Universität Stuttgart

Datenanalysen für industrielle Anwendungsfälle: Datencharakteristika und Nutzung von Domänenwissen

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Venia Legendi in Informatik als Habilitationsschrift angenommene Abhandlung

vorgelegt von

Dr. rer. nat. Peter Reimann

aus Stuttgart

Hauptberichter: Prof. Dr.-Ing. habil. Bernhard Mitschang

Mitberichter: Prof. Dr. rer. nat. Volker Markl

Institut für Parallele und Verteilte Systeme

2025

Vorwort

Die vorliegende Habilitationsschrift fasst zentrale Ergebnisse von Forschungsarbeiten zusammen, an denen ich als Leiter der Nachwuchsgruppe „IKT-Plattform für die Produktion“ an der Graduiertenschule GSaME (Graduate School of Excellence advanced Manufacturing Engineering)[1] der Universität Stuttgart beteiligt war. Der Forschungsschwerpunkt dieser Nachwuchsgruppe lag im Bereich des Datenmanagements, der Datenbereitstellung sowie der Datenanalyse für industrielle Anwendungsfälle im produzierenden Gewerbe. Diese Anwendungsfälle können unter dem Begriff *Industrial Analytics* zusammengefasst werden. Hierbei werden Daten aus verschiedenen Phasen eines Produktlebenszyklus akquiriert und analysiert. In der Produktentwicklung können bspw. Daten aus Produktsimulationen analysiert werden, um Empfehlungen für die Verbesserung eines Produktdesigns herzuleiten. In der Produktion kann eine Analyse von Sensordaten die Fehlerdiagnose in einer Produktionslinie unterstützen.

Im Rahmen meiner Tätigkeit als Leiter der Nachwuchsgruppe habe ich zahlreiche solcher industriellen Anwendungsfälle für Datenanalysen untersucht. Dabei stellte sich heraus, dass in einem Großteil der Anwendungsfälle die verfügbaren Daten komplexe Charakteristika aufweisen, die gängige Verfahren zum Datenmanagement und zur Datenanalyse vor Herausforderungen stellen. In der vorliegenden Habilitationsschrift habe ich die häufigsten solcher spezifischen und komplexen Datencharakteristika kategorisiert und diskutiert. Beispielsweise liegen Daten häufig in heterogenen Formaten und in isolierten

[1] GSaME: https://www.gsame.uni-stuttgart.de/

Datensilos vor, sodass in vielen Anwendungsfällen nur sehr wenige Daten zur Verfügung stehen. Ebenso erhöht die hohe Produktvielfalt in Unternehmen die Anzahl und Komplexität der in den Daten auftretenden Muster und Zusammenhänge. Diese Muster und Zusammenhänge sowie deren statistische Verteilung in Daten können sich sogar über die Zeit ändern (sog. Datensatzverschiebungen). Werden solche komplexen Datencharakteristika nicht adäquat adressiert, führt dies zu mäßigen Analyseergebnissen, z. B. zu einer geringen Vorhersagegenauigkeit in einer Klassifikation.

In einigen Projekten meiner Nachwuchsgruppe hat sich gezeigt, dass die spezifischen und komplexen Datencharakteristika industrieller Anwendungsfälle durch die Einbeziehung von Domänenwissen effektiver adressiert werden können. Im Rahmen meiner Forschungstätigkeit als Leiter der Nachwuchsgruppe habe ich daher den Stand der Wissenschaft hinsichtlich grundlegender Ansätze zur Einbeziehung dieses Domänenwissens in die drei Schritte Datenakquise, Datenvorbereitung und Datenanalyse von Datenanalyseprozessen untersucht. Neben Ansätzen, die im Rahmen der Nachwuchsgruppe entwickelt wurden, umfasst dies auch mehrere Ansätze aus der verwandten Literatur. Der primäre Fokus liegt dabei auf formal repräsentiertem Domänenwissen, das in einer mathematisch-logischen bzw. in einer digital erfassten Struktur vorliegt und meist Kausalitätswissen über bestimmte Sachverhalte und Zusammenhänge in einer Anwendungsdomäne beschreibt. Beispiele für entsprechende Wissensrepräsentationsformen sind Fallbasen, Regelbasen, Glossare, semantische Netze wie Taxonomien, Ontologien oder Wissensgraphen, probabilistische Graphen sowie mathematische Simulationsmodelle. Die Ansätze, die solche Wissensformen im Rahmen von Datenanalyseprozessen nutzen, habe ich insbesondere dahingehend untersucht, inwieweit sie die typischen komplexen Datencharakteristika industrieller Anwendungsfälle adressieren können. Eine weitere wesentliche Fragestellung war, ob und unter welchen Bedingungen diese Ansätze auf andere Anwendungsfälle und deren Daten übertragbar sind, für die sie in der Literatur noch nicht evaluiert wurden. Die vorliegende Habilitationsschrift fasst die wesentlichen Ergebnisse dieser Untersuchungen zusammen.

Das Verfassen der vorliegenden Habilitationsschrift sowie die Erarbeitung der entsprechenden Forschungsergebnisse wären nicht ohne eine intensive wissenschaftliche Betreuung und Förderung an der GSaME sowie am Institut

für Parallele und Verteilte Systeme (IPVS) der Universität Stuttgart möglich gewesen. An erster Stelle möchte ich mich insbesondere bei Herrn Prof. Bernhard Mitschang für seine stete und konstruktive Unterstützung, seine wertvollen Ratschläge und die vielen interessanten Diskussionen und Denkanstöße bedanken. Herrn Prof. Volker Markl danke ich für seine Bereitschaft, den Mitbericht im Zuge meines Habilitationsverfahrens zu übernehmen. Großen Dank schulde ich auch den Doktorandinnen und Doktoranden der GSaME, die Teil meiner Nachwuchsgruppe waren und die durch ihre Arbeiten einen unverzichtbaren Beitrag zu meiner Forschung geleistet haben. Hierbei sind insbesondere Vitali Hirsch, Cornelia Kiefer, Alejandro Villanueva, Christian Weber, Yannick Wilhelm und Julian Ziegler zu nennen. Ein besonderer Dank geht zudem an Holger Schwarz und insbesondere an Michael Behringer für die kostbare Zeit, die sie für das Korrekturlesen dieser Arbeit und für wertvolle Kommentare und Hinweise aufgewendet haben. Weiterhin danke ich Christoph Stach, Christoph Gröger und Herrn Prof. Ulf Schreier für interessante und hilfreiche Diskussionen. Schließlich möchte ich diese Schrift zwei ganz besonderen Menschen widmen und ihnen meinen aufrichtigen Dank für ihre Geduld und moralische Unterstützung während meiner intensiven Forschungsarbeit aussprechen: meinen Eltern Christa und Gunter Reimann.

Peter Reimann
Stuttgart im Mai 2025

Inhaltsverzeichnis

Abkürzungsverzeichnis 11

1 Einleitung 15

2 Grundlagen 21
2.1 Industrial Analytics 21
2.2 Arten formal repräsentierten Domänenwissens 26
2.3 Vorgehensmodelle für Datenanalyseprozesse 37
2.3.1 Generisches Vorgehensmodell 37
2.3.2 Einbeziehung des Wissens von Domänenexperten in Vorgehensmodelle für Datenanalyseprozesse 45

3 Datencharakteristika in industriellen Anwendungsfällen 49
3.1 Industrielle Anwendungsfälle für Datenanalysen 49
3.1.1 Anwendungsfälle in der Produktentwicklung 51
3.1.2 Anwendungsfälle in der Produktion 52
3.1.3 Anwendungsfälle im Vertrieb 54
3.1.4 Anwendungsfälle in der Produktnutzung 54
3.1.5 Anwendungsfälle zur Prozessanalyse 55
3.2 Datencharakteristika in industriellen Anwendungsfällen 56
3.2.1 Heterogene Datenformate und isolierte Datensilos . . . 57
3.2.2 Datencharakteristika als Ursachen für Bias in Daten . . 58
3.2.3 Datencharakteristika in nichtstationären Umgebungen . 69

3.3 Datencharakteristika in anderen Anwendungsdomänen 74
3.3.1 Heterogene Datenformate und isolierte Datensilos . . . 76
3.3.2 Datencharakteristika als Ursachen für Bias in Daten . . 77
3.3.3 Datencharakteristika in nichtstationären Umgebungen . 80

4 Effekte datengetriebener Ansätze 83
4.1 Datenakquise und Datenexploration 85
4.1.1 Ansätze und Tools zur Datenintegration 85
4.1.2 Öffentlich verfügbare Datenrepositorys 86
4.1.3 Datengeneratoren 88
4.2 Datenvorbereitung 89
4.2.1 Sampling-Verfahren 90
4.2.2 Statistische Feature-Selektion 92
4.2.3 Binärisierung einer Multi-Klassen-Klassifikation 93
4.2.4 Tools zur Spezifikation von Datenvorbereitungs-Pipelines 96
4.3 Datenanalyse und Modellbildung 98
4.3.1 Ensemble-Methoden für die Klassifikation 98
4.3.2 AutoML und Meta-Learning 102
4.3.3 Ansätze zur Adressierung von Datensatzverschiebungen 104
4.4 Zusammenfassung 108

5 Nutzung von Domänenwissen zur Datenakquise und -exploration 111
5.1 Semantische Informationen in Metadatenmodellen zur Unterstützung von Domänenexperten bei der Datenakquise 113
5.1.1 Metadaten zur Unterstützung der Datenakquise und Datenexploration 113
5.1.2 Domänenwissen und semantische Informationen in Metadatenmodellen 115
5.2 Metadatenmodell zur Unterstützung der Datenakquise in der virtuellen Produktentwicklung 118
5.2.1 Bestandteile des Metadatenmodells 119
5.2.2 Zentrale Ergebnisse der Validierung 122
5.2.3 Übertragbarkeit des Metadatenmodells auf andere Anwendungsdomänen 124

5.3 Nutzung einer Taxonomie zur Generierung von Daten mit einer Gruppenheterogenität und Gruppenungleichverteilung 125
5.3.1 Vorgehensweise des Ansatzes zur Datengenerierung . . 126
5.3.2 Zentrale Evaluationsergebnisse 128
5.3.3 Übertragbarkeit des Ansatzes zur Datengenerierung auf andere Anwendungsdomänen und Datencharakteristika 130
5.4 Zusammenfassung 131

6 Nutzung von Domänenwissen zur Datenvorbereitung 133
6.1 Feature-Engineering mit Regeln oder semantischen Netzen . . 134
6.2 Festlegung der Labels mit Regeln oder semantischen Netzen . . 137
6.3 Anreicherung der Daten mit Simulationsergebnissen 139
6.4 Datenpartitionierung mit Constraint-basierter Clusteranalyse . 142
6.5 Verfahren zur Datenpartitionierung mittels einer Taxonomie . . 145
6.5.1 Nutzung einer Taxonomie zur Datenpartitionierung . . 146
6.5.2 Zentrale Evaluationsergebnisse 152
6.5.3 Übertragbarkeit des Verfahrens auf andere Anwendungsdomänen 156
6.6 Ontologiebasierte Ansätze zur Spezifikation von Pipelines für die Datenvorbereitung 161
6.7 Zusammenfassung 165

7 Nutzung von Domänenwissen zur Datenanalyse und Modellbildung 167
7.1 Modellverwaltungsplattformen als Expertensysteme zum Vorschlag von ML-Modellen 167
7.1.1 Metadaten zur Beschreibung von ML-Modellen 169
7.1.2 AssistML-Ansatz als Expertensystem zur Empfehlung von ML-Lösungen 171
7.1.3 Plattform mit semantischen Informationen in Modell-Metadaten 175
7.1.4 Zentrale Evaluationsergebnisse 177
7.1.5 Adressierung der Datencharakteristika industrieller Anwendungsfälle 181
7.1.6 Übertragbarkeit von AssistML und der MMP auf andere Anwendungsdomänen 185

7.2 Explizite Nutzung von Ontologien, Wissensgraphen oder Simulationsmodellen im Lernprozess von ML-Modellen 186
7.2.1 Einbindung des Domänenwissens als Teil der Struktur eines ML-Modells 187
7.2.2 Nutzung des Domänenwissens in Lernalgorithmen . . . 188
7.2.3 Nutzung des Domänenwissens zur Bewertung von ML-Modellen 189
7.2.4 Adressierung der Datencharakteristika industrieller Anwendungsfälle 189
7.3 Inferenz und Vorhersagen mit probabilistischen Graphen 191
7.3.1 Erzeugung probabilistischer Graphen durch Domänenexperten sowie Ergänzung der Graphen durch datengetriebene Lernverfahren 192
7.3.2 Adressierung der Datencharakteristika industrieller Anwendungsfälle 193
7.4 Entscheidungsfusion in Ensembles aus datengetriebenen ML-Modellen und wissensbasierten Modellen 195
7.4.1 PUSION: Generisches Rahmenwerk zur Optimierung der Entscheidungsfusion in Ensembles 197
7.4.2 Zentrale Evaluationsergebnisse für PUSION 199
7.4.3 Adressierung der Datencharakteristika industrieller Anwendungsfälle 202
7.4.4 Übertragbarkeit von PUSION auf andere Anwendungsdomänen 204
7.5 Zusammenfassung 205

8 Zusammenfassung und Ausblick 207
8.1 Zentrale Ergebnisse der Diskussion 207
8.2 Ausblick 211

Literaturverzeichnis 213

Abbildungsverzeichnis 251

Tabellenverzeichnis 255

Abkürzungsverzeichnis

AutoML	Automated Machine Learning
AutomationML	Automation Markup Language
BARENTS	tailorable data preparation zone for Data Lakes
CAD	Computer-aided Design
CAE	Computer-aided Engineering
CAEX	Computer-aided Engineering Exchange
CAT	Computer-Aided Testing
CEG	Cause & Effect Graph
CFD	Compuational Fluid Dynamics
COMPAS	Correctional Offender Management Profiling for Alternative Sanctions
CPI	Class Partitioning according to Imbalance – Klassenaufteilung gemäß Ungleichgewicht
CRISP-DM	Cross-Industry Standard Process for Data Mining
CRM	Customer Relationship Management
CSMM	Core Scientific Metadata Model
CSV	Comma-separated Values
DB	Datenbank
DSGVO	Datenschutz-Grundverordnung
EEG	Elektroenzephalographie
EKG	Elektrokardiogramm
EMG	Elektromyographie

ENG	Elektroneurographie
EoL	End-of-Line(-Prüffeld)
ERP	Enterprise Resource Planning
ETL	Extraktion, Transformation, Laden (von Daten)
FEM	Finite-Elemente-Methode
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode and Effects and Criticality Analysis
HANDLE	Handling metAdata maNagement in Data LakEs
i.O.	in Ordnung
IA	Industrial Analytics
IKT	Informations- und Kommunikationstechnologie
IOT	Internet der Dinge
iIOT	industrielles Internet der Dinge
IT	Informationstechnologie
JSON	JavaScript Object Notation
JT	Jupiter Tesselation
KEEL	Knowledge Extraction based on Evolutionary Learning (Repository)
KDD-Prozess	Kowledge Discovery in Databases Process
KNIME	Konstanz Information Miner
KNN	K-nächster-Nachbar
MEB	Merkmalentstehungsbaum
MEDAL	MEtadata model for DAta Lakes
MES	Manufacturing Execution System
ML	Maschinelles Lernen
MLOps	Machine Learning Operations
MMP	Model Management Platform
n.i.O.	nicht in Ordnung
OPC UA	Open Platform Communications Unified Architecture
OvO	One-vs-One
OvA	One-vs-All
PLM	Product Lifecycle Management
PMML	Predictive Model Markup Language
PUSION	Python Universal Fusion

RA	Rework Attempts – Nacharbeitsversuche
RUS	Random Undersampling
ROS	Random Oversampling
SCADA	Supervisory Control and Data Acquisition
SDB	Scientific Data Bag
SEMMA	Sample, Explore, Modify, Model, and Assess
SMOTE	Synthetic Minority Oversampling Technique
SPH	Segmentation according to Product Hierarchy – Segmentierung gemäß Produkthierarchie
TDSP	(Microsoft) Team Data Science Process
UCI ML	University of California Irvine Machine Learning (Repository)
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UWD	Ursache-Wirkungs-Diagramm
XML	Extensible Markup Language
WA	Web Analytics
W3C	World Wide Web Consortium

Kapitel 1
Einleitung

Industrieunternehmen des produzierenden Gewerbes sehen enorme Potenziale in der Analyse ihrer Daten [DB21; Grö21; SAH18]. Entsprechende industrielle Anwendungsfälle für Datenanalysen werden nach Gröger [Grö22] unter dem Begriff *Industrial Analytics* zusammengefasst. Das primäre Ziel für Unternehmen ist, aus den Analyseergebnissen Handlungsempfehlungen abzuleiten, mit denen Produkte, Prozesse und Dienstleistungen innerhalb der industriellen Wertschöpfung verbessert werden können. Dabei werden Daten aus verschiedenen Phasen eines Produktlebenszyklus analysiert. In der Produktentwicklung können bspw. Simulationsdaten analysiert werden, um Empfehlungen zur Verbesserung eines Produktdesigns zu erhalten [ZRKM20]. In der Produktion kann eine Analyse von Sensordaten die Fehlerdiagnose von Produktdefekten unterstützen [GSH+20; HRKM18]. Mit Daten aus der Produktnutzung kann z. B. die Ursache für einen Produktausfall bestimmt werden [KM16].

Zur Umsetzung von Datenanalysen existieren etablierte Vorgehensmodelle wie der *Cross-Industry Standard Process for Data Mining (CRISP-DM)* [CCK+00] oder der *Kowledge Discovery in Databases Process* (KDD-Prozess) [FPS96]. Aus solchen Vorgehensmodellen lässt sich ein generisches Vorgehensmodell mit sechs Schritten ableiten, das in dieser Habilitationsschrift als Bezugsrahmen zur Einordnung aller untersuchten Ansätze dient. Im ersten Schritt wird die fachliche Problemstellung untersucht, wobei u. a. die mit der Datenanalyse zu

erreichende Zielsetzung formuliert wird. Der zweite Schritt umfasst die Akquise relevanter Daten aus verfügbaren Quellen sowie eine Datenexploration, um erste Zusammenhänge in den Daten zu erkennen. Anschließend werden die Daten im dritten Schritt vorbereitet, wobei bspw. die Datenqualität verbessert wird. Im vierten Schritt wird die eigentliche Datenanalyse durchgeführt, indem geeignete Analysealgorithmen auf die Daten angewendet werden. Die Analyseergebnisse werden in den letzten beiden Schritten hinsichtlich der am Anfang definierten Zielsetzung bewertet und im Anwendungsfall eingesetzt.

Den zentralen Untersuchungsgegenstand dieser Habilitationsschrift bilden Problemstellungen, die sich für Datenanalysen aus den spezifischen und meist komplexen Datencharakteristika industrieller Anwendungsfälle ergeben. Nachdem Kapitel 2 grundlegende Begriffe und Konzepte wie Industrial Analytics, verschiedene Kategorien von Domänenwissen sowie Vorgehensmodelle für Datenanalyseprozesse erläutert, werden in Kapitel 3 auf Basis einer Analyse mehrerer industrieller Anwendungsfälle die häufigsten solcher spezifischen Datencharakteristika kategorisiert und diskutiert. Beispielsweise liegen Daten häufig in heterogenen Formaten und in isolierten Datensilos vor [Grö18]. Dies erschwert die Akquise von Daten, die für eine Analyse benötigt werden [ZRKM20]. Oftmals stehen nur sehr wenige Daten in ausreichender Qualität für die Datenanalyse zur Verfügung [HRM19; KM16]. Ebenso erhöht die hohe Produktvielfalt in Unternehmen die Anzahl und Komplexität der in den Daten auftretenden Muster, sodass es Analysealgorithmen häufig nicht gelingt, alle Muster korrekt zu erkennen [HRM19; SRYD14; TGS+19; TRSM23; WRGM21]. Diese Muster sowie deren statistische Verteilung in Daten können sich zudem im Rahmen sog. Datensatzverschiebungen über die Zeit ändern [MRA+12; SRWM22; WHC+16]. Werden solche Datencharakteristika bei Datenanalysen nicht adäquat adressiert, führt dies zu mäßigen Analyseergebnissen, z. B. zu einer geringen Vorhersagegenauigkeit in einer Klassifikation.

Als Analyseverfahren kommen in den untersuchten industriellen Awendungsfällen primär Verfahren des überwachten maschinellen Lernens (ML) zum Einsatz, insbesondere für Klassifikationen [CCS+18; DB21; HRKM18; LLE16; TGS+19; WWIT16]. In einigen Awendungsfällen werden auch Regressionen [HMB+23], Textanalysen [KM16; KRM19a] oder Prozessanalysen [BWD+21; DRG21] eingesetzt. Dabei werden strukturierte Daten aus

Informationssystemen, Sensordaten und auch unstrukturierte Daten wie z. B. Textdaten oder Geometriedaten analysiert. Zwar stehen in dieser Habilitationsschrift industrielle Anwendungsfälle im Fokus, es werden aber an geeigneten Stellen Querbezüge zu anderen Anwendungsfällen und deren Daten außerhalb des industriellen Bereichs diskutiert, wie bspw. für medizinische Diagnosen mit Zeitreihendaten oder Bilddaten [CRM+20; KU15; MGM20].

In Kapitel 4 werden mehrere rein datengetriebene Ansätze diskutiert, die jeweils einzelne der komplexen Datencharakteristika industrieller Anwendungsfälle adressieren können. Dabei stehen Ansätze im Fokus, die in den drei Schritten Datenakquise, Datenvorbereitung sowie Datenanalyse des o. g. generischen Vorgehensmodells für Datenanalyseprozesse eingesetzt werden, da die komplexen Datencharakteristika insbesondere in diesen Schritten zu Problemen führen und daher auch dort adressiert werden sollten. Im Schritt der Datenakquise werden bspw. Ansätze zur Datenintegration [LN07] oder Datengeneratoren [FSC+20; RHW21] diskutiert. Beispiele für Ansätze, die spezifische Datencharakteristika in der Datenvorbereitung adressieren, sind Verfahren zum Sampling [GFB+12; HG09; SWK09; WY12] oder zur statistischen Feature-Selektion [KR10]. Im Schritt der Datenanalyse werden u. a. Ensemble-Methoden [WGC14] oder AutoML-Systeme (*Automated Machine Learning*) [HZC21; XWL+21; ZH21] untersucht. Grundlage für die Diskussion in Kapitel 4 sind mehrere vergleichende Studien zu datengetriebenen Ansätzen, die in den vom Autor dieser Schrift betreuten Projekten entstanden sind. Die Ergebnisse dieser Studien werden in den Gesamtkontext der Schrift eingeordnet und durch weitere Diskussionen des Autors ergänzt. Das zentrale Ergebnis ist, dass rein datengetriebene Ansätze zur Datenakquise, Datenvorbereitung und Datenanalyse jeweils nur wenige der komplexen Datencharakteristika industrieller Anwendungsfälle adressieren. Dabei haben viele der Ansätze negative Effekte auf die nicht berücksichtigten Datencharakteristika, was sich häufig negativ auf die Güte der Analyseergebnisse auswirkt.

Verschiedene Arbeiten zeigen, dass viele der komplexen Datencharakteristika industrieller Anwendungsfälle durch die Einbeziehung von Domänenwissen effektiver adressiert werden können (siehe bspw. [HRT+23; Rei23; RMB+23; WRGM21]). Grundsätzlich kann Domänenwissen in informelles Erfahrungswissen von Domänenexperten sowie in Domänenwissen, das in einer formalen

Repräsentationsform vorliegt, eingeteilt werden [Leh21; Pol09]. In der Praxis wird in den Schritten Datenakquise, Datenvorbereitung sowie Datenanalyse des o. g. generischen Vorgehensmodells für Datenanalyseprozesse häufig nur sporadisch das informelle Erfahrungswissen von Domänenexperten einbezogen, bspw. von Produktentwicklungs- oder Produktionsingenieuren. Zum Beispiel können Experten gebeten werden, einzelne Klassenlabels in einem Datensatz festzulegen [RBE+20; SRYD14]. Dass Domänenexperten ihr Wissen nur sporadisch für wenige Teilaufgaben einbringen, ist vor allem darin begründet, dass der dafür von ihnen zu erbringende Aufwand oft unverhältnismäßig hoch ist.

Der Fokus dieser Habilitationsschrift liegt daher auf der Diskussion von Ansätzen, die formal repräsentiertes Domänenwissen in Datenanalyseprozessen einbeziehen. Formal repräsentiertes Domänenwissen liegt in einer mathematisch-logischen bzw. in einer digital erfassten Struktur vor, die bereits im Vorfeld von ggf. mehreren Domänenexperten definiert wurde. Es beschreibt meistens Kausalitätswissen über bestimmte Sachverhalte und Zusammenhänge in einer Anwendungsdomäne. Beispiele entsprechender Wissensrepräsentationsformen, die auch primär in dieser Arbeit untersucht werden, sind Fallbasen [SS09a], Regelbasen [Ros03], Glossare oder kontrollierte Vokabulare [ANSI/NISO10], semantische Netze wie Taxonomien, Ontologien oder Wissensgraphen [EW16; Sow91], probabilistische Graphen [KF09] sowie mathematische Simulationsmodelle [BZBP13]. Die Nutzung formal repräsentierten Domänenwissens hat folgende Vorteile: (1) Da es in der jeweiligen Anwendungsdomäne bereits vorliegt, müssen Domänenexperten selbst nicht direkt in Datenanalyseprozesse involviert werden, um das Domänenwissen einzubeziehen. (2) Formal repräsentiertes Kausalitätswissen, das von mehreren Domänenexperten definiert wurde, ist i. d. R. umfassender als das informelle Erfahrungswissen einzelner Domänenexperten. Daher können damit potenziell auch die komplexen Datencharakteristika industrieller Anwendungsfälle besser adressiert werden.

Allerdings fehlt in diesem Themenbereich der Einbeziehung von Domänenwissen in Datenanalyseprozessen eine systematische und umfassende wissenschaftliche Diskussion verschiedener Fragestellungen. Insbesondere betrachtet ein Großteil der einschlägigen Literatur jeweils nur eine Teilmenge der möglichen Ansätze zur Einbeziehung von Domänenwissen. Ebenso werden häufig nur die Effekte auf eine Teilmenge der komplexen Datencharakteristika

industrieller Anwendungsfälle untersucht. Oftmals wird primär betrachtet, inwieweit die Ansätze helfen, mit einer kleinen Menge an verwertbaren Daten umzugehen [GNH22; RMB+23; WRGM21]. Die vorliegende Habilitationsschrüft schließt diese Lücke, indem sie in Kapitel 5 bis 7 systematisch den Stand der Wissenschaft hinsichtlich Ansätzen zur Einbeziehung verschiedener Arten formal repräsentierten Domänenwissens in unterschiedliche Aufgaben von Datenanalyseprozessen diskutiert. Dabei werden grundlegende Ansätze analysiert, die jeweils in einem der Schritte Datenakquise (Kapitel 5), Datenvorbereitung (Kapitel 6) sowie Datenanalyse (Kapitel 7) des o. g. generischen Vorgehensmodells für Datenanalyseprozesse verortet werden. Die Ansätze werden hinsichtlich der unterstützten Aufgabe in Datenanalyseprozessen sowie hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens kategorisiert. Ebenso wird diskutiert, welche der in Kapitel 3 erläuterten komplexen Datencharakteristika die Ansätze erfolgreich adressieren und auf welche Datencharakteristika sie keine oder sogar negative Effekte haben.

Im Schritt der Datenakquise (Kapitel 5) werden Metadatenmodelle diskutiert, in denen Metadaten mit Begriffen aus Glossaren oder semantischen Netzen verschlagwortet werden können, um somit die Akquise von Daten aus heterogenen und isolierten Datenquellen zu erleichtern [EGG+21a; GLN+05; ZRKM21]. Weiterhin wird ein Datengenerator betrachtet, der ein semantisches Netz nutzt, um Daten mit realitätsnahen Datencharakteristika zu erzeugen [TRSM23]. Für die Datenvorbereitung (Kapitel 6) finden sich in der Literatur vor allem Ansätze, die eine Regelbasis, semantische Netze oder Simulationsmodelle zum Feature-Engineering, zur Festlegung der Labels in Daten oder zur Anreicherung von Daten verwenden [GNH22; ITW05; RBE+20; SRYD14]. Außerdem werden in dieser Schrift Verfahren diskutiert, die Regeln oder semantische Netze zur Partitionierung von Daten in einzelne Teilmengen verwenden [HRT+23; WCRS01]. Ebenso sind ontologiebasierte Ansätze von Interesse, die die Spezifikation von Datenvorbereitungs-Pipelines vereinfachen [SBE+21]. Im Kontext der Datenanalyse (Kapitel 7) werden Modellverwaltungsplattformen [VSL+16; ZCD+18] untersucht, die ML-Modelle mit Metadaten beschreiben, die erneut mit Begriffen aus Glossaren oder semantischen Netzen verschlagwortet werden können [WHR20]. Weitere Ansätze nutzen Ontologien, Wissensgraphen oder Simulationsmodelle, um bspw. Teilstrukturen von ML-Modellen vorzugeben

Tabelle 1.1: Ansätze zur Einbeziehung formal repräsentierten Domänenwissens in Datenanalyseprozessen, die im Rahmen von Projekten entstanden sind, die vom Autor dieser Schrift betreut wurden.

Ansatz	**Publikationen**	**Abschnitt**
Metadatenmodell zur Unterstützung der Datenakquise in der Produktentwicklung	[ZRKM20; ZRKM21]	Abschnitt 5.2 *(Datenakquise)*
Nutzung einer Taxonomie zur Generierung von Daten	[TRSM23]	Abschnitt 5.3 *(Datenakquise)*
Nutzung einer Taxonomie zur Partitionierung von Daten	[HRM20a; HRT+23]	Abschnitt 6.5 *(Datenvorbereitung)*
Modellverwaltungsplattformen zum Vorschlag von Modellen des maschinellen Lernens (ML-Modelle)	[VRWM23; VWRM21; WHR20; WR20]	Abschnitt 7.1 *(Datenanalyse)*
Rahmenwerk zur Optimierung der Entscheidungsfusion in Ensembles	[WRG+23]	Abschnitt 7.4 *(Datenanalyse)*

oder den Lernprozess für ML-Modelle einzuschränken [GNH22; RMB+23]. Weiterhin kann Domänenwissen direkt in probabilistischen Graphen modelliert und somit für eine Inferenz verwendet werden [Nea03]. Schließlich können neben datengetriebenen ML-Modellen auch wissensbasierte Modelle, wie probabilistische Graphen, Simulationsmodelle oder semantische Netze in einer Ensemble-Methode kombiniert werden [WRG+23; WRGM21].

Neben einem Überblick über verschiedene grundlegende Ansätze umfassen Kapitel 5 bis 7 auch eine Kategorisierung und Diskussion von Ansätzen, die im Rahmen von Projekten entstanden sind, die vom Autor dieser Habilitationsschrift betreut wurden. Tabelle 1.1 fasst diese Ansätze sowie die jeweiligen Publikationen, in denen weitere Details erläutert werden, zusammen. Speziell für diese Ansätze wird in der vorliegenden Schrift zusätzlich erörtert, ob und unter welchen Bedingungen sie auf andere Anwendungsfälle sowie deren Daten übertragbar sind, für die sie in den Publikationen noch nicht evaluiert wurden. Dies schließt die Übertragbarkeit auf weitere industrielle Anwendungsfälle, aber auch auf andere Anwendungsdomänen wie datengetriebene medizinische Diagnosen ein [CRM+20; KU15; MGM20]. Kapitel 8 fasst schließlich die zentralen Erkenntnisse aus der Diskussion in dieser Arbeit zusammen und erläutert mögliche Fragestellungen, die einer weiteren Bearbeitung bedürfen.

Kapitel 2
Grundlagen

In diesem Kapitel wird der Kontext der vorliegenden Habilitationsschrift festgelegt, indem zentrale Begriffe und grundlegende Konzepte erläutert werden. Die im Rahmen dieser Arbeit untersuchten industriellen Anwendungsfälle stehen im Kontext von Industrial Analytics nach der Definition von Gröger [Grö22], die in Abschnitt 2.1 zusammengefasst wird. Anschließend kategorisiert Abschnitt 2.2 verschiedene Arten formal repräsentierten Domänenwissens, die in dieser Arbeit primär untersucht werden. Der letzte Abschnitt 2.3 leitet aus existierenden Vorgehensmodellen für Datenanalyseprozesse ein generisches Vorgehensmodell ab, dessen Schritte in den folgenden Kapiteln als Bezugsrahmen zur Einordnung von Lösungsansätzen zur Nutzung von Domänenwissen in Datenanalyseprozessen dient.

2.1 Industrial Analytics

Industrial Analytics bezeichnet die Analyse der in Unternehmen anfallenden Daten, um Erkenntnisse zu erhalten, mit denen die industrielle Wertschöpfung unterstützt und verbessert werden kann [Grö22]. Dabei wird eine produktorientierte Sichtweise der Wertschöpfung verfolgt, d. h. sie wird anhand des industriellen Produktlebenszyklus definiert [Sta20]. Abbildung 2.1 zeigt typische Phasen eines Produktlebenszyklus. Ebenso zeigt sie die zentralen Schritte

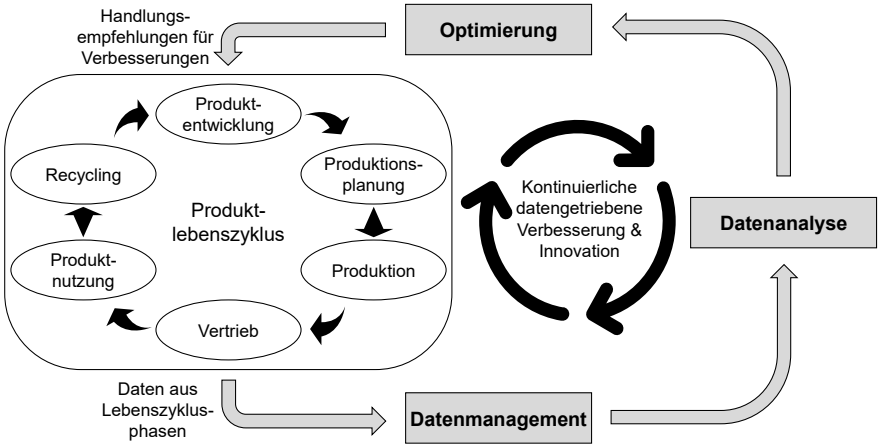


Abbildung 2.1: *Industrial Analytics Cycle*: Kontinuierlicher Lernzyklus im Produktlebenszyklus, vgl. Gröger [Grö22].

eines Lernzyklus nach Gröger [Grö18; Grö22], der eine kontinuierliche datengetriebene Verbesserung und Innovation von Prozessen und Produkten ermöglichen soll. Im ersten Schritt *Datenmanagement* dieses Lernzyklus werden Daten aus den Phasen des Produktlebenszyklus akquiriert und für die nachfolgende Analyse vorbereitet. Im Allgemeinen speichern diese Daten Informationen zu bestimmten, für den Anwendungsfall relevanten Ereignissen oder Entitäten. In einer Produktionslinie können Daten bspw. die Ausführung einzelner Produktionsschritte, die produzierten Produkte bzw. Produktteile aber auch Maschinenfehler oder Probleme bzgl. der Produktqualität repräsentieren.

Im Schritt der *Datenanalyse* werden die vorbereiteten Daten mithilfe unterschiedlicher Analysetechniken ausgewertet [Cao17]. Dies bezieht sich insbesondere auf deskriptive, diagnostische und prädiktive Datenanalysen [LKS13]. In *deskriptiven* Analysen werden in der Vergangenheit oder in der Gegenwart aufgetretene Ereignisse in den Daten identifiziert und näher untersucht. Beispielsweise kann ein statistischer Report eine Übersicht geben, welche Maschinenfehler an welchen Schritten einer Produktionslinie wie häufig aufgetreten sind oder wie viele Produkte jeweils fehlerhaft produziert wurden. Der primäre Fokus *diagnostischer* Analyseverfahren ist eine Ursachenanalyse, insbesondere um herauszufinden, warum die identifizierten Ereignisse aufgetreten sind. In

einer Produktionslinie können z. B. über Textanalysen oder weitere statistische Auswertungen die Ursachen eingegrenzt werden, warum bestimmte Produkte Qualitätsprobleme aufweisen [KM16; KRM19a]. *Prädiktive* Verfahren versuchen, bestimmte Ereignisse vorherzusagen. Als Beispiel kann mittels Techniken des Data Mining oder des maschinellen Lernens in einzelnen Schritten einer Produktionslinie vorhergesagt werden, ob die Produkte am Ende der Linie Fehler aufweisen werden oder nicht [GSH+20; WSR+20].

Der letzte Schritt *Optimierung* im Lernzyklus nach Gröger [Grö22] betrachtet *präskriptive* Verfahren, d. h. es werden auf Basis der zuvor erzeugten Analyseergebnisse konkrete Handlungsempfehlungen zur Verbesserung der Prozesse und Produkte abgeleitet [LKS13]. Dabei stehen Handlungsempfehlungen zur Erreichung eines bestimmten vordefinierten Ziels im Vordergrund, z. B. die Verbesserungen der Produktqualität oder die Vermeidung von Maschinenfehlern. Der gesamte Lernzyklus beschreibt ein iteratives Vorgehen. Die verbesserten Prozesse und Produkte erzeugen wieder neue Daten, die als Grundlage für weitere Datenanalysen sowie Prozess- und Produktverbesserungen dienen.

Gröger [Grö22] charakterisiert Anwendungsfälle des Industrial Analytics zudem über eine Abgrenzung zu Datenanalysen aus dem Bereich von Webanwendungen. Dies wird auch *Web Analytics* genannt [ZP15], das sich insbesondere bezüglich der Zielsetzung der Datenanalyse sowie der Charakteristika der zur Verfügung stehenden Daten von Industrial Analytics unterscheidet (siehe Tabelle 2.1). Web Analytics wird vor allem von Internetunternehmen wie Amazon, Alphabet oder Meta vorangetrieben. Ein häufiges Ziel ist die Analyse des Verhaltens von Nutzern auf Webseiten, um damit z. B. digitales Marketing zu unterstützen. Die Daten für Web Analytics kommen oft aus Logs von Webservern oder von Informationssystemen, die von den entsprechenden Webanwendungen genutzt werden, z. B. von Webshops mit Daten zu Produkten und Kundenbestellungen. Als Datenanalysen werden vielfach statistische Auswertungen der Daten oder Trendanalysen durchgeführt [ZP15]. Zudem finden im Bereich sozialer Netzwerke Analysen auf Graphdaten Anwendung [OR02].

Die Hauptakteure bei Industrial Analytics sind Industrieunternehmen des produzierenden Gewerbes. Solche Unternehmen möchten mit Datenanalysen ihre Produkte, Prozesse und Dienstleistungen verbessern [Grö22]. Die Daten unterscheiden sich wesentlich von denen aus dem Bereich Web Analytics. Einige

Tabelle 2.1: Wesentliche Unterscheidungsmerkmale zwischen Web Analytics und Industrial Analytics, vgl. Gröger [Grö22].

	Web Analytics	**Industrial Analytics**
Hauptakteure	Internetunternehmen	Unternehmen des produzierenden Gewerbes
Häufiges Ziel der Datenanalyse	Analyse des Nutzerverhaltens, digitales Marketing	Analyse und Verbesserung der Produkte, Prozesse und Dienstleistungen
Typische Datenquellen	Logs von Webservern, Informationssysteme von Webanwendungen	Informationssysteme der Unternehmen (ERP, CRM, PLM, MES, ...), iIoT-Sensordaten, von Menschen erzeugte Daten
Typische Datenanalysen	Statistische Auswertungen, Trendanalysen, Graphdatenanalysen	Statistische Auswertungen, Reporting, Data Mining, maschinelles Lernen

der Daten für Industrial Analytics kommen aus den Informationssystemen der Unternehmen, die Daten über verschiedene Phasen eines Produktlebenszyklus erfassen und verwalten. Dies sind z. B. ERP-[1], CRM-[2] oder PLM-Systeme[3] sowie ein MES[4]. Weitere Daten werden in dem sog. *industriellen Internet der Dinge (iIoT)* über Sensoren akquiriert, z. B. Daten von Maschinen in einer Produktionslinie oder von IoT-fähigen Produkten. Zudem erzeugen häufig noch die menschlichen Akteure selbst Daten, indem sie diese bspw. direkt in digitalen Dokumenten speichern oder indem sie Beiträge in unternehmensinternen sozialen Medien eintragen [KHW+17].

Gemäß verschiedener Übersichtsartikel kommen in Anwendungsfällen des Industrial Analytics neben statistischen Auswertungen und Reporting-Techniken vermehrt Verfahren des Data Mining und des maschinellen Lernens zum Einsatz [CCS+18; DB21; Grö22; KBT11; SAH18; WWIT16]. Dies trifft mit sehr wenigen Ausnahmen auch auf die in dieser Arbeit primär betrachteten industriellen Anwendungsfälle zu (siehe Abschnitt 3.1). Insbesondere werden in dieser Arbeit vor allem Anwendungsfälle für überwachte Lernverfahren betrachtet.

[1] ERP: Enterprise Resource Planning
[2] CRM: Customer Relationship Management
[3] PLM: Product Lifecycle Management
[4] MES: Manufacturing Execution System

In den meisten Fällen werden überwachte Lernverfahren für prädiktive Datenanalysen eingesetzt, sie können aber auch innerhalb diagnostischer oder präskriptiver Analysen verwendet werden [LKS13]. Entsprechende Lernalgorithmen analysieren sogenannte Trainingsdaten, um darin enthaltene Muster oder Zusammenhänge zu erkennen [Mit97; MRT18]. Sie bilden die erkannten Muster in statistischen Modellen ab, die auch Modelle des maschinellen Lernens oder kurz *ML-Modelle* genannt werden. Diese Modelle können anschließend auf neue Daten angewendet werden, um festzustellen, welche der erkannten Muster sich in den neuen Daten wiederfinden. Typischerweise werden dabei bestimmte Variablen der Daten vorhergesagt, deren Werte in den Trainingsdaten vorhanden, aber in den neuen Daten noch unbekannt waren.

Bei überwachten Lernverfahren besteht ein *Datensatz* $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ aus n *Dateninstanzen* mit Eingabedaten x_i sowie Ausgabedaten y_i. Eine Dateninstanz repräsentiert ein Ereignis oder eine Entität aus dem realen Anwendungsfall. Jedes x_i ist ein Vektor aus dem Feature-Raum $\mathcal{F} = \{f_1, \ldots, f_F\}$ mit F Dimensionen. Die einzelnen Features $f_1, \ldots, f_F$ repräsentieren die in den Daten enthaltenen Attribute der realen Entitäten, z. B. Sensorsignale der Maschinen in einer Produktionslinie [GSH+20]. Jedes Feature $f_j \in \mathcal{F}$ kann einen eigenen Wertebereich $\mathcal{W}(f_j)$ haben, z. B. für kategorische oder numerische Werte. Jedes x_i ist folglich ein Element des kartesischen Produkts der Wertebereiche der Features, d. h. $x_i \in \mathcal{X} = \mathcal{W}(f_1) \times \cdots \times \mathcal{W}(f_F)$. Die Ausgabedaten y_i einer Dateninstanz repräsentieren das sogenannte *Label* der jeweiligen Entität. Jedes Label y_i kann ein Element aus einer Menge möglicher Labels $\mathcal{Y}$ sein. In einer Klassifikation ist $\mathcal{Y}$ eine Menge diskreter Klassen, während es in einer Regression numerische Werte sind [HKP12].

In den Trainingsdaten liegen die Werte sowohl der Eingabedaten x_i als auch der Ausgabedaten y_i für jede der n Dateninstanzen $(x_i, y_i) \in \mathcal{D}$ bereits vor. Lernalgorithmen analysieren diese Trainingsdaten, um eine Funktion $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ zu lernen. Diese Funktion $\mathcal{M}$ beschreibt die in den Daten erkannten Muster oder Zusammenhänge zwischen den Eingabedaten $x_i \in \mathcal{X}$ und den Ausgabedaten $y_i \in \mathcal{Y}$. Je nach Verfahren und Lernalgorithmus wird die Funktion $\mathcal{M}$ entweder in einer symbolischen Repräsentationsform wie Entscheidungsbäume oder in einer nicht-symbolischen Form wie neuronale Netze erzeugt [HKP12; Mit97]. Diese symbolische oder nicht-symbolische Repräsentationsform stellt

das ML-Modell dar. Dieses Modell wird anschließend auf neue Daten angewendet, bei denen lediglich die Eingabedaten x_i bekannt sind. Somit kann für jede der neuen Dateninstanzen mit der im ML-Modell repräsentierten Funktion $\mathcal{M}$ das entsprechende Label y_i vorhergesagt werden.

In einer Klassifikation kann zudem noch zwischen einer binären Klassifikation, einer Multi-Klassen-Klassifikation und einer Multi-Label-Klassifikation unterschieden werden [MRT18]. Die Unterscheidungsmerkmale sind, wie viele diskrete Klassen in $\mathcal{Y}$ vorhanden sind und wie viele Klassen für jede Dateninstanz in ihren Ausgabadeten y_i vorliegen bzw. vorhergesagt werden. In einer *binären Klassifikation* besteht $\mathcal{Y}$ aus genau zwei Klassenlabels, die häufig als positive und negative Klasse bezeichnet werden. Jede Dateninstanz (x_i, y_i) wird dabei in y_i genau einer dieser beiden Klassen zugeordnet. Eine *Multi-Klassen-Klassifikation* bezeichnet Lernaufgaben mit mehr als zwei Klassen, d. h. $\mathcal{Y}$ besteht aus $|\mathcal{Y}| > 2$ Klassenlabels. Hier wird jede Dateninstanz in ihrem y_i ebenfalls nur genau einer dieser Klassenlabels zugeordnet, d. h. $y_i \in \mathcal{Y}$. Der Unterschied zu einer *Multi-Label-Klassifikation* besteht darin, dass jeder Instanz in y_i sogar eine Menge mehrerer Labels zugeordnet werden kann, d. h. $y_i \subseteq \mathcal{Y}$. Die Anzahl der Labels in y_i kann dabei zwischen jeder Dateninstanz variieren. Im Fall der Multi-Label Klassifikation bildet die oben genannte Funktion $\mathcal{M}$ die Eingabedaten x_i folglich auf die Potenzmenge von $\mathcal{Y}$ ab, d. h. $\mathcal{M} : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$.

2.2 Arten formal repräsentierten Domänenwissens

Domänenwissen beschreibt Wissen über Zusammenhänge bzgl. der Entitäten und Ereignisse sowie der Prozesse der jeweiligen Anwendungsdomäne, z. B. Wissen über Produkte und Produktionsprozesse eines Unternehmens. Es kann generell in explizites und implizites Wissen eingeteilt werden [DJR+20; Leh21; Pol09; Row07]. Explizites Wissen ist sprachlich vermittelbar und kann daher leicht wieder- und weitergegeben werden. Somit ist es möglich, explizites Wissen in strukturierter Form zu speichern. Implizites Wissen ist hingegen nicht gänzlich sprachlich beschreib- und vermittelbar. Hierunter fallen z. B. persönliche Erfahrungswerte, Vorstellungen oder eingespielte Handlungsroutinen der Domänenexperten, welche diese aber nicht explizit begründen können. Implizites Wissen liegt folglich in keiner digital erfassten Struktur vor.

Tabelle 2.2: Wesentliche Unterschiede zwischen formal repräsentiertem Domänenwissen und informellem Wissen der Domänenexperten.

Formal repräsentiertes Domänenwissen	liegt in mathematisch-logischer, **digital erfasster Struktur** vor; immer **explizites** Wissen; von Domänenexperten zuvor erfasstes **Kausalitätswissen**
Informelles Domänenwissen	**Erfahrungswissen** der Domänenexperten; liegt **nicht** in **digital** erfasster Struktur vor; meistens **implizites**, selten explizites Wissen

Als weiteres Unterscheidungsmerkmal kann Domänenwissen in einer formal repräsentierten Form vorliegen oder informelles Wissen der Domänenexperten sein. Tabelle 2.2 fasst die wesentlichen Unterschiede zusammen. *Formal repräsentiertes Domänenwissen* liegt in einer meist mathematisch-logischen oder digital erfassten Struktur vor. Dies können z. B. Regeln [GA11], semantische Netze [Rei10] oder Simulationsmodelle [BZBP13] sein, die bestimmte Sachverhalte oder Zusammenhänge in einer Anwendungsdomäne beschreiben oder formalisieren. Diese Art des Domänenwissens ist immer explizit und beschreibt meist Kausalitätswissen [YQL11], das häufig von mehreren Domänenexperten im Vorfeld erfasst und formal definiert wurde. *Informelles Wissen der Domänenexperten* ist deren Erfahrungswissen, das nicht in einer formalen oder digitalen Struktur vorliegt. In den meisten Fällen handelt es sich um implizites Wissen. In einzelnen Fällen kann es explizites Wissen sein, wenn es prinzipiell sprachlich beschreibbar ist, aber noch nicht formal erfasst wurde.

Informelles Wissen wird in Datenanalyseprozssen einbezogen, indem direkt das Erfahrungswissen der Domänenexperten zu bestimmten Problemen abgefragt wird. Beispielsweise können Domänenexperten gefragt werden, welche Features f_j im Datensatz $\mathcal{D}$ für eine konkrete Analyse relevant sind. Ein weiteres häufig auftretendes Beispiel ist, dass Domänenexperten die Labels y_i einzelner Dateninstanzen der Trainingsdaten festlegen. Allerdings ist der Aufwand für Domänenexperten, ihr informellen Wissen auf diese Weise einzubringen, in den meisten Fällen unangemessen hoch. Zum Beispiel müssen sie die Labels y_i einzeln für jede Dateninstanz im Datensatz $\mathcal{D}$ festlegen, was bei einer relativ großen Datenmenge einen immensen Aufwand darstellt [RBE+20; SRYD14]. Daher liegt der Fokus dieser Habilitationsschrift auf Ansätzen, die

formal repräsentiertes Domänenwissen in Datenanalyseprozessen verwenden. Die Nutzung formal repräsentierten Domänenwissens hat folgende Vorteile:

- Formal repräsentiertes Domänenwissen liegt in der Anwendungsdomäne bereits vor. Daher müssen Domänenexperten selbst nicht mehr direkt involviert werden, um das Domänenwissen in Datenanalyseprozessen einzubeziehen, sodass für sie kein zusätzlicher Aufwand entsteht.
- Formal repräsentiertes Domänenwissen wurde meistens von mehreren Domänenexperten erfasst und über einen längeren Zeitraum gepflegt. Es ist daher i. d. R. umfassender als das informelle Domänenwissen einzelner Domänenexperten, sodass es potenziell auch die komplexen Datencharakteristika in industriellen Anwendungsfällen besser adressieren kann.

Im Folgenden werden die für Datenanalyseprozesse wichtigsten und daher in dieser Habilitationsschrift primär untersuchten Arten formal repräsentierten Domänenwissens beschrieben. Abbildung 2.2 stellt für jede dieser Arten dar, in welcher formalen Struktur das Domänenwissen abgebildet wird.

Fallbasis in Expertensystemen: Expertensysteme speichern in einer Wissensbasis die Kenntnisse von einem oder mehreren Domänenexperten über die in einer Anwendungsdomäne häufig auftretenden Probleme und entsprechende Maßnahmen zur Problemlösung [Pup91; SS09a]. Expertensysteme können von anderen Nutzern dazu verwendet werden, um aus der Wissensbasis Handlungsempfehlungen zur Lösung konkreter Probleme abzuleiten.

Fallbasierte Systeme sind eine mögliche Grundlage zur Umsetzung von Expertensystemen [AP94]. Die Wissensbasis wird hierbei durch eine sog. Fallbasis oder Falldatenbank repräsentiert, in der bereits gelöste Probleme mittels einer Problembeschreibung sowie einer Dokumentation einer erfolgreichen Problemlösung hinterlegt sind. Zur Lösung eines neuen Problems sucht das fallbasierte System in seiner Fallbasis nach einem möglichst ähnlichen und zuvor schon gelösten Problem. Die Lösung dieses ähnlichen Problems aus der Fallbasis wird anschließend auf das neue Problem angepasst und angewendet. Dieses Vorgehen wird auch fallbasiertes Schließen bezeichnet [Ric03].

Ein gängiges Beispiel für eine Wissenbasis eines fallbasierten Systems in produzierenden Unternehmen ist die Falldatenbank, die mittels der in der DIN-

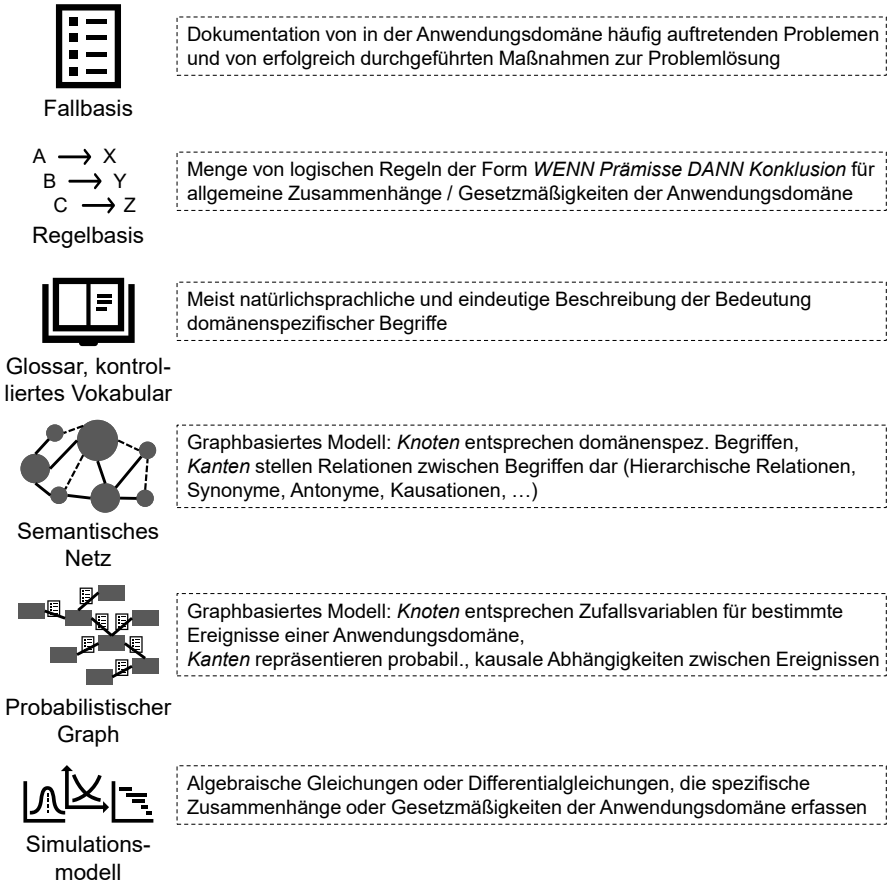


Abbildung 2.2: Im Rahmen dieser Habilitationsschrift primär untersuchte Arten formal repräsentierten Domänenwissens sowie die jeweils genutzte formale Struktur zur Abbildung des Domänenwissens.

Norm EN 60812 beschriebenen Fehlzustandsart- und -auswirkungsanalyse (engl. *Failure Mode and Effects Analysis – FMEA*) erzeugt wird [DIN06; Ise06]. Die FMEA ist eine systematische Analysetechnik des Qualitätsmanagements, die i. d. R. von erfahrenen Ingenieuren bereits früh in der Produktentwicklungsphase oder der Produktionsplanungsphase des Produktlebenszyklus durchgeführt wird und für die nachfolgenden Phasen die größten Fehlerrisiken vorab vermeiden soll [DIN06; Ise06]. So erfassen Ingenieure bspw. bereits

vor der Produktnutzung oder vor der Produktion mögliche Produkt- oder Produktionsfehler sowie passende Maßnahmen zur Fehlerbehebung oder Fehlervermeidung [LLL13]. Die Ergebnisse einer FMEA werden über strukturierte Arbeitsblätter erfasst, in denen die Fehlerart, die dazugehörige Fehlerursache, der Fehlereffekt auf die Produktion oder Produktnutzung sowie mögliche Maßnahmen zur Fehlererkennung, Fehlervermeidung und Fehlerbehebung beschrieben werden [DIN06]. Weiterhin kann die qualitative FMEA durch eine quantitative Analyse zur Wichtigkeit eines Fehlers ergänzt werden (engl. *Failure Mode and Effects and Criticality Analysis – FMECA*) [DIN06; Ise06].

Regelbasis in Expertensystemen: Eine Alternative für die Wissensbasis in Expertensystemen ist eine Regelbasis, die durch ein regelbasiertes System verwaltet wird [GA11; Ros03]. Die Regelbasis besteht dabei aus einer Menge von meist logischen Regeln in der Form *WENN Prämisse DANN Konklusion*. Eine Regel beschreibt typischerweise, dass unter spezifischen Vorbedingungen (Prämisse) ein bestimmtes Ereignis oder ein Effekt (Konklusion) eintritt. So kann bspw. für einen Prüfstand in einer Produktionslinie, der eine Menge von Sensorsignalen zur Überprüfung der produzierten Produkte erfasst, eine Regel zur Fehlererkennung spezifiziert werden [HRKM18]. Die Prämisse beschreibt dabei den Fall, dass die Werte einzelner Sensorsignale ober- oder unterhalb eines Toleranzbandes dieser Signale liegen. Tritt diese Prämisse ein, indiziert die Konklusion der Regel, dass ein Qualitätsproblem des Produkts vermutet werden sollte. Während die Einträge einer Fallbasis in fallbasierten Systemen häufig vorkommende Beispiele konkreter Probleme beschreiben, stellen die Regeln eines regelbasierten Systems eher allgemeine Sachverhalte, Zusammenhänge oder Gesetzmäßigkeiten der jeweiligen Anwendungsdomäne dar. Die Regeln können dabei hinsichtlich ihrer Ausdrucksmächtigkeit bspw. in Form von Ausdrücken der Aussagenlogik, der Prädikatenlogik [BE05] oder der Beschreibungslogik [BCM+07] spezifiziert sein.

Glossare und kontrollierte Vokabulare: Ein Glossar enthält eine Liste relevanter Begriffe einer Anwendungsdomäne sowie eine Beschreibung der Bedeutung dieser Begriffe oder auch mögliche Übersetzungen in andere Sprachen [ANSI/NISO10]. Die Begriffe werden dabei meist natürlichsprachlich

beschrieben. Dabei liegt der Fokus primär auf einer Erklärung und Abgrenzung einzelner Begriffe, sodass ein Glossar in einer Anwendungsdomäne für ein klares Begriffsverständnis sorgt. Ebenso werden durch die Beschreibung der Begriffe auch die wesentlichen Eigenschaften der durch die Begriffe bezeichneten Dinge und Entitäten der Anwendungsdomäne festgelegt.

In einem kontrollierten Vokabular werden die Begriffe eindeutig beschrieben und definiert, sodass keine Homonyme auftreten. Damit werden durch eine Disambiguierung Mehrdeutigkeiten der Begriffe vermieden. In einigen kontrollierten Vokabularen treten sogar keine Synonyme auf. Glossare und insbesondere kontrollierte Vokabulare ermöglichen u. a. eine Verschlagwortung und damit begriffliche Indexierung anderer Dinge einer Anwendungsdomäne, z. B. auch von Daten und Informationen [EGG+21a; WWR18; ZRKM21].

Semantische Netze: Ein semantisches Netz ist ein graphbasiertes Modell zur Wissensrepräsentation, das die relevanten Begriffe einer Anwendungsdomäne sowie die Beziehungen zwischen diesen Begriffen formalisiert und strukturiert [Rei10; Sow91]. Die Knoten des graphbasierten Modells repräsentieren die domänenspezifischen Begriffe, während die Kanten Beziehungen bzw. Relationen zwischen den Begriffen darstellen. Im Allgemeinen können in einem semantischen Netz folgende Arten von Relationen definiert werden [Sow91]:

- *Hierarchische Relation zwischen Ober- und Unterbegriffen:* Der Unterbegriff ist eine Spezialisierung des Oberbegriffs und erbt von diesem sämtliche Eigenschaften und alle weiteren Relationen. Beispielsweise sind 'Verbrennungsmotor' und 'Elektromotor' Unterbegriffe des Oberbegriffs 'Motor'. Die Relation ist transitiv, sodass ein Unterbegriff wiederum ein Oberbegriff eines noch spezifischeren Unterbegriffs sein kann, z. B. sind 'Ottomotor' und 'Dieselmotor' Unterbegriffe von 'Verbrennungsmotor'.
- *Hierarchische Instanzrelation:* Diese Relation verbindet eine konkrete Instanz mit deren abstrakten Klasse. Instanzen unterscheiden sich insofern von Unterbegriffen, dass Instanzen konkrete Individuen repräsentieren, während Unterbegriffe abstrakte Klassen der Anwendungsdomäne beschreiben. Der konkrete Motor im Fahrzeug des Autors dieser Habilitationsschrift ist zum Zeitpunkt der Veröffentlichung der Schrift bspw. eine Instanz der abstrakten Klasse 'Elektromotor'.

- *Hierarchische partitative Relation bzw. Aggregation:* Diese Relation beschreibt eine *ist-Teil-von*-Beziehung zwischen Begriffen. Beispielsweise besteht ein Verbrennungsmotor u. a. aus mehreren Zylindern, während ein Elektromotor u. a. aus einem statischen Magnet und einem drehbaren Elektromagnet besteht.
- *Synonymie bzw. Äquivalenzrelation:* Diese nicht-hierarchische Relation zwischen zwei Begriffen spezifiziert, dass die miteinander verknüpften Begriffe bedeutungsgleich sind. So haben bspw. 'Ottomotor' und 'Benzinmotor' grundsätzlich die gleiche Bedeutung.
- *Antonymie:* Die mit dieser Relation verknüpften Begriffe haben eine gegensätzliche Bedeutung. So sind 'laufend' und 'ausgeschaltet' gegensätzliche Begriffe hinsichtlich des Zustands eines Verbrennungsmotors.
- *Kausation:* Diese Relation ordnet Begriffe zueinander, die Ereignisse und Zustände beschreiben. Dabei spezifiziert die Kausation, dass ein bestimmtes Ereignis einen Zustand oder ein anderes Ereignis verursacht. So kann bspw. das Ereignis 'Anlassen' einen Verbrennungsmotor vom Zustand 'ausgeschaltet' in den Zustand 'laufend' überführen.

Zudem können Begriffe über *Eigenschaften* näher beschrieben werden. So können bestimmte Typen von Motoren bspw. über Leistungskennzahlen wie die maximale PS-Leistung oder das maximale Drehmoment beschrieben werden.

Verschiedene Arten von semantischen Netzen schränken die Menge der zulässigen Relationen ein [Rei10; Sow91]. Nachfolgend werden mit Taxonomien, Thesauri, Ontologien und Wissensgraphen die wichtigsten im Rahmen dieser Habilitationsschrift untersuchten Arten von semantischen Netzen beschrieben. Die Definition von Eigenschaften der Begriffe ist typischerweise bei allen semantischen Netzen erlaubt. Abbildung 2.3 fasst für jede Art von semantischen Netzen zusammen, welche Relationen zusätzlich erlaubt sind.

In einer *Taxonomie* sind neben Eigenschaften von Begriffen nur hierarchische Relationen zwischen Ober- und Unterbegriffen zulässig. Taxonomien werden z. B. in der Biologie zur hierarchischen Kategorisierung von Lebewesen verwendet. In produzierenden Unternehmen werden sie u. a. genutzt, um Produkte und Produktvarianten des Unternehmens in einer Produkthierachie zu gruppieren und zu strukturieren [AK04; HRT+23; SRYD14]. Weiterhin existieren

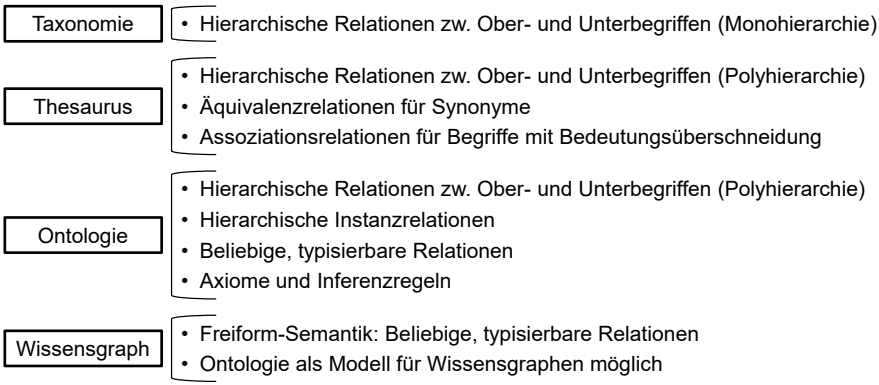


Abbildung 2.3: Wichtigste Arten semantischer Netze sowie die jeweils zugelassenen Relationen zwischen domänenspezifischen Begriffen.

einige Standards zur unternehmensübergreifenden Produktkategorisierung, die wie eine Taxonomie aufgebaut sind [HLS07]. Ein bekannter Standard ist ECLASS[1], der über verschiedene Branchen hinweg die Kategorisierung von Produkten und Dienstleistungen sowie deren semantische Beschreibung über Merkmale ermöglicht. Dieser Standard bietet eine Taxonomie über vier Hierarchieebenen von Ober- und Unterbegriffen. Dies sind Sachgebiete bzw. Industriebranchen sowie Hauptgruppen, Gruppen und Untergruppen der Produkte und Dienstleistungen, die innerhalb der Branchen vertrieben werden.

Thesauri sind semantische Netze, mit denen neben Eigenschaften und Relationen zwischen Ober- und Unterbegriffen auch Begriffe über Äquivalenzrelationen als Synonyme definiert werden können. Zusätzlich unterstützen Thesauri mit Assoziationsrelationen eine verwandte Art von Relationen, die zwei Begriffe miteinander verknüpft, die nicht vollständig semantisch äquivalent sind, aber eine Überschneidung in ihrer Bedeutung haben. Weiterhin können in Thesauri im Gegensatz zu Taxonomien auch Polyhierarchien definiert werden. In diesem Fall kann ein Unterbegriff mehr als einen Oberbegriff haben.

Ontologien unterstützen explizite Konstrukte für hierarchische Ober- und Unterbegriffsrelationen sowie für hierarchische Instanzrelationen zwischen Klassen und ihren Instanzen [SS09b]. Anders als die anderen Arten semanti-

[1] ECLASS: https://eclass.eu/

scher Netze erlauben Ontologien damit, explizit Instanzrelationen von anderen hierarchischen Relationen zu unterscheiden. Darüber hinaus können in Ontologien beliebig weitere Relationen zwischen Begriffen bzw. Klassen spezifiziert werden. Jede Klasse und jede Relation kann dabei typisiert werden, um die genaue Semantik der Klasse oder der Relation zu beschreiben. So können Relationen über ihre Typisierung bspw. einer der weiteren oben gelisteten Arten von Relationen zugeordnet werden, also z. B. Aggregation, Synonymie, Antonymie oder Kausation. Es sind darüber hinaus aber auch weitere anwendungsspezifische Typen von Relationen wie auch von Klassen denkbar, sodass Ontologien insgesamt eine sehr hohe semantische Ausdrucksmächtigkeit besitzen. Ontologien erlauben zudem die Spezifikation von zwei Arten von Regeln. So können mit Axiomen, die immer wahr sein müssen, Regeln für die Integrität konkreter Instanzen einer Ontologie definiert werden. Die zweite Art von Regeln sind Inferenzregeln, mit denen aus einer konkreten Instanz einer Ontologie, d. h. aus der entsprechenden Fakten- bzw. Wissensbasis, neue Fakten bzw. neues Wissen abgeleitet werden können.

Die Automation Markup Language (AutomationML) ist eine in produzierenden Unternehmen verbreitete XML-basierte Beschreibungssprache, die einen Datenaustausch zwischen unterschiedlichen Software-Lösungen zur Anlagenplanung ermöglicht [Dra10]. Mit AutomationML sowie dem übergeordneten Datenformat Computer Aided Engineering Exchange (CAEX) kann in einer Klassen- und Objektstruktur eine Ontologie mit den meisten der in semantischen Netzen verwendbaren Relationstypen definiert werden, um somit die Topologie und Struktur von Produktionsanlagen und Produkten abzubilden.

Wissensgraphen bieten ebenso Möglichkeiten, um in einem graphbasierten Modell die für eine Anwendungsdomäne relevanten Begriffe, Entitäten oder Ereignisse sowie Beziehungen zwischen diesen zu spezifizieren. Dabei werden Wissensgraphen in der Literatur nicht immer einheitlich von Ontologien oder anderen semantischen Netzen abgegrenzt [EW16]. Wissensgraphen unterstützen eine Freiform-Semantik. Dies bedeutet, dass für Wissensgraphen nicht zwingend ein festes Schema definiert werden muss, sondern beliebige domänenspezifische Begriffe und Beziehungen spezifiziert werden können. Die semantische Interpretation der Begriffe und Relationen ist in diesem Fall der Freiform-Semantik aber nutzer- bzw. anwendungsabhängig. Soll die Semantik

möglichst eindeutig definiert werden, kann alternativ eine Ontologie als Modell für Wissensgraphen definiert werden. Ein Wissensgraph ist in diesem Fall eine konkrete Instanz der Ontologie. Prinzipiell können Begriffe und Beziehungen in Wissensgraphen auch über die semantisch weniger aussdrucksstarken Thesauri oder Taxonomien typisiert werden.

Probabilistische Graphen: Weitere graphbasierte Ansätze zur Wissensrepräsentation sind probabilistische Graphen [CGH97; KF09]. Gängige Beispiele sind Markov-Modelle [Fin03] und Bayessche Netze [JN07]. Dabei stellen die Knoten dieser probabilistischen Graphen Zufallsvariablen dar. Dies können z. B. bestimmte Ereignisse oder Zustände in der jeweiligen Anwendungsdomäne sein, die mit einer bestimmten Wahrscheinlichkeit oder Häufigkeit auftreten. In einer Produktionslinie können die Knoten und Zufallsvariablen bspw. bestimmte Fehlerarten, Fehlereffekte oder Fehlerursachen repräsentieren [Sad20]. Die in einem probabilistischen Graphen modellierten Kanten beschreiben bedingte Abhängigkeiten zwischen den Knoten und deren Zufallsvariablen. Häufig werden damit kausale Relationen zwischen den durch die Zufallsvariablen repräsentierten Ereignissen oder Zuständen definiert, sodass die Kanten den Kausationsrelationen von semantischen Netzen nahe kommen. Im Gegensatz zu allgemeinen semantischen Netzen können bei den Knoten bzw. Kanten in probabilistischen Graphen bedingte Wahrscheinlichkeitsverteilungen zwischen den verknüpften Zufallsvariablen festgelegt werden. Damit wird insbesondere festgelegt, mit welcher Wahrscheinlichkeit ein bestimmtes Ereignis einen konkreten Zustand bzw. ein anderes Ereignis verursacht. Ebenso kann damit unsicheres Wissen über kausale Relationen in einer Anwendungsdomäne beschrieben werden, wenn nicht alle möglichen Kausalitäten sicher erfasst werden können. Ist die Wahrscheinlichkeit einer einzelnen Zufallsvariable bekannt, können z. B. über die Bayessche Inferenz [BS00] auch die Wahrscheinlichkeiten aller im Graphen abhängigen Zufallsvariablen berechnet werden.

Entsprechen wie im oben genannten Beispiel die Knoten Fehlerarten, Fehlereffekte und Fehlerursachen, so sagen die bedingten Wahrscheinlichkeitsverteilungen an den Kanten zwischen diesen Knoten aus, mit welcher Wahrscheinlichkeit ein Fehler einen bestimmten Fehlereffekt zur Folge hat oder welche Ursache mit welcher Wahrscheinlichkeit zu einem konkreten Fehler

führt [Sad20]. Tritt in einer Produktionslinie ein konkreter Fehler auf, ist die Wahrscheinlichkeit der Zufallsvariable bzw. des Knotens der entsprechenden Fehlerart 100 %. Über den probabilistischen Graphen und die darin modellierten bedingten Wahrscheinlichkeitsverteilungen können dann mittels Inferenz die wahrscheinlichsten Fehlereffekte sowie Fehlerursachen bestimmt werden.

Mathematische Simulationsmodelle: Simulationen imitieren das Verhalten bzw. die Änderungen des Zustands anderer Experimente, Prozesse oder Systeme [BZBP13; Har96]. Das Verhalten wird meist über mathematische, physikalische Simulationsmodelle beschrieben und abstrahiert. Diese Simulationsmodelle werden wiederum in computerbasierten Simulationen umgesetzt und berechnet [Hum90]. Eine wesentliche Motivation für computerbasierte Simulationen ist, dass die Durchführung der alternativen realen Experimente häufig zu zeitaufwendig oder zu kostenintensiv ist. Dies trifft in industriellen Anwendungsfällen z. B. auf Produkttests wie einen Crash-Test zu, die zunehmend virtuell am Computer berechnet werden [Hau81; VK11; ZRKM20].

Die mathematischen Simulationsmodelle werden i. d. R. von Domänenexperten spezifiziert. Diese Experten erfassen in den Modellen die in der Anwendungsdomäne bekannten Zusammenhänge und Gesetzmäßigkeiten bestimmter anwendungsspezifischer Problemstellungen. Relativ einfache Zusammenhänge können über analytische Modelle, insbesondere über algebraische Gleichungen oder Gleichungssysteme, modelliert werden. Diese analytischen Modelle können i. d. R. direkt von Computern berechnet werden. Die meisten mathematischen Simulationsmodelle bilden aber komplexere Zusammenhänge über ein System von Differenzialgleichungen ab [BDH11]. Differenzialgleichungen besitzen häufig keine direkte analytische Lösung [Hum90], insbesondere da sie das typischerweise zeitabhängige Verhalten des zu simulierenden Systems auf zeitkontinuierliche Weise beschreiben. Sie können daher nicht direkt von Computern berechnet werden und müssen stattdessen vorher mit numerischen Methoden diskretisiert und ggf. auch approximiert werden. Ein gängiges Beispiel für eine numerische Diskretisierungsmethode für Differenzialgleichungen ist die Finite-Elemente-Methode (FEM) [GRS07; ZTZ13].

2.3 Vorgehensmodelle für Datenanalyseprozesse

Verschiedene Konsortien schlagen Vorgehensmodelle vor, die konkrete Schritte mit definierten Aufgaben vorgeben, um Datenanalysen umzusetzen. Die gängigsten sind der *Cross-Industry Standard Process for Data Mining (CRISP-DM)* [CCK+00], der *Kowledge Discovery in Databases (KDD) Process* [FPS96], der *Microsoft Team Data Science Process* (TDSP)[1] und der Prozess *Sample, Explore, Modify, Model, and Assess* (SEMMA)[2] des SAS-Instituts. Die Schritte dieser Vorgehensmodelle unterscheiden sich primär nur in ihrer Terminologie und in Detailaspekten. Sie besitzen ausreichend Gemeinsamkeiten, um aus ihnen das in Abschnitt 2.3.1 vorgestelle generische Vorgehensmodell zur Umsetzung von Datenanalysen ableiten zu können. Schließlich wird in Abschnitt 2.3.2 diskutiert, wie in den einzelnen Schritten dieses generischen Vorgehensmodells in der praktischen Umsetzung von Datenanalyseprozessen Domänenwissen bzw. das Wissen von Domänenexperten einbezogen wird. Die Schritte des generischen Vorgehensmodells sowie die Diskussion in diesem Abschnitt dienen in den folgenden Kapiteln als Bezugsrahmen zur Einordnung der in dieser Habilitationsschrift behandelten Problemstellungen und Lösungsansätze.

2.3.1 Generisches Vorgehensmodell

Das generische Vorgehensmodell beinhaltet insgesamt die sechs nachfolgend erläuterten Schritte. Abbildung 2.4 stellt diese Schritte sowie die wichtigsten jeweils durchzuführenden Aufgaben dar. Zu beachten ist, dass die Schritte nicht zwingend strikt sequenziell durchgeführt werden müssen, sondern dass in praktischen Umsetzungen des Vorgehensmodells auch Iterationen zwischen einzelnen Schritten sinnvoll sein können [CCK+00; FPS96]. Tabelle 2.3 fasst zusammen, inwieweit die sechs Schritte des generischen Vorgehensmodells innerhalb der Vorgehensmodelle CRISP-DM, KDD, Microsoft TDSP und SEMMA vorkommen. Zur Illustration der Schritte wird nachfolgend an geeigneten Stellen auf ein Beispiel zur Qualitätskontrolle von Produkten innerhalb einer aus mehreren Maschinen bestehenden Produktionslinie verwiesen [WSR+20].

[1] Microsoft TDSP: `https://docs.microsoft.com/de-de/azure/architecture/data-science-process/overview`

[2] SEMMA: `https://documentation.sas.com/doc/en/emref/14.3/n061bzurmej4j3n1jnj8bbjjm1a2.htm`

Abbildung 2.4: Schritte des generischen Vorgehensmodells zur Umsetzung von Datenanalysen sowie zentrale Aufgaben in den sechs Schritten.

Verständnis über Problemstellung herstellen: In diesem ersten Schritt wird die fachliche Problemstellung untersucht, um Ziele und Anforderungen an die zu entwickelnde Datenanalyselösung aus Sicht dieser domänenspezifischen Problemstellung abzuleiten. Darüber hinaus werden sämtliche Ressourcen erfasst, die für das Projekt genutzt werden können, z. B. die zur Verfügung stehenden Datenquellen, Software und Hardware, aber auch Personal zur Umsetzung des Projekts. Schließlich wird ein Projektplan erstellt, der festlegt, wie genau und mit welchen Ressourcen die nachfolgenden Schritte des Vorgehensmodells durchgeführt werden sollen. Im Beispiel der Qualitätskontrolle von Produkten in einer Produktionslinie [GSH+20; WSR+20] werden als Ziele festgelegt, möglichst alle Maschinenfehler zu erkennen und frühzeitig sowie korrekt vorherzusagen, ob ein Maschinenfehler zu einem Qualitätsproblem

des Produkts führt. Die zur Verfügung stehenden Datenquellen beinhalten insbesondere sämtliche Sensordaten der Maschinen in der Produktionslinie.

In den Vorgehensmodellen CRISP-DM und Microsoft TDSP entspricht dieser erste Schritt dem '*Business Understanding*'. Der KDD- und der SEMMA-Prozess definieren keinen expliziten Schritt, betonen aber, dass vor der Durchführung des jeweiligen Vorgehensmodells ein Verständnis über die mit der Datenanalyse zu lösende fachliche Problemstellung geschaffen werden muss [FPS96].

Datenakquise und Datenexploration: In diesem Schritt werden die Daten aus den zur Verfügung stehenden Datenquellen akquiriert und gesammelt, um sie initial zu untersuchen und besser verstehen zu können. Die erfassten und gesammelten Daten werden auf einer Meta-Ebene näher beschrieben, z. B. in welchen Datenformaten sie vorliegen und was einzelne Datenfelder bzw. Features semantisch bedeuten. Im Beispiel der Qualitätskontrolle in einer Produktionslinie [GSH+20; WSR+20] können Sensordaten der Maschinen erfasst sowie auf einer Meta-Ebene die Bedeutungen der einzelnen Sensoren der Maschinen beschrieben werden. Weiterhin kann auf dieser Meta-Ebene zusammengefasst werden, wie viele Dateninstanzen zur Verfügung stehen und wie viele reale Entitäten bzw. Ereignisse, z. B. Produkte und Maschinenfehler, durch diese Dateninstanzen repräsentiert werden.

Die gesammelten Daten werden mit primär deskriptiven und diagnostischen Analyseverfahren initial untersucht und exploriert, um bspw. erste Korrelationen in den Daten zu erfassen. Damit können u. a. Hypothesen über Zusammenhänge in den Daten aufgestellt werden, die in den späteren Schritten des Vorgehensmodells näher untersucht werden. Ebenso können die Daten hinsichtlich verschiedener Kriterien der Datenqualität [WS96] untersucht werden, z. B. ob die Daten korrekt und vollständig sind. Erkannte Probleme mit der Datenqualität, z. B. dass einige Dateninstanzen fehlerhafte Sensorwerte enthalten, werden in diesem Schritt genau dokumentiert, damit sie in den nachfolgenden Schritten berücksichtigt werden können.

In CRISP-DM wird dieser Schritt '*Data Understanding*' genannt. Microsoft TDSP beschreibt mit '*Data Acquisition and Understanding*' einen sehr ähnlichen Schritt, während der KDD-Prozess im Schritt '*Selection*' primär die Akquise sowie die Auswahl passender Daten aus mehreren Datenquellen betont. Der

Tabelle 2.3: Ausprägungen und Bezeichnungen der Schritte des generischen Vorgehensmodells zur Umsetzung von Datenanalysen in CRISP-DM, KDD, Microsoft TDSP und SEMMA.

Schritt	**CRISP-DM**	**KDD**	**Micros. TDSP**	**SEMMA**
Verständnis über Problemstellung herstellen	'Business Understanding'	vor Durchführung des Vorgehensmodells	'Business Understanding'	vor Durchführung des Vorgehensmodells
Datenakquise und Datenexploration	'Data Understanding'	'Selection' (primär Datenakquise)	'Data Acquisition and Understanding'	zwei Schritte: 'Sample' und 'Explore'
Datenvorbereitung	'Data Preparation'	zwei Schritte: 'Cleansing' und 'Transformation'	Teil von 'Data Acquisition and Understanding'	'Modify'
Datenanalyse und Modellbildung	'Modeling'	'Data Mining'	'Modeling'	'Model'
Evaluation der Analyseergebnisse	'Evaluation'	'Interpretation'	'Customer Acceptance'	'Assess'
Anwendung der Analyseergebnisse	'Deployment'	Teil von 'Interpretation'	'Deployment'	Teil von 'Assess'

Prozess SEMMA betrachtet hier zwei Schritte, die aber zusammen die gleichen wie die oben beschriebenen Aufgaben beinhalten: '*Sample*' für die Datenakquise und '*Explore*' für die initiale Datenexploration.

Datenvorbereitung: In diesem Schritt erfolgt die Vorbereitung der zuvor erfassten Daten, sodass diese für die im nächsten Schritt durchzuführende Datenanalyse und Modellbildung angepasst werden. Dazu werden auch die Erkenntnisse aus der im Schritt zuvor durchgeführten Datenexploration ausgenutzt. So können bspw. Sampling-Verfahren oder Verfahren zur Feature-Selektion auf die Daten angewendet werden, um insbesondere diejenigen Dateninstanzen und Features aus den Daten zu löschen, die laut den Ergebnissen der Datenexploration eine zu schlechte Datenqualität oder eine zu geringe

Korrelation mit Ausgabewerten besitzen. Wenn möglich, wird die Qualität der Daten über entsprechende Verfahren zur Datenbereinigung verbessert, z. B. indem fehlende bzw. fehlerhafte Feature-Werte, wie fehlerhafte Sensorwerte der Maschinen im Beispiel der Qualitätskontrolle, ergänzt bzw. korrigiert werden. Ebenso können die Daten um neue Daten angereichert werden, die zusätzliche Informationen liefern. Hierbei können über Verfahren zum Feature-Engineering oder zur Feature-Konstruktion neue Features erzeugt werden. So können im Anwendungsfall bspw. die Sensorwerte eines Sensortyps, z. B. eines Temperatursensors, über mehrere Maschinen hinweg aggregiert und diese aggregierten Werte als weiteres Feature zur Verfügung gestellt werden. Weiterhin können über Datengeneratoren [FSC+20; GPM+14; RHW21; TRSM23] aus den bisherigen Daten auch zusätzliche Dateninstanzen erzeugt werden.

Falls die Daten aus mehreren einzelnen Datensätzen bestehen, z. B. da sie aus mehreren Quellsystemen stammen, werden diese einzelnen Datensätze noch zusammengeführt und integriert. Somit wird ein einzelner Datensatz $\mathcal{D}$ für die nachfolgende Datenanalyse und Modellbildung erstellt. Im Bedarfsfall werden im Datensatz $\mathcal{D}$ syntaktische Anpassungen des Datenformats durchgeführt, sodass $\mathcal{D}$ bzw. einzelne Features in $\mathcal{D}$ so angepasst werden, dass die im nachfolgenden Schritt verwendeten Analysetools oder -algorithmen mit den Daten umgehen können. Dies schließt bspw. die Kodierung kategorischer Features mittels *One-Hot-Encoding* ein. Der Schritt der Datenvorbereitung liefert neben dem aufbereiteten Datensatz $\mathcal{D}$ eine Dokumentation der angewendeten Datenvorbereitungstechniken, um diese später nachvollziehen zu können.

Dieser Schritt der Datenvorbereitung entspricht in CRISP-DM dem Schritt '*Data Preparation*' sowie im SEMMA-Prozess dem Schritt '*Modify*'. Im KDD-Prozess ist die Datenvorbereitung hingegen in zwei Schritte '*Cleansing*' zur Datenbereinigung und '*Transformation*' für weitere Datentransformationen aufgeteilt. In Microsoft TDSP gibt es keinen dedizierten Schritt zur Datenvorbereitung, die meisten der oben beschriebenen Aufgaben werden aber bereits im Schritt '*Data Acquisition and Understanding*' durchgeführt.

Datenanalyse und Modellbildung: Ziele dieses Schritts sind die Auswahl, die Umsetzung und der Test unterschiedlicher Verfahren und Algorithmen zur Datenanalyse und Modellbildung. Dazu werden abhängig von der im ersten

Schritt definierten Zielsetzung der Datenanalyse und von den Eigenschaften der zuvor erfassten und vorbereiteten Daten eine Menge prinzipiell geeigneter Analyseverfahren ausgewählt. Hier kann bspw. eine Auswahl eines geeigneten Verfahrens des Data Mining getroffen werden, also ob eine Assoziationsregelanalyse, eine Cluster-Analyse, eine Klassifikation oder eine Regression infrage kommt [HKP12]. Je nach gewähltem Verfahren sowie je nach den Eigenschaften der Daten kann zudem die Menge an zu evaluierenden Analysealgorithmen eingeschränkt werden. Im Beispiel der Qualitätskontrolle von Produkten kann die Vorhersage, ob ein produziertes Produkt ein Qualitätsproblem aufweist, als binäre Klassifikation umgesetzt werden. Die positive Klasse bedeutet, dass ein Qualitätsproblem wahrscheinlich ist, während die negative Klasse darauf hindeutet, dass kein Qualitätsproblem vorliegt. Als zu evaluierende Klassifikationsalgorithmen kann in diesem Beispiel je nach Eigenschaften der Daten eine Wahl zwischen Algorithmen für symbolische ML-Modelle wie Entscheidungsbäume oder für nicht-symbolische Modelle wie neuronale Netze getroffen werden [Mit97] (siehe auch Abschnitt 2.1).

Anschließend werden die ausgewählten Analyseverfahren und Algorithmen auf die Daten angewendet und anhand eines zuvor definierten Test-Designs bewertet. Das Test-Design definiert dabei insbesondere, auf welcher Teilmenge der Daten die Algorithmen angewendet sowie die Modelle gebildet werden, auf welcher Teilmenge die Modelle getestet werden und mit welchen Metriken die Tests erfolgen sollen. Dies muss auf die domänenspezifische Zielsetzung der Datenanalyse abgestimmt werden. In überwachten Lernverfahren wie einer Klassifikation kann der Datensatz $\mathcal{D}$ bspw. in disjunkte Teilmengen für Trainingsdaten $\mathcal{D}_{train} \subset \mathcal{D}$ und Testdaten $\mathcal{D}_{test} \subset \mathcal{D}$ aufgeteilt werden. Es ist auch eine Kreuzvalidierung denkbar, bei der in mehreren Iterationen unterschiedliche Teilmengen von Trainings- und Testdaten gebildet werden [WFH11]. Auf den Trainingsdaten $\mathcal{D}_{train}$ wird der Analysealgorithmus angewendet und so ein Modell gebildet. Anschließend wird das Modell auf die Testdaten $\mathcal{D}_{test}$ angewendet und mit geeigneten Metriken bewertet. Die gängigsten Metriken im Beispiel einer Klassifikation sind die Korrektklassifikationsrate (engl. *Accuracy*), die Sensitivität bzw. Wahr-Positiv-Rate (engl. auch *Recall*), die Spezifizität oder Wahr-Negativ-Rate sowie der positive Vorhersagewert (engl. *Precision*) [HKP12]. Im Beispiel der Qualitätskontrolle ist insbesondere die Wahr-Positiv-Rate geeignet. Diese gibt

den Anteil an, wie viele der tatsächlich aufgetretenen Qualitätsprobleme, die in den Testdaten also der positiven Klasse zugeordnet sind, von dem trainierten Klassifikationsmodell auch als positive Klasse vorhergesagt werden. Wenn in dieser Habilitationsschrift an entsprechenden Stellen bzgl. der Bewertung von Klassifikationsmodellen die konkrete Metrik nicht relevant ist, werden die Metriken mit dem Begriff *Vorhersageleistung* der Modelle abstrahiert. Anhand der mit dem Test-Design und den Metriken durchgeführten Bewertung der erstellten Modelle werden einige Kandidatenmodelle ausgewählt, die beim Test am besten abgeschnitten haben. Diese Kandidatenmodelle werden im nächsten Schritt des Vorgehensmodells genauer untersucht.

Dieser zentrale Schritt eines Datenanalyseprozesses wird in allen vier betrachteten Vorgehensmodellen mit weitgehend ähnlichen Aufgaben und Ergebnissen beschrieben, aber teils unterschiedlich benannt. CRISP-DM und Microsoft TDSP nennen ihn beide '*Modeling*', während er im Prozess SEMMA mit '*Model*' und im KDD-Prozess mit '*Data Mining*' bezeichnet wird.

Evaluation der Analyseergebnisse: Die im vorherigen Schritt anhand des Test-Designs ausgewählten Kandidatenmodelle werden in diesem Schritt weitergehend evaluiert, um insbesondere zu überprüfen, inwieweit sie die im ersten Schritt festgelegten domänenspezifischen Ziele erreichen. Dabei werden nicht nur die Testergebnisse aus dem vorherigen Schritt auf die domänenspezifischen Ziele abgebildet, sondern auch noch weitere Anforderungen berücksichtigt. So kann bspw. überprüft werden, welche monetären Kosten bei der Anwendung der Kandidatenmodelle entstehen oder sogar eingespart werden können [HRM20b]. Weiterhin kann untersucht werden, wie viele Daten die Modelle im laufenden Betrieb benötigen und ob diese Datenmenge überhaupt zur Verfügung steht. Die Evaluationsergebnisse werden zudem zum Zwecke der Nachvollziehbarkeit dokumentiert. Wenn kein Modell die domänenspezifischen Ziele erreicht oder die Anforderungen erfüllt, können alle oder einzelne Schritte des Vorgehensmodells iteriert werden und dabei u. a. die domänenspezifischen Ziele und/oder die zu verwendeten Daten geändert werden. Wenn mehrere der Kandidatenmodelle die domänenspezifischen Ziele erreichen und alle Anforderungen erfüllen, muss eines dieser Modelle ausgewählt werden, das im nächsten Schritt im Anwendungsfall eingesetzt wird. Bei komplexen

Modellen, die nicht selbsterklärend sind, kann diese Auswahl durch Ansätze zur Erklärung der Modelle (engl. *Explainable AI*) unterstützt werden [BH21].

Auch dieser Schritt wird in allen Vorgehensmodellen mit teils unterschiedlichen Namen, aber ähnlichen Aufgaben und Ergebnissen beschrieben. So heißt er in CRISP-DM '*Evaluation*' und in SEMMA '*Assess*'. Im KDD-Prozess wird er mit '*Interpretation*' und in Microsoft TDSP mit '*Customer Acceptance*' bezeichnet.

Anwendung der Analyseergebnisse: Schließlich soll das im Schritt zuvor ausgewählte Kandidatenmodell im Anwendungsfall tatsächlich eingesetzt werden. Dazu wird in diesem Schritt des Vorgehensmodells ein Plan zum Deployment des Modells erstellt, der sämtliche Rahmenbedingungen der Modellanwendung beschreibt und damit genau festlegt, wie das Modell in der Praxis eingesetzt werden soll. Im Beispiel der Qualitätskontrolle muss ein Plan festgelegt werden, wie das entsprechende binäre Klassifikationsmodell direkt innerhalb der Produktionslinie eingesetzt werden kann und welche Daten es dabei von welchen Sensoren der Maschinen abrufen muss. Unter Umständen muss sogar die Produktionsumgebung selbst angepasst werden. Wenn zuvor z. B. keine datengetriebene Lösung in dieser Umgebung eingesetzt wurde, muss das Konzept zur Fehlererkennung und Qualitätskontrolle in der Produktionsumgebung auf diese neue datengetriebene Lösung angepasst werden.

Weiterhin werden Pläne zur Überwachung des im Anwendungsfall eingesetzten Modells entworfen, um dessen erfolgreiche Nutzung über einen längeren Zeitraum zu gewährleisten. Beispielsweise müssen die Vorhersagen und die zugrundeliegenden Daten eines Klassifikations- oder Regressionsmodells ständig überwacht werden, um zu überprüfen, ob die Anzahl der fehlerhaften Vorhersagen zunimmt und ob sich im Laufe der Zeit der Inhalt oder die statistische Verteilung der Daten ändert [MRA+12; WHRS19]. Im Bedarfsfall muss das Modell an den neuen Inhalt oder die neue statistische Verteilung der Daten angepasst werden [LHW18]. Schließlich wird noch ein Abschlussbericht für das gesamte Projekt, also für die Umsetzung aller Schritte des Vorgehensmodells erstellt. Dieser Bericht soll genau dokumentieren, was gut und was schlecht gelaufen ist, um die Durchführungen nachfolgender Projekte zur Umsetzung von Datenanalysen zu verbessern.

Sowohl in CRISP-DM als auch im Microsoft TDSP wird dieser Schritt '*Deployment*' bezeichnet. Der KDD-Prozess sieht die Anwendung der Analyseergebnisse als letzten Teil des oben bereits erwähnten Schritts '*Interpretation*', um aus der Datenanalyse Wissen abzuleiten, mit dem die anwendungsspezifische Problemstellung gelöst werden kann. Der SEMMA-Prozess beschreibt analog die Anwendung der Analyseergebnisse kurz als Teil des Schritts '*Assess*'.

2.3.2 Einbeziehung des Wissens von Domänenexperten in Vorgehensmodelle für Datenanalyseprozesse

Die Einbeziehung des Wissens von Domänenexperten spielt in allen betrachteten Vorgehensmodellen zur Umsetzung von Datenanalyseprozessen eine Rolle (siehe bspw. [CCK+00; FPS96]). Allerdings wird in den Beschreibungen der Vorgehensmodelle nicht diskutiert, wie formal repräsentiertes Domänenwissen, das z. B. in den in Abschnitt 2.2 beschriebenen Strukturen vorliegt, in die einzelnen Schritte der Vorgehensmodelle einbezogen werden kann. Stattdessen liegt der Fokus primär auf der Einbeziehung des informellen Wissens von Domänenexperten, also deren Erfahrungswissen, das nicht in einer formalen oder digitalen Struktur vorliegt und in vielen Fällen auch implizites Wissen darstellt. Dies bedeutet, dass Domänenexperten meist selbst in einzelne Schritte und Aufgaben der Vorgehensmodelle einbezogen werden und dabei ihr Wissen zur Bearbeitung der entsprechenden Aufgaben beisteuern [SRM25].

Domänenexperten sind die Hauptakteure im ersten sowie im vorletzten Schritt des in Abschnitt 2.3.1 dargestellten generischen Vorgehensmodells, d. h. bei der Herstellung des Verständnisses über die Problemstellung sowie bei der Evaluation der Analyseergebnisse [CCK+00] (siehe Abbildung 2.5). In diesen Schritten wird ihr informelles Wissen über die domänenspezifische Problemstellung benötigt, um die wesentlichen Aufgaben der Schritte durchführen zu können. Dies betrifft im ersten Schritt des Vorgehensmodells insbesondere die Festlegung der domänenspezifischen Ziele und weiterer Anforderungen an eine Datenanalyselösung. Im vorletzten Schritt zur Evaluation der Analyseergebnisse werden Domänenexperten vor allem benötigt, um zu überprüfen, ob die mit der Datenanalyse erstellten Kandidatenmodelle die zuvor definierten Ziele erreichen und alle Anforderungen erfüllen.

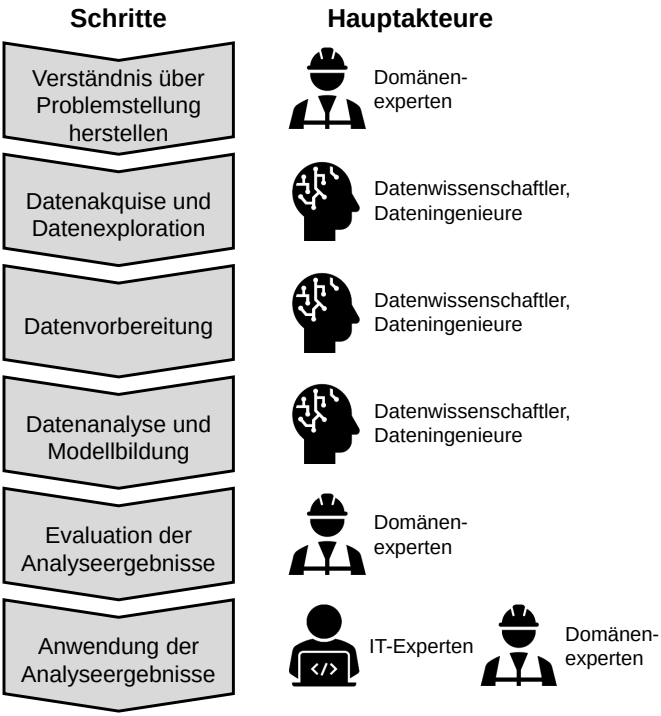


Abbildung 2.5: Zuordnung der wichtigsten Hauptakteure, die ihr jeweiliges Fachwissen in die Aufgaben der einzelnen Schritte des generischen Vorgehensmodells aus Abbildung 2.4 einbeziehen.

Auch bei vielen der Aufgaben im letzten Schritt zur Anwendung der Analyseergebnisse müssen häufig Domänenexperten eingebunden werden. Wenn IT-Experten bspw. einen Plan zum Deployment eines Modells erstellen, müssen Domänenexperten überprüfen, ob dieser Plan in der Praxis umgesetzt werden kann. Hier können z. B. rechtliche oder sicherheitskritische Aspekte eine Rolle spielen. Ebenso müssen Domänenexperten überprüfen, ob für den Einsatz des Modells Anpassungen an der Produktivumgebung notwendig sind und ob diese Anpassungen aus technischen oder betriebswirtschaftlichen Gesichtspunkten durchführbar sind. Dies betrifft bspw. das in Abschnitt 2.3.1 erwähnte Konzept zur Qualitätskontrolle in einer Produktionslinie, das ggf. erst an eine datengetriebene Lösung angepasst werden muss.

Die Hauptakteure der anderen drei Schritte zur Datenakquise und Datenexploration, zur Datenvorbereitung sowie zur Datenanalyse und Modellbildung sind in der Regel Datenwissenschaftler und Dateningenieure (engl. *Data Engineers*). Diese besitzen aber häufig kaum Wissen über die Anwendungsdomäne und über domänenspezifische Problemstellungen. Das Domänenwissen wird aber dennoch für die Umsetzung einzelner Aufgaben in diesen Schritten benötigt [BFSM22; BHFM20; SRYD14]. Daher werden sporadisch in einzelnen Aufgaben dieser drei Schritte des Vorgehensmodells Domänenexperten einbezogen. In der Praxis stellt sich dies häufig so dar, dass für konkrete Aufgaben direkt das informelle Wissen von den Domänenexperten abgefragt wird [VGR20]. So können Domänenexperten bspw. bei der Datenakquise gebeten werden, die semantische Bedeutung bestimmter Daten oder Features zu beschreiben. Ebenso können sie während der initialen Datenexploration bei der Aufstellung der Hypothesen helfen und überprüfen, welche Hypothesen in der domänenspezifischen Problemstellung überhaupt Sinn ergeben [CCK+00]. Im weiteren Verlauf des Vorgehensmodells können Domänenexperten bspw. Features aus dem Datensatz benennen, die entweder relevant oder irrelevant für die jeweilige Problemstellung der Datenanalyse sind [BHTM22; VLT10]. Ebenso können Domänenexperten die Identifikation und Behebung von Problemen bzgl. der Datenqualität unterstützen [BFSM22; BHV+23; KRM19b]. Speziell in Datenanalysen des überwachten Lernens müssen Domänenexperten häufig die Labels einzelner Dateninstanzen festlegen [RBE+20; SRYD14]. Dass Domänenexperten nur sporadisch für solche einzelnen Aufgaben eingesetzt werden, liegt vor allem daran, dass der Aufwand, den sie dabei aufbringen müssen, häufig unangemessen hoch ist. Schließlich wird bei den oben genannten beispielhaften Ansätzen das informelle Wissen in der Regel auf der Ebene einzelner Features oder einzelner Dateninstanzen und daher sehr häufig von Domänenexperten abgefragt [BFSM22; SRYD14].

Als eine wesentliche Schlussfolgerung lässt sich festhalten, dass in der praktischen Umsetzung des in Abschnitt 2.3.1 vorgestellten Vorgehensmodells fast ausschließlich nur informelles Wissen der Domänenexperten eingebunden wird. In den drei Schritten zur Datenakquise und Datenexploration, zur Datenvorbereitung sowie zur Datenanalyse und Modellbildung sind Domänenexperten zudem nicht die Hauptakteure, sodass ihr informelles Wissen hier

nur sporadisch bei wenigen Aufgaben von ihnen direkt abgefragt wird. Gerade in diesen drei Schritten müssen aber die in industriellen Anwendungsfällen vorherrschenden komplexen Datencharakteristika primär adressiert werden. Daher steht die Nutzung von Domänenwissen in diesen drei Schritten im Fokus dieser Habilitationsschrift. Ebenso fokussiert die Schrift nicht auf informelles Wissen, sondern auf formal repräsentiertes Domänenwissen, das in einer digital erfassten Form bereits in den jeweiligen Anwendungsdomänen vorliegt. Wie in Abschnitt 2.2 beschrieben, ist formal repräsentiertes Domänenwissen meist umfassender als das informelle Wissen einzelner Domännenexperten. Ebenso müssen diese Domänenexperten in diesem Fall nicht mehr direkt involviert werden, um das Wissen in einzelnen Aufgaben der drei relevanten Schritte des Vorgehensmodells einzubinden, sodass für die Domänenexperten kein zusätzlicher Aufwand entsteht. Entsprechende Ansätze zur Nutzung formal repräsentierten Domänenwissens in den Schritten zur Datenakquise und Datenexploration, zur Datenvorbereitung sowie zur Datenanalyse und Modellbildung werden in den Kapiteln 5, 6 und 7 diskutiert.

Kapitel 3

Datencharakteristika in industriellen Anwendungsfällen

In Abschnitt 3.1 werden zentrale Klassen realer industrieller Anwendungsfälle für Datenanalysen zusammengefasst. Anschließend trägt Abschnitt 3.2 die in solchen industriellen Anwendungsfällen häufig vorkommenden komplexen Datencharakteristika zusammen. Abschnitt 3.3 diskutiert, dass diese Datencharakteristika auch in anderen Anwendungsdomänen auftreten, insbesondere bei datengetriebenen medizinischen Diagnosen [CRM+20; KU15; MGM20].

3.1 Industrielle Anwendungsfälle für Datenanalysen

Die im Rahmen dieser Habilitationsschrift untersuchten Anwendungsfälle stehen primär im Kontext von *Industrial Analytics* nach Gröger [Grö22], also Datenanalysen innerhalb der industriellen Wertschöpfung und innerhalb des Produktlebenszyklus. Die nachfolgenden Abschnitte 3.1.1 bis 3.1.4 beschreiben exemplarisch verschiedene industrielle Anwendungsfälle für Datenanalysen in einzelnen Phasen des in Abbildung 2.1 gezeigten Produktlebenszyklus. Abschnitt 3.1.5 betrachtet zusätzlich spezielle Anwendungsfälle für eine Analyse

Tabelle 3.1: Übersicht wesentlicher Merkmale der untersuchten industriellen Anwendungsfälle für Datenanalysen. Bei der Zielsetzung wird in Klammern angegeben, ob die Anwendungsfälle eher Industrial Analytics **(IA)** oder Web Analytics **(WA)** zuzuordnen sind.

Phase im Produktlebenszyklus	**Zielsetzung der Datenanalyse**	**Häufigste Analyseverfahren**	**Typische Daten**
Produktentwicklung	**(IA)** Vorhersage der Qualität und Funktionsfähigkeit von Produktdesigns	computerbasierte Simulation, Regression	CAD-Geometrien, JT-Standard, proprietäre Dateien (Binär, Text, CSV)
Produktion	**(IA)** Qualitätskontrolle, Fehlererkennung und Fehlerdiagnose	Klassifikation, Regression, Textanalyse	Sensordaten, Zeitreihendaten, Textdaten, MES, SCADA
Vertrieb	**(WA)** Klassifikation von Produkten in Produkttypen	Klassifikation	Textdaten, JSON-Dokumente
Produktnutzung	**(IA)** Vorhersage von Produktausfällen, Ursachenanalyse für Qualitätsprobleme	Klassifikation, Textanalyse	tabellarisch strukturierte Produktstammdaten, Textdaten
Übergeordnet	**(IA)** Prozessanalyse und -optimierung	Process Mining	Maschinenlogs, MES, ERP, PLM

und Optimierung von Prozessen im gesamten Produktlebenszyklus. Weitere im Rahmen dieser Arbeit untersuchten, aber nachfolgend nicht im Detail diskutierten Anwendungsfälle werden von Durnagöz et al. [DEF+24; DHMR24], Gerling et al. [GSH+20], Gröger [Grö15], Helwig et al. [HPS15], Hermann et al. [HCG+22; HMB+23], Klaiber et al. [KHS+24], Leitner et al. [LLE16], Mayr et al. [MMR+24], Shoshi et al. [SXF+24; SXF+25], Thalmann et al. [TGS+19], Wagner et al. [WKR+21] sowie in verschiedenen Übersichtsartikeln zu solchen Anwendungsfällen beschrieben [CCS+18; DB21; Grö22; KBT11; SAH18; WWIT16]. Tabelle 3.1 fasst die wesentlichen Merkmale aller untersuchten Anwendungsfälle hinsichtlich der betrachteten Zielsetzung, der am häufigsten genutzten Datenanalyseverfahren sowie der typischerweise verwendeten Daten zusammen.

3.1.1 Anwendungsfälle in der Produktentwicklung

Die Produktentwicklung, d. h. insbesondere der Entwurf von Produktdesigns, wird heutzutage vermehrt virtuell durchgeführt und durch Computerprogramme unterstützt, um Zeit und Kosten zu sparen. Anwendungsbereiche sind bspw. das *Computer-aided Design* (CAD) oder *Computer-aided Engineering* (CAE) [DWGB04; VK11]. Gerade im CAE-Bereich werden häufig computerbasierte Simulationen eingesetzt, um die zuvor entwickelten Produktdesigns hinsichtlich ihrer technischen Funktionsfähigkeit zu analysieren, ohne tatsächliche Prototypen bauen und diese physisch testen zu müssen. Diese Simulationen werden folglich zu einer Vorhersage genutzt, welche Qualität bzw. Funktionsfähigkeit die Produkte mit den jeweils entwickelten Produktdesigns später haben werden. Da computerbasierte Simulationen i. d. R. sehr rechenintensiv sind, werden sie häufig nur für bestimmte Szenarien mit einer eingeschränkten Auswahl der möglichen Eingabeparameter ausgeführt. Hier können datengetriebene Regressionen genutzt werden, um die Simulationen näherungsweise zu ersetzen bzw. um Simulationsergebnisse für die nicht simulierten Szenarien bzw. Bereiche der Eingabeparameter abzuschätzen [PZL+18; QSJ+05].

Speziell im CAE-Bereich gibt es eine Vielzahl an konkurrierenden Softwarelösungen für die Produktentwicklung, wobei die jeweiligen Softwareanbieter unterschiedliche Datenformate sowie Schnittstellen für den Datenzugriff unterstützen. Computerbasierte Simulationen basieren typischerweise auf Softwareprogrammen zur numerischen Strömungsmechanik (engl. Compuational Fluid Dynamics, CFD) [FP08] oder zur Finite-Elemente-Methode (FEM) [ZTZ13]. Diese Programme verwenden meist CAD-Geometrien als zentrale Eingabedaten. Obwohl es mit *Jupiter Tesselation*[1] (JT) einen von der ISO genormten Standard für solche CAD-Daten gibt, verwenden viele Softwareanbieter immer noch eigene proprietäre binäre oder textuelle Dateiformate für die Eingabedaten [MB08; ZRKM21]. Die Ausgabedaten der Simulationen werden von den Softwareprogrammen ebenso in anwendungsspezifischen Dateiformaten gespeichert. Hier ist eine ganze Bandbreite von Formaten zu finden, angefangen von Binärformaten über Textdateien bis zu CSV-basierten Dateien (Comma-Separated Values) [ZRKM21].

[1] Jupiter Tesselation: `https://www.iso.org/standard/62770.html`

3.1.2 Anwendungsfälle in der Produktion

In der Phase Produktion werden in dieser Arbeit primär Anwendungsfälle zur Qualitätskontrolle [ISO15] für Produkte in einem Produktionsprozess untersucht. Hierbei ist insbesondere die Fehlererkennung und Fehlerdiagnose von Belang [Ise06]. Ziel der Fehlererkennung ist es, zu erkennen, ob fehlerhafte Prozessbedingungen vorliegen oder Produkte defekt sind. In der Fehlerdiagnose werden der Ort und die Art eines Fehlers sowie mögliche Fehlerursachen bestimmt. Letzteres dient als Grundlage, um die Fehler zu beheben.

Wilhelm et al. [WRGM21; WSR+20] betrachten Datenanalysen zur Fehlererkennung in einzelnen Schritten eines Produktionsprozesses, z. B. in einzelnen Montageschritten einer Montagelinie. Sie verwenden hierfür Zeitreihendaten, die von Sensoren der Maschinen oder der Produktionssysteme geliefert werden. Weiterhin können Daten aus einem MES oder einem SCADA-System (engl. *Supervisory Control and Data Acquisition*) der Produktionssteuerung ausgelesen werden. Die Fehlererkennung wird über ein binäres Klassifikationsverfahren umgesetzt, wobei die Klasse 'negativ' einer normalen Ausführung des Produktionsprozesses entspricht, während die Klasse 'positiv' einen Fehler anