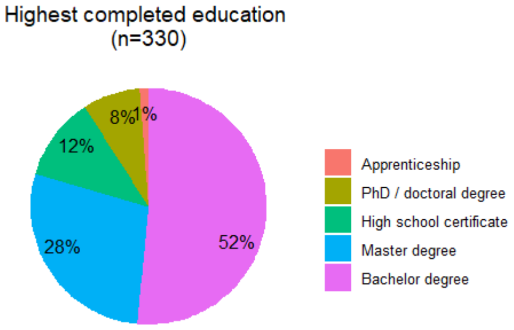


Figure 5.2: Demographics: highest completed education

Highest completed education	**Nr. of respondents**
Bachelor degree	171
Master degree	92
High school certificate	38
PhD / doctoral degree	26
Apprenticeship	3

Table 5.1: Demographics: highest completed education

The participants' main educational background appeared to mainly in computer science (46%) and statistics / data science (16%), followed by business, electrical and electrical engineering, physics, mathematics, and chemistry each in almost equal parts (see Figure 5.3 and Table 5.2).

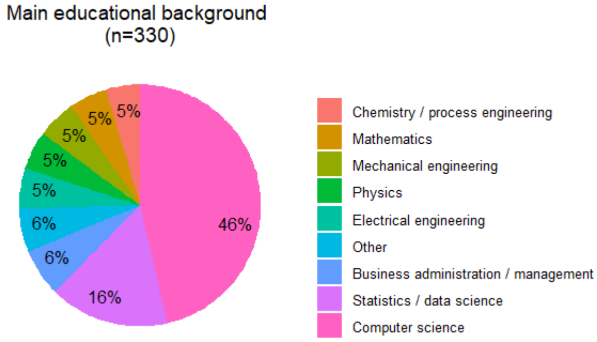


Figure 5.3: Demographics: main educational background

Main educational background	**Nr. of respondents**
Computer science	152
Statistics / data science	53
Business administration / management	20
Other	20
Electrical engineering	17
Physics	17
Mechanical engineering	17
Mathematics	17
Chemistry / process engineering	17

Table 5.2: Demographics: main educational background

The main business areas or domains, in which the participants work, are very diverse, with 21% being employed in the banking, insurance and financial services sector, 12% in industrial manufacturing, 9% in government / public sector, then 7-8% in health and medical services, education, research, energy, utilities, natural resources, retail, and consumer goods, as shown by Figure 5.4. Around 3-5% stated they work in the transportation, pharmaceuticals, life sciences, real estate, and construction sector (see Figure 5.4 and Tab 5.3).

When it comes to the most recent job role, 38% stated that they work in data analysis, data science or statistics related job, while further 20% work as software engineer, 16% in general IT or project management, and 11% in management or business administration, according to Figure 5.5. The related absolute numbers are provided in table 5.4.

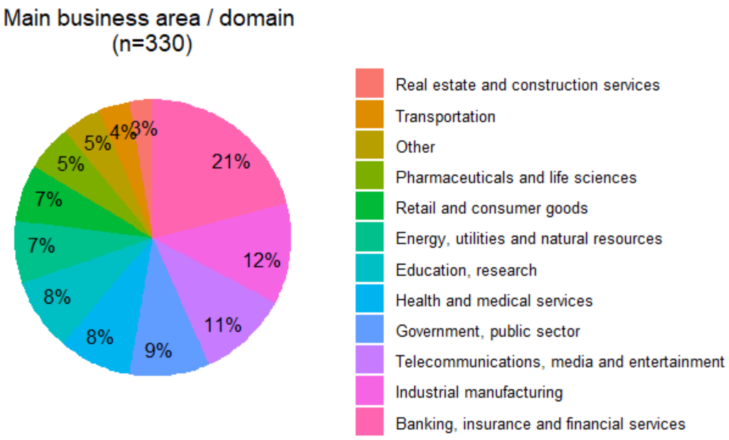


Figure 5.4: Demographics: main business area

Main business area	**Nr. of respondents**
Banking, insurance and financial services	69
Industrial manufacturing	40
Telecommunications, media and entertainm.	36
Government, public sector	30
Health and medical services	26
Education, research	26
Energy, utilities and nat. res.	23
Retail and consumer goods	23
Pharmaceuticals and life sciences	17
Other	17
Transportation	13
Real estate and construction services	10

Table 5.3: Demographics: main business area

The overall work experience ranges mainly between 3 to 5 years (i.e. among 35% of participants), and 1 to 2 years (i.e. among 19% of participants), as it can be seen from Figure 5.6. Around 30% of the participants' work experience is over 6 years, and 15% have less than one year in work experience (see Fig. 5.6 and Table 5.5).

The size of the organization also differed, but 44% of the participants are employed by large businesses counting over 500 employees overall as to Figure 5.7. Medium-sized (i.e. 50-500 employees) and small businesses (i.e. <50 employees) each have a share of around 20% (see Fig. 5.7). 12% of the participants decided not to declare their organization's size (see Fig. 5.7). The related absolute numbers are provided in table 5.6.

With respect to their age group, 50% of the survey respondents were between 26 and

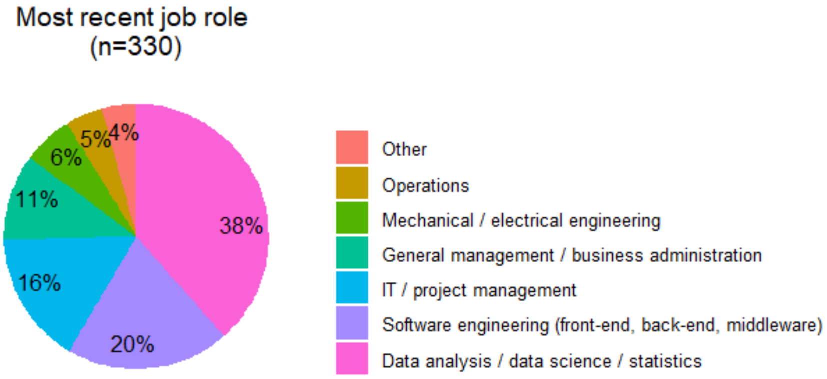


Figure 5.5: Demographics: latest job role

Most recent job role	**Nr. of respondents**
Data analysis / data science / statistics	125
Software engineering	66
IT / project management	53
General mgmt. / business admin.	36
Mechanical / electrical engineering	20
Operations	17
Other	13

Table 5.4: Demographics: most recent job role

35 years old, 35% between 18 and 25 years old, and 9% between 36 and 45 years old, according to Figure 5.8. About 4% were 46-55 years old, and the remaining participants did not declare this information (see Fig. 5.8). For absolute numbers refer to Table 5.7.

The gender distribution among the participants was 69% male, 27% female, and 4% who did not share this information, as it can be seen from Figure 5.9. The related absolute numbers are provided in Table 5.8.

The participants were also asked what programming languages they actively use at work, which led to the results in Figure 5.10, while participants were allowed to select more than one programming language in case this applied. According to Figure 5.10, 51% of them use Python, and 27% R at work (5.10). This is followed by around 17-20% of respondents who use Java, JavaScript, or C/C++/C# in practice (5.10). Further 20% did not use any programming languages at work (5.10). Less than 7% used Matlab/Octave, SAS, SQL, Scala, Visual Basic, TypeScript, or Kotlin (5.10). In the open comment field, less than 1% mentioned that they use either PHP, CUDA, Angular, Mathematica, Rust, Go,

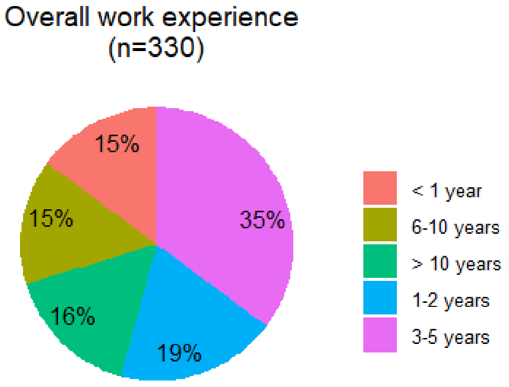


Figure 5.6: Demographics: overall work experience

Overall work experience	Nr. of respondents
3-5 years	115
1-2 years	62
> 10 years	53
6-10 years	50
< 1 year	50

Table 5.5: Demographics: overall work experience

Size of organization	Nr. of respondents
Large business (> 500 employees)	145
Small business (< 50 employees)	83
Medium-sized business (50-500 employees)	63
Don't want to share this information	39

Table 5.6: Demographics: size of organization

or Smalltalk (5.10).

In the section on handling incomplete datasets and missing values in practice, the participants were presented with cases in which missing rates of 5%, 20%, 40%, and 70% were given. They were asked to judge the situation subjectively, and select the most applicable answer option. As it is illustrated by Figure 5.11 and 5.9, in the scenario with 5% missing values in the dataset under concern, 38% said they would delete the observations (rows) and/or features (columns) that contain missing values, and then use

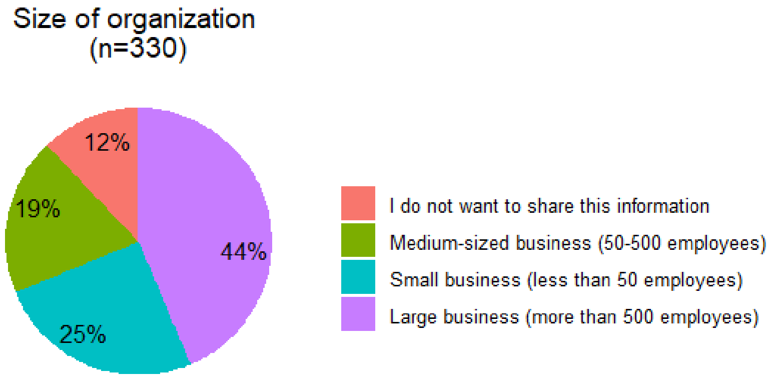


Figure 5.7: Demographics: size of organization

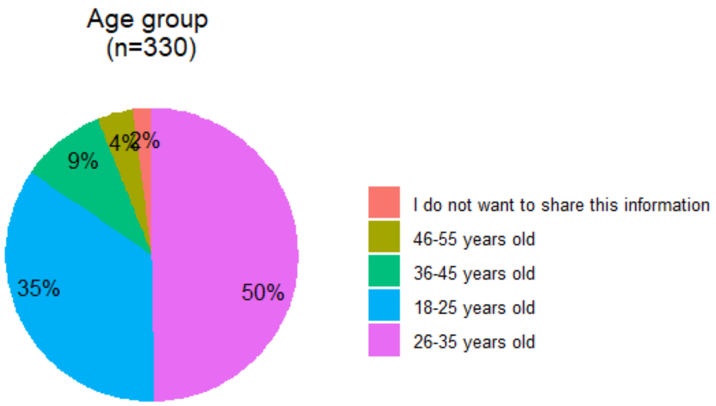


Figure 5.8: Demographics: age group

the dataset. 30% would think about a tailored imputation strategy for the setting, and would compare several methods, whereas 20% would simply replace all missing values in a column by the same value, i.e. do a replacement by constant. 8% stated that they would apply the one specific imputation technique that they always use for missing values (see Fig. 5.11).

In contrast to the 5% scenario, when the missing value percentage is increased to 20%, most participants, i.e. 44% said that they would think about a tailored imputation strategy, and compare several methods, as it is visible from Figure 5.12. 18% stated that they would not use the dataset at all due to its incompleteness. So, the point where simple missing value handling methods are applied by a large part of the respondents appears to be somewhere between the 5% and 20% missing value percentage. The related absolute numbers are provided in table 5.10.

Age group	Nr. of respondents
26-35 years old	165
18-25 years old	116
36-45 years old	30
46-55 years old	13
Don't want to share	6

Table 5.7: Demographics: age group

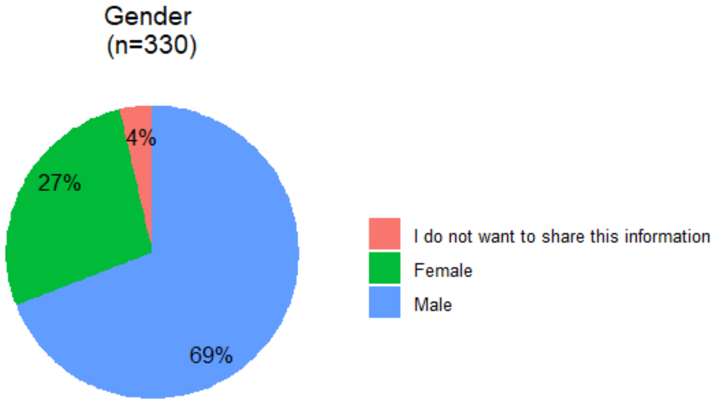


Figure 5.9: Demographics: gender

Gender	Nr. of respondents
Male	228
Female	89
Don't want to share	13

Table 5.8: Demographics: gender

In case of 40% missing values, the percentage of respondents who would not use the dataset due to its incompleteness surges to 80%, while 13% still think about a tailored imputation strategy, as shown by Figure 5.13. Simpler methods to replace the missing values would be used by only less than 5% of the participants (see Fig. 5.13). The related absolute numbers are provided in table 5.11.

In the scenario where a share of 70% of data is missing, again 80% indicated that they would not use the dataset at all because it is incomplete, and further 14% would still think about a tailored imputation strategy, as it is illustrated by Figure 5.14. The related absolute numbers are provided in table 5.12. Again, simpler methods to replace the missing values would be used by only approximately 5% of the participants (see Fig.

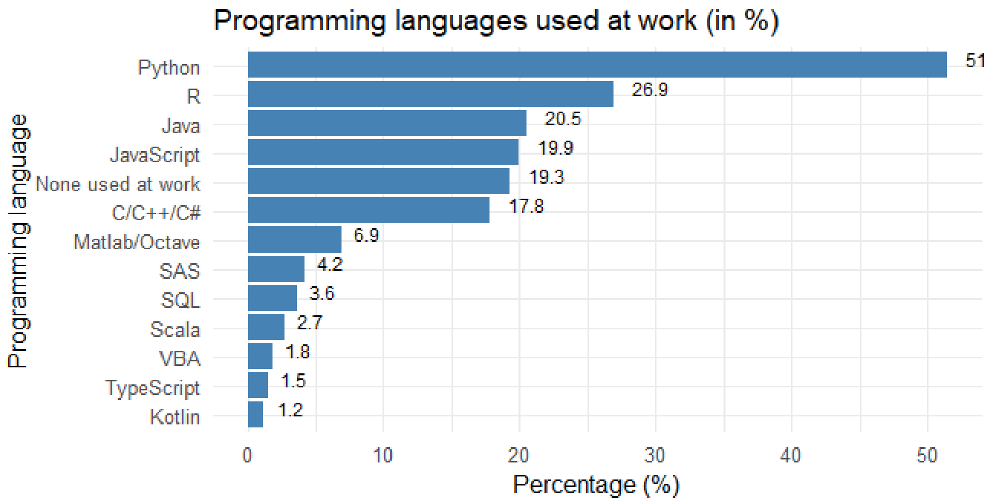


Figure 5.10: Programming languages used at work

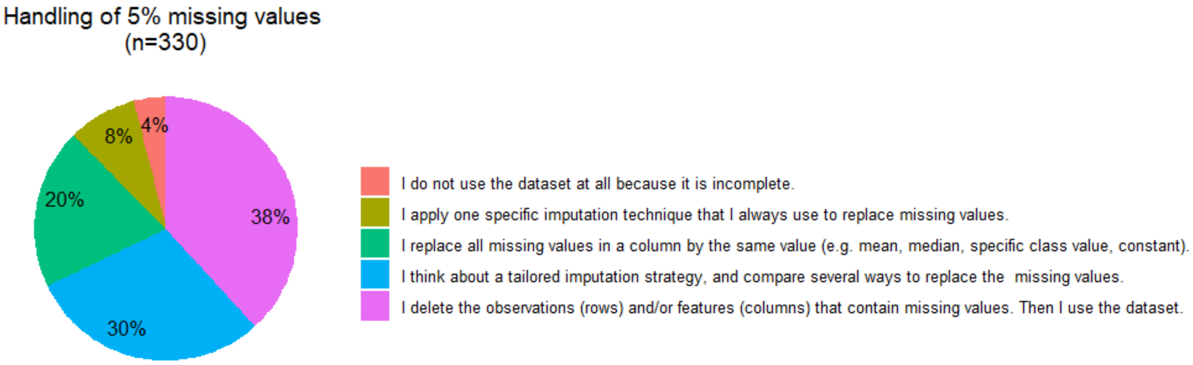


Figure 5.11: Scenario: 5% missing values

5.14). Based on the Figures 5.13 and 5.14, it can be said that those missing value rates are already too high to still think about a strategy to tackle them, according to the majority of respondents, which would largely lead to purging the whole dataset.

As for the overall approach to handling incomplete data at the organization where the respondents work, 33% indicated that every time when data is missing in a new dataset, a dedicated strategy is elaborated to tackle them (see Figure 5.15). The related absolute numbers are provided in table **??**. Figure 5.15 also shows that 24% would describe their organization's approach in a way that they would most commonly just delete the rows and / or columns affected by missing values, and continue using the dataset afterwards. 21% of the respondents stated that their company policy would rather not use incomplete datasets at all, and only consider complete ones. 14% would perform replacement by constant on the missing value entries, and 8% said that in their organization the one

Scenario: 5% missing values	Nr. of respondents
Delete the observations / features	125
Tailored imputation strategy	99
Replace all missing by same value	66
Apply one specific technique	26
Don't use the dataset	14

Table 5.9: Scenario: 5% missing values

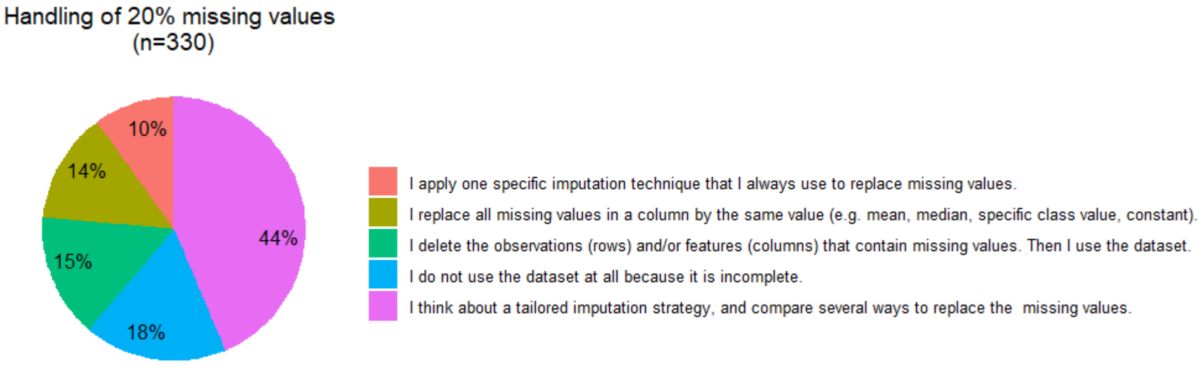


Figure 5.12: Scenario: 20% missing values

Scenario: 20% missing values	Nr. of respondents
Tailored imputation strategy	144
Don't use the dataset	58
Delete the observations / features	50
Replace all missing by same value	45
Apply one specific technique	33

Table 5.10: Scenario: 20% missing values

technique they always use would be chosen in any case (see Fig. 5.15).

30% of the respondents described their perceived importance of the topic of handling incomplete data at their organization as rather important, as shown in Figure 5.16. The related absolute numbers are provided in table 5.14. Around 24% said that they perceive it as indifferent, and 19% stated that for their organization the topic appears to be highly important. According to around 26% of the participants, the topic is rather not important or very unimportant at their organization.

The next section dealt with the knowledge of methods to handle incomplete datasets in practice, in relation to which the participants indicated their own experience, which is

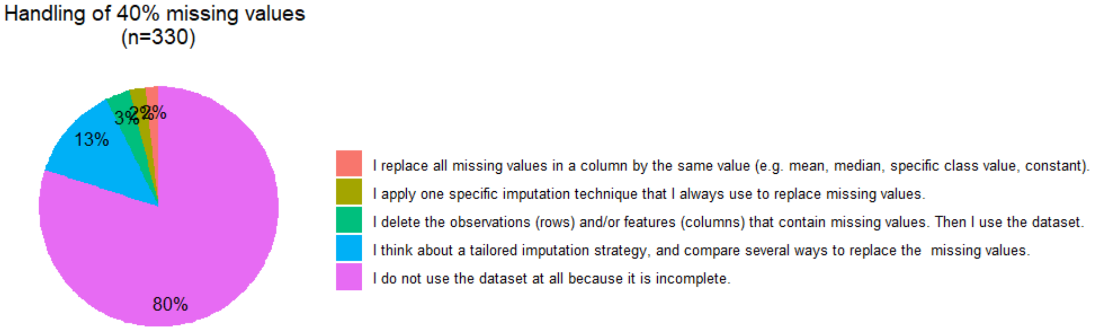


Figure 5.13: Scenario: 40% missing values

Scenario: 40% missing values	**Nr. of respondents**
Don't use the dataset	264
Tailored imputation strategy	43
Delete the observations / features	10
Apply one specific technique	7
Replace all missing by same value	6

Table 5.11: Scenario: 40% missing values

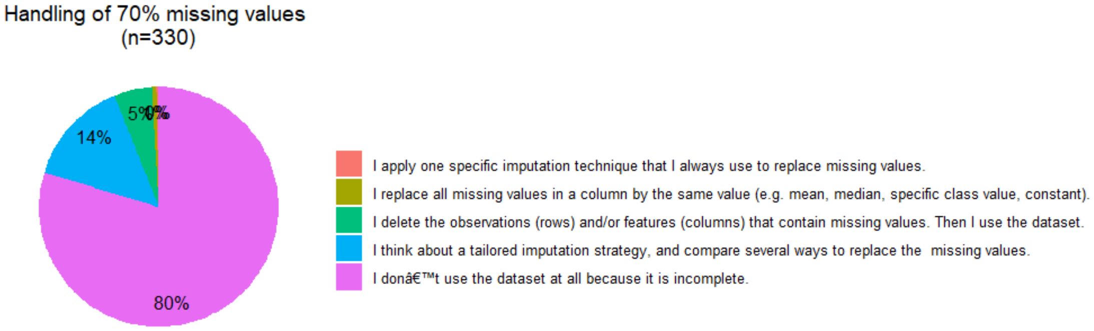


Figure 5.14: Scenario: 70% missing values

shown in Figure 5.17 and 5.18 (using all participant's answers). The Figures 5.17 and 5.18 illustrate that the most used methods in practice among all participants are mean / mode / median replacement and replacement by constant, which were both used by almost 50-60% of the respondents, followed by regression with around 25-30%, and kNN actively used by around 20%. Approximately 10% of the participant had used CART, random forest, SVM, LOCF, multiple imputation, and PCA techniques in context with missing value handling (see Fig. 5.17 and 5.18). The in practice least used methods for dealing with missing values from the ones given in the survey are hot-deck, EM, ANN,

Scenario: 70% missing values	Nr. of respondents
Don't use the dataset	264
Tailored imputation strategy	46
Delete the observations / features	17
Replace all missing by same value	2
Apply one specific technique	1

Table 5.12: Scenario: 70% missing values

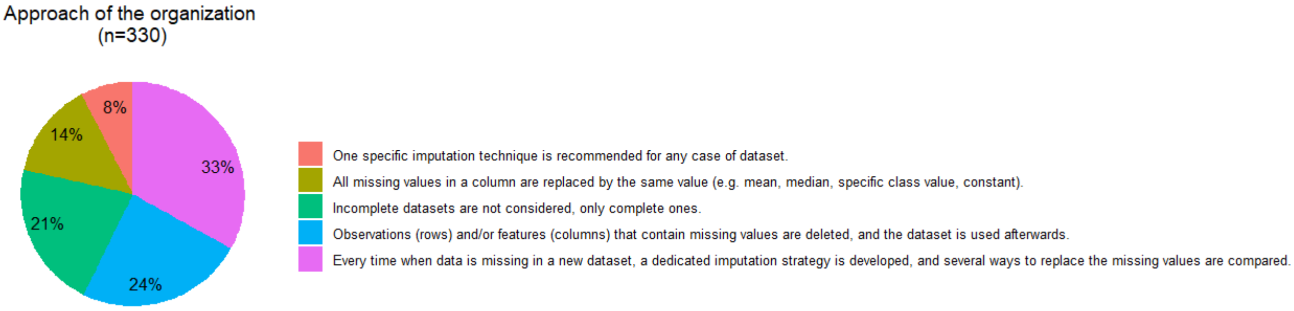


Figure 5.15: Approach of the organization

Approach of the organization	Nr. of respondents
Dedicated imputation strategy	109
Delete observations / features	79
Incomplete datasets are not used	70
All missing values replaced by constant	46
One specific imputation technique always used	26

Table 5.13: Approach of the organization

association rules, Naive Bayes, and XGBoost. It also emerged that most participants of the survey know mean / mode / median replacement, replacement by constant, and kNN, but for instance EM, hot-deck, association rules, XGBoost are not generally known much (see Fig. 5.17 and 5.18). Still, to over 20% of the respondents, the methods CART, kNN, random forest, SVM, ANN, Naive Bayes, PCA, and regression are known from another context where they used it, but they have not used it in practice for missing value handling (5.17, 5.18). For each method, around 10% of the participants did some tutorials, but had not actively applied it in practice (5.17, 5.18).

When the view is narrowed from all participants to merely the ones whose job role is placed in the area of data analysis / data science / statistics, the above findings are confirmed, but the share of respondents not knowing the methods is lower, and at the

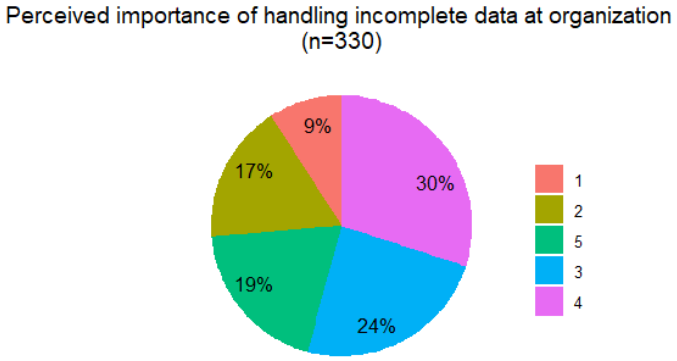


Figure 5.16: Perceived importance of topic at organization

Perceived importance	**Nr. of respondents**
4	99
3	80
5	63
2	57
1	31

Table 5.14: Perceived importance

same time, the share of respondents who actively use the methods is higher, as it is illustrated by Figures 5.19 and 5.20. From the data analyst / data scientist / statistician perspective, mean / mode / median replacement, replacement by constant, kNN, and regression are the best known and most actively used techniques to handle missing values (5.19, 5.20).

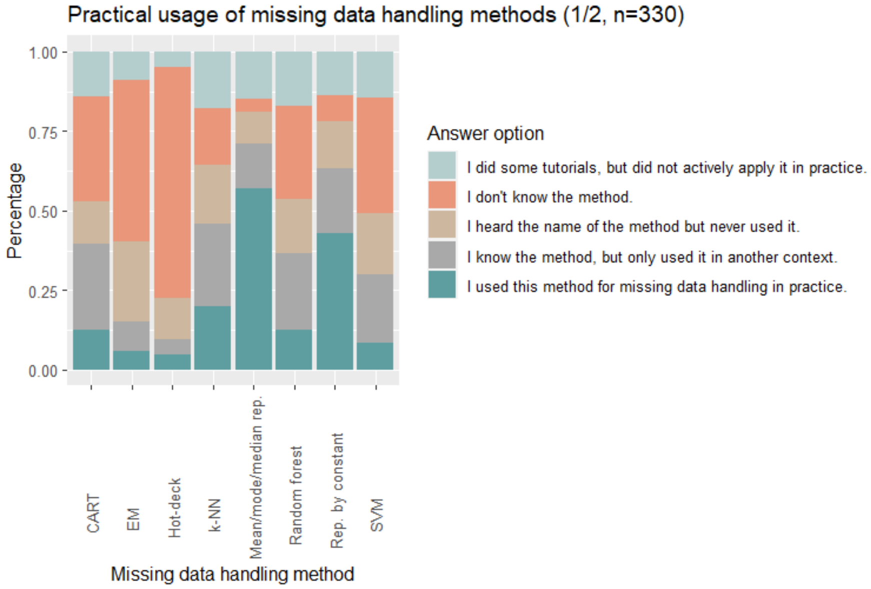


Figure 5.17: Practical usage of methods (1/2) (n=330)

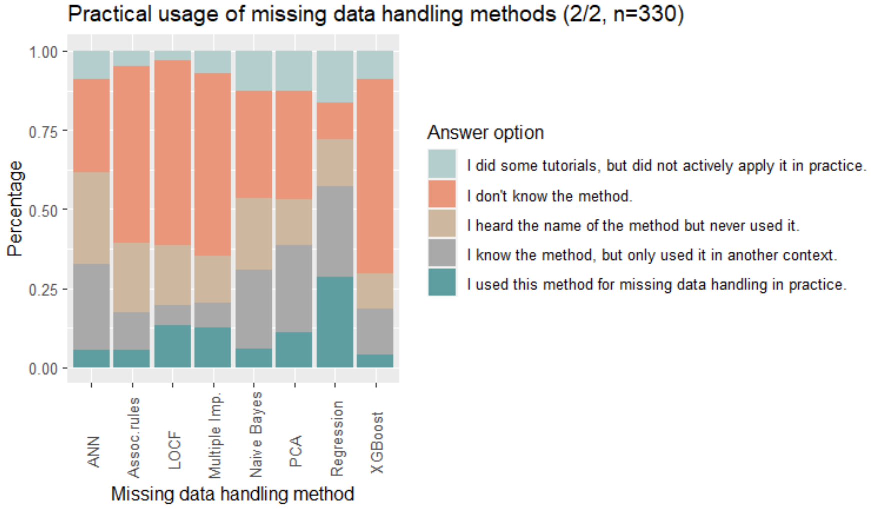


Figure 5.18: Practical usage of methods (2/2) (n=330)

In line with the survey, the participants were also asked to state their preference for a specific missing value handling method, in case they do have one. According to Figure 5.21 (see Table 5.15 for absolute values), 22% stated that mean / median / mode replacement is their preferred technique, which is the largest group of respondents. In contrast, 21% said that they did generally not tackle incomplete datasets in their job role. Further around 8% preferred kNN or linear / logistic regression methods, and 5% replacement by constant. ANN (i.e. neural networks), CART, Naive Bayes, association rules, EM, PCA,

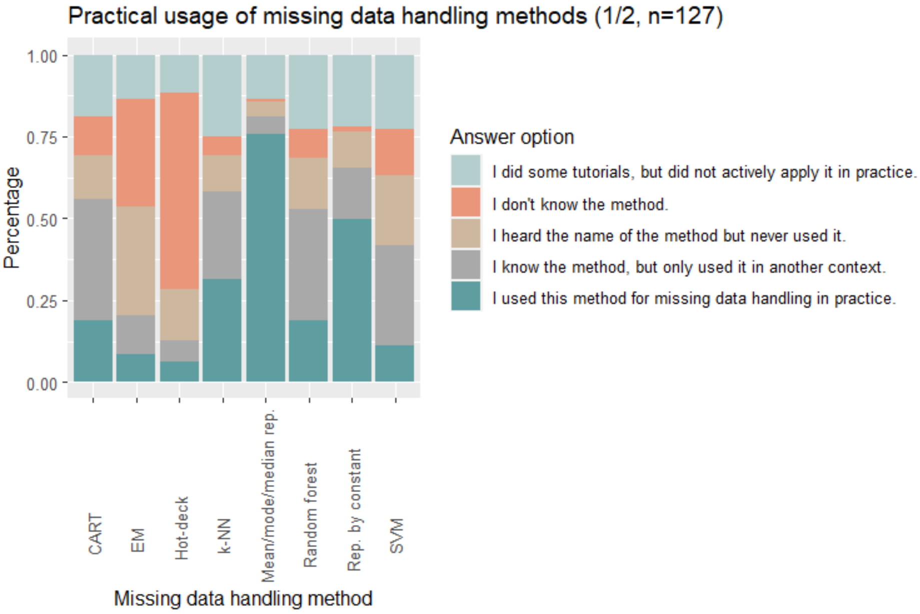


Figure 5.19: Practical usage of methods (1/2) (n=127)

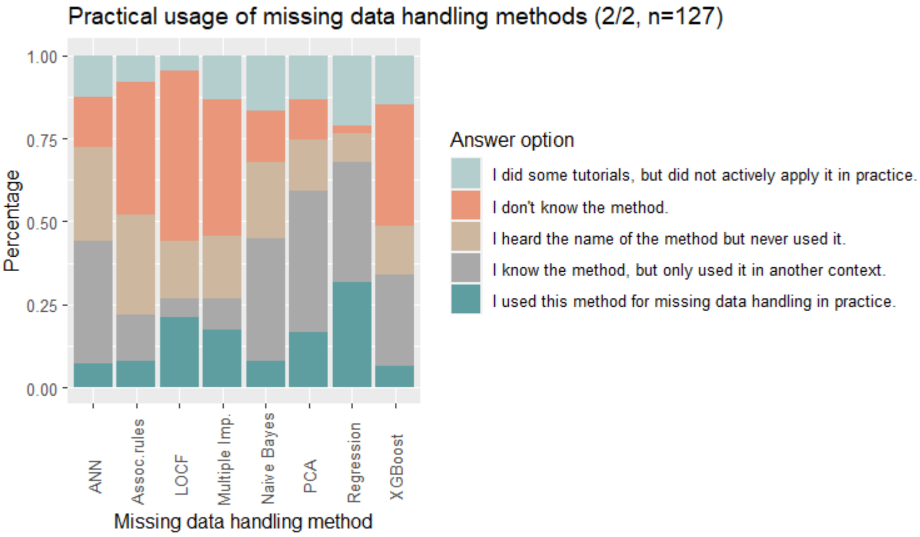


Figure 5.20: Practical usage of methods (2/2) (n=127)

XGBoost, hot-deck, and LOCF were mentioned least as preferred methods, according to Figure 5.21.

A series of missing data handling software libraries and packages from the languages Python and R were presented to the survey participants in order for them to answer whether they did not know the package / library, had actively used it in practice, or they

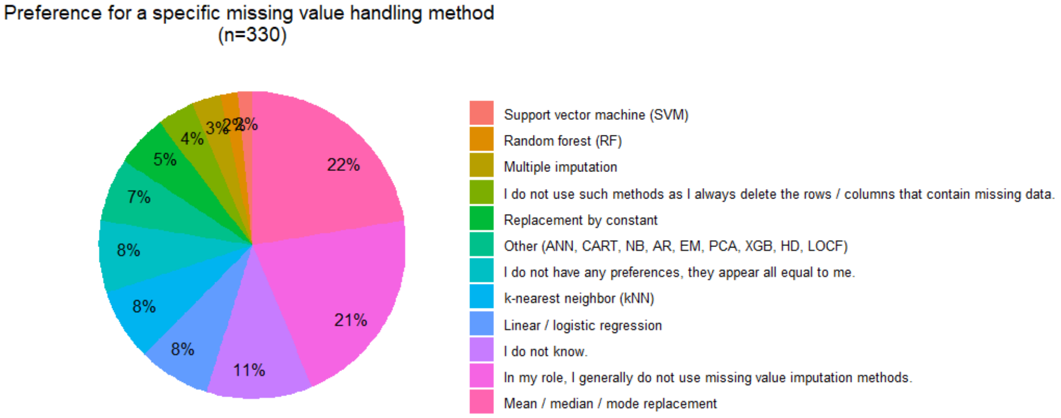


Figure 5.21: Preference for method

Preference for method	**Nr. of respondents**
Mean / median / mode replacement	72
No contact with imputation in job	69
I do not know	36
Linear / logistic regression	26
k-nearest neighbor (kNN)	25
No preferences	25
Other (ANN, CART, NB, AR, EM, PCA, XGB, HD, LOCF)	23
Replacement by constant	17
Delete rows / columns only	13
Multiple imputation	10
Random forest (RF)	7
Support vector machine (SVM)	7

Table 5.15: Preference for method

heard about it but did not use it in practice. The results were gathered once across all participants, then among people who worked in data science, data analysis, or statistics only. Figure 5.22, which considers all respondents, shows that around 70% (or more) of them had heard about the packages and libraries, but did not actively use it in their job role. Around 20% each had actively used *Datawig* (Python), *IterativeImputer* (Python), *KNNImputer* (Python) and *XGBRegressor* (R), however, at the same time the largest number of respondents also mentioned those libraries as unknown to them (see Fig. 5.22). For instance *Amelia* (R), *Hmisc* (R), *mi* (R), *missForest* (R), *missMDA* (R), *naniar* (R), and *VIM* (R) exhibited a very low share of respondents not knowing them, but likewise a large share of people who had not actively used them at work (see Fig. 5.22. When the scope of the data is narrowed down to only data scientists, data analysts, and statisticians, it can be seen that the share of respondents who actively used the respective libraries and packages is higher than for the whole participant base, according to Figure 5.23. Likewise, from the data science, data analysis, and statistics perspective, the packages and libraries that were most actively used in practice were *Datawig* (Python), *Hmisc* (R), *IterativeImputer* (Python), *KNNImputer* (Python), and *XGBRegressor* (Python), plus a higher number of participants who had practical experience with *MICE* (R) (see Fig. 5.23. Even among the data scientists, data analysts, and statisticians, the approximately 50%, and for some methods even a higher share of respondents, had heard about the library or package, but had not actively used it at work (see Fig. 5.23).

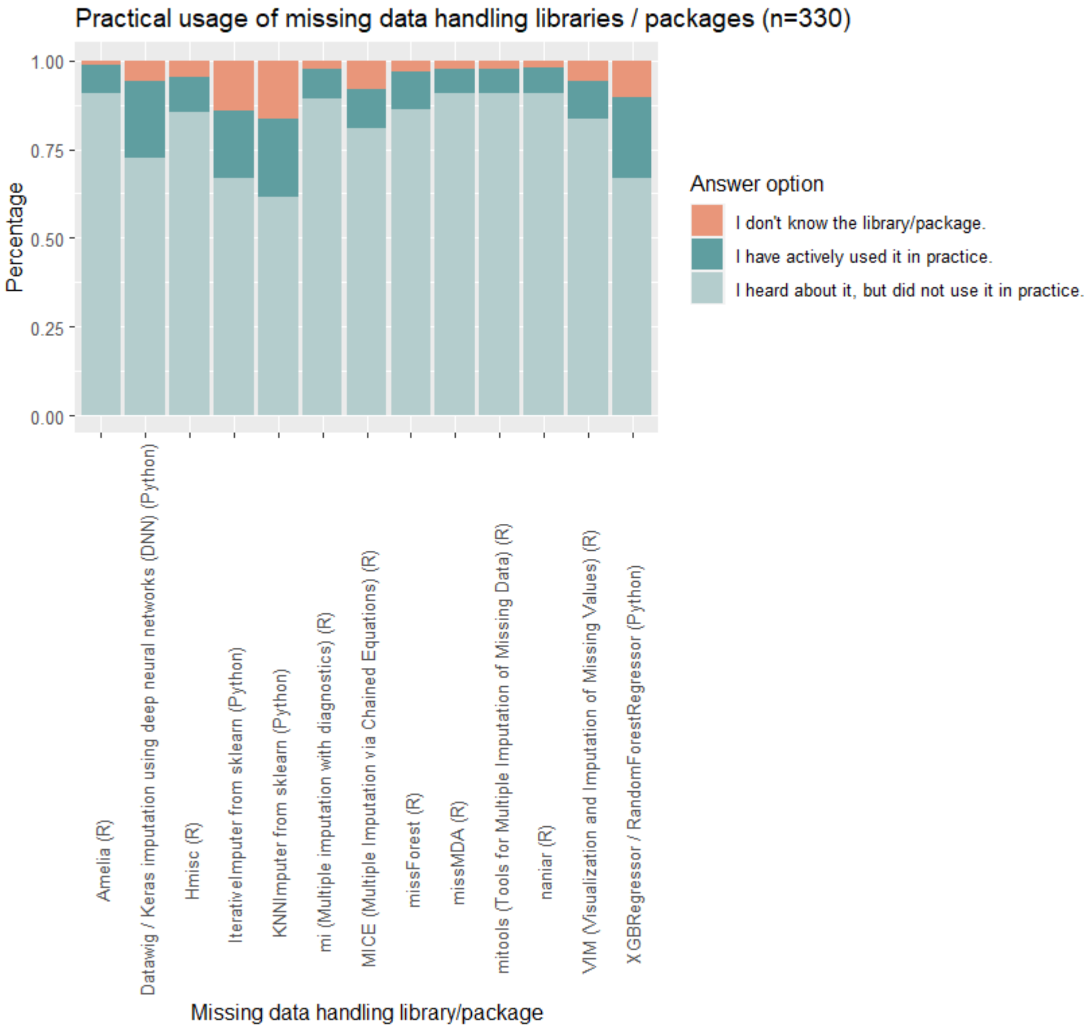


Figure 5.22: Practical usage of libraries / packages (n=330)

In the last section of the survey, the choice of methods and learning about methods to handle missing data were investigated. The participants rated the respective influencing factors of the choice of missing value treatment methods on a scale from 1 (i.e. low

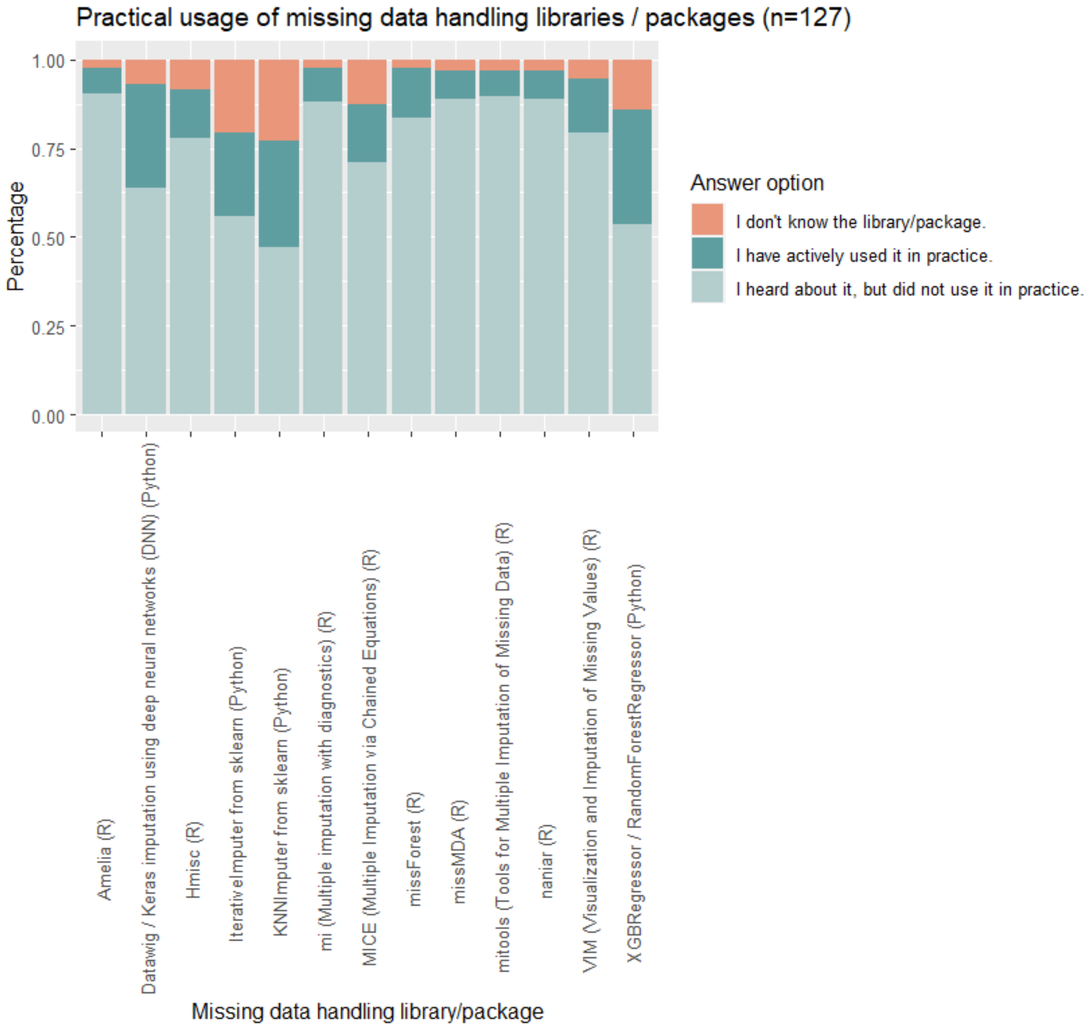


Figure 5.23: Practical usage of libraries / packages (n=127)

influence) to 5 (i.e. high influence). As shown by Figure 5.24, the characteristic that a method is “fast and easy to implement“ was rated highest with an average score of 3.9. So, out of the given characteristics it was the most influencing factor for the choice of a missing value treatment method. The characteristic that the method is *recommended at workplace* was rated as 3.68 on average. *Useful tutorials being available* obtained a mean rating of 3.42, and *method already used during education* 3.35. The lowest average score was assigned to *reading papers about the method* with 3.22 on the scale. Overall, it can be see from Figure 5.24 that the influence of the given factors is neither particularly low nor particularly high, but rather between 3 and 4 on a scale from 1 to 5.

When asked about how they perceived the extent of practical guidance experienced during their education with regard to handling missing data, 37.9% of the respondents

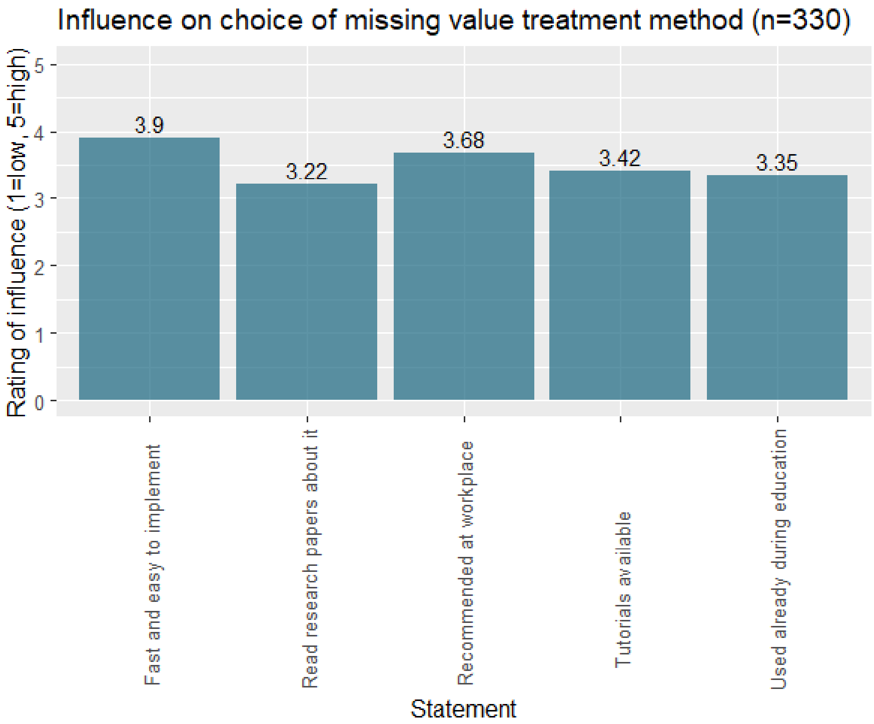


Figure 5.24: Factors influencing the choice of method

mentioned that it had been mentioned as such but that they actually applied only one or two simpler methods, and did not go into details. 22.7% of the respondents indicated that they had not obtained any practical guidance on this topic during their education at all, whereas 16.7% mentioned that they covered the topic in-depth and that they also gained practical insights by comparing several approaches in a more detailed way (see Figure **??**). 13.3% said that the topic was covered, but nothing was implemented in a practical setting, and 9.4% said that they only remembered that there was a theoretical session on it (see Figure **??**). To sum up, over 80% indicated that they did not cover the topic in-depth in practical sessions during their education (see Figure **??**).

Subsequently, the participants were asked to indicate how important they see practical guidance during education or how much they would recommend it, the results of which are summarized by Figure 5.25 and Table 5.16. . 34% respectively 33% each think that practical guidance on handling missing data would be very helpful resp. rather helpful, while only 9% respectively 2% consider it rather not helpful or not helpful at all (see Figure 5.25).

In line with the final comments section, some participants added their thoughts in the form of open text. A large number of the comments pointed out that missing values are an issue in practice that should ideally obtain more attention, but is in reality still not taken as seriously by management or due to time constraints at work in many cases.

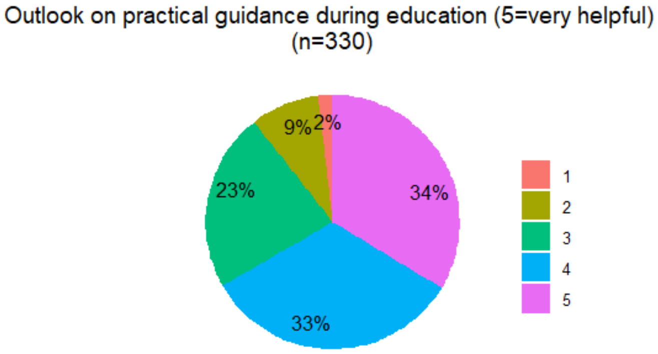


Figure 5.25: Outlook on practical guidance

Outlook guidance	**Nr. of respondents**
5	112
4	108
3	75
2	29
1	6

Table 5.16: Outlook guidance

Careful study of research papers and available options would be necessary to examine the difference in performance, however, such a scenario appears to be rather an ideal case, and not directly transferable to everyday life where time and resource constraints occur. The respondents also mentioned that it would be recommendable to not even let missing values occur, i.e. tackle the issue earlier at the root already.

CHAPTER 6

Statistical Experiments

The chapter comprises firstly the Experiment Framework Definition 6.1 section, in which the general experimental design is outlined that is relevant across the two case studies. The section Open-source Case Study 6.2 covers the data description, exploratory analysis, experiment implementation, and summary of the results. The section Industry Case Study 6.3 comprises the same structure of sub-sections in order for the results to be in a comparable format to the open-source case study. Finally, the section Comparison of Results 6.4 evaluates the results of the two case studies side by side, so that the best and worst performing missing value handling methods based on the underlying datasets can emerge more clearly.

6.1 Experiment Framework Definition

The chapter *Statistical Experiments* 6 comprises an open-source case study and an industry-specific case study, for which the common case study and experiment structure is outlined in this section. The Figure 6.1 depicts the **case study structure**, which starts with introductory remarks on the chosen dataset and its verbal description. In the next phase, the dataset is presented in an exploratory way using descriptive statistics and visualizations. Additional information about the explanatory variables (to put it differently features or predictors) and the target variable (i.e. the dependent variable) including their types is provided.

In the third phase, the statistical experiments are carried out, which is illustrated in greater detail by Figure 6.2 that comprises the necessary experiment steps. The final phase wraps up the case study by providing a summary of the outcomes and visualizing the results amongst other things in terms of performance metrics measured.

When it comes to the **statistical experiment set-up**, as already mentioned, Figure 6.2 indicates the details. As such, the experiment set-up is based upon the structure proposed

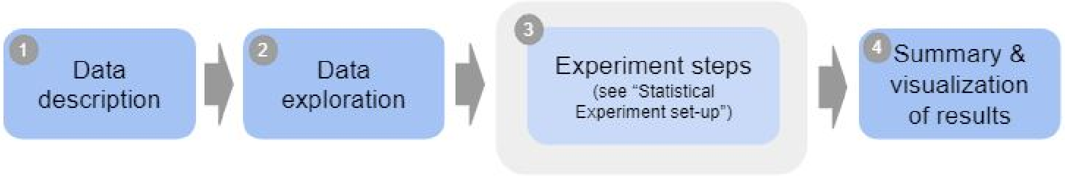

1 Data description
2 Data exploration
3 Experiment steps (see "Statistical Experiment set-up")
4 Summary & visualization of results

Figure 6.1: Structure of case study

by Farhangfar (2008) who conducted similar statistical experiments, but adapted in a way to fit the goals of the experiments used in this research project [21]. As more extensive methods for handling missing data are available in R, this programming language is chosen to implement the statistical experiments for the two case studies.

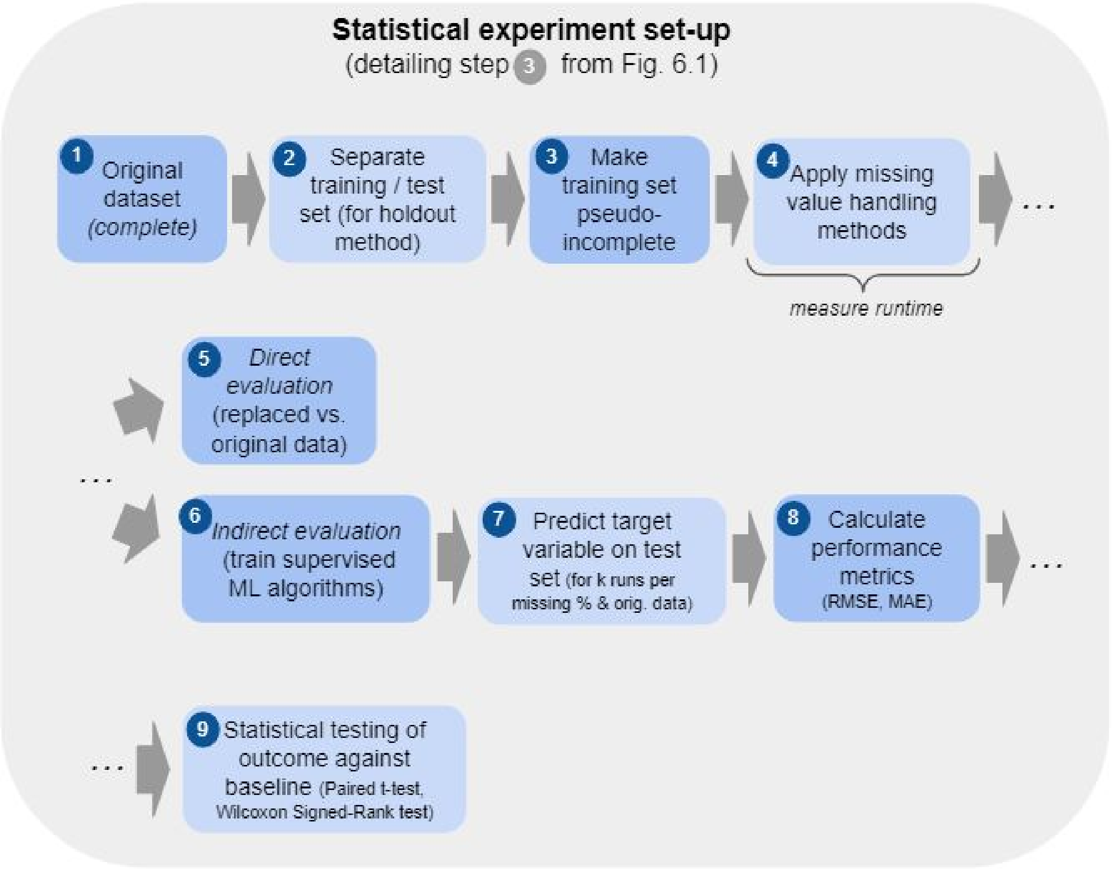

Statistical experiment set-up
(detailing step 3 from Fig. 6.1)
1 Original dataset (complete)
2 Separate training / test set (for holdout method)
3 Make training set pseudo-incomplete
4 Apply missing value handling methods
measure runtime
...
...
5 Direct evaluation (replaced vs. original data)
6 Indirect evaluation (train supervised ML algorithms)
7 Predict target variable on test set (for k runs per missing % & orig. data)
8 Calculate performance metrics (RMSE, MAE)
...
...
9 Statistical testing of outcome against baseline (Paired t-test, Wilcoxon Signed-Rank test)

Figure 6.2: Overview of statistical experiments (adapted from Farhangfar (2008)) [21]

The statistical experiment starts from the original dataset, which is complete, i.e. it does not contain any missing values. In the next step, the data is split according to the chosen split-ratio between train and test set, as a separation is necessary for the application for the holdout method. Furthermore, the target variable is separated from the features. The holdout method requires that there is one training set for fitting the machine learning model that is not overlapping with the test set, with the latter being used to make predictions by the means of the previously fitted machine learning model.

The training set is subsequently made pseudo-incomplete meaning that a specific percentage of data are randomly chose to be replaced by `NA` values so that they appear to be missing. In doing so, different runs are carried out to get different versions of the training dataset having a certain missing data percentage.

In the following step, several methods (chosen based on the outcomes of the literature review (4) as well as the online survey (5)) are applied to the datasets containing missing values, and the runtime to make them complete again is measured as a performance criterion.

As soon as the dataset is made complete again, the direct evaluation can be carried out in order to compare the original dataset to the dataset in which the missing values were imputed on a column by column basis.

For the indirect evaluation, which compares the performance of supervised machine learning algorithms on the imputed dataset to the performance on the original dataset and / or to different machine learning algorithms, several further steps are to be taken. The chosen supervised machine learning algorithms are trained based on the pseudo-complete datasets, i.e. the models are fitted. Then the model is used to generate predictions on the test dataset, which was earlier set aside. In this case, a numeric target variable is predicted. Therefore, regression methods are used among the supervised machine learning techniques. Once the predictions are completed, the performance can be measured in terms of RMSE and MAE, which evaluate the differences between the predictions and the original values of the numeric target variable.

In the final step, the performance outcomes are statistically tested for four missing value percentages using the different randomly generated runs of pseudo-complete training datasets. The performance results from the different runs are tested against baseline, i.e. against a the results of a specific missing data handling method (e.g. mean imputation) and / or of the original dataset. With regard to the statistical tests, the case study deals with repeated measures data, which means that paired t-test (as a parametric test[1]), and Wilcoxon Signed-Rank test (as its non-parametric[2] equivalent) can be carried out to compare the performance scores.

[1] Note: The parametric test makes certain assumptions on the distribution such as normality.

[2] Note: The non-parametric tests does not make any assumptions on the distribution of the data.

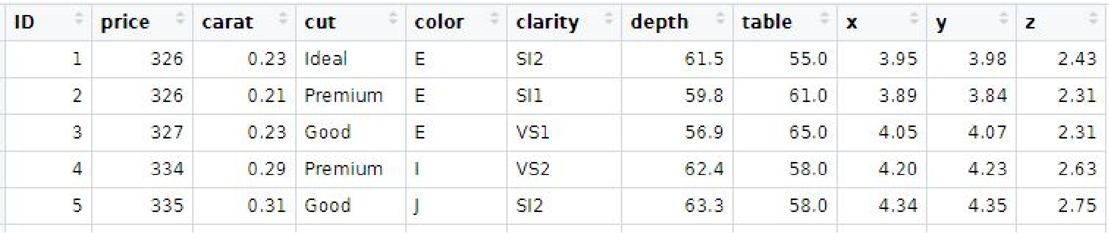
6.2 Open-source Case Study

The section Open-source Case Study 6.2 covers the data description, exploratory analysis, experiment implementation, and summary of the results. The general case study structure is illustrated in Fig. 6.1.

6.2.1 Dataset Description

For the open-source case study a dataset from the raw materials industry is analyzed, more precisely from the diamond industry. The data[3] was extracted from the pool of Kaggle and Google research datasets[4]. For this project, a (randomly selected) subset of 10% of the data is used due to the computational complexity required for the subsequently applied missing handling methods.

The *diamonds* dataset contains 5394 observations (i.e. rows) and 11 features (i.e. columns). It is structured in a way that the unique ID heads the columns, and the second column is the target variable *price*, which describes the price (in United States Dollar (USD)) per observation. The Figure 6.3 presents a short excerpt of the *diamonds* dataset, which illustrates the rows and columns layout.



ID	price	carat	cut	color	clarity	depth	table	x	y	z
1	326	0.23	Ideal	E	SI2	61.5	55.0	3.95	3.98	2.43
2	326	0.21	Premium	E	SI1	59.8	61.0	3.89	3.84	2.31
3	327	0.23	Good	E	VS1	56.9	65.0	4.05	4.07	2.31
4	334	0.29	Premium	I	VS2	62.4	58.0	4.20	4.23	2.63
5	335	0.31	Good	J	SI2	63.3	58.0	4.34	4.35	2.75

Figure 6.3: Excerpt of *diamonds* dataset

The ID column contains the unique identifier for each observation and is not part of the statistical analysis, whereas the latter columns are target variable (which only refers to the *price*) or features, which are *carat*, *cut*, *color*, *clarity*, *depth*, *table*, x, y, and z. Three of the variables (i.e. *cut*, *color*, and *clarity*) are nominal, which means categorical variables. The remaining ones are of ratio type, which means numeric variables. The dataset as such does not contain any missing values in its original form.

The table 6.1 below summarizes the type information regarding the columns, provides a short description of the respective column's meaning, and its range or categories. The column *price* is annotated as target variable, which is the numeric variable that should be predicted in the indirect evaluation track using regression methods.

[3] Note: The data source references refers to all places in the chapter mentioning the *diamonds* dataset.
[4] Data Source: `www.kaggle.com/datasets/nancyalaswad90/diamonds-prices`

Summary of dataset characteristics			
Variable name	**Type**	**Description**	**Range**
`ID`	Ratio	Unique identifier	1 - 53943
`price` (target variable)	Ratio	Price of diamond	336 - 18757
`carat`	Ratio	Carat of diamond	0.23 - 5.01
`cut`	Nominal	Type of cut	*Ideal*, *Premium*, *Very Good*, *Good*, *Fair*
`color`	Nominal	Color of diamond	*E*, *I*, *J*, *H*, *F*, *G*, *D*
`clarity`	Nominal	Type of clarity	*SI1*, *SI2*, *VS1*, *VS2*, *VVS1*, *VVS2*, *I1*, *IF*
`depth`	Ratio	Depth of diamond	51.00 - 71.60
`table`	Ratio	Table information	43.00 - 70.00
`x`	Ratio	Measure x in mm	0.00 - 10.74
`y`	Ratio	Measure y in mm	0.00 - 10.54
`z`	Ratio	Measure z in mm	0.00 - 6.98

Table 6.1: Summary of *diamonds* dataset characteristics

6.2.2 Exploratory Analysis

In line with the exploratory analysis, descriptive statistics are visualized in order to introduce the content of the *diamonds* dataset prior to the analysis. With regard to the categorical variable *cut*, the most frequently occurring type is of the label *Ideal*, followed then by *Premium* and *Very Good*, *Good*, and only very few of type *Fair*, as shown in Figure 6.4. The histogram of the second categorical variable is presented by Figure 6.5 refers to *color*, which is again not evenly distributed, but has most occurrences in the types *G*, *E*, *F*, and *H*, with a lower number of occurrences in type *D*, *I*, and *J*. Figure 6.6 shows the histogram of clarity types, which comprises mostly the labels *SI1*, *SI2*, *VS1*, and *VS2*, and a lower number of the labels *I1*, *IF*, *VVS1*, and *VVS2*.

Furthermore, the Figure 6.7 represents the distribution of the numeric variables contained in the dataset (including the target variable *price*). From the figure, it can be seen that the distribution of *depth* and *table* are rather concentrated around a smaller range of values, while the distribution of x, y, y, *carat*, and *price* respectively is less condensed and spread across a larger range. Between the numeric variables the correlation can be computed to see how each pair of them is related. As it can be seen from Figure 6.8, the positive correlations between *carat* and x, y and x, y and *carat*, z and x, z and *carat*, as well as z and y are among the highest in the dataset. The variable *price* is also rather positively correlated with x, y, *carat*, and z, according to Figure 6.8. The variable *table* appears to be only slightly positively or almost uncorrelated with other variables in the dataset. The variable *depth* is slightly negatively correlated with *table*, and very slightly positively correlated with z, but uncorrelated with the remaining variables.

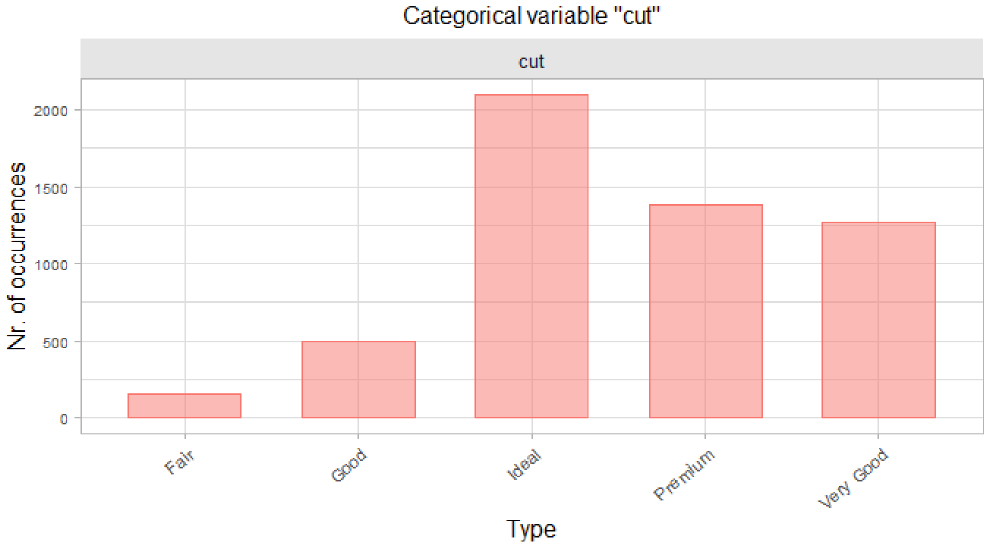


Figure 6.4: Histogram of the categorical variable *cut*

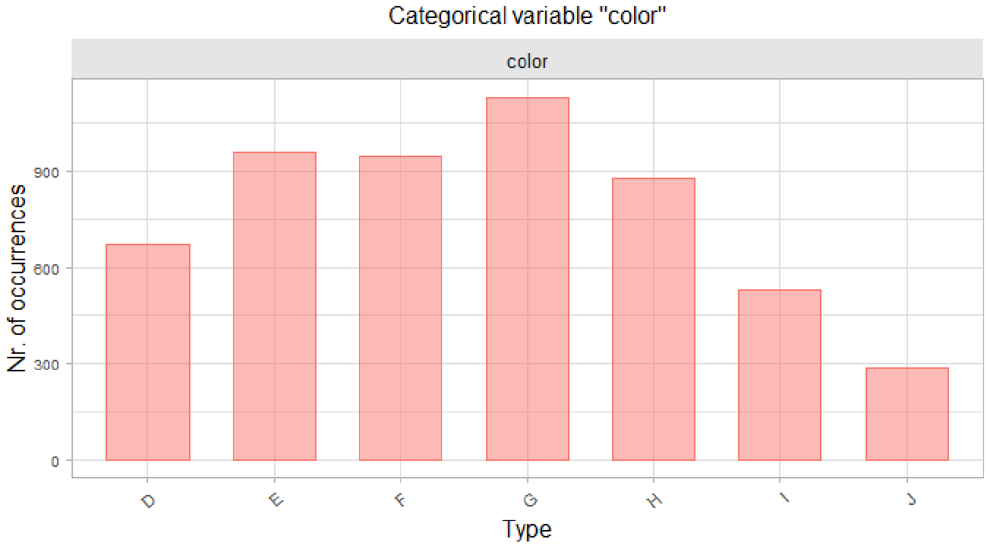


Figure 6.5: Histogram of the categorical variable *color*

6.2.3 Experiment Implementation

The experiment was implemented according to the set-up shown in Figure 6.2. The **original dataset** (see step 1 in Fig. 6.2) was used as a basis, and was read in a way that strings were treated as factors. The original dataset was prepared (as a `data.frame` data type) in way that facilitated the later pre-processing and analysis steps: *ID* and *price* (i.e. the target variable) were in the first part, followed by the categorical variables,

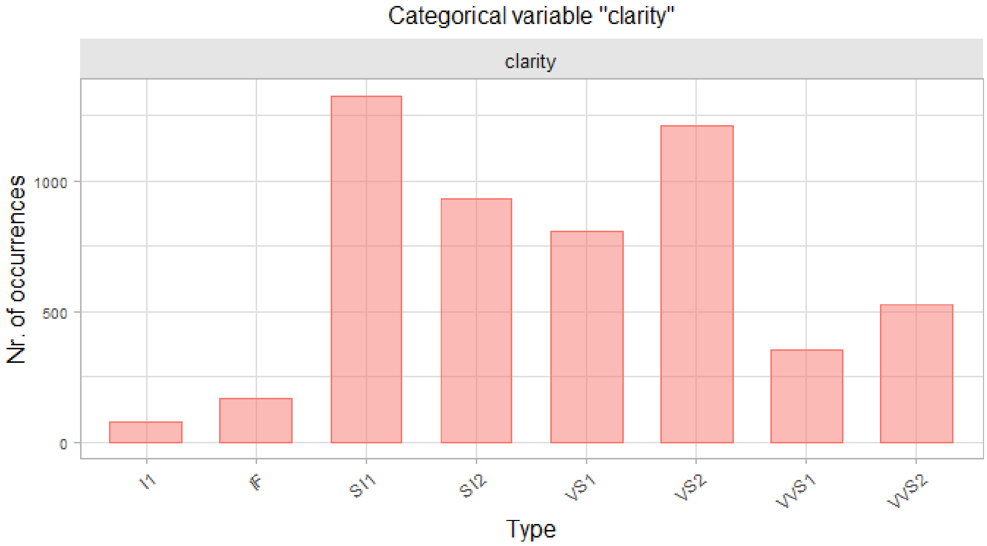


Figure 6.6: Histogram of the categorical variable *clarity*

and finally the numeric variables.

The data was subsequently **split into training and test data** (see step 2 in Fig. 6.2) along 20 random splits, with the randomness being captured by seed values to make the experiment reproducible. As a split ratio 80-20 was chosen for the train-test percentages, the 20 random split combinations were prepared in order to use them for the holdout method. According to this, on every training dataset a model is fitted, and the corresponding test data is the basis for the predictions.

The split, but still complete data had to undergo a process of **creating a pseudo-incomplete dataset** (see step 3 in Fig. 6.2). For this reason, the features were separated from the target variable and the ID column, and random row-column index combinations were generated in order to obtain the entries to delete from the training data. Different missing data percentages were considered, namely 5%, 10%, 20%, and 40%. The missing percentage of 70% which was also mentioned in the prior online survey was dropped given the fact that most theoretical works as well as participants stated that they never considered a dataset with such a high missing data percentage, and if they did, the data quality was too low for further processing[36].

In the next phase, the missing values had to be dealt with. Thus, **missing value handling methods** were applied (see step 4 in Fig. 6.2) to make the pseudo-incomplete complete again, and at the same time the **execution time was measured** as a performance metric for all 20 runs (per missing entries percentage and handling method). For the datasets, the missingness pattern MCAR was assumed as the datasets were randomly made incomplete. The missing handling methods were chosen based on the results from the literature review and the online survey:

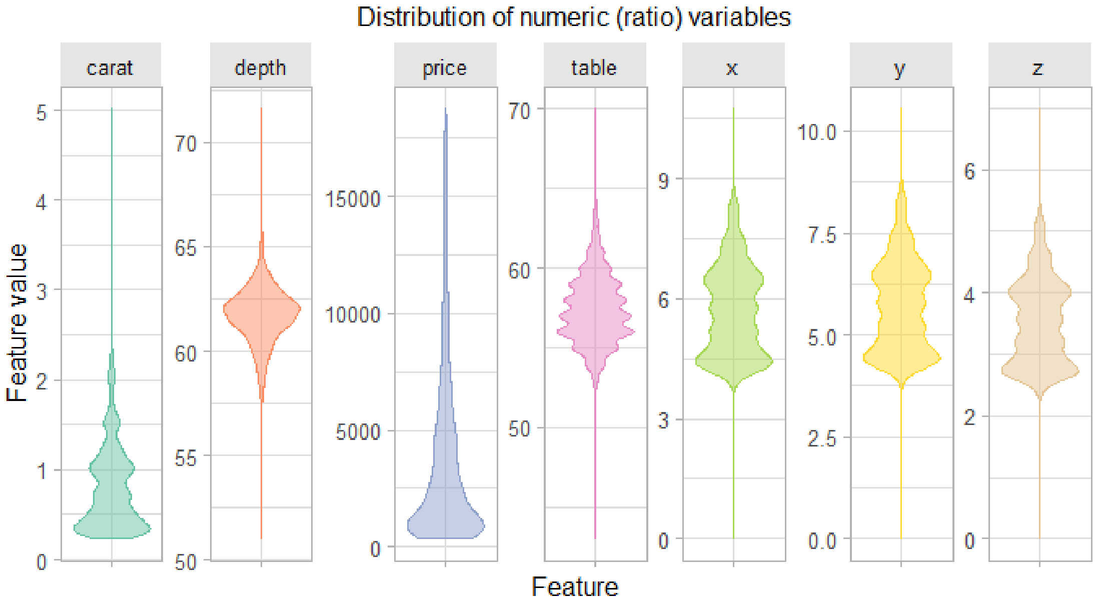


Figure 6.7: Distribution of the numeric variables

- Mean imputation
- Median imputation
- kNN imputation
- CART imputation
- Random Forest (RF) imputation
- Regression imputation
- XGBoost imputation

Theory suggested as machine-learning based methods in this specific order kNN, RF, SVM, CART / DT (trees), Naive Bayes, LDA[5], XGBoost, and neural networks (e.g. variational autoencoder, NNRW)[36]. In contrast to this, the ranking of suggestions appeared to be different as to the online survey, in which the participants mentioned mean / median / mode imputation, kNN, CART / DT (trees), DT, Multiple Imputation, linear regression, and LOCF in this order. Artificial neural networks, association rules, XGBoost, Naive Bayes, SVM, EM, Hot-Deck, and PCA were used rarely in practice, according to the online survey results. Among the in practice most frequently used libraries were `mi`, `Hmisc`, `mice`, `missForest`, and `VIM`, whereas `Amelia`, `missMDA`,

[5] Note: for categorical data only.

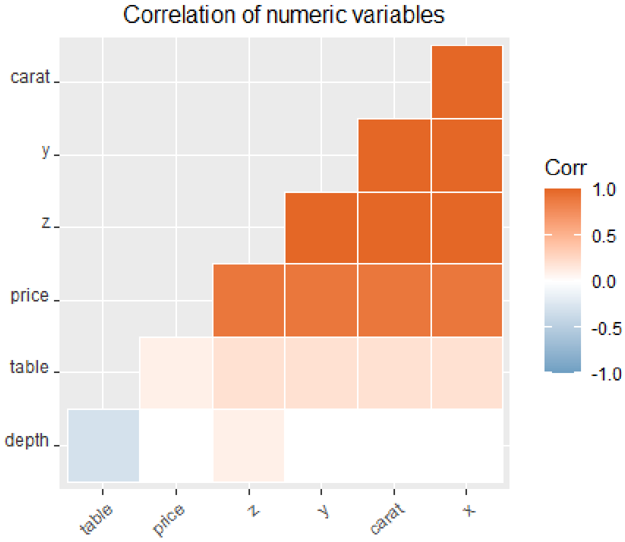


Figure 6.8: Correlation between numeric variables

mitools, and naniar were not much used. These insights were also reflected in the choice of packages in line with the experiment implementation.

Mean and median imputation were carried out with the help of base R, i.e. without any specific libraries. The respective mean and median were used to replace numeric variables, while the mode was chosen to replace categorical variables[6].

For kNN imputation, the package VIM which was also mentioned in the online survey and in the literature review, and required the precondition of library laeken, was applied[7]. The number k was set to 5 neighbors, and the weighted mean was calculated from them.

Regarding CART imputation, the package mice from R was chosen[8]. The CART method was implemented having a minimum bucket of 100 observations per terminal node.

As for RF imputation, the library randomForest provided the rfImpute() function[9]. The formula object in the function was created in a way that the target variable *price* was explained by all features, and the imputation was carried out in three iterations as well as with 100 trees being generated.

Regression imputation was carried out using the package mice which provides the method norm.boot that performs linear regression with bootstrapping for numeric variables,

[6] Mode imputation: stackoverflow.com/questions/67128217/mode-imputation-for-categorical-variables-in-a-dataframe

[7] laeken: www.rdocumentation.org/packages/VIM/versions/6.1.1/topics/kNN

[8] mice: rforpoliticalscience.com/2020/07/28/impute-missing-values-with-mice-package-in-r/ and https://rdrr.io/cran/mice/man/mice.impute.cart.html

[9] rfImpute: math.furman.edu/ dcs/courses/math47/R/library/randomForest/html/rfImpute.html

and in addition mode imputation for categorical data types had to be chosen due to the need for type matching [10].

The XGBoost imputation was done using the package `mixgb` that offers the function `mixgb()` which takes `m` (as the number of imputed datasets), `maxit` (as the number of imputation iterations as parameters [11].

As for the results of the imputation, data checks were carried out to verify the correctness of the actual imputed values, and whether all missing values had been imputed. In addition to this, with each method, the runtime results were captured in a `data.frame` with the columns *method*, *missing_percentage*, and *run1* through *run20*, and written to a CSV file.

When it comes to the subsequent **direct evaluation** (see step 5 in Fig. 6.2), the original (training) data was compared to the replaced or so-called "pseudo-complete" data. For the categorical columns, the percentage of wrong categories imputed was calculated based on the total number of replaced (prior empty) values, whereas for the numeric columns, the RMSE of imputed values compared to the original values was calculated (per method and share of missing values considered).

In line with the **indirect evaluation** phase (see step 6 in Fig. 6.2), support-vector regression (SVR) which is a commonly used supervised machine learning method and regression method and was not chosen as a missing handling method, was applied to fit a statistical model based on the original as well as the imputed training data[12]. The SVR model was called as follows: `svm(formula = price ~ ., data)`. It used a radial SVM kernel, cost parameter of 1, gamma value of 0.04167, and epsilon value of 0.1 with 299 support vectors. Later on, the **predictions for the test datasets** (see step 7 in Fig. 6.2) were made using the previously fitted model. The target variable *price* was predicted for the test datasets for 20 runs per missing data percentage and handling method. In line with this, the well-known and widely used regression performance metrics RMSE and MAE were calculated for each one of the outcome cases. The results were finally put together into one `data.frame` and written to a CSV file. Figure 6.9 depicts an excerpt of the RMSE results `data.frame` which contains all 20 runs and the calculated RMSE metrics per method and missing value percentage.

The actual values were plotted against the predicted values of the target variable for the original and imputed dataset such as the one shown in Figure 6.10 (based on the original dataset) to visually explore the differences[13].

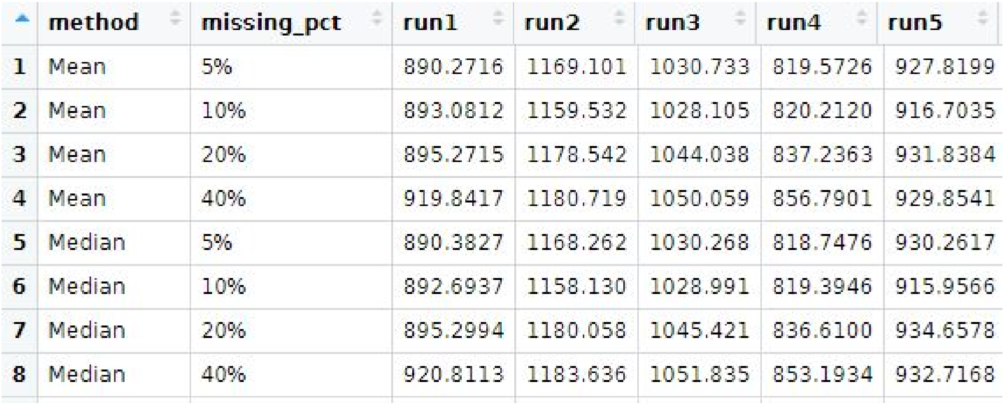
The datasets containing missing values were also analyzed graphically with respect to how many data entries per variable were missing, and how they combine, which is illustrated

[10] `mice`: rforpoliticalscience.com/2020/07/28/impute-missing-values-with-mice-package-in-r/

[11] `mixgb`:cran.r-project.org/web/packages/mixgb/vignettes/Using-mixgb.html; rdrr.io/github/agnesdeng/misle/man/Mixgb.html

[12] SVR: www.datatechnotes.com/2019/09/support-vector-regression-example-with.html

[13] Predicted vs. target: juliejosse.com/wp-content/uploads/2018/06/DataAnalysisMissingR.html



	method	missing_pct	run1	run2	run3	run4	run5
1	Mean	5%	890.2716	1169.101	1030.733	819.5726	927.8199
2	Mean	10%	893.0812	1159.532	1028.105	820.2120	916.7035
3	Mean	20%	895.2715	1178.542	1044.038	837.2363	931.8384
4	Mean	40%	919.8417	1180.719	1050.059	856.7901	929.8541
5	Median	5%	890.3827	1168.262	1030.268	818.7476	930.2617
6	Median	10%	892.6937	1158.130	1028.991	819.3946	915.9566
7	Median	20%	895.2994	1180.058	1045.421	836.6100	934.6578
8	Median	40%	920.8113	1183.636	1051.835	853.1934	932.7168

Figure 6.9: Excerpt of RMSE results `data.frame`

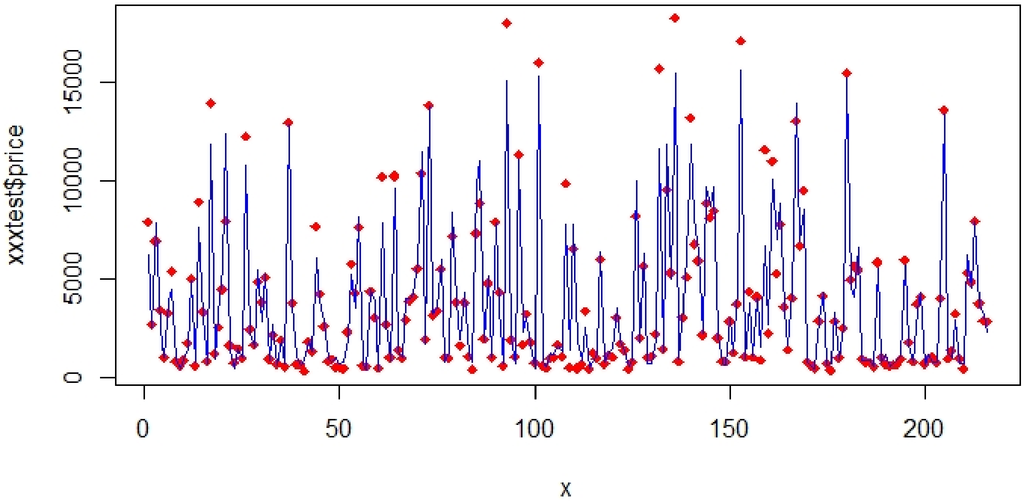


Figure 6.10: Excerpt of RMSE results `data.frame`

by the exemplary plots in Figure 6.11 that uses a dataset with 40% missing data entries as underlying data.

The **performance results** (see step 8 in Fig. 6.2) were then **visualized** and compared against the baseline, i.e. the original dataset and the mean imputation case. The visualizations together with the statistical significance outcomes are summarized in the subsection *Experiment Results* 6.2.4. As for the **statistical significance tests** (see step 9 in Fig. 6.2), the results of the 20 runs on randomly split datasets using holdout method were tested against the baseline, i.e. the original dataset and / or the mean imputation case, the former for RMSE and MAE metrics, and the latter for RMSE, MAE, and

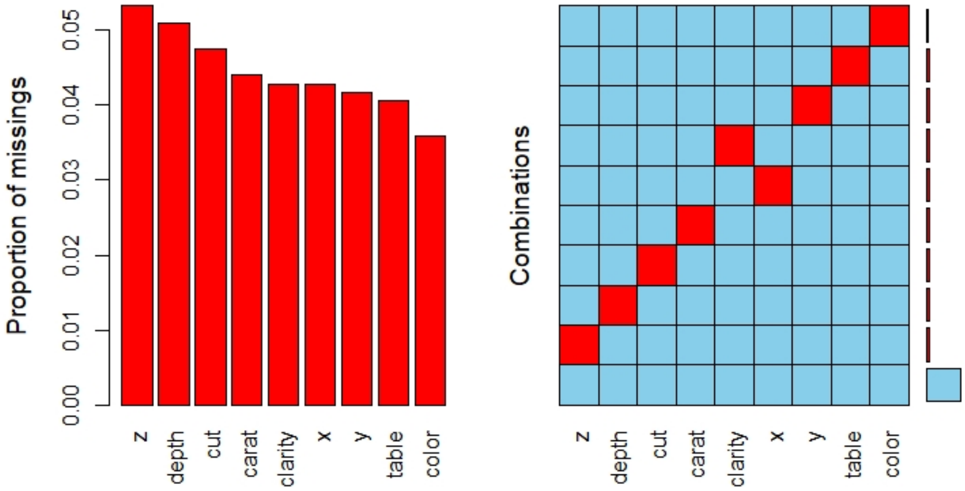


Figure 6.11: Exemplary missing values analysis

execution time. As the case at hand was seen as a paired setting, which means that repeated measures were taken on the same data several times, paired t-tests and Paired Samples Wilcoxon Signed-Rank tests were carried out. Paired t-tests are parametric tests which means that some assumptions (such as ratio target variable, independent observations, normality of the target variable, and non-existence of outliers) required by the data are made[14]. To ensure the quality of the significance testing conclusion, Paired Samples Wilcoxon Signed-Rank tests were executed as a complement to the paired t-tests as they do not require any assumptions on the data distribution, i.e. they are non-parametric[15].

[14] Paired t-tests: www.datanovia.com/en/lessons/t-test-assumptions/paired-t-test-assumptions/

[15] Wilcoxon Paired Samples Signed-Rank Test: www.geeksforgeeks.org/wilcoxon-signed-rank-test-in-r-programming/

6.2.4 Experiment Results

The experiment results comprise the outcomes of the direct and indirect evaluation as well as of the statistical significance tests. Concerning the **direct evaluation**, categorical features and numeric features were analyzed separately. Different metrics were used respectively. The imputed and non-imputed categorical features were compared according to the misclassification rate (in %), i.e. the share of incorrectly classified imputed entries, which are depicted in Table 6.2 that contains the misclassification rate per missing data percentage and categorical variable (being one of *cut*, *color*, or *clarity*). Across the different methods the misclassification rates per missing data percentage and categorical feature were very similar, thus the metrics are shown in the Table 6.2 only once. It also emerges from Table 6.2 that the misclassification rate tends to increase with the missing data percentage, but is in general rather high (i.e. over 60%) throughout all missing data percentages and categorical features considered. In the scenario with 40% of the data entries missing, XGBoost and RF exhibited a slightly worse imputation performance than the other methods.

Direct evaluation: categorical variables		
Missing data %	Variable	Misclassification rate (%)
5%	cut color clarity	67% 60% 75%
10%	cut color clarity	70% 100% 100%
20%	cut color clarity	73% 92% 76%
40%	cut color clarity	70% 100% 84%

Table 6.2: Direct evaluation of imputed categorical variables

Likewise, the direct evaluation performance was measured per missing value percentage and handling method for the numeric variables, more precisely *depth*, *table*, x, y, and z, as it can be seen from Table 6.3. The best performance in terms of RMSE (comparing the original dataset to the imputed one) yielded per missing value percentage and method is highlighted in bold. In the scenario with 5% missing data, the RF method performed best in terms of RMSE, with the XGBoost coming close to the former's result in the variable x. In the case of 10% missing data, the RF method was across all variables again the best, but kNN also achieved a good performance, for two variables even better than RF, even better than XGBoost. When considering 20% of missing data, the RF method still performed best or second best across all but one variable. Mean as well as median performed best in relation to two variables in this case, and kNN in one variable. In the 40% missing data case, the mean and median methods prevailed in three variables, and the RF method achieved the best performance for two variables. Overall, it can be said from the direct evaluation results shown in Table 6.3 that RF handled the missing data situation best in the highest number of scenarios, followed by kNN, XGBoost, and in one scenario even mean and median imputation.

Direct evaluation: numeric variables						
Missing %	Method	RMSE depth	RMSE table	RMSE x	RMSE y	RMSE z
5%	Mean	1.9195	3.0275	1.4869	1.4761	0.9234
	Median	1.9199	3.0303	1.4870	1.4761	0.9235
	kNN	1.9195	3.0307	1.4870	1.4745	0.9229
	CART	1.9243	3.0379	1.4870	1.4746	0.9223
	RF	**1.9186**	**3.0252**	**1.4866**	**1.4742**	**0.9219**
	Regression	1.9220	3.0424	1.4877	1.4745	0.9225
	XGBoost	1.9214	3.0338	**1.4867**	1.4745	0.9221
10%	Mean	1.9197	3.0281	1.4869	1.4761	0.9229
	Median	1.9199	3.0330	1.4869	1.4761	0.9230
	kNN	**1.9192**	3.0322	**1.4868**	**1.4743**	0.9226
	CART	1.9265	3.0423	1.4872	1.4746	0.9222
	RF	**1.9193**	**3.0272**	**1.4869**	**1.4743**	**0.9219**
	Regression	1.9208	3.0468	1.4899	1.4756	0.9225
	XGBoost	1.9215	3.0399	1.4870	1.4748	**0.9220**
20%	Mean	1.9167	3.0249	**1.4856**	1.4749	**0.9217**
	Median	1.9162	3.0284	**1.4856**	1.4750	**0.9217**
	kNN	1.9196	3.0322	1.4869	**1.4742**	0.9223
	CART	1.9244	3.0423	1.4872	1.4751	0.9224
	RF	**1.9156**	**3.0272**	1.4869	**1.4743**	**0.9219**
	Regression	1.9262	3.0468	1.4900	1.4764	0.9240
	XGBoost	1.9227	3.0399	1.4874	1.4747	0.9221
40%	Mean	1.9121	3.0196	**1.4802**	**1.4699**	**0.9179**
	Median	1.9122	3.0246	**1.4803**	**1.4702**	**0.9180**
	kNN	1.9173	3.0291	1.4859	1.4737	0.9205
	CART	1.9343	3.0568	1.4875	1.4753	0.9226
	RF	**1.9117**	**3.0186**	1.4868	1.4744	0.9223
	Regression	1.9227	3.0386	1.4871	1.4748	0.9244
	XGBoost	1.9263	3.0426	1.4874	1.4764	0.9227

Table 6.3: Direct evaluation of imputed numeric variables in terms of RMSE *(Note: best performance highlighted in bold)*

Throughout the missing data handling process, the **execution times of the imputation**

methods were measured for each run and missing data percentage, for which the outcomes are represented in numeric terms (rounded to four decimal places) in Table 6.4 and are visualized in Figure 6.12. It can be seen from Figure 6.12 and Table 6.4 that the runtimes of the CART method were substantially higher than for any other method, and reached approximately 12 to 15 seconds per dataset for all kinds of missing percentages. The next runtime-intense method was XGBoost which took around 4 to 5 seconds per dataset to impute the missing entries. Then the regression and RF method ranged from approximately 1 to 2 seconds in imputation time per dataset (see Table 6.4). The kNN method appeared to be the least runtime-intense among the machine learning methods used with an execution time of approximately 0.1 to 0.5 seconds per imputation. The fastest methods were mean and median imputation which took only around 0.002 to impute the missing entries across all settings.

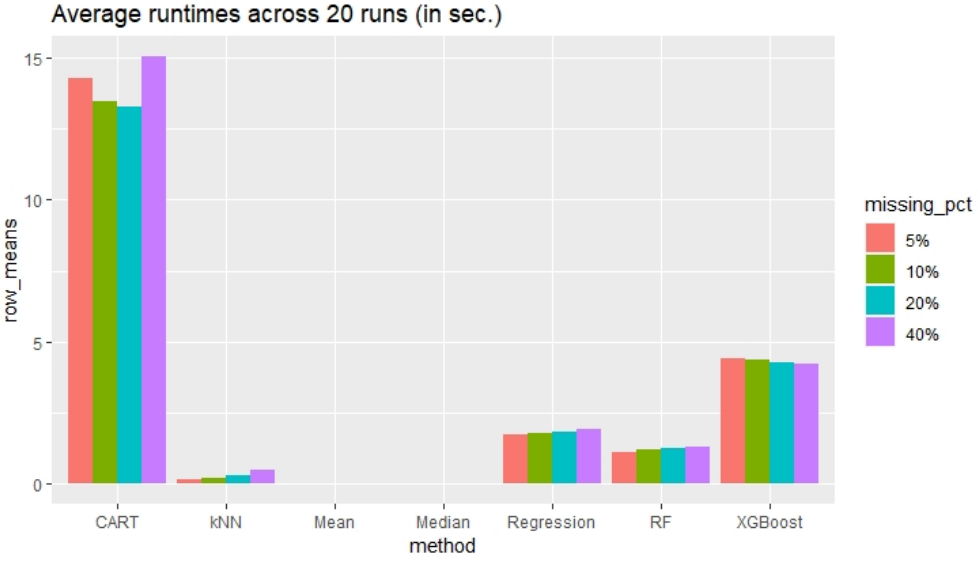


Figure 6.12: Average execution times across 20 runs

When it comes to the **indirect evaluation**, SVR models were fitted based on the training data, and the models were used to make predictions for the test data (per random split, missing data percentage, and missing data handling method). The regression performance was measured in terms of RMSE and MAE.

Figure 6.13 visualizes the average RMSE metrics across the 20 runs using randomly split train and test datasets for varying missing data percentages and handling methods. In Figure 6.13, the average RMSE of the SVR models fitted based on the original, i.e. non-imputed, training is shown as a dashed, horizontal line in the plot and more precisely amounts to 545.0577, which is naturally at or below all methods' RMSE level. From the figure, it emerges that the mean and median imputation methods exhibited the highest RMSE across all settings, which means that they perform worst, followed by CART, regression, and RF methods. The best performing methods in terms of average RMSE

Execution times (in sec.) across 20 runs		
Missing data %	Method	Execution time (sec.)
5%	Mean	**0.0016**
	Median	0.0021
	kNN	0.1159
	CART	14.3134
	RF	1.1182
	Regression	1.7330
	XGBoost	4.4096
10%	Mean	**0.0026**
	Median	**0.0025**
	kNN	0.1817
	CART	13.4544
	RF	1.1818
	Regression	1.7685
	XGBoost	4.3642
20%	Mean	**0.0022**
	Median	**0.0022**
	kNN	0.3026
	CART	13.2997
	RF	1.2566
	Regression	1.8202
	XGBoost	4.2674
40%	Mean	**0.0016**
	Median	0.0033
	kNN	0.4868
	CART	15.0388
	RF	1.3013
	Regression	1.8945
	XGBoost	4.2351

Table 6.4: Execution times (in seconds) across 20 runs *(Note: best performance highlighted in bold)*

across runs were kNN, XGBoost, and in the 5% as well as 20% case also RF. In Table 6.5 the best results per missing data percentage and method combination, i.e. lowest RMSE, are highlighted in bold.

The MAE performance metrics across 20 runs, illustrated by Figure 6.14 and Table 6.6, paint a similar picture as the RMSE did before. Again, the average MAE metric across the 20 runs is indicated as a dashed, horizontal line in the plot at 995.1534 (see 6.14). In contrast to the RMSE metric, with regard to MAE the kNN method performed best, followed by RF. Regression was also among the best two methods in the 10% missing data scenario and XGBoost in the 40% missing data scenario.

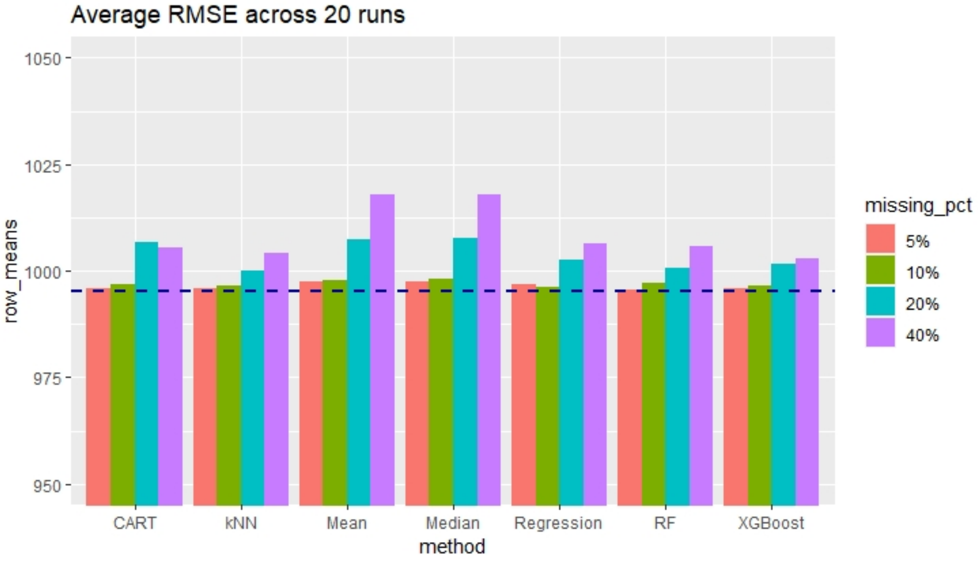


Figure 6.13: Average RMSE across 20 runs

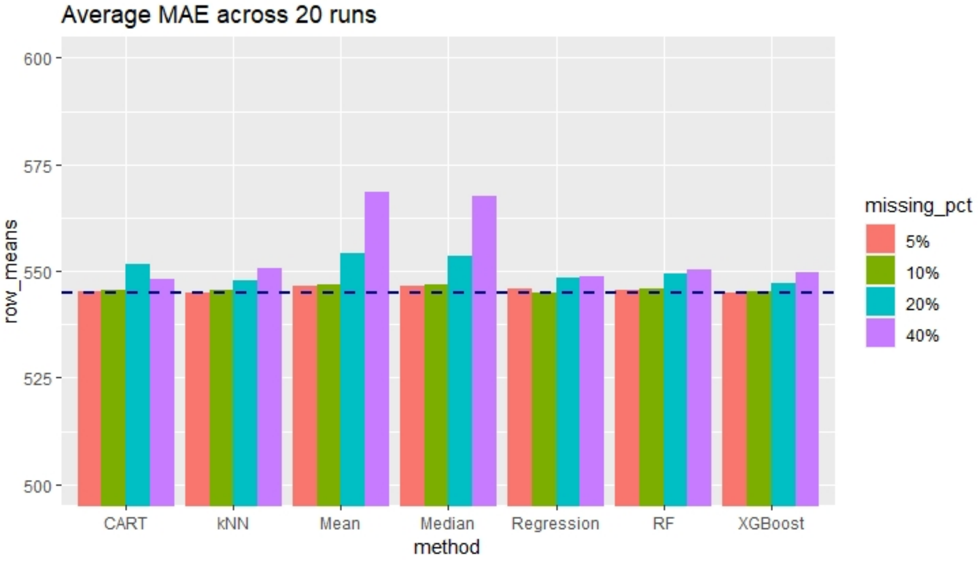


Figure 6.14: Average MAE across 20 runs

To analyze the generated execution time, RMSE, and MAE even further, **statistical significance tests** were performance based on the 20 runs carried out for each method and missing data percentage combination. The tests were applied in a paired setting, which means that repeated measures of the same data were done several times. The paired t-test[16] was chosen as a parametric variant, and the Paired Samples Wilcoxon

[16] Paired t-tests: www.datanovia.com/en/lessons/how-to-do-a-t-test-in-r-calculation-and-reporting

Average RMSE across 20 runs		
Missing data %	Method	Average RMSE
5%	Mean	997.58
	Median	997.57
	kNN	**995.73**
	CART	995.75
	RF	**995.72**
	Regression	996.87
	XGBoost	996.02
10%	Mean	997.87
	Median	997.99
	kNN	**996.42**
	CART	996.79
	RF	997.22
	Regression	**996.30**
	XGBoost	996.50
20%	Mean	1007.39
	Median	1007.68
	kNN	**999.90**
	CART	1006.71
	RF	**1000.81**
	Regression	1002.66
	XGBoost	1001.61
40%	Mean	1017.76
	Median	1017.93
	kNN	**1004.13**
	CART	1005.37
	RF	1005.84
	Regression	1006.32
	XGBoost	**1003.06**

Table 6.5: Average RMSE across 20 runs *(Note: best performance highlighted in bold)*

Signed-Rank test[17] as a non-parametric version. Later on, the outcome of both tests was compared, and it turned out that they yielded the same significance result in all cases.

With reference to execution time, the mean method was chosen as a baseline, and the remaining methods were tested against it, using the 20 runs carried out each. Median and kNN did not perform significantly differently from the mean method (baseline), i.e. they were not significantly faster or lower. However, between mean method and CART, RF, regression, as well as XGBoost the result of the statistical tests were significant for

[17] Wilcoxon Paired Samples Signed-Rank Test: www.geeksforgeeks.org/wilcoxon-signed-rank-test-in-r-programming

Average MAE across 20 runs		
Missing data %	Method	Average MAE
5%	Mean	546.58
	Median	546.55
	kNN	**545.01**
	CART	545.25
	RF	545.67
	Regression	545.86
	XGBoost	**545.02**
10%	Mean	546.83
	Median	546.85
	kNN	546.63
	CART	546.54
	RF	546.82
	Regression	**545.04**
	XGBoost	**545.21**
20%	Mean	554.07
	Median	553.43
	kNN	**547.85**
	CART	551.53
	RF	549.50
	Regression	548.34
	XGBoost	**547.30**
40%	Mean	568.52
	Median	567.75
	kNN	550.82
	CART	**548.21**
	RF	550.24
	Regression	**548.78**
	XGBoost	549.79

Table 6.6: Average MAE across 20 runs *(Note: best performance highlighted in bold)*

all missing value percentages.

As for the RMSE, the mean method was again used as a baseline. In the 5% and 10% missing data settings none of the statistical tests performed were significant, whereas in the 20% case kNN, RF, and XGBoost exhibited a significant different to the mean method. In the 40% missing data setting, the kNN, RF, CART, regression, and XGBoost results were significant compared to the mean method.

When considering MAE as a performance metric, none of the statistical tests yielded significant results in the 5% and 10% missing data percentage cases. However, in the 20% case kNN, RF, and XGBoost were significant, and in the 40% case kNN, CART, RF, regression, and XGBoost led to a significant test result.

When assuming the RMSE performance on the original dataset as a baseline, regression and XGBoost exhibited a significant test result in the 5% missing data percentage case, while in the scenarios with 10% missing data, none of the methods led to a significantly different result. In the 20% missing data setting, kNN was not significant, while all others appeared to lead to a significantly different result. In the case with 40% missing data entries, all methods led to a significant test result compared to the original dataset RMSE performance.

If the MAE performance based on the original dataset is used as a baseline, none of the statistical tests are significant in the 5% and 10% missing data settings. In the 20% missing entries case, kNN and XGBoost do not lead to a significant result, but median, CART, RF, and regression are significantly different from the baseline. In the 40% missing entries case, finally all methods are significantly different from the baseline.

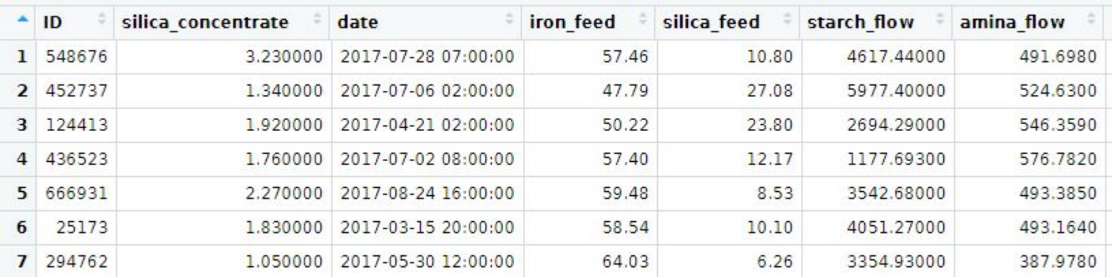
6.3 Industry Case Study

The section Industry Case Study 6.3 covers the data description, exploratory analysis, experiment implementation, and summary of the results. The general case study structure is illustrated in Fig. 6.1.

6.3.1 Dataset Description

For the industry case study a dataset from the manufacturing industry is used, which deals with an actual process in a mining plant that serves for quality control purposes[18] and comprises a series of sensor measurements. The dataset[19] was extracted from Kaggle[20] and is publicly available for reproducibility and verification. In line with this project, a random subset of 1% out of the over 700,000 sensor records is selected from the original dataset so that the computational complexity is kept viable in view of the missing data handling and regression algorithms to apply later on.

The *industry* case dataset used in this project contains 7375 observations (i.e. rows) and 25 features (i.e. columns). It is structured in a way that the unique ID heads the columns, the second column is the target variable *silica_concentrate* which describes the concentration of silica recorded per observation, and the third column the *date* when the sensor measurement was recorded. The remaining columns are considered as features in this project. The Figure 6.15 presents an excerpt of the *industry* dataset, which illustrates the rows and columns layout.



	ID	silica_concentrate	date	iron_feed	silica_feed	starch_flow	amina_flow
1	548676	3.230000	2017-07-28 07:00:00	57.46	10.80	4617.44000	491.6980
2	452737	1.340000	2017-07-06 02:00:00	47.79	27.08	5977.40000	524.6300
3	124413	1.920000	2017-04-21 02:00:00	50.22	23.80	2694.29000	546.3590
4	436523	1.760000	2017-07-02 08:00:00	57.40	12.17	1177.69300	576.7820
5	666931	2.270000	2017-08-24 16:00:00	59.48	8.53	3542.68000	493.3850
6	25173	1.830000	2017-03-15 20:00:00	58.54	10.10	4051.27000	493.1640
7	294762	1.050000	2017-05-30 12:00:00	64.03	6.26	3354.93000	387.9780

Figure 6.15: Excerpt of *industry* dataset

The data which stems from an actual iron ore manufacturing plant was collected with the purpose of detecting possible impurity of the ore earlier than previously possible [17]. Drumond (2018) stated in their paper related to the dataset that the traditional process of detecting the "quality in the flotation concentrate" took at least 120 minutes in the lab [17]. Due to the time-consuming manual analysis process, the researchers suggested to try to find ways to automate the quality control by using machine learning

[18] Manufacturing plant process: www.kaggle.com/datasets/edumagalhaes/quality-prediction-in-a-mining-process

[19] Note: The data source references refers to all places in the chapter mentioning the *industry* or *manufacturing* dataset.

[20] Data Source: www.kaggle.com/datasets/edumagalhaes/quality-prediction-in-a-mining-process

techniques and automation in the form of a so-called “soft sensor“ that predicts the silica concentrate (which is also the target variable in this project) based on the collected sensor measurements [17]. Figure 6.16 depicts the raw materials, such as iron ore, manufacturing or mining process which involves sensors that measure and record parameters for the later analysis and quality control in the (froth) flotation process [50].

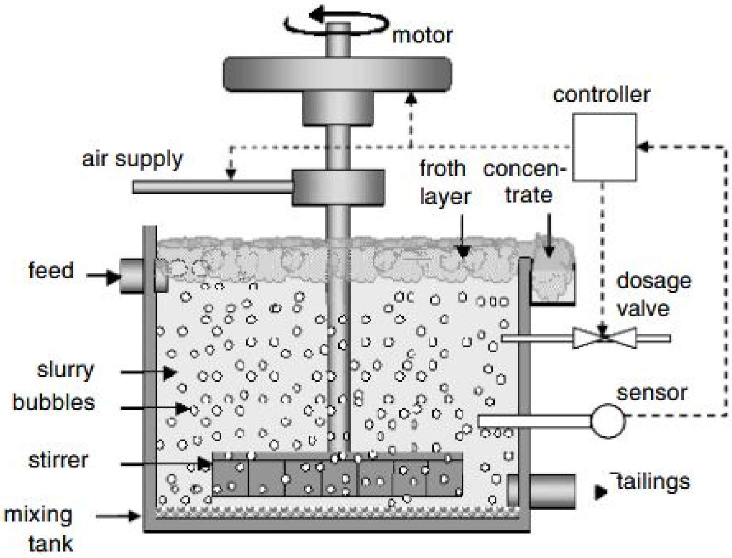


Figure 6.16: Manufacturing process: “froth flotation cell with sensor, actuators, and control system“ by Kramer (2012) [50]

The ID column contains the unique identifier for each observation and is not part of the statistical analysis, the variable *silica*is the numeric target variable that “measures the impurity in the ore concentrate to support manufacturing engineers in taking corrective actions early in the process“[21], and likewise the date is not used as a predictor, whereas the latter columns are features, which are *iron_feed*, *silica_feed*, *starch_flow*, *amina_flow*, *ore_pulp_flow*, *ore_pulp_ph*, *ore_pulp_density*, *flotation_col1_air_flow* through *flotation_col7_air_flow*, *flotation_col1_level* through *flotation_col7_level*, and *iron_concentrate*. All features are of ratio type, i.e. numeric variables. The dataset as such does not contain any missing values in its original form.

Table 6.7 below summarizes the type information regarding the columns, provides a short description of the respective column’s meaning, and its range or categories. The column *price* is annotated as target variable, which is the numeric variable that should be predicted in the indirect evaluation track using regression methods.

[21] Source: www.kaggle.com/datasets/edumagalhaes/quality-prediction-in-a-mining-process

Summary of dataset characteristics			
Variable name	**Type**	**Description**	**Range**
`ID`	Ratio	Unique identifier	15 - 737275
`silica_concentrate` (target)	Ratio	Impurity of ore	0.60 - 5.53
`date`	Nominal	Date	Mar.-Sept. 2017
`iron_feed`	Ratio	Iron feed	42.74 - 65.78
`silica_feed`	Ratio	Silica feed	1.31 - 33.40
`starch_flow`	Ratio	Starch flow	0.58 - 6289.01
`amina_flow`	Ratio	Amina flow	241.90 - 739.20
`ore_pulp_flow`	Ratio	Pulp flow	376.30 - 418.60
`ore_pulp_ph`	Ratio	Pulp pH	8.75 - 10.81
`ore_pulp_density`	Ratio	Pulp density	1.52 - 1.83
`flotation_col1_air_flow`	Ratio	Air flow 1	175.80 - 368.80
`flotation_col2_air_flow`	Ratio	Air flow 2	175.90 - 360.30
`flotation_col3_air_flow`	Ratio	Air flow 3	176.50 - 305.80
`flotation_col4_air_flow`	Ratio	Air flow 4	293.30 - 305.60
`flotation_col5_air_flow`	Ratio	Air flow 5	287.00 - 308.50
`flotation_col6_air_flow`	Ratio	Air flow 6	192.20 - 370.90
`flotation_col7_air_flow`	Ratio	Air flow 7	186.10 - 371.2
`flotation_col1_level`	Ratio	Level 1	151.50 - 861.80
`flotation_col2_level`	Ratio	Level 2	212.70 - 828.40
`flotation_col3_level`	Ratio	Level 3	126.80 - 886.80
`flotation_col4_level`	Ratio	Level 4	162.50 - 679.50
`flotation_col5_level`	Ratio	Level 5	167.70 - 675.00
`flotation_col6_level`	Ratio	Level 6	162.30 - 698.60
`flotation_col7_level`	Ratio	Level 7	176.40 - 657.00
`iron_concentrate`	Ratio	Iron ore	62.06 - 68.01

Table 6.7: Summary of *industry* dataset characteristics

6.3.2 Exploratory Analysis

In this section, the *industry* dataset is subject to exploratory analysis, which involves checking the distribution of the numeric target variable as well as features and the correlation between the columns before the actual experiment is carried out.

From Figure 6.17, it can be seen that the distribution of the variables *amina_flow*, *starch_flow*, and *ore_pulp_ph* is quite evenly spread, while the distribution of *iron_feed* has an accumulation around the mean and the upper values, and *ore_pulp_flow* accumulates rather around the mean. The target variable *silica_feed* is rather evenly spread, but with an accumulation in the lower part, according to Figure 6.17. In contrast to this, Figure 6.18 shows that most flotation air flow columns have concentrations around several values, but the values are not evenly spread, although an accumulation around the mean is visible. Only the variable *ore_pulp_density* is quite evenly spread across the distribution based on Figure 6.18. Figure 6.19 illustrates the distribution of the flotation level columns, which all exhibit a similar distribution pattern that is concentrated on the mean region, and then fades out towards the upper and lower ends of the distribution.

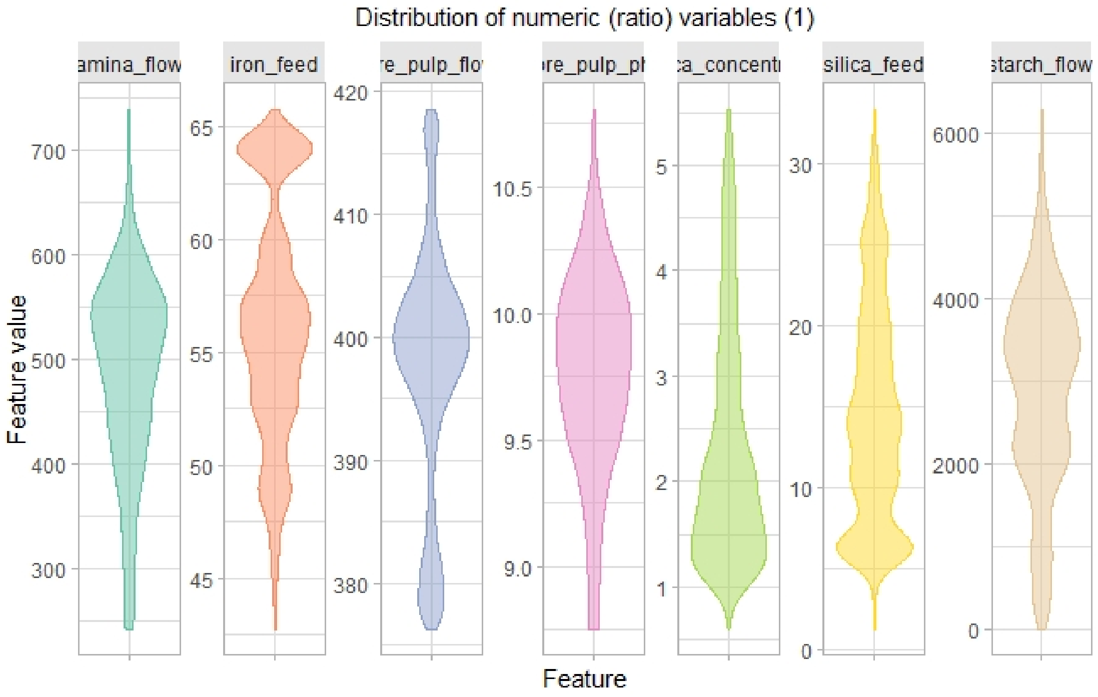


Figure 6.17: Distribution of the numeric variables *amina_flow*, *iron_feed*, *ore_pulp_flow*, *ore_pulp_ph*, *silica_concentrate*, and *silica_feed*, *starch_flow*

The correlation between the numeric variables is calculated in order to grasp the relationship between them, which is summarized by Figure 6.20. The Figure 6.20 illustrates that there is a highly positive correlation among the flotation air flow columns as well as the flotation level columns respectively. Given the fact that those correlations are highly positive, it is considered sufficient to keep only one flotation air flow and flotation level column respectively for the subsequent data analysis and application of imputation as well as machine learning methods. Given the mentioned variables also exhibited a very similar distribution in the exploratory analysis, keeping all of them does not add much information. The remaining variables have a low to medium positive or negative correlation. The target variable *silica_concentrate* has a low to medium negative or positive correlation with most of the features, a higher negative correlation with *iron_feed* and *iron_concentrate*, and a medium positive correlation with *amina_flow*. To sum up, the dataset finally comprises ten numeric features, and one target variable after integrating the alterations suggested in line with the exploratory analysis.

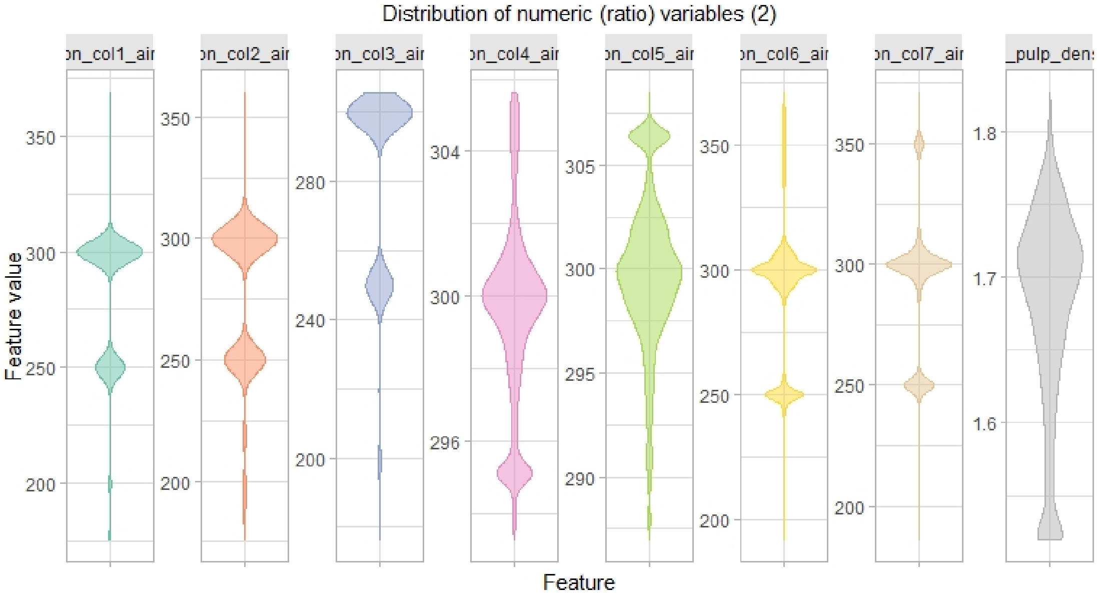


Figure 6.18: Distribution of the numeric variables *flotation_col1_air_flow* through *flotation_col7_air_flow*, and *ore_pulp_density*

6.3.3 Experiment Implementation

Just like the first experiment, the second experiment was also implemented according to the set-up shown in Figure 6.2. The **original dataset** did not contain any categorical features that would re-appear in the subsequent machine learning models. Thus, the purely numeric data did not require any specific pre-processing. The column *ID* was kept heading the data columns. The second column was the target variable *silica_concentrate*, followed by the numeric features.

The experiment conducted on the *industrial sensors* dataset **mirrored the first experiment** in order to make the two experiments comparable and reproducible. The data was split along 20 randomly chosen splits into 80% training data and 20% test data in preparation for the holdout method to be applied later in the regression setting for the indirect evaluation.

The data was transformed into several versions of **pseudo-incomplete datasets** that contained different missing data percentages, i.e. 5%, 10%, 20%, and 40%, in all columns except the *ID* and *silica_concentrate* columns, assuming the MCAR missingness pattern. Then the **missing value handling methods** were applied in order to create complete datasets again, and at the same the **runtime was measured** for each one of the 20 runs. The same missing handling methods as in the first experiment were applied (using the same statistical software libraries and packages as mentioned in the open-source case study 6.2), and were chosen based on the results from the literature review and the online survey:

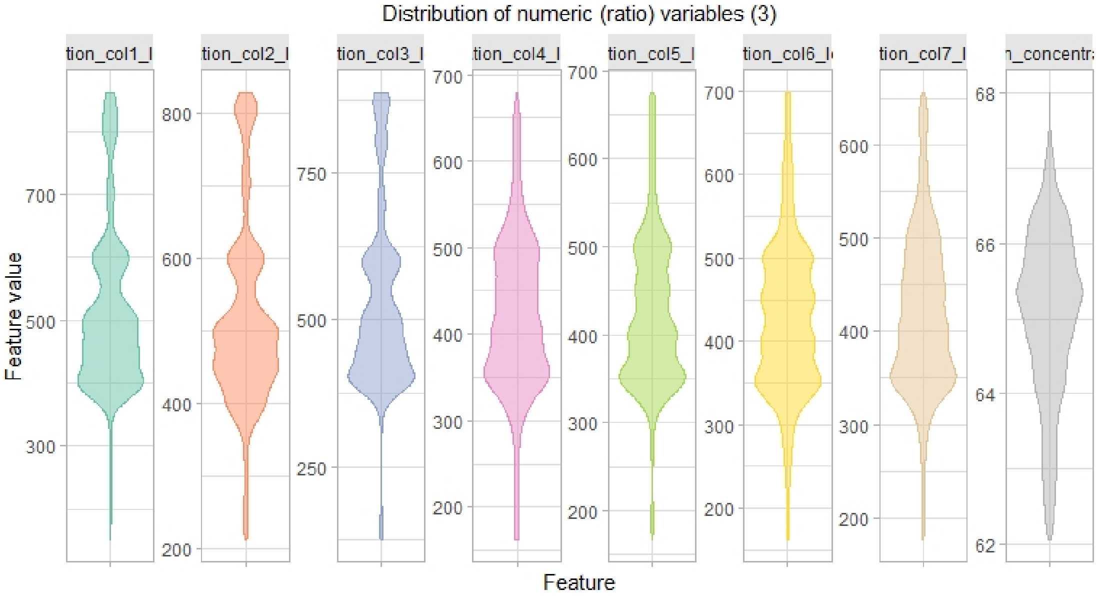


Figure 6.19: Distribution of the numeric variables

- Mean imputation
- Median imputation
- kNN imputation
- CART imputation
- RF imputation
- Regression imputation
- XGBoost imputation

The outcomes of imputing the missing values were subject to a series of checks to verify whether the values were actually imputed and correct. The runtime results per method were captured in a `data.frame` with the columns *method*, *missing_percentage*, and *run1* through *run20*, and written to a CSV file. Concerning the **direct evaluation**, the original (training) data was confronted with the “pseudo-complete“ data. For the numeric columns, the RMSE of imputed values compared to the original values was calculated per method and percentage of missing values.

During **indirect evaluation** phase, SVR was applied to fit a statistical model based on the original as well as the imputed training data[22]. As in the first experiment, the

[22] https://www.datatechnotes.com/2019/09/support-vector-regression-example-with.html

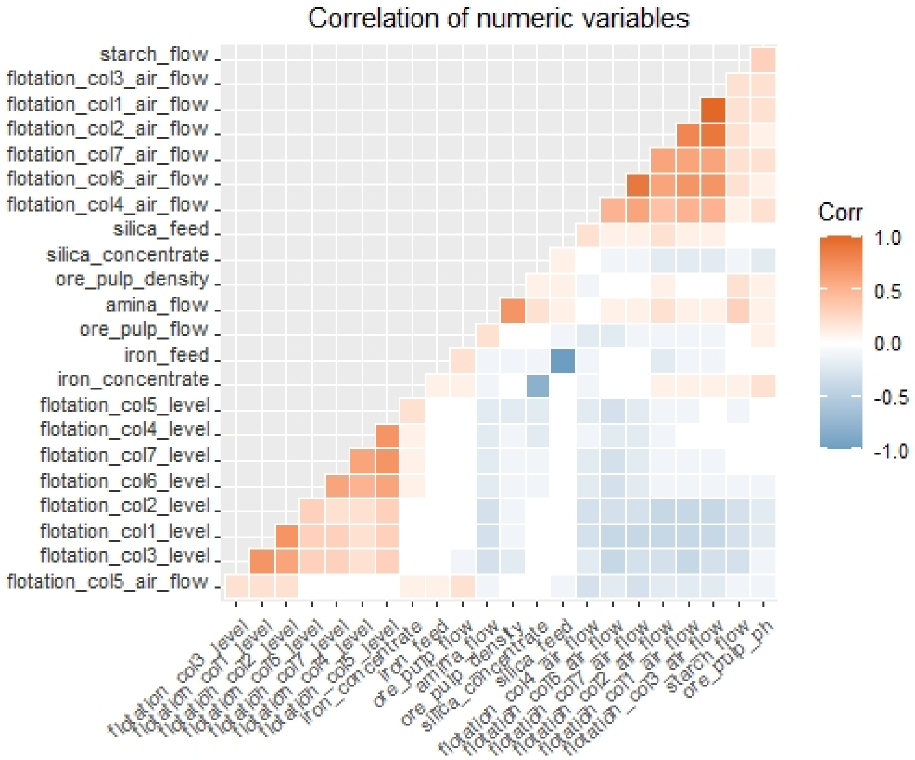


Figure 6.20: Correlation between numeric variables

SVR model was called using the target variable and the features: `svm(formula = silica_concentrate ~ ., data)`. Likewise, it used a radial SVM kernel, cost parameter of 1, gamma value of 0.04167, and epsilon value of 0.1 with 299 support vectors. Then the target variable *silica_concentrate* was predicted for the test datasets for 20 runs per missing data percentage and handling method. In line with this, the well-known and widely used regression performance metrics RMSE and MAE containing all 20 runs and the calculated RMSE metrics per method and missing value percentage were computed. An outcome `data.frame` was created and written to a CSV file. In addition to this, the actual values were plotted against the predicted values of the target variable for the original and predicted dataset such as the one shown in Figure 6.21 (based on the original dataset)[23].

The missing values in the datasets were also analyzed in terms of plots regarding the number of missing values per feature which is shown by the Figure 6.22 to which underlies one of the datasets with 40% missing values.

The **performance results** were finally **visualized** and compared to the specified baseline, i.e. either the original data or the data imputed with the mean. The visualizations

[23] Predicted vs. actual: juliejosse.com/wp-content/uploads/2018/06/DataAnalysisMissingR.html

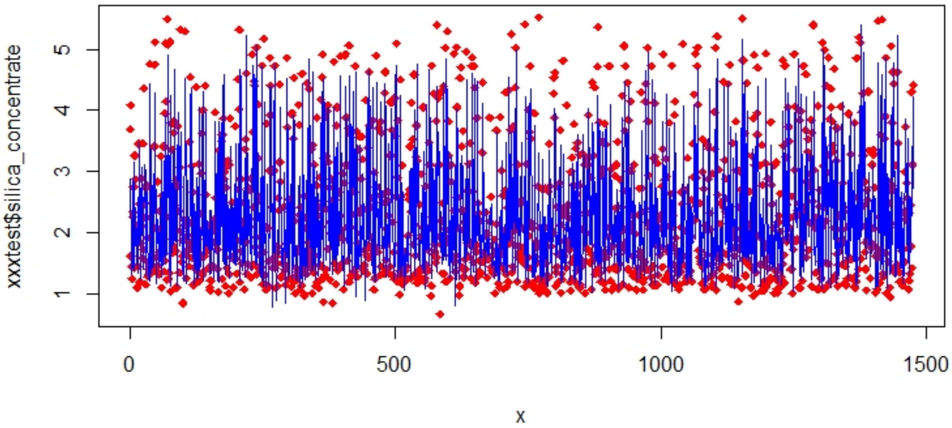


Figure 6.21: Actual vs. predicted values based on SVR using original data

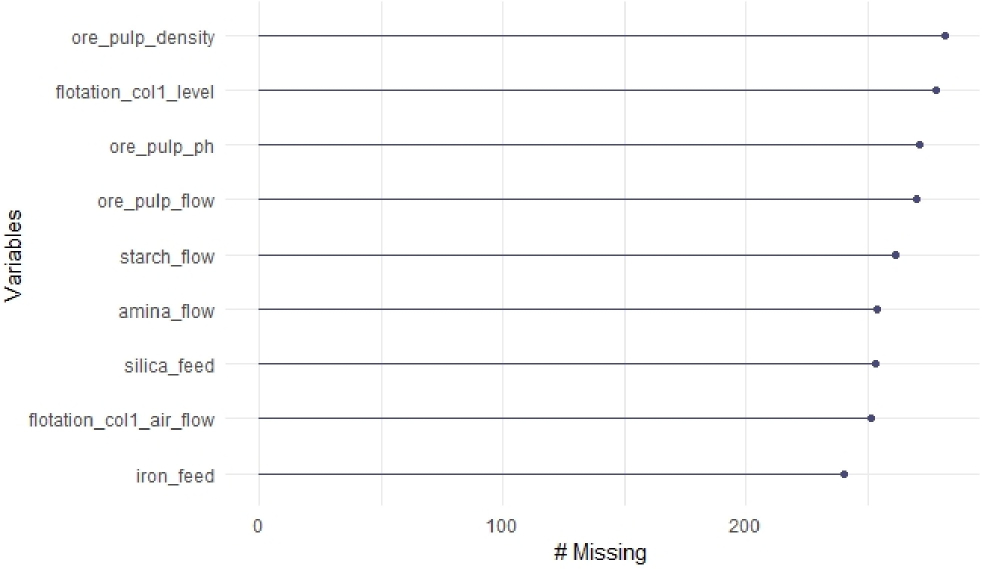


Figure 6.22: Number of entries missing per variable

together with the statistical significance outcomes are summarized in the subsection *Experiment Results* 6.2.4. The **statistical significance tests** were carried out just as outlined in the open-source case study 6.2, and comprised both paired t-tests[24] as well as Paired Samples Wilcoxon Signed-Rank tests[25].

[24] Paired t-tests: www.datanovia.com/en/lessons/t-test-assumptions/paired-t-test-assumptions/

[25] Wilcoxon Paired Samples Signed-Rank Test: www.geeksforgeeks.org/wilcoxon-signed-rank-test-in-r-

6.3.4 Experiment Results

In this chapter, the outcomes of the direct evaluation, the indirect evaluation, and the statistical significance tests are presented. With regard to the **direct evaluation**, the dataset contained merely numeric features, which were analyzed using the RMSE between the imputed dataset and the original dataset respectively. The features `iron_feed`, `silica_feed`, and `starch_flow` had one as an outcome in all scenarios, while for the remaining features, the results (rounded to four decimal places) are summarized in the tables 6.8 (containing the variables `amina_flow`, `ore_pulp_flow`, `ore_pulp_ph`, `ore_pulp_density`) and 6.9 (containing the variables `flotation_col1_air_flow`, `flotation_col1_level`, `iron_concentrate`).

From Table 6.8 and Table 6.9, it can be see that for a missing data rate of 5% the kNN, and RF methods are performing best in terms of (lowest) RMSE across the features. Only with regard to one feature, the CART and XGBoost methods' metrics are best. In the case of 10% missing data, RF performs best throughout the variables, except for `flotation_col1_air_flow`, for which CART is best. In the scenario with 20% missing data, RF still performs quite well, but mean and median imputation also lead to quite good RMSE results. In the 40% missing data case, mean, median, and RF are again the best methods with respect to performance.

Throughout the missing data handling process, the **execution times of the imputation methods** were measured for each run and missing data percentage, for which the outcomes are represented in numeric terms (rounded to four decimal places) in Table 6.4 and are visualized in Figure 6.12. It can be seen from Figure 6.12 and Table 6.4 that the runtimes of the CART method were substantially higher than for any other method, and reached approximately 12 to 15 seconds per dataset for all kinds of missing percentages. The next runtime-intense method was XGBoost which took around 4 to 5 seconds per dataset to impute the missing entries. Then the regression and RF method ranged from approximately 1 to 2 seconds in imputation time per dataset (see Table 6.4). The kNN method appeared to be the least runtime-intense among the machine learning methods used with an execution time of approximately 0.1 to 0.5 seconds per imputation. The fastest methods were mean and median imputation which took only around 0.002 to impute the missing entries across all settings.

The **execution times of the imputation methods** were measured in seconds over the 20 runs, which is illustrated by Table 6.10 and Figure 6.23. From the Table 6.10 and Figure 6.23, it emerges that the RF method has the longest average execution time, which is over 30 seconds per run, followed by CART and kNN with a maximum execution each of around 10 to 15 seconds for a run. Then regression and XGBoost methods take a runtime of around five to seven seconds, and mean as well as median imputation methods have the shortest runtime, which is approximately 0.002 to 0.004 seconds. On the whole, the mean and median imputation methods always hat the shortest runtimes as they

programming/

Direct evaluation: numeric variables (in terms of RMSE)					
Miss.%	Method	amina flow	ore pulp flow	ore pulp ph	ore pulp density
5%	Mean	124.1063	13.0474	**0.5237**	**0.0937**
	Median	124.1007	13.0496	0.5238	**0.0937**
	kNN	**124.0605**	**13.0470**	**0.5237**	**0.0937**
	CART	124.1850	13.0552	0.5242	**0.0937**
	RF	**124.0829**	**13.0451**	**0.5235**	**0.0937**
	Regression	124.2735	13.0710	0.5255	0.0939
	XGBoost	124.1031	13.0493	0.5240	**0.0937**
10%	Mean	124.0499	13.0451	0.5235	**0.0937**
	Median	124.0611	13.0507	0.5235	**0.0937**
	kNN	124.0432	13.0478	0.5236	**0.0937**
	CART	124.2809	13.0673	0.5244	0.0938
	RF	**124.0238**	**13.0441**	**0.5233**	**0.0937**
	Regression	124.6259	13.0808	0.5253	0.0939
	XGBoost	124.1619	13.0732	0.5245	0.0938
20%	Mean	**123.8707**	**13.0312**	0.5230	**0.0935**
	Median	**123.8915**	13.0340	0.5230	**0.0935**
	kNN	124.0813	**13.0513**	0.5238	0.0936
	CART	124.4003	13.1022	0.5250	0.0939
	RF	123.9507	13.0326	**0.5229**	**0.0935**
	Regression	124.6419	13.1369	0.5277	0.0940
	XGBoost	124.3615	13.0864	0.5250	0.0938
40%	Mean	**123.5136**	**12.9882**	**0.5210**	**0.0932**
	Median	**123.5241**	**12.9911**	**0.5211**	**0.0933**
	kNN	124.0344	13.0425	0.5236	0.0936
	CART	124.7418	13.1202	0.5260	0.0940
	RF	**123.7101**	13.0019	0.5214	**0.0933**
	Regression	124.7810	13.1850	0.5290	0.0942
	XGBoost	124.4768	13.1138	0.5259	0.0940

Table 6.8: Direct evaluation of imputed numeric variables in terms of RMSE *(Note: best performance highlighted in bold)*

required the least computational complexity, while for instance methods such as RF, kNN, regression, or XGBoost are more computation-intense.

When it comes to the **indirect evaluation**, the average RMSE across 20 runs was measured for the sensors dataset, which is summarized by Table 6.11 and Figure 6.24. In the scenario with 5% missing data, the CART and RF methods are performing best with a RMSE of around 0.5461, which almost corresponds to the RMSE on the original dataset which was recorded as 0.5458 (see Table 6.11 and dashed line in Figure 6.24). In the 10% missing data case, RF and XGBoost are the best methods, but are closely

Direct evaluation: numeric variables (in terms of RMSE)				
Missing %	Method	flotation air flow	flotation level	iron concentrate
5%	Mean	39.2940	174.9449	1.4881
	Median	38.3253	175.0017	1.4881
	kNN	39.2802	**174.8637**	1.4881
	CART	**39.2533**	175.0948	1.4881
	RF	39.2835	**174.8640**	1.4881
	Regression	39.3207	175.2730	1.4881
	XGBoost	**39.2501**	175.0504	1.4881
10%	Mean	39.2901	**174.8277**	1.4881
	Median	39.3481	174.9078	1.4881
	kNN	39.3027	174.9509	1.4881
	CART	**39.2567**	175.2017	1.4881
	RF	39.2840	**174.8109**	1.4881
	Regression	39.3973	175.8792	1.4881
	XGBoost	**39.2523**	175.0454	1.4881
20%	Mean	**39.2476**	**174.6376**	1.4881
	Median	39.3980	174.7324	1.4881
	kNN	39.3067	174.9202	1.4881
	CART	39.2851	175.6417	1.4881
	RF	**39.2483**	**174.7166**	1.4881
	Regression	39.4411	175.7975	1.4881
	XGBoost	39.2604	175.2260	1.4881
40%	Mean	**39.1519**	**173.9140**	1.4881
	Median	39.4159	**173.9727**	1.4881
	kNN	39.3071	174.8340	1.4881
	CART	39.3158	175.5035	1.4881
	RF	**39.1869**	**174.3888**	1.4881
	Regression	39.5537	176.2457	1.4881
	XGBoost	39.3280	175.4446	1.4881

Table 6.9: Direct evaluation of imputed numeric variables in terms of RMSE *(Note: best performance highlighted in bold)*

followed by kNN and CART. If the missing rate is 20%, the XGBoost method is slightly better than the kNN and RF methods. Finally, in the 40% missing data scenario, the kNN and CART methods perform best with regard to RMSE, followed by XGBoost. Thus, it can be seen that CART, kNN, XGBoost, and RF achieved the best performance.

The second performance metric measured in line with the indirect evaluation was MAE, whose results are depicted in Figure 6.25 (with the MAE of 0.3887 based on the original dataset as a dashed horizontal line) and Table 6.12. It can be seen that kNN and CART are in most scenarios among the best methods with regard to MAE performance, most of

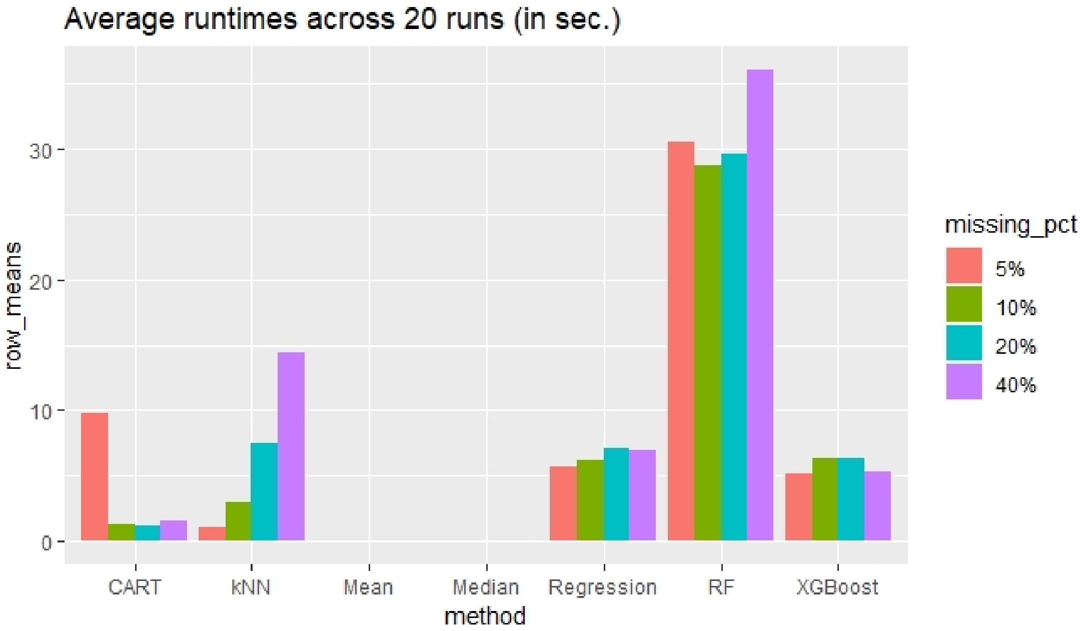


Figure 6.23: Average execution times across 20 runs

the time closely followed by XGBoost, and then RF, while regression, mean, and median imputation are not so favorable related to MAE.

A series of **statistical significance tests** (i.e. paired t-tests and Paired Samples Wilcoxon Signed-Rank tests) were carried out to analyze execution time, RMSE, and MAE in greater detail based on the 20 runs of data generated for each missing data handling method and missing data percentage. Like it was the case in the open-source case study (see 6.2), the parametric and non-parametric tests also came to the same overall conclusion, meaning they were either both significant or not significant for each tests performed.

Concerning the significance tests on execution time, the respective data from the runs were tested against the mean imputation method runtime as a baseline. This procedure resulted in the 5%, 10%, 20%, and 40% missing data case in significant results for all imputation methods, which means that their runtime was in every case significantly different (namely longer) from the one of the mean imputation method.

When it comes to the RMSE as a performance metric, the data from the 20 runs were first tested against the mean imputation method as a baseline. This resulted in the 5% missing data rate scenario in the CART, RF, and XGBoost methods yielding a significant result against the mean imputation method, while the median, kNN, and regression methods results were not significantly different from the mean imputation method. In the 10% missing values case, kNN, RF, XGBoost were significant (i.e. much lower than that of the mean), and CART became non-significant in contrast to the 5% missing

Execution times (in sec.) across 20 runs		
Missing data %	Method	Execution time (sec.)
5%	Mean	**0.0019**
	Median	0.0042
	kNN	1.0575
	CART	9.7549
	RF	30.4148
	Regression	5.7309
	XGBoost	5.2484
10%	Mean	**0.0021**
	Median	0.0041
	kNN	3.0564
	CART	1.2784
	RF	28.7110
	Regression	6.1969
	XGBoost	6.3812
20%	Mean	**0.0018**
	Median	0.0059
	kNN	7.4662
	CART	1.1570
	RF	29.5367
	Regression	7.1308
	XGBoost	6.3657
40%	Mean	**0.0039**
	Median	**0.0040**
	kNN	14.4620
	CART	1.6123
	RF	36.0018
	Regression	6.9424
	XGBoost	5.2927

Table 6.10: Execution times (in seconds) across 20 runs *(Note: best performance highlighted in bold)*

data scenario. The median and regression imputation method stayed non-significant. In the 20% missing data scenario, the kNN, CART, RF, and XGBoost imputation method results were significant again, and the RMSE results of median as well as regression were not. In the 40% missing data case, kNN, CART, RF, and XGBoost yielded a significant test result, while the median and regression method were not significant based on the RMSE data.

To analyze the RMSE data from a second angle, it was tested against the RMSE results based on the original dataset in line with the SVR. In the 5% missing rate scenario, the mean, median, regression, and XGBoost methods led to a significant test result,

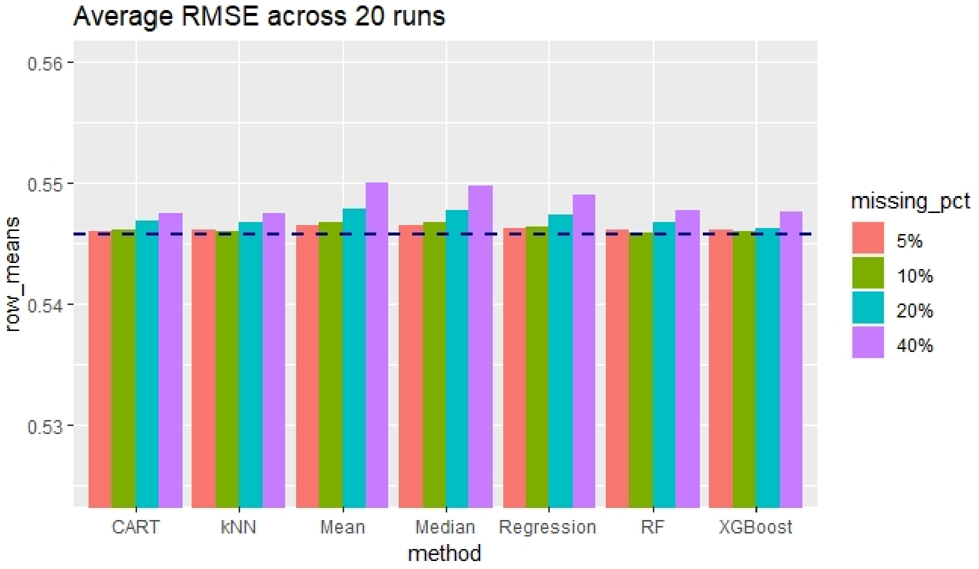


Figure 6.24: Average RMSE across 20 runs

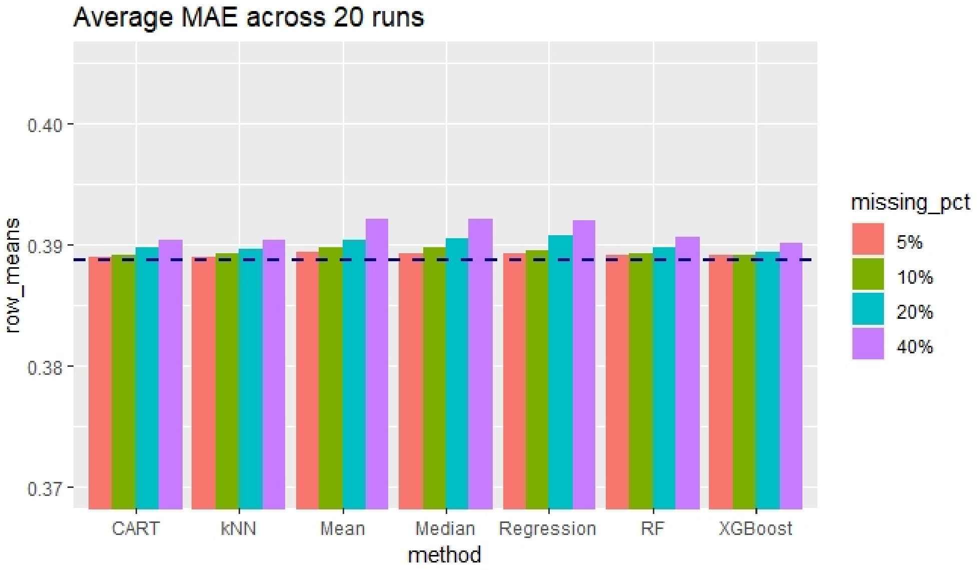


Figure 6.25: Average MAE across 20 runs

which indicates a significant difference (in this case to the worse, i.e. larger RMSE). The RMSE results from the kNN, CART, and RF runs were not significantly different from the RMSE metric on the original dataset. In the 10% missing rate case, the mean and median method results were significantly different from the results stemming from the original dataset, while kNN, CART, RF, regression, and XGBoost method results were not. In the 20% and 40% missing rate scenario, all methods' results tested against the

Average RMSE across 20 runs		
Missing data %	Method	Average RMSE
5%	Mean	0.5465
	Median	0.5465
	kNN	0.5462
	CART	**0.5461**
	RF	**0.5461**
	Regression	0.5462
	XGBoost	0.5462
10%	Mean	0.5468
	Median	0.5467
	kNN	**0.5461**
	CART	**0.5461**
	RF	**0.5459**
	Regression	0.5464
	XGBoost	**0.5460**
20%	Mean	0.5478
	Median	0.5478
	kNN	**0.5467**
	CART	0.5469
	RF	**0.5467**
	Regression	0.5474
	XGBoost	**0.5463**
40%	Mean	0.5500
	Median	0.5498
	kNN	**0.5475**
	CART	**0.5475**
	RF	0.5478
	Regression	0.5490
	XGBoost	**0.5476**

Table 6.11: Average RMSE across 20 runs *(Note: best performance highlighted in bold)*

results on the original dataset were significantly different.

As for the MAE performance metric, the results recorded from the considered imputation methods were tested against the mean imputation method. In the 5% missing rate scenario, the RF method led to a significantly different result. The remaining methods' results were not significant. In the 10%, 20%, and 40%, the same significance test conclusion were made, namely the kNN, CART, RF, and XGBoost method results were significant (with a much lower MAE than the mean imputation method), but the median and regression method results were not significantly different.

The second perspective, which takes the MAE results based on the original dataset as a

Average MAE across 20 runs		
Missing data %	Method	Average MAE
5%	Mean	0.3893
	Median	0.3893
	kNN	**0.3890**
	CART	**0.3890**
	RF	0.3891
	Regression	0.3892
	XGBoost	0.3891
10%	Mean	0.3898
	Median	0.3898
	kNN	**0.3892**
	CART	**0.3891**
	RF	0.3893
	Regression	0.3895
	XGBoost	**0.3891**
20%	Mean	0.3904
	Median	0.3905
	kNN	**0.3896**
	CART	0.3897
	RF	0.3897
	Regression	0.3907
	XGBoost	**0.3894**
40%	Mean	0.3921
	Median	0.3922
	kNN	**0.3904**
	CART	**0.3903**
	RF	0.3906
	Regression	0.3920
	XGBoost	**0.3901**

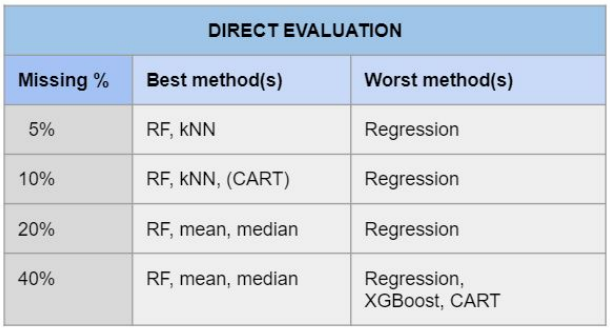
Table 6.12: Average MAE across 20 runs *(Note: best performance highlighted in bold)*

baseline, led to the conclusion that in the 5% missing rate case, the mean, median, kNN, RF, regression, and XGBoost method results were significantly different in terms of MAE. They were all much larger than the MAE on the original dataset. The CART method MAE results were not significantly different from the MAE on the original dataset. In the 10% missing rate case, the mean, median, kNN, RF, and regression method results were significant, while the CART and XGBoost method results were not. In the 20% and 40% missing rate scenario, all methods' results were significantly different from the MAE results on the original dataset.

6.4 Comparison of Results

In this chapter, the results in terms of performance metrics recorded in line with the open-source case study (see 6.2) and the industry case study (see 6.3) are compared on the one hand in terms of best and worst missing value handling methods in the various missing rate scenarios. On the other hand, statistical significance tests were carried out on the experiment runs, whose results are contrasted at this point.

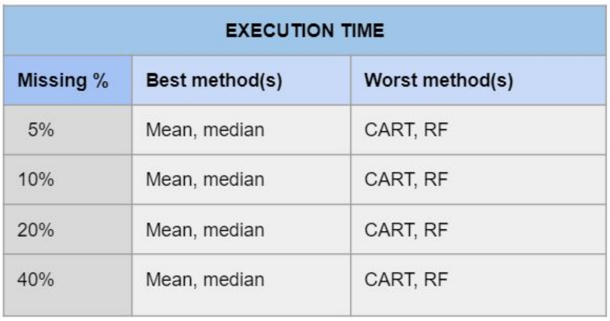
In the **direct evaluation** the results across both case studies were similar and led to an extraction of the best methods, namely RF and kNN in the 5% and 10% missing rate scenarios, and RF, mean as well as median in the 20% and 40% missing rate scenarios, as shown in Figure 6.26. The corresponding worst missing data handling methods in both case studies was regression across all missing rate scenarios, and in the 40% missing rate scenario XGBoost and CART in addition.



DIRECT EVALUATION		
Missing %	**Best method(s)**	**Worst method(s)**
5%	RF, kNN	Regression
10%	RF, kNN, (CART)	Regression
20%	RF, mean, median	Regression
40%	RF, mean, median	Regression, XGBoost, CART

Figure 6.26: Comparison regarding direct evaluation

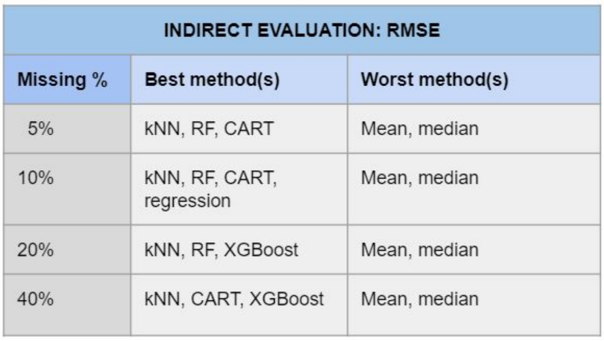
As for the **execution time** of the missing handling methods, mean and median imputation were the fastest methods across all missing rate scenarios, while CART and RF were the most resource and computation intensive in both case studies (see Figure 6.27.



EXECUTION TIME		
Missing %	**Best method(s)**	**Worst method(s)**
5%	Mean, median	CART, RF
10%	Mean, median	CART, RF
20%	Mean, median	CART, RF
40%	Mean, median	CART, RF

Figure 6.27: Comparison regarding execution time

In line with the **indirect evaluation**, the **RMSE performance metrics** were best for the kNN, RF, and CART methods in the 5% and 10% missing rate scenarios, while for higher missing rates only kNN remained among the best across both scenarios consistently, as Figure 6.28 shows. In the 20% missing rate case, kNN, RF, and XGBoost were best, while in case of 40% missing data kNN, CART, and XGBoost were best. The worst methods across both case studies and all missing rate scenarios were mean as well as median imputation.

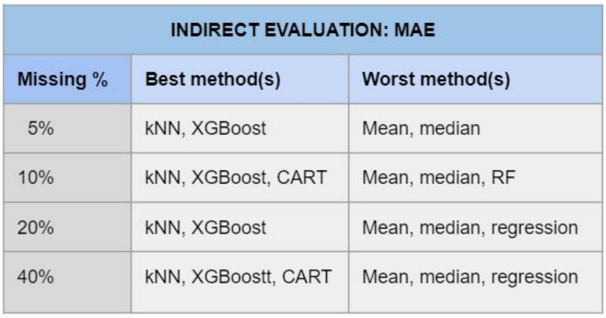


INDIRECT EVALUATION: RMSE

Missing %	Best method(s)	Worst method(s)
5%	kNN, RF, CART	Mean, median
10%	kNN, RF, CART, regression	Mean, median
20%	kNN, RF, XGBoost	Mean, median
40%	kNN, CART, XGBoost	Mean, median

Figure 6.28: Comparison regarding RMSE

Furthermore, **MAE performance** was measured in line with the **indirect evaluation**. Figure 6.29 illustrates that the kNN and XGBoost turned out as the best methods across both case studies and all missing rate scenarios with regard to MAE, while CART method was among the best performing methods only in the 10% and 40% missing rate scenarios. The worst methods across both case studies related to the MAE were mean and median, and in the 20% and 40% missing rate scenarios regression as well.

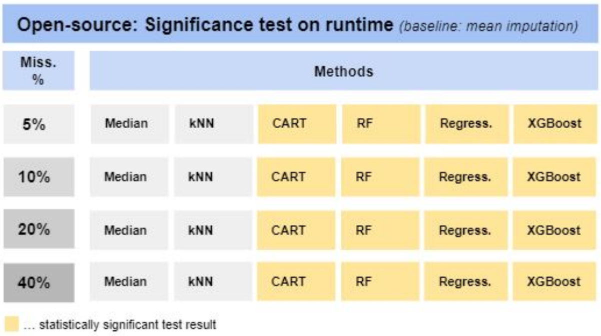


INDIRECT EVALUATION: MAE

Missing %	Best method(s)	Worst method(s)
5%	kNN, XGBoost	Mean, median
10%	kNN, XGBoost, CART	Mean, median, RF
20%	kNN, XGBoost	Mean, median, regression
40%	kNN, XGBoostt, CART	Mean, median, regression

Figure 6.29: Comparison regarding MAE

As 20 runs of random data splits were used to evaluate the runtime, RMSE, and MAE performance, several **statistical significance tests** were carried out in a paired setting. The first one was performed based on the **runtime data** for both the open-source as well as the industry data **with the mean imputation method as a baseline**. Figure 6.30 and 6.31 illustrate all significance tests performed, with the significant results being highlighted in yellow. Both figures show a similar pictures which indicates that almost all methods (except for the median and kNN imputation in the open-source case study) are significantly different from the baseline, i.e. mean imputation. They had much longer runtimes than mean imputation in the case studies.

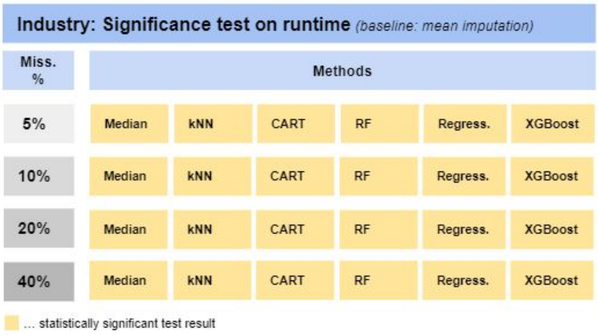


Open-source: Significance test on runtime *(baseline: mean imputation)*

Miss. %	Methods					
5%	Median	kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
10%	Median	kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
20%	Median	kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
40%	Median	kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.30: Open: Significance test on runtime *(baseline: mean imputation)*



Industry: Significance test on runtime *(baseline: mean imputation)*

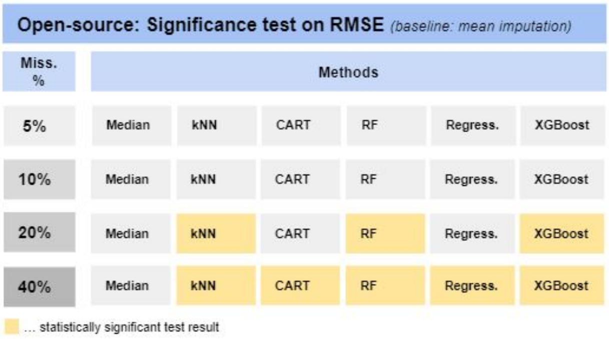
Miss. %	Methods					
5%	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
10%	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
20%	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
40%	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.31: Industry: Significance test on runtime *(baseline: mean imputation)*

When considering the **RMSE with the mean imputation method results** as a **baseline**, Figure 6.32 and 6.33 illustrate the significance test results. They differ between the two case studies in a way that for instance in the 5% and 10% missing rate scenarios in the open-source case study no significant difference between the mean imputation and other methods is detected, but in the industry case study significant differences occur already with CART, RF, kNN, and XGBoost methods in the lower missing rate scenarios. In the higher two missing rate scenarios in both case studies, machine learning methods (i.e. kNN, CART, RF, and XGBoost) notably play a role as their RMSE appears to be significantly different. The machine learning methods' RMSE is typically much lower under those conditions.

RMSE with the SVR results on the original dataset as a **baseline** was carried out as well to gain insights into the performance of the SVR based on the original non-imputed data compared to the imputed datasets, Figure 6.34 and 6.35 illustrate the significance test results. The results were quite similar across the case studies, with the exception that the mean and median imputation methods results were already significantly different from the results yielded on the original data in the lower two missing rate scenarios (see

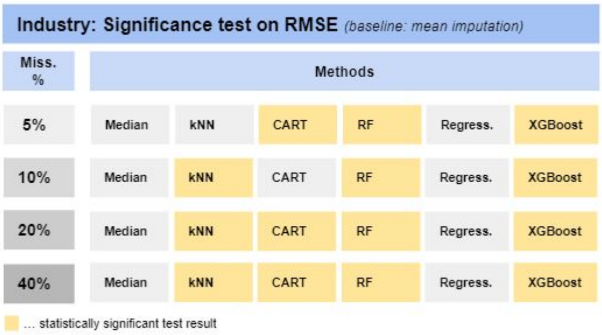


Open-source: Significance test on RMSE *(baseline: mean imputation)*

Miss. %	Methods					
5%	Median	kNN	CART	RF	Regress.	XGBoost
10%	Median	kNN	CART	RF	Regress.	XGBoost
20%	Median	🟨 kNN	CART	🟨 RF	Regress.	🟨 XGBoost
40%	Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.32: Open: Significance test on RMSE *(baseline: mean imputation)*

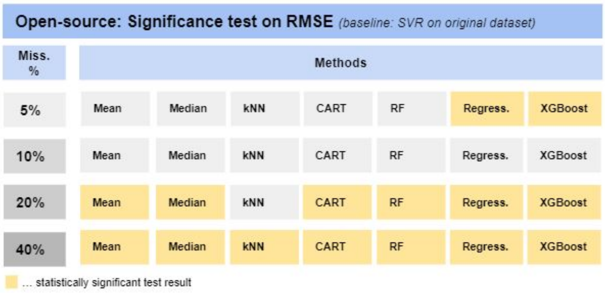


Industry: Significance test on RMSE *(baseline: mean imputation)*

Miss. %	Methods					
5%	Median	kNN	🟨 CART	🟨 RF	Regress.	🟨 XGBoost
10%	Median	🟨 kNN	CART	🟨 RF	Regress.	🟨 XGBoost
20%	Median	🟨 kNN	🟨 CART	🟨 RF	Regress.	🟨 XGBoost
40%	Median	🟨 kNN	🟨 CART	🟨 RF	Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.33: Industry: Significance test on RMSE *(baseline: mean imputation)*



Open-source: Significance test on RMSE *(baseline: SVR on original dataset)*

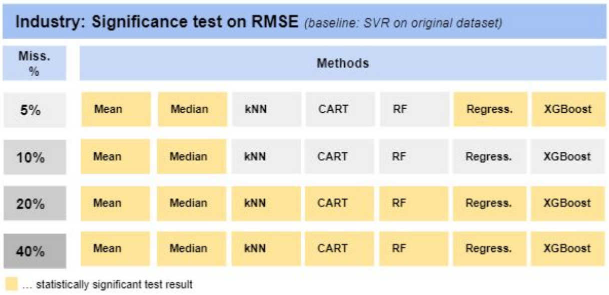
Miss. %	Methods						
5%	Mean	Median	kNN	CART	RF	🟨 Regress.	🟨 XGBoost
10%	Mean	Median	kNN	CART	RF	Regress.	XGBoost
20%	🟨 Mean	🟨 Median	kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
40%	🟨 Mean	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.34: Open: Significance test on RMSE *(baseline: SVR on original data)*

6.35). Apart from that, the RMSE was significantly different for all methods in the higher two missing rate scenarios. In effect, it was much larger. But in the lower two missing rate scenarios, CART, kNN, RF, and XGBoost offered good opportunities to get performance results close to the ones yielded on the original dataset.

As for the **MAE metric** with the **mean imputation as a baseline**, the MAE results were significantly different mainly in the 20% and 40% missing rate scenarios in both case studies (see Figure 6.36 and 6.37). The larger the share of missing data, the more apparent the difference between the mean or median methods and the machine learning



Industry: Significance test on RMSE *(baseline: SVR on original dataset)*

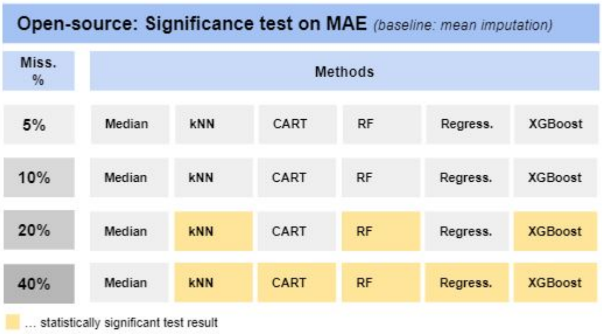
Miss. %	Methods						
5%	🟨 Mean	🟨 Median	kNN	CART	RF	🟨 Regress.	🟨 XGBoost
10%	🟨 Mean	🟨 Median	kNN	CART	RF	Regress.	XGBoost
20%	🟨 Mean	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
40%	🟨 Mean	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.35: Industry: Significance test on RMSE *(baseline: SVR on original data)*

methods became, as the MAE for the latter was comparatively lower.

When using the **SVR on the original data** as a **baseline for the significance test related to the MAE**, the results in the open-source and the industry case study differ in a way that in the open-source study the majority of tests in the 5% and 10% missing rate scenarios are not significant, so the MAE results from the imputed datasets are still rather similar to the ones based on the original dataset (see Figure 6.38 and 6.39). However, for the higher two missing rate scenarios, the difference gets significant, which means that the MAE results are not near the ones based on the original dataset.

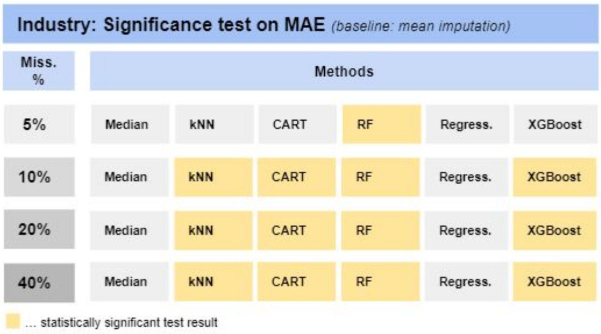


Open-source: Significance test on MAE *(baseline: mean imputation)*

Miss. %	Methods					
5%	Median	kNN	CART	RF	Regress.	XGBoost
10%	Median	kNN	CART	RF	Regress.	XGBoost
20%	Median	🟨 kNN	CART	🟨 RF	Regress.	🟨 XGBoost
40%	Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.36: Open: Significance test on MAE *(baseline: mean imputation)*

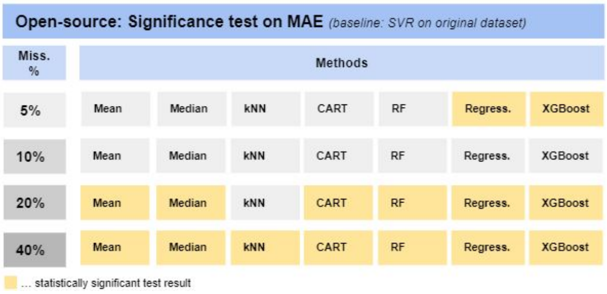


Industry: Significance test on MAE *(baseline: mean imputation)*

Miss. %	Methods					
5%	Median	kNN	CART	🟨 RF	Regress.	XGBoost
10%	Median	🟨 kNN	🟨 CART	🟨 RF	Regress.	🟨 XGBoost
20%	Median	🟨 kNN	🟨 CART	🟨 RF	Regress.	🟨 XGBoost
40%	Median	🟨 kNN	🟨 CART	🟨 RF	Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.37: Industry: Significance test on MAE *(baseline: mean imputation)*

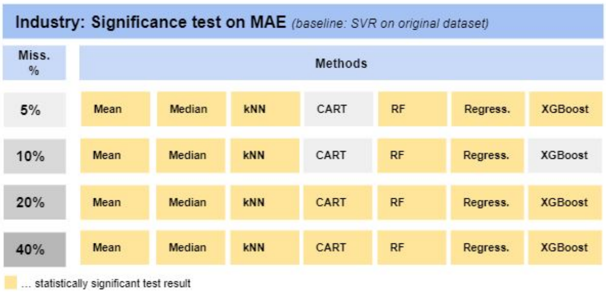


Open-source: Significance test on MAE *(baseline: SVR on original dataset)*

Miss. %	Methods						
5%	Mean	Median	kNN	CART	RF	🟨 Regress.	🟨 XGBoost
10%	Mean	Median	kNN	CART	RF	Regress.	XGBoost
20%	🟨 Mean	🟨 Median	kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
40%	🟨 Mean	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.38: Open: Significance test on MAE *(baseline: SVR on original data)*



Industry: Significance test on MAE *(baseline: SVR on original dataset)*

Miss. %	Methods						
5%	🟨 Mean	🟨 Median	🟨 kNN	CART	🟨 RF	🟨 Regress.	🟨 XGBoost
10%	🟨 Mean	🟨 Median	🟨 kNN	CART	🟨 RF	🟨 Regress.	XGBoost
20%	🟨 Mean	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost
40%	🟨 Mean	🟨 Median	🟨 kNN	🟨 CART	🟨 RF	🟨 Regress.	🟨 XGBoost

🟨 ... statistically significant test result

Figure 6.39: Industry: Significance test on MAE *(baseline: SVR on original data)*

CHAPTER 7

Discussion and Limitations

7.1 RQ1: Data quality dimensions and relation to completeness

The first research question aimed at shedding light on the data quality dimensions as well as specifically data completeness and with a view to CPPS, and was formulated as follows:

RQ1: ***Which dimensions does data quality have, and what is the relation of it to data completeness?***

In view of the Related Work 2, it turned out that data quality of such cannot be described by merely one all-encompassing and uniform definition, but rather comprises a combination of dimensions [45] [4] [85]. Thus, data quality is multi-faceted, and the notion "dimension" was chosen in order to convey the meaning of making data quality evaluable or measurable in a quantitative way [43] [18]. Data quality corresponds to the degree how well the data sources available to the individual or organization analyzing it matches with the actual data [38].

One of the most widely used data quality models was developed by the ISO, and contains *inherent* as well as *system-dependent* (i.e. software- and hardware-related) dimensions [42] [48]. The inherent dimensions comprise accuracy, completeness, consistency, credibility, currentness, accessibility, compliance, efficiency, and confidentiality, which highlights the relation of data completeness to data quality as one of its main dimensions [42] [48]. Furthermore, research by for example Jayawardene (2013) [43], Nikiforova (2018) [64], Cai (2015) [10], Song (2020) [81], Günther (2019) [34], Liu (2020) [56], and Geo (2019) [29] amongst others all outlined data completeness as one of the main dimensions of data quality, notably in a domain like IoT and industrial applications, which "relies heavily on the quality of the data collected by devices", where the loss of data entries might occur due to various issues [56]. Data completeness can be hampered in this particular domain by inconsistent data, diverging formats, missing values, duplicates, or erroneous sensor measurements [99] [56]. According to Liu et al. (2020), the by far most frequently pointed out issue in data preparation and analysis was "missing data" [56], which finally emphasized again the assignment of a higher weight to treating incomplete data, and recognizing its impact on overall data quality.

At this point, it has to be mentioned that the literature analyzed was limited to the resources mentioned in the bibliography and the Related Work 2. Further data quality dimensions might play a role, but would still not question the status of data completeness as one of the dimensions contributing to data quality. Newer studies in reference to the topic of data quality dimensions in relation to data completeness might have been published during the time of writing and after the extraction of the data sources for this piece of work.

7.2 RQ2: Techniques to handle data completeness

The goal of the second research question was to compare the theoretical (i.e. academic) via a literature review, and practical perspective of approaches by the means of an online survey to deal with missing values, and it was phrased in the following way:

RQ2: ***What methods and models are used in research and in the industry to inspect, evaluate, and improve the handling of missing values throughout the lifecycle?***

In line with the Related Work 2 and the Literature Review 4, it turned out that data science often requires dealing with incomplete data, and that datasets would only be considered up to a maximum of 50% of missing values, otherwise they would be too unreliable for further processing [52] [13]. This suggestion from the literature reviewed was later on also confirmed by the online survey, in which around 80% of the participants stated that they would not use the presented datasets with 40% and 70% of the data entries missing.

According to the Related Work 2 and the Literature Review 4, around 70% of the people involved in data preparation indicated that they had used missing value handling methods beyond the deletion of columns / rows [90], and mean imputation as well as regression imputation were preferred in a practical setting [90] [31] [98]. However, the theoretical view recommends statistical and machine learning based techniques in order to aim for even higher performance advances [90] [52] [73]. The proposition regarding the usage of handling methods was again largely confirmed by the online survey, in which approximately 70-80% of the participants said that they engage in more than the deletion of columns / rows with regard to datasets that contain up to 20% missing values, for which they think about a dedicated strategy (see Figure 5.14 and 5.13).

The Literature Review 4 also indicated an interest for using increasingly machine-learning based and statistical imputation methods beyond the simple ones from the theoretical perspective (from 2010 to 2021 as depicted in Fig. 2.5), in view of those methods' performance advantages [36] [98] [31]. Still, in the publications it was acknowledged that such comprehensive methods could lead to higher computation effort and longer execution times in the missing data imputation process [36] [31]. According to previous literature surveys by Young (2011) and Franca (2021), practitioners rarely used more advanced statistical or machine learning based techniques, notably due to the availability of easily usable software, execution time (notably for some machine learning based methods), and the "time as well as statistical knowledge" necessary for implementing and comparing outcomes [98] [26].

In the Literature Review 4, Hasan (2021) revealed that (from 2010 to 2021) that among the machine learning based imputation methods in a theoretical setting, the most frequently used method by far was kNN, followed by RF, SVM, BPCA, and DT (see Figure 2.7) [36]. In very few publications, data was imputed using Naive Bayes, self-organizing maps, SVR, LDA, (linear) regression, XGBoost, or ANN [36]. The Related Work 2 and Literature Review 4 showed that in research publications, all kinds of missing rates have been analyzed, but they usually focus on one of three ranges of missing rates, namely below 30%, 30% to 50%, or over 50% [2]. Among the models well suited to industrial data, kNN, SVM, mean replacement, along with simpler statistical methods for miss rate percentages lower than approximately 30 to 40% have been mentioned, while over 40% the features are usually purged [11] [89] [22].

In extension to research publications, this thesis newly contributes beyond the state of the art by adding insights from practitioners on how high missing data rates could be so that the dataset stays still relevant for further analysis, as well as on the use of specific handling methods, software packages and libraries to deal with missing data, to the observations made by Young (2011) [98], Gond(2021) [31], Hasan (2021) [36], Feldman (2018) [23], and Alamoodi (2021) [2], amongst others.

The 330 online survey participants largely had completed a bachelor or master degree, with over 60% of them in statistics, data science and computer science, and around 40% working in data analysis, data science or a statistics related job (see 5.2, 5.3). Approximately 70% indicated that they had more than 3 years of work experience in their current job. What was previously not covered by the literature was the usage of programming languages amongst others for the purpose of handling data completeness, which turned out to be 51% Python and 27% R, according to the online survey (see 5.10). Row / column deletion or complete case analysis was not much considered by the previous publications, but was included in the online survey, which effectively revealed that in a scenario with 5% missing entries, around 40% delete the missing observations or features as a whole from the dataset (see 5.11). 58% apply imputation techniques, but around half of them apply very simple methods such as constant replacement (incl. mean / mode / median imputation) or using always the same handling (see 5.11). As a large amount of publications deal with more complex methods, the online survey results indicated that notably in scenarios with rather low missing rates, practitioners might not choose the use more elaborate methods, but rather favor simple ones.

However, when presented with a dataset containing 20% missing entries, 44% stated that they would think about a tailored and more elaborate imputation strategy, while only 15% would opt for row / column deletion, and 14% for constant replacement (see 5.12). For 18%, the dataset would even have to large of a missing rate, so they would not use it at all (see 5.12). This is also a fact that has not been tackled by publications that even in such rather low missing rate scenarios, a considerable number of practitioners would refrain from using the dataset. For the larger two analyzed missing rate scenarios, the online survey results confirmed the previous publications' findings, saying that most practitioners would not use such datasets. In the 40% and 70% missing rate scenario, 80% of the participants said they would not further process such a dataset, whereas only about 15% would try to deal with it (see 5.14 and 5.13).

Young (2011) [98], Gond(2021) [31], Hasan (2021) [36], Feldman (2018) [23], and Alamoodi (2021) [2] amongst others promoted the use of more advanced techniques for handling missing values. However, in practice the organizations' approaches diverge substantially from that ideal in a way that only 33% of the survey respondents said that they developed dedicated strategies to handle missing values for every dataset. The remaining 67% of the respondents either used complete case analysis, did not consider incomplete datasets at all, replaced by constant, or used only one single method of choice for every dataset (see 5.15. Hence, this elucidated the divergence between theory and practice once more. In a theoretical sense, all options for more comprehensive strategies that could enhance data quality overall would be available, but organizations do apparently not seamlessly put them into practice for various reasons. One of those reasons might be the fact that more than 50% of the organizations perceive handling incomplete data somewhat important or indifferent, but not very important, and yet over 25% do not consider it important (see 5.16. This could actually hint towards the need for raising awareness of the topic.

In the literature, missing data handling methods such as CART, EM, hot-deck, kNN, mean / mode / median / constant replacement, RF, SVM, ANN, association rules, LOCF, multiple imputation, Naive Bayes, PCA, regression, and XGBoost have often been mentioned as commonly used tools to impute missing data entries [20] [36] [21] [66]. However, practitioners had a different view of that. Even when considering only respondents whose current job role is placed in the area of data analysis / data science / statistics, 75% actively use the mean / median / mode replacement and 50% the constant replacement methods in practice, which is the highest usage among all missing data handling methods mentioned in the online survey (see 5.19, 5.20). When it comes to specific preferences for methods, mean / median / mode replacement was mentioned as the top choice (by 22% of respondents), followed by regression and kNN (by 8% each) (see 5.21). More comprehensive machine learning methods turned out to be not too popular (see 5.21). This was followed by only around 30% active usage of kNN and regression respectively. Approximately 10% of the participants had used CART, RF, SVM, LOCF, multiple imputation, and PCA techniques in context with missing value handling (see 5.17, 5.18). The least used methods by practitioners for dealing with missing values from the ones given in the survey are hot-deck, EM, ANN, association rules, Naive Bayes, and XGBoost. Still, to over 20% of the respondents, the methods CART, kNN, RF, SVM, ANN, Naive Bayes, PCA, and regression are known from another context where they used it, but they have not used it in practice for missing value handling (see 5.17, 5.18). This only somewhat reflects the development of the usage of missing data handling methods in publications as Hasan (2021) summarized that in a theoretical setting, the most frequently used method by far was kNN, followed by RF, SVM, BPCA, and DT (see Fig. 2.7) [36]. Still, concerning the least frequently used methods, theory and practice corresponded which comprise Naive Bayes, SVR, LDA, (linear) regression, XGBoost, or artificial neural networks [36].

The usage of software libraries and packages in relation to handling incomplete data has so far not been covered by any survey or publication. From the online survey results, it can be seen that, even if the view is narrowed to participants who work in data science related jobs, only a selection of libraries such as the various imputers and regressors from Python or `mice`, `VIM` and `Hmisc` from R are used actively in practice by a maximum of 25%. To a large number of participants (i.e. ranging from 50% to 80% per library), the library is known to them, but not actively used in practice (see 5.20). In contrast to this, research publications often refer to specific software libraries and suggest them for handling missing data in practice such as those by Erhan (2021) [20], Farhangfar (2008) [21], or Okafor (2021) [66].

Finally, the question of why certain methods are preferred in practice, and others are not much used, is raised. The participants' aggregated answer to what influences their choice of missing value treatment method most is quite revealing. The most important aspect to them is indeed that a method is fast and easy to implement, and secondly they closely follow recommendations by their workplace (see 5.24). While the availability of tutorials related to methods and whether the method was already used or taught during their education was even more important than reading research papers about the suitability of methods and their application (see 5.24). In line with the final comments section, some participants added their thoughts in the form of open text. A large number of the comments pointed out that missing values are an issue in practice that should ideally obtain more attention, but is in reality still not taken as serious by management or due to time constraints at work in many cases. Careful study of research papers and available options would be necessary to examine the difference in performance, however, such a scenario appears to be rather an ideal case, and not directly transferable to everyday life where time and resource constraints occur. The respondents also mentioned that it would be recommendable to not even let missing values occur, i.e. tackle the issue earlier at the root already.

What might limit the findings related to this research question is that the literature review is based on a specific search string which might could have overlooked relevant publications. The search was limited to the publications databases IEEE Xplore, ACM Digital Library, Web of Science, and Scopus, and to the time range from 1999 to 2022 which could have left out relevant publications. Even in 2022, further relevant publications might become available after the time of extraction or writing. Concerning the online survey, it was limited in time to a duration of around one month when answers were accepted. In case the online survey submission period would have been longer, a higher number of answers might have been generated. The list of missing value handling methods and software libraries suggested for evaluation did not represent an exhaustive list, which could have led to it leaving out relevant options. In addition to this, a representation bias or response bias might have occurred among the respondents, as the survey participants were sourced mainly via personal contacts and a subsequent snowballing effect. The online survey finally also bears the risks that its results might not be generalizable to the whole population.

7.3 RQ3: Open-source and industry use cases on missing value imputation

The third research question refers to two case study comprising statistical experiments to apply missing value handling methods, and was formulated as follows:

RQ3: ***Which methods of handling incomplete data are most suitable with a view to enhancing data quality based on the selected data sets?***

The overall RQ3 is split into two parts, RQ3a and RQ3b, which refer to the open-source case study and the industry case study (with the latter considering manufacturing sensors data) to determine which methods for missing data handling are most suitable, and how they compare across the case studies.

As it was mentioned in the Related Work 2, the framework from Hasan (2021) for the execution of statistical experiments was applied in line with both case studies focusing on indirect evaluation, extended by the direct evaluation phase, and runtime measurement as performance metrics to measure performance from various angles, which was depicted in Figure 6.2 [36]. Indeed, the number of publications using direct evaluation has surged over the last decade, and the usage of indirect evaluation has stayed stable (see 2.8) [36]. A similar experiment set-up like Hasan's (2021) had previously also been used in publications by Alamoodi (2021), Syafie (2018) and Dhungana (2021) suggested a similar experiment design [2] [84] [16]. In the Literature Review 4, Hasan (2021) revealed that most research studies either concentrated on the direct or the indirect evaluation, but did not apply both [36]. Thus, in this thesis, both performance evaluation techniques and additionally the recording of execution time as a performance metric was chosen in extension of the state of the art evaluation approaches. According to Hasan (2021), 60% to 70% of the studies concentrated on simple statistical methods for imputation [36]. In view of the above, this thesis aimed to incorporate in publications much used (e.g. kNN, RF, DT) and not so frequently used (e.g. linear regression, XGBoost) machine learning based imputation methods [36] (see also 2.7.

Research studies concentrate usually focus on one of three ranges of missing rates, namely below 30%, 30% to 50%, or over 50% [2]. In this thesis, the missing rates of 5%, 20%, 40%, and 70% were already used in the online survey, where the participants largely voted for not using the dataset with 70% of entries missing. For this reason, the missing rates were narrowed down to 5%, 10%, 20%, 40% (assuming the MCAR missingness mechanism) to obtain more fine-grained insights into the performance metrics for those four scenarios.

The most used performance metric mentioned in publications that were screened in line with the Related Work 2 for regression settings was RMSE, sometimes in combination with imputation execution time [16] [2] [36]. To extend to the state of the art, the performance metrics RMSE, MAE, and runtime were combined in this thesis for a comprehensive evaluation.

As for the choice of missing data handling methods, the goal of this thesis was to bring together methods frequently used in practice as well as methods suggested in research publications. Thus, both simpler techniques like mean / median / constant replacement and linear regression as well as machine learning techniques like kNN, RF, CART (i.e. DT), and XGBoost were applied. Overall, the statistical experiments aimed to evaluate various missing value imputation methods in the direct and indirect evaluation setting plus the execution time, and with varying missing value rates. The results from both case studies are compared subsequently in order to review differences and similarities in the scenarios. Ehrlinger (2018) analyzed the treatment of missing values in the area of Industry 4.0 analytics, from which emerged that most research on missing values did not refer to industrial analytics, but rather to social science analytics concentrating on survey data [19]. In view of the above, this thesis aimed to gain further insights into missing value treatment in the industrial manufacturing domain using sensors data and to compare to another domain's dataset.

For the implementation, the choice of R software libraries and packages reflected those indicated by the online survey participants as being actively used in their job such as `VIM`, `mice`, `mi`, and `randomForest`. For every imputation method and share of missing values, 20 runs were carried out so that enough performance result data for statistical significance testing in a paired setting (by the means of paired t-tests and Paired Samples Wilcoxon Signed-Rank tests) was collected in order to extend the state of the art studies by the significance analysis.

When it comes to the comparison of results of the two case studies, the direct evaluation revealed similarities between the best and worst methods across all missing data rates (see 6.26). The RF and kNN methods were the best methods, and linear regression consistently came out as worst (regarding the RMSE performance metric). This would also entail for practitioners who stated (linear) regression as one of their favorite methods that they might be well advised to re-think their preference, and consider some of the better performing methods, as even mean and median replacement perform better in the 20% and 40% missing rate scenarios than linear regression. In relation to execution time, the results are again similar across both case studies and for all missing data rates. The fastest methods concerning this criterion are mean and median replacement, whereas the most computation-intense ones are CART and RF (see 6.27). So, mean and median replacement could be an option if the most important criterion is a short runtime. However, if other criteria matter, the former are not recommended.

In line with the indirect evaluation, kNN and RF prevailed as best performing methods with regard to RMSE across all missing rates scenarios. In the 5% and 10% missing rate scenarios CART was among the best methods too, while in the 20% and 40% missing rate scenarios XGBoost was among the best. A similar picture is shown by the comparison of the MAE results, from which kNN and XGBoost emerge yet again as best methods. Mean and median replacement are the worst methods with regard to RMSE and MAE out of the tested ones. Furthermore, regression does not perform well either in the scenarios with higher missing rates.

Hence, for practitioners kNN and XGBoost could be a viable choice as they are among the methods regarding RMSE and MAE performance, and at the same time not among the worst concerning execution time. In fact, kNN turned out as a particularly favorable method as it has good RMSE, MAE, and runtime results, and even its runtime is not significantly different from the one of the mean replacement in the open-source case study (6.30). However, this finding might not be generalizable as there is a significant difference for the kNN method when using the industry data (6.31).

From both the industry as well as the open-source dataset results, it could be seen that the RMSE and MAE performance was significantly different from the mean imputation results for kNN, CART, RF, and XGBoost in the scenarios with a share of 20% and 40% data missing. This gives a hint towards better choices regarding the RMSE metric compared to the simple mean imputation baseline, and similarly "bad" method such as median and regression imputation. When the regression performance on the original dataset is taken as a baseline, a key finding was that kNN, CART, and RF imputation did not lead to significant performance loss in either one of the case studies in the 5% and 10% missing data rates scenarios. But in the higher missing rate scenarios, any imputation method still comes at a substantial performance disadvantage. For the MAE performance metric, the situation differed slightly as even for the 5% and 10% missing data rates scenarios only CART and XGBoost imputed datasets came close to the performance based on the original dataset in both case studies, and notably it was restricted to that in the industry case study. Overall, comparing the results of the statistical experiments demonstrated that the two datasets led to rather similar performance metrics findings despite the different constraints and dataset shapes considered for the analysis. In addition to this, the mean and median replacement methods that are preferred in practice next to regression are fast, but do not exhibit good RMSE and MAE performance metrics. Therefore, the trade-off between execution time and other performance metrics such as RMSE or MAE has to be critically reflected upon. Even though it might not have been the fastest algorithm, kNN imputation still offered a good performance fit for all cases and related to all performance metrics analyzed, while still keeping execution time reasonably low.

On the whole, it has to be mentioned at this point that the aforementioned findings also come with several limitations. Firstly, the experiment set-up was chosen in alignment to Hasan's (2021) framework [36]. Other frameworks could have led to different phases, processes, and outcomes. What is more, the performance comparison as well as the significance testing was based on 20 randomly selected runs with fixed seed values to control for the randomness and make the experiments reproducible. The choice of n as 20 runs could be changed and could lead to different results. The impact of the randomly selected seed values on the results was not analyzed further either. Furthermore, the statistical experiments looked at the outcomes of six missing data handling methods, even though there are many more available. So, another choice of methods might have led to a different outcome. Likewise, the SVR method was applied in line with the indirect evaluation which is only one method out of many that could have been chosen and compared. Even though it was applied with all the same parameters in all scenarios, the effect of another regression method was not examined. Finally, the experiments were carried out based on the two named datasets, which could constitute a too low number of datasets to obtain generalizable findings.

CHAPTER 8

Conclusion and Future Work

This thesis aimed to analyze and enhance data quality from the completeness dimension perspective, and with a focus on the manufacturing domain. As a sufficiently high data quality is crucial for the successful use of industrial applications for instance in CPPS engineering, tackling incomplete data in the most effective way possible, can essentially help to obtain more promising results [7] [37]. Even though complete data would be an ideal case to start data analysis, notably manufacturing data is in reality rarely complete and the reasons for lost data entries are manifold. Handling incomplete data means taking steps to eliminate the missing observations or features, or to choose methods to impute the missing entries. At this point, the goal was to find out what approaches practitioners in comparison to methods suggested in research publications choose in order to cope with missing data, and to subsequently analyze based on two case studies how performance metrics of several missing data handling methods compare. Similarities and differences between the results generated based on the dataset of the open-source as well as the manufacturing case study were inspected so that overlaps could be identified.

In line with the online survey, practitioners turned out to favor mean, median, and constant replacement as well as (linear) regression imputation to handle missing values, i.e. they had a preference for simple methods. From the segment of machine learning methods, which was not so much used among the online survey participants, the favored alternative was the kNN method (followed by a smaller number of people who used CART and RF) which was also quite well known among the practitioners. Besides, it turned out that ANN, association rules, LOCF, Naive Bayes, SVM, EM, and XGBoost were neither much used nor much known with regard to handling missing data among practitioners, even when the view was narrowed to participants who work in data science related jobs.

However, their choice also depends on the share of missing values in the dataset. For datasets with lower than 40% of the data entries missing, most practitioners stated that they would try to handle missing data, whereas for datasets with missing rates higher than 40% they would drop the whole dataset and not consider it further for analysis. Overall, the highest rated criterion for choosing a specific missing data handling method was that the method is fast and easy to implement, and second to this that the workplace gave a recommendation for a particular method. Some practitioners participating in the online survey also cited the availability of tutorials or previous experience with a method from their education as a decision criterion for a specific method. Reading research papers and their method presentation or conclusion section was, however, least frequently mentioned as a criterion to choose a particular missing data handling technique. The majority of practitioners mentioned that during their education either nothing regarding the topic occurred or they only learned one to two simple methods but no details. In view of this, they stated that they would see more thorough education about the topic as beneficial.

In the statistical experiments the mean replacement, median replacement, kNN, CART, RF, and XGBoost imputation methods, which were selected based on the Literature Review 4 and Online Survey 5.2 results, were applied to two datasets in line with the open-source case study and the industry case study. The thesis intended thereby to gain insights into how selected methods perform on different datasets in the direct, indirect evaluation, and runtime review. Thus, the statistical experiments were used to develop a "virtual sensor" (cf. "soft sensor") that evaluates the imputation techniques in order to determine whether the performance of subsequent machine learning methods can be influenced and improved by the choice of the missing value handling method, notably in the CPPS context.

From the direct evaluation, RF emerged as the best method across all missing value percentages analyzed (followed by kNN in the lower and mean / median replacement in the higher missing rate scenarios), while (linear) regression came out as least performant. With regard to execution time, the mean and median replacement methods achieved the best performance, while CART and RF took longest to execute.

In line with the indirect evaluation, the kNN method appeared to be the among the best performing based on RMSE for all missing data rates, while RF and CART performed well in three scenarios, and XGBoost in the scenarios with 20% and 40% data missing. The least performant methods with respect to the RMSE criterion were mean and median replacement. As for the MAE, the situation was much like the one using RMSE, as the kNN performed best in all scenarios again. But in contrast to the RMSE evaluation, regarding MAE XGBoost performed best in all scenarios as well. CART was among the best only in the scenarios with 10% and 40% missing data. In relation to both, the RMSE and MAE performance metrics, mean and median replacement came out as the least performing methods in all scenarios, which were complemented by the (linear) regression method in the 20% and 40% missing data rate scenario.

From the above analyses, it can be seen that the choice of missing data handling method can have an influence on the performance metrics set for the direct as well as the indirect evaluation. The indications for suitable and well performing methods were essentially very similar across the open-source and industry case study.

Future research on this topic should, however, consider more datasets or case studies than the two observed in this thesis so that the results could be better suited to generalize from them. Additionally, further recent research that was published after the time of extraction and writing should be integrated. The online survey comprised a limited number of participants due to time constraints. So, it would be advised to repeat the online survey over a longer time period in order to obtain more participants, and check for the reproducibility as well as stability of its results. As in this thesis, only a limited number of missing data handling method were used both in the online survey as well as in the statistical experiments, it is finally recommended to expand this in the future to a higher number of methods, evaluate more missing data rates, and create a more comprehensive and fine-grained scenario analysis, while also checking for the potential effect of the choice of random seed values.

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 9

ANN artificial neural network. 20, 22, 45, 68, 69, 71, 123–125, 131

AUC Area under the ROC (Receiver Operating Characteristic) curve. 17, 24

BPCA Bayesian principal component analysis. 24, 123, 125

CART classification and regression trees. 22, 46, 68, 69, 71, 86, 87, 93, 96–98, 104, 107–118, 124, 127, 128, 131, 132

CPPS Cyber-Physical Production Systems. 1, 4, 9–12, 31, 32, 46, 121, 131, 132

CV cross-validation. 17

DQ Data Quality. 12, 14, 15

DT decision tree. 24, 86, 123, 125–127

EM expectation-maximization. 19, 20, 22, 48, 52, 53, 68, 69, 71, 86, 124, 131

EMSE empirical software engineering. 29

GAN generative adversarial network. 50, 51

IoT Internet of Things. 14

IS information systems. 27

ISO International Organization for Standardization. 2, 3, 12, 13, 15, 29, 121

kNN k-nearest neighbor. 20, 22, 24, 25, 45–49, 51–55, 58, 69–71, 86, 87, 91–94, 96–98, 104, 107–118, 123–128, 131, 132

LDA linear discriminant analysis. 53, 86, 123, 125

LOCF last observation carried forward. 45, 46, 50, 68, 72, 86, 124, 131

MAE Mean Absolute Error. 17, 48, 49, 81, 88, 89, 93–95, 97, 98, 105, 109, 110, 112–114, 116, 118–120, 127, 128, 132, 136, 137, 139, 140

MAPE mean absolute percentage error. 48, 49

MAR Missing at Random. 18, 19, 50, 52

MCAR Missing Completely at Random. 18, 26, 50, 52, 53, 85, 103, 127

MICE multiple imputation by chained equations. 45, 55

ML machine learning. 23, 135

MLP multi-layered perceptron. 20

MNAR Missing Not at Random. 18, 50

MSE Mean Squared Error. 17, 50, 51

MVI missing value imputation. 23, 25

NFL No Free Lunch. 22

NNRW neural network with random weights. 55, 86

OLS ordinary least squares. 51

PCA principal component analysis. 47, 52, 68, 69, 71, 86, 124, 125

RBF radial basis function. 53, 54

RF random forest. 24, 86, 87, 91–94, 96–98, 104, 107–118, 123–128, 131, 132

RMAE Root Mean Absolute Error. 17

RMSE Root Mean Square Error. 17, 25, 48–53, 55, 81, 88, 89, 91–98, 104, 105, 107–113, 115–119, 127, 128, 132, 136, 137, 139, 140

RNN recurrent neural networks. 22

ROC receiver operating characteristic. 16, 24

RQ Research Questions. 5

SLR systematic literature review. 35

SMOTE synthetic minority oversampling technique. 46

SVD singular value decomposition. 47, 52

SVM support vector machine. 20, 22, 45, 46, 53, 54, 58, 68, 69, 86, 88, 105, 123–125, 131

SVR support-vector regression. 88, 93, 104–106, 111, 117–120, 123, 125, 129, 137

USD United States Dollar. 82

VAE variational autoencoder. 55

Bibliography

[1] MB Abhishek and Neelawar Shekar Vittal Shet. Data processing and deploying missing data algorithms to handle missing data in real time data of storage tank: A cyber physical perspective. In *2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, pages 1–6, 2019.

[2] A. Alamoodi, Bilal Bahaa, A. Zaidan, O.s Albahri, Juliana Chen, Mohammed Chyad, Salem Garfan, and Ahmed Aleesa. Machine learning-based imputation soft computing approach for large missing scale and non-reference data imputation. *Chaos, Solitons & Fractals*, 151:111–236, 10 2021.

[3] Rasim Alguliyev, Yadigar Imamverdiyev, and Lyudmila Sukhostat. Cyber-physical systems and their security issues. *Computers in Industry*, 100:212–223, 2018.

[4] Clemens Arbesser, Florian Spechtenhauser, Thomas Muhlbacher, and Harald Piringer. Visplause: Visual data quality assessment of many time series using plausibility checks. *IEEE Transactions on Visualization and Computer Graphics*, 23:641 – 650, 01 2016.

[5] Rachad Atat, Lingjia Liu, Jinsong Wu, Guangyu Li, Chunxuan Ye, and Yi Yang. Big data meet cyber-physical systems: A panoramic survey. *IEEE Access*, 6(1):73603–73636, 2018.

[6] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. Methodologies for data quality assessment and improvement. *ACM Comput. Surv.*, 41(3):1–52, July 2009.

[7] S. Biffl, M. Eckhart, A. Lüder, and E. Weippl. *Security and Quality in Cyber-Physical Systems Engineering*. Springer Publishing, Cham (Switzerland), 2019.

[8] Stefan Biffl, Arndt Lüder, and Detlef Gerhard. *Multi-disciplinary engineering for cyber-physical production systems: Data models and software solutions for handling complex engineering projects*. Springer, 2017.

[9] Wahyu Bowo, Agus Suhanto, Meisuchi Naisuty, Syukron Ma'mun, Achmad Hidayanto, and Ika Habsari. Data quality assessment: A case study of pt jas using tdqm framework. *2019 Fourth International Conference on Informatics and Computing (ICIC)*, pages 1–6, 10 2019.

[10] Li Cai and Yangyong Zhu. The challenges of data quality and data quality assessment in the big data era. *Data Science Journal*, 14:1–10, 05 2015.

[11] Caoimhe Carbery, Roger Woods, Cormac McAteer, and David M Ferguson. Missingness analysis of manufacturing systems: A case study. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, pages 1–12, 2022.

[12] M.H. Cartwright, M.J. Shepperd, and Q. Song. Dealing with missing software project data. In *Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717)*, pages 154–165, 2003.

[13] Y.S. Chuo, J.W. Lee, and C.H. Mun. Artificial intelligence enabled smart machining and machine tools. *Journal of Mechanical Science and Technology*, 36:1–23, 01 2022.

[14] Corinna Cichy and Stefan Rass. An overview of data quality frameworks. *IEEE Access*, pages 24634–24648, 02 2019.

[15] David Corrales, Juan Corrales, and Agapito Ledezma Espino. How to address the data quality issues in regression models: A guided process for data cleaning. *Symmetry*, 10:1–20, 04 2018.

[16] Hariom Dhungana, Francesco Bellotti, Riccardo Berta, and Alessandro Gloria. Performance comparison of imputation methods in building energy data sets. *Applications in Electronics Pervading Industry, Environment and Society*, pages 144–151, 01 2021.

[17] Pablo Drumond, Daniele Kappes, Cássio Moraes, Eduardo Oliveira, and Mariana Teixeira. Soft sensor: Traditional machine learning or deep learning. pages 231–242, 10 2018.

[18] Ashley Edelen and Wesley Ingwersen. The creation, management, and use of data quality information for life cycle assessment. *The International Journal of Life Cycle Assessment*, 23:759–772, 07 2017.

[19] Lisa Ehrlinger, Thomas Grubinger, Bence Varga, Mario Pichler, Thomas Natschläger, and Jürgen Zeindl. Treating missing data in industrial data analytics. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pages 148–155, 2018.

[20] Laura Erhan, Mario Di Mauro, Ashiq Anjum, Ovidiu Bagdasar, Wei Song, and Antonio Liotta. Embedded data imputation for environmental intelligent sensing: A case study. *SENSORS*, 21(23), DEC 2021.

[21] Alireza Farhangfar, Lukasz Kurgan, and Jennifer Dy. Impact of imputation of missing values on classification error for discrete data. *PATTERN RECOGNITION*, 41(12):3692–3705, DEC 2008.

[22] Alireza Farhangfar, Lukasz A. Kurgan, and Witold Pedrycz. A novel framework for imputation of missing values in databases. *IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS PART A-SYSTEMS AND HUMANS*, 37(5):692–709, SEP 2007.

[23] Michael Feldman, Adir Even, and Yisrael Parmet. A methodology for quantifying the effect of missing data on decision quality in classification problems. *Communications in Statistics - Theory and Methods*, 47:1–52, 01 2017.

[24] Donatella Firmani, Massimo Mecella, Monica Scannapieco, and Carlo Batini. On the meaningfulness of "big data quality" (invited paper). *Data Science and Engineering*, 1(1):6–20, 2016.

[25] Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo, Maria G Xibilia, et al. *Soft sensors for monitoring and control of industrial processes*, volume 22. Springer, 2007.

[26] Cinthya França, Rodrigo Couto, and Pedro Braconnot Velloso. Missing data imputation in internet of things gateways. *Information*, 12:425–447, 10 2021.

[27] Salvador García, Sergio Ramírez-Gallego, Julián Luengo, José Benítez, and Francisco Herrera. Big data preprocessing: methods and prospects. *Big Data Analytics*, 1:1–22, 11 2016.

[28] Milot Gashi, Patrick Ofner, Helmut Ennsbrunner, and Stefan Thalmann. Dealing with missing usage data in defect prediction: A case study of a welding supplier. *Computers in Industry*, 132:1–10, November 2021.

[29] Mouzhi Ge, Stanislav Chren, Bruno Rossi, and Tomáš Pitner. Data quality management framework for smart grid systems. *22nd International Conference on Business Information Systems (BIS2019)*, pages 1–13, 06 2019.

[30] Karina Gibert, Miquel Sànchez-Marrè, and Joaquín Izquierdo. A survey on preprocessing techniques: Relevant issues in the context of environmental data mining. *AI Communications*, 29:1–37, 11 2016.

[31] Vikesh Kumar Gond, Aditya Dubey, and Akhtar Rasool. A survey of machine learning-based approaches for missing value imputation. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1–8, 2021.

[32] Aurora Gonzalez-Vidal, Punit Rathore, Aravinda S. Rao, Jose Mendoza-Bernal, Marimuthu Palaniswami, and Antonio F. Skarmeta-Gomez. Missing data imputation with bayesian maximum entropy for internet of things applications. *IEEE INTERNET OF THINGS JOURNAL*, 8(21):16108–16120, NOV 1 2021.

[33] Robert Groves, Floyd Fowler, and Coupter Mick. Survey methodology. *Wiley & Sons Publications (2nd Edition)*, 2009.

[34] Lisa Günther, Eduardo Colangelo, Hans-Hermann Wiendahl, and Christian Bauer. Data quality assessment for improved decision-making: a methodology for small and medium-sized enterprises. *Procedia Manufacturing*, 29:583–591, 01 2019.

[35] Muhammad H., Nur Dalila K.A., Noorita Tahir, Zatul Iffah Abd Latiff, Jusoh Huzaimy, and Akimasa Yoshikawa. Missing data imputation of magdas-9's ground electromagnetism with supervised machine learning and conventional statistical analysis models. *Alexandria Engineering Journal*, 61:1–11, 06 2021.

[36] Md Hasan, Md. Ashraful Alam, Shidhartho Roy, Aishwariya Dutta, Jawad Md, and Sunanda Das. Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021). *Informatics in Medicine Unlocked*, 27:1–24, 11 2021.

[37] Q. Peter He and Jin Wang. Statistical process monitoring as a big data analytics tool for smart manufacturing. *Journal of Process Control*, 67:35–43, 2018. Big Data: Data Science for Process Control and Operations.

[38] Bernd Heinrich, Diana Hristova, Mathias Klier, Alexander Schiller, and Michael Szubartowicz. Requirements for data quality metrics. *Journal of Data and Information Quality*, 9:1–32, 01 2018.

[39] Alan Hevner, Alan R, Salvatore March, Salvatore T, Park, Jinsoo Park, Ram, and Sudha. Design science in information systems research. *Management Information Systems Quarterly*, 28:75–106, 03 2004.

[40] Azira Ibrahim, Ibrahim Mohamed, and Nurhizam Safie. Factors influencing master data quality: A systematic review. *International Journal of Advanced Computer Science and Applications*, 12:1–12, 01 2021.

[41] Syed Imtiaz and Sirish Shah. Treatment of missing values in process data analysis. *The Canadian Journal of Chemical Engineering*, 86:838 – 858, 10 2008.

[42] ISO. Iso-iec 25012 - data quality. `https://www.iso.org/standard/35736.html`, 2011.

[43] Vimukthi Jayawardene, Shazia Sadiq, and Marta Indulska. The curse of dimensionality in data quality. *Proceedings of the 24th Australasian Conference on Information Systems*, 01 2013.

[44] Suraj Juddoo and Carlisle George. A qualitative assessment of machine learning support for detecting data completeness and accuracy issues to improve data analytics in big data for the healthcare industry. In *2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM)*, 11 2020.

[45] Suraj Juddoo, Carlisle George, Penny Duquenoy, and David Windridge. Data governance in the health industry: Investigating data quality dimensions within a big data context. *Applied System Innovation*, 1:1–16, 11 2018.

[46] A. Juneja and N. N. Das. Big data quality framework: Pre-processing data in weather monitoring application. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 559–563, 2019.

[47] I. Kadence. What is survey design? *Kadence International*.

[48] Kalinka Kaloyanova, Ina Naydenova, and Zlatinka Covacheva. Addressing data quality in healthcare. In *CEUR Workshop Proceedings*, volume 1, pages 155–164, 05 2021.

[49] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7–15, 2009.

[50] A. Kramer, Sebastian Gaulocher, M. Martins, and Laurindo Leal Filho. Surface tension measurement for optimization of flotation control. *Procedia Engineering*, 46:111–118, 12 2012.

[51] Moh Abdul Latief, Alhadi Bustamam, and Titin Siswantining. Performance evaluation xgboost in handling missing value on classification of hepatocellular carcinoma gene expression data. In *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–6, 2020.

[52] Chia-Yen Lee and Chen-Fu Chien. Pitfalls and protocols of data science in manufacturing practice. *Journal of Intelligent Manufacturing*, pages 1–19, 11 2020.

[53] Jay Lee, Chao Jin, and Zongchang Liu. *Predictive Big Data Analytics and Cyber Physical Systems for TES Systems*, pages 97–112. Springer International Publishing, Cham, 2017.

[54] JuneHyuck Lee, Sang Do Noh, Hyun-Jung Kim, and Yong-Shin Kang. Implementation of cyber-physical production systems for quality prediction and operation control in metal casting. *Sensors*, 18(5):1–17, 2018.

[55] W. Lin and Chih-Fong Tsai. Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, 53:1487–1509, 2019.

[56] Caihua Liu, Patrick Nitschke, Susan Williams, and Didar Zowghi. Data quality and the internet of things. *Computing*, 102:573–599, 02 2020.

[57] Yuehua Liu, Tharam Dillon, Wenjin Yu, Wenny Rahayu, and Fahed Mostafa. Missing value imputation for industrial iot sensor data with large gaps. *IEEE INTERNET OF THINGS JOURNAL*, 7(8):6855–6867, AUG 2020.

[58] Panagiotis Loukopoulos, George Zolkiewski, Ian Bennett, Pericles Pilidis, Fang Duan, and David Mba. Dealing with missing data as it pertains of e-maintenance. *Journal of Quality in Maintenance Engineering*, 23:1–24, 07 2017.

[59] Ana Lucas. Corporate data quality management: From theory to practice. In *5th Iberian conference on information systems and technologies (CISTI), 2010*, pages 1 – 7, 07 2010.

[60] Zheng Lv, Jun Zhao, and Ying Liu. Data imputation for gas flow data in steel industry based on non-equal-length granules correlation coefficient. *Information Sciences*, 367:311–323, 06 2016.

[61] Paul McMahon, Tieling Zhang, and Richard A. Dwight. Approaches to dealing with missing data in railway asset management. *IEEE Access*, 8:48177–48194, 2020.

[62] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda. Cyber-physical systems in manufacturing. *CIRP Annals*, 65(2):621–641, 2016.

[63] Phu H. Nguyen, Sagar Sen, Nicolas Jourdan, Beatriz Cassoli, Per Myrseth, Mikel Armendia, and Odd Myklebust. Software engineering and ai for data quality in cyber- physical systems - sea4dq'21 workshop report. *SIGSOFT Softw. Eng. Notes*, 47(1):26–29, jan 2022.

[64] Anastasija Nikiforova. Open data quality. *CEUR Workshop Proceedings*, pages 1–10, 08 2018.

[65] Heru Nugroho and Kridanto Surendro. Missing data problem in predictive analytics. In *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, ICSCA '19, page 95–100, New York, NY, USA, 2019. ACM - Association for Computing Machinery.

[66] Nwamaka U. Okafor and Declan T. Delaney. Missing data imputation on iot sensor networks: Implications for on-site sensor calibration. *IEEE SENSORS JOURNAL*, 21(20):22833–22845, OCT 15 2021.

[67] Ajaya Kumar Pani, Krunal G Amin, and Hare Krishna Mohanta. Data driven soft sensor of a cement mill using generalized regression neural network. In *2012 International Conference on Data Science Engineering (ICDSE)*, pages 98–102, 2012.

[68] Baba Piprani and Denise Ernst. A model for data quality assessment. In *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*, pages 750–759, 11 2008.

[69] Dmitriy P. Plakhotnikov and Elena E. Kotova. Design and analysis of cyber-physical systems. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, number 1, pages 589–593, 2021.

[70] Reinhold Ploesch. Design science for software & systems engineering. In *Institute of Business Informatics at JKU - Johannes Kepler Universität Linz (Austria)*, pages 1–45, 2015.

[71] Kumar Rahul and R K Banyal. Data cleaning mechanism for big data and cloud computing. In *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 195–198, 2019.

[72] Theofanis Raptis, Andrea Passarella, and Marco Conti. Data management in industry 4.0: State of the art and open challenges. *IEEE Access*, 1:1–43, 07 2019.

[73] Adil Rasheed, Omer San, and Trond Kvamsdal. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*, 8:21980–22012, 2020.

[74] Roozbeh Razavi-Far, Ehsan Hallaji, Maryam Farajzadeh-Zanjani, Ranim Aljoudi, and Mehrdad Saif. A critical study on the impact of missing data imputation for classifying intrusions in cyber-physical water systems. In *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6, 2021.

[75] Kenneth Rothman. Curbing type i and type ii errors. *European Journal of Epidemiology*, 25:223–224, 03 2010.

[76] Aulia Deandra Rusdah and Hendri Murfi. Xgboost in handling missing values for life insurance risk prediction. *SN Applied Sciences*, 2:1336–1348, 08 2020.

[77] Ambika Sadhu, Rishabh Soni, and Mridul Mishra. Pattern-based comparative analysis of techniques for missing value imputation. In *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pages 513–518, 2020.

[78] Sunghyun Sim, Hyerim Bae, and Yulim Choi. Likelihood-based multiple imputation by event chain methodology for repair of imperfect event logs with missing data. In *2019 International Conference on Process Mining (ICPM)*, pages 9–16, 2019.

[79] Harpreet Singh. Big data, industry 4.0 and cyber-physical systems integration: A smart industry context. *Materials Today: Proceedings*, 46:157–162, 2021. 2nd International Conference on Manufacturing Material Science and Engineering.

[80] R Sivakani and Gufran Ahmad Ansari. Imputation using machine learning techniques. In *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pages 1–6, 2020.

[81] Shaoxu Song and Aoqian Zhang. Iot data quality. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 3517–3518, New York, NY, USA, 2020. Association for Computing Machinery.

[82] V Sowmya and N Kayarvizhy. An efficient missing data imputation model on numerical data. In *2021 2nd Global Conference for Advancement in Technology (GCAT)*, pages 1–8, 2021.

[83] Aditya Sundararajan and Arif Sarwat. *Evaluation of Missing Data Imputation Methods for an Enhanced Distributed PV Generation Prediction*, pages 590–609. 01 2020.

[84] Lukman Syafie, Fitriyani Umar, Aliyazid Mude, Herdianti Darwis, Herman, and Harlinda. Missing data handling using the naive bayes logarithm (nbl) formula. In *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, pages 318–321, 2018.

[85] Ikbal Taleb, Hadeel El Kassabi, Mohamed Serhani, Rachida Dssouli, and Chafik Bouhaddioui. Big data quality: A quality dimensions evaluation. In *The 2016 Second IEEE International Conference on Cloud and Big Data Computing (CBDCom 2016, 18-21 July, 2016, Toulouse, France)*, pages 1–8, 07 2016.

[86] John R. Venable, Jan Pries-Heje, and Richard Baskerville. Choosing a design science research methodology. In *ACIS Proceedings - Australasian Conference on Information Systems*, pages 1–12, 2017.

[87] Chunzhi Wang, Natalya Shakhovska, Anatoliy Sachenko, and Myroslav Komar. A new approach for missing data imputation in big data interface. *Information Technology And Control*, 49:541–555, 12 2020.

[88] Huan Wang, Yibin Chen, Bingyang Shen, Di Wu, and Xiaojuan Ban. Generative adversarial networks imputation for high rate missing values. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 586–590, 2018.

[89] Yanxia Wang, Kang Li, Shaojun Gan, and Che Cameron. Missing data imputation with ols-based autoencoder for intelligent manufacturing. *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS*, 55(6, 2):7219–7229, NOV-DEC 2019.

[90] T. Kirk White, Jerome P. Reiter, and Amil Petrin. Imputation in U.S. Manufacturing Data and Its Implications for Productivity Dispersion. *The Review of Economics and Statistics*, 100(3):502–509, 07 2018.

[91] Roel Wieringa. *Design Science Methodology for Information Systems and Software Engineering - Springer-Verlag Berlin Heidelberg 2014*. 01 2014.

[92] Diamond Williams and Herman Tang. Data quality management for industry 4.0: A survey. *Software Quality Professional - Milwaukee Bd. 22*, pages 26–35, 03 2020.

[93] Shu Xu, Bo Lu, Michael Baldea, Thomas Edgar, Willy Wojsznis, Terry Blevins, and Mark Nixon. Data cleaning in the process industries. *Reviews in Chemical Engineering*, 31:453–490, 10 2015.

[94] Qishan Yang, Mouzhi Ge, and Markus Helfert. Guidelines of data quality issues for data integration in the context of the tpc-di benchmark. In *The International Conference on Enterprise Information System (ICEIS), 26-29 Apr, 2017, Porto, Portugal*, pages 1–10, 12 2017.

[95] Zoujing Yao and Chunhui Zhao. Figan: A missing industrial data imputation method customized for soft sensor application. *IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING*.

[96] Yumeng Ye, Bingyi Zhong, Sapna Srimal, Awaad Alsarkhi, and John Talburt. A study on the impact of missing values in probabilistic matching. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 158–163, 2018.

[97] Yi You, Yong Hu, and Siqi Bu. Pmu data issues and countermeasure techniques in cyber-physical power systems: A survey. In *2021 IEEE Sustainable Power and Energy Conference (iSPEC)*, volume 1, pages 4278–4285, 2021.

[98] W. Young, G. Weckman, and W. Holland. A survey of methodologies for the treatment of missing values within datasets: limitations and benefits. *Theoretical Issues in Ergonomics Science*, 12(1):15–43, 2011.

[99] Lina Zhang, Dongwon Jeong, and Sukhoon Lee. Data quality management in the internet of things. *Sensors (Basel, Switzerland)*, 21:1–21, 08 2021.

[100] XiaoChi Zhang, Lars Kuchinke, Marcella L. Woud, Julia Velten, and Jürgen Margraf. Survey method matters: Online/offline questionnaires and face-to-face or telephone interviews differ. *Computers in Human Behavior*, 71:172–180, 2017.

[101] Ming Zhu and Xingbing Cheng. Iterative knn imputation based on gra for missing values in tplms. In *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, volume 01, pages 94–99, 2015.

Appendix

Initial List of Publications

001 R. Razavi-Far, E. Hallaji, M. Farajzadeh-Zanjani, R. Aljoudi and M. Saif. 2021. A Critical Study on the Impact of Missing Data Imputation for Classifying Intrusions in Cyber-Physical Water Systems. In *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*. pp. 1-6, doi: 10.1109/IECON48115.2021.9589513.

002 H. Zhou, K. -M. Yu, M. -G. Lee and C. -C. Han. 208. The Application of Last Observation Carried Forward Method for Missing Data Estimation in the Context of Industrial Wireless Sensor Networks. In *2018 IEEE Asia-Pacific Conference on Antennas and Propagation (APCAP)*. pp. 1-2, doi: 10.1109/APCAP.2018.8538147.

003 Y. Ye, B. Zhong, S. Srimal, A. Alsarkhi and J. Talburt. 2018. A Study on the Impact of Missing Values in Probabilistic Matching. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. pp. 158-163, doi: 10.1109/CSCI46756.2018.00038.

004 R.-H. Chen and C.-M. Fan. 2012. Treatment of missing values for association rule-based tool commonality analysis in semiconductor manufacturing. In *2012 IEEE International Conference on Automation Science and Engineering (CASE)*. pp. 886-891, doi: 10.1109/CoASE.2012.6386510.

005 R. Chen and C. Fan. 2012. Markov-chain based missing value estimation method for tool commonality analysis in semiconductor manufacturing. 2012. Proceedings of the 2012 Winter Simulation Conference (WSC). pp. 1-12, doi: 10.1109/WSC.2012.6465277.

006 M. Zhu and Xingbing Cheng. 2015. Iterative KNN imputation based on GRA for missing values in TPLMS. In *4th International Conference on Computer Science and Network Technology (ICCSNT)*. pp. 94-99, doi: 10.1109/ICCSNT.2015.7490714.

007 S. Juddoo and C. George. 2020. A Qualitative Assessment of Machine Learning Support for Detecting Data Completeness and Accuracy Issues to Improve Data Analytics in Big Data for the Healthcare Industry. In *2020 3rd International*

Conference on Emerging Trends in Electrical, Electronic and Communication s Engineering (ELECOM). pp. 58-66, doi: 10.1109/ELECOM49001.2020.929 7009.

008 X. Song, Y. Guo, N. Li and J. Liu. 2021. Dynamic Missing Data Recovery Method with Low Complexity in Internet of Things. In *2021 7th International Conference on Computer and Communications (ICCC)*. pp. 2091-2095, doi: 10.1109/ICCC54389.2021.9674265.

009 M. Abhishek and N. S. V. Shet. 2019. Data Processing and deploying missing data algorithms to handle missing data in real time data of storage tank: A Cyber Physical Perspective. In *2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*. pp. 1-6, doi: 10.1109/ICECIE47765.2019.8974816.

010 H. Kaneyasu, D. Nobayashi, K. Tsukamoto, T. Ikenaga and M. Lee. 2022. Data Completeness-aware Transmission Control for Large Spatio-Temporal Data Retention. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*. pp. 1-5, doi: 10.1109/ICCE53296.2022.9730495.

011 H. Wang, Y. Chen, B. Shen, D. Wu and X. Ban. 2018. Generative Adversarial Networks Imputation for High Rate Missing Values. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018*. pp. 586-590, doi: 10.1109/Cybermatics_2018.2018.00121.

012 L. Ehrlinger, T. Grubinger, B. Varga, M. Pichler, T. Natschläger and J. Zeindl. 2018. Treating Missing Data in Industrial Data Analytics. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*. pp. 148-155, doi: 10.1109/ICDIM.2018.8846984.

013 P. Fountas and K. Kolomvatsos. 2020. A Continuous Data Imputation Mechanism based on Streams Correlation. In *2020 IEEE Symposium on Computers and Communications (ISCC)*. pp. 1-6, doi: 10.1109/ISCC50000.2020.9219548.

014 K. Rahul and R. K. Banyal. 2019. Data Cleaning Mechanism for Big Data and Cloud Computing. In *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*. pp. 195-198.

015 V. H. Umathe and G. Chaudhary. 2015. Imputation methods for incomplete data. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. pp. 1-4, doi: 10.1109/ICIIECS.2015.7193063.

016 A. K. Pani and H. K. Mohanta. 2013. A hybrid soft sensing approach of a cement mill using principal component analysis and artificial neural networks. In *2013*

3rd IEEE International Advance Computing Conference (IACC). pp. 713-718, doi: 10.1109/IAdCC.2013.6514314.

017 A. K. Pani and H. K. Mohanta. 2013. A hybrid soft sensing approach of a cement mill using principal component analysis and artificial neural networks. In *2013 3rd IEEE International Advance Computing Conference (IACC)*. pp. 713-718, doi: 10.1109/IAdCC.2013.6514314.

018 Yongshuai Shao, Z. Chen, Fangfang Li and Chong Fu. 2016. Reconstruction of big sensor data. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. pp. 1-6, doi: 10.1109/CompComm.2016.7924653.

019 B. Twala and M. Cartwright. 2005. Ensemble imputation methods for missing software engineering data. In *11th IEEE International Software Metrics Symposium (METRICS'05)*. pp. 10 pp.-30, doi: 10.1109/METRICS.2005.21.

020 Harp, Steven Goldman, Robert Samad, Tariq. 1996. Imputation of Missing Data Using Machine Learning Techniques. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. pp. 140-145.

021 Y. Su, Z. Yang, N. Guo and H. Yang. 2021. Improving Quality of Smart Grid Data by Functional Data Analysis. In *2021 2nd International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)*. pp. 12-17, doi: 10.1109/ISCEIC53685.2021.00010.

022 X. Miao, Y. Gao, L. Chen, H. Peng, J. Yin and Q. Li. 2021. Towards Query Pricing on Incomplete Data (Extended Abstract). In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. pp. 2348-2349, doi: 10.1109/ICDE51399.2021.00260.

023 W. Han and M. Jochum. 2020. Machine Learning Approach for Data Quality Control of Earth Observation Data Management System. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. pp. 3101-3103, doi: 10.1109/IGARSS39084.2020.9323615.

024 Kin Seng Lei and Feng Wan. 2010. Pre-processing for missing data: A hybrid approach to air pollution prediction in Macau. In *2010 IEEE International Conference on Automation and Logistics, 2010*. pp. 418-422, doi: 10.1109/ICAL.2010.5585320.

025 M. M. Islam, G. Lee and S. N. Hettiwatte. 2017. Missing measurement estimation of power transformers using a GRNN. In *2017 Australasian Universities Power Engineering Conference (AUPEC)*. pp. 1-5, doi: 10.1109/AUPEC.2017.8282431.

026 C. Fiot, A. Laurent and M. Teisseire. 2007. Approximate Sequential Patterns for Incomplete Sequence Database Mining. In *2007 IEEE International Fuzzy Systems Conference*. pp. 1-6, doi: 10.1109/FUZZY.2007.4295445.

027 A. K. Pani, K. G. Amin and H. K. Mohanta. 2012. Data driven soft sensor of a cement mill using generalized regression neural network. In *2012 International Conference on Data Science Engineering (ICDSE)*. pp. 98-102, doi: 10.1109/ICDSE.2012.6281902.

028 S. Yan, D. Chen, S. Wang and S. Liu. 2020. Quality prediction method for aluminum alloy ingot based on XGBoost. In *2020 Chinese Control And Decision Conference (CCDC)*. pp. 2542-2547, doi: 10.1109/CCDC49329.2020.9164112.

029 Z. Lv, W. Deng, Z. Zhang, N. Guo and G. Yan. 2019. A Data Fusion and Data Cleaning System for Smart Grids Big Data. In *2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*. pp. 802-807, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00119.

030 S. Sim, H. Bae and Y. Choi. 2019. Likelihood-based Multiple Imputation by Event Chain Methodology for Repair of Imperfect Event Logs with Missing Data. In *2019 International Conference on Process Mining (ICPM)*. pp. 9-16, doi: 10.1109/ICPM.2019.00013.

031 Saleem, Asma, Khadim Hussain Asif, Ahmad Ali, Shahid Mahmood Awan and Mohammad A. Alghamdi. 2014. Pre-processing Methods of Data Mining. In *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (2014)*. pp. 451-456.

032 M. H. Cartwright, M. J. Shepperd and Q. Song. 2003. Dealing with missing software project data. In *Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717)*. pp. 154-165, doi: 10.1109/METRIC.2003.1232464.

033 W. Liu, D. Wei and F. Zhou. 2018. Fault diagnosis based on deep learning subject to missing data. In *2018 Chinese Control And Decision Conference (CCDC)*. pp. 3972-3977, doi: 10.1109/CCDC.2018.8407813.

034 C. T. Tran, M. Zhang and P. Andreae. 2016. Directly evolving classifiers for missing data using genetic programming. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. pp. 5278-5285, doi: 10.1109/CEC.2016.7748361.

035 Yan, Xiaobo Xiong, Weiqing Hu, Liang Wang, Feng Zhao, Kuo. 2015. Missing Value Imputation Based on Gaussian Mixture Model for the Internet of Things. In *Mathematical Problems in Engineering*. pp. 1-8. 10.1155/2015/548605.

036 Agbo, Benjamin, Yongrui Qin and Richard Hill. 2020. Best Fit Missing Value Imputation (BFMVI) Algorithm for Incomplete Data in the Internet of Things. In *IoTBDS 2020*. pp. 1-8.

037 A. González-Vidal, P. Rathore, A. S. Rao, J. Mendoza-Bernal, M. Palaniswami and A. F. Skarmeta-Gómez. 2021. Missing Data Imputation With Bayesian Maximum Entropy for Internet of Things Applications. In *IEEE Internet of Things Journal*, vol. 8, no. 21. pp. 16108-16120, doi: 10.1109/JIOT.2020.2987979.

038 Gyeong Ho Lee, Jaeseob Han, Jun Kyun Choi. 2021. MPdist-based missing data imputation for supporting big data analyses in IoT-based applications. In *Future Generation Computer Systems*, vol. 125. pp. 421-432, https://doi.org/10.1016/j.future.2021.06.042.

039 N. U. Okafor and D. T. Delaney. 2021. Missing Data Imputation on IoT Sensor Networks: Implications for on-Site Sensor Calibration. In *IEEE Sensors Journal*, vol. 21, no. 20. pp. 22833-22845, doi: 10.1109/JSEN.2021.3105442.

040 Y. Liu, T. Dillon, W. Yu, W. Rahayu and F. Mostafa. 2020. Missing Value Imputation for Industrial IoT Sensor Data With Large Gaps. In *IEEE Internet of Things Journal*, vol. 7, no. 8. pp. 6855-6867, doi: 10.1109/JIOT.2020.2970467.

041 Y. Wang, K. Li, S. Gan and C. Cameron. 2019. Missing Data Imputation With OLS-Based Autoencoder for Intelligent Manufacturing. In *IEEE Transactions on Industry Applications*, vol. 55, no. 6. pp. 7219-7229, Nov.-Dec. 2019, doi: 10.1109/TIA.2019.2940585.

042 Stefan Steiner, Yan Zeng, Timothy M. Young, David J. Edwards, Frank M. Guess and Chung-Hao Chen. 2016. A Study of Missing Data Imputation in Predictive Modeling of a Wood-Composite Manufacturing Process. In *Journal of Quality Technology*, vol. 48, no. 3. pp. 284-296, doi: 10.1080/00224065.2016.11918167.

043 Radhakrishna Vangipuram, Rajesh Kumar Gunupudi, Puligadda Kumar Veereswara and Janaki V. 2020. A machine learning approach for imputation and anomaly detection in IoT environment. In *Expert Systems*. pp. 1-37. doi: 10.1111/exsy.12556.

044 M. F. Pirehgalin and B. Vogel-Heuser. 2018. Estimation of Missing Values in Incomplete Industrial Process Data Sets Using ECM Algorithm. In *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. pp. 251-257, doi: 10.1109/INDIN.2018.8471950.

045 Wang Huan, Yuan Zhaolin, Chen Yibin, Shen Bingyang and Wu Aixiang. 2019. An Industrial Missing Values Processing Method Based on Generating Model. In *Computer Networks*. pp. 1-158. doi: 10.1016/j.comnet.2019.02.007.

046 Kwak Doh-Soon, Kim Kwang-Jae. 2012. A data mining approach considering missing values for the optimization of semiconductor-manufacturing processes. In *Expert System Applications*, vol. 39. pp. 2590-2596. doi: 10.1016/j.eswa.2011.08.114.

047 Z. Yao and C. Zhao. 2021. FIGAN: A Missing Industrial Data Imputation Method Customized for Soft Sensor Application. In *IEEE Transactions on Automation Science and Engineering*, doi: 10.1109/TASE.2021.3132037.

048 Luo Lijia, Bao Shiyi and Peng Xin. 2019. Robust monitoring of industrial processes using process data with outliers and missing values. In *Chemometrics and Intelligent Laboratory Systems*, pp. 1-192. 103827. 10.1016/j.chemolab.2019.103827.

049 Linsheng Zhong, Yuqing Chang, Fuli Wang and Shihong Gao. 2022. Distributed Missing Values Imputation Schemes for Plant-Wide Industrial Process Using Variational Bayesian Principal Component Analysis. In *Industrial Engineering Chemical Research*, vol. 61. pp. 580593. doi: https://doi.org/10.1021/acs.iecr.1c03860.

050 Bechny Michal, Sobieczky Florian, Zeindl Jürgen and Ehrlinger Lisa. 2021. Missing Data Patterns: From Theory to an Application in the Steel Industry. In *Conference: SSDBM 2021: 33rd International Conference on Scientific and Statistical Database Management*. pp. 214-219. doi: 10.1145/3468791.3468841.

051 Alireza Farhangfar, Lukasz Kurgan and Jennifer Dy. 2008. Impact of imputation of missing values on classification error for discrete data. In *Pattern Recognition*, vol. 41, no. 12. pp. 3692-3705, doi: https://doi.org/10.1016/j.patcog.2008.05.019.

052 Cho Eunnuri, Chang Tai-Woo and Hwang Gyusun. 2022. Data Preprocessing Combination to Improve the Performance of Quality Classification in the Manufacturing Process. In *Electronics*, vol. 11. pp. 477. doi: 10.3390/electronics11030477.

053 Karkare Rasika, Paffenroth Randy and Apelian Diran. 2022. Self-Supervised Deep Hadamard Autoencoders for Treating Missing Data: A Case Study in Manufacturing. In *Integrating Materials and Manufacturing Innovation*. pp. 1-11. doi: 10.1007/s40192-022-00254-7.

054 Xu Zhou, Xiaofeng Liu, Gongjin Lan and Jian Wu. 2021. Federated conditional generative adversarial nets imputation method for air quality missing data. In *Knowledge-Based Systems*, vol. 228. pp. 1-12. doi: https://doi.org/10.1016/j.knosys .2021.107261.

055 Soroush Ojagh, Francesco Cauteruccio, Giorgio Terracina and Steve H.L. Liang. 2021. Enhanced air quality prediction by edge-based spatiotemporal data preprocessing. In *Computers Electrical Engineering*, vol. 96. pp. 1-12. doi: https://doi.org/10.1016/j.compeleceng.2021.107572.

056 V. Khadse, P. N. Mahalle and S. V. Biraris. 2018. An Empirical Comparison of Supervised Machine Learning Algorithms for Internet of Things Data. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697476.

057 Tran Cao Truong, Zhang Mengjie, Andreae Peter and Xue Bing. 2017. Multiple Imputation and Genetic Programming for Classification with Incomplete Data. In *The Genetic and Evolutionary Computation Conference, GECCO 2017At:*

Berlin, GermanyVolume: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, Berlin, Germany, July 15-19, 2017. pp. 1-8. doi: 10.1145/3071178.3071181.

058 A. Farhangfar, L. A. Kurgan and W. Pedrycz. 2007. A Novel Framework for Imputation of Missing Values in Databases. In *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 5. pp. 692-709. doi: 10.1109/TSMCA.2007.902631.

059 C. Zhang, Q. Guo and Y. Li. 2020. Fault Detection in the Tennessee Eastman Benchmark Process Using Principal Component Difference Based on K-Nearest Neighbors. In *IEEE Access*, vol. 8. pp. 49999-50009. doi: 10.1109/ACCESS.2020.2977421.

060 Choi Chanyoung, Hae-Ik Jung and Jaehyuk Cho. 2021. An Ensemble Method for Missing Data of Environmental Sensor Considering Univariate and Multivariate Characteristics. In *Sensors*, Basel, Switzerland. pp. 1-22. doi: https://doi.org/10.3390/s21227595.

061 S. Sanyal and P. Zhang. 2018. Improving Quality of Data: IoT Data Aggregation Using Device to Device Communications. In *IEEE Access*, vol. 6. pp. 67830-67840. doi: 10.1109/ACCESS.2018.2878640.

062 Azimi Iman, Pahikkala Tapio, Rahmani Amir M., Niela-Vilen Hannakaisa, Axelin Anna and Liljeberg Pasi. 2019. Missing Data Resilient Decision-making for Healthcare IoT through Personalization: A Case Study on Maternal Health. In *Future Generation Computer Systems*, vol. 96. pp. 297-308. doi: 10.1016/j.future.2019.02.015.

063 Oh Eunseo and Lee Hyunsoo. 2020. An Imbalanced Data Handling Framework for Industrial Big Data Using a Gaussian Process Regression-Based Generative Adversarial Network. In *Symmetry*, vol. 12, no. 669. pp. 1-19. doi: 10.3390/sym12040669.

064 Ni Jiacheng, Li Li, Qiao Fei and Wu Qidi. 2014. A GS-MPSO-WKNN method for missing data imputation in wireless sensor networks monitoring manufacturing conditions. In *Transactions of the Institute of Measurement and Control*, vol. 36. pp. 1083-1092. doi: 10.1177/0142331214534291.

065 Guo Cen, Hu Wenkai, Yang Fan and Huang Dexian. 2020. Deep Learning Technique for Process Fault Detection and Diagnosis in the Presence of Incomplete Data. In *Chinese Journal of Chemical Engineering*, vol. 28. pp. 2358–2367. doi: 10.1016/j.cjche.2020.06.015.

066 Y.-L. Liang, C.-C. Kuo and C.-C. Lin. 2019. A Hybrid Memetic Algorithm for Simultaneously Selecting Features and Instances in Big Industrial IoT Data for Predictive Maintenance. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*. pp. 1266-1270, doi: 10.1109/INDIN41052.2019.8972199.

067 Nelwamondo Fulufhelo, Mohamed Shakir and Marwala Tshilidzi. 2007. Missing Data: A Comparison of Neural Network and Expectation Maximisation Techniques. In *Current Science*, vol. 93. pp. 1514-1521.

068 Xuegang Luo, Junrui Lv and Juan Wang. 2021. Missing Data Reconstruction Based on Spectral k-Support Norm Minimization for NB-IoT Data. In *Mathematical Problems in Engineering*. pp. 1-11. doi: 10.1155/2021/1336900.

069 Srinivasan Karthik, Currim Faiz, Ram Sudha, Lindberg Casey and Sternberg Esther. 2016. Feature Importance and Predictive Modeling for Multi-source Healthcare Data with Missing Values. In *The 6th International Conference on Digital Health Conference*. pp. 47-54. doi: 10.1145/2896338.2896347.

070 X. Jiang, Z. Tian and K. Li. 2021. A Graph-Based Approach for Missing Sensor Data Imputation. In *IEEE Sensors Journal*, vol. 21, no. 20. pp. 23133-23144. doi: 10.1109/JSEN.2021.3106656.

071 Andrews Mark, Jones Gavin, Leyde Brian, Xiong Lie, Xu Max and Chien Peter. 2019. A Statistical Imputation Method for Handling Missing Values in Generalized Polynomial Chaos Expansions. In *ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition*. pp. 1-20. doi: 10.1115/GT2019-91035.

072 Erhan Laura, Mario Di Mauro Ashiq Anjum, Ovidiu Bagdasar, Wei Song and Antonio Liotta. 2021. Embedded Data Imputation for Environmental Intelligent Sensing: A Case Study. In *Sensors*, vol. 21. Basel, Switzerland. pp. 1-17. doi: https://doi.org/10.3390/s21237774.

073 Cheng Hongju, Shi Yushi, Wu Leihuo, Guo Yingya and Xiong Naixue. 20212. An Intelligent Scheme for Big Data Recovery in Internet of Things Based on Multi-Attribute Assistance and Extremely Randomized Trees. In *Information Sciences*, vol. 557. pp. 66-83. doi: 10.1016/j.ins.2020.12.041.

074 Severson Kristen A., Molaro Mark C. and Braatz Richard D. 2017. Principal Component Analysis of Process Datasets with Missing Values. In *Processes*, vol. 5, no. 38. pp. 1-18. https://www.mdpi.com/2227-9717/5/3/38.

075 Teh Hui, Kempa-Liehr Andreas and Wang Kevin. 2020. Sensor data quality: a systematic review. In *Journal of Big Data*, vol. 7. pp. 1-49. doi: 10.1186/s40537-020-0285-1.

076 Azimi Shelernaz and Pahl Claus. 2021. The Effect of IoT Data Completeness and Correctness on Explainable Machine Learning Models. In *Database and Expert Systems Applications - 32nd International Conference, DEXA 2021, September 27–30, 2021 Proceedings, Part II*. pp. 151-160. doi: https://doi.org/10.1007/978-3-030-86475-0_15.

077 Han Li, Zhao Liu and Ping Zhu. 2021. An Engineering Domain Knowledge-Based Framework for Modelling Highly Incomplete Industrial Data. In *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 17, no. 4. pp. 1-19. doi: 10.4018/IJDWM.2021100103.

078 He D.K., Chu T.S., Lang Y.B. and Sun G.X. 2015. Comparison of Missing Data Imputation Methods for Leaching Process Modelling. In *2015 International Conference on Artificial Intelligence and Industrial Engineering*. pp. 497-500. doi: 10.2991/aiie-15.2015.134.

079 Mante Jeanet, Gangadharan Nishanthi, Sewell David, Turner Richard and Field Ray. 2019. A heuristic approach to handling missing data in biologics manufacturing databases. In *Bioprocess and Biosystems Engineering*, vol. 42. pp. 657–663. doi: 10.1007/s00449-018-02059-5.

080 Yin Shen, Wang Guang and Yang Xu. 2014. Robust PLS approach for KPI-related prediction and diagnosis against outliers and missing data. In *International Journal of Systems Science*, vol. 45. pp. 1464-5319. doi: 10.1080/00207721.2014.886136.

081 Marco S. Reis and Pedro M. Saraiva. 2006. Generalized Multiresolution Decomposition Frameworks for the Analysis of Industrial Data with Uncertainty and Missing Values. In *Industrial Engineering Chemical Research*, vol 45, no. 18. pp. 6330-6338. doi: https://doi.org/10.1021/ie051313b.

082 Anagnostopoulos Christos. 2016. Quality-optimized predictive analytics. In *Applied Intelligence*, vol. 45. pp. 1034–1046. doi: 10.1007/s10489-016-0807-x.

083 Van Stein Bas and Kowalczyk Wojtek. 2016. An Incremental Algorithm for Repairing Training Sets with Missing Values. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2016, Part II*. pp. 175-186. doi: 10.1007/978-3-319-40581-0_15.

084 Ruan Wenjie, Xu Peipei, Sheng Quan, Falkner Nickolas, Li Xue and Zhang Wei Emma. 2017. Recovering Missing Values from Corrupted Spatio-Temporal Sensory Data via Robust Low-Rank Tensor Completion. In *Database Systems for Advanced Applications (DASFAA 2017), Part I*. pp. 607-622. doi: 10.1007/978-3-319-55753-3_38.

085 Kang Wensheng. 2011. Missing-Data Imputation in Nonstationary Panel Data Models. In *Missing Data Methods: Time-Series Methods and Applications*. pp. 235-251. doi: 10.1108/S0731-9053(2011)000027B007.

086 S. N. Haider, Q. Zhao and B. K. Meran, Automated data cleaning for data centers: A case study. In *2020 39th Chinese Control Conference (CCC)*. pp. 3227-3232, doi: 10.23919/CCC50068.2020.9189357.

087 Farhangfar Alireza, Kurgan Lukasz and Pedrycz Witold. 2004. Experimental analysis of methods for imputation of missing values in databases. In *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5421. pp. 172-182. doi: 10.1117/12.542509.

088 Martinez Luengo Maria, Shafiee Mahmood and Kolios Athanasios. 2019. Data management for structural integrity assessment of offshore wind turbine support structures: data cleansing and missing data imputation. In *Ocean Engineering*, vol. 173. pp. 867-883. doi: 10.1016/j.oceaneng.2019.01.003.

089 Z. Chai, C. Zhao, B. Huang and H. Chen, A Deep Probabilistic Transfer Learning Framework for Soft Sensor Modeling With Missing Data. In *IEEE Transactions on Neural Networks and Learning Systems*. pp. 1-12. doi: 10.1109/TNNLS.2021.3085869.

090 - Tran Cao Truong, Zhang Mengjie, Andreae Peter and Xue Bing. 2016. A Wrapper Feature Selection Approach to Classification with Missing Data. In *Lecture Notes in Computer Science - European Conference on the Applications of Evolutionary Computation - EvoApplications 2016: Applications of Evolutionary Computation*. pp. 685-700. doi: 10.1007/978-3-319-31204-0_44.

091 Qin Yongsong, Zhang Shichao and Zhang Chengqi. 2010. Combining kNN Imputation and Bootstrap Calibrated Empirical Likelihood for Incomplete Data Analysis. In *IJDWM*, vol. 6. pp. 61-73. doi: 10.4018/jdwm.2010100104.

092 Sarkar Sobhan, Pramanik Anima, Khatedi Nikhil and Maiti Jhareswar. 2020. An Investigation of the Effects of Missing Data Handling Using 'R'-Packages. In *Data Engineering and Communication Technology*. pp. 275-285. doi: 10.1007/978-981-15-1097-7_24.

093 J. Choi, Y. Son and M. K. Jeong. 2021. Restricted Relevance Vector Machine for Missing Data and Application to Virtual Metrology. In *IEEE Transactions on Automation Science and Engineering*, pp. 1-12. doi: 10.1109/TASE.2021.3111096.

094 Tkachenko Roman, Izonin Ivan, Kryvinska Natalia, Dronyuk Ivanna and Zub Khrystyna. 2020. An Approach towards Increasing Prediction Accuracy for the Recovery of Missing IoT Data based on the GRNN-SGTM Ensemble. In *Sensors*, vol. 20, no. 2625. pp. 1-15. doi: 10.3390/s20092625.

095 Lakshminarayan Kamakshi, Steven A. Harp and Tariq Samad. 1999. Imputation of Missing Data in Industrial Databases. In *Applied Intelligence*, vol. 11. pp. 259-275.

096 Gupta Prakhar and Raghavan Srinivasan. 2011. Missing Data Prediction and Forecasting for Water Quantity Data. In *Modeling, Simulation and Control - International Proceedings of Computer Science and Information Technology*. pp. 98-102.

097 - Tian Jing, Yu Bing, Yu Dan and Ma Shilong. 2013. Clustering-Based Multiple Imputation via Gray Relational Analysis for Missing Data and Its Application to Aerospace Field. In *The Scientific World Journal*. pp. 1-10. doi: 10.1155/2013/720392.

098 A. Peterkova, M. Nemeth and A. Bohm. 2018. Computing Missing Values Using Neural Networks in Medical Field. In *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)*. pp. 000151-000156. doi: 10.1109/INES.2018.8523857.

099 Z. B. Othmane, D. Bodénès, C. de Runz and A. A. Younes. 2018. A Multi-sensor Visualization Tool for Harvested Web Information: Insights on Data Quality. In *2018 22nd International Conference Information Visualisation (IV)*. pp. 110-116. doi: 10.1109/iV.2018.00029.

100 Caiafa Cesar, Sun Zhe, Tanaka Toshihisa Marti-Puig Pere and Solé-Casals Jordi. 2021. Machine Learning Methods with Noisy, Incomplete or Small Datasets. In *Applied Sciences*, vol. 11, no. 4132. pp. 1-4. doi: 10.3390/app11094132.

101 X. Yuan, Z. Ge, B. Huang and Z. Song. 2016. A Probabilistic Just-in-Time Learning Framework for Soft Sensor Development With Missing Data. In *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3. pp. 1124-1132. doi: 10.1109/TCST.2016.2579609.

102 - B. Twala and M. Cartwright. 2005. Ensemble imputation methods for missing software engineering data. In *11th IEEE International Software Metrics Symposium (METRICS'05)*. pp. 10-30. doi: 10.1109/METRICS.2005.21.

103 Céline Fiot, Anne Laurent and Maguelonne Teisseire. 2007. SPoID: Do Not Throw Meaningful Incomplete Sequences Away! In *New Dimensions in Fuzzy Logic and Related Technologies*, vol. 1. pp. 329-336.

104 N. Solomakhina, T. Hubauer, S. Lamparter, M. Roshchin and S. Grimm. 2014. Extending statistical data quality improvement with explicit domain models. In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*. pp. 720-725. doi: 10.1109/INDIN.2014.6945602.

105 A. Baggag et al. 2021. Learning Spatiotemporal Latent Factors of Traffic via Regularized Tensor Factorization: Imputing Missing Values and Forecasting. In *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6. pp. 2573-2587. doi: 10.1109/TKDE.2019.2954868.

106 W. Han and M. Jochum. 2020. A Machine Learning Approach for Data Quality Control of Earth Observation Data Management System. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. pp. 3101-3103. doi: 10.1109/IGARSS39084.2020.9323615.

107 Llanes-Santiago Orestes, B. C. Rivero-Benedico, S. C. Gálvez-Viera, E. F. Rodríguez-Morant, R. Torres-Cabeza and A. J. Silva-Neto. 2019. A Fault Diagnosis Proposal with Online Imputation to Incomplete Observations in Industrial Plants. In *Revista Mexicana de Ingeniería Química*. pp. 1-16. doi: https://doi.org/10.24275/uam/izt/dcbi/revmexingquim/2019v18n1/Llanes.

108 R. Gong, S. H. Huang and T. Chen. 2008. Robust and Efficient Rule Extraction Through Data Summarization and Its Application in Welding Fault Diagnosis. In *IEEE Transactions on Industrial Informatics*, vol. 4, no. 3. pp. 198-206. doi: 10.1109/TII.2008.2002920.

109 Kadlec Petr and Gabrys Bogdan. 2008. Soft Sensor Based on Adaptive Local Learning. In *Advanced Neuro Informatics Processing Part I*, vol. 5506. pp. 1172-1179. doi: 10.1007/978-3-642-02490-0_142.

110 Zulkepli Fatin, Ibrahim Roliana and Saeed Faisal. 2017. Data Preprocessing Techniques for Research Performance Analysis. In *Recent Developments in Intelligent Computing, Communication and Devices*. pp.157-162. doi: 10.1007/978-981-10-3779-5_20.

111 Liu Yu, Gong Xian, Wang Zhi, Liu Wei and Nie Zuo. 2009. Multiple Imputation for Missing Data in Life Cycle Inventory. In *Materials Science Forum*. pp. 610-613. doi: 10.4028/www.scientific.net/MSF.610-613.21.

112 Farhan Javeria. 2015. Overview of missing physical commodity trade data and its imputation using data augmentation. In *Transportation Research Part C: Emerging Technologies*, vol. 54. pp. 1-14. doi: 10.1016/j.trc.2015.02.021.

113 S. Wang and G. Mao. 2018. Missing Data Estimation for Traffic Volume by Searching an Optimum Closed Cut in Urban Networks. In *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1. pp. 75-86. doi: 10.1109/TITS.2018.2801808.

114 Prabhu Amogh, Edgar Thomas and Good Richard. 2009. Missing data estimation for run-to-run EWMA-controlled processes. In *Computers Chemical Engineering*, vol. 33. pp. 1861-1869. doi: 10.1016/j.compchemeng.2009.05.010.

115 Comandella Daniele, Gottardo Stefania, Rio-Echevarria Iria and Rauscher Hubert. 2020. Quality of physicochemical data on nanomaterials: an assessment of data completeness and variability. In *Nanoscale*, vol. 12. pp. 1-14. doi: 10.1039/C9NR08323E.

116 K. Kim, K. Kim, C. Jun, I. Chong and G. Song. 2019. Variable Selection Under Missing Values and Unlabeled Data in Semiconductor Processes. In *IEEE Transactions on Semiconductor Manufacturing*, vol. 32, no. 1. pp. 121-128. doi: 10.1109/TSM.2018.2881286.

117 Wang Xiaomeng, Nauck Detlef, Spott Martin and Kruse Rudolf. 2007. Intelligent data analysis with fuzzy decision trees. In *Soft Computing*, vol. 11. pp. 439-457. doi: 10.1007/s00500-006-0108-0.

118 Shardt Yuri and Brooks Kevin. 2018. Automated System Identification in Mineral Processing Industries: A Case Study using the Zinc Flotation Cell. In *IFAC-PapersOnLine - Conference Paper Archive*, vol. 51. pp. 132-137. doi: 10.1016/j.ifacol.2018.09.288.

119 Venkatesan Rajkumar, Bleier Alexander, Reinartz Werner and Ravishanker Nalini. 2019. Improving customer profit predictions with customer mindset metrics through multiple overimputation. In *Journal of the Academy of Marketing Science*, vol. 47. pp. 771-794. doi: 10.1007/s11747-019-00658-6.

120 R. Ratolojanahary, R. Houé Ngouna, K. Medjaher, F. Dauriac, M. Sebilo. 2019. Groundwater quality assessment combining supervised and unsupervised methods. In *IFAC-PapersOnLine - Conference Paper Archive*, vol. 52, no. 10. pp. 340-345. doi: https://doi.org/10.1016/j.ifacol.2019.10.054.

121 W. Shao, Z. Ge, L. Yao and Z. Song. 2020. Bayesian Nonlinear Gaussian Mixture Regression and its Application to Virtual Sensing for Multimode Industrial Processes. In *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2. pp. 871-885. doi: 10.1109/TASE.2019.2950716.

122 Yang Qishan, Ge Mouzhi and Helfert Markus. 2016. Guidelines of Data Quality Issues for Data Integration in the Context of the TPC-DI Benchmark. In *Proceedings of the 19th International Conference on Enterprise Information Systems - ICEIS*, vol. 1. pp. 135-144.

123 Shi Tao and Cressie Noel. 2007. Global statistical analysis of MISR aerosol data: A massive data product from NASA's Terra satellite. In *Environmetrics*, vol. 18. pp. 665 - 680. 10.1002/env.864.

124 D. Matheson, Chaoying Jing and F. Monforte. 2004. Meter data management for the electricity market. In *2004 International Conference on Probabilistic Methods Applied to Power Systems*. pp. 118-122.

125 Nordloh Vito, Roubícková Anna and Brown Nick. 2020. Machine Learning for Gas and Oil Exploration. In *University of Edinburgh Publications*. pp. 3009-3017. doi: 10.3233/FAIA200476.

126 A. Hinojosa and S. Stoyanov. 2018. Data Driven Predictive Model to Compact a Production Stop-on-Fail Test Set for an Electronic Device. In *2018 International Conference on Computing, Electronics Communications Engineering (iCCECE)*. pp. 59-64. doi: 10.1109/iCCECOME.2018.8658941.

127 Goldrick Stephen, Sandner Viktor, Cheeks Matthew, Turner Richard, Farid Suzanne, McCreath Graham and Glassey Jarka. 2019. Multivariate Data Analysis Methodology to Solve Data Challenges Related to Scale-Up Model Validation and Missing Data on a Micro-Bioreactor System. In *Biotechnology Journal*, vol. 15. doi: 10.1002/biot.201800684.

128 O. Shafiq, R. Alhajj and J. G. Rokne. 2014. Handling incomplete data using semantic logging based Social Network Analysis Hexagon for effective application monitoring and management. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*. pp. 634-641. doi: 10.1109/ASONAM.2014.6921652.

129 Trendowicz Adam, Siebert Julian, Jedlitschka Andreas and Martinez Fernandez S. 2019. Data Preparation - Tackle the Most Effort-Prone Phase in Data Projects. In *20th International Conference on Product-Focused Software Process*. pp. 770-771.

130 Zhu Fangzhou, Luo Chen, Yuan Mingxuan, Zhu Yijian, Zhang Zhengqing, Gu Tao, Deng Ke, Rao Weixiong and Zeng Jia. 2016. City-Scale Localization with Telco Big Data. In *CIKM'16: Proceedings of the 2016 ACM Conference on Information and Knowledge Management*. pp. 439-448. doi: 10.1145/2983323.2983345.

131 Al-Qerem Ahmad, Al-Naymat Ghazi, Alhasan Mays and Al-Debei Mutaz. 2020. Default Prediction Model: The Significant Role of Data Engineering in the Quality of Outcomes. In *The International Arab Journal of Information Technology*, vol. 17. pp. 635-644. doi: 10.34028/iajit/17/4A/8.

132 Brown Marvin and Kros John. 2003. Data mining and the impact of missing data. In *Industrial Management and Data Systems*, vol. 103. pp. 611-621. doi: 10.1108/02635570310497657.

133 - Caoimhe M Carbery. 2022. Missingness analysis of manufacturing systems: A case study. In *Journal of Engineering Manufacture*. pp. 1-12. doi: https://doi.org/10.1177/09544054221076631.

134 Dhungana Hariom, Bellotti Francesco, Berta Riccardo and Gloria Alessandro. 2021. Performance Comparison of Imputation Methods in Building Energy Data Sets. In *University of Genova Publications*. pp. 1-12. doi: 10.1007/978-3-030-66729-0_17.

135 V. K. Gond, A. Dubey and A. Rasool. 2021. A Survey of Machine Learning-Based Approaches for Missing Value Imputation. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*. pp. 1-8. doi: 10.1109/ICIRCA51532.2021.9544957.

136 Loukopoulos Panagiotis, Zolkiewski George, Bennett Ian, Pilidis Pericles, Duan Fang and Mba David. 2017. Dealing with missing data as it pertains of e-maintenance. In *Journal of Quality in Maintenance Engineering*, vol. 23. pp. 1-19. doi: 10.1108/JQME-08-2016-0032.

137 P. McMahon, T. Zhang and R. A. Dwight. 2020. Approaches to Dealing With Missing Data in Railway Asset Management. In *IEEE Access*, vol. 8, pp. 48177-48194. doi: 10.1109/ACCESS.2020.2978902.

138 A. Sadhu, R. Soni and M. Mishra. 2020. Pattern-based Comparative Analysis of Techniques for Missing Value Imputation. In *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*. pp. 513-518. doi: 10.1109/ICCCA49541.2020.9250825.

139 Xu Shu, Lu Bo, Baldea Michael, Edgar Thomas, Wojsznis Willy, Blevins Terry and Nixon Mark. 2015. Data cleaning in the process industries. In *Reviews in Chemical Engineering*, vol. 31. pp. 453-490. doi: 10.1515/revce-2015-0022.

140 Young William, Weckman Gary and Holland William. 2011. A survey of methodologies for the treatment of missing values within datasets: Limitations and benefits. In *Theoretical Issues in Ergonomics Science*, vol. 12. pp. 15-43. doi: 10.1080/14639220903470205.

141 Feldman Michael, Even Adir and Parmet Yisrael. 2017. A Methodology for Quantifying the Effect of Missing Data on Decision Quality in Classification Problems. In *Communications in Statistics - Theory and Methods*, vol. 47. pp. 1-22. doi: 10.1080/03610926.2016.1277752.

Selected Publications

001 R. Razavi-Far, E. Hallaji, M. Farajzadeh-Zanjani, R. Aljoudi and M. Saif. 2021. A Critical Study on the Impact of Missing Data Imputation for Classifying Intrusions in Cyber-Physical Water Systems. In *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*. pp. 1-6, doi: 10.1109/IECON48115.2021.958951 3.

003 Y. Ye, B. Zhong, S. Srimal, A. Alsarkhi and J. Talburt. 2018. A Study on the Impact of Missing Values in Probabilistic Matching. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. pp. 158-163, doi: 10.1109/CSCI46756.2018.00038.

006 M. Zhu and Xingbing Cheng. 2015. Iterative KNN imputation based on GRA for missing values in TPLMS. In *4th International Conference on Computer Science and Network Technology (ICCSNT)*. pp. 94-99, doi: 10.1109/ICCSNT.2015.7490714.

007 S. Juddoo and C. George. 2020. A Qualitative Assessment of Machine Learning Support for Detecting Data Completeness and Accuracy Issues to Improve Data Analytics in Big Data for the Healthcare Industry. In *2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communication s Engineering (ELECOM)*. pp. 58-66, doi: 10.1109/ELECOM49001.2020.929 7009.

009 M. Abhishek and N. S. V. Shet. 2019. Data Processing and deploying missing data algorithms to handle missing data in real time data of storage tank: A Cyber Physical Perspective. In *2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*. pp. 1-6, doi: 10.1109/ICECIE47765.2019.8974816.

011 H. Wang, Y. Chen, B. Shen, D. Wu and X. Ban. 2018. Generative Adversarial Networks Imputation for High Rate Missing Values. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018*. pp. 586-590, doi: 10.1109/Cybermatics_2018.2018.00121.

012 L. Ehrlinger, T. Grubinger, B. Varga, M. Pichler, T. Natschläger and J. Zeindl. 2018. Treating Missing Data in Industrial Data Analytics. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*. pp. 148-155, doi: 10.1109/ICDIM.2018.8846984.

014 K. Rahul and R. K. Banyal. 2019. Data Cleaning Mechanism for Big Data and Cloud Computing. In *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*. pp. 195-198.

020 Harp, Steven Goldman, Robert Samad, Tariq. 1996. Imputation of Missing Data Using Machine Learning Techniques. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. pp. 140-145.

027 A. K. Pani, K. G. Amin and H. K. Mohanta. 2012. Data driven soft sensor of a cement mill using generalized regression neural network. In *2012 International Conference on Data Science Engineering (ICDSE)*. pp. 98-102, doi: 10.1109/ICDSE.2012.6281902.

030 S. Sim, H. Bae and Y. Choi. 2019. Likelihood-based Multiple Imputation by Event Chain Methodology for Repair of Imperfect Event Logs with Missing Data. In *2019 International Conference on Process Mining (ICPM)*. pp. 9-16, doi: 10.1109/ICPM.2019.00013.

032 M. H. Cartwright, M. J. Shepperd and Q. Song. 2003. Dealing with missing software project data. In *Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717)*. pp. 154-165, doi: 10.1109/METRIC.2003.1232464.

037 A. González-Vidal, P. Rathore, A. S. Rao, J. Mendoza-Bernal, M. Palaniswami and A. F. Skarmeta-Gómez. 2021. Missing Data Imputation With Bayesian Maximum Entropy for Internet of Things Applications. In *IEEE Internet of Things Journal*, vol. 8, no. 21. pp. 16108-16120, doi: 10.1109/JIOT.2020.2987979.

039 N. U. Okafor and D. T. Delaney. 2021. Missing Data Imputation on IoT Sensor Networks: Implications for on-Site Sensor Calibration. In *IEEE Sensors Journal*, vol. 21, no. 20. pp. 22833-22845, doi: 10.1109/JSEN.2021.3105442.

040 Y. Liu, T. Dillon, W. Yu, W. Rahayu and F. Mostafa. 2020. Missing Value Imputation for Industrial IoT Sensor Data With Large Gaps. In *IEEE Internet of Things Journal*, vol. 7, no. 8. pp. 6855-6867, doi: 10.1109/JIOT.2020.2970467.

041 Y. Wang, K. Li, S. Gan and C. Cameron. 2019. Missing Data Imputation With OLS-Based Autoencoder for Intelligent Manufacturing. In *IEEE Transactions on Industry Applications*, vol. 55, no. 6. pp. 7219-7229, Nov.-Dec. 2019, doi: 10.1109/TIA.2019.2940585.

047 Z. Yao and C. Zhao. 2021. FIGAN: A Missing Industrial Data Imputation Method Customized for Soft Sensor Application. In *IEEE Transactions on Automation Science and Engineering*, doi: 10.1109/TASE.2021.3132037.

051 Alireza Farhangfar, Lukasz Kurgan and Jennifer Dy. 2008. Impact of imputation of missing values on classification error for discrete data. In *Pattern Recognition*, vol. 41, no. 12. pp. 3692-3705, doi: https://doi.org/10.1016/j.patcog.2008.05.019.

058 A. Farhangfar, L. A. Kurgan and W. Pedrycz. 2007. A Novel Framework for Imputation of Missing Values in Databases. In *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 5. pp. 692-709. doi: 10.1109/TSMCA.2007.902631.

072 Erhan Laura, Mario Di Mauro Ashiq Anjum, Ovidiu Bagdasar, Wei Song and Antonio Liotta. 2021. Embedded Data Imputation for Environmental Intelligent Sensing: A Case Study. In *Sensors*, vol. 21. Basel, Switzerland. pp. 1-17. doi: https://doi.org/10.3390/s21237774.

107 Llanes-Santiago Orestes, B. C. Rivero-Benedico, S. C. Gálvez-Viera, E. F. Rodríguez-Morant, R. Torres-Cabeza and A. J. Silva-Neto. 2019. A Fault Diagnosis Proposal with Online Imputation to Incomplete Observations in Industrial Plants. In *Revista Mexicana de Ingeniería Química*. pp. 1-16. doi: https://doi.org/10.24275/uam/izt/dcbi/revmexingquim/2019v18n1/Llanes.

133 Caoimhe M Carbery. 2022. Missingness analysis of manufacturing systems: A case study. In *Journal of Engineering Manufacture*. pp. 1-12. doi: https://doi.org/10.1177/09544054221076631.

136 Loukopoulos Panagiotis, Zolkiewski George, Bennett Ian, Pilidis Pericles, Duan Fang and Mba David. 2017. Dealing with missing data as it pertains of e-maintenance. In *Journal of Quality in Maintenance Engineering*, vol. 23. pp. 1-19. doi: 10.1108/JQME-08-2016-0032.

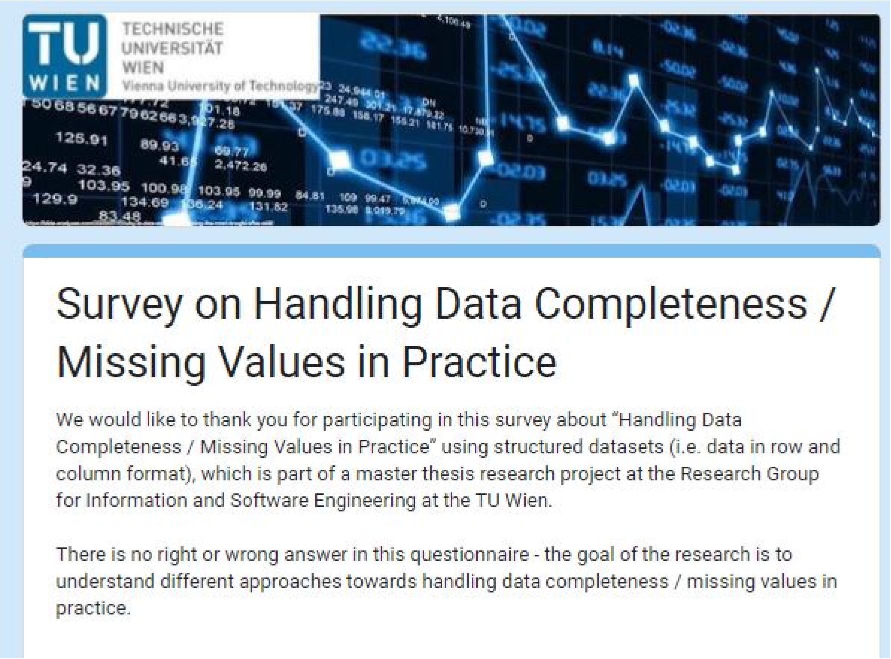
137 P. McMahon, T. Zhang and R. A. Dwight. 2020. Approaches to Dealing With Missing Data in Railway Asset Management. In *IEEE Access*, vol. 8, pp. 48177-48194. doi: 10.1109/ACCESS.2020.2978902.

138 A. Sadhu, R. Soni and M. Mishra. 2020. Pattern-based Comparative Analysis of Techniques for Missing Value Imputation. In *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*. pp. 513-518. doi: 10.1109/ICCCA49541.2020.9250825.

140 Young William, Weckman Gary and Holland William. 2011. A survey of methodologies for the treatment of missing values within datasets: Limitations and benefits. In *Theoretical Issues in Ergonomics Science*, vol. 12. pp. 15-43. doi: 10.1080/14639220903470205.

141 Feldman Michael, Even Adir and Parmet Yisrael. 2017. A Methodology for Quantifying the Effect of Missing Data on Decision Quality in Classification Problems. In *Communications in Statistics - Theory and Methods*, vol. 47. pp. 1-22. doi: 10.1080/03610926.2016.1277752.

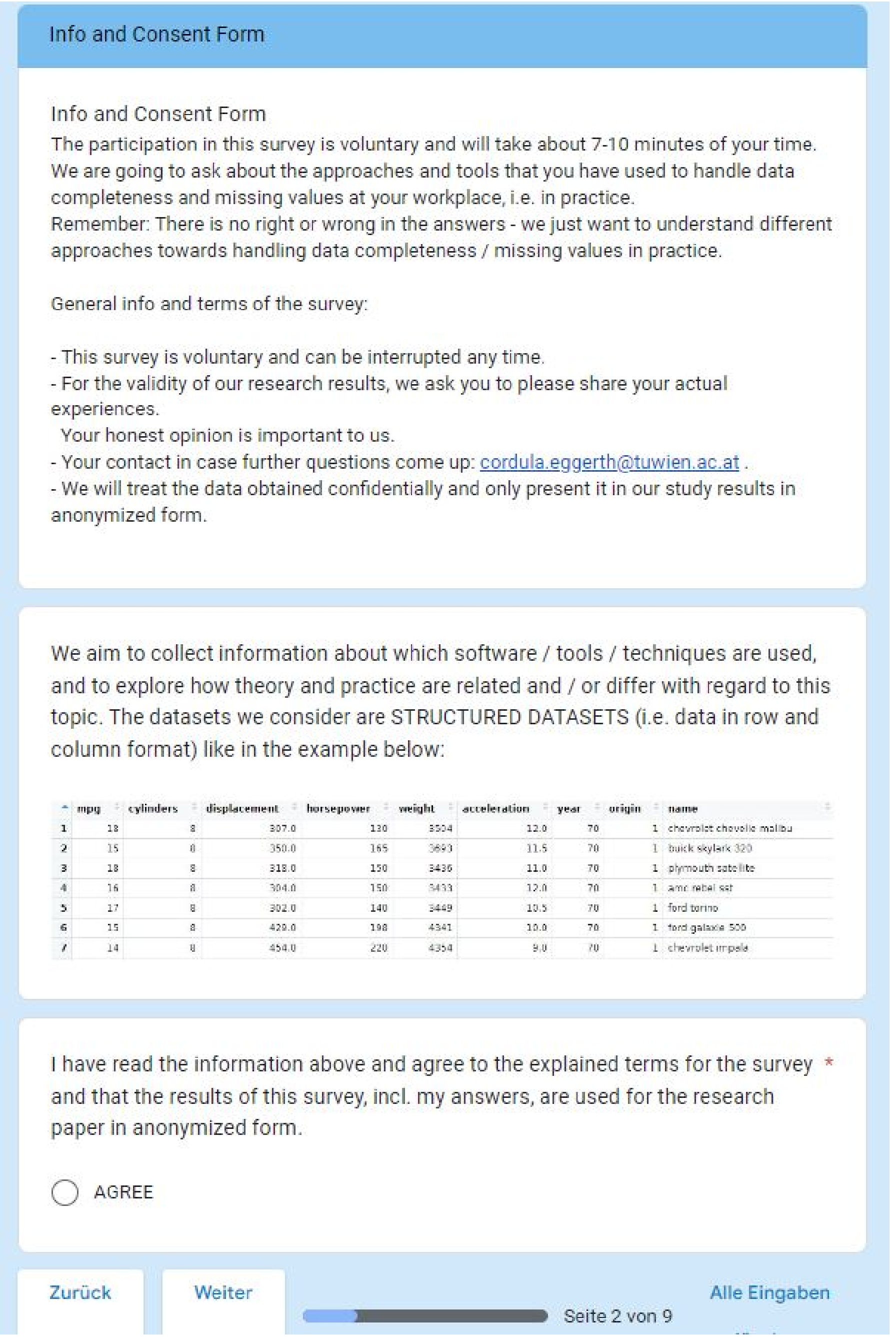
Online Survey: Questionnaire



Survey on Handling Data Completeness / Missing Values in Practice

We would like to thank you for participating in this survey about "Handling Data Completeness / Missing Values in Practice" using structured datasets (i.e. data in row and column format), which is part of a master thesis research project at the Research Group for Information and Software Engineering at the TU Wien.

There is no right or wrong answer in this questionnaire - the goal of the research is to understand different approaches towards handling data completeness / missing values in practice.



Info and Consent Form

Info and Consent Form

The participation in this survey is voluntary and will take about 7-10 minutes of your time.
We are going to ask about the approaches and tools that you have used to handle data completeness and missing values at your workplace, i.e. in practice.
Remember: There is no right or wrong in the answers - we just want to understand different approaches towards handling data completeness / missing values in practice.

General info and terms of the survey:

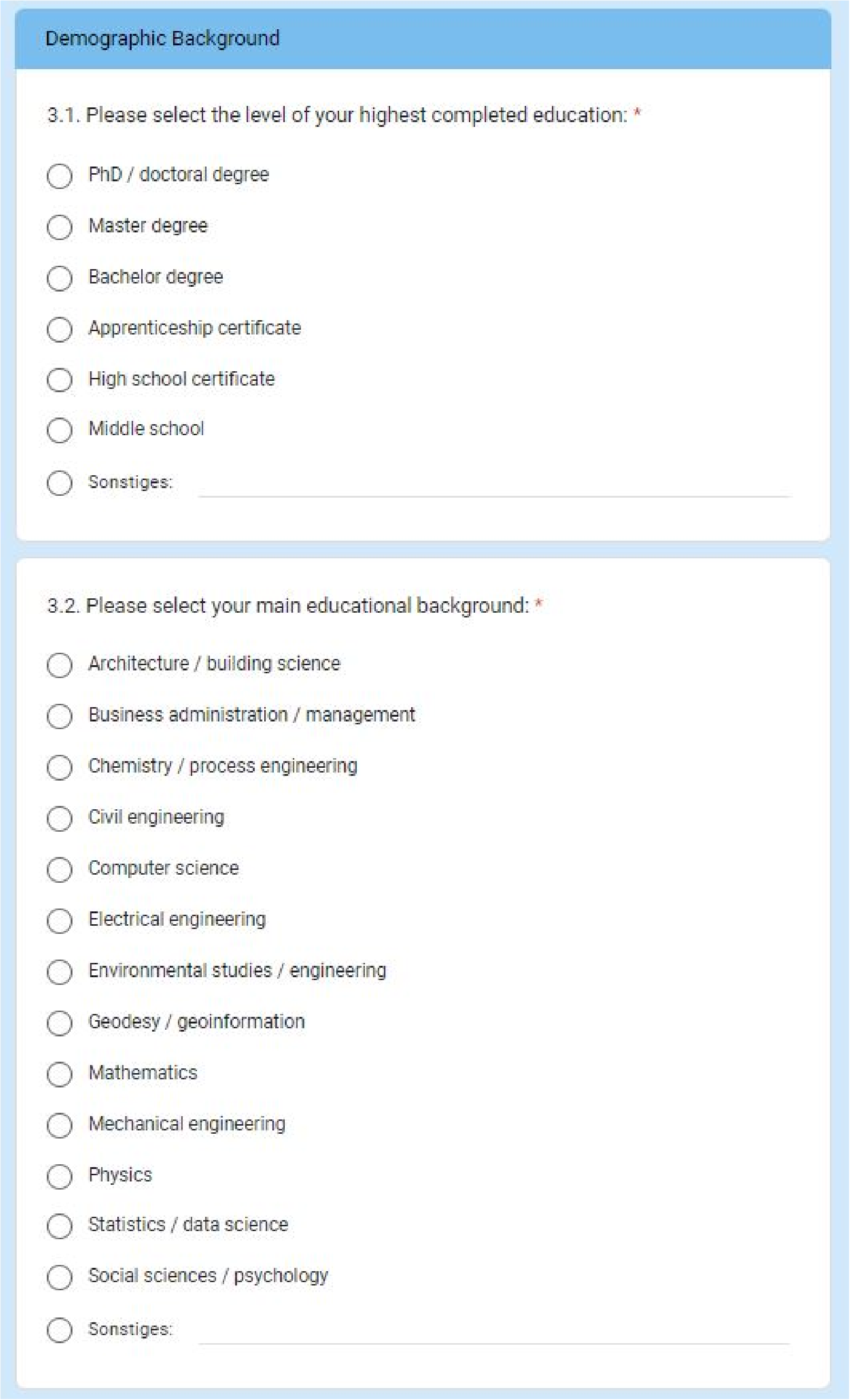
- This survey is voluntary and can be interrupted any time.
- For the validity of our research results, we ask you to please share your actual experiences.
 Your honest opinion is important to us.
- Your contact in case further questions come up: cordula.eggerth@tuwien.ac.at .
- We will treat the data obtained confidentially and only present it in our study results in anonymized form.

We aim to collect information about which software / tools / techniques are used, and to explore how theory and practice are related and / or differ with regard to this topic. The datasets we consider are STRUCTURED DATASETS (i.e. data in row and column format) like in the example below:

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
1	18	8	307.0	130	3504	12.0	70	1	chevrolet chevelle malibu
2	15	8	350.0	165	3693	11.5	70	1	buick skylark 320
3	18	8	318.0	150	3436	11.0	70	1	plymouth satellite
4	16	8	304.0	150	3433	12.0	70	1	amc rebel sst
5	17	8	302.0	140	3449	10.5	70	1	ford torino
6	15	8	429.0	198	4341	10.0	70	1	ford galaxie 500
7	14	8	454.0	220	4354	9.0	70	1	chevrolet impala

I have read the information above and agree to the explained terms for the survey and that the results of this survey, incl. my answers, are used for the research paper in anonymized form. *

○ AGREE



Demographic Background

3.1. Please select the level of your highest completed education: *

○ PhD / doctoral degree

○ Master degree

○ Bachelor degree

○ Apprenticeship certificate

○ High school certificate

○ Middle school

○ Sonstiges: ____________________

3.2. Please select your main educational background: *

○ Architecture / building science

○ Business administration / management

○ Chemistry / process engineering

○ Civil engineering

○ Computer science

○ Electrical engineering

○ Environmental studies / engineering

○ Geodesy / geoinformation

○ Mathematics

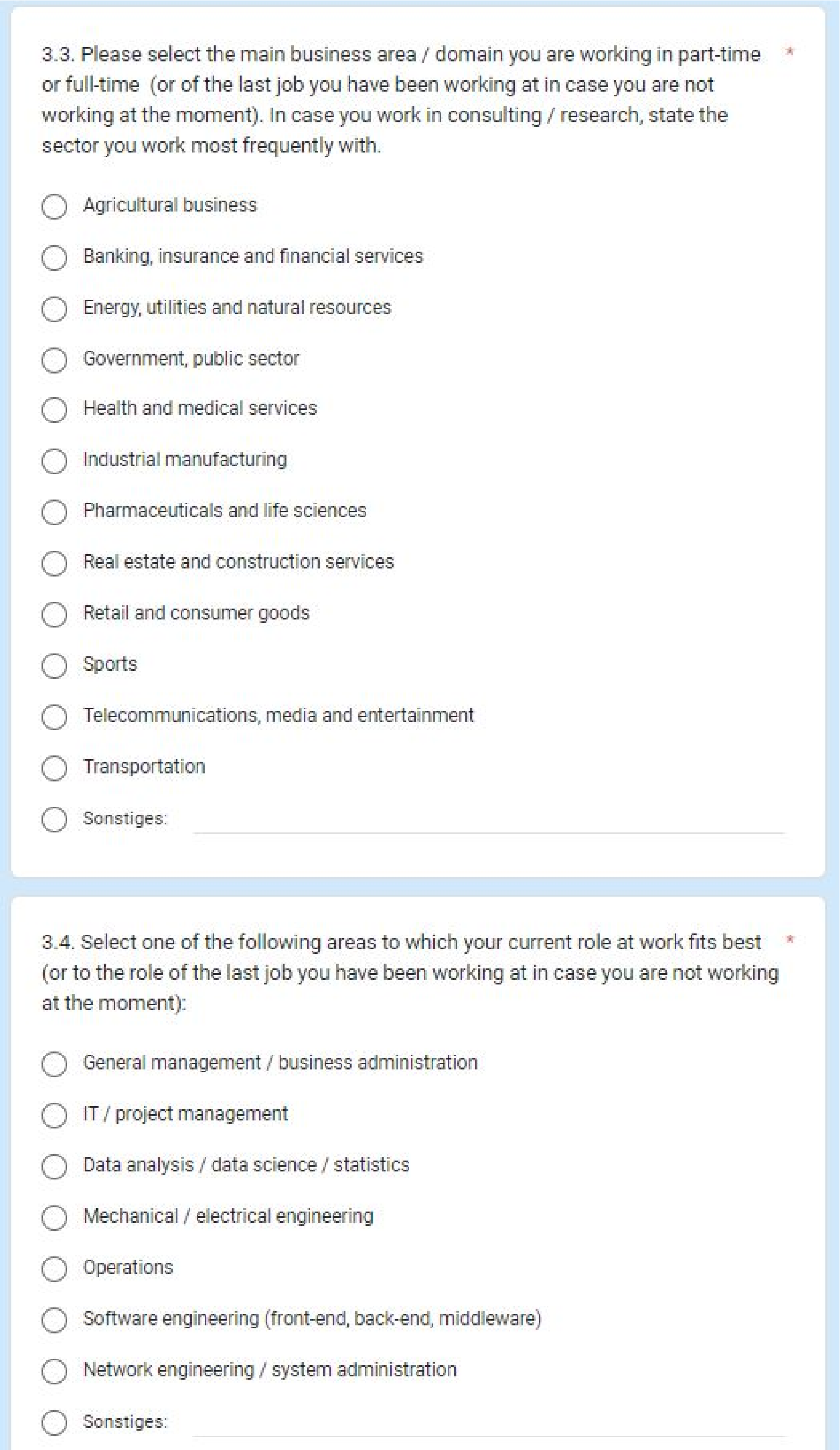
○ Mechanical engineering

○ Physics

○ Statistics / data science

○ Social sciences / psychology

○ Sonstiges: ____________________



3.3. Please select the main business area / domain you are working in part-time or full-time (or of the last job you have been working at in case you are not working at the moment). In case you work in consulting / research, state the sector you work most frequently with. *

○ Agricultural business

○ Banking, insurance and financial services

○ Energy, utilities and natural resources

○ Government, public sector

○ Health and medical services

○ Industrial manufacturing

○ Pharmaceuticals and life sciences

○ Real estate and construction services

○ Retail and consumer goods

○ Sports

○ Telecommunications, media and entertainment

○ Transportation

○ Sonstiges: ____________________

3.4. Select one of the following areas to which your current role at work fits best (or to the role of the last job you have been working at in case you are not working at the moment): *

○ General management / business administration

○ IT / project management

○ Data analysis / data science / statistics

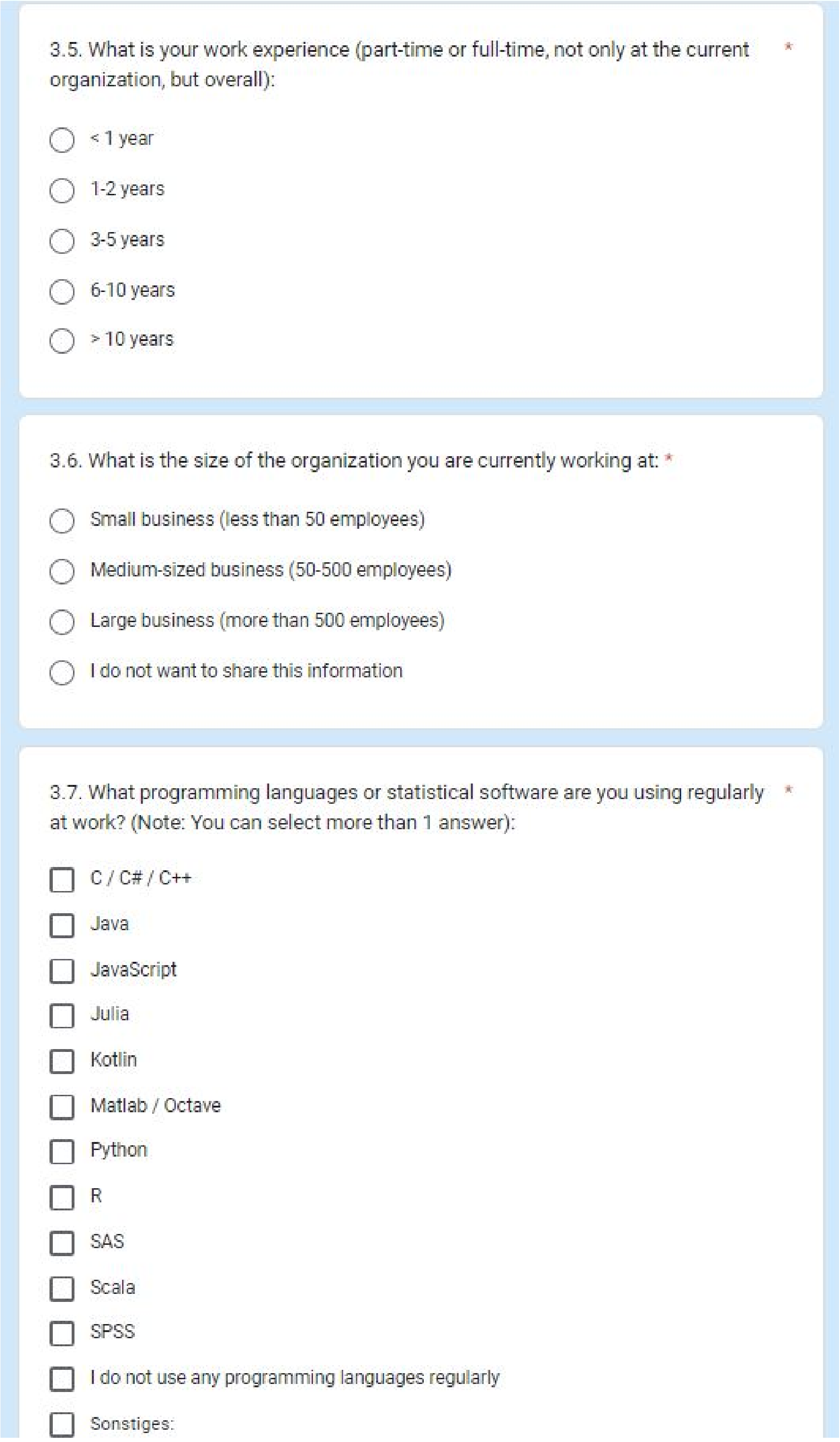
○ Mechanical / electrical engineering

○ Operations

○ Software engineering (front-end, back-end, middleware)

○ Network engineering / system administration

○ Sonstiges: ____________________



3.5. What is your work experience (part-time or full-time, not only at the current organization, but overall): *

○ < 1 year

○ 1-2 years

○ 3-5 years

○ 6-10 years

○ > 10 years

3.6. What is the size of the organization you are currently working at: *

○ Small business (less than 50 employees)

○ Medium-sized business (50-500 employees)

○ Large business (more than 500 employees)

○ I do not want to share this information

3.7. What programming languages or statistical software are you using regularly at work? (Note: You can select more than 1 answer): *

□ C / C# / C++

□ Java

□ JavaScript

□ Julia

□ Kotlin

□ Matlab / Octave

□ Python

□ R

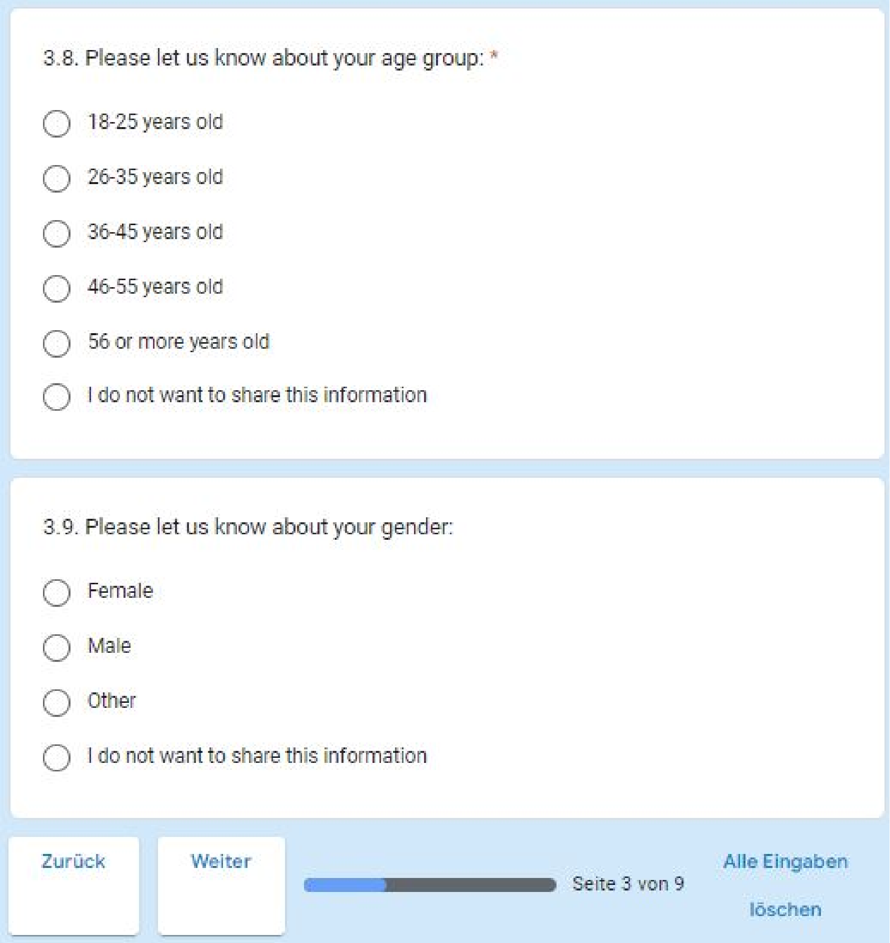
□ SAS

□ Scala

□ SPSS

□ I do not use any programming languages regularly

□ Sonstiges:



3.8. Please let us know about your age group: *

○ 18-25 years old

○ 26-35 years old

○ 36-45 years old

○ 46-55 years old

○ 56 or more years old

○ I do not want to share this information

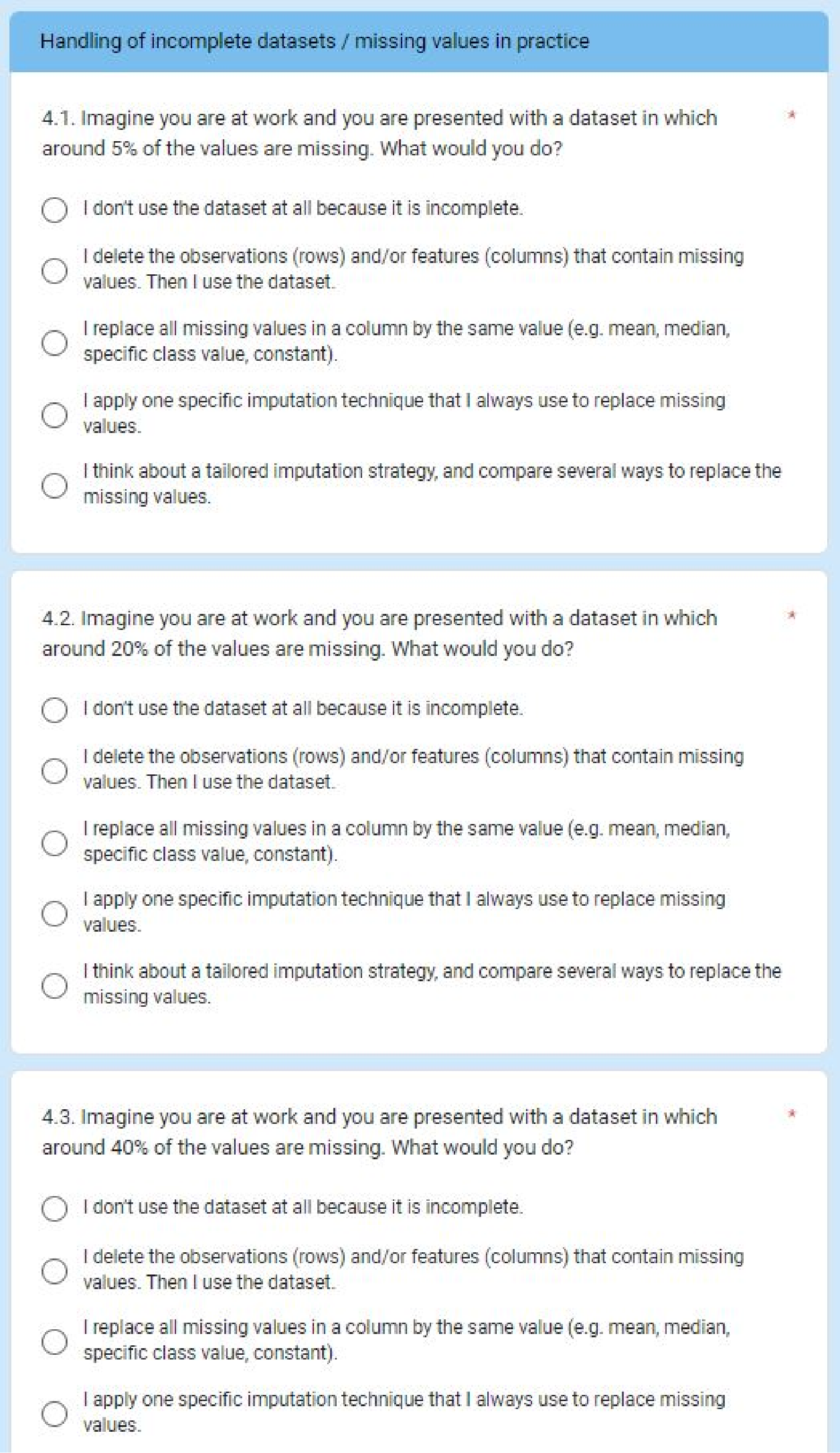
3.9. Please let us know about your gender:

○ Female

○ Male

○ Other

○ I do not want to share this information



Handling of incomplete datasets / missing values in practice

4.1. Imagine you are at work and you are presented with a dataset in which around 5% of the values are missing. What would you do? *

○ I don't use the dataset at all because it is incomplete.

○ I delete the observations (rows) and/or features (columns) that contain missing values. Then I use the dataset.

○ I replace all missing values in a column by the same value (e.g. mean, median, specific class value, constant).

○ I apply one specific imputation technique that I always use to replace missing values.

○ I think about a tailored imputation strategy, and compare several ways to replace the missing values.

4.2. Imagine you are at work and you are presented with a dataset in which around 20% of the values are missing. What would you do? *

○ I don't use the dataset at all because it is incomplete.

○ I delete the observations (rows) and/or features (columns) that contain missing values. Then I use the dataset.

○ I replace all missing values in a column by the same value (e.g. mean, median, specific class value, constant).

○ I apply one specific imputation technique that I always use to replace missing values.

○ I think about a tailored imputation strategy, and compare several ways to replace the missing values.

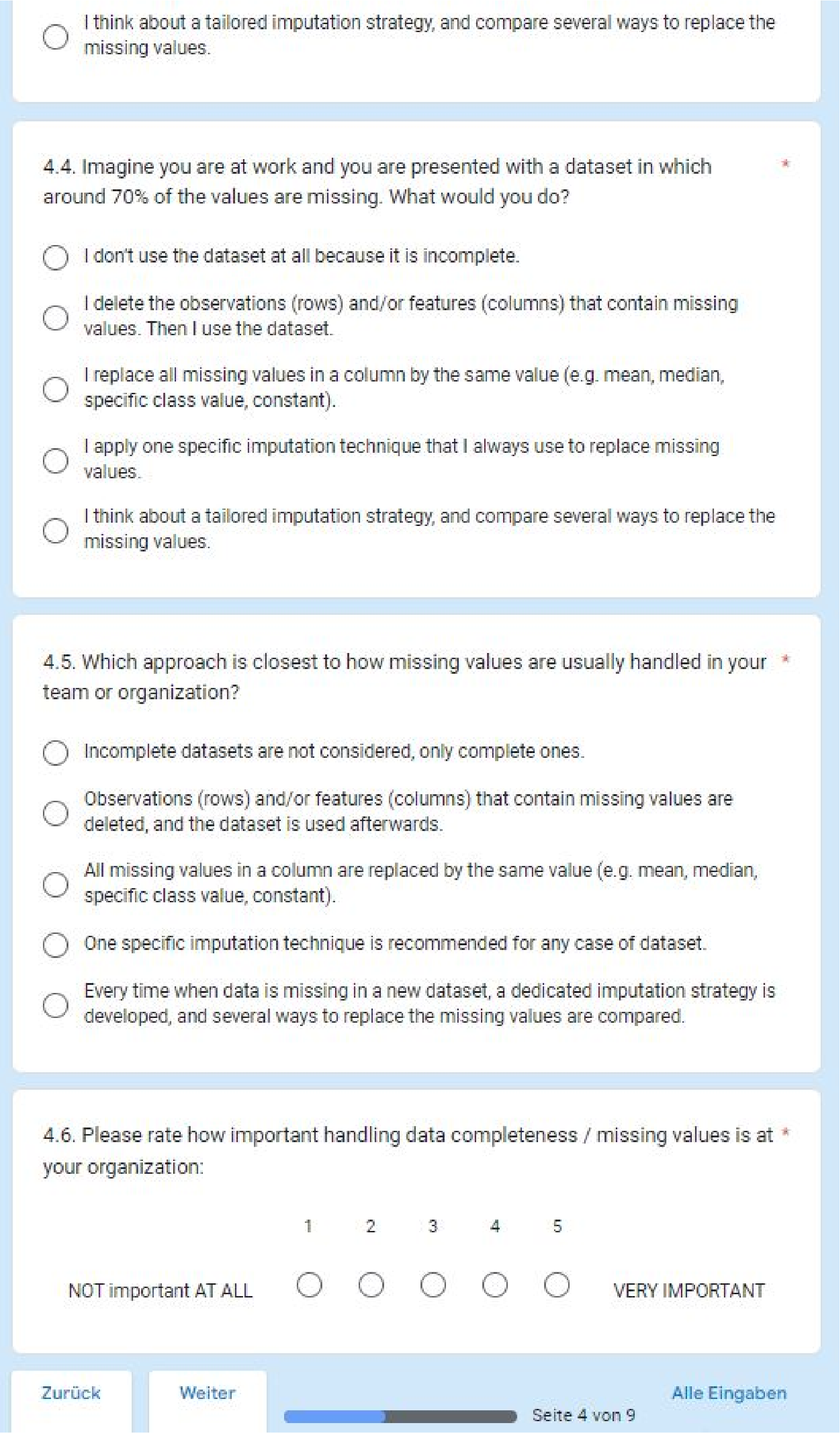
4.3. Imagine you are at work and you are presented with a dataset in which around 40% of the values are missing. What would you do? *

○ I don't use the dataset at all because it is incomplete.

○ I delete the observations (rows) and/or features (columns) that contain missing values. Then I use the dataset.

○ I replace all missing values in a column by the same value (e.g. mean, median, specific class value, constant).

○ I apply one specific imputation technique that I always use to replace missing values.



○ I think about a tailored imputation strategy, and compare several ways to replace the missing values.

4.4. Imagine you are at work and you are presented with a dataset in which around 70% of the values are missing. What would you do? *

○ I don't use the dataset at all because it is incomplete.

○ I delete the observations (rows) and/or features (columns) that contain missing values. Then I use the dataset.

○ I replace all missing values in a column by the same value (e.g. mean, median, specific class value, constant).

○ I apply one specific imputation technique that I always use to replace missing values.

○ I think about a tailored imputation strategy, and compare several ways to replace the missing values.

4.5. Which approach is closest to how missing values are usually handled in your team or organization? *

○ Incomplete datasets are not considered, only complete ones.

○ Observations (rows) and/or features (columns) that contain missing values are deleted, and the dataset is used afterwards.

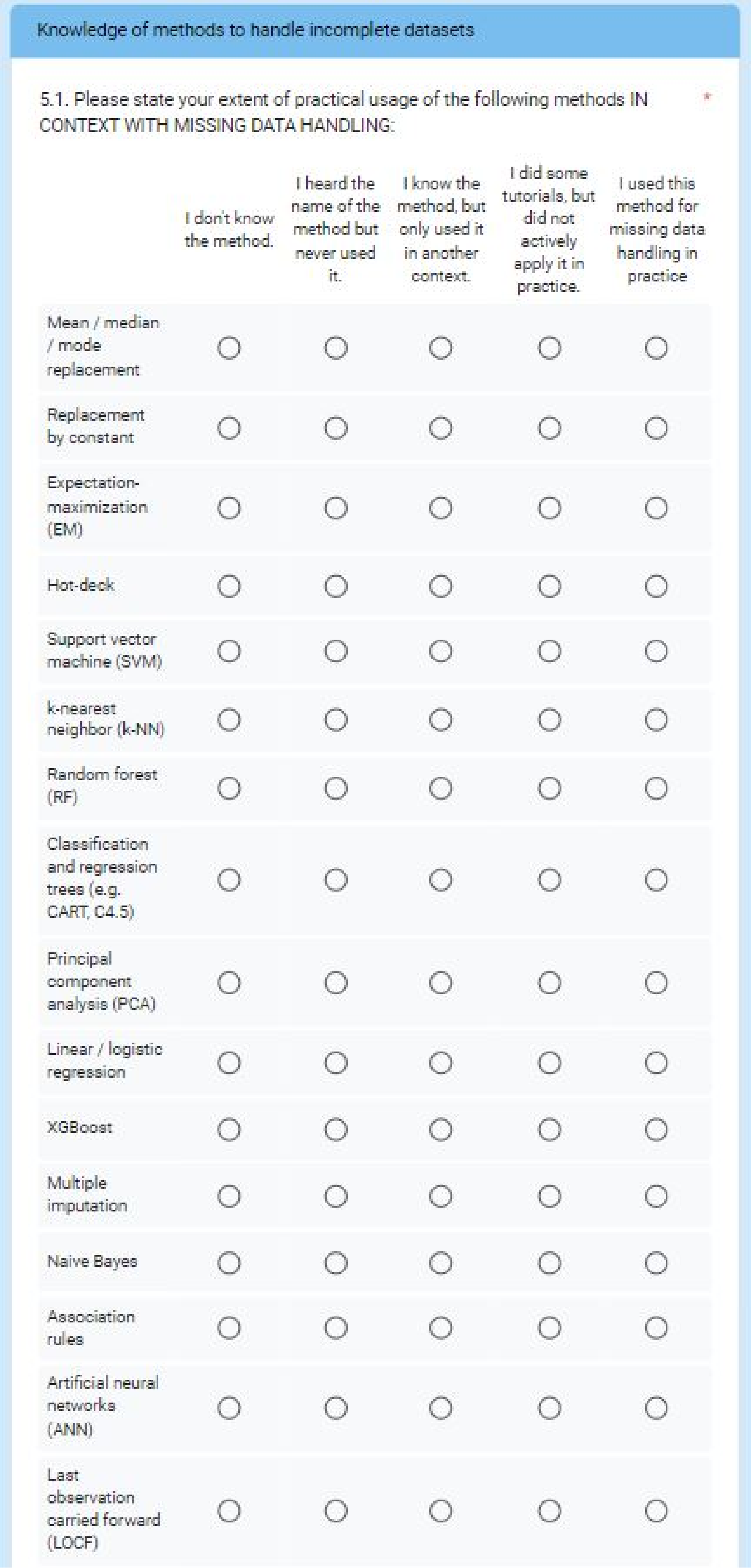
○ All missing values in a column are replaced by the same value (e.g. mean, median, specific class value, constant).

○ One specific imputation technique is recommended for any case of dataset.

○ Every time when data is missing in a new dataset, a dedicated imputation strategy is developed, and several ways to replace the missing values are compared.

4.6. Please rate how important handling data completeness / missing values is at your organization: *

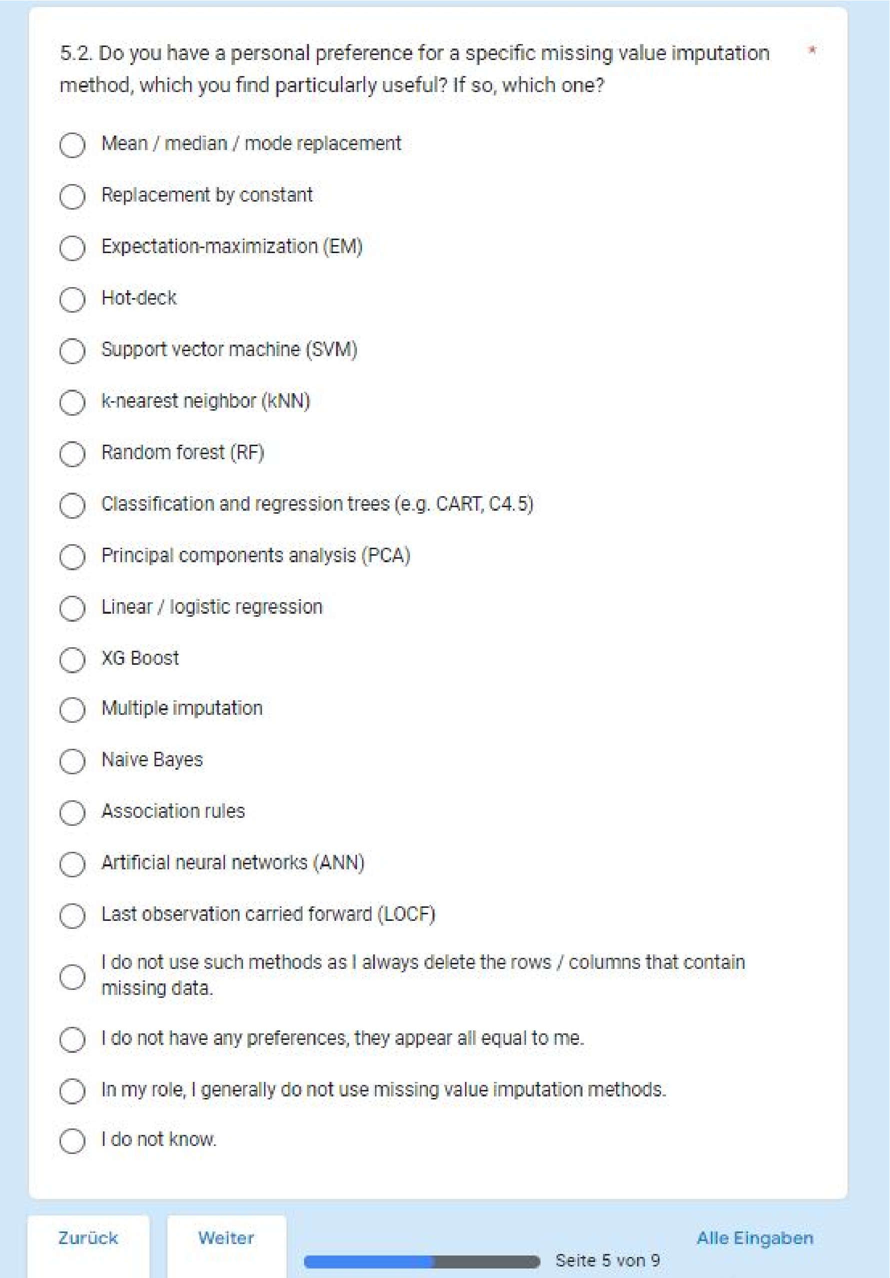
	1	2	3	4	5	
NOT important AT ALL	○	○	○	○	○	VERY IMPORTANT



Knowledge of methods to handle incomplete datasets

5.1. Please state your extent of practical usage of the following methods IN CONTEXT WITH MISSING DATA HANDLING: *

	I don't know the method.	I heard the name of the method but never used it.	I know the method, but only used it in another context.	I did some tutorials, but did not actively apply it in practice.	I used this method for missing data handling in practice
Mean / median / mode replacement	○	○	○	○	○
Replacement by constant	○	○	○	○	○
Expectation-maximization (EM)	○	○	○	○	○
Hot-deck	○	○	○	○	○
Support vector machine (SVM)	○	○	○	○	○
k-nearest neighbor (k-NN)	○	○	○	○	○
Random forest (RF)	○	○	○	○	○
Classification and regression trees (e.g. CART, C4.5)	○	○	○	○	○
Principal component analysis (PCA)	○	○	○	○	○
Linear / logistic regression	○	○	○	○	○
XGBoost	○	○	○	○	○
Multiple imputation	○	○	○	○	○
Naive Bayes	○	○	○	○	○
Association rules	○	○	○	○	○
Artificial neural networks (ANN)	○	○	○	○	○
Last observation carried forward (LOCF)	○	○	○	○	○



5.2. Do you have a personal preference for a specific missing value imputation method, which you find particularly useful? If so, which one? *

○ Mean / median / mode replacement

○ Replacement by constant

○ Expectation-maximization (EM)

○ Hot-deck

○ Support vector machine (SVM)

○ k-nearest neighbor (kNN)

○ Random forest (RF)

○ Classification and regression trees (e.g. CART, C4.5)

○ Principal components analysis (PCA)

○ Linear / logistic regression

○ XG Boost

○ Multiple imputation

○ Naive Bayes

○ Association rules

○ Artificial neural networks (ANN)

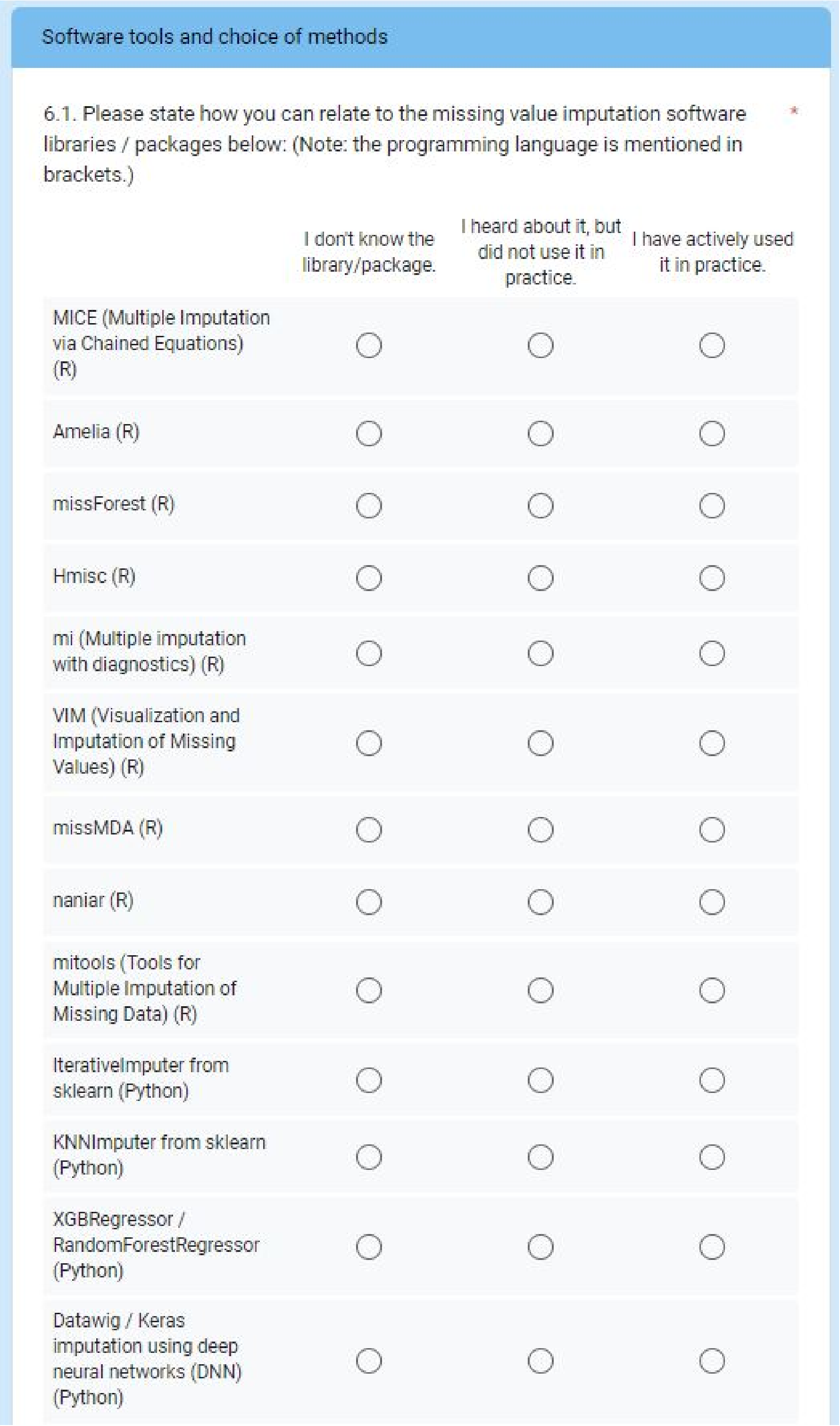
○ Last observation carried forward (LOCF)

○ I do not use such methods as I always delete the rows / columns that contain missing data.

○ I do not have any preferences, they appear all equal to me.

○ In my role, I generally do not use missing value imputation methods.

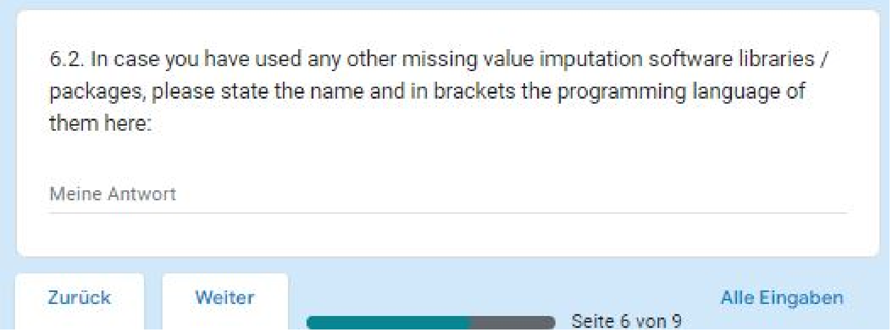
○ I do not know.



Software tools and choice of methods

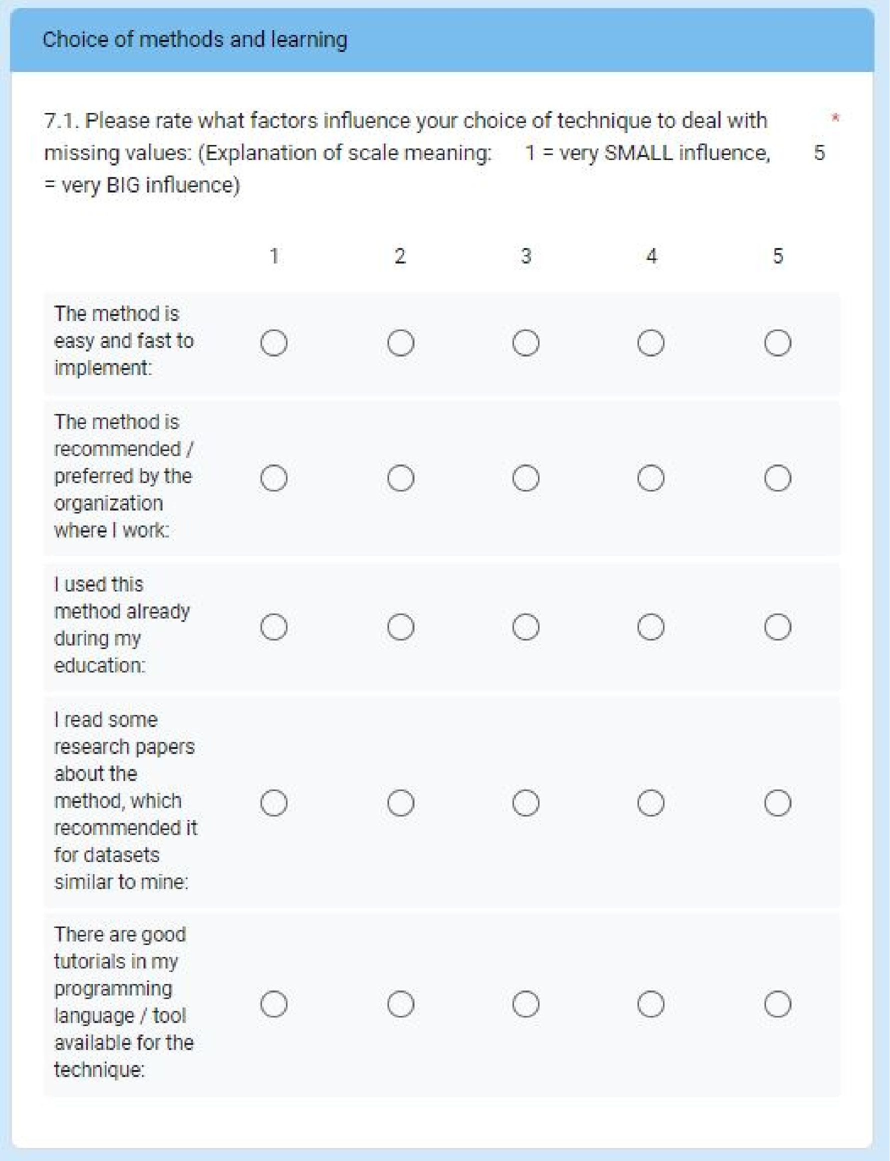
6.1. Please state how you can relate to the missing value imputation software libraries / packages below: (Note: the programming language is mentioned in brackets.) *

	I don't know the library/package.	I heard about it, but did not use it in practice.	I have actively used it in practice.
MICE (Multiple Imputation via Chained Equations) (R)	○	○	○
Amelia (R)	○	○	○
missForest (R)	○	○	○
Hmisc (R)	○	○	○
mi (Multiple imputation with diagnostics) (R)	○	○	○
VIM (Visualization and Imputation of Missing Values) (R)	○	○	○
missMDA (R)	○	○	○
naniar (R)	○	○	○
mitools (Tools for Multiple Imputation of Missing Data) (R)	○	○	○
IterativeImputer from sklearn (Python)	○	○	○
KNNImputer from sklearn (Python)	○	○	○
XGBRegressor / RandomForestRegressor (Python)	○	○	○
Datawig / Keras imputation using deep neural networks (DNN) (Python)	○	○	○



6.2. In case you have used any other missing value imputation software libraries / packages, please state the name and in brackets the programming language of them here:

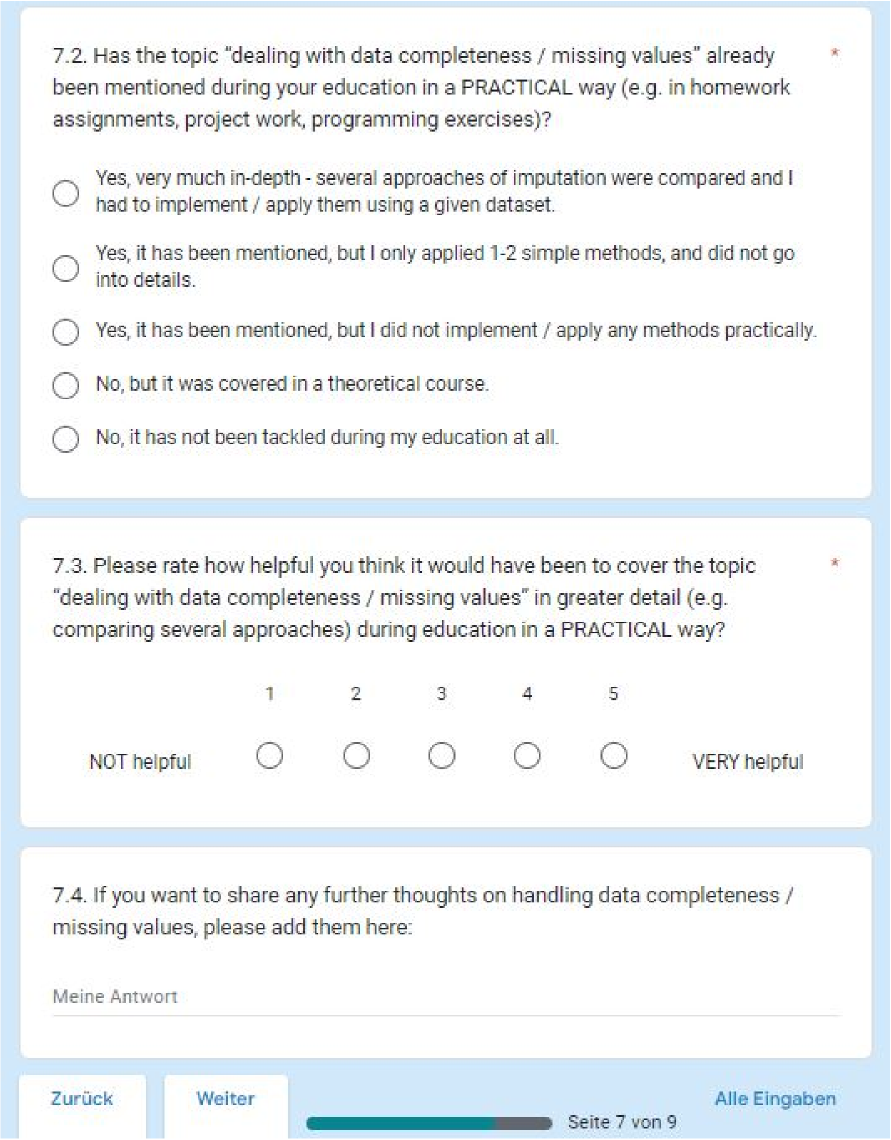
Meine Antwort



Choice of methods and learning

7.1. Please rate what factors influence your choice of technique to deal with missing values: (Explanation of scale meaning: 1 = very SMALL influence, 5 = very BIG influence) *

	1	2	3	4	5
The method is easy and fast to implement:	○	○	○	○	○
The method is recommended / preferred by the organization where I work:	○	○	○	○	○
I used this method already during my education:	○	○	○	○	○
I read some research papers about the method, which recommended it for datasets similar to mine:	○	○	○	○	○
There are good tutorials in my programming language / tool available for the technique:	○	○	○	○	○



7.2. Has the topic "dealing with data completeness / missing values" already been mentioned during your education in a PRACTICAL way (e.g. in homework assignments, project work, programming exercises)? *

○ Yes, very much in-depth - several approaches of imputation were compared and I had to implement / apply them using a given dataset.

○ Yes, it has been mentioned, but I only applied 1-2 simple methods, and did not go into details.

○ Yes, it has been mentioned, but I did not implement / apply any methods practically.

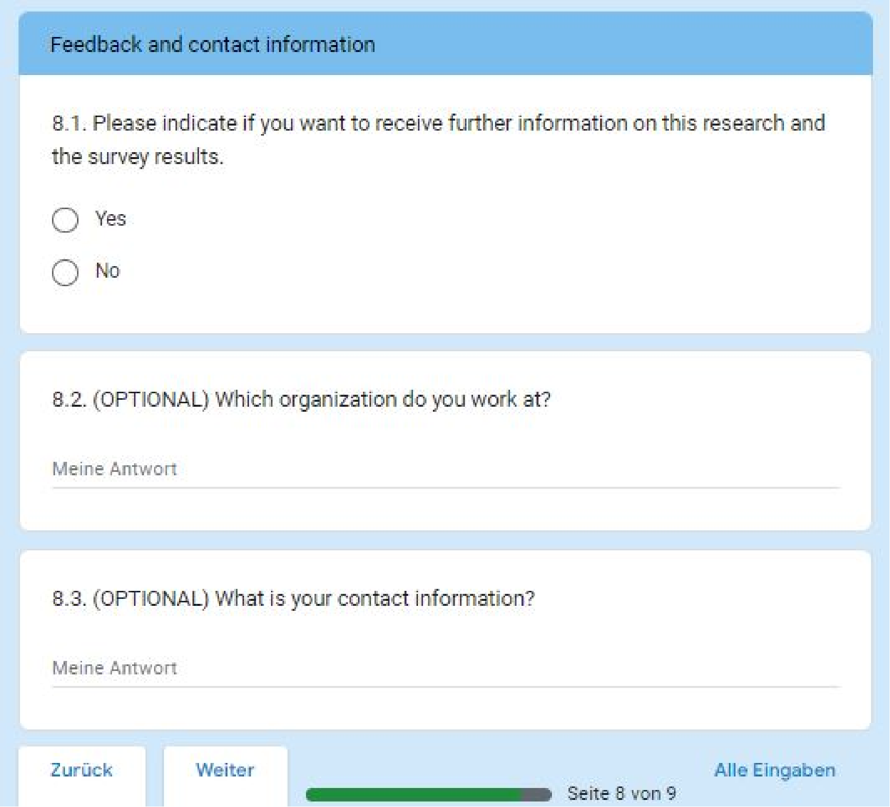
○ No, but it was covered in a theoretical course.

○ No, it has not been tackled during my education at all.

7.3. Please rate how helpful you think it would have been to cover the topic "dealing with data completeness / missing values" in greater detail (e.g. comparing several approaches) during education in a PRACTICAL way? *

	1	2	3	4	5	
NOT helpful	○	○	○	○	○	VERY helpful

7.4. If you want to share any further thoughts on handling data completeness / missing values, please add them here:



Feedback and contact information

8.1. Please indicate if you want to receive further information on this research and the survey results.

○ Yes

○ No

8.2. (OPTIONAL) Which organization do you work at?

Meine Antwort

8.3. (OPTIONAL) What is your contact information?

Meine Antwort