TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Diplomarbeit

Design and Development of Automatic Recommendation Generation Module of Prescriptive Maintenance Model (AutoPriMa)

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs

unter der Leitung von

Univ.-Prof. Dr.-Ing. Dipl. Wirtsch.-Ing. Prof. eh. Dr. h.c. Wilfried Sihn

(E330 Institut für Managementwissenschaften, Bereich: Betriebstechnik, Systemplanung und Facility Management

Dr.-Ing. Fazel Ansari

(E330 Institut für Managementwissenschaften, Bereich: Betriebstechnik, Systemplanung und Facility Management, Research Group of Smart and Knowledge-Based Maintenance)

eingereicht an der Technischen Universität Wien

Fakultät für Maschinenwesen und Betriebswissenschaften

von

Ing. Linus Kohl, BSc

1028804

Ernst-Melchior-Gasse 11/3/10

1020 Wien

Ort, Datum

Vorname Nachname

TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Design and Development of Automatic Recommendation Generation Module of Prescriptive Maintenance Model (AutoPriMa)

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Ort, Datum

Vorname Nachname

Danksagung

Mein Dank gilt allen Personen die mich beim Umsetzung der vorliegenden Diplomarbeit unterstützt haben.

Besondern möchte ich Herrn Univ.-Prof. Dr.-Ing. Dipl. Wirtsch.-Ing. Prof. eh. Dr. h.c. Wilfried Sihn und Herrn Dr.-Ing. Fazel Ansari vom Institut für Managementwissenschaften, der Forschungsgruppe Smart and Knowledge-Based Maintenance danken. Ich möchte mich an dieser Stelle vor allem für die Möglichkeit bedanken eine Arbeit über ein so spannendes Thema verfassen zu dürfen. Ebenfalls möchte ich mich für das hilfreiche und reichhaltiges Feedback bei allen Fragen sehr herzlich bedanken.

Meinen Kommilitonen und Arbeitskollegen möchte ich an dieser Stelle auch meinen Dank aussprechen, welche immer ein offenes Ohr für Fragen hatten.

Weiters möchte ich mich auch bei meinen Eltern bedanken, die mir die Möglichkeit gegeben haben nzu studieren, sonder auch in meiner Studienzeit während allen Höhen und Tiefen zur Seite gestanden sind. Bei meiner Freundin möchte ich mir sehr für ihre Geduld und unterstützt, nicht nur während dieser Diplomarbeit, sonder auch in den Jahren davor bedanken.

Contents

Abstract

Industry 4.0 creates a change in maintenance. Due to the rise of Cyber-Physical Production Systems (CPPS) and the availability of sensor data, maintenance was changed from descriptive to prescriptive maintenance. The Internet of Things (IoT), Data Science and Artifical Intelligence (AI) all play a vital role in the development of manufacturing technology[1]. Predictive and prescriptive maintenance is expected to grow by approximately 39% to a total of, 10.96$B by 2022[2]. Smart Manufacturing Leadership Coalition (SMLC) has also predicted that the following targets can be achieved by data driven analytics in smart manufacturing (1), 30% reduction in capital intensity, (2) up to 40% reduction in product cycle times, and (3) overarching positive impact across energy and productivity[3]. Lueth K. et al. (2016) stated in their report that 79% of all decision makers of Original Equipment Manufacturers will see predictive and prescriptive maintenance as one of the most important applications in the next 1-3 years.

In the area of prescriptive analytics the goal is to find the best course of action for a given problem, by using techniques like recommendation engines and neural networks[4] for solving a problem. Those techniques can then be converted for use in maintenance. A rising demand for prescriptive maintenance, which offers decision support can be anticipated, while currently predictive maintenance mostly consists of inappropriate maintenance strategies and conditions[5]. According to Cheng et al. (2018) and R. Ranjan (2014) state of the art decision-making processes combine different data sources with data science methods to either improve the system intelligence[6][7] or establish an automated big data pipeline[8] Cheng et al. (2018) and R. Ranjan (2014). The concept Knowledge Based Maintenance (KBM)[9][10][11][12] is a key enabler for digital transformation to prescriptive maintenance.

As stated by Ansari, Glawar, et al. (2019), the PriMa model and its four-step methodology have been introduced and an applied as part of a proof-of-concept study, however while the paper specifies the methodology and approach in detail, it does not go into detail on how to achieve problem 1 (P1) the data input into the data warehouse, problem 2 (P2) how to build aggregator functions and most importantly, how to handle the feedback loop between the Knowledge-Base and the Decision Support Dashboard problem 3 (P3).

This works aims to design an automated PriMa model, specifically focusing on the knowledge pipeline from the textual data from maintenance reports to the recommendation of a solution for the problem identified in the report. These questions have been answered by looking into the requirements given by O'Donovan et al. (2015) for the data ingest process and proposing an own requirement list for a data warehouse solution (P1). In the next step three machine learning (ML) algorithms, namely Hamilton Monte-Carlo (HMC), Random Forest (RF) and Neural Networks (NN) reasoning have been generated

[1] Sharma et al., 2014.
[2] Analytics, 2017.
[3] Coalition, 2011.
[4] Gartner, 2019b.
[5] Ansari, Glawar, et al., 2019.
[6] Nemeth et al., 2018.
[7] Betti et al., 2019.
[8] O'Donovan et al., 2015.
[9] Ansari, Glawar, et al., 2019.
[10] Matyas, 2018.
[11] Ansari, Khobreh, et al., 2018.
[12] Ansari, Uhr, et al., 2014.

and minimum working examples provided. Their outputs later on have been aggregated by a weighted hybrid function (P2). For the knowledge pipeline a Natural Language Processing (NLP) algorithm was applied which uses maintenance reports and extracts nouns and verbs. Those than can be matched against an ontology by using case-based reasoning (CBR) with the help of SPARQL. To sum up, the present thesis contributes on design and technical realization of the knowledge pipeline in the context of maintenance by analyzing technical requirements and developing a proof of concept demonstrator.

Kurzfassung

Mit Industry 4.0 wurde eine neue Ära in der Instandhaltung eingeleitet. Mit dem Aufkommen von Cyber-Physical Production Systems (CPPS) und der ständigen Verfügbarkeit von Sensordaten änderte sich die Wartung von der vorausschauenden zur präskriptiven Instandhaltung. Internet of Things (IoT), Data Science und Artifical Intelligence (AI) spielen daher eine wichtige Rolle bei der Entwicklung von Fertigungstechnologien[13]. Es wird erwartet, dass die vorschreibende und präskriptive Instandhaltung bis 2022[14] um etwa 39% auf jährlich 10,96$ Milliarden wächst. Es wird auch von der Smart Manufacturing Leadership Coalition (SMLC) dargelegt, dass die folgenden Ziele durch datengesteuerte Analysen in der intelligenten Fertigung erreicht werden können (1) 30% Reduzierung der Kapitalintensität, (2) bis zu 40% Reduzierung der Produktzykluszeiten und (3) übergreifende positive Auswirkungen auf Energie und Produktivität[15]. Lueth K. et al. (2016) erklärten in ihrem Bericht, dass 79% aller Entscheidungsträger von Original Equipment Manufacturers die vorausschauende und präskriptiven Instandhaltung als eine der wichtigsten Entwicklungen in den nächsten 1-3 Jahren sehen.

In der prädiktiven Datenanalyse ist es das Ziel die best mpglichste Handlungsalternative zu finden um ein gegebenes Problem mit Hilfe von Techniken wie Empfehlungsdienst und Neuronalen Netzwerk[16] zu lösen. Während die prädiktive Instandhaltung in der aktuellen Situation meist aus unangemessenen Instandhaltungsstrategien und -bedingungen besteht[17], versucht die präskriptive Instandhaltung mit modernsten Entscheidungsprozessen verschiedene Datenquellen zu kombinieren und mit Data Science Methoden zur Verbesserung der Systemintelligenz[18][19][20], oder mit einer automatisierten Big Data Pipeline[21] Cheng et al. (2018) und R. Ranjan (2014) die Instandhaltungskennzahlen zu verbessern. Der Fehler ist fehlendes Wissen, so dass das Konzept Knowledge Based Maintenance (KBM)[22][23][24][25], ein Schlüsselfaktor für die digitale Transformation zur präskriptiven Instandhaltung sein kann.

Das PriMa-Modell und seine Vier-Schritte-Methodik wurde schon von Ansari, Glawar, et al. (2019) angewendet und an einem praktischen Beispiel erprobt. Während das Paper die Methodik und den Ansatz im Detail beschreibt, geht es nicht im Detail darauf ein, wie man Problem 1 (P1) die Dateneingabe in das Data Warehouse, Problem 2 (P2) den Aufbau von Aggregatorfunktionen und vor allem den Umgang mit der Feedbackschleife zwischen der Knowledge-Base und dem Decision Support Dashboard Problem 3 (P3) lösen kann.

Die genannten Fragen wurden in dieser Arbeit beantwortet, indem die Anforderungen von O'Donovan et al. (2015) an den Datenerfassungsprozess umgesetzt und eine eigene Anforderungsliste für eine Data Warehouse Lösung (P1) vorgeschlagen wurde. Im näch-

[13] Sharma et al., 2014.
[14] Analytics, 2017.
[15] Coalition, 2011.
[16] Gartner, 2019b.
[17] Ansari, Glawar, et al., 2019.
[18] Nemeth et al., 2018.
[19] Betti et al., 2019.
[20] J. Lee, Lapira, et al., 2013.
[21] O'Donovan et al., 2015.
[22] Ansari, Glawar, et al., 2019.
[23] Matyas, 2018.
[24] Ansari, Khobreh, et al., 2018.
[25] Ansari, Uhr, et al., 2014.

sten Schritt wurden drei ML-Algorithmen, nämlich ein Random Forest (RF), ein Neural Network (NN) und ein Bayes'sches Netzwerk generiert und Minimum Working Examles bereitgestellt. Ihre späteren Ergebnisse wurden durch eine gewichtete Hybridfunktion (P2) aggregiert. Für die Wissenspipeline wurde ein Natural Language Processing (NLP)-Algorithmus verwendet, der einen Instandhaltungsbericht als Input verwendet und Substantive und Verben extrahiert. Diese werden dann gegen eine Ontologie Datenbank abgeglichen. Dies geschieht mit Hilfe von CBR, was hier mit SPARQL umgesetzt wurde. Zusammenfassend lässt sich sagen, dass die vorliegende Arbeit einen Beitrag zum Design und zur technischen Realisierung der Knowledge Pipeline im Rahmen der Instandhaltung leistet, indem sie technische Anforderungen analysiert und einen Proof of Concept-Demonstrator entwickelt.

1 | Introduction

With Industry 4.0, a new era of manufacturing started and disrupted the whole value chain. With so called smart factories the possibly of lot size one arises. That offers a customer the possibility of ordering a car in a custom color, with certain car seats and even an individual chassis height. For producing goods in lot size one a wholistic production landscape must be planned, where all parts of the process from the machines to ordering process of the customer are connected and the gathered data is collected and analysed. This shows how many different areas are influenced by the changes triggered by Industry 4.0. In the ideal case all of those connected machines and services collect data in order to gain more insights. These insights can be leveraged to develop better manufacturing techniques as well as more efficient maintenance strategies.

Information from the production process and indeed all areas can be collected and evaluated. For this thesis, data concerning maintenance will be tge if primary interest. The term smart maintenance, or maintenance 4.0 derives from the word Industry 4.0[1]. Prior to elaborating on maintenance 4.0, the term Industry 4.0 should be explained in more detail.

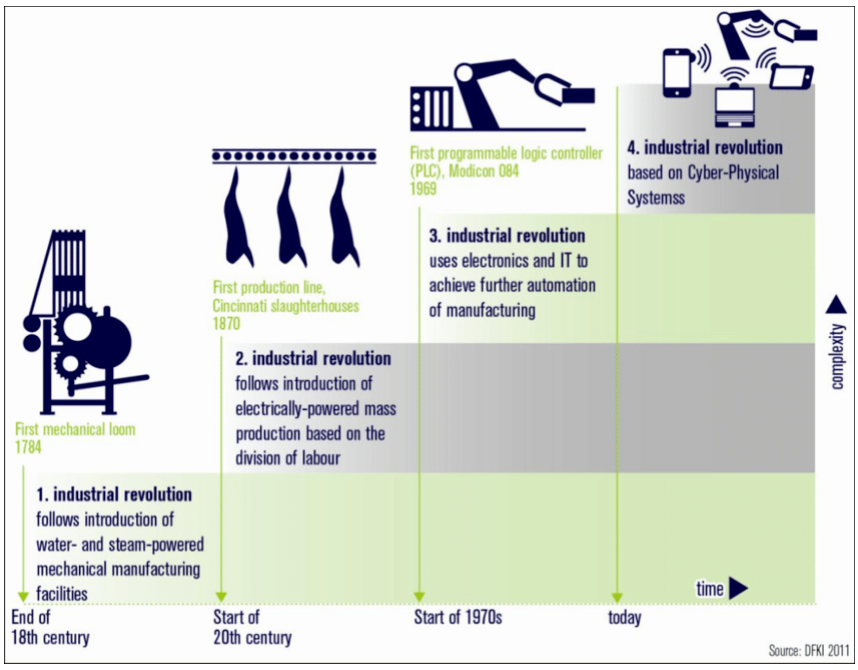


Figure 1.1: Industrie 4.0. Reprinted from Hirsch-Kreinsen (2016).

The term: "**Industrie 4.0**" originates from a project in the high tech of the German federal government, where they started a high tech strategy called "Industrie 4.0"[2]. The

[1] Matyas, 2018, p. 138-143.
[2] BMBF, 2017.

word "Industry 4.0" refers to the 4^{th} industrial revolution (Figure 1.1) driven by cyber physical systems, which can now be available in quantity with the IPv62 protocol[3] which offers many more addresses and enables direct networking between smart objects[4].

Manufacturing sector is undergoing a transformation towards tis fourth stage of industrialization. Where the first stage was the start of industrialization, machines powered by water and steam, the second revolution took place at the turn of the 20^{th} century when electrical power enabled mass production. In the third revolution, which started in the 1970s the use of information technology increased the amount of automation in manufacturing. So with the rise of wireless connected embedded systems which are connected to the Internet the merge of the physical and the digital world became reality. The automation of automation has also become a reality, which is reflected for example, in automated decision support. Industry 4.0 also led to a significant drop in costs in various sectors such as inventory, manufacturing, logistics and maintenance (Table 1.1).

Area	**Effect**	**Potential**
Inventory Costs	• Reduction of safety inventories • Avoidance of Bullwhip and Burbidge effects	-30% to -40%
Manufacturing Costs	• Improved OEE • Process circles • Vertical and horizontal personal flexibility	-60% to -70%
Logistics Costs	• Higher automation	-10% to -20%
Complexity Costs	• Extended performance ranges • Reduced troubleshooting	-60% to -70%
Quality Costs	• Near real time quality loops	-10% to -20%
Maintenance Costs	• Optimal stock • Condition-based maintenance • Dynamic priorisation	-20% to -30%

Table 1.1: Initial assessment of the potential cost reduction. Reprinted from Bauernhansl (2014).

Industry 4.0 is also changing the skills employers are looking for. With the changing environment, workers have to get used to lifelong learning[5]. This results in reskilling and upskilling on the job. One way to gain experience with cyber physical systems in smart factories is learning factories, these are useful for both academia and industry for research in smart production or new ways of hybrid learning which links world of work to academia[6]. The implementation of Industry 4.0, especially in Germany, focuses on three key areas[7]:

- **Horizontal integration through value networks** by creating forms of coopera-

[3]IPv62 is an internet protocol which allows enough IP addresses for complex networks
[4]Henning Kagermann, 2013.
[5]Schlund Sebastian, 2014.
[6]Ansari, Hold, et al., 2018.
[7]Henning Kagermann, 2013.

tion between different companies.

- **End-to-end engineering across the entire value chain** can be achieved by end-to-end integration throughout the engineering process in the whole value chain of the product.

- **Vertical integration and networked manufacturing systems** by integrating manufacturing systems in business and so creating digital connections which enable data from the sensors of the machines in the shop floor all the way up to the ERP systems.

Internet of Things (IoT) is an enabling technology in Industry 4.0, which itself is enabled by a technological surge in telecommunication, computer technologies in recent years. Those devices, make a production landscape possible where each machine is connected to a central hub. This allows continuous updates to all vital information from the production and logistics process, and this data bears relevant information for many applications. In maintenance it allows a real time glimpse into the current situation at the production plant. Those insights make predictive and prescriptive maintenance possible. More than ten years ago Gubbi et al., 2013 predicted that "The next wave in the era of computing will be outside the realm of the traditional desktop. In the Internet of Things (IoT) paradigm, many of the objects that surround us will be on the network in one form or another. Radio Frequency Identification (RFID) and sensor network technologies will rise to meet this new challenge, in which information and communication systems are invisibly embedded in the environment around us."

IoT was enabled by RDF, Bluetooth, WiFi, 4G-LTE and 5G, which will lead to even more use cases. 5G offers a larger bi-directional bandwith and is able to provide huge broadcasting data, supporting up to 60.000 connections[8]. IoT was identified as one of the emerging technologies eight years ago by Gartner's Hype Cycle Special Report for 2011 and was forecasted to be market ready in 5-10 years[9]. If Google trends (Figure 1.2) is taken into consideration the popularity of the search term IoT did not wane in the last 9 years, instead it steadily rose. So three IoT elements have been identified which enable seamless ubicomp[10]:

- **Hardware:** Sensors, actuators and embedded communication hardware

- **Middleware:** On demand storage and tools available for data analytics

- **Presentation (Visualization and Interpretation):** Novel and easy to understand visualization and interpretation tools

Cyber Physical Productionn Systems (CPPS) are enabled through IoT, which allows companies to incorporate their own network in their entire manufacturing process. Therefore systems like smart machines, warehousing systems and production facilities are summarized with the term Cyber Physical Systems. Cyber Physical Systems ... are integration of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. as stated by Lee[11]. CPS are "developed digitally and feature end-to-end ICT-based integration, from inbound logistics to production,

[8] Hossain, 2013.
[9] Gubbi et al., 2013.
[10] Gubbi et al., 2013.
[11] E. Lee et al., 2017, p. 7.

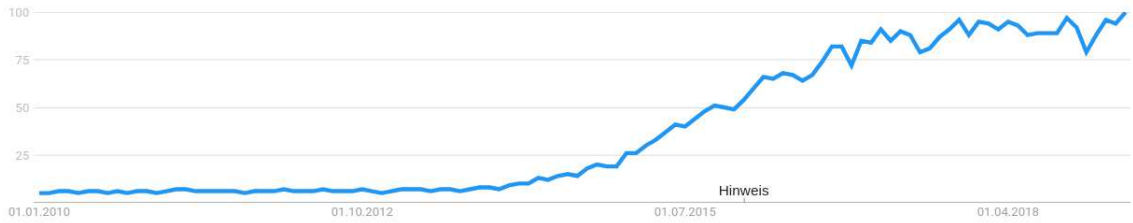


Figure 1.2: Google trends for IoT between 01.01.2010 and 12.04.2019. Reprinted from trends.google.com (2019).

marketing, outbound logistics and service"[12].

Previously disconnected entities, like machines, are now digitalized and network connected. In other words, a physical object like a machine can be augmented with two supplementary layers Figure (1.3), the cloud and service. Where the cloud enables exchange between the machine and other machines, users or services. In the service layer the real value of CPS is created. Here the data gathered by the machines is aggregated and algorithms transformed so new insights are gained.

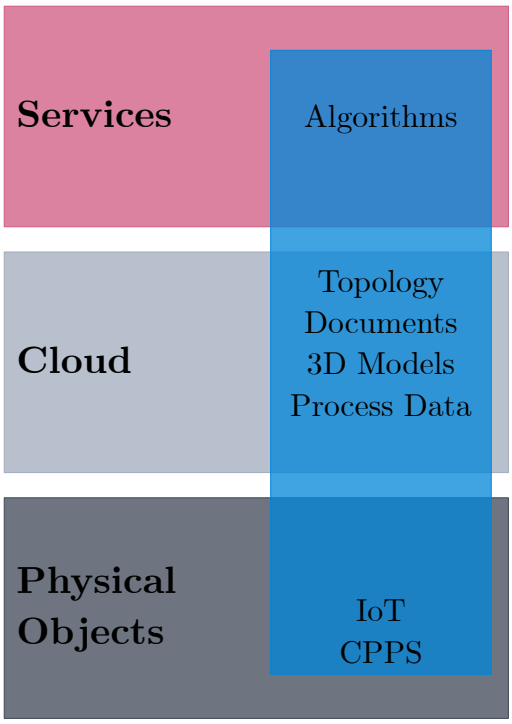


Figure 1.3: Three levels form a Cyber Physical System in Industrie 4.0 Drath et al. (2014).

CPPS are based on a 5C architecture as seen in Figure (1.4) introduced by Vogel-Heuser, Diedrich, et al. (2013). In the bottom layer "Smart Connection Level" the basic functionality of CPPS is defined. Self-connect and self-sensing is here seen as a critical feature for self aware information which helps them so self-predict their potential issues. The next step for CPPS is the "Cyber Level" where each machine creates its own twin

[12]Hirsch-Kreinsen, 2016, p. 14.

by using its features and information from its sensors[13]. This twin can be used for self-compare for peer-to-peer performance. Those self-assessments and self-evaluation are done and presented in a info graphic or dashboard. In the top level "Configuration" the machine can be reconfigured based on the priority and risk criteria to achieve the performance[14][15].

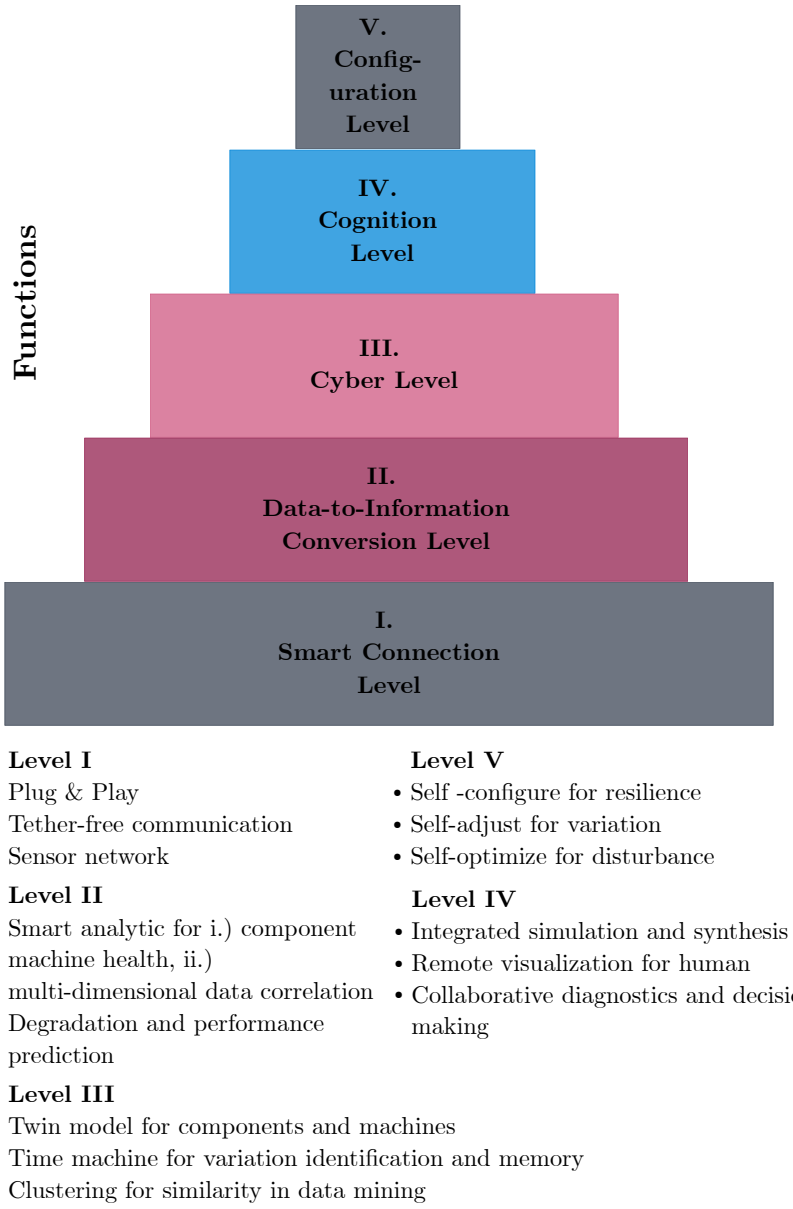


Figure 1.4: 5C Architecture of CPPS. Reprinted from Vogel-Heuser, J. Lee, et al. (2015) based on Vogel-Heuser, Diedrich, et al. (2013).

Industry 4.0 creates a change in maintenance. With the rise of (CPPS) and the availability of sensor data, maintenance changed from descriptive to prescriptive maintenance. IoT, Data Science and Artificial Intelligence (AI) play a vital role in the develop-

[13] Uhlemann et al., 2017.
[14] J. Lee, Bagheri, et al., 2014.
[15] J. Lee, Lapira, et al., 2013.

ment of manufacturing technology and operation management strategies[16]. Predictive and prescriptive maintenance is expected to grow about 39% to 10.96$B annuallyby 2022[17], or in case of Germany the transformation to Industry 4.0 could be worth up to 267B[18]. It is also outlined by Smart Manufacturing Leadership Coalition (SMLC) that the following targets can be achieved by data driven analytics in smart manufacturing (1) 30% reduction in capital intensity, (2) up to 40% reduction in product cycle times, and (3) overarching positive impact across energy and productivity[19].

In the report by Lueth K. et al., 2016[20] it is stated that 79% of all decision makers of Original Equipment Manufacturers see predictive and prescriptive maintenance as one of the most important applications in the next 1-3 years. Decision support systems are also mentioned by 58% as important.

A rising demand for prescriptive maintenance which offers decision support can be anticipated, as in the current situation predictive maintenance mostly consists of inappropriate maintenance strategies and conditions[21]. State of the art decision-making processes combine different data sources with data-science methods to either improve the system intelligence[22]][2324], or to establish an automated big data pipeline[25] Cheng et al., 2018 and R. Ranjan (2014). The concept Knowledge Based Maintenance (KBM)[26272829] is a key enabler for digital transformation to prescriptive maintenance.

The PriMa Model and its Four-Step methodology has been introduced and applied to a proof of concept study according to Ansari, Glawar, et al. (2019). While the paper specifies the methodology and approach in detail it does not go into detail on how to achieve problem 1 (P1) the data input into the data warehouse, problem 2 (P2) how to build aggregator functions and most importantly, how to handle the feedback loop between the Knowledge-Base and the Decision Support Dashboard problem 3 (P3). Therefore, this work aims on providing sufficient answers to open questions to enable a smooth implementation of PriMa.

1.0.1 | Problem Definition and Research Goal

With this thesis a comparison of state-of-the-art technologies, which can be used in realizing PriMa will be given. Prescriptive maintenance is an area where not only prior knowledge from maintenance and manufacturing come into play, but also a deep understanding for smart devices, IoT, Big Data, Machine Learning and consequently programming is needed.

[16] Sharma et al., 2014.
[17] Analytics, 2017.
[18] Heng, 2014.
[19] Coalition, 2011.
[20] Lueth K. et al., 2016.
[21] Ansari, Glawar, et al., 2019.
[22] Nemeth et al., 2018.
[23] Betti et al., 2019.
[24] J. Lee, Lapira, et al., 2013.
[25] O'Donovan et al., 2015.
[26] Ansari, Glawar, et al., 2019.
[27] Matyas, 2018.
[28] Ansari, Khobreh, et al., 2018.
[29] Ansari, Uhr, et al., 2014.

This makes prescriptive maintenance not only a challenging field, but also because of its interdisciplinary so promising for further investigations. So streaming technologies such as Kafka and RabbitMQ, and Data warehouse solutions like Amazon Web Service Google Cloud Platform (GCP), Microsoft Azure are evaluated. With these findings the mentioned problem (P1) will be answered. The answer will be provided by a comparison of popular streaming technologies and data warehouse technologies. With a technology review these technologies are also evaluated in terms of how well they can be combined with each other.

In case of the analytic toolbox of PriMa, methods like HMC, RF, NN and Text Mining as well as CBR will be discussed and presented with short code examples. The aim of these code examples is to show applicability for data sets in the maintenance sector. With findings from those applications the creation of an aggregation algorithm (P2) will be discussed. In this case, a literature review will be done and a code example will be presented with those findings. Here the literature review will provide the basis and aim, and aggregation algorithm will be developed based on these.

In the case of the Decision support, dashboard solutions will be discussed with a focus on their role in machine learning pipelines. After a technology analysis, those findings will be linked to the results of the following knowledge pipeline and how to implement this loop into the dashboard (P3).

The main focus of this research work is to establish the feedback loop, or knowledge pipeline, between the Maintenance Knowledge Base and the dashboard (P3). With the help of a literature review, the most important parts for creating and updating a Knowledge Base will be filtered and based on these specifications the knowledge pipeline will be build. Special attention will be given on how to generate knowledge from the information provided by the dashboard and on how to feed that information and its uncertainty back into the Knowledge Base. In particular the main research question regarding (P3) is: How to build a cutting-edge knowledge pipeline for prescriptive maintenance based on the four layer architecture of PriMa?

Sub questions for P1 and P2 are:

- How to stream continuous data from various sources into a data warehouse solution of Prima (i.e. Data Governance)?
- How to aggregate machine learning algorithm and how to validate their outcome? (i.e. From Data to Knowledge Intelligence)

The defined non-goal was to develop a software prototype.

1.0.2 | Methodology

As seen in Figure (1.5) a holistic view of all areas needed for PriMa[30] is given at the beginning of this thesis, and with this knowledge the requirement analysis for the technology reviews for (P1) and (P2) are made. Then each step of PriMa will be analysed and a Theory Building for the objectives of each part and a Solution Technology, see Figure (1.5), on how to implement this area will be given. The Naturalistic Evaluation, see Figure (1.5), of the PriMa Model will not be done because this has been done extensively by Ansari, Glawar, et al. (2019).

[30] Ansari, Glawar, et al., 2019.

In the first section, the architecture of PriMa will be explained. In the first layer two common state of the art data streaming tools, Kafka and RabbitMQ will be compared and analysed concerning their suitability to implementing PriMa. Also in Layer I, the needed storage solutions will be analysed. The main focus with data warehousing will lie in usability, compatibility, flexibility, extendibility and cost effectiveness. In this case, comparison of available technologies will be made without a deep literature review in place because those are mostly state of the art technologies and a mayor benefit here is their availability and potential for quick operational readiness. So a qualitative technology review will be done and the key findings will be used in the comparison of the streaming and warehouse solutions as part of the Theory Building and possible Artificial Evaluation, see Figure (1.5).

In the case of the Analytics Toolbox each algorithm will be compared regarding to findings in the qualitative literature review. So, for example, a confusion matrix will be calculated and compared to each other, on basis of the same data set, and in regard to its strength and weaknesses in the application and literature. For each algorithm a minimum working example (MWE) for the Solution Technology Invention, see Figure (1.5), will be provided to show its applicability. The mentioned Analytics Toolbox in Ansari, Glawar, et al., 2019 focuses on their application and thus, the focus will be in comparing those solutions without researching their specific finesse. After this ways of aggregation for those algorithms will be shown.

Furthermore dashboard solutions and their seamless integration into a potential pipeline will be compared. Again, as with data streaming and data warehousing the dashboarding is mainly focused on discussing openly available solutions. Further on a qualitative literature review on knowledge from data coming from continuous and unstructured data (free text) will be done and those results will be put into an MWE. Before this step a model Knowledge Base (KB) will be introduced, an algorithm on how to update this KB and then how to implement semantic-based Learning/Reasoning, again see Solution Technology Illustration (1.5). All of those mentioned points of the Knowledge Pipeline are designed to satisfy the feedback loop between the dashboard and the knowledge base (Knowledge Pipeline).

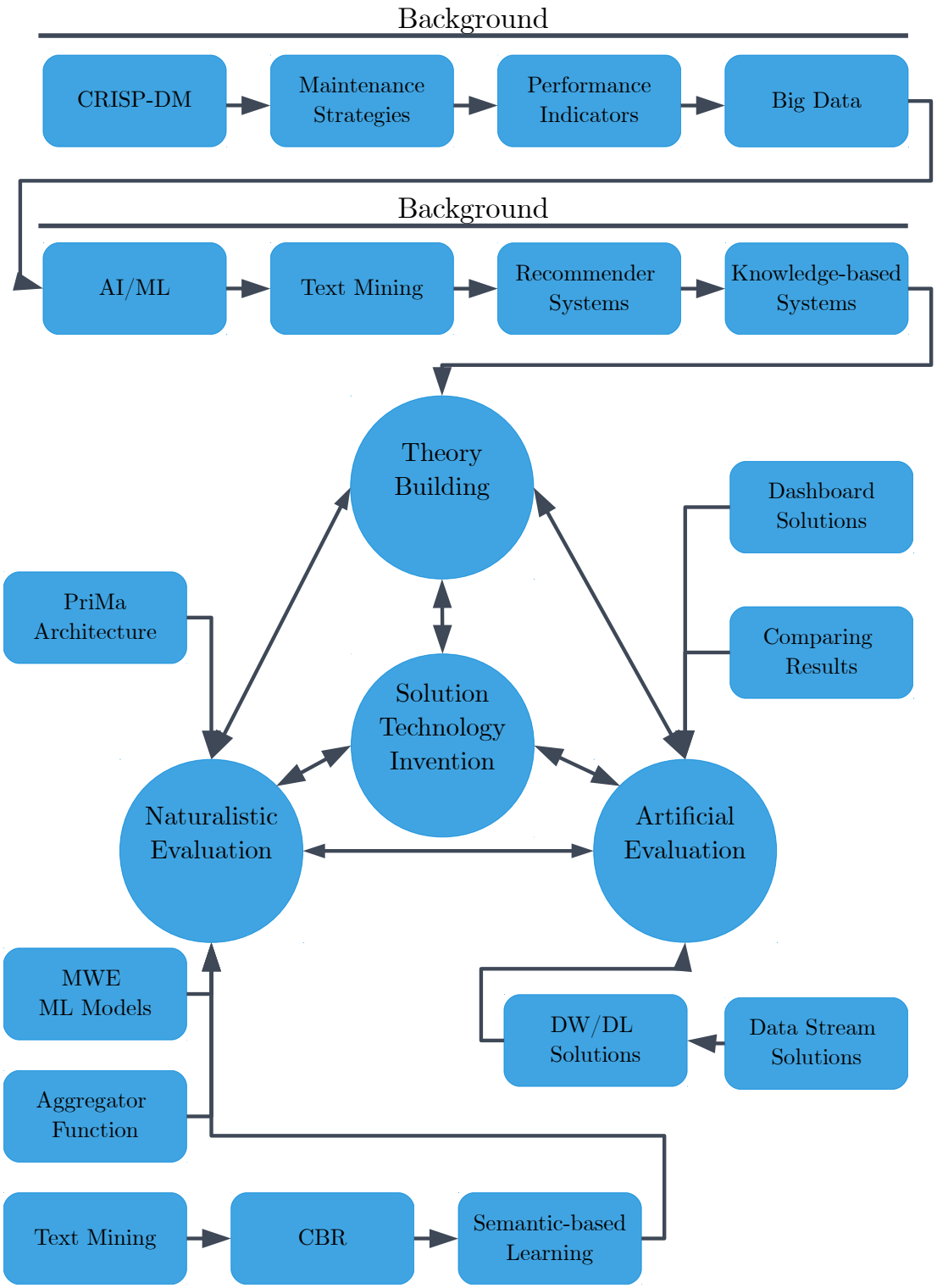


Figure 1.5: Methodology of the Master Thesis.

2 | Background

2.1 | CRISP-DM

CRISP-DM stands for CRoss-Industry Standard Process for Data Mining and it is a comprehensive data mining methodology and process model. "CRISP-DM breaks down the life cycle of a data mining project into six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment."[1]

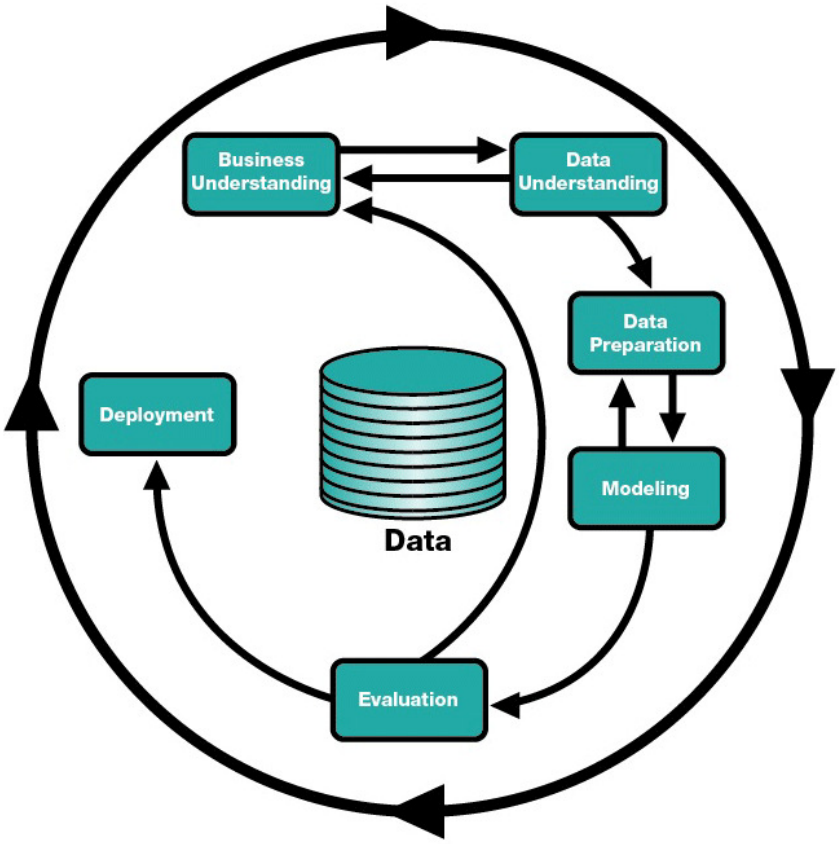


Figure 2.1: Process diagram showing the relationship between the different phases of CRISP-DM Chapman et al. (2000)

The phases of the CRISP-DM model are as described by Shearer (2000) and Chapman et al. (2000). The first similarity, that both are circular flow diagrams, is obvious, when looking at the two flow diagrams more in detail more similarities can be seen in the different steps.

[1] Shearer, 2000.

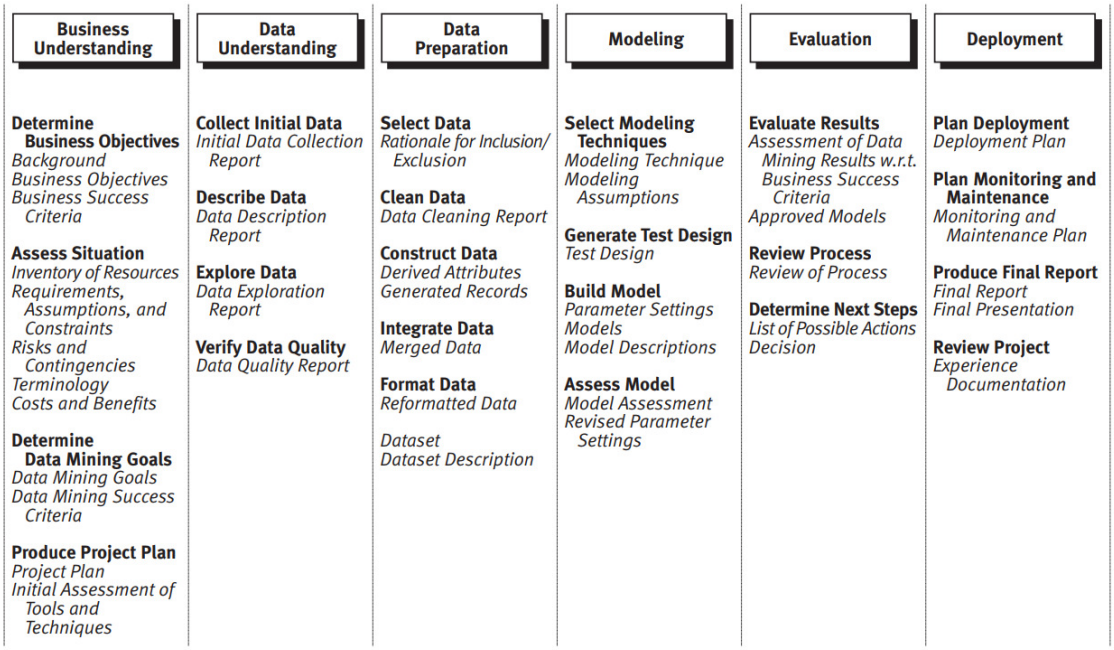


Figure 2.2: : Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model Chapman et al. (2000)

Phase I: Business Understanding
It is most essential to understand the project objectives from a business perspective and so to be able to convert this knowledge into a data mining problem. From there a preliminary plan can be developed to achieve these objectives. For example the primary business goal could be to determine when current customers are prone to move to a competitor.

When this primary business goal is fixed the next step is to determine the measure of success. This could be reducing lost customers by 10%. Unattainable goals should be avoided, each success criteria should be linked to one specific business objective.

The data mining goals state objectives in the project in business terms, if they can not be translated into data mining goals it should be considered to redefine the problem. It is essential for achieving the goals to produce a project plan, where the data mining goals are stated, steps are outlined, a timeline is proposed, a risk assessment is done and the used tools and techniques are stated.

The Business understanding and the Retrieve step are in principle the same, since in both cases it is a matter of gaining a deeper understanding of the problem. Both CRISP-DM and CBR focus on the problem and the understanding of the problem.

Phase II: Data Understanding
In this phase it is important to become familiar with the data to identify data quality problems, to discover initial insights, detect interesting subsets to form hypothesis about hidden information.

When collecting the data the analyst loads the data and if necessary integrates this data. Problems should be reported. While describing the data the analyst examines the surface properties, like format, quantity, number of records and fields in each table, the identities of the fields. When exploring the data the goal is to tackle the data mining ques-

tions by querying, visualizing and reporting. After this a data exploration report should be generated.

The data quality is very important and should be separately examined. Questions like is the data complete, are there missing values, are there missing attributes, blank fields, plausibility of values, consistent spelling of values and so on.

Phase III: Data Preparation
In this phase the dataset will be prepared to be fed into the modeling, so the transformation from the raw data via selection of attributes, transforming and cleaning the data for modeling tools. Those necessary five steps are selection of data, the cleansing of data, the construction of data, the integration of data, and the formatting of data.

The select data task must be decided if the data will be used for the analyses, based on certain criteria, including the relevance for the data mining goals, quality and technical constraints, like data volume and data types. When cleaning data subsets must be selected or missing data estimated through modeling analyses. It is important in this step to outline how each quality problem was addressed.

At the construct data task the analyst develops entirely new records (e.g. empty purchase) or creating derived attributes. When integrating the data often information from multiple tables or records are combined to create new records or values, but it involves also data aggregation. While formating data the analyst will change the format, or design, or if necessary remove illegal characters from strings, or trimming them to maximum length.

Phase IV: Modeling
In this phase various modeling techniques are selected and applied to the data sets and calibrated to optimal values. If a technique requires specific requirements, stepping back to the data preparation phase might be necessary. When choosing the modeling technique it is referred to choosing one or more specific modeling techniques like decision tree building, with C4.5 or neural network generation with back propagation.

After building the model the analyst must test the model in relation to quality, validity and running empirical tests to determine the strength of the model. In supervised data mining tasks such as classification it is common to use error rates as a quality measure. Then the data set is separated in a training and a testing dataset. So the model is trained on the training set and then later validated against the testing dataset, to see if it can predict the history before it is used to predict the future.

After testing the data analyst runs the model on the prepared data set to create models. While assessing the model the analyst interprets the model according to her domain knowledge, the success criteria and the desired test design. Business analyst should be included to interpret the results in a business context.

Phase V: Evaluation
Before deploying the model the analyst should evaluate the model and review its construction to see if it achieves the business objectives. In the step of evaluating the results the analyst summarizes the assessment in terms of business success criteria, and a final statement if the project meets the initial business objectives. In the review process it is important to go through the data mining engagement to determine if any important factors have been overlooked. Then it must be decided whether to finish the project or to

initiate further iterations in the project.

The Revision task and the Evaluation phase are again very similar and it is necessary to check whether the proposed solution really fits the objectives set. A metric should be considered to objectively evaluate the performance of the algorithm and compare it with alternatives later.

Phase VI: Deployment
After the creation the knowledge gained in the project must be organized and presented in a way that the customer can use it, so a strategy for the deployment must be made. OIf the data mining results becomes part of the day-to-day business monitoring and maintenance are important issues. In the end a final report must be made, it can be only a summary of the project and experiences or in is a more comprehensive presentation of the data mining results. After this the projects should be reviewed and the analyst should asses failures and success as well as ares of improvement for future projects.

2.2 | State of the Art Maintenance Strategies

As indicated in the CRISP-DM model, see chapter (2.1), the business understanding is the most essential task in order to convert this knowledge into a data mining problem. In the area of maintenance the maintenance strategy used gives away what types of data have to bee collected. This is important because only if the right data types are collected the necessary insights can be gained.

Maintenance strategies are rules that specify which actions are to be carried out on which aggregates or components at which times. The areas economic efficiency - safety - availability are important to minimize costs and maximize plant availability. Future trends predict fewer personnel to operate more complex plants. Indirect maintenance cost, especially downtime is often caused by a lack of transparency in processes and inadequate planning of maintenance activities. There is no uniform maintenance strategy that could be applied everywhere. The more plan intensive the operation the more continuous the production process the higher the downtime costs and the greater is the importance of maintenance[2][3].

The ideal maintenance concept includes an optimal mix of failure elimination, preventive maintenance, condition oriented maintenance and predictive maintenance. Those goals must be in maximal reliability with minimal costs at the same time. This very complex problem of which machine or plant are maintained needs an extensive analysis. The following criteria are impotent for such an analysis:

- Chaining of installations
- Redundancy of systems
- Validity of quality, environmental and safety standards
- Working time agreements
- Repair time

[2]Matyas, 2018, p. 119-120.
[3]Ben-Daya, 2009.

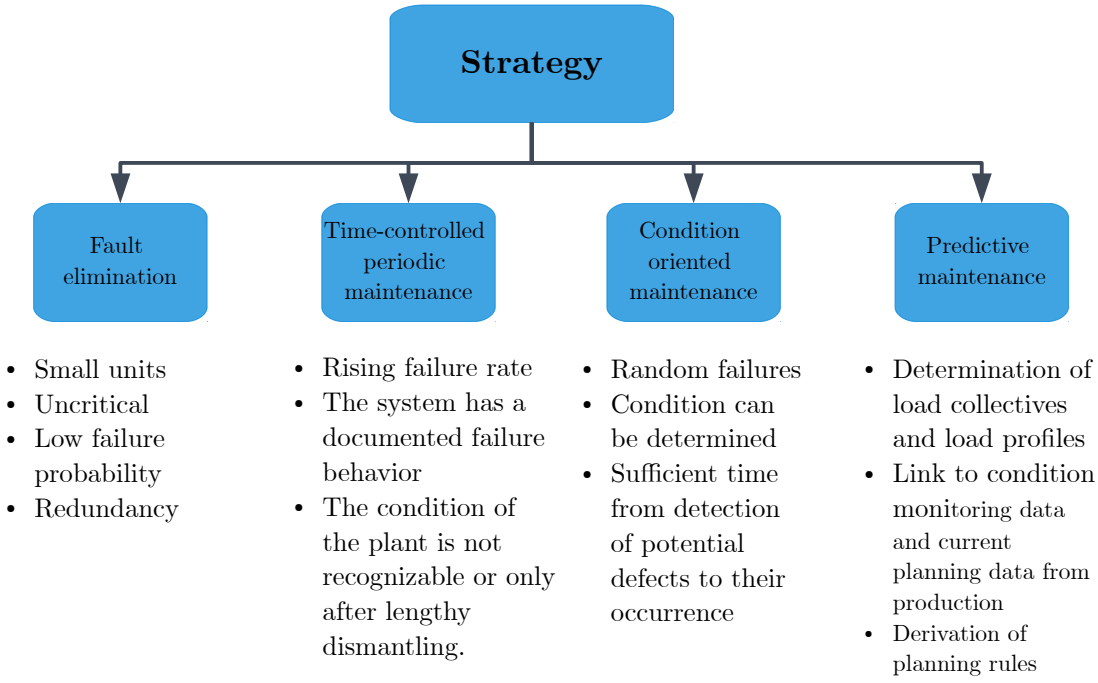


Figure 2.3: Maintenance strategies Matyas (2018, p. 120).

- Spare parts availability
- Material buffer between the plants
- Peak loads due to seasonal market demand and availability of raw material

2.2.1 | Fault elimination

With this method, the machines are operated without significant effort for inspection and maintenance until damage occurs. This often leads to the destruction of the machine, but at the same time a maximal maintenance interval is possible. Every standstill occurs unexpectedly and the failure is out of the operator's control. Operative planning in the production becomes nearly impossible and in the case of planned tasks (which are all other maintenance strategies) downtimes are smaller as seen in Figure (2.4). The concept of fault elimination is seldom useful in modern industry plants.

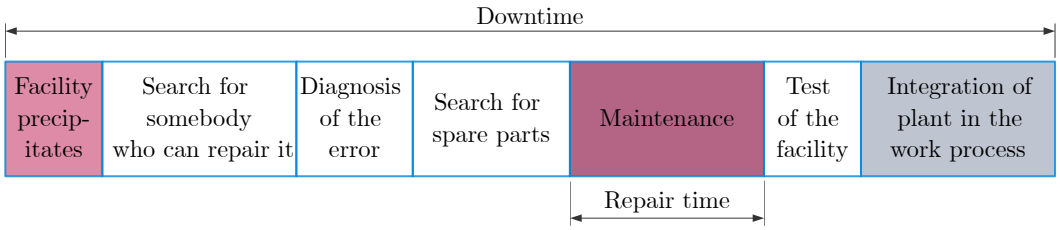


Figure 2.4: Plant downtime for unplanned measures Matyas (2018, p. 121).

2.2.1.1 Time-controlled periodic maintenance

Time-controlled periodic maintenance is a common method to preventively overhaul or replace certain parts when they have reacher a certain life span, regardless of their actual

condition. This course of action is meaningful if either the effects are to safety or the approximate life span of the product and the majority of the plant components remain functional.

On order to keep the level of maintenance costs and system downtime costs low it is necessary to plan preventive measures according to actual usage stock. A way to minimize those costs would be an early replacement 5 minutes before the damage occurs. Changing the replacement after the damage leads to a sudden increase in wear and tear of the other components. It can be assumed that planed repair measure can be carried out more quickly. The planning of measures to prevent damage and breakdowns is due to the following four unpleasant characteristics of damage and is extremely complex:

- Different "mean time between two damages"
- Different scattering of the service life
- Poor damage documentation
- Insufficient statistical experience of damage

The problem of determining the optimal (minimum cost) interval for Preventive maintenance measures can be solved by condition-based maintenance. To create of maintenance plans and maintenance strategies, knowledge of the mean time between failures, frequency distributions and previous damage experiences and documentation is of importance.

2.2.1.2 Condition-oriented maintenance

As can be derived from the described properties of damage a periodic repair is almost always uneconomical, because it must be carried out much more often than condition based maintenance. In addition, there is always the risk that preventive interventions could damage other previously functional parts. Exceptions when periodic maintenance is economical:

- Periodic alternation of fluid supplies
- Cleaning and renewal of filters which cannot be inspected
- The replacement of components, the repair costs of which are in relation to the downtime costs are very low

In condition-based maintenance, the maintenance measures are geared as closely as possible to the actual degree of wear and tear of the maintenance object. With suitable monitoring systems it is possible to inform about deviation from the required performance. This ensures that the maintenance interval is adjusted to the supply. Arguments for condition-based maintenance:

- Costs
- Risk of breaking something with preventive measures
- Diagnosis of occurred damages and reduction of consequential damages

As a rule, malfunctions are preceded by a certain warning before they occur which is described as a potential disruption. But the interval can be between a few milliseconds and some years. Condition related measures are taken when potential malfunctions become clear so that countermeasures can be made before damage occurs. Condition-based maintenance is based on the assumption that most malfunctions do not occur suddenly but develop over time and are preceded by warning signals. These signals are called potential interferences. They can be graphically represented in a so-called PF curve, Figure (2.5). P is the point at which a potential interference is detected and F is the downtime[4].

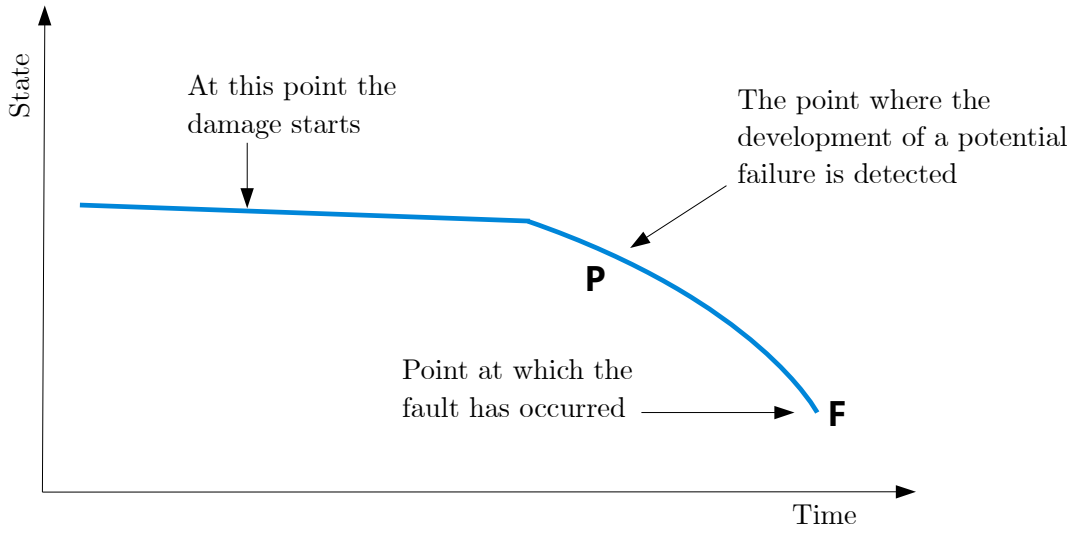


Figure 2.5: PF curve Matyas (2018, p. 125).

So condition-related measures are taken to determine when a potential malfunction will occur and countermeasures can be taken before damage occurs. So this time, which is between P and F in Figure (2.5) is an early warning time. The most challenging effort in condition-based maintenance is to find a way of predicting failures as early as possible. Correctly applied, condition monitoring is a very good method to avoid unexpected failures to prevent. After this step condition-maintaining measures should be applied so that the plant operates in an optimal level. Through early detection of damage, the safety of the system can be improved at any time. At the same time the operating costs can be kept as low as possible. Early detection of anomalies in operation can reduce costs in the following areas[5]:

- Operating costs
- Plant costs
- Repair costs
- Failure costs

2.2.1.3 Predictive maintenance

"Predictive maintenance is a management technique that, simply stated, uses regular evaluation of actual operating condition of plant equipment, production systems, and plant

[4]Matyas, 2018, p. 124-125.
[5]Matyas, 2018, p. 124-125.

management functions to optimize total plant operation."[6]

The three classical maintenance strategies - failure elimination, time-controlled periodic maintenance and condition-based maintenance often not sufficient enough nowadays due to the increasing complexity of production processes. Improved plant availability is usually accompanied by increased maintenance and repair costs, which often results in a waste of resources, as the maintenance measures are often initiated at the wrong time[7][p. 136 -137].

Deploying those classic techniques in mass production can still be done efficiently, but can not cope with customer driven production which does not result in load collectives. Above all in flexible manufacturing systems with a high variation of the production program and without fixed load collectives, there is a need for a foresighted, anticipatory and holistic maintenance strategy, which includes both sensor and control systems. The following factors are taken into account: the quality and machine data as well as the historical knowledge about failure events[8].

Four main principles for predictive maintenance have been identified by[9]:

- Preserving the system function
- Identification of the particular failure modes that can potentially cause functional failure
- Prioritizing key functional failures
- Selection of applicable and effective maintenance tasks for high priority items

The prognosis of wear and its effects on the manufacturing process is a central topic in the planning of maintenance strategies. Those models are mostly based on historical data, or from long term studies. Based on these data statistical models are build which can predict component wear. Traditional methods of quality management like six sigma use production or process data to make predictions for product quality. Those models do not take machine data, or production planning and control data into account[10].

So predictive maintenance is more than just measuring the operation condition of plants. Predictive maintenance permits accurate evaluation of all functional groups, like maintenance within a company. Properly applied, it can identify most, if not even all factors which limit the total plant efficiency[11]. The following tools are useful in predictive maintenance:

- Vibration monitoring and analysis
- Thermography
- Tribology
- Ultrasonics
- Operating Dynamics Analysis

[6]Mobley, 2014, Section 3.3.
[7]Matyas, 2018.
[8]Matyas, 2018, p. 136 -137.
[9]Ben-Daya, 2009, p. 400.
[10]Matyas, 2018, p. 136 -137.
[11]Mobley, 2014, section 3.3.

2.2.1.4 Smart Maintenance

At the moment the world of manufacturing is at a turning point. Industry 4.0 is becoming a reality and heralds a new age of manufacturing and therefore maintenance. So maintenance must become smart in order to give the right answers to those new questions. As introduced in chapter (1) smart devices in IoT which evolved in to CPPS can be connected to powerful computing power or cloud services and are so able to give maintenance staff insight into data in a way which was not possible before. The problems of the existing maintenance approaches are the partial view of condition-oriented strategies and the lack of anticipatory suitability of holistic strategies such as "Total Productive Maintenance" (TPM)[12]. Often the informations of machine conditions are incomplete or late which prevents from finding the exact time or optimum time for wear and tear which is based on the current status in production and the required production quality should be coordinated.

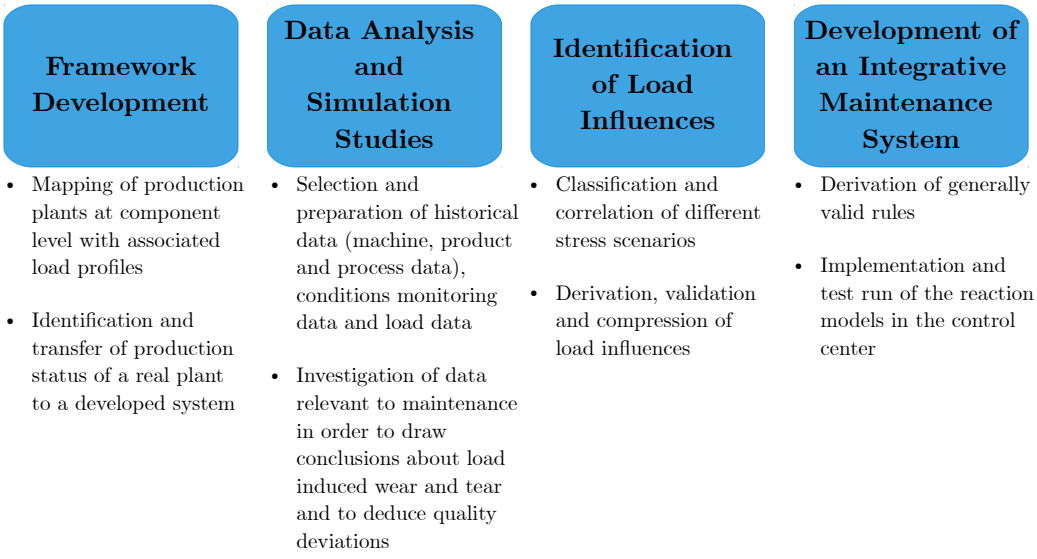


Figure 2.6: Practical implementation of anticipatory maintenance planning Matyas (2018, p. 142).

Additionally there are service life calculations for plant components, but they are not realistic because the real loads deviate from the theoretical load profile. Due to the poor quality of maintenance data and their inadequate linking and evaluation, it has proved difficult to derive a minimum from the maintenance costs and overall system downtime costs. The correlation of these data in connection with suitable system models can, however, be used directly as a basis for decision-making for the optimization of the system maintenance times, product quality and energy consumption.

So good data collection is a key element to smart maintenance[13]. To collect the right data critical components must be identified, physical parameters denided and the right monitors to be chosen so that the relevant data which represent system health are collected[14]. This can be achieved for example with WIFI technologies for remote data collection, cellular communication networks and compact microcomputer such as the Raspberry

[12] Matyas, 2018, p. 140.
[13] Matyas, 2018, p. 141-142.
[14] Gombé et al., 2017.

Pi[15].

Data mining techniques can be used for the recognition of meaningful patterns, correlations and exploitable knowledge in apparently disjointed data which offers a multitude of application possibilities. In the field of maintenance, data mining methods are successfully applied to make statements regarding failure patterns in the sense of predictive maintenance planning. So in this case the remaining service life is carried out on basis of the changes in the quality assurance measurements at the product instead of condition monitoring. It can be shown that data trends with clear trend breaks can be observed in measurement series at the measured points. The challenge is to extract unique characteristica which point to changing trends[16].

So this can be achieved by the following four step model, see Figure (2.6) introduced by[17]:

- **Step 1:** To understand the behavior of the system it is divided into its maintenance relevant components.
- **Step 2:** In parallel to step 1 historical machine, product and precess data are structurally processed.
- **Step 3:** Now the load profiles or machines of the same type, but with different product ranges an be compared and correlated.
- **Step 4:** In the last step, generally valid rules are developed from the previously determined data records.

Relevant data sources can be failure data, which is often a critical requirement because they directly correspond to maintenance[18][19], but also purely qualitative data[20][21], and cost-based criticality[22] can be used for assessing criticality, or cost deployment[23] to assign costs to machine downtimes.

2.3 | Performance Indicators in Maintenance

As seen in the chapter (2.1) business understanding and evaluation are phases needed needed in order to identify the underlying problems and later on to see if the measures taken have made the right impact. This means that KPIs are part of the business understanding and evaluation phase in CRIPSP-DM. Performance indicators are summarized information in numbers about technical and operational drivers. They must meet the need for information and are differently prepared for the various forms of information. Important aspects in the creation of key figures and key figure systems are user-friendliness and traceability of the key figures. It is also important to prepare those figures in a timely manner. Key figures depict business operations in a concise and objective form and are mainly used for decision making. Their importance is increasing. Key figures also offer the

[15] Bumblauskas et al., 2017.
[16] Matyas, 2018, p. 143.
[17] Matyas, 2018.
[18] Stadnicka et al., 2014.
[19] Bengtsson, 2011.
[20] Márquez et al., 2016.
[21] Bengtsson, 2011.
[22] Moore et al., 2006.
[23] Yamashina et al., 2002.

possibility of finding cost originators and can be used to Show cause-effect relationships[24]. In summary, one can say that key figures serve the following purposes:

- Create transparency
- Improve the objectivity of decision making
- Support the formulation of goals and forecasts
- Make performance and potential for improvement measurable
- Enable trend observations to be made
- Support the Continual Improvement Process (CIP) process
- Create opportunities for orientation and comparison
- Document conditions

Different organizational level need different key figures. This different preparation for the management level, the planning level and control level and for the executing level.

So there are various ways on how to use those key figures, see Figure (2.7). They can be used in controlling as a focus of own improvements like cost bearers, service quality and in customer surveys. In the external communication it is important to showcase the own performance capability in addition to special strengths, problematic weaknesses, outsourcing discussions. The own knowhow should also be offensively used for disruption analyses, information systems, plant optimization[25].

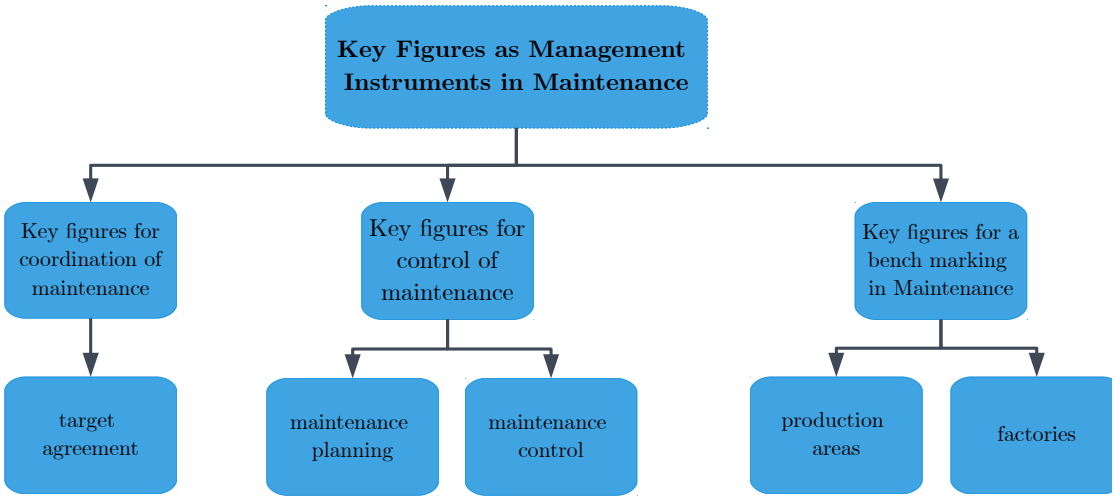


Figure 2.7: Maintenance strategies K. (2015)

Key figures can also be sorted according to their application[26]:

- Cost ratios
- Key figures for assessing planning quality

[24] Matyas, 2018, p. 97-98.
[25] K., 2015.
[26] Matyas, 2018, p. 98-102.

Leadership level	Summarization level	Key figure character	Example
Executive level			
technical management	whole company	global key figures with holistic character	maintenance efficiency
maintenance management	factory	goals, structures, trends	plant efficiency
operations management		global analysis in maintenance	budget plan
controlling			
Planning and control level			
workshop management	factory	key figures for plant or divisions	Personnel costs
operating engineers	operational divisions	structure, costs	material cost share
operations management			external service share
Operational level			
Foreman	plant	high details	plant availability
work scheduler	assembly	object and order level	share of overtime
craftsmen	order	assembly	execution time

Table 2.1: Key figure assignment to management level. Reprinted from Matyas (2018, p. 98).

- Workload indicators
- Indicators of labour productivity, and
- Structuring key figures of the organizational plan

So the following examples can be given:

2.3.0.1 Cost key figures:

$$\text{Maintenance intensity in \%} \quad \frac{\text{Annual maintenance costs}}{\text{Replacement value of the asset}} * 100 \tag{2.1}$$

$$\text{Maintenance cost rate MU/h} \quad \frac{\text{Maintenance costs}}{\text{Wage hours spent}} \tag{2.2}$$

Key figures for assessing planning quality:

$$\text{Failure time share in \%} \quad \frac{\text{Failure time per plant}}{\text{Operating time per plant}} * 100 \tag{2.3}$$

$$\text{Maintenance quote in h/piece} \quad \frac{\text{Maintenance hours spent}}{\text{Produced quantity}} \tag{2.4}$$

Workload indicators:

$$\text{Work overhang in days} \quad \frac{\text{Orders still to be executed in h}}{\text{Craft capacities in h/day}} \tag{2.5}$$

Indicators of labour productivity:

$$\text{Failure rate in \%} \quad \frac{\text{Downtimes of the maintenance employees}}{\text{Attendance time}} * 100 \tag{2.6}$$

Structuring key figures of the organizational plan:

$$\text{Maintenance cost ratio in MU/person} \quad \frac{\text{Total maintenance costs}}{\text{Maintenance personnel}} \tag{2.7}$$

Production efficiency indicators:[27]

$$\text{Overall Equipment Efficiency in \%} \quad \text{Availability} * \text{Performance} * \text{Quality} \tag{2.8}$$

$$\text{Availability} \quad \frac{\text{Run Time}}{\text{planned Production Time}} \tag{2.9}$$

2.4 | Big Data

As seen in the chapter (2.1) data understanding is needed in order to identify data quality problems and to discover insights. Therefore data has to be stored in a way that reflects the necessities of the use case, which is in this thesis the area of maintenance.

Data is the raw material for prescriptive maintenance, it is generated by machines, sensors, reports and invoices on a daily, hourly, minutely, secondly or even smaller basis. Processing this amount of information is a chance and a challenge at the same time. The growth and volume of data appears to be a remarkable problem for capturing, handling and processing those informations. So there are a lot of definitions out there which try to capture what data means in our time. A famous quote is from the British Mathematician Clive Humby who is credited to have coined the phrase.

"Data is the new oil. Its valuable, but if unrefined it cannot really be used. It has to be changed into gas, plastic, chemicals, etc to create a valuable entity that drives profitable activity; so must data be broken down, analyzed for it to have value. But there are other definitions out which try to capture what impact data has."

"Data are becoming the new raw material of business: an economic input almost on a par with capital and labour."[28]

[27] Farinha, 2018.
[28] Cukier, 2010.

"..data can create significant value for the world economy, enhancing the productivity and competitiveness of companies and the public sector and creating substantial economic surplus for consumers."[29]

"Data is the new gold"[30]

So data can be considered as the oil of the 21^{st} century? According to Marr it is not the new oil because it is not a single-use commodity, because data can be shared and reused for new purposes, and whereas oil is a finite resource, new data is becoming increasingly available and can be harvested nearly infinitely. Also the value of oil is based on its rarity, whereas data becomes more and more valuable the more it is reused. Data can and must be accumulated but it does not grow more valuable without the right data governance. Without data governance data is just cost and liability[31].

Gartner group coined 3Vs of Big Data[32], some time later IBM[33] added a 4th V to the Big Data landscape, as seen in Figure (2.8),[34]. Those Vs can be extended to 10, as introduced by Ansari et. al.[35]

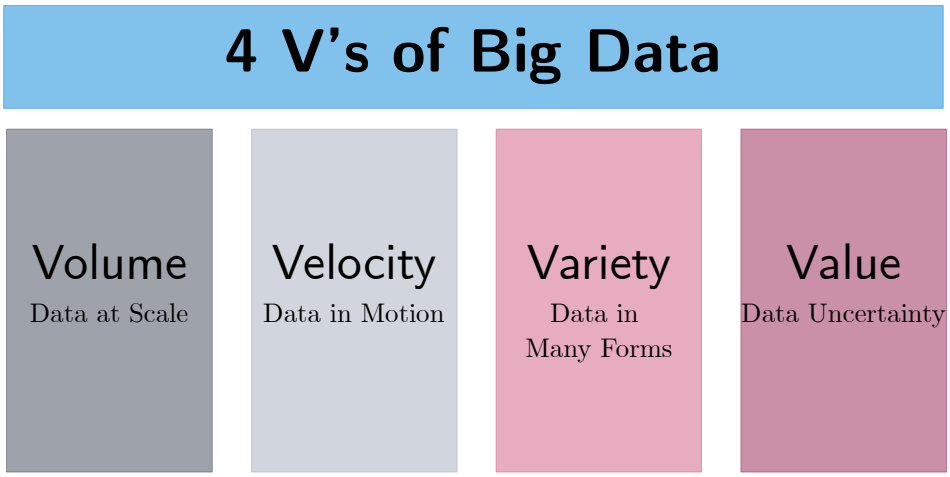


Figure 2.8: 4 V's of Big Data, based on Corrigan (2013)

- **Volume:** It presents the physical volume of data, and moreover its growing volume.
- **Velocity:** With the accelerated growth of data, comes change. Data can change dynamically, or be obsolete in some few seconds.
- **Variety:** Data can represent a variety of data types, formats and structures (structured, semi-structured, unstructured), encodings, syntax, semantics and so on.
- **Veracity:** The accuracy of data, how truthful is a data set?

[29] Manyika et al., 2011.
[30] Kroes, 2011.
[31] Marr, 2018.
[32] Stephen, 2011.
[33] IBM, 2019.
[34] Jagadish, 2015.
[35] Matyas, 2018.

Big Data can enable business intelligence, but there have been changes in recent times[36]. Analytics are a necessity for success and in the market and decision should be data driven and not led by intuition. The traditional approach is driven by information requirements like:

- Business users determine what questions to ask
- IT structures the data to answer that question (sales reports, profitability analysis)

Where the more recent vision is a data-driven discovery, predictive and prescriptive, optimization which:

- IT delivers an information platform for creative discovery
- Business explores what questions could be asked (e.g. brand sentiment, trends in consumer behavior etc.)
- Integration of semi-structured and unstructured data
- Use of big data technologies (NoSQL, Hadoop stack etc.)
- Do it all in (near) real-time (not just offline reporting and analytics)
- Integrate BI into data-driven business processes

2.4.1 | Data Warehouse

A data warehouse is a system for reporting and data analysis and is a core component of business intelligence according to Dedi et al. (2016). It can be defined as a collection integrated databases designed to support a decision support system (DSS)[37]. Another system called Data Vault is proposed by Lindstedt et al. (2009). In the Data Vault system the data is loaded from the source system in its original format. So the Data Vault is build around Hubs, Links and Satellites[38]. The alternative view of a data warehouse which will be presented here from Kimball et al. (2013) sees a data warehouse as a collection of data marts. Those can be used for querying and reporting and connected using conformed dimensions. So there is no need to replicate all data from the source system.

So what are the requirements for data warehouseing in business intelligence? One set of requirements is:

- "The DW/BI system must make information easily accessible
- The DW/BI system must present information consistently
- The DW/BI system must adapt to change
- The DW/BI system must present information in a timely way.
- The DW/BI system must be a secure bastion that protects the information assets
- The DW/BI system must serve as the authoritative and trustworthy foundation for improved decision making.
- The business community must accept the DW/BI system to deem it successful"[39]

[36] Kiesling, 2018.
[37] Inmon, 2005.
[38] Jovanovic et al., 2014.
[39] Kimball et al., 2013.

Looking at Figure (2.9) five main areas can be seen. The first which is the source database can be an operational application, or On Line Transaction Processing (OLTP), so a system which supports one or more types of business transactions .

- **Landing Area** is a data base which is able to store a single data extract of a subset of one of the Source databases. Its schema is identical with the subset of the source database.
- **Staging Area** is a database which is able to sore matching data extracts from various Landing Areas in an integrated format. The data warehouse must wait for the upload of the Staging Areas. It is identically to the schema of the data warehouse.
- **Data Warehouse** is a database which contains the history of all Staging Areas. Its schema is normally a Third Normal Form.
- **Data Market** is a database which is on disk or main memory, and contains the data describing the performance of one or more business transactions which are taken form the data warehouse. The schema of a Data Mart often has the form of one or more "stars".

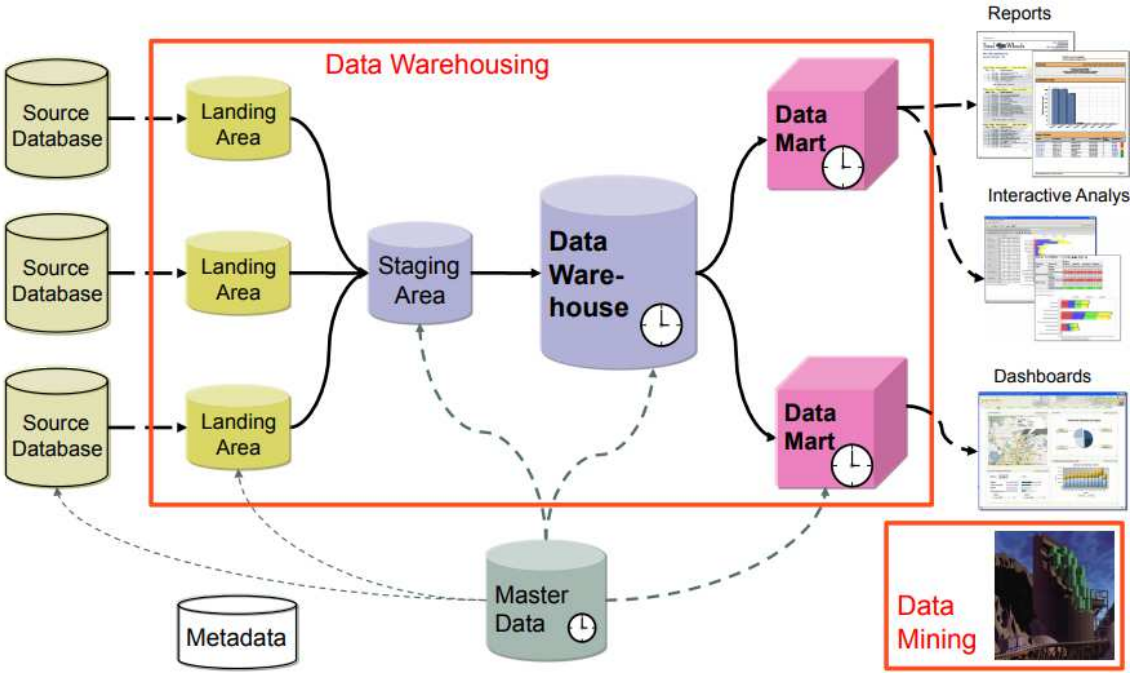


Figure 2.9: Data Warehouse Reference Architecture Marti (2012)

In the business intelligence applications of Big Data there are two main paradigms. Online analytical processing (OLAP) and online transaction processing (OLTP). Where OLAP is the answer to multi-dimensional queries which need to be computed swiftly, OLTP is the traditional transaction oriented way of data base transactions.

OLAP is all about getting the data out and analyzing the data describing business transactions. The goal is reducing the response time. Its results are large result sets with a lot of join operations, but no immediate updates and/or inserts. Periodic batch inserts are used here. Also a controlled redundancy, which is a necessity for performance reasons, because denormalized schemas, materialized views and indexes are used[40]. Typical applications are:

[40]Marti, 2012.

- Management Information Systems (MIS)
- Decision Support Systems (DSS)
- Statistical Databases
- Scientific databases, Bio-Informatic

OLTP focuses on getting the data in and capturing data which describes business transactions. Here the goal is many short and small transactions, like point queries, single row updates and/or inserts. Uncontrolled redundancies should be avoided and normalized database schemas used. So there is always consistent and up to date database[41]. Typical applications are:

- Flight reservation systems
- Procurement
- Order Management

2.4.2 | Data Lakes

A data lake is a collection of storage instances of various data assets additional to the originating data sources. These assets are stored in a near-exact, or even exact, copy of the source format."[42]

Data lakes are created to present an unrefined view of data. So those information hidden in the data must be extracted by an analyst in order to gain information out of the data. The data analyst, therefore, to compromise with the restrictions of classic data stores such as data marts or data warehouses. So according to Kiesling (2018) typical characteristics are:

- Single store of enterprise data with a flat architecture
- Schema and data requirements are not defined until the data is queried
- Extract, Load Transform (ELT) rather than Extract, Transform, Load (ETL)
- Promises cost-effective scalability and flexibility
- Allows new kinds of analyses and insights
- Promises low long-term cost of ownership

ETL is seen as an continuos, ongoing process which is well defined. So data is extracted from homogenous or heterogenous data sources. In the next step these data are cleansed, enriched, transformed and then stored in either a data warehouse or data lake[43]. ELT is a variant of ETL where the extracted data is first loaded into a target system. So transformations happen after the data is loaded into the target system. This target system must be powerful to handle those transformations[44].

How could someone define the relation between data warehouse and a data lake? A data lake is not a replacement for a traditional data warehouse, for example because it

[41] Marti, 2012.
[42] Gartner, 2019a.
[43] V. Ranjan, 2009.
[44] V. Ranjan, 2009.

uses schema on read as an alternative approach to deal with unknown questions. There are no established reference architectures, but proposed solutions as seen in Figure (2.10), for data lakes. Data lakes are still very tool focused but there is a trend in data lakes which sees them maturing because there are efforts to build architectures, structures, models and data governance[45][46]. In praxis data lakes have a polyglot persistence and are realized in ways such as:

- Analytics on multiple platforms
- On premise and/or in the cloud
- Combination of traditional data warehouse, Hadoop data lakes, MPPs, NewSQL, NoSQL, Search..
- Data in lakes accessible via SQL-on-Hadoop or SerDes on raw data

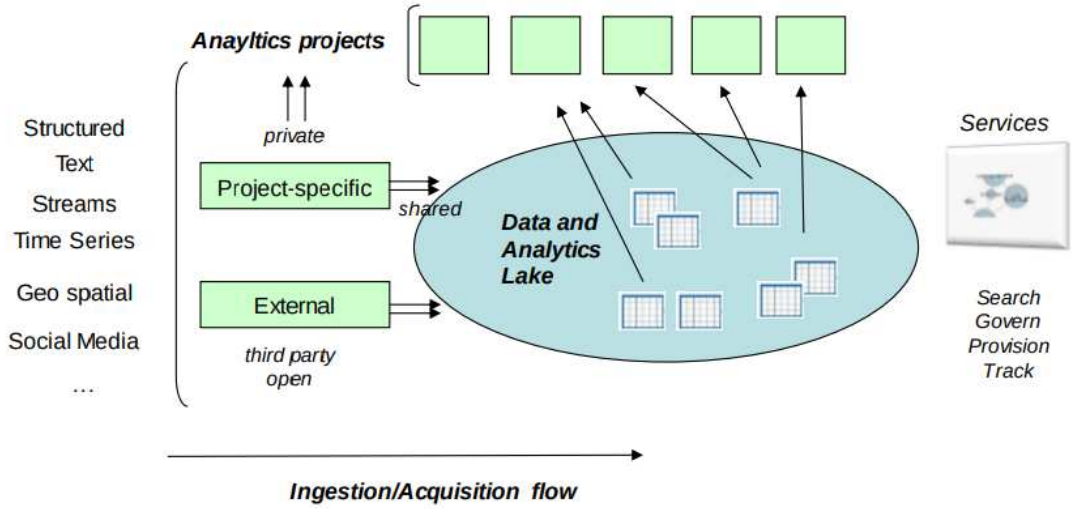


Figure 2.10: Data Lake Logical Architecture Terrizzano et al. (2015)

2.5 | Data Streams

As seen in the chapter (2.1) data understanding is needed in order to identify data quality problems and to discover insights. Therefore data streaming is an essential part of the data understanding phase. Distributed systems involve thousands of entities and are potentially distributed all over the world, where the location may change over the lifetime of the systems. So there is the need for more flexible systems, which reflect the decoupled nature of applications. Two popular open source systems are Apache Kafka and RabbitMW which commercially-supported pubsub systems. They have distinctive architectural differences. In Kaka producers publish bathes of messages to a disk based append log which is topic specific. Any number of consumers can subscribe to this topic and pull the stores messages through and index mechanism. In RabbitMQ producers publish batches of messages with a routing key to a network. In this network routing decisions happen dn the messages end um in queues where consumers can gat at messages through a push pr pull mechanism[47].

[45] Terrizzano et al., 2015.
[46] Kiesling, 2018.
[47] Dobbelaere et al., 2017.

So the basic scheme of the publish/subscribe paradigms is the ability for subscribers to express their intrust in an event or a pattern of events to be notified generated by a publisher. So producers publish information in a software bus and consumers subscribe to this bus. So the basic system, as seen in figure (2.11), relies on even notification[48].

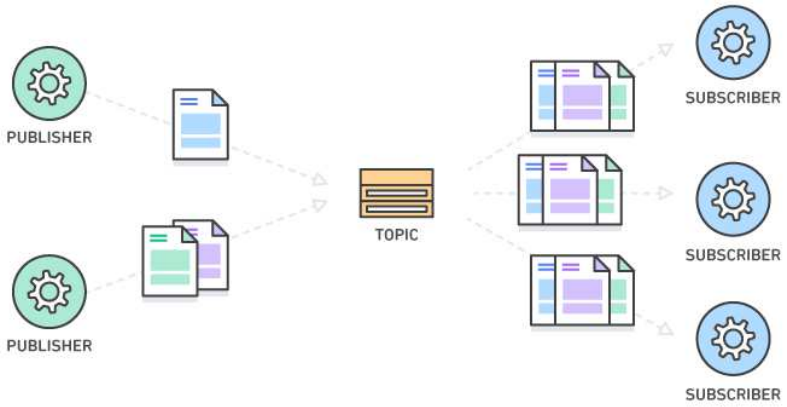


Figure 2.11: Publisher/subscriber architecture, based on AWS, 2019

Core Functionalities

The most fundamental function is de decoupling between publisher and subscriber. The publisher subscriber coordination scheme can be described in three dimensions:[49][50]

- Entity decoupling: publisher und consumers need not to be aware of each other.
- Time decoupling: publisher and consumer do not need to be active at the same time.
- Synchronization decoupling: the communication between publisher and consumer does not need to block the producer or consumer.

Also very important in such a system is the routing logic. So a system which decides if and which packet will end up at a consumer. So there are two main subscription schemes:[51]

- Topic based: The publisher tags its massages to a set of topics. Those topics can be used to filter which messages go to which consumer.
- Content based: In this case all fields of the message can be used for filtering. The constrains for filtering can be logically combined.

Quality-of-Service Guarantees (QoS)

A subscriber/publisher system is also described by a large set of required and desired guarantees which are referred as QoS[52]. Those are:

- Correctness can be described by three primitives: no-loss, no-duplication, no-disorder. Based on this two other criteria are relevant: Delivery guarantees and ordering guarantees[53].

[48] Eugster et al., 2003.
[49] Eugster et al., 2003.
[50] Dobbelaere et al., 2017.
[51] Dobbelaere et al., 2017.
[52] Eugster et al., 2003.
[53] Sheykh Esmaili et al., 2011.

- Reliability is the ability to perform even if one or several of the hard- or software components fail.
- Availability is the capacity of system to maximize its uptime
- Transactions are used to group messages into atomic units. So either all of a message is sent or nothing.
- Scalability is the concept of being able to continuously evolve in order to support a growing amount of tasks.
- Efficiency is has two measures latency, the response time, and the throughout or bandwidth.

Requirements in IoT

Before going into detail about on different messaging systems it is important to set the requirements for messaging systems in IoT. For industrial cyber physical systems Cheng et al. (2018) defined two main requirements:

- High throughput, where huge amounts of data from IoT sources can be transmitted to effectively and in real time
- Fault tolerant data ingestion

For big data pipelines in maintenance O'Donovan et al. (2015) has identified the problem of the data transmission across the factory. So the requirements for a system which can fulfill all needs in a big data pipeline would be:

- Legacy integration
- Cross-network communication
- Fault tolerance
- Extensibility
- Scalability
- Openness and accessibility

2.6 | Case-based Reasoning

Case-based reasoning can be found in the modeling stage if seen in the sense of the CRISP-DM model, as seen in chapter (2.1).

2.6.1 | Case-based Reasoning - Fundamentals

CBR is a paradigm suitable to solve problems by utilizing specific knowledge of previously experienced problems (cases). This makes CBR different from other major AI approaches.[54]. This "reasoning and remembering approach"[55] was strongly influenced by studies of the human brain[56] and was adopted for machines Case-based reasoning is both ... the ways people use cases to solve problems and the ways we can make machines use

[54] Aamodt et al., 1994.
[55] D. B. Leake, 1996.
[56] De Mantaras et al., 2005.

them[57]. CBR can be seen as a circular process[58], which shares many similarities with the CRISP-DM cycle, as seen in chapter (2.1).

The central idea is to store past experience in cases which are used for solving new problems. This is done by recalling similar experiences, the reuse of those experiences in a new context and storing the new experience in memory. CBR is a subset of Artificial Intelligence (AI) and belongs to the methods of supervised learning in Machine Learning (ML). Learning in CBR is based on analogies and therefore not by deduction or induction. As usually in model-based approaches data is used to create a model and most calculations are done while building the model[59].

Advantages:

- Avoidance of high knowledge acquisition effort
- Simpler maintenance of the knowledge in the system
- Facilitation of intelligent retrieval
- High quality of solutions for poorly understood domains
- High user acceptance

So typical application fields are:

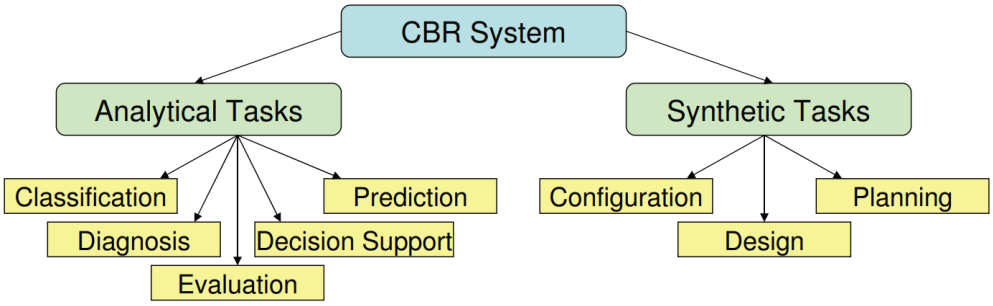


Figure 2.12: Typical application fields, reprinted from Gabel (2010)

Five main types of CBR have been identified by[60]:

- **Exemplar-based reasoning:** The "classical" view.[61] A concept is defined extensionally as a set of exemplars. Solving a problem is seen as a classification problem. For example finding the right class for the unclassified example. A set of classes constitutes the set of possible solutions
- **Instance-based reasoning:** This syntactic approach compensates a lack of general background knowledge with a large number of instances. The instances are normally simple feature vectors. This is a non-generalized approach to the concept learning problem addressed by classic machine learning (ML) methods.

[57] Kolodner, 2014, p. 27.
[58] Aamodt et al., 1994.
[59] Gabel, 2010.
[60] Aamodt et al., 1994.
[61] Smith et al., 1981.

- **Memory-based reasoning:** This approach sees collection of cases as a large memory and reasoning is seen as accessing and searching this memory. It uses parallel processing techniques, and the access and storage rely on purely syntactic criteria, or the try to utilize general domain knowledge.

- **Case-based reasoning:** This typical CBR method differs from other in the richness of information contained in a case and a certain complexity in respect to its internal organization. A typical case-based method is also able to modify, a retrieved solution wen applied in a different problem, it also utilizes general background knowledge. This approach leans heavily on cognitive psychology theories.

- **Analogy-based reasoning:** This approach is very similar to the CBR approach, but it is often used to characterize methods to solve past problems form different domains, whereas normally CBR approaches focus on cases from one domain.

2.6.2 | Case-based Reasoning - Cycle

There are two main parts, or views on how to describe CBR methods, and they are both complementary:

- A model of the CBR cycle

- A task-method for CBR

The CBR cycle (2.13) can be described by its four processes[62]:

- **Retrieve** the most similar case or cases

- **Reuse** all necessary information and knowledge on how to solve this problem

- **Revise** the proposed solution

- **Retain** the useful parts of this experience for future problem solving

Which fits into the task orientated view of knowledge level modeling from Van de Velde, 1993. In this concept a system is viewed as an agent with goals, and those goals set tasks to achieve them. A system can be described by tasks, methods and domain knowledge. Tasks apply one or more methods, and those methods need knowledge to achieve its tasks. So a task-method structure (2.14) was defined by Aamodt et al. (1994). The tasks are linked by lines which are decompositions and the relationship is downwards, the tippled lines indicate alternative methods applicable for solving a task.

From top down it is problem solving and learning from experience with its method CBR. CBR has four tasks itself as seen in (2.13) retrieve, reuse, revise, and retain. The retrieve task for example itself is partitioned in identify, search, initial match and select. All four major tasks have to be completed to complete the top level task. The task-method decomposition model(2.14) has no control structure.

- **Retrieve**

  - **identify features** To identify a problem in knowledge-intensive methods an attempt is made to understand the problem. To understand a problem noisy

[62] Aamodt et al., 1994.

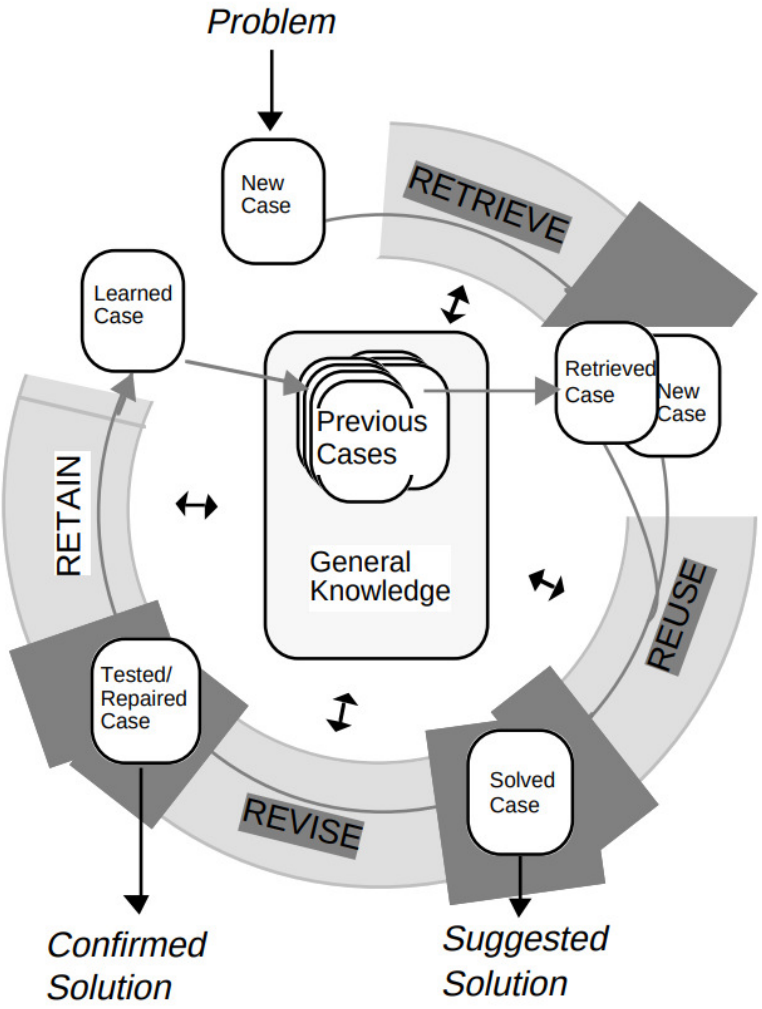


Figure 2.13: The CBR cycle Aamodt et al., 1994

problem descriptors have to be filtered out. Others than those given as input maybe use a general knowledge model, or are retrieve from the case base and use features of that case which are expected features. So the checking of expectations must be done within the knowledge model.

- **collect description**
- **interpret problem**
- **infer descriptors**

- **search**
- **initially match** The task of finding a good match must be split into two subtasks, the first one is the initial matching process where a set of plausible matches is retrieved, and the best one is selected. The latter task is the select task. For retrieving there are three ways: "By following direct index pointer from problem features, bu searing an index structure, or bu searching in sa model of general domain knowledge"[63].
- **select** The select task is done after a set of similar cases was chosen and a best best must now be selected. The best match is usually determined after evaluating the set of initial matches more closely. If a a match is not strong

[63] Aamodt et al., 1994.

enough a different link is followed to find a more closely related case. During the selection process consequences and expectations are made and attempts to justify them. This can be done by using a system of general domain knowledge or by the user through confirmation or adding additional information.

- **Reuse**
  - **copy** In simple classification tasks differences are abstracted and the solution is transfered to the new case. This is simple type of reuse, but other systems consider differences and so an adaption process takes place.
  - **adapt** An old case can be reused by reusing the old case solution or by transformational reuse with reusing the past method that construed the solution which is called derivational reuse. In the first the past solution is not directly the new one but there is some knowledge about it in the form of transformational operators, which can be applied. In the derivational reuse it is looked at how the case was solved. So the retrieved case holds informations about the used method for solving and its justification like the used operators, subgoals considered, generated alternatives, failed searches and so on. So the those things are replayed with the new case.
- **Revise**
  - **evaluate solution** This task normally takes place outside the CBR system, in this step the applying if the solution is evaluated in a real environment.
  - **repair fault** The repair fault task involves to steps, first the detection of errors and second the retrieving or generation of an explanation for them. The solution repair task is the second task of the revision phase.
- **Retain**
  - **integrate** This final step is the updating of the knowledge base with the new case knowledge. If there is no new case and an index set has been constructed it is the main step in the retain phase. By modifying the indexes the CBR system learns to become a better similarity assessor and it is an important step in CBR learning. The strength or importance of an index is very important and are adjusted due to the success or failure of the case to solve an input problem. If some features are judged relevant they are strengthened and others are weakened if they did lead to unsuccessful case retrievals.
    - **rerun problem**
    - **update general knowledge**
    - **adjust indexes**
  - **index** The indexing problem is one of the most focused problems in CBR. It is about what can of index is to be used for future retrieval and how to structure the search space of indexes. Direct indexes are actually knowledge acquisition problems and should be analyzed as part of the domain knowledge analysis in the modeling step.
    - **determine indexes**
    - **generalize determine indexes**
  - **extract** In CBR the is in most times solved by the use of a previous case If it was solved by user input or other methods a new case must be constructed. Anyhow a decision of what to use as a learning source. Failures, so information from the Revise task can also be extracted and retained, either as failure case

or within a total problem. So when a failure is encountered the system can be reminded to a similar previous failure.

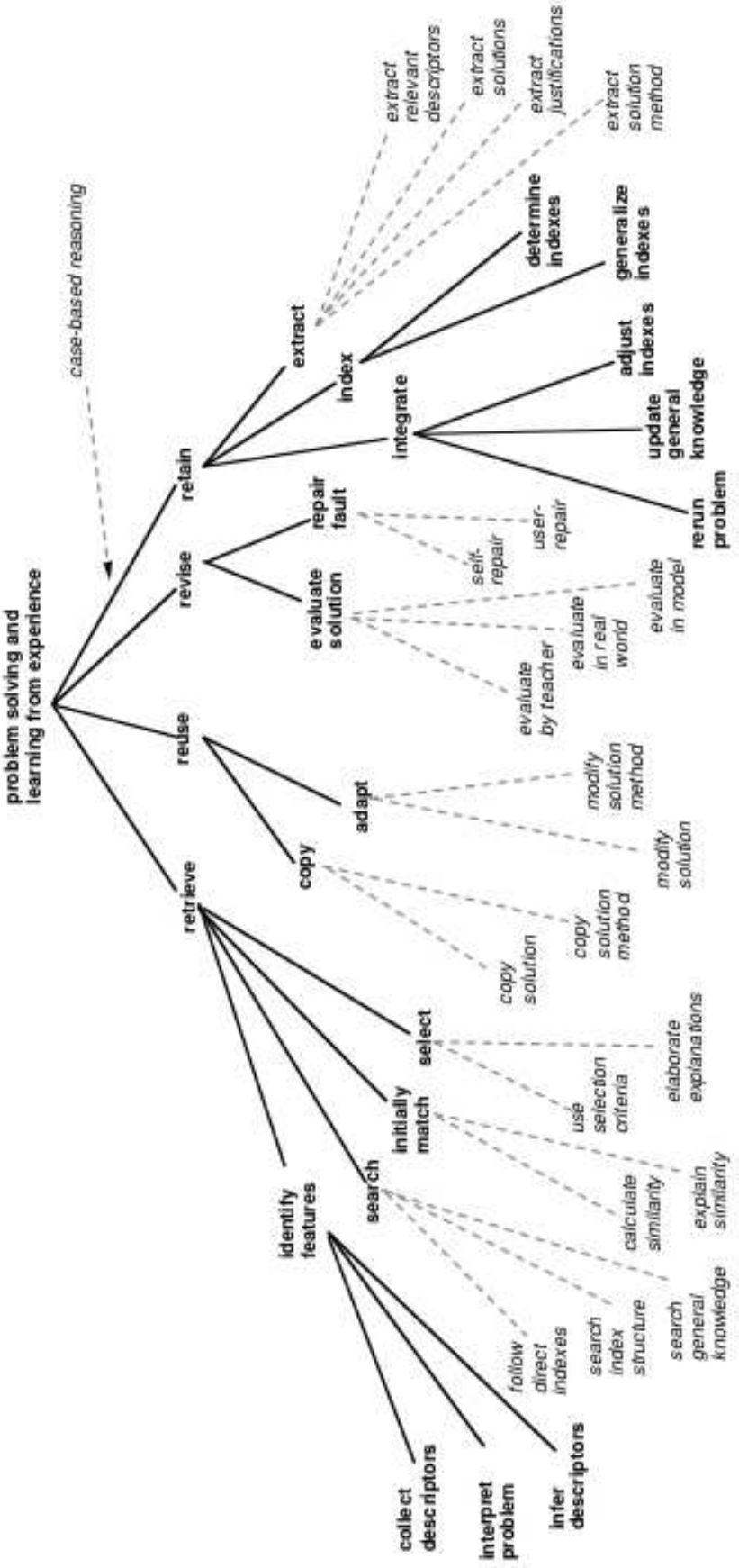


Figure 2.14: A task-method decomposition of CBR by Aamodt et al. (1994)

The Case and Knowledge

A case can be an abstraction of events that are limited in space and time in the cognitive science view, contrary from a technical point of view a case is a description of a problem, which in combination of certain experience can be used to solve the case[64]. A case consists of mandatory and optional parts.

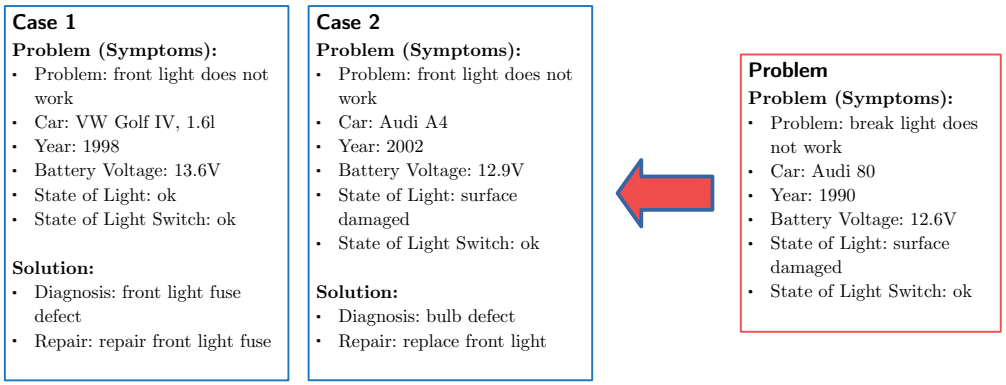


Figure 2.15: Solving a New Diagnostic Problem, reprinted from Gabel (2010)

So CBR is considered as a problem solving method and when faced with a problem a solution must be provided. With a new problem the case memory must be searched, however often it does not start with a complete problem description. This can mean that the available information is not sufficient to successfully solve the case[65].

- **Diagnosis:** First symptoms given, but not enough to identify the cause
- **Design:** When the first functional descriptions are given, but not enough to choose the right parts the layout problem
- **Consulting:** The customer has some ideas, the are not detailed enough to propose a clear specification.

Using the CBR circle (2.13) the first part would be Retrieving the case, after this comes the crucial step of Reuse. Here the solution must be adapted. In this case a new diagnoses could be break light fuse defect and it could be repaired by repairing the break light fuse. After this comes the Retain part, where if the solution was correct it should be stored as a case in the case base.

Knowledge

"In order to solve problems one needs knowledge"[66]

The knowledge of a CBR system comes form its vocabulary which represents the knowledge. Retrieval is done by similarity assessment. So the similarity between two cases can be measured and is a core step of CBR. The information about the new problem which is query must cover all necessary information to be applicable:

- target of problem

[64] Gabel, 2010.
[65] Lenz et al., 2003.
[66] Michael M. Richter, 2010.

- constraints
- characteristics

The solution contains all information to describe the solution for the problem correctly. A feedback if the solution was correct must also be added.

- solution itself
- justifications
- possible alternative solutions
- steps that were tried, but failed

Every case is represented by pairs of attributes and their belonging values, so a set of attributes $A_1, ..., A_n$ is fixed and to each attribute A_i there is an associated domain D_i so each attributes value is defined by $a_i \in D_i$. Those attributes can be numerical, symbolic, or textual (strings). The choice of attributes and their corresponding domains needs general knowledge, namely vocabulary knowledge. Domains are mainly chosen by the requirements of similarity computation and solution adaption. So the choice of attributes is an important step, they must allow the decision if a case is similar or not, they should not be redundant and should represent independent properties of the case.

Advantages	**Disadvantages**
straightforward representation	no structural or relational information
easy to understand and implement	no ordering information is representable
cases are easy to store (databases)	
efficient retrieval	

Table 2.2: Advantages and Disadvantages of case representation formalism

There can also be an object-oriented case representation

- Graph- and Tree-based representation
- First-Order-Based Case Representation
- Hierarchical case representation
- generalized cases

2.6.3 | Similarity

Similarity is the central notion in CBR and is always considered between problems, so the selection of cases in the retrieve phase is based on similarity of a given query. Similarity is can be easily adapted to the current problem. There are three observations made by Gabel (2010):

- There is no universal similarity; similarity always relates to a certain purpose.
- Similarity is not necessarily transitive
- Similarity does not have to be symmetric

Similarity can be modeled by a similarity measure and a distance measure.

Similarity measure: A Similarity Measure on a set M is a real-valued function $sim : M^2 \to [0, 1]$. So similarity is:

- reflexive iff. $\forall x \in M : sim(x, x) = 1$
- symmetric iff. $\forall x, y \in M : sim(x, y) = sim(y, x)$

Each similarity measure induces a similarity relation R_{sim}.

Distance measure: A Similarity Measure on a set M is a real-valued function $sim : M^2 \to \Re_0^+$. So distance is:

- reflexive iff. $\forall x \in M : d(x, x) = 0$
- symmetric iff. $\forall x, y \in M : d(x, y) = d(y, x)$

Each distance measure induces a similarity relation R_d.
So there is a relation between similarity and distance. Definition: A similarity measure sim and a distance measure d are called Compatible if and only if: $\forall x, y, u, v \in M : R_{sim}(x, y, u, v) \longleftrightarrow R_d(x, y, u, v)$. Similarity can be measured in attributes-value based case representation the following way.

Hamming Distance:
The Hamming Distance is for binary features, where $H(x, y)$ is the number of attributes with differing values, and H is a distance measure.

$$H(x, y) = n - \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} (1 - x_i)(1 - y_i) \tag{2.10}$$

Simple Matching Coefficient (SMC):
The Simple Matching Coefficient transforms the Hamming Distance into a compatible similarity measure and is not restricted to binary features. So it can be used for nominal discrete variables without natural ordering like color and for ordinal discrete variables with a natural ordering like school grades.

$$a = \sum x_i y_i \tag{2.11}$$
$$b = \sum x_i (1 - y_i) \tag{2.12}$$
$$c = \sum (1 - x_i) y_i \tag{2.13}$$
$$d = \sum (1 - x_i)(1 - y_i) \tag{2.14}$$
$$SMC(x, y) = 1 - \frac{n - (a + d)}{n} = \frac{(a + d)}{n} = 1 - \frac{(b + c)}{n} \tag{2.15}$$

Measures for Real-Valued Attribute:

- City-blocked metric d_1

$$d_1 = \sum_{i=1}^{n} |x_i - y_i| \tag{2.16}$$

- Euclidean distance d_2

$$d_2 = ||x - y|| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.17}$$

It is the sum of the square of each distance and because squares of the distance are summed, the largest value may dominate. The Euclidean distance measure is appropriate for example wen data is not standardized, but the distance measure can be greatly effected by the scale of the data[67].

Weighted Euclidean distance d_{2w} The weighted Euclidean distance takes care of the fact that not all attributes are of the same importance with the help of weights. The weights w_i must be chosen carefully, because the higher the weight the more the influence on the measure and on the chance to be selected as the nearest neighbor.

$$d_{2w} = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2} \tag{2.18}$$

Measures for Sparsely Filled Cases:
In cases like online shops the value 0 is predominating, which must be taken into consideration by the similaritydistance measure.
Maybe it is desired to consider two customers similar when they bought the same product often. In this case the Cosine Similarity Measure, which is based on the inner product, ca be used.

$$SMC_{00}(x, y) = \frac{(\sum_{i=1}^{n} I(x_i = y_i)) - f}{n - f} \tag{2.19}$$

$$cos(x, y) = \frac{x^T y}{||x|| * ||y||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.20}$$

Local-Global Principle

Here the description of cases and queries are only about real valued vectors $u \in U = \Re^n$. The elements of a vector always correspond to some attribute A_i. The Local-Global Principle which is for similarities and distances can be formulated as following way[68]:

There are similarity measures sim_i on the domains of the attributes A_i (on the reals in our case) and some composition function $COMP : /R_n \Rightarrow R$ such that Maybe it is desired to consider two customers similar when they bought the same product differently often. In this case the Cosine Similarity Measure, which is based on the inner product, ca be used.

$$SIM([q1, ..., qn], [u1, ..., un]) = COMP(sim1(q1, u1), ..., sim(qn, un)) \tag{2.21}$$

The measures sim_i are called local measures and SIM is the global measure. A similarity measure SIM is called composite if it can be combined from some local similarity measures as in 2.21.

Local Similarity

The local similarity measures sim_i which is defined as $sim_i : \Re \times \Re \Rightarrow \Re$ determines the similarity $sim(qi, ci)$ for the values of a query$q = (q1, ..., qn)$ and a case c= $(c1, ..., cn)$ at the Attribute A_i.

Local similarity expresses the acceptance for different values of the Attribute. So it is possible to match vague notions in queries like "end of the week" with concrete values in

[67] GUPTA, 2014.
[68] Burkhard, 2004.

the case like "Friday". It is also possible to compare concrete values in a query with vague notions in the case, or vice versa. It is even possible to use vague queries with vague cases, which follows operations from fuzzy logic[69]. Interesting to point out is also that functions of linguistic terms usually have non-zero values only in a very limited region of their domain, whereas measures from distances have non-zero values even for great distances. For unacceptability there is the way of using negative similarity values. When using weighted sums a negative local value decreases the global acceptance of a case c for the query q.

Similarity Tables
Similarity tables are for attributes with symbolic type like the type of RAM

	SD	DDR	RD
SD	1.0	0.9	0.75
DDR	0.5	1.0	0.75
RD	0.25	0.5	1.0

Table 2.3: Similarity table Gabel (2010)

Difference-Based Similarity Functions Difference-based similarity functions are for attributes with a numeric type, and the similarity is measured as the numerical difference between the case and the query.

Local Similarities for Taxonomies It must be assumed that the considered objects can be ordered in Taxonomy or set of notions. So the similarity for leaf nodes is defined in the following way:

- each inner node must be assigned to a similarity value e
- the successor nodes become larger similarity values
- the similarity between two leaf nodes is calculated by the similarity value by the deepest common predecessor

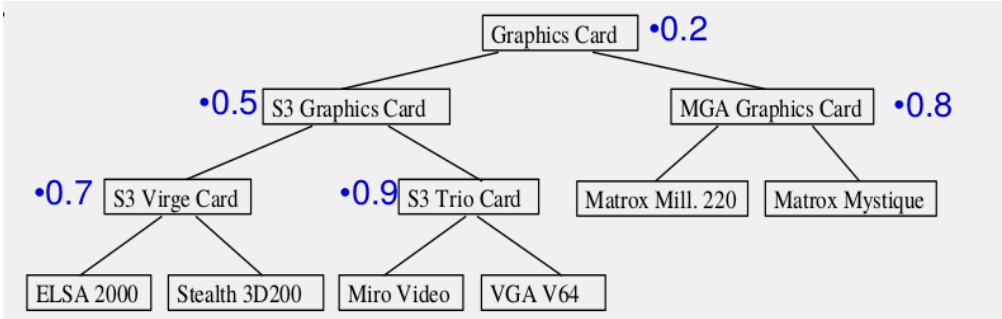


Figure 2.16: Local Similarities for Taxonomies, reprinted from Michael M. Richter (2010)

Global Similarity "A higher global similarity must be supported by at least one higher local similarity"[70]

The Global Monotonicity Axiom states:

$$SIM(u,v) > SIM(u,w) \rightarrow \exists i \in \{1,...,n\} : sim_i(u_i,v_i) > sim_i(u_i,w_i) \tag{2.22}$$

[69] Burkhard and Michael M Richter, 2001.
[70] Burkhard, 2004.

There is a situation where this axiom does not hold, for example in the well known XOR-problem form neural nets. A new attribute the $XOR(u, v)$ would be needed in order to represent the XOR-problem by monotonous global similarities, so an extension of the language it self. This leads to the point that it is important to find attributes such that the monotonicity axiom holds. They should also be easy to compute, which limits them further.

2.7 | Random Forest

A decision tree, or the advanced random forest are ML techniques which can be applied in the modeling phase of the CRISP-DM model, as seen in chapter (2.1).

2.7.1 | Decision tree

"Decision trees are a simple model type: they make a prediction that is piecewise constant."[71]

Tree based methods involve stratifying and segmenting the predictor space into simple regions. Usually the mean or the mode of the data set is used for prediction. One of the great benefits is that tree-based methods are simple and can be easily interpreted. The three most common ways of improving decision trees are bagging, boosting and random forests, which will be explained in this section.

Decision trees can be applied to classification and regression problems. In regression problems, a given number of factors is used for determining a certain outcome. So the first split will be done at the most dominant factor and so two regions are obtained. The predictor space is divided into high-dimensional rectangles or boxes, and the goal is to find boxes that minimize the RSS given by

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j}) \tag{2.23}$$

and a top-down greedy approach is applied. Top-down because it begins at the top of the tree and successively splits the predictor space and greedy because in each step the best split is made, so there is no looking ahead. In recursive binary splitting first predictor is selected and a cut point so that the splitting leads to the greatest possible reduction in RSS. Another way is tree pruning, where a very large tree is grown which is then prune back to a subtree, which is done by K-fold cross-validation.

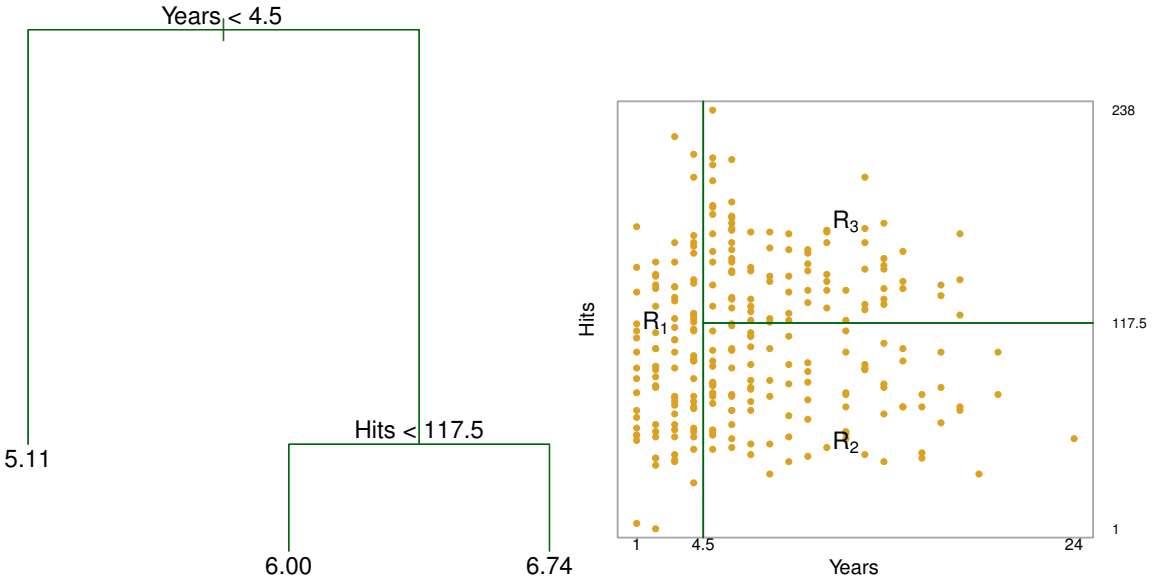
With classification trees the concept is similar, except that is used to predict a qualitative response, rather than a quantitative. So the process is very similar where regressive binary splitting is used to grow a classification three. A alternative to the RSS is the classification error rate

$$E = 1 - max_k(\hat{p_{mk}}) \tag{2.24}$$

or the Gini index which is often referred as a measure of node purity.

$$G = \sum_{k=1}^{K} \hat{p_{mk}}(1 - \hat{p_{mk}}) \tag{2.25}$$

[71] Zumel et al., 2014.



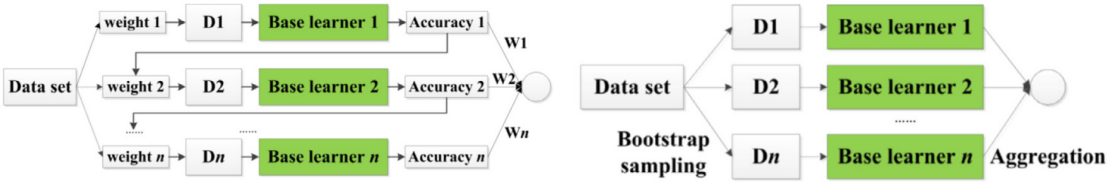
(a) For the Hitters data, a regression tree for prediction the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year, reprinted from James et al. (2014, p. 304)

(b) The three-region partition for the Hitters data set from the regression tree, reprinted from James et al. (2014, p. 305)

Figure 2.17: Tree-based methods

2.7.1.1 Bagging, Boosting and Random Forrest

"The idea of ensemble learning is to build a prediction model by combining the strengths of a collection of simpler base models".[72] Very popular ensemble learning methods are bagging and boosting.



(a) Bagging strategy, reprinted from S. Zhong et al. (2015)

(b) Boosting strategy, reprinted from S. Zhong et al. (2015)

Figure 2.18: Tree-based methods

Bagging

"Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a statistical learning method."[73] So bagging, as seen in figure (2.18b) is used to reduce the variance and hence increase the prediction accuracy, so may training sets are taken from the population and separate prediction models using each training set and the the results are averaged. Normally access to may training set is not possible or limited to they are bootstraped, by taking repeated samples form a single training set.

[72] Hastie et al., 2009, p. 624.
[73] James et al., 2014, p. 316.

Boosting

"Like bagging, boosting is a general approach that can be applied to many statistical learning methods for regression or classification."[74][Boosting, as seen in figure (2.18a) works simmilar to bagging, except that the trees are grown sequentially and each tree uses information from the previous tree. This has the advantage that not a single large decision tree is fitted to the data, which amounts to fitting the data hard[75] and therefore over fitting, instead boosting learns slow.

Random Forests

Random Forests are an improvement of bagged trees because they provide a tweak which are decorrelated tree. A number of decision trees are build on bootstrapped training samples, but each time they split a random sample of predictors m is chosen as split candidates from the full possible set of predictors p, typically $m \approx \sqrt{p}$[76]. So this avoids the problem in a collection of bagged trees that most trees will choose the dominant predictor.

2.8 | Deep Learning

Deep Learning is a technique which can be applied in the modeling phase, which is part of the CRISP-DM model, as seen in chapter (2.1).

Deep Feedforward Networks, also called multilayer perceptions (MLPs) are the quintessential deep learning models, they are mostly nonlinear statistical models. The try to approximate some function f^*, so a classifier $y = f^*(x)$ maps some input x to a category y. In a feedforward network it defines the mapping $y = f^*(x;\theta)$ and learns the value of the parameter θ, that results from the best function approximation. Those are feedforward networks because the information flow starts at the input x and goes through the intermediate computations which are used to define f and finally through the output f. They are called networks because the are normally composed of of many different function.

These chain structures can be in the form of $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$, which can be visualized as seen in fig (2.19).[77]. The learning algorithm it self decides hot use use those layers and the desired output, the layers in between, which do not show the desired output are called hidden layers.

For K-classification there are K units at the top, with the kth unit modelling the probability of the class k. So there are K target measures Y_k with $k-1, ..., K$, which are codes as 0 or 1. The derived features Z_m are created from those linear combinations of the inputs and so the target Y_k is just a function of linear combinations of Z_m.

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T), m = 1, \ldots, M \tag{2.26}$$

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, \ldots, K \tag{2.27}$$

$$f_k(X) = g_k(T), K = 1, \ldots, K \tag{2.28}$$

The activation function $\sigma(v)$ is usually a sigmoid function $\sigma(v) = \frac{1}{(1+e^{-v})}$ see fig.(2.20)

[74] James et al., 2014, p. 321.
[75] James et al., 2014, p. 321.
[76] James et al., 2014, p. 320.
[77] Goodfellow et al., 2016.

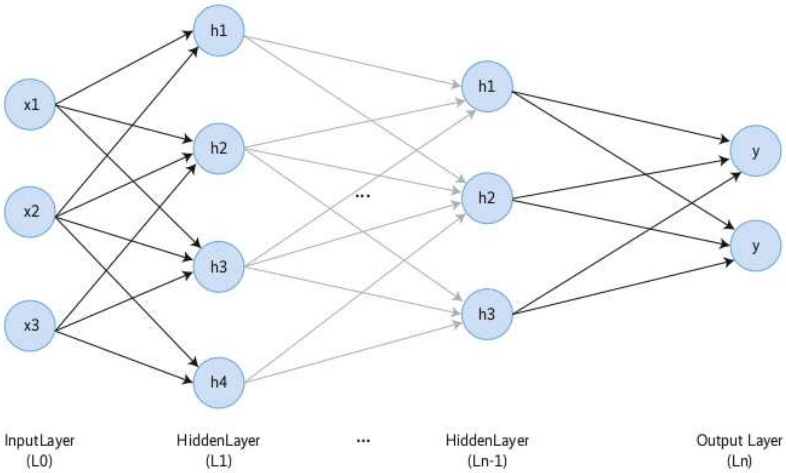


Figure 2.19: A feed forward neuronal network, reprinted from Tilly (2018)

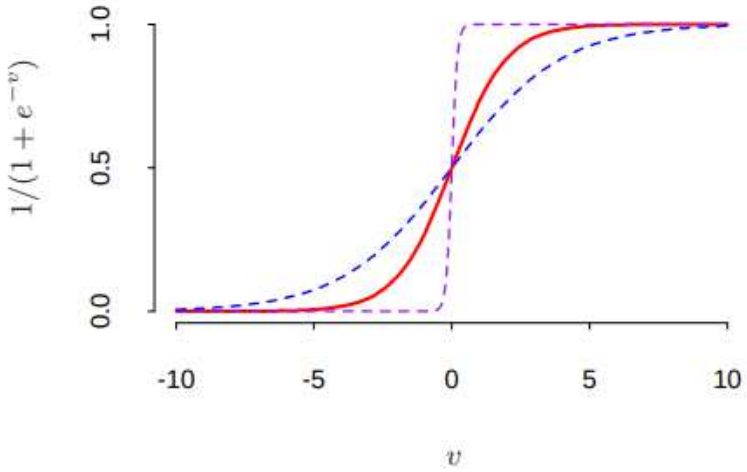


Figure 2.20: Plot of the sigmoid function (red curve), which is commonly used in hidden layers of a neural network. The other curves (blue, purple)are scaled, with a parameter which controls the activation rate, reprinted from Hastie et al. (2009)

Back-Propagation, often called backprop, is an algorithm where the information from the cost is allowed to then flow back through the network in order to compute the gradient[78]. Computing the gradient numerically can be computationally expensive, so the backprop algorithm uses an inexpensive method. The backprop refers only to the way of computing the gradient, while another algorithm such as stochastic gradient descent is used to perform learning using the gradient.

2.9 | Bayesian Networks

The HMC model, which is based on Bayes theorem can be used in the Modeling phase of the CRISP-DM model, as seen in chapter (2.1).

Bayes theorem eq. (2.29) allows to convert a prior probability, with its distribution

[78] Goodfellow et al., 2016.

$p(w)$ into a posterior probability w by incorporating evidence from the observed data $D = t_1, ..., T_N$.

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})}{p(\mathcal{D})} \tag{2.29}$$

The quantity $p(\mathcal{D}|\boldsymbol{w})$ is evaluated for the observed data $\mathcal{D}$ and can be seen as a function of $\boldsymbol{w}$, which is called the likelihood function. "It expresses how probable the observed data set is for different settings of the parameter vector $\boldsymbol{w}$."[79]

The denominator of the Bayes theorem eq. (2.29) can be expressed in terms of the prior distribution and the likelihood function:

$$p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w})d\boldsymbol{w} \tag{2.30}$$

Bayesian inference in machine learning[80] can help to give answers to typical problems like "I have **A**, what is **B**?". Bayes' rule provides a logic of uncertainty, so it is possible to reason with a given A the likelihood of B, with the conditional probability $P(B\|A)$. So this leads to a parametrized model which can be described as $P(B\|A) = f(A; \boldsymbol{w})$, where $\boldsymbol{w}$ denotes a vector of all adjustable parameters. In Bayesian inference parameters like w are treated as random variables, like A and B. So before obtaining the posterior distribution over $\boldsymbol{w}$ a prior distribution $p(\boldsymbol{w})$ must be specified.

2.9.0.1 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) allows sampling from a large class of distributions and is great with scaling of the dimensionality of the sample space[81].

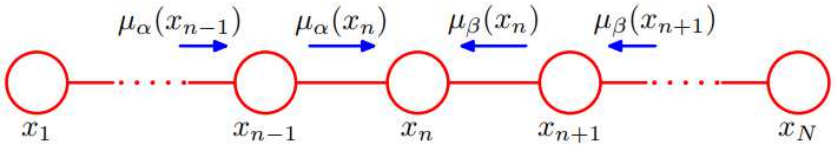


Figure 2.21: The marginal distribution $p(x_n)$ for a node x_n along the chain, reprinted from Bishop (2006, p. 397)

A first-order Markov chain can be defined as a series of random variables $\boldsymbol{z}^{(1)}, ..., \boldsymbol{z}^{(M)}$ which hold the conditional independence property for $m \in 1, ..., M-1$

$$p(\boldsymbol{z}^{(m+1)}|\boldsymbol{z}^{(1)}, ..., \boldsymbol{z}^{(m)}) = p(p(\boldsymbol{z}^{(m+1)}\|p(\boldsymbol{z}^{(m)})$$

which can be visualized as seen in fig. (2.21). To use Markov chain for sampling it must be set up such as that the desired distribution is invariant. A homogenous Markov chain with transition probabilities $T(\boldsymbol{z}', \boldsymbol{z})$, the distribution $p^*(\boldsymbol{z})$ is invariant if

$$p^*(\boldsymbol{z}) = \sum_{\boldsymbol{z}'} T(\boldsymbol{z}', \boldsymbol{z})p^*(\boldsymbol{z}) \tag{2.31}$$

[79] Bishop, 2006.
[80] Tipping, 2004.
[81] Bishop, 2006.

A sufficient, but not necessary condition is that the transition probabilities are chosen to satisfy the property of detailed balance to

$$p^*(\boldsymbol{z})T(\boldsymbol{z}, \boldsymbol{z}') = p^*(\boldsymbol{z}')T(\boldsymbol{z}', \boldsymbol{z}) \tag{2.32}$$

2.9.0.2 Gibbs Sampling

Gibbs Sampling is widely applicable and is based on Markov chain Monte Carlo algorithm. When considering a distribution $\boldsymbol{z}^{(1)}, ..., \boldsymbol{z}^{(M)}$ from which a sample is chosen for the initial state of the Markov chain, the Gibbs sampling procedure involves replacing the value of one of the variables with a value from the distribution of that variable, which is based on the variable conditioned of the remaining variables. So z_i is replaced bu the value drawn from the distribution $p(z_i|\boldsymbol{z}\beta)$. This procedure is repeated by cycling through the variables or by updating the variable to be chosen at each step[82].

Collapsed Gibbs Sampling
Collapsed Gibbs Sampling was introduced by[83] and the basic idea is to iterate one or more variables while sampling for some other variable. Where a Gibbs sampler would sample form $p(A|B, C)$, to $p(B|A, C)$ and then to $p(C|A, B)$ a collapsed Gibbs sampler can replace the sampling step for A with a sample for the marginal distribution $p(A|C)$, with B integrated out.

2.10 | Text Mining

Text mining instead to the other techniques presented in this thesis works not with structured data. It useses unstructured data, but the concept is still a part of the modeling phase of the CRISP-DM model, as seen in chapter (2.1).

Most information today is provided as text. In order to process unstructured data, for example free text, it is necessary to use slightly different techniques than with structured data. Here comes text mining into play. With text mining it is possible to extract meaningful information out of documents.

As stated by Feldman et al. (2007) "Text mining is a new and exciting area of computer science research that tries to solve the crisis of information overload by combining techniques from data mining, machine learning, natural language processing (NLP), information retrieval (IR), and knowledge management"[84]

Kodratoff (1999) defines Knowledge Discovery in Texts as the "science that discovers knowledge in texts, where knowledge is taken with the meaning used in KnowledgeDiscovery in Data [..], that is: the knowledge extracted has to be grounded in the real world, and will modify the behavior of a human or mechanical agent."[85]

So NLP is all about enabling computers to understand human generated language, whether this is about text or speech. Here the focus will be on textual data because in maintenance most information is written and will be provided as a document. In those reports for example the maintenance officer will describe the current situation of a machine

[82] Bishop, 2006.
[83] Liu, 1994.
[84] Feldman et al., 2007, p. x.
[85] Kodratoff, 1999.

and if necessary defines the problem at hand. If parts are damages those are named so the reader knows which parts are to be replaced. Those for humans relative simple tasks need certain tools for computers in order to understand this. Here comes NLP into play. NLP is defined as:

As express bu Lidd (2001) "Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications."[86]

A typical first approach to indexing and querying is[87]:

- **Tokenization:** In this step longer strings of text are split into smaller pieces (tokens). So larger chunks can be tokenized into sentences, sentences into word and so on. Sometimes segmentation is used when referring to the split down of large chunks of text into paragraphs or words. In this case tokenization is referred to the breakdown to words.

- **Stop word removal:** Here words which have little meaning are removed. They might carry little meaning because of their frequency, or from a conceptual point of view. This is done because words that occur in many of the documents in the collection carry little meaning in from a frequency point of view[88].

- **Normalization** This is done to put everything on the same basis. Text will be converted to the same case, punctuation is removed, numbers are converted to their word equivalents and so on. So all words are on equal footing. There are two distinct steps:

  - Stemming, which is the process of eliminating affixes (suffixes, prefixes, infixes and circumfixes). "Stemming tends to help as many queries as it hurts."[89]
  - Lemmatization is related to stemming, but it is able to to capture canonical forms baed on the word's lemma. So "saw" is a past tense word, and its lemma is "see". It could also be a noun and the lemma would be the full form. Often words are similar, like "number and "numb". In those cases the lemmatizer has to determine the words part-of-speech, before choosing a lemma footcite[90].

After the text has been sufficiently prepared different methods can be applied to extract information. Here a short overview of several frameworks will be provided. From the simple but efficient bag-of-words, to vector space model, the more sophisticated CIMAWA[91] to a very different approach using neural networks.

Bag-of-words

Bag-of-words is one of the most popular methods for object categorization. The basic idea is to quantize each extracted key point into one of visual words. So clustering algorithm like k-means is applied. This model omits grammar and word order and only focuses on frequency. So the text is represented by a bag of words, where the word bag refers to

[86] Lidd, 2001, p. 1.
[87] Hiemstra et al., 2001.
[88] Hiemstra et al., 2001.
[89] Hiemstra et al., 2001.
[90] Hiemstra et al., 2001.
[91] Klahold et al., 2013.

concept of multisets[92].

Vector space model

It is a very simple data structure without any semantic information, but enables very efficient analysis of huge document collections. In the vector space model a document is represented as a vector in in n dimensional space. This enables comparison o documents by simple vector operations. So each element of the vector represents a word. or a group of words of the document collection. So the size of the vector is defined by the number of words. The simplest way is is using binary term vectors. So each vector element is set to one if the corresponding word is used or zero if not. So the encoding will result in a simple Boolean comparison or search[93].

CIMAWA

CIMAWA stands for Concept for the Imitation of the Mental Ability of Word Association, and it expresses a the association between keywords with a numeric value. For this the associated Gravity approach was developed. This relies on the word associations, generated by CIMAWA, combined with a clustering algorithm for multitopic detection. In short CIMAWA is a measure indicator for strongness of the word x in association withe word y. This is based on a certain windows size ws for the co-occurrence $Cooc(x, y)$ damped by a factor ζ. The next step is the calculation of the associative gravity force which is based on the attraction of each word. With the results of the previous calculations the words can be clustered in word clusters, each representing a subtopic[94].

Deep Learning for NLP

Many NLP frameworks suffer from the curse of dimensionality while learning joint probability functions of language models. So a lot of research was put into adapting Deep Learning for NLP. The following different approaches have achieved impressive results[95]:

- Word Embeddings: Distributional vectors which are based on the so-called distributional hypothesis. Word embeddings are pre-trained and the task is to predict a word based on its context.
- Recurrent Neural Network (RNN): Those Neural networks are very effective at processing sequential information. The strength of of RNN is to memorize the results of previous computations and use it for the current one.
- Reinforcement Learning: Here agents are trained to perform discrete actions by a reward system. It is often used for text summarization.

2.11 | Self-Explanatory Dasboard, Decision Support and Recommender Systems

2.11.0.1 Dashboards for Vizualization of KPIs

The word dashboard is self comes from panel placed in front of a carriage to protect the driver from beeing hit by mud or dirt. In automobiles nowadays it is a control panel in front of the driver, showing information about speed, rotation and so on. Dashboards

[92] Y. Zhang et al., 2010.
[93] Hotho et al., 2005.
[94] Klahold et al., 2013.
[95] Young et al., 2018.

in the business community are recognized as a performance management system. Key performance indicators are displayed at one glance to give the decision maker a better overview and alerts can be received[96][97].

Since the rise of Big Data and the exponential growth in data volume dashboards became more and more overloaded and complex in the need to display as many KPIs as possible[98]. This lead dashboards to become SMART (synergetic, monitor KPIs, accurate, responsive, and timely) and IMPACT (interactive, more data history, personalized, analytical, collaborative, and traceability)[99].

Often the problem is that IT engineers are not familar with the domain knowledge, and domain experts are not aware of IT technologies that can help them create dashboards. Tools like Tableau[100] and Power BI[101] can help non IT professionals to analyse huge amounts of data and create dashboards without analyzing IT professionals. Design principles for dashboards have been created in order to make dashboards more self explainatory[102]:

1. **Tuftes Data-Ink Ratio** is defined by the ratio between proportion of a graphic;s ink devoted to non redundant display information. So minimize this five principles are suggested:
   - Above all else, show the data
   - Maximize the data-ink ratio
   - Erase nondata ink
   - Erase redundant data ink
   - Revise and edit
2. **Fews Highlighting and Organizing Objective** is derived from the data-ink ratio and has to objectives reducing the non-data ink and enhancing the data ink.
3. **Shneidermans Visual Information Seeking Mantra** is a task taxonomy consisting of the tasks overview, zoom, filter, details on demand, relate, history and extract.
4. **Munzners Nested Model for Visualization Design** is four level model for visualization design and evalutation. Starting at the top with characterizing the problems, mapping those into abstract operations, designing visual encoding and interactions and creating an algorithm to execute that design automatically.

Based on the before mentioned principles Lin et al. (2018) proposes a design principle derived from the 5S workplace organization method.

- **Principle 1: Seeing Both the Forest and Trees** has two implications, first that the whole forest must be classified into different regions based on certain characteristics so that the user can identify the interesting part. The second one is to see the forest and the trees on one dashboard.

[96] Park et al., 2015.
[97] Podgorelec et al., 2011.
[98] Park et al., 2015.
[99] Malik, 2005.
[100] *Tableau* 2019.
[101] *Power BI* 2019.
[102] Lin et al., 2018.

- **Principle 2: Simplicity Through Self-Selection** one major principle is simplicity. To achieve this goal the following principles have been introduced: self-selection, significance and synthesis.
- **Principle 3: Simplicity Through Significance** is realized when information is easily actionable and is the main goal of any dashboard. An effective dashboard facilitates identification of information which can be used to make effective decisions.
- **Principle 4: Simplicity Through Synthesis** is done when a dashboard is able to reveal several charts with related but different indicators. It is proven that decision-making is difficult if different indicators show conflicting results.
- **Principle 5: Storytelling** is shown in literature that storytelling is an increasingly important point in dashboard design.

2.11.0.2 Decision Support Systems

A decision support system (DSS) is according to Aronson et al. (2005) a system which its central purpose is to support and improve decision making. To fulfill such a task a DSS must be adaptive, easy to use, robust and complete on important issues. Bonczek et al. (1980) defines DSS as a computer-based system consisting of three interacting components[103]:

- language system
- knowledge system
- problem system

According to Foster et al. (2005) defines an intelligent decision support system (IDSS) also as a Knowledge-Based DSS, which replaces the model management system with Expert system or other intelligent decision making functionality. Those are added to enhance the model based management, and can be achieved by using parts of AI, including NLP. This allows also making decisions with uncertainty. So it can not only give a recommendation, but also give a confidence level that the recommendation is a good one.

In maintenance a DSS is a computerized information system which contains domain-specific knowledge an analytical decision models which assist the decision maker by presenting information and various alternatives[104]. This combined with computational tools like a knowledge base, neural networks, fuzzy logic and Bayesian theory enhance DSS[105]. Especially in CBR has been used for decision support in areas of interactive troubleshooting like maintenance[106][107]. Yu et al. (2003) did create the a e-maintenance system for decision support, where the goal was to combine as mich knowledge from experts (agents) as possible and then negotiate between them. The approach from Benkaddour et al. (2016) was similar, they introduced also a collaborative decision support system for machine breakdowns in the nonwovens industry. In both papers the necessity for a collaboration between two or more agent for decision support is shown. Decision support systems play an important role in manufacturing. The question would be how to realise such systems. Besides PriMa a system from Benkaddour et al. (2016) and from Chan et al. (2000) have been chosen as examples for decision support systems in manufacturing.

[103] Bonczek et al., 1980.
[104] Wang, 1997.
[105] Yam et al., 2001.
[106] Yu et al., 2003.
[107] Benkaddour et al., 2016.

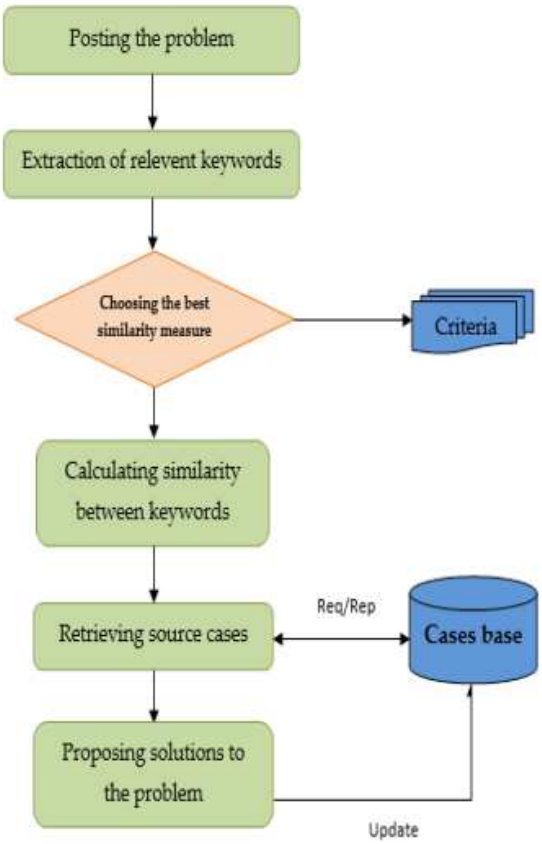


Figure 2.22: Researching solutions using CBR system, based on Benkaddour et al. (2016)

It can be seen where as Benkaddour et al. (2016) focuses solely on CBR, as seen in Figure (2.23), where as Chan et al. (2000), as seen in Figure (2.24) uses fuzzy and neural network decison support tools. By combining those two approaches analytic hierarchy process (AHP) is used. AHPC is a hierarchical decision-making model comprised of a goal, criteria and probably serval sub-criteria for each problem and decision[108]. In summary it can be said that Benkaddour et al. (2016) relies solely on past decisions, where as Chan et al. (2000) more or less ignores this know-how and focuses on other AI methods to get decision support. It must be noted that neural networks and fuzzy logic also need historical data in some way.

2.11.0.3 Recommender Systems

Recommender systems (RS) are a popular tool supporting automatic, context-based retrieval of resources[109]. RS are software tools and techniques which provides suggestions. RS are primarily focused on two groups, persons who lack sufficient personal experience or competence to evaluate all possibilities, or second a person faced with a potentially overwhelming number of alternatives. In their simplest way recommendations are offered as a ranked lists. RS are especially very popular in e-commerce and new e-business, like highly rated internet sites as YouTube, Amazon, Netflix, Tripadvisor and so on[110].

In those services mostly content-based filtering and collaborative filtering comes into play, because they are well suited for the recommendation of quality and taste products.

[108] Chan et al., 2000.
[109] Tschinkel et al., 2015.
[110] Ricci et al., 2011, p. 1-4.

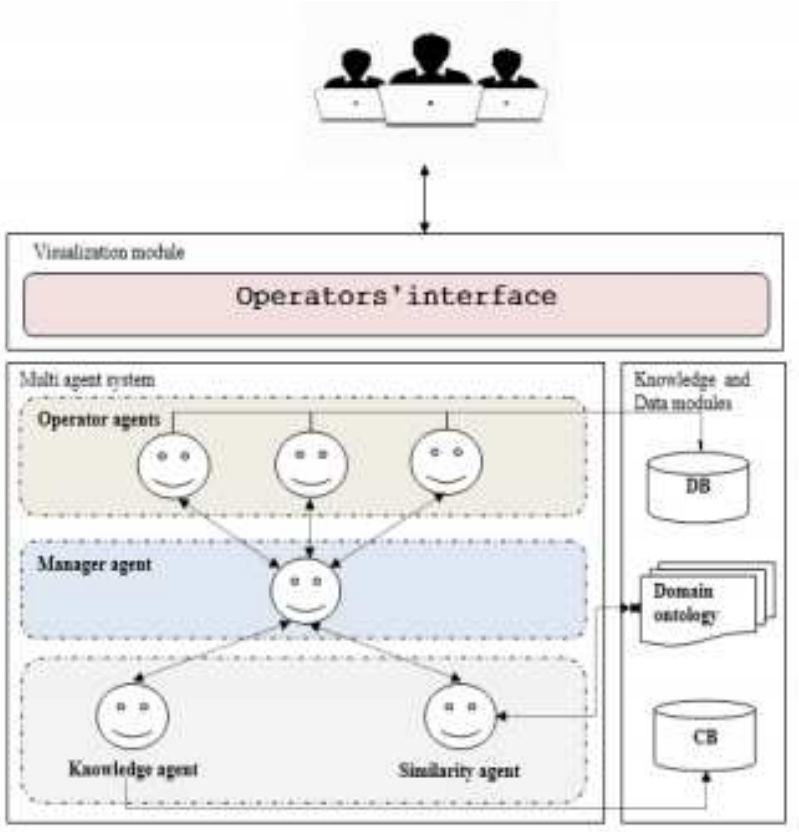


Figure 2.23: The multi agent system, based on Benkaddour et al. (2016)

In content-based filtering the goal is to match items which are similar to items previously liked, or bought by the user. The advantage is that no information about other uses is needed and specific features of a user can be captured. This is also a disadvantage because a lot of domain knowledge is needed for for building the knowledge base and it can only make recommendations building on existing features[111]. Collaborative filtering uses the ratings provided by multiple users to make recommendations. This results in no need of previous domain knowledge, it offers the discovery of new solutions and it needs no contextual features. New items and side features are a big drawback for collaborative filtering[112]. To combine both approaches hybrid recommender systems can be built. Those hybrid systems can be achieved by various approaches[113]:

- Weighted
- Switching
- Mixed
- Feature Combination
- Cascade

RS aim at two tasks. The generation of a recommendation und the use of the user of feedback after the recommendation. According to Sielis et al. (2015) RS have a cyclic architecture with four big steps:

[111] Aggarwal, 2016, p. 139.
[112] Aggarwal, 2016, p. 8.
[113] Burke, 2002a.

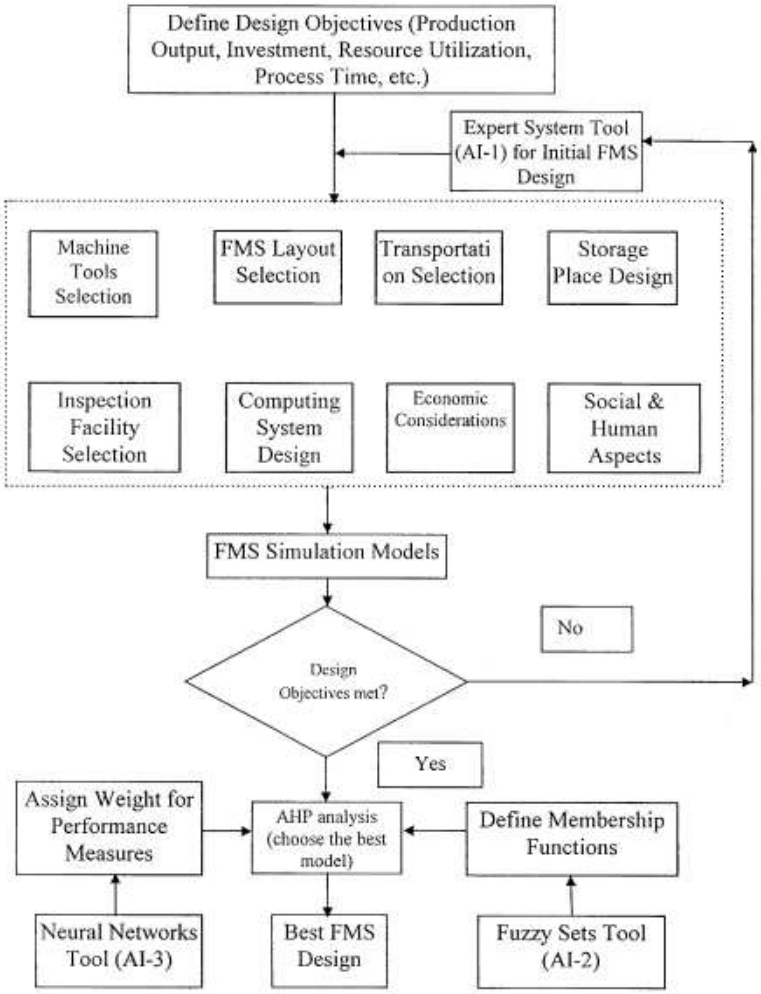


Figure 2.24: Outlinene of the intelligent decision support system for the FMS design, based on Chan et al. (2000)

- Data collection
- Data filtering
- Rank the recommended items
- Presentation of data

For example in case of apartments those methods are not suitable because collecting ratings for an apartment is not useful because they are not sold very often and buyers tastes do change because their preferences change over time. A Knowledge-based recommender helps to tackle those challenges. Knowledge-based recommender systems have also the advantage that they do not suffer from cold start problems, but they suffer from the so called knowledge acquisition bottleneck. This means that the engineers must work on converting the knowledge possessed bu domain experts to formal, executable representations[114].

There are two basic types of knowledge-based recommenders:

[114]Ricci et al., 2011, p. 187-188.

- Case-based
- Constrain-based

In case of knowledge both are similar, their main distinction is the way that similarity is calculated. Case-based recommenders determine similarity using similarity metrics. For Constrain-based recommenders knowledge bases which contain explicit rules are the main source[115].

2.12 | Knowledge-Based Systems, Knowledge-Base and Ontology

2.12.1 | Knowledge-Based Systems

"Knowledge-based systems are Information and communications technology (ICT)-based applications that can undertake analysis and research to reach conclusions about a challenge with an expert level equivalent or close to the human level."[116]

They can also be described as computer programs which exhibit a high level of intelligent performance similar to a human expert[117]. According to Schmalhofer (2001) they consist of four parts:

- A knowledge base
- A search or interferance system
- A knowledge acquisition system
- A user interface or communication system

So an expert system can ask a user questions, offer advice and demonstrate how it came to solution given. The interference system is different from other software. The algorithm which is used to solve the problem is a heuristic for reasoning[118].

2.12.1.1 Knowledge-Base

For Giaretta et al. (1995) knowledge bases are a specific version of ontologies. So ontologies are the underlying system of a knowledge base and express them in suitable, formal structures. So a knowledge base must be true in every possible world of the underlying conceptualization. A knowledge base consists of T-Box and A-Box[119], where the T-Box is for the terminological knowledge, which also includes concepts and roles and the A-Box is about the assertion nof the modelled world which belong to the concept.

Ontologies can be used for recommender systems, because an ontology contains a set of concepts like entities, attributes and properties related to a domain along with their definitions. Those are represented by languages like Web Ontology language (OWL) and Resource Description Framework (RDF). So moreover those ontologies can be used alongside other tools such das data mining and machine learning to give better results. So ontology based recommenders are knowledge-based recommender systems which need a

[115] Ricci et al., 2011, p. 188.
[116] Babu et al., 2017.
[117] Schmalhofer, 2001.
[118] Ferguson et al., 2003.
[119] Dengel, 2012.

knowledge base to store those information[120].

For case-based recommender systems specific cases are specified and used as anchor pints as can be seen in Figure (2.25). With the defined similarity metrics which are carefully defined in a domain specific way the query is matched to a case. The returned result can be used as target cases with some user modifications[121].

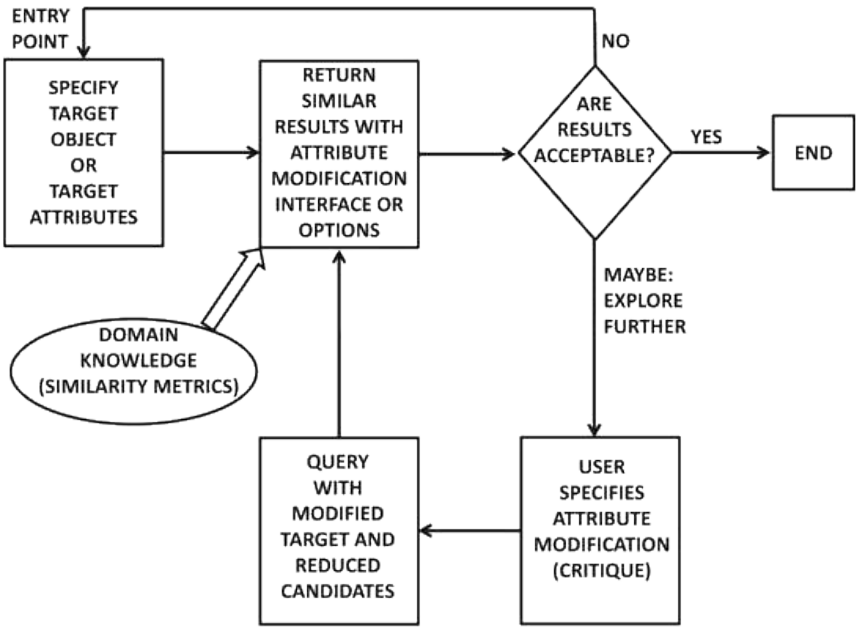


Figure 2.25: Interactive process of a case-based interaction in a knowledge based recommender system from Aggarwal (2016).

So in hybrid ontology-based recommendation learning ratings are coupled with ontological domain knowledge which improves similarity matching. So once the ontological concepts are fully mapped, normal recommendation approaches are applied. This coupled with the lack of the cold start problem, sparsity and overspecialization makes ontology based recommender systems well suited for recommending learning resources. To compute the similarity between the ontology and the query the cosine similarity is well suited according to Tarus et al. (2017).

Knowledge bases can include rich information from structured and unstructured data with different semantics and so knowledge bases gained more attention in recent times. In recent years several typical knowledge bases have been constructed. Those include academic projects like YAGO, NELL, DBpedia, and DeepDive, as well as commercial projects, such as Microsofts Satori and Googles Knowledge Graph[122]. A knowledge base for a recommender system, here for movies, can be divided into three parts according to F. Zhang et al. (2016):

- **Structural knowledge** can be defined as a heterogenous network with multiple type of entities and multiple links to express the structure of the knowledge base.
- **Textual Knowledge** gives the main information about the entity it represents

[120] Tarus et al., 2017.
[121] Aggarwal, 2016.
[122] F. Zhang et al., 2016.

- **Visual Knowledge** can add information to the textual knowledge.

2.12.1.2 Ontologies

"An ontology is a set of definitions of content-specific knowledge representation primitives: classes, relations, functions, and object constants."[123]

Ontologies are artifacts and represent the knowledge represented to a specific topic or domain. Ontologies have gained more attained in recent years to the rise of the semantic web including standard ontology representation languages (especially OWL) and the availability of tools to manipulate them, like parser libraries, reasoners and ontology editors (like Protege)[124].

Ontologies also follow an evolutionary cycle, see (2.26), proposed by Zablith et al. (2015), which has fast similarities with the CRISP-DM model, see Figure (2.1). The ontology in the middle servers as input for all tasks and it receives input from the Managing Changes task. When the changes are applied the cycle is repeated. The five main steps are:

- Detecting the need for evolutionary
- Suggesting Changes
- Validating Changes
- Assessing Evolution impact
- Managing Changes

Dengel (2012) describes a ontologies as a formal, explicit specification with a common conceptualization:

- the use of common symbols and terms in the sense of syntax,
- the common understanding concerning their meaning, i.e. semantics,
- the classification of terms in the form of a taxonomy,
- the networking of concepts with the help of associative relations while simultaneously
- Determination of rules and definitions about which relations are meaningful and allowed.

Description logic is the base of an ontology according to Ansari, Khobreh, et al. (2018) and Russell et al. (2009). Ontologies consist of classes, relations and rules which describe conceptualizations as wall as instances. Those describe individual elements of a domain. So those elements are described by Dengel (2012) as:

- Classes or often also concepts describe different categories of terms. Those are mostly referred as taxonomies, hierarchies of terms. So inheritance mechanisms are already cindered and can be applied.
- Relationships describe dependencies between classes. In taxonomies the 'is one' relation is described per default. All others most be described additionally. Relations also often describe other properties of classes.

[123] Gruber, 1992.
[124] Zablith et al., 2015.

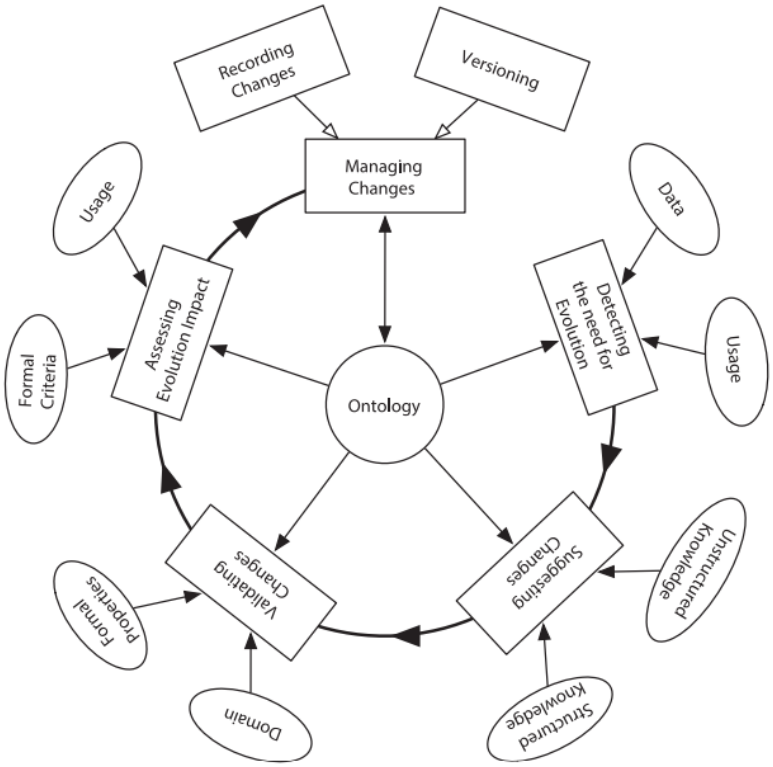


Figure 2.26: Ontology evolution cycle from Zablith et al. (2015).

- Rules of axioms are used to describe the circumstances of domains which are true.
- Instances represent real existing elements of a domain.

In this example (2.12.1.2) <Type>,<Machine> and <CNC> are defined. The CNC machine EMCO CONCEPT TURN 460 is defined with <Type> in the property <cnc> in the class <CNC>. The instance <ID> is as <Machine> as <CNC>. Based on Wikipedia (2019) and[125].

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns="http://localhost:8080/OWLBuergerInformation.owl#"
xml:base="http://localhost:8080/OWLBuergerInformation.owl">

<owl:Ontology rdf:about=""/>

<owl:Class rdf:ID="Type"/>
<owl:Class rdf:ID="Machine"/>
<owl:Class rdf:ID="CNC">
  <rdfs:subClassOf rdf:resource="#Machine"/>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Type"/>
      <owl:hasValue rdf:resource="#cnc" rdf:type="#Type"/>
```

[125] W3, 2019.

```
        </owl:Restriction>
      </owl:equivalentClass>
    </owl:Class>

    <owl:ObjectProperty rdf:ID="Type"
       rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
      <rdfs:range rdf:resource="#Type"/>
      <rdfs:domain rdf:resource="#Machine"/>
    </owl:ObjectProperty>
    <owl:DatatypeProperty rdf:ID="name"
       rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:domain rdf:resource="#Machine"/>
    </owl:DatatypeProperty>
    <owl:DatatypeProperty rdf:ID="name"
       rdf:type="http://www.w3.org/2002/07/owl#FunctionalProperty">
      <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:domain rdf:resource="#Machine"/>
    </owl:DatatypeProperty>

    <Machine rdf:ID="CNC1" name="EMCO␣CONCEPT␣TURN␣460">
      <Type rdf:resource="#female"/>
    </Machine>
  </rdf:RDF>
```

Listing 2.1: OWL code example

In computer science and especially AI ontologies are used for knowledge sharing and reuse effort for efficient engineering of knowledge-based systems. So Knowledge Management is defined by Staab et al. (2010) as:

- Systematically managed organizational activity
- Strategic key resources viewn as implicit and explicit knowledge
- Aims at improving the handling of knowledge
- To achieve organizational goals
- By employing tools, techniques, and theories from areas like IT, strategic planning, business process management and others
- To achieve a planned impact on people, processes , technology and culture

2.12.1.3 Knowledge graph

There are many definitions out there on how to best describe knowledge graphs. The word it self was coined by Google to describe any graph-based knowledge base[126], but they are not merely a graphical representation of ontologies[127]. Two have been selected, because their definitions is most suitable for knowledge graphs in manufacturing, especially maintenance.

[126] Färber et al., 2018.
[127] Ehrlinger et al., 2016.

A knowledge graph (i) mainly describes real world entities and their interrelations, organized in a graph, (ii) defines possible classes and relations of entities in a schema, (iii) allows for potentially interrelating arbitrary entities with each other and (iv) covers various topical domains.[128]

Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities.[129]

First there are various ways to express knowledge in graphical forms, Dengel (2012)uses Sowas distinction:

- **Definition networks** get their knowledge from hierarchical categorization of concepts. A famous example is the Pophyrian tree, and such networks are also used for taxonomies.
- **Assertional networks** they additionally allow non taxonomical (hierarchical) relations. They allow non taxionomical and non partonomical relations.
- **Implicational networks** are acyclic directed graphs that represent causal dependencies between events represented as graph nodes. Bayesian networks, as seen in chapter (2.9.0.2) are prominent representatives.
- **Excetuable networks** are a special type of semantic networks that allow the dynamics of a system or a process to be to depict, for example Petri nets.
- **Learning networks** can change over time by expanding themselves with new nodes and edges, or reducing themselves when certain nodes or edges disappear. Weigths can be allocated to those nodes, for example the probability from Bayesian networks.
- **Hybrid networks** combine aspects from the networks above. So Bayesian networks can become learning networks if the necessary learning algorithms are implemented.

Those networks can be graphically represented in the way of concept maps. They have been developed by Josef Novak in the 1960s as a way for knowledge structuring and visualization. For Novak a concept map is an abstract description of specific ideas of knowledge domain and should not be confused with a text analysis method of the same name.

Here Ehrlinger et al. (2016) argues that ontologies are not really different from knowledge bases but are sometimes erroneously classified as being database schemas. But ontologies, especially consist not only of classes and properties but also hold instances. So it can be argued that knowledge graphs are very large ontologies. Other say that knowledge graphs are superior ontologies with built in reasoners[130]. One of the most cited knowledge graph engines is Google Knowledge Graph. There is hardly any information available on the technology, but there is some on Yahoo's Spark and the Knowledge Vault. Both use Semantic Web standards such as RDF.

[128] Paulheim, 2017.
[129] Kroetsch et al., 2015.
[130] Ehrlinger et al., 2016.

3 | Realising PriMa

3.1 | PriMa Architecture

The focus of this thesis lies in building a cutting edge knowledge pipeline and to automatize PriMa[1]. In order to perform this tasks an understanding of the underlying architecture of PriMa must be gained. The architecture of PriMa, as seen in Figure (3.1) builds upon a four layer model. Those layers are:

- Data management
- Predictive data analytic toolbox
- Recommender and decision-support dashboard
- Semantic-based learning and reasoning

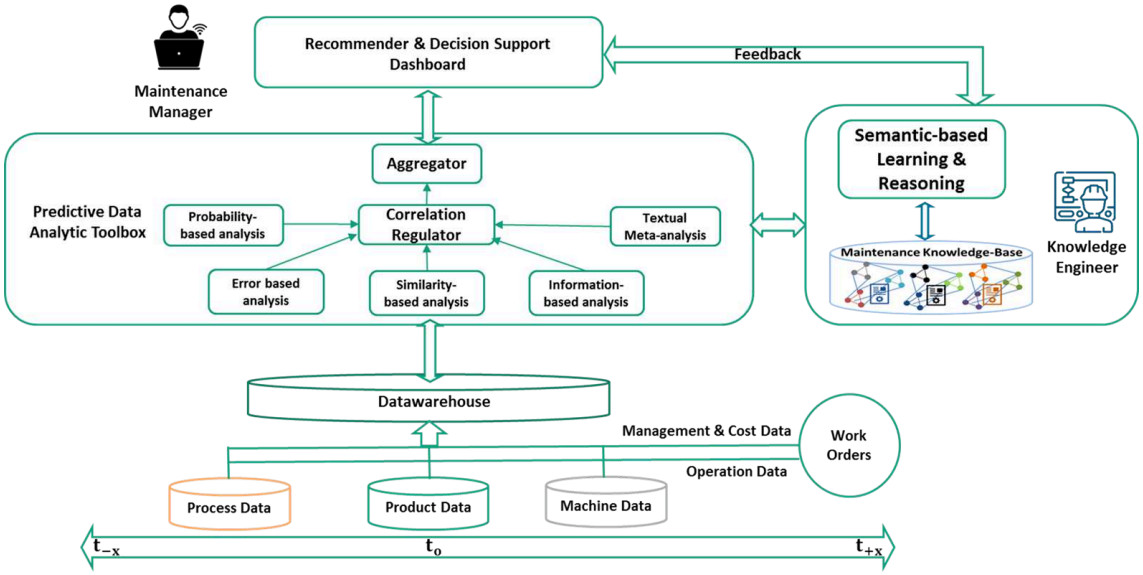


Figure 3.1: Overall architecture of PriMa, by Ansari, Glawar, et al. (2019)

3.1.1 | Data Management

This layer consists of the data input and the data warehousing solution. The data warehouse must be able to collect maintenance records and cost data as well as operational data from three dimensions[2]. Those have been extended by a fourth dimension expert Knowledge.

[1] Ansari, Glawar, et al., 2019.
[2] Ansari, Glawar, et al., 2019.

- Machines
- Processes
- Products
- Expert Knowledge

The data flow in Figure (3.1) consists in horizontal and vertical view of all processes, actors and production (maintenance) management systems. In the horizontal view either maintenance operations, or maintenance management, where as in the vertical view the semantic interlinking of operations and management data is the focus. The PriMa model must be able to deal with multimodality of maintenance records.

The data warehouse, as seen in Figure (3.2), is designed with Data Vault 2.0. It consists of four main hubs which are:

- Maintenance organization
- Production planing and controlling
- Cost controlling
- Event tracking

Maintenance records consist of heterogenous data, which means that they may be directly analyzed or require processing. The structured comes from condition monitoring systems and includes sensor data from conditions and environmental records. The unstructured or semi-structured data consist of maintenance records, but also audio, images or video records. Those need preprocessing, which is at the moment not included in PriMa.

3.1.2 | Predictive data analytic toolbox

In this layer ML and knowledge discovering algorithms are used in order to gain information from the data ingested before. In the papers from Ansari, Glawar, et al. (2019) and Nemeth et al. (2018) four families of ML algorithms were discussed in more detail for use in PriMa:

- information-based
- similarity-based
- probability-based
- error-based learning

3.1.3 | Recommender and decision-support dashboard

The goal of the third layer is to present the outcome of the previous one in a meaningful way. For this the outcome of the data analytic algorithms must be prepared to recommend actions and decision alternatives. Before that the data must be correlated to avoid errors and then aggregated.

With those results a link must be generated to the maintenance knowledge base. For this reasoning methods like CBR come in hand, as seen in Section (2.6.3). The link must be activated in each decision-making and problem-solving iteration. This means not only querying existing knowledge, but also enriching existing one and generating new one.

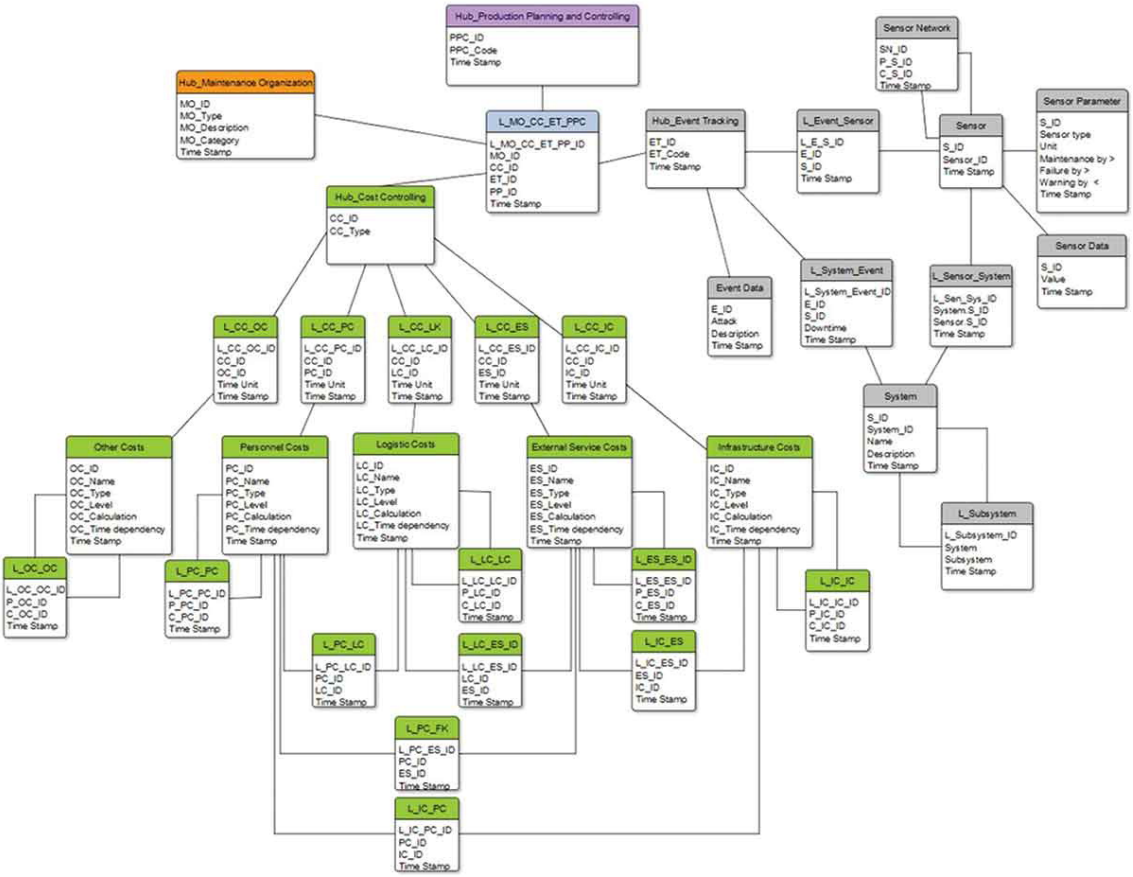


Figure 3.2: PriMas database schema Interlinking multimodal data and building a scalable data warehouse, by Ansari, Glawar, et al. (2019)

3.1.4 | Architecture comparison

While automatizing PriMa the design of its architecture should be discussed. Other authors like O'Donovan et al. (2015), Cheng et al. (2018) and R. Ranjan (2014) also proposed big data pipelines with similar goals.

Starting with data data ingest, the first difference is, that in PriMa data ingest process is discussed, but not in great detail. Cheng et al. (2018) discusses the data ingest, as seen in Figure (3.3), is in much more detail. Especially concerning the type of gateways needed for the IoT devices is discussed in depth, because they have been identified as most important. Also the ingestion engine is discussed in great detail[3]. O'Donovan et al. (2015) discusses briefly the common interfaces for IoT devices and its gateways. R. Ranjan (2014) does not focus on the protocols and gateways needed, but focuses on the ingestion process it self, as seen in Figure (3.4).

It can also be seen in in these Figures (3.3), (3.4) and (3.5) that they all use a data storage based on NoSQL. Where a good database schema for a relational database provides more speed while querying, as seen in Section (2.4.2), a NoSQL data base is more flexible, because the schema it self must not be defined beforehand. It's also noticeable that the other architectures put the analytical layer before the data storage.

The analytical layer is very similar in the four examined papers. Most do not discuss

[3]Cheng et al., 2018.

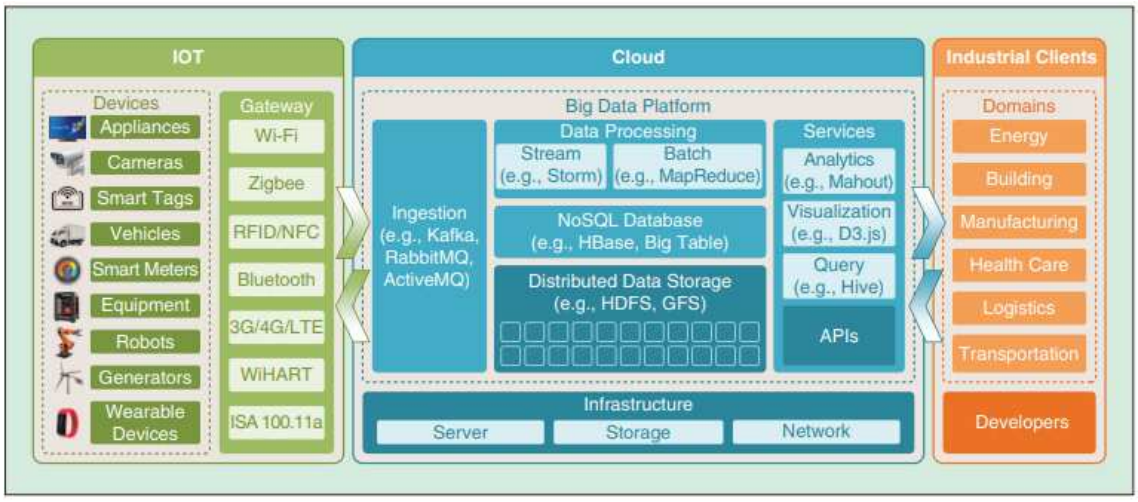


Figure 3.3: The industrial CPS systems powered by big data. Smart devices could communicate with the cloud using protocols like AMQP. The big data platform hosted in the cloud enables efficient data ingestion, warehouse, processing, analytics, and visualization. Industrial clients from energy, building, manufacturing, health care, logistics and transportation domains can gain insight into the big data through the service interfaces provided by the cloud. RFID: radio-frequency identification; 3G: third generation, LTE: long-term evolution; ISA: International Society of Automation; AMQP: Advanced Message Queuing Protocol; GFS: Google File System; API: Application Programming Interface; NFC: near-field communication; WiHART: Wireless Highway Addressable Remote Transducer Protocol, by Cheng et al. (2018)

their used methods except Ansari, Glawar, et al. (2019), however go in detail into the requirements[4][5]. For others the framework (for example Apache Storm, Hadoop Mahout, Apache Spark) is the main discussion point and go here into great depth reviews[6][7]. Important to notice is that only Ansari, Glawar, et al. (2019) discusses structured and unstructured data mining algorithms at all, only R. Ranjan (2014) mentions briefly that a use of NoSQL databases is suitable for unstructured data.

It is important to point out that due to their different designs (analytical layer before data storage) different frameworks for data analytics are discussed[8][9][10]. They focus is mainly on data stream analytics, instead of classic analytical approaches.

The recommender and decision-support dashboard as described in Ansari, Glawar, et al. (2019) is quite unique, only O'Donovan et al. (2015) mentions the need of a dashboard, but neglects the recommendation, or decision-support character of the dashboard. All of the examined papers mention the decision-support character of data analytics but do not specify on how to provide those support. Only R. Ranjan (2014) mentions briefly application programming interfaces (API) which allow additional decision-support applications easy communication.

[4] O'Donovan et al., 2015.
[5] R. Ranjan, 2014.
[6] R. Ranjan, 2014.
[7] Cheng et al., 2018.
[8] O'Donovan et al., 2015.
[9] Cheng et al., 2018.
[10] R. Ranjan, 2014.

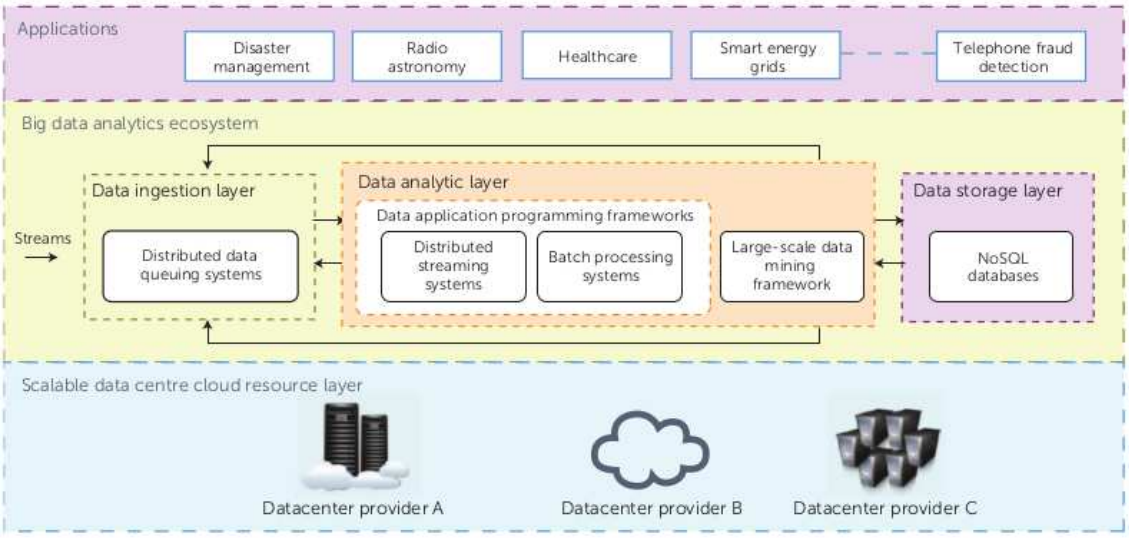


Figure 3.4: A high-level architecture of large-scale data processing service. The big data analytics architectures have three layers: data ingestion, analytics, and storage and the first two layers communicate with various databases during execution, by R. Ranjan (2014)

Layer	**IBDP**	**ICPS**	**LSDPS**	**PriMa**
Data ingestion	Defined	Defined	Not Defined	Not Defined
Data store	Non-Relational	Non-Relational	Non-Relational	Relational
Analytics layer	Stream Analytics	Stream Analytics	Stream Analytics and conventional Analytics	Conventional Analytics
Recommender system	Not	Not	No	Yes but not defined
Semantic layer	No	No	No	Yes

Table 3.1: Comparison of Industrial Big Data Pipeline (IBDP)[11], Industrial CPS (ICPS)[12], Large-Scale Data Processing Service (LSDPS)[13] and PriMa[14].

Semantic-based learning and reasoning is the core contribution of the PriMa system to the field of prescriptive maintenance systems. The unique approach combines traditional expert knowledge with the capabilities of big data analytics. All other systems do not leverage this knowledge and rely solely on the information from their algorithms. R. Y. Zhong et al. (2017) sees knowledge in the data used, but is not investigation knowledge from other sources, where as Legat et al. (2014), Engel et al. (2018) and Ullrich (2016) who work with knowledge in form of ontologies. However, they do not do not provide a framework like PriMa. Those findings have been summarized in Table (3.1). The biggest differences in comparison to the other architectures is he different data store, the missing stream analytics, the recommender system and the unique semantic layer.

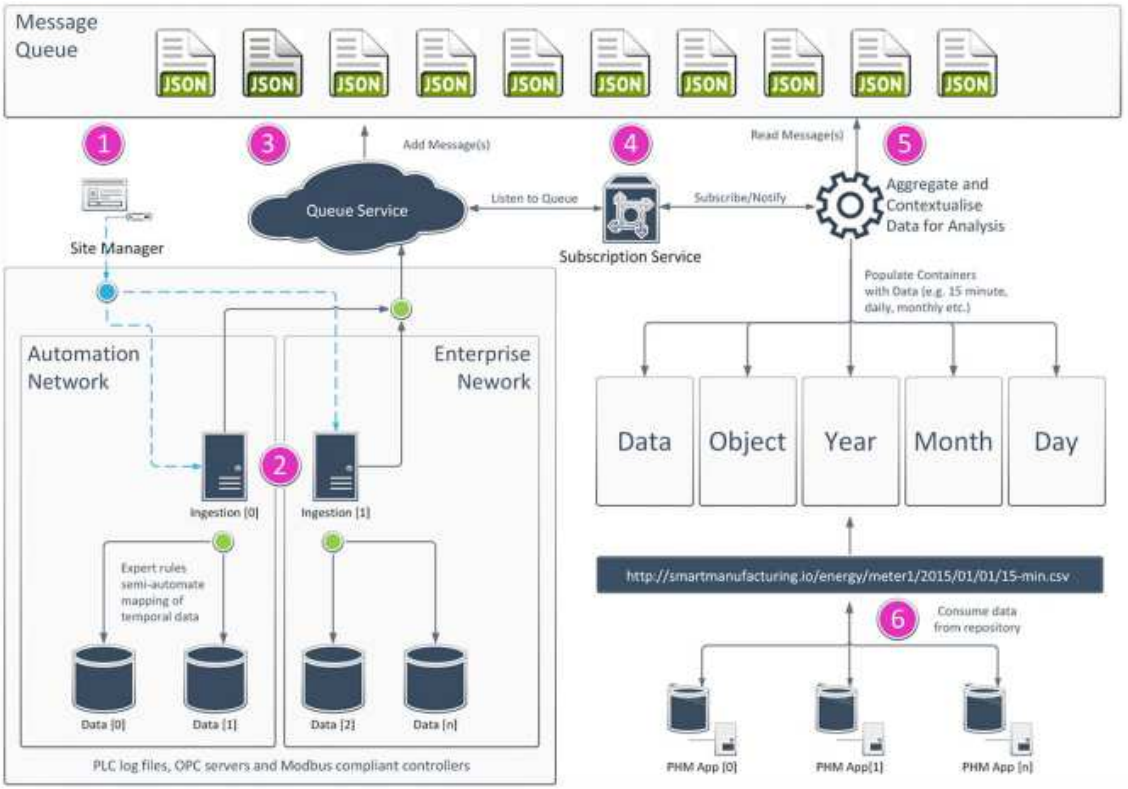


Figure 3.5: Big data pipeline architecture and workflow, by O'Donovan et al. (2015)

3.2 | An automated PriMa model

A basic requirement for a maintenance model is the possibility to apply it in a real world application. This has been done extensively by Ansari, Glawar, et al. (2019) using a four-step methodology[15], as seen in Figure (3.6). The model used is based on the CRISP-DM model, as explained in chapter (2.1). The six phases of CRISP-DM have been transformed into four stages. In the fours step model:

- Data Acquisition and preprocessing: Consisting of domain experts as well as data and knowledge engineers
- Data Analysis and Simulation: Which focuses on building and simulating a predictive model
- Reaction Mode: In this step a set of rules is defined which satisfy the given requirements
- Prescriptive Maintenance Decision Support System: Focuses on building a framework which includes prescriptive measures as well as the development of a maintenance control centre.

Those steps have been implemented when developing the the approach shown in Figure (3.7). While Ansari, Glawar, et al. (2019) proved that PriMa can be used in a real life manufacturing context, this thesis shows a way of how to automatize those approaches by defining frameworks for the data ingest, suggesting data storage approaches which fulfill demands identified, as seen in (2.4.2), but also from other literature sources which

[15] Matyas et al., 2017.

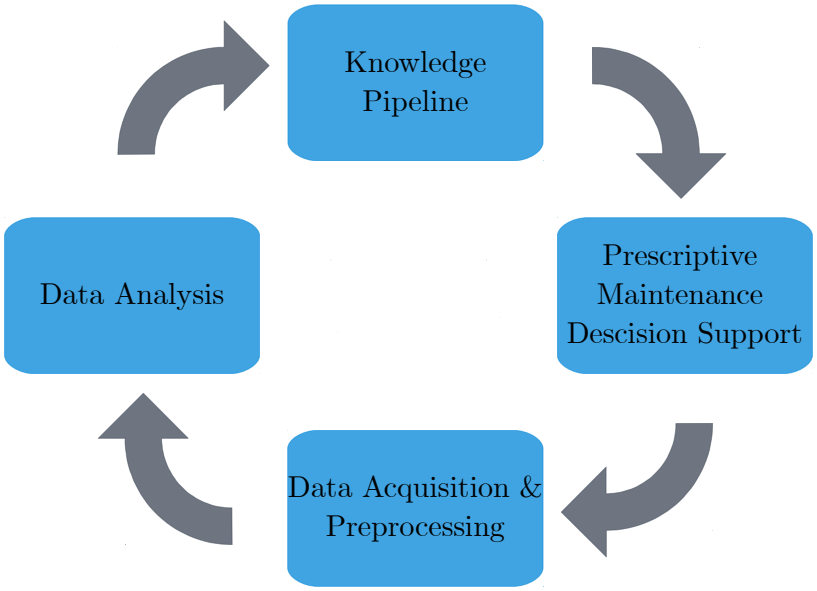


Figure 3.6: Procedural approach for prescriptive maintenance planning, adopted from Ansari, Glawar, et al. (2019)

provide specific requirements for the data storage in an IoT/CPPS context[16][17][18][19].

The data analysis layer will be designed as shown in Ansari, Glawar, et al. (2019) and minimum working examples (MWE) examples of three algorithms will be provided. To fulfill the automatization approach their output will be automatically aggregated.

For the knowledge pipeline it will be shown that text from a maintenance report can be automatically extracted and used for a CBR approach. The case based used will be an ontology based knowledge-base. With the reasoning approach provided by CBR a both way update process with the information's of the data analysis layer will be realized.

In the end all of those information will be gathered in a dashboard which fulfills the identified requirements from chapter (2.11.0.3). For a better understanding a mock-up will be shown.

[16] O'Donovan et al., 2015.
[17] Cheng et al., 2018.
[18] R. Ranjan, 2014.
[19] R. Y. Zhong et al., 2017.

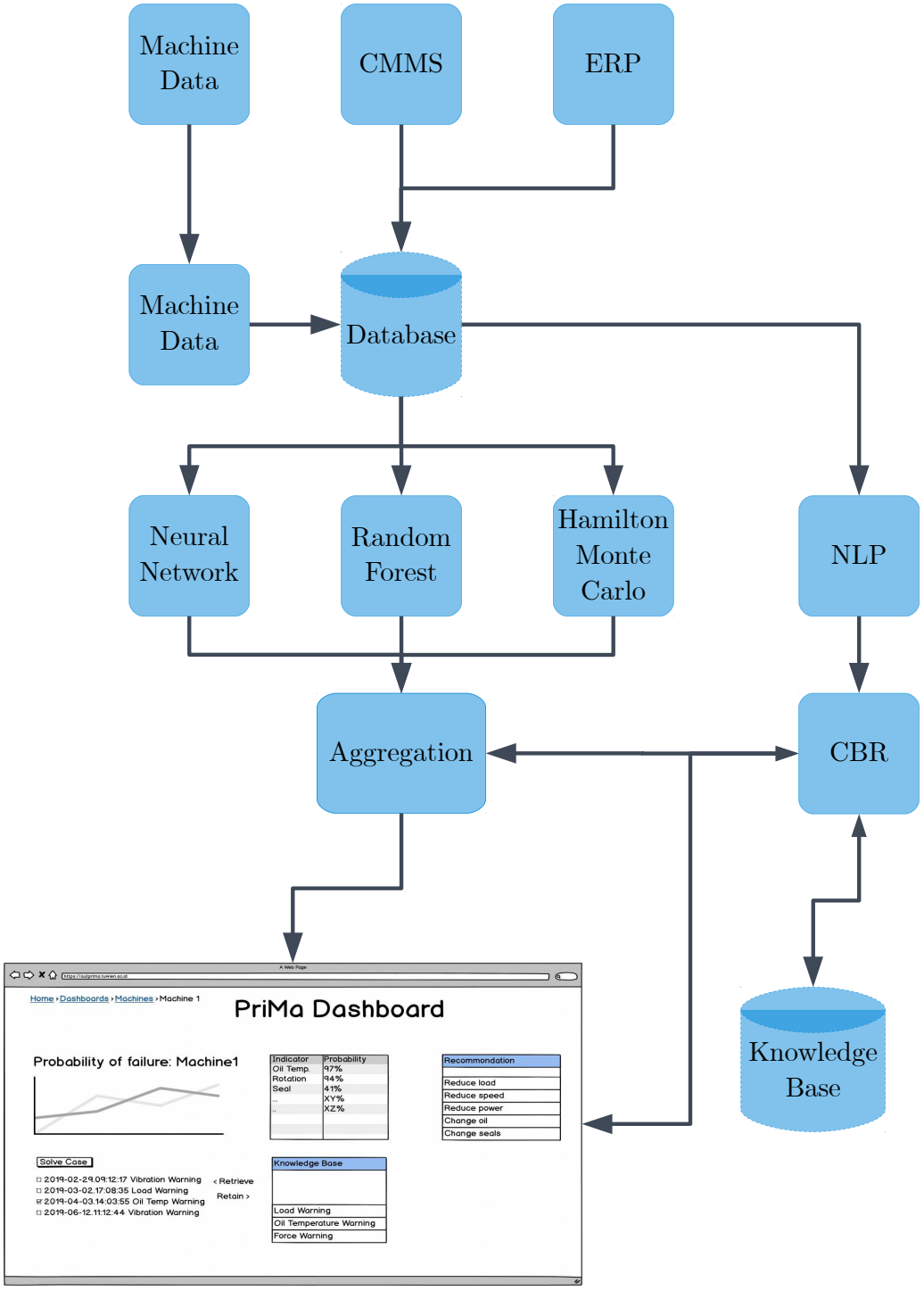


Figure 3.7: Flowchart of AutoPriMa

3.3 | Realisation of Layer one: Data management

3.3.1 | Data Streams

It is crucial for an automated maintenance model like AutoPriMa[20] to be able to handle a high volume of data stream, with the possibility to stream various types of data, ideally in real-time.

Any data ingestion process in the manufacturing areas, especially maintenance, must meet some certain requirements. O'Donovan et al. (2015) identified the following criteria:

- **Legacy integration**: Due to the coexistence of legacy devices in the industrial big data pipeline those must be integrated and using the ingestion engine.
- **Cross-network communication**: Ingestion engines can operate across different networks due to lack of local dependencies.
- **Fault tolerance**: The ingestion engine must be reliable and resilient. In case of a fault the instructions must be able to assign to a different ingestion process.
- **Extensibility**: The big data pipeline must be design in a way that new data sources an be easily integrated.
- **Scalability**: The ingestion process must be scalable horizontally, as to add multiple ingestion engines.
- **Openness and accessibility**: The ingestion engine should use open standards which facilitate communication and data exchange among other components of the pipeline.

For transmitting data various gateways are available[21] namely:

- Wi-Fi
- Zigbee
- RFIDNFC
- Bluetooth
- 3G4GLTE
- WiHART
- ISA 100.11a

The above-mentioned gateways support transmitting data with the with the help of a protocol. They vary from technology to technology, but when Wi-Fi, or 3G4GLTE is used the more popular protocols are listed as below[22][23][24]:

- AMQP

[20] Ansari, Glawar, et al., 2019.
[21] Cheng et al., 2018.
[22] Cheng et al., 2018.
[23] R. Ranjan, 2014.
[24] O'Donovan et al., 2015.

- MQTT
- OPNWIRE
- REST
- STOMP
- XMPP

For IoT devices streaming technologies like ActiveMQ, Apache Kafka and RabbitMQ become very popular in recent time. In a recent study about Kafka streaming 86% of the responders reported that the number of their systems Kafka is rising[25]. Those publishsubscribe massaging systems provide various benefits over requestresponse systems, among other things the fact that the various speeds of the different sites do not effect the transmitting speed of the whole system[26]. Apache Kafka and RabbitMQ have been selected, because of their popularity[27], for further investigation. Before a comparison is made a short description of both services is provided.

Protocol	**Strength**	**Weakness**
MQTT	Simple	High latency in high sampling rate
	Lightweight	No automatic discovery
	WebSocket support	Data type simple
	Implementations for embedded devices	Not secure at protocol level
AMQP	Very extended	Not for realtime
	High interoperability	No automatic discovery
	Designed for critical applications	No ensures interoperability
XMPP	Simple	High bandwidth consumption
	Widely supported	Data type simple
	Extensible	poor fault resistance

Table 3.2: Strength and weaknesses for each Machine-to-Machine protocol. Modified from Talaminos-Barroso et al. (2016).

In Table (3.2) a comparison of different Machine-to-Machine protocols can be seen. Three protocols and their most important advantages and disadvantages have been chosen from a study from Talaminos-Barroso et al. (2016).

Apache Kafka

Apache Kafka is an open-source, distributed, publishsubscribe messaging broker. It maintains feeds of categorized topics. A producer can publish messages to all topics, as seen in Figure (3.8). Those messages are stored and replicated. The distributed nature of Kafka offers the possibility to device each topic into multiple partitions, where each broker maintains one ore more of the partitions, because of lead balancing. The consumer which subscribes to one or more topics acquired the data from the brokers. The fault tolerance

[25] Dauber, 2019.
[26] Cheng et al., 2018.
[27] Dobbelaere et al., 2017.

of Kafka comes from its replication. Each partition has one broker acting as leader, which is in charge of all readwrite requests. If the leader fails, a new one is selected from among the remaining followers[28][29]. It offers three key capabilities as follows[30]:

- Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system.
- Store streams of records in a fault-tolerant durable way.
- Process streams of records as they occur.

The best areas of application for Kafka are[31]:

- Pub/Sub Messaging when routing is simple and, or the throughput per topic is beyond what RabbitMQ can handle
- Scalable Ingestion System, Kafka offers high input and is already integrated in platforms such as Apache Spark and Apache Flink
- Data-layer Infrastructure, because of the durability and efficient multicast Kafka can serve as a connector to various batch and streaming services
- Capturing Change Feeds, Kafka's log centric design makes ich excellent for this purpose
- Stream Processing, Kafka offers a lightweight stream processing library

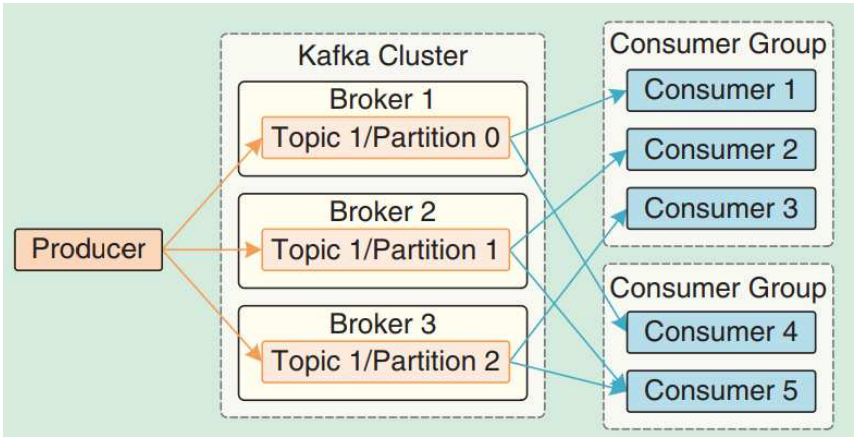


Figure 3.8: The schema of Apache Kafka, reprinted from Cheng et al. (2018)

RabbitMQ

Like Kafka, RabbitMQ is an opens-source massaging broker. It is written in Erlang and is therefore best suited for distributed applications and offers high parallelism, and fault tolerance. As seen in Figure (3.9) messages are published to exchanges, those then distribute copies of those messages to ques using rules, which are called bindings. With the optional routing key used by those bindings it is possible to bind a queue to an exchange. A broker can provide four exchange types, a topic exchange type is used for implementing a publishsubscribe pattern, namely[32][33]:

[28] Cheng et al., 2018.
[29] Dobbelaere et al., 2017.
[30] Foundation, 2019.
[31] Dobbelaere et al., 2017.
[32] Cheng et al., 2018.
[33] Dobbelaere et al., 2017.

- direct
- fanout
- topic
- headers

A exchange route messages with one or more queue based on the matching between the routing key and the pattern used for the exchange.

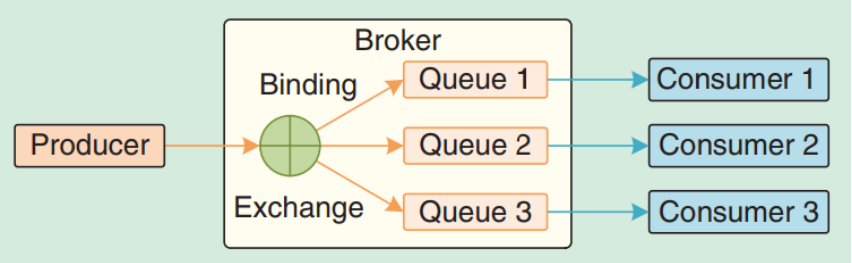


Figure 3.9: The schema of RabbitMQ, reprinted from Cheng et al. (2018)

The best areas of application for RabbitMQ are[34]:

- PubSub Messaging is what RabbitMQ was created for.
- Request-Response Messaging, is a strong suit of RabbitMQ, where it offers a lot ofn support for Remote Procedure Call (RPC) style communication
- Operational Metrics Tracking is a something where RabbitMQ's complex filtering comes in hand if realtime processing is needed.
- Underlying Layer for IoT Applications Platform. RabbitMQ can be connected to individual operator nodes with a sub 5ms latency, up to 40Kpps for a single node, excellent visibility on internal matrices, easy test and debug cycles and support for MQTT with a sophisticated routing capability and the possibility to handle very large number of streams

Apache Kafka vs. RabbitMQ

As mentioned supported messaging protocols are a very important selection criteria for a message broker. Nearly all mentioned protocols are supported by RabbitMQ, as seen in Table (3.3), whereas Kafka uses its own binary protocol. This is due to the restrictions of classic protocols, and it is possible to establish a truly distributed messaging system with Kafka's binary protocol[35].
It must be said that there are a number of open-source projects which give Kafka to use protocols like MQTT, but Kafka it self does not natively support any protocols.

Some requirements can not be covered by Apache Kafka, or RabbitMQ on its own, so a combined use of both can be the best option[36]:

- **Option 1**: RabbitMQ, followed by Kafka is the best choice if RabbitMQ is the best architecture, but the streams need long term storage
- **Option 2**: Kafka, followed by RabbitMQ is the best choice if the throughput is very high, but a complex routing is needed.

[34] Dobbelaere et al., 2017.
[35] Cheng et al., 2018.
[36] Dobbelaere et al., 2017.

Protocol	Apache Kafka	RabbitMQ
AMQP	No	0-8,-9,1
MQTT	No	Yes
Openwire	No	No
REST	No	Yes
STOMP	No	Yes
XMPP	No	Over gateway

Table 3.3: Protocols supported by Apache Kafka and RabbitMQ. Adopted from Cheng et al., 2018.

3.3.2 | Data Warehouse/Data Lake

After all data have been collected and routed to the right recipient they must be stored, in order to be available for further investigation. In the PriMa model the data store is designed as seen in Figure (3.2). Relational data bases have a fixed schema and are secure and are able to make complex queries. Non-relational databases are well suited for large amounts of data with little structure. They are alo very good when a massive growth is expected, but making big joints over large data sets is not a strong suit of non-relational databases.

With the help of the requirements formulated in Section (2.4.2) and the requirements defined by Ansari, Glawar, et al. (2019) and O'Donovan et al. (2015) the following criteria have been defined for the technical implementation of PriMa's data warehouse. The following criteria have been adopted from the data ingest from O'Donovan et al. (2015) and adopted for data storage:

- **Legacy integration**: The warehouse of PriMa must be able to integrate a various field of data sources[37].
- **Fault tolerance and high availability**: The data warehouse must be replicated, and the high availability must be implemented on the infrastructure site.
- **Extensibility**: The storage system of a prescriptive maintenance system is not non-political structure but compromises of different elements, which need to be flexible enough to be extended during its lifetime.
- **Scalability**: Given the nature of a big data pipeline high volumes of data input must
- **Security**: The data warehouse must serve as an trustworthy foundation for decision making and so be secure against manipulation.
- **Openness and accessibility**: The data warehouse must support open formats an available data exchange through various channels (eg. HTTPS, API,...)

The goal is to transform and combine those requirements defined in Ansari, Glawar, et al. (2019) and those developed in this thesis into a industry solution. Three major players in the cloud computing market have been identified and their products will be compared. Those companies are Amazon Web Services (AWS)[38], Google Cloud Platform (GCP)[39],

[37] Ansari, Glawar, et al., 2019.
[38] AWS, 2019.
[39] Platfrom, 2019.

Microsoft Azure (MA)[40]. They all offer Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

Type of Storage	AWS	GCP	AM
File Storage	Amazon Elastic File System	Cloud Filestore	Azure Files
Object Storage	Amazon Simple Storage Services	Google Cloud Storage	Blob Storage
Block Storage	Amazon Elastic Block Store	Persistent Disk	Azure Managed Disks
Backup	Amazon S3 can be used	No backup solution	Backup (built in)

Table 3.4: Data storage comparison of Amazon Web Services (AWS)[41], Google Cloud Platform (GCP)[42], Microsoft Azure (MA)[43] .

The first comparison will be based on the IaaS for storage, as seen in Table (3.4). The price of the storage is not part of this comparison, because they are quite similar and without a detailed use case it is hard to build up reliable numbers. The solutions offered range from simple file storage to, object storage for unstructured data to block storage. Their services are very similar, however MA takes definitely the lead here by providing an implemented backup and disaster recovery.

For ML and AI applications computing power is a major decision reason. As seen in Table (3.5) the services are very similar, all competitors offer auto scaling, but GCP offers predefined machines in addition. However, AWS and MA offer a service which lets you create a virtual private server.

Type of Computing Service	AWS	GCP	AM
PaaS	Elastic Beanstalk	Google App Engine	Azure Cloud Services
Scaling	AWS Auto Scaling	Done by Instance groups	Azure Autoscale
Virtual server	Amazon Elastic Compute Cloud	Compute Engine	Virtual Machines

Table 3.5: Compute service comparison of Amazon Web Services (AWS)[44], Google Cloud Platform (GCP)[45], Microsoft Azure (MA)[46] .

When it comes to managing cloud resources over a complex infrastructure, good tools are extremely helpful. Such tools are for example for provisioning, deployment and monitoring of resources. Also deployment templates and central access control are very comfortable for administrating cluster environments. AWS and MA have more services to offer as it can be seen in Table (3.6).
It can be said that Amazon Web Services and Microsoft Azure offers the most major cloud service. Microsoft has an edge especially when it comes to integration into an existing IT landscape, where many companies already use Microsoft services.

[40] Azure, 2019.

Type of Management Service	AWS	GCP	AM
Server management	AWS Systems Manager	NA	Azure Operational Insight
Deployment templates	AWS CloudFormation	Resource Manager	Azure Resource Manager
Monitoring	Amazon CloudWatch	Google StackDriver	Azure Monitor/Log Analytics/Application Insight
Automation	AWS OpsWorks/	Compute Engine	Azure Automation

Table 3.6: Management service comparison of Amazon Web Services (AWS)[47], Google Cloud Platform (GCP)[48], Microsoft Azure (MA)[49] .

3.4 | Realisation of Layer two: Predictive data analytic toolbox

For building and training the models[50] the approach as shown in Figure (3.10) was used. First the columns needed for the input features have been selected and then the data set was split into test and training data. In the next step a design of a model has been chosen and trained by the train data. To test the training accuracy the test data used on the trained model and a confusion matrix has been created in order to show the accuracy of the chosen ML or DL model. In the following subsections a RF Section (2.7.1.1) will be presented. This ensemble learning technique is ideally suited for classification, and very simple to implement. The next model will be a HMC algorithm, subsection (2.9.0.2), which is more complex and allows far richer information, because a whole distribution is delivered. The last model is a NN, subsection (2.8), where a multi-layer perceptron classifier with three layers and 30 nodes in total was used. The data set used is a production data set from a research project in the area of Industry 4.0.

3.4.1 | Random Forest

For the random forest approach the scikit-learn library was chosen. It offers many tools for easy application of ML. In this example, Listing (1), after importing all necessary libraries the train/test split is manually implemented. This is done to show in a few lines how such an approach works. 75% of the data set have been chosen for training, and the rest for testing. To avoid bias, the selection was done by a random function. The algorithm presented here has been designed with the help of the following document[51].

In Listing (3) the process of feature engineering is shown in brief. First, the `FanOn` column is dropped and then the training data set is factorized. So all objects are encoded as enumerated type or as a categorical variable.

Now the random forest classifier is generated, as seen in Listing (3). Scikit-learn uses per default bootstrapping and the Gini function, Equation (2.25), is used for measuring the quality of the split.

[50] All code developed and the data set used in this chapter can be found on Zenodo (Kohl, 2019). The code it self has been run on the Little Big Data Cluster of the TU.it of the TU Wien*dataLab - Little Big Data Cluster* 2019.

[51] Navlani, 2018.

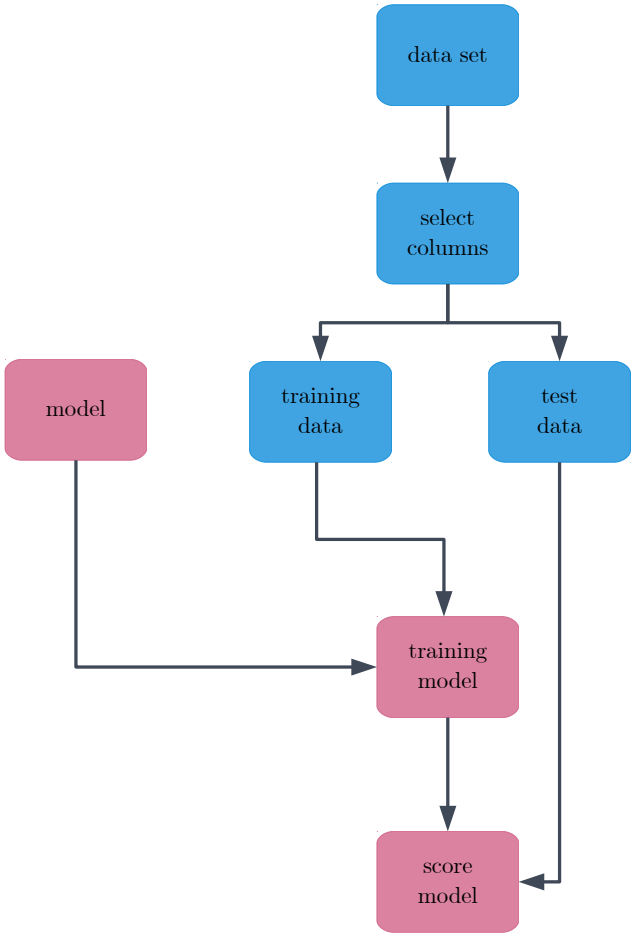


Figure 3.10: Flow chart of the creation of the machine learing models

Scikit offers a confusion matrix function, but as in case of the test split it is here briefly shown how to implement a confusion matrix with pandas, Listing (4). In the output, Listing (5), it can be seen that only 18 out of 1363 have been categorized as false negative. This means that roughly 99% of the test data set has been correctly categorized.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import export_graphviz
from sklearn.ensemble import RandomForestClassifier

data_forest['is_train'] = np.random.uniform(0, 1, len(data_forest))
 <= .75
train, test = data_forest[data_forest['is_train']==True],
              data_forest[data_forest['is_train']==False]
```

Listing 1: Library import and selecting variables for categorization

```
features = data_forest.drop(["FanOn"], axis = 1)
features = features.columns

y = pd.factorize(train['FanOn'])[0]
```

Listing 2: Feature engineering

```
clf = RandomForestClassifier(n_jobs=2, random_state=0)
clf.fit(train[features], y)
```

Listing 3: Building the random forest model

```
preds = clf.predict(test[features])
pd.crosstab(test['FanOn'], preds, rownames=['True'],
colnames=['Predicted'])
```

Listing 4: Generating the confusion matrix for the test data set of the random forest model

```
[[ 586    0]
[  18 1363]]
```

Listing 5: Output of the confusion matrix for the test data set of the random forest model

3.4.2 | Applying Bayesian Networks

With the theoretical background established in Section (2.9.0.2) a detailed explanation of the implementation of the Bayes theorem in PriMa will follow. To be precise a Hamilton Monte-Carlo algorithm has been used, and PyMC3[52] for the implementation. It has been used because it offers an easy to use high level API which allows relative easy training of Bayesian networks. Again the course of action as shown in Figure (3.10) was taken.

It is assumed that the data set is already loaded. The following packages must be imported in order to perform the necessary steps, for training a Bayesian network in Python with PyMC3. The library Theano was used because it offers easy handling of multi-dimensional arrays. Panda offers easy manipulation of numerical tables and scikit-learn-learn is one of the most popular ML/DL libraries for Python. The algorithm presented here has been designed with the help of the following document[53].

After the import the data set was prepared for the testtraining split. In this case scikit-learn offers a function `train_test` just for this. Working with pandas allows the use of regular expressions, so it is relatively easy to label the data. Per default the `train_size` variable of the `train_test_split` is set to 0.25.

```
from sklearn.model_selection import train_test_split
import theano
import theano.tensor as tt
import pandas as pd

X_hmc = data_hmc.iloc[:, [False, True, True, True, False, True,
True, True, True, True, True, True, True, True, True,]]
y_hmc = data_hmc.iloc[:, [False, False, False, False, True, False,
False, False, False, False, False, False, False, False, False]]
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

Listing 6: Library import and test/train split

After splitting the data set values from the data frames must be assign to variables and those will then be transformed into a NumPy array as seen in Listing (7). This is because PyMC3 can't work with pandas data frames, but they are very practicable for nearly all other data manipulation task.

In this step, Listing (8) the model is built. The assumption is that the classes are roughly linearly separable, so a multinomial logistic regression is used. PyMC3 uses symbolic inputs for latent variables, so in order to evaluate an expression knowledge about those variables is needed, because fixed values need to be provided.
The function `sample_node`, as seen in Listing (9), all symbolic dependencies. The code also applies 100 sampled probabilities for each observation.

[52] *PYMC3 - Probabilistic Programming in Python* 2019.
[53] *Variational API quickstart* 2019.

```
        X_train_hmc = X_train_hmc.values
        y_train_hmc = y_train_hmc.values

        converter = []

        for i in y_train_hmc:
            #print(i[0])
            converter.append(i[0])

        y_train_hmc = np.asarray(converter, dtype=np.float64)
```

Listing 7: Converting the data frame

```
    Xt = theano.shared(X_train_hmc)
    yt = theano.shared(y_train_hmc)

    with pm.Model() as hmc_model:
        # Coefficients for features
        b = pm.Normal('b', 0, sd=1e2, shape=(14, 2))
        # Transoform to unit interval
        a = pm.Flat('a', shape=(2,))
        p = tt.nnet.softmax(Xt.dot(b) + a)
        observed = pm.Categorical('obs', p=p, observed=yt)'
```

Listing 8: Building the model

```
    with hmc_model:
        inference = pm.SVGD(n_particles=500, jitter=1)
        approx = inference.approx
        test_probs = approx.sample_node(p,
        more_replacements={Xt: X_test}, size=100)
        train_probs = approx.sample_node(p)
```

Listing 9: Building the model, `sample_node`

Here, in Listing (10) first the symbolic expressions are sampled for accuracy scores and then it is calculated. Those calls are cached in order to reuse them afterwards.

```
#symbolic expressions
test_ok = tt.eq(test_probs.argmax(-1), y_test)
train_ok = tt.eq(train_probs.argmax(-1), y_train)
test_accuracy = test_ok.mean(-1)
train_accuracy = train_ok.mean(-1)

eval_tracker = pm.callbacks.Tracker(
    test_accuracy=test_accuracy.eval,
    train_accuracy=train_accuracy.eval
)

inference.fit(400, obj_optimizer=pm.adamax(learning_rate=0.1),
                callbacks=[eval_tracker]);
```

Listing 10: Training the network

In Figure (3.11) the trainings progress can be seen.

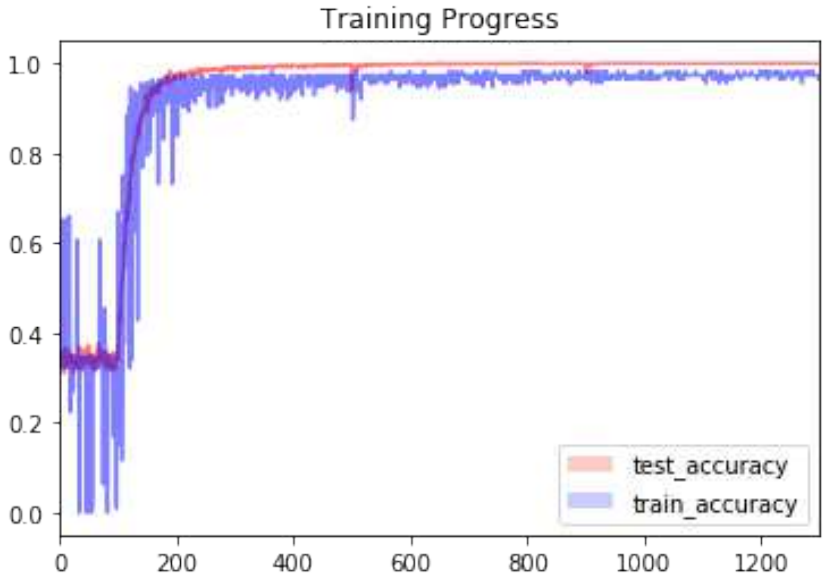


Figure 3.11: Training progress of the Bayesian network

3.4.3 | Neural Networks

For the neural network scikit-learn[54] has been used. It is a software library for machine learning and was originally developed as a Google Summer of Code project. The algorithm presented here has been designed with the help of the following document[55]. The NN developed here has the sole goal of classification. The first step is always the import of the libraries needed. It is assumed that the data set has been previously been loaded. In the following step the `X_nn` values used for categorization are selected.

[54] *scikit-learn* 2019.
[55] Robinson, 2019.

```
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier


X_nn = data_nn.iloc[:, [False, True, True, True, False, True,
True, True, True, True, True, True, True, True, True,]]
y_nn = data_nn.iloc[:, [False, False, False, False, True, False,
False, False, False, False, False, False, False, False,
False]]
```

Listing 11: Library import and selecting variables for categorization

In the next step, as seen in Listing (12) first all unique non numerical values are converted to numerical, orcategorical values. This must not be done in this case, but it is good practice, as well as feature scaling. The features are scaled so they can be uniformly evaluated. This is only done on the training data, because the real data will never be scaled.

```
le = preprocessing.LabelEncoder()
y_nn = y_nn.apply(le.fit_transform)

X_train_nn, X_test_nn, y_train_nn, y_test_nn =
train_test_split(X_nn, y_nn, test_size = 0.20)

scaler = StandardScaler()
scaler.fit(X_train_nn)

X_train_nn = scaler.transform(X_train_nn)
X_test_nn = scaler.transform(X_test_nn)
```

Listing 12: Feature scaling and test/train split

The model building is done as shown in Listing (13). Here a multi-layer perceptron classifier was choosen. The `hidden_layer_sizes` Parameter, is used to set the size of the hidden layers. A three layer architecture with 10 nodes each was chosen and `max_iter` was set to 1000, which is the number of iteration per epoch. One epoch is a combination of one cycle of the feed-forward and back propagation phase. scikit-learn uses per default the `relu` activation function in combination with the `adam` cost optimizer.

```
mlp_nn = MLPClassifier(hidden_layer_sizes=(10, 10, 10),
max_iter=1000)
mlp_nn.fit(X_train_nn, y_train_nn.values.ravel())
```

Listing 13: Building the model

The final stage is testing the model with the test data set, as seen in Listing (14). The evaluation is done by a confusion matrix, Listing (15). It can be seen that 99% of the data has been classified correctly, and only 17 out of 1546 have been classified false negative.

```
predictions_nn = mlp_nn.predict(X_test_nn)

print(confusion_matrix(y_test_nn,predictions_nn))
print(classification_report(y_test_nn,predictions_nn))
```

Listing 14: Evaluating the model

```
[[ 413    0]
[  17 1116]]
             precision    recall  f1-score   support

          0       0.96      1.00      0.98       413
          1       1.00      0.98      0.99      1133

avg / total       0.99      0.99      0.99      1546
```

Listing 15: Output of the confusion matrix for the test data set of the neural network model

3.4.4 | Text Mining

For text mining the framework spaCy[56] has been chosen. What spaCy offers is an industrial-strength natural language processing API which makes tasks like splitting, stop word removal, stemming/lemming and so on very easy. In offers named entity recognition and its pre trained CNN make NLP quite easy. In case of maintenance a special vocabulary would be necessary in order to recognise words the rigth way. For example CNC machine is declared as company, where in the setting of manufacturing computer numerical control machines are meant. The spaCy NLP framework also offers easy integration into scikit-learn and Tensorflow, which is very pracicable for building pipelines.

[56] *spaCy* 2019.

```
import spacy

# Load English tokenizer, tagger, parser, NER and word vectors
nlp = spacy.load("en_core_web_sm")

# Process whole documents
text = ("The lathe machine is leaking oil,
because the sealing is damaged.")
doc = nlp(text)

# Analyze syntax
print("Noun phrases:", [chunk.text for chunk in doc.noun_chunks])
print("Verbs:",
[token.lemma_ for token in doc if token.pos_ == "VERB"])

# Find named entities, phrases and concepts
for entity in doc.ents:
    print(entity.text, entity.label_)
```

Listing 16: Building the NLP model

As it can be seen in Listing (16) that the model building is farley easy. Here the goal has been to use the framework to extract the nous and verbs out of the sentence. The nouns can be correlated to classes in ontologies and the verbs as the data properties. In Listing (17) the result can be seen.

```
Noun phrases: ['The lathe machine', 'oil', 'the sealing']
Verbs: ['be', 'leak', 'be', 'damage']
```

Listing 17: Out put of the NLP model

3.4.5 | Aggregation

In order to combine multiple ML algorithms an aggregation function needs to be chosen. So according to Aggarwal (2016) there are three primary ways of creating hybrid recommender systems:

- **Ensemble design**: Off-the-shelf algorithms are combined into a single and more robust output.
- **Monolithic design**: Various data types are used for an integrated recommendation algorithm.
- **Mixed systems**: Here multiple recommendation algorithms are used as black-boxes and their results are presented side by side.

So in must be asked on how to combine those algorithms if ensemble learning is for example chosen. According to Burke, 2002b seven types of hybrid recommender systems can be distinguished:

- **Weighted**: Scores of several recommender systems are combined into one unified score.
- **Switching**: The algorithm switches, depending on the needs, between different recommender systems.
- **Cascade**: One recommender refines the recommendations for the following recommender.
- **Feature augmentation**: The output of one recommender is the feature input for the next one.
- **Feature combination**: The feature of different data sources are combined for the use of one single recommender system
- **Meta-level**: Here the model used by one system is the input to another. Combinations are typically content-based and collaborative systems
- **Mixed**: Several recommendations are presented to the user at the same time.

AutoPriMa uses a feature combination, because data from different sources is combined to generate one output. In this step at the second layer a weighted approach will be chosen. This is done due to the fact that it is easier to implement. The weights could be heuristic, formal statistical models, or domain knowledge. In this example ale weights α_i are equal. So the prediction $\hat{R}$ is calculated by summarizing the individual recommendations from the different algorithms $\hat{R}_i$ times the individual weight α_i.

$$\sum_{n=1}^{q} \alpha_i * \hat{R}_i \tag{3.1}$$

```
rec = 1/3*pred_rf + 1/3*pred_nn + 1/3*pred_hmc
```

Listing 18: Weighted approach

3.5 | Realisation of Layer three: Recommender and decision-support dashboard

In the third layer of the PriMa architecure, as seen in Figure (3.1) not only the results of the recommendation from the layer two are presented to the user, but also possible solutions based on the knowledge base. In Figure (3.12) a mock-up of a dashboard solution can be seen. While designing this solution especially the data-ink ratio and the principles developed in Section (2.11.0.3) have been kept in mind. A dashboard must allow the user to quickly capture complex data. In the case of PriMa it must also be able to scope with displaying recommendations which come from the knowledge base. Those recommendations are generated by using CBR. CBR has, as explained in Subsection (2.6.3), four different stages.
To explain the functionality of the AutoPriMa dashboard, the five different sections in figure (3.1) are numbered.

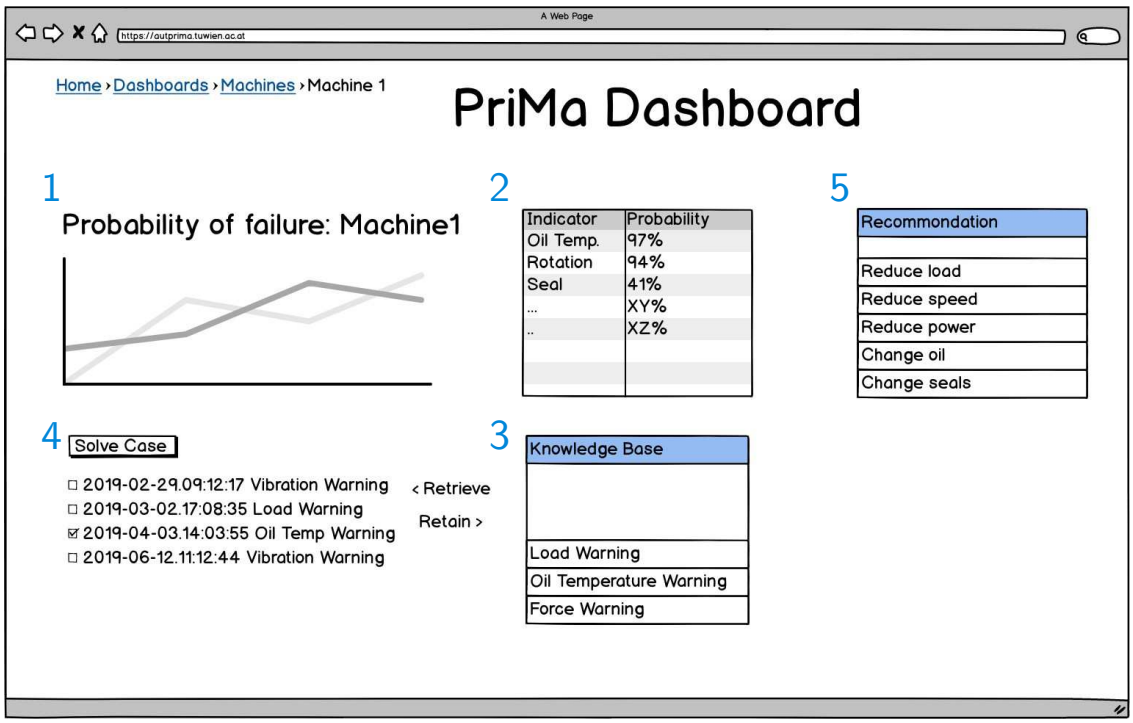


Figure 3.12: Mock-up of the PriMa dashboard

- Section 1: The probability generated by the aggregation function, Listing (18). This shows the maintenance engineer the current trend and gives a feeling if the plant, or machine in this example is heading in the right direction. The goal of this graph is only to inform and give a long term view. As seen in the previous chapter (3.4.5) no prediction is implemented, but this dashboard is designed in a way that prediction can be easily implemented.

- Section 2: Lists the indicators which lead to the probability of failure. One way would be to extract them from the RF model. This is done to give the maintenance engineer some understanding of the facts which lead to AutoPriMa's assessment. Such information is important for a better acceptance.

- Section 3: The knowledge base offers past solved cases based on the analysis of the recommendation engine and the maintenance reports. This is also the first step where the CBR cycle comes into play. When selecting a case it can be retrieved. When retrieving a solution of a past problem the operator gets detailed instructions on what to do in section 5 recommendation.

- Section 4: When a certain situation was not part of the knowledge base the case will be saved but not as a solution. So when the operator finds a solution the case is selected and retained. When retaining a case a solution must be added and will be added to the knowledge base. The algorithm which decides which cases are temporarily saved is not part of this thesis.

- Section 5: Offers recommendations based on the result of the CBR algorithm. An

addition not shown in this mock-up would be the possibility of a pop-up which gives more detailed explanations on how to achieve the suggested tasks.

3.6 | Realisation of Layer four: Knowledge Pipeline

The knowledge pipeline enables PriMa to combine predictions from the traditional machine learning models with the knowledge from experts. This knowledge pipeline uses ontologies, as shown in chapter (2.12.1.3) for storing information about the machines and its used parts, as well as the relations between the machines and their parts. Those relations are in this case information about the malfunction. The information about those errors come from maintenance reports.

Those reports are fed into the NLP engine, as seen in chapter (3.4.5). This algorithm transforms each report into nouns and verbs. The nouns are the machines or parts and the verbs are their relation. For example *The lathe machine leaks oil.* There is the classes *lathe machine* and *oil*, and the object property *leaks*. So information on how to solve the leaking lathe machine must be stored in either the class *oil* or in the object property *leaks*.

Not only oil can leak, but also cooling liquid and so the solution to both problems is quite similar. It can be stored in the object property *leaks*. It is important to notice that when the ontology becomes more and more complex such simple statements may not be true. When a report reads *The bearing of the lathe machine vibrates* the solution is eventually to fixate the bearing. But also an engine vibrates and an abnormally vibration engine must be repaired in a different way. In ontologies thresholds for object properties can be defined. This means for example if the cooling fluid gets hotter than 75°C and the ontologie is queried a numeric value must be delivered in order to make the connection. So while designing a complex ontology such thoughts must be kept in mind in order to build a system which enables growth and satisfied all requirements of the production plant.

For the knowledge base an ontology was with Protègè[57]. As seen in Figure on (3.13). It can be seen that each machine has a connection to some parts. Each red dotted line is a object property like *leaks*.

CBR normally works with similarity measures as seen with numerical values in the code provided[58]. In this case SPARQL[59] has been used to build the CBR cycle. This has been done by various researchers, like Bach et al., 2012 for MyCBR3 and others[60][61][62]. A SPARQL queries are always compromise of the followings[63]:

- Prefix declarations,for URIs
- Data set definition

[57] *Protégé* 2019.
[58] Kohl, 2019.
[59] Eric Prud'hommeaux, n.d.
[60] Kushwaha et al., 2013.
[61] Bruneau et al., 2017.
[62] Gaillard et al., 2014.
[63] Feigenbaum, 2009.

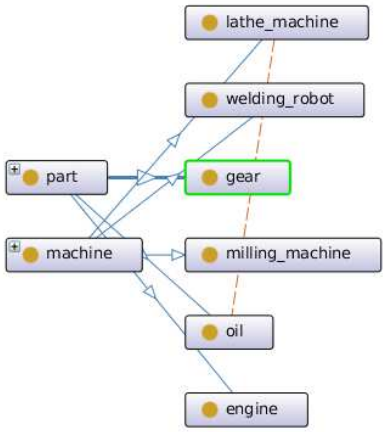


Figure 3.13: The used ontology

- A result clause
- The query pattern
- Query modifiers

A SPARQL query must always consist of triplets like *?person foaf:name ?name*[64]. According to the CBR cycle, Figure (2.13), the first step is the case retrieval. In this step the reasoning algorithm looks for a similar case. This can be done by an *ASK*, as seen in listing (19), query in SPARQL. This query returns *TRUE* when the case exists or false *FALSE* if no similar case exists.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prima: <http://www.semanticweb.org/linus/ontologies/
2019/5/untitled-ontology-11#>
ASK { ?part prima:partOf ?machine }
```

Listing 19: ASK query

When the case exists it must be retrieved in order to be reused. This is done by the *SELECT* statement. In *SELECT* statement only the variables used in the tripled of the *WHERE* clause can be selected. So the *SELECT* statement as seen in listing (20), queries for all classes which are connected by a certain object property and have a *prima:solution*. In the *prima:solution* a solution for the case can be found. As explained above the classes come from the NLP model. SPARQL can only use

[64] Feigenbaum, 2009.

uniform resource identifiers (URI)[65]. This means, if Python is used, the list from the NLP model must be run against a dictionary in order to retrieve the URI for lathe machine.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prima: <http://www.semanticweb.org/linus/ontologies/
2019/5/untitled-ontology-11#>
SELECT ?machine ?part ?solution
     WHERE { ?part prima:partOf ?machine .
         OPTIONAL {?part prima:solution ?solution.} }
```

Listing 20: SELECT query

When the *ASK* statement returns *FALSE* the maintenance officer suggest a solution and with this information the new case will be retained to the knowledge base. As seen in listing (21) a *WHERE* clause is here needed too. This one is needed in order to check if the object property and necessary connections between the classes exist. In this case again URIs have to be used in order to create a concrete connection between to classes with a certain object property.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prima: <http://www.semanticweb.org/linus/ontologies/
2019/5/untitled-ontology-11#>
CONSTRUCT {?part prima:partOf ?machine }
    WHERE { ?part prima:partOf ?machine .
    OPTIONAL {?part prima:solution ?solution.} }
```

Listing 21: CONSTRUCT query

This shows that with the help of SPARQL a whole CBR cycle can be recreated. All listings above must be packed in *if* statements and a dictionary, if Python is used, must be created in order to query for specific cases. All queries presented here only give generic answers. Certain prefixes can also be added to the custom ontology, so the queries would not need to look for *pima:solution*, but could use *prima:solution* instead. This would also create the possibility of integrating the probabilities from the aggregated ML algorithms. Those information could be stored in *prima:probability*. So when querying cases are only retrieved if the probability is high enough. It is also important to notice that cases which do not exist in the case base, must be stored in a temporary data store. Only when the maintenance engineer finds a suitable solution than a new connection can be made and the solution updated.

[65] *http://www.ontologydesignpatterns.org/ont/dul/DUL.owl/unique*

4 | Conclusion

This thesis established a knowledge pipeline which shows us that unstructured data, text, from maintenance reports can be used for solving the problems mentioned in the report. This was done by applying CBR with SPARQL on an ontology. Beforehand text mining was applied to extract certain words from the maintenance report. Not only known problems can be solved, but also new problems and their solutions can be added to the knowledge base. So a lerning system is generated which is able to grow over time. This gives AutoPriMa an edge over other architectures[1][2][3].

4.0.0.1 Keyfinding 1

In the introduction a sub question was raised: "How to stream continuous data from various sources into a data warehouse solution of PriMa¿' This leads to the point of which ingestion platform to use. The main selling point for RabbitMQ is definitely its broader support of protocols, especially MQTT, which is widely used in the industrial context. Kafka does not directly support MQTT, but there are some open-source projects which enable MQTT support for Kafka. So if the data volume is so high that RabbitMQ cannot handle it Kafka can be an option. To avoid problems with outliers a combined use of, RabbitMQ and Kafka is a very elegant way to scope with potential sensor malfunctions. This is also important when discussing the data store used. The architectural details explained in Ansari, Glawar, et al. (2019) can be fulfilled by all inspected providers. When looking at the criteria developed in chapter (3.3.2)) Azure is the most suitable solution. This is mostly due to the good legacy integration because of the tight integration of other Microsoft products. The security aspect with its advanced backup and disaster recovery solutions included in the Azure package makes Microsoft Azure the preferred platform. Those services are extremely valuable for companies. PriMa will be most likely deployed to highly advanced factories with a high degree of automatization. A failure of the storage system would be catastrophic in such a scenario.

4.0.0.2 Keyfinding 2

The second research question raised in this thesis is "How to aggregate machine learning algorithm and how to validate their outcome? (i.e. From Data to Knowledge Intelligence)¿' Therefore three algorithms have been chosen to achieve these goals. The first one is a RF, which performs very well with it's classification task on the used data set. The confusion table shows a 99% accuracy. The NN was also implemented with scitkit-learn and as seen in Listing (15) the confusion matrix shows only 17 false negative detections and therefore a 99% accuracy. The HMC algorithm was far more complex to implement and training the model takes a significant amount of time. The achieved results are, again as with the RF and the NN, very good. In case of the HMC model the training progress can be seen

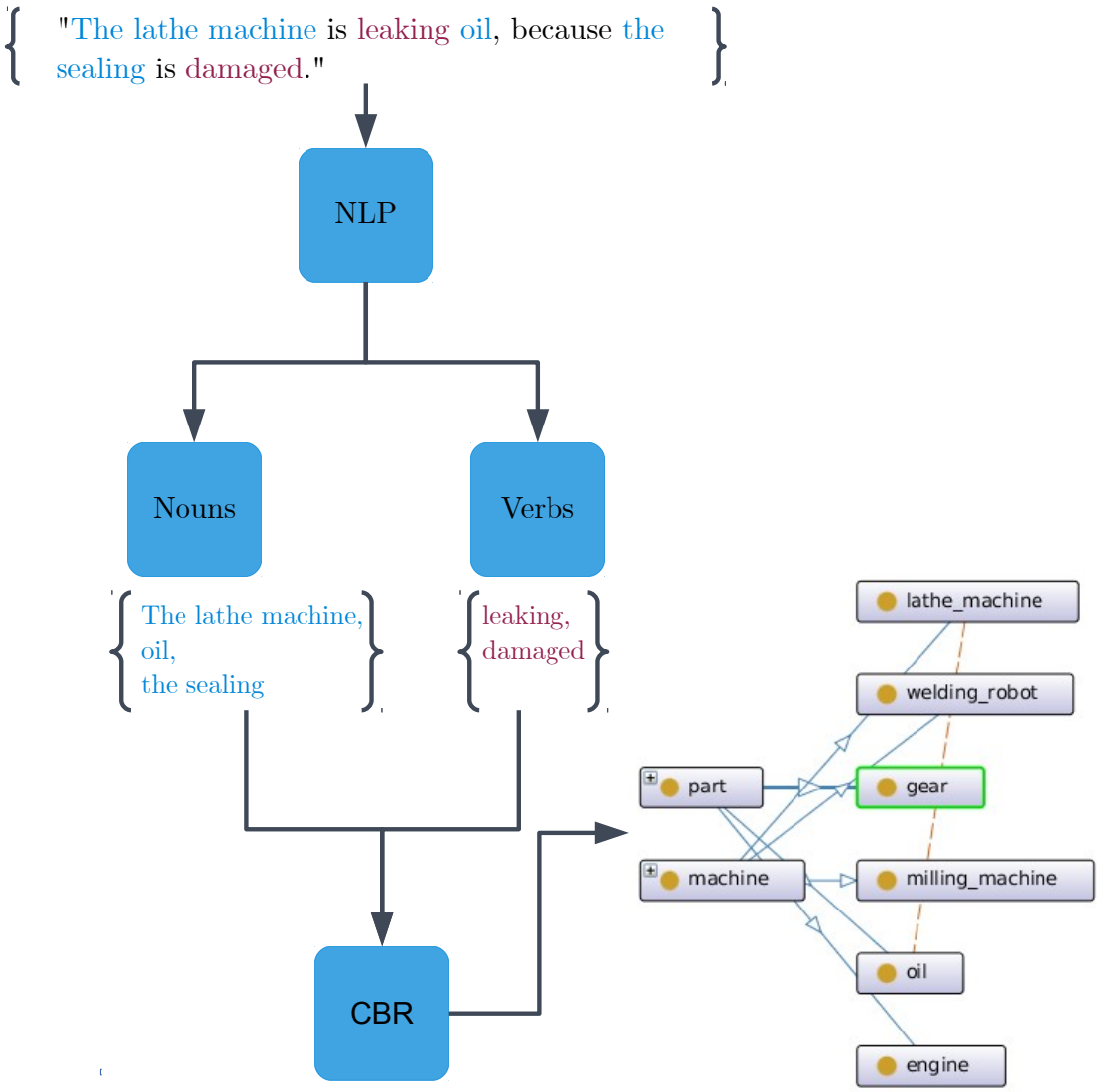
[1] Cheng et al., 2018.
[2] O'Donovan et al., 2015.
[3] R. Ranjan, 2014.

in Figure (3.11). The aggregation of all three models was done with a simple weighted hybrid model. The weights have not been adjusted and each algorithm is contributing a third to the end result. It can be said that the stated research question was answered in full and the results are satisfactory. It shows that three algorithm can be applied to a problem and a single result forged out of those inputs.

4.0.0.3 Keyfinding 3

The main research question of this thesis was "How to build a cutting-edge knowledge pipeline for prescriptive maintenance based on the four layer architecture of PriMa¿' This was done with a combination of NLP and CBR, where CBR uses SPARQL as a reasoning engine. The information flow can be seen in Figure (4.1). A sentence from the maintenance report is split into nouns and verbs, where the nouns correspond to classes and the verbs to object properties. Then CBR, with the help of SPARQL, looks up if a certain combination exists, if this is true, than the solution is retrieved, and if not a new connection is established. This workflow can be implement into existing maintenance workflows, because the routine of creating reports feeds into AutoPriMa. It also leads to shorter training time for new maintenance engineers. The new hires can build upon old, solved cases and therefore it reduces mistakes at the beginning of the training period and shortens their training time. The knowledge which can be built upon is expert knowledge, which allows all maintenance engineers to leverage this knowledge. Therefore it also allows experienced staff, who are not so familiar with a particular problem to build upon this knowledge-base and solve those less frequent problems. This leads to the main strength of the cutting edge knowledge pipeline, its extendability. It is a learning system and does not only show a certain snapshot of time in the sense of knowledge but it extends over time, which each new solution. So AutoPriMa combines expert knowledge and machine-made knowledge to a single, growing solution. This gives AutoPriMa an definite edge over frameworks like O'Donovan et al. (2015), Cheng et al. (2018) and R. Ranjan (2014).

The biggest limitation is that AutoPriMa is not able to scope with long reports and the process of updating the knowledge is not fully defined. This includes how the unsolved case is stored and which workflow is to be followed when solving and consequently storing the new case. The *INSERT* query could be used for updating existing cases, but again a defined workflow needs to be constructed here.



SPARQL:
ASK The lathe machine leaking oil IN PriMa

if ASK == TRUE
    SELECT The lathe machine leaking oil
If ASK == FALSE
    CONSTRUCT The lathe machine leaking oil

Figure 4.1: AutoPriMa's knowledge pipeline

5 | Outlook

The goal of PriMa and therefore AutoPriMa, is to be a virtual assistant system in Smart Factories, consequently different aspects have to be improved. In layer one, data management, security is an important aspect. While a service like Microsoft Azure offers reasonable security, the security of the connected devices was not discussed in this thesis. Although MQTT offers reasonable security, the sending devices must be included in a wholistic security concept. Not only classical security threads are part of such a concepts, but also sensor malfunctions, which could corrupt the whole data base and this would lead to mistrained ML algorithms.

In the layer of the predictive data analytics toolbox the aspect of parallelisation is neither discussed nor implemented. While Apache Spark[1] offers good support for RF, Deep Learning is currently not supported and so it must be combined with Keras[2] or Tensorflow[3]. But there is the Deep Learning Pipeline[4] project from Databricks which tries to bridge this gap so it is only a matter of time before these projects will be added to the official API and enable parallel deep learning with Apache Spark. Parallelisation of Bayesian networks with PyMC3 is more complex and it must be done manually using Python functionality. It should be also mentioned that as seen in the previous chapters, the algorithms just classify a certain state, but make no predictions about future trends. This would be a very important feature and therefore an important aspect to enhancement for future implementation.

In layer four, which includes the knowledge pipeline, the update functionality of SPARQL should be implemented into the knowledge pipeline. At the moment when a new case emerges, or an update of an old case is needed, a new triple is added. An update function would be a considerably more elegant approach to solve this problem. Additionally, all parts of the CBR cycle should be packed into one script which has the NLP model as an input.

PriMa, on the whole can be enhanced by adding additional layers to its architecture. Figure (5.1) also shows in a schematic way, in purple, a possibility to connect PriMa to a production planning system (PPS). Here for example Reinforcement Learning could be used for planning the maintenance activities and a digital twin of the factory would be the potential training ground. Also Blockchain technology could be used to make maintenance contracts for different machines, which are potentially owned by another company. The PPS would also be able to schedule the workload to other similar machines.

[1] *Apache Spark* 2019.
[2] *Keras* 2016.
[3] *Tensorflow* 2019.
[4] *Deep Learning Pipelines* 2019.

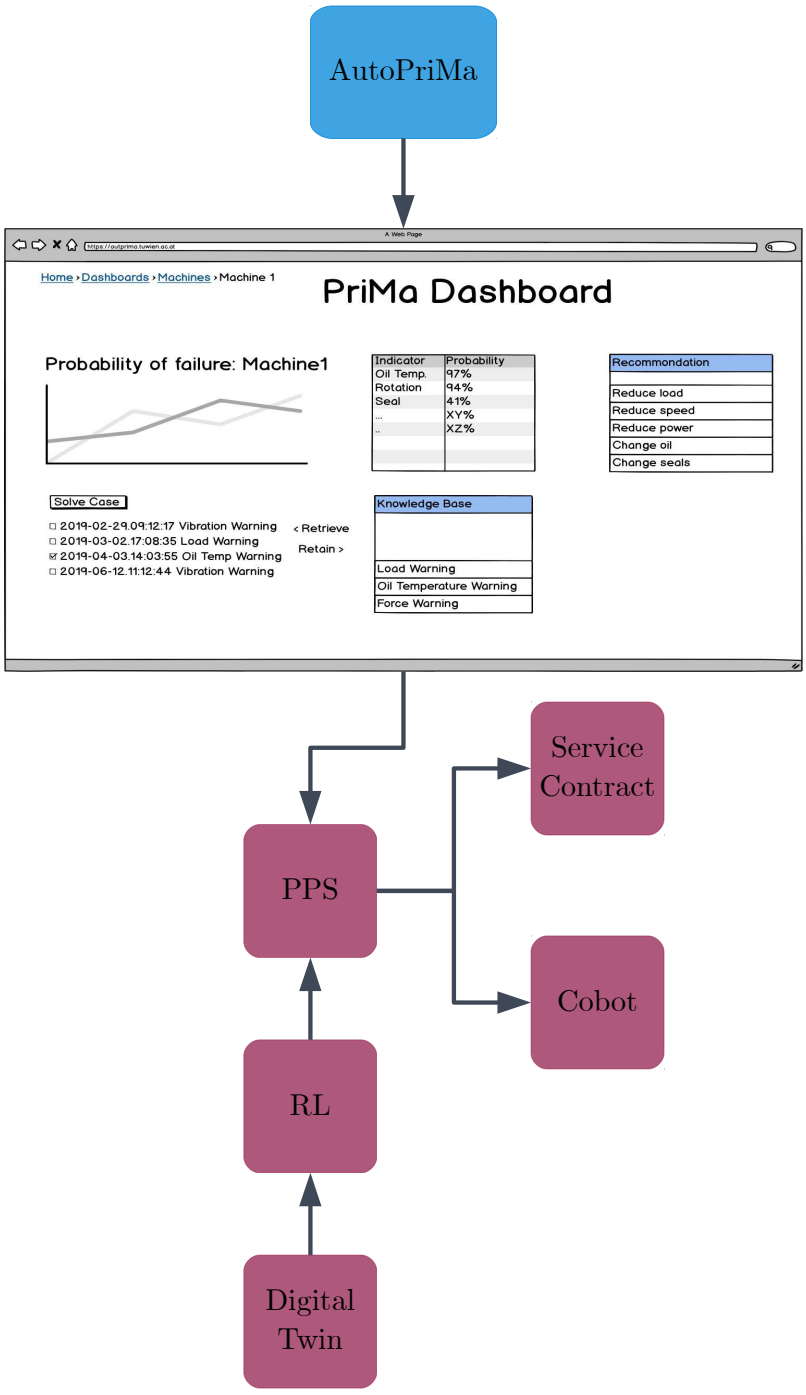


Figure 5.1: Possible extensions for AutoPriMa

List of Figures

List of source codes

List of Tables

[5] O'Donovan et al., 2015.
[6] Cheng et al., 2018.
[7] R. Ranjan, 2014.
[8] Ansari, Glawar, et al., 2019.
[9] AWS, 2019.
[10] Platfrom, 2019.
[11] Azure, 2019.
[12] AWS, 2019.
[13] Platfrom, 2019.
[14] Azure, 2019.
[15] AWS, 2019.
[16] Platfrom, 2019.
[17] Azure, 2019.

Bibliography

Aamodt, Agnar and Enric Plaza (1994). "Case-based reasoning: Foundational issues, methodological variations, and system approaches". In: *AI communications* 7.1, pp. 39–59.

Aggarwal, Charu C (2016). "Knowledge-based recommender systems". In: *Recommender systems*. Springer, pp. 167–197.

Analytics, IoT (Mar. 2017). *Predictive Maintenance Market Report 201722*. Research rep. Market Report. Hamburg, Germany.

Ansari, Fazel, Robert Glawar, and Tanja Nemeth (2019). "PriMa: a prescriptive maintenance model for cyber-physical production systems". In: *International Journal of Computer Integrated Manufacturing*, pp. 1–22.

Ansari, Fazel, Philipp Hold, et al. (Oct. 2018). "AUTODIDACT: Introducing the Concept of Mutual Learning into a Smart Factory Industry 4.0". In:

Ansari, Fazel, Marjan Khobreh, et al. (2018). "A problem-solving ontology for human-centered cyber physical production systems". In: *CIRP Journal of Manufacturing Science and Technology* 22, pp. 91–106.

Ansari, Fazel, Patrick Uhr, and Madjid Fathi (2014). "Textual meta-analysis of maintenance managements knowledge assets". In: *International Journal of Services, Economics and Management* 6.1, pp. 14–37.

Apache Spark (June 30, 2019). URL: https://spark.apache.org/.

Aronson, Jay E, Ting-Peng Liang, and Efraim Turban (2005). *Decision support systems and intelligent systems*. Vol. 4. Pearson Prentice-Hall.

AWS, Amazon (May 14, 2019). *Pub/Sub Messaging - Asynchronous event notifications*. URL: https://aws.amazon.com/pub-sub-messaging/.

Azure, Microsoft (June 17, 2019). *Microsoft Azure*. URL: https://azure.microsoft.com/en-us/.

Babu, Suresh C., Shailendra N. Gajanan, and J. Arne Hallam (2017). "Chapter 16 - Designing a Decentralized Food System to Meet Nutrition Needs: An Optimization Approach". In: *Nutrition Economics*. Ed. by Suresh C. Babu, Shailendra N. Gajanan, and J. Arne Hallam. San Diego: Academic Press, pp. 327–342. ISBN: 978-0-12-800878-2. DOI: https://doi.org/10.1016/B978-0-12-800878-2.00016-5. URL: http://www.sciencedirect.com/science/article/pii/B9780128008782000165.

Bach, Kerstin and Klaus-Dieter Althoff (2012). "Developing case-based reasoning applications using mycbr 3". In: *International Conference on Case-Based Reasoning*. Springer, pp. 17–31.

Bauernhansl, Thomas (2014). "Die Vierte Industrielle Revolution – Der Weg in ein wertschaffendes Produktionsparadigma". In: *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Springer Fachmedien Wiesbaden, pp. 5–35. DOI: 10.1007/978-3-658-04682-8_1. URL: https://doi.org/10.1007/978-3-658-04682-8_1.

Ben-Daya, Mohamed (2009). *Handbook of maintenance management and engineering*. Dordrecht [u.a.]: Springer. ISBN: 1848824718.

Bengtsson, Marcus (Jan. 2011). "Classification of Machine Equipment". In:

Benkaddour, Fatima Zohra, Noria Taghezout, and Bouabdellah Ascar (2016). "Novel Agent Based-approach for Industrial Diagnosis: A Combined Use between Case-based Reasoning and Similarity Measures." In: *International Journal of Interactive Multimedia & Artificial Intelligence* 4.2.

Betti, Alessandro et al. (2019). "Predictive Maintenance in Photovoltaic Plants with a Big Data Approach". In: *arXiv preprint arXiv:1901.10855*.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0387310738.

BMBF (Aug. 2017). *Industrie 4.0 Innovationen für die Produktion von morgen*. Tech. rep. Bundesministerium für Bildung und Forschung.

Bonczek, Robert H, Clyde W Holsapple, and Andrew B Whinston (1980). "The evolving roles of models in decision support systems". In: *Decision Sciences* 11.2, pp. 337–356.

Bruneau, Olivier et al. (2017). "A SPARQL Query Transformation Rule Language — Application to Retrieval and Adaptation in Case-Based Reasoning". In: *Case-Based Reasoning Research and Development*. Ed. by David W. Aha and Jean Lieber. Cham: Springer International Publishing, pp. 76–91. ISBN: 978-3-319-61030-6.

Bumblauskas, Daniel et al. (2017). "Smart Maintenance Decision Support Systems (SMDSS) based on corporate big data analytics". In: *Expert Systems with Applications* 90, pp. 303–317. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.08.025. URL: http://www.sciencedirect.com/science/article/pii/S095741741730564X.

Burke, Robin (2002a). "Hybrid recommender systems: Survey and experiments". In: *User modeling and user-adapted interaction* 12.4, pp. 331–370.

— (2002b). "Hybrid recommender systems: Survey and experiments". In: *User modeling and user-adapted interaction* 12.4, pp. 331–370.

Burkhard, Hans-Dieter (2004). "Case completion and similarity in case-based reasoning". In: *Computer Science and Information Systems* 1.2, pp. 27–55.

Burkhard, Hans-Dieter and Michael M Richter (2001). "On the notion of similarity in case based reasoning and fuzzy theory". In: *Soft computing in case based reasoning*. Springer, pp. 29–45.

Chan, Felix TS, Bing Jiang, and Nelson KH Tang (2000). "The development of intelligent decision support tools to aid the design of flexible manufacturing systems". In: *International journal of production economics* 65.1, pp. 73–84.

Chapman, Pete and Clinton (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. Tech. rep. The CRISP-DM consortium. URL: http://www.crisp-dm.org/CRISPWP-0800.pdf.

Cheng, Bo et al. (2018). "Industrial cyberphysical systems: Realizing cloud-based big data infrastructures". In: *IEEE Industrial Electronics Magazine* 12.1, pp. 25–35.

Coalition, Smart Manufacturing Leadership (2011). "Implementing 21st century smart manufacturing". In: *Workshop summary report*.

Corrigan, David (Oct. 22, 2013). *Building Confidence in Big Data - IBM Smarter Business 2013*. URL: https://www.slideshare.net/ibmsverige/building-confidence-in-big-data.

Cukier, K. (2010). *Data, Data Everywhere: A Special Report on Managing Information*. The economist. Economist Newspaper. URL: https://books.google.at/books?id=jfpSmwEACAAJ.

dataLab - Little Big Data Cluster (July 2, 2019). URL: https://www.it.tuwien.ac.at/hpc/data_lab/.

Dauber, Luanne (July 18, 2019). *The 2017 Apache Kafka Survey: Streaming Data on the Rise*. URL: https://www.confluent.io/blog/2017-apache-kafka-survey-streaming-data-on-the-rise/.

De Mantaras, Ramon Lopez et al. (2005). "Retrieval, reuse, revision and retention in case-based reasoning". In: *The Knowledge Engineering Review* 20.3, pp. 215–240.

Dedi, Nedim and Clare Stanier (2016). "An evaluation of the challenges of multilingualism in data warehouse development". In:

Deep Learning Pipelines (June 30, 2019). URL: `https://docs.databricks.com/applications/deep-learning/single-node-training/deep-learning-pipelines.html`.

Dengel, Andreas (2012). *Semantische Technologien: Grundlagen Konzepte Anwendungen*. ger. Heidelberg: Spektrum Akademischer Verlag. ISBN: 9783827426635.

Dobbelaere, Philippe and Kyumars Sheykh Esmaili (2017). "Kafka versus rabbitmq". In: *arXiv preprint arXiv:1709.00333*.

Drath, Rainer and Alexander Horch (June 2014). "Industrie 4.0: Hit or Hype? [Industry Forum]". In: *Industrial Electronics Magazine, IEEE* 8, pp. 56–58. DOI: `10.1109/MIE.2014.2312079`.

Ehrlinger, Lisa and Wolfram WöSS (2016). "Towards a Definition of Knowledge Graphs." In: *SEMANTiCS (Posters, Demos, SuCCESS)* 48.

Engel, G., T. Greiner, and S. Seifert (June 2018). "Ontology-Assisted Engineering of CyberPhysical Production Systems in the Field of Process Technology". In: *IEEE Transactions on Industrial Informatics* 14.6, pp. 2792–2802. ISSN: 1551-3203. DOI: `10.1109/TII.2018.2805320`.

Eric Prud'hommeaux, Andy Seabor (n.d.). *SPARQL Query Language for RDF*. URL: `https://www.w3.org/TR/rdf-sparql-query/`.

Eugster, Patrick Th et al. (2003). "The many faces of publish/subscribe". In: *ACM computing surveys (CSUR)* 35.2, pp. 114–131.

Färber, Michael et al. (2018). "Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago". In: *Semantic Web* 9.1, pp. 77–129.

Farinha, J.M.T. (2018). *Asset Maintenance Engineering Methodologies*. CRC Press. ISBN: 9781351869324. URL: `https://books.google.at/books?id=NERnDwAAQBAJ`.

Feigenbaum, Lee (June 9, 2009). *SPARQL by example*. URL: `https://www.w3.org/2009/Talks/0615-qbe/`.

Feldman, R., J. Sanger, and Cambridge University Press (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press. ISBN: 9780521836579. URL: `https://books.google.at/books?id=U3EA%5C_zX3ZwEC`.

Ferguson, Stuart and Rodney Hebels (2003). "CHAPTER 5 - Generic data management software". In: *Computers for Librarians (Third Edition)*. Ed. by Stuart Ferguson and Rodney Hebels. Third Edition. Topics in Australasian Library and Information Studies. Chandos Publishing, pp. 143–166. ISBN: 978-1-876938-60-4. DOI: `https://doi.org/10.1016/B978-1-876938-60-4.50011-2`. URL: `http://www.sciencedirect.com/science/article/pii/B9781876938604500112`.

Foster, Darren, Carolyn McGregor, and Samir El-Masri (2005). "A survey of agent-based intelligent decision support systems to support clinical management and research". In: *proceedings of the 2nd international workshop on multi-agent systems for medicine, computational biology, and bioinformatics*, pp. 16–34.

Foundation, Apache Software (2019). *Apache Kafka*. URL: `http://kafka.apache.org/`.

Gabel, Thomas (Nov. 5, 2010). *Problem Solving by Case-Based Reasoning PART 1*. `http://www2.informatik.uni-freiburg.de/~ki/teaching/ss10/gki/ai41-2x2.pdf` online accessed: 2018-12-15. URL: `http://www2.informatik.uni-freiburg.de/~ki/teaching/ss10/gki/ai41-2x2.pdf` (visited on 12/15/2018).

Gaillard, Emmanuelle et al. (2014). "Tuuurbine: A Generic CBR Engine over RDFS". In: *Case-Based Reasoning Research and Development*. Ed. by Luc Lamontagne and Enric Plaza. Cham: Springer International Publishing, pp. 140–154. ISBN: 978-3-319-11209-1.

Gartner (2019a). *Gartner*. URL: https://www.gartner.com/it-glossary/data-lake (visited on 05/14/2019).

— (July 11, 2019b). *Prescriptive Analytics - Gartner IT Glossary*. URL: https://www.gartner.com/it-glossary/prescriptive-analytics/.

Giaretta, Pierdaniele and Nicola Guarino (1995). "Ontologies and knowledge bases towards a terminological clarification". In: *Towards very large knowledge bases: knowledge building & knowledge sharing* 25, p. 32.

Gombé, Bérenger Ossété et al. (2017). "A SAW wireless sensor network platform for industrial predictive maintenance". In: *Journal of Intelligent Manufacturing*. Thanks to Springer Verlag editor. The definitive version is available at : http://www.springer.com/business+%26+management/operations+research/journal/10845 Source : http://www.sherpa.ac.uk/romeo/search.php, pp. 1–12. DOI: 10.1007/s10845-017-1344-0. URL: http://oatao.univ-toulouse.fr/18693/.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Gruber, Thomas R (1992). *Ontolingua: A mechanism to support portable ontologies*.

Gubbi, Jayavardhana et al. (2013). "Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future generation computer systems* 29.7, pp. 1645–1660.

GUPTA, G.K. (2014). *INTRODUCTION TO DATA MINING WITH CASE STUDIES*. PHI Learning. ISBN: 9788120350021. URL: %5Curl%7Bhttps://books.google.at/books?id=fzB9BAAAQBAJ%7D.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York. ISBN: 9780387848587. URL: %5Curl%7Bhttps://books.google.at/books?id=tVIjmNS3Ob8C%7D.

Heng, Stefan (2014). "Industry 4.0: Huge potential for value creation waiting to be tapped". In: *Deutsche Bank Research*.

Henning Kagermann Wolfgang Wahlster, Johannes Helbig (Apr. 2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0*. Tech. rep. National academy of science and engineering.

Hiemstra, Djoerd and Franciska MG de Jong (2001). "Statistical Language Models and Information Retrieval: natural language processing really meets retrieval". In: *Glot international* 5.8, pp. 288–293.

Hirsch-Kreinsen, Hartmut (Jan. 2016). *Industry 4.0" as Promising Technology: Emergence, Semantics and Ambivalent Character*.

Hossain, Saddam (2013). "5G wireless communication systems". In: *American Journal of Engineering Research (AJER)* 2.10, pp. 344–353.

Hotho, Andreas, Andreas Nürnberger, and Gerhard PaaSS (2005). "A brief survey of text mining." In: *Ldv Forum*. Vol. 20. 1. Citeseer, pp. 19–62.

IBM (May 12, 2019). *The 4 Vs of Big Data*. URL: https://www.ibmbigdatahub.com/tag/587.

Inmon, W.H. (2005). *Building the Data Warehouse*. Timely, practical, reliable. Wiley. ISBN: 9780764599446. URL: https://books.google.at/books?id=rnG3vjy7iPoC.

Jagadish, H.V. (2015). "Big Data and Science: Myths and Reality". In: *Big Data Research* 2.2. Visions on Big Data, pp. 49–52. ISSN: 2214-5796. DOI: https://doi.org/10.1016/j.bdr.2015.01.005. URL: http://www.sciencedirect.com/science/article/pii/S2214579615000064.

James, Gareth et al. (2014). *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.

Jovanovic, Vladan, Danijela Subotic, and Stevan Mrdalj (2014). "Data modeling styles in data warehousing". In: *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, pp. 1458–1463.

K., Matyas (2015). "Instandhaltungs- und Zuverlässigkeitsmanagement". In:

Keras (June 30, 2016). URL: https://keras.io/.

Kiesling, Elmar (2018). *Business Intelligence - Data Warehousing (Part 1)*.

Kimball, R. and M. Ross (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley. ISBN: 9781118732281. URL: https://books.google.at/books?id=4rFXzk8wAB8C.

Klahold, Andre et al. (2013). "Using word association to detect multitopic structures in text documents". In: *IEEE Intelligent Systems* 29.5, pp. 40–46.

Kodratoff, Yves (1999). "Knowledge discovery in texts: a definition, and applications". In: *International Symposium on Methodologies for Intelligent Systems*. Springer, pp. 16–29.

Kohl, Linus (June 2019). *Auto PriMA*. DOI: 10.5281/zenodo.3249987. URL: https://doi.org/10.5281/zenodo.3249987.

Kolodner, Janet (2014). *Case-based reasoning*. Morgan Kaufmann.

Kroes, Neelie (2011). "Data is the new gold". In: URL: http://europa.eu/rapid/press-release_SPEECH-11-872_en.htm?locale=en (visited on 05/11/2019).

Kroetsch, M and G Weikum (2015). "Special Issue on Knowledge Graphs". In: *Journal of Web Semantics. Zugriff am* 4, p. 2018.

Kushwaha, Nidhi, Rajesh Mahule, and Om Prakash Vyas (2013). "Utilizing DBpedia via CBR Approach of Recommender System". In:

Leake, David B (1996). *Case-Based Reasoning: Experiences, lessons and future directions*. MIT press.

Lee, E.A. and S.A. Seshia (2017). *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. The MIT Press. MIT Press. ISBN: 9780262340526. URL: https://books.google.at/books?id=3BHiDQAAQBAJ.

Lee, Jay, Behrad Bagheri, and Hung-An Kao (2014). "Recent advances and trends of cyber-physical systems and big data analytics in industrial informatics". In: *International proceeding of int conference on industrial informatics (INDIN)*, pp. 1–6.

Lee, Jay, Edzel Lapira, et al. (2013). "Recent advances and trends in predictive manufacturing systems in big data environment". In: *Manufacturing letters* 1.1, pp. 38–41.

Legat, Christoph et al. (2014). "Semantics to the Shop Floor: Towards Ontology Modularization and Reuse in the Automation Domain". In: *IFAC Proceedings Volumes* 47.3. 19th IFAC World Congress, pp. 3444–3449. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20140824-6-ZA-1003.02512. URL: http://www.sciencedirect.com/science/article/pii/S1474667016421385.

Lenz, Mario et al. (2003). *Case-based reasoning technology: from foundations to applications*. Vol. 1400. Springer.

Lidd, Elizabeth D. (2001). *Natural Language Processing*.

Lin, Ching-Yi et al. (2018). "5S Dashboard Design Principles For Self-service Business Intelligence Tool Users". In: *Journal of Big Data Research* 1.1, p. 5.

Lindstedt, D., K. Graziano, and H. Hultgren (2009). *The Business of Data Vault Modeling*. LuLu Enterprise. ISBN: 9781435719149. URL: https://books.google.at/books?id=KNxdBx2Il7cC.

Liu, Jun S. (1994). "The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem". In: *Journal of the American Statistical*

Association 89.427, pp. 958–966. DOI: 10.1080/01621459.1994.10476829. eprint: \url{https://doi.org/10.1080/01621459.1994.10476829}. URL: %5Curl%7Bhttps://doi.org/10.1080/01621459.1994.10476829%7D.

Lueth K. Patsioura C., Williams Z. and Kermani Z. (Dec. 2016). *INDUSTRIAL ANALYTICS 2016/2017 The current state of data analy tics usage in industrial companies*. Research rep. Market Report. Hamburg, Germany.

Malik, Shadan (2005). *Enterprise Dashboards - Design and Best Practices for IT*. New York: John Wiley & Sons. ISBN: 978-0-471-74193-0.

Manyika, J. et al. (2011). *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. McKinsey. ISBN: 9780983179696. URL: https://books.google.at/books?id=vN1CYAAACAAJ.

Márquez, Adolfo Crespo et al. (2016). "Criticality Analysis for Maintenance Purposes: A Study for Complex In-service Engineering Assets". In: *Quality and reliability engineering international* 2.32, pp. 519–533.

Marr, Bernard (Mar. 5, 2018). "Here's Why Data Is Not The New Oil". In: *Forbes*. URL: https://www.forbes.com/sites/bernardmarr/2018/03/05/heres-why-data-is-not-the-new-oil/#265fe0a73aa9 (visited on 05/11/2019).

Marti, Robert (2012). *Data Warehousing*.

Matyas, Kurt (2018). *Instandhaltungslogistik: Qualität und Produktivität steigern*. Carl Hanser Verlag GmbH Co KG.

Matyas, Kurt et al. (2017). "A procedural approach for realizing prescriptive maintenance planning in manufacturing industries". In: *CIRP Annals* 66.1, pp. 461–464.

Mobley, R. Keith (2014). *Maintenance engineering handbook*. eng. Eighth edition. New York: McGraw-Hill education. ISBN: 0071826610.

Moore, W.J. and Andrew Starr (Aug. 2006). "An intelligent maintenance system for continuous cost-based prioritisation of maintenance activities". In: *Computers in Industry* 57, pp. 595–606. DOI: 10.1016/j.compind.2006.02.008.

Navlani, Avinash (May 16, 2018). *Datacamp - Understanding Random Forests Classifiers in Python*. URL: https://www.datacamp.com/community/tutorials/random-forests-classifier-python.

Nemeth, Tanja et al. (2018). "PriMa-X: A reference model for realizing prescriptive maintenance and assessing its maturity enhanced by machine learning". In: *Procedia CIRP* 72.1, pp. 1039–1044.

O'Donovan, P. et al. (2015). "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities". In: *Journal of Big Data* 2.1, p. 25. ISSN: 2196-1115. DOI: 10.1186/s40537-015-0034-z. URL: https://doi.org/10.1186/s40537-015-0034-z.

Park, Yeonjeong and I-H Jo (2015). "Development of the learning analytics dashboard to support students learning performance". In: *Journal of Universal Computer Science* 21.1, p. 110.

Paulheim, Heiko (2017). "Knowledge graph refinement: A survey of approaches and evaluation methods". In: *Semantic web* 8.3, pp. 489–508.

Platfrom, Google Cloud (June 17, 2019). *Google Cloud Platform*. URL: https://cloud.google.com/?hl=en.

Podgorelec, V and S Kuhar (2011). "Taking advantage of education data: Advanced data analysis and reporting in virtual learning environments". In: *Elektronika ir Elektrotechnika* 114.8, pp. 111–116.

Power BI (July 12, 2019). URL: https://powerbi.microsoft.com.

Protégé (2019). URL: https://protege.stanford.edu/.

PYMC3 - Probabilistic Programming in Python (June 19, 2019). URL: https://docs.pymc.io/.

Ranjan, R. (May 2014). "Streaming Big Data Processing in Datacenter Clouds". In: *IEEE Cloud Computing* 1.1, pp. 78–83. ISSN: 2325-6095. DOI: 10.1109/MCC.2014.22.

Ranjan, Vikas (2009). *A comparative study between ETL (Extract, Transform, Load) and ELT (Extract, Load and Transform) approach for loading data into data warehouse.* Tech. rep. viewed 2010-03-05, http://www. ecst. csuchico. edu/˜ juliano/csci693

Ricci, Francesco, Lior Rokach, and Bracha Shapira (2011). "Introduction to recommender systems handbook". In: *Recommender systems handbook*. Springer, pp. 1–35.

Richter, Michael M. (2010). *Machine Learning - Case Based Reasoning.* URL: http://pages.cpsc.ucalgary.ca/~mrichter/ML/ML%202010/Experience%20and%20CBR/CBR%202010.pdf.

Robinson, Scott (July 25, 2019). *Introduction to Neural Networks with Scikit-Learn.* URL: https://stackabuse.com/introduction-to-neural-networks-with-scikit-learn/.

Russell, Stuart and Peter Norvig (2009). *Artificial Intelligence: A Modern Approach*. 3rd. Upper Saddle River, NJ, USA: Prentice Hall Press. ISBN: 0136042597, 9780136042594.

Schlund Sebastian Hämmerle Moritz, Tobias Strölin (2014). *Industrie 4.0 Eine Revolution der Arbeitsgestaltung*. Research rep. Fraunhofer-Institut für Arbeitswirtschaft und Organisation Stuttgart.

Schmalhofer, F. (2001). "Expert Systems in Cognitive Science". In: *International Encyclopedia of the Social & Behavioral Sciences*. Ed. by Neil J. Smelser and Paul B. Baltes. Oxford: Pergamon, pp. 5128–5135. ISBN: 978-0-08-043076-8. DOI: https://doi.org/10.1016/B0-08-043076-7/01615-6. URL: http://www.sciencedirect.com/science/article/pii/B0080430767016156.

scikit-learn (June 20, 2019). URL: https://scikit-learn.org.

Sharma, Pawan and Mahendra Sharma (2014). "Artificial intelligence in advance manufacturing technology-a review paper on current application". In: *Int J Eng Manag Sci* 1.1, pp. 4–7.

Shearer, Colin (2000). "The CRISP-DM Model: The New Blueprint for Data Mining". In: *Journal of Data Warehousing*.

Sheykh Esmaili, Kyumars et al. (2011). "Changing flights in mid-air: a model for safely modifying continuous queries". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. ACM, pp. 613–624.

Sielis, George A, Aimilia Tzanavari, and George A Papadopoulos (2015). "Recommender systems review of types, techniques, and applications". In: *Encyclopedia of Information Science and Technology, Third Edition*. IGI Global, pp. 7260–7270.

Smith, Edward E and Douglas L Medin (1981). *Categories and concepts*. Vol. 9. Harvard University Press Cambridge, MA.

spaCy (June 22, 2019). URL: https://spacy.io/.

Staab, Steffen and Rudi Studer (2010). *Handbook on ontologies*. Springer Science & Business Media.

Stadnicka, Dorota, Katarzyna Antosz, and R.M. Chandima Ratnayake (2014). "Development of an empirical formula for machine classification: Prioritization of maintenance tasks". In: *Safety Science* 63, pp. 34–41. ISSN: 0925-7535. DOI: https://doi.org/10.1016/j.ssci.2013.10.020. URL: http://www.sciencedirect.com/science/article/pii/S0925753513002531.

Stephen, Prentice; Yvonne Genovese (June 17, 2011). *Pattern-Based Strategy: Getting Value From Big Data*. Research rep.

Tableau (July 12, 2019). URL: https://www.tableau.com.

Talaminos-Barroso, Alejandro et al. (2016). "A Machine-to-Machine protocol benchmark for eHealth applications Use case: Respiratory rehabilitation". In: *Computer Methods and Programs in Biomedicine* 129, pp. 1–11. ISSN: 0169-2607. DOI: https://doi.org/

10.1016/j.cmpb.2016.03.004. URL: http://www.sciencedirect.com/science/article/pii/S0169260715302959.

Tarus, John K., Zhendong Niu, and Abdallah Yousif (2017). "A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining". In: *Future Generation Computer Systems* 72, pp. 37–48. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2017.02.049. URL: http://www.sciencedirect.com/science/article/pii/S0167739X17303254.

Tensorflow (June 30, 2019). URL: https://www.tensorflow.org/.

Terrizzano, Ignacio G et al. (2015). "Data Wrangling: The Challenging Yourney from the Wild to the Lake." In: *CIDR*.

Tilly, Marcel (2018). "Das Gehirn des Rechners". In: *iX Developer*.

Tipping, Michael E (2004). "Bayesian inference: An introduction to principles and practice in machine learning". In: *Advanced lectures on machine Learning*. Springer, pp. 41–62.

trends.google.com (Apr. 12, 2019). *Google Trends - IoT*. URL: https://trends.google.com/trends/explore?date=2010-01-01%202019-04-12&q=IoT.

Tschinkel, Gerwald et al. (2015). "The Recommendation Dashboard: A System to Visualise and Organise Recommendations". In: *2015 19th International Conference on Information Visualisation*. IEEE, pp. 241–244.

Uhlemann, Thomas, Christian Lehmann, and Rolf Steinhilper (Dec. 2017). "The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0". In: *Procedia CIRP* 61, pp. 335–340. DOI: 10.1016/j.procir.2016.11.152.

Ullrich, Carsten (2016). "An Ontology for Learning Services on the Shop Floor." In: *International Association for Development of the Information Society*.

Van de Velde, Walter (1993). "Issues in knowledge level modelling". In: *Second generation expert systems*. Springer, pp. 211–231.

Variational API quickstart (2019). URL: https://docs.pymc.io/notebooks/variational_api_quickstart.html.

Vogel-Heuser, Birgit, Christian Diedrich, and Manfred Broy (2013). "Anforderungen an CPS aus Sicht der Automatisierungstechnik". In: *at–Automatisierungstechnik at–Automatisierungstechnik* 61.10, pp. 669–676.

Vogel-Heuser, Birgit, Jay Lee, and Paulo Leitão (Jan. 2015). "Agents enabling cyber-physical production systems". In: *at - Automatisierungstechnik* 63. DOI: 10.1515/auto-2014-1153.

W3 (May 24, 2019). *OWL Web Ontology Language*. (2019-05-24). URL: https://www.w3.org/TR/owl-semantics/examples.html (visited on 05/24/2019).

Wang, Huaiqing (1997). "Intelligent agent-assisted decision support systems: integration of knowledge discovery, knowledge analysis, and group decision support". In: *Expert Systems with Applications* 12.3, pp. 323–335.

Wikipedia (May 24, 2019). *Web Ontology Language*. (2019-05-24). URL: https://de.wikipedia.org/wiki/Web_Ontology_Language (visited on 05/24/2019).

Yam, RCM et al. (2001). "Intelligent predictive decision support system for condition-based maintenance". In: *The International Journal of Advanced Manufacturing Technology* 17.5, pp. 383–391.

Yamashina, Hajime and Takashi Kubo (Nov. 2002). "Manufacturing cost deployment". In: *International Journal of Production Research* 40, pp. 4077–4091. DOI: 10.1080/00207540210157178.

Young, Tom et al. (2018). "Recent trends in deep learning based natural language processing". In: *ieee Computational intelligenCe magazine* 13.3, pp. 55–75.

Yu, Ren, Benoit Iung, and Hervé Panetto (2003). "A multi-agents based E-maintenance system with case-based reasoning decision support". In: *Engineering applications of artificial intelligence* 16.4, pp. 321–333.

Zablith, Fouad et al. (2015). "Ontology evolution: a process-centric survey". In: *The knowledge engineering review* 30.1, pp. 45–75.

Zhang, Fuzheng et al. (2016). "Collaborative knowledge base embedding for recommender systems". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, pp. 353–362.

Zhang, Yin, Rong Jin, and Zhi-Hua Zhou (2010). "Understanding bag-of-words model: a statistical framework". In: *International Journal of Machine Learning and Cybernetics* 1.1-4, pp. 43–52.

Zhong, Ray Y et al. (2017). "Big Data Analytics for Physical Internet-based intelligent manufacturing shop floors". In: *International journal of production research* 55.9, pp. 2610–2621.

Zhong, Shisheng, Xiaolong Xie, and Lin Lin (2015). "Two-layer random forests model for case reuse in case-based reasoning". In: *Expert Systems with Applications* 42.24, pp. 9412–9425. ISSN: 0957-4174. DOI: \url{https://doi.org/10.1016/j.eswa.2015.08.005}. URL: %5Curl%7Bhttp://www.sciencedirect.com/science/article/pii/S0957417415005448%7D.

Zumel, N. and J. Mount (2014). *Practical Data Science with R*. Manning Publications Company. ISBN: 9781617291562. URL: %5Curl%7Bhttps://books.google.at/books?id=tJ-HngEACAAJ%7D.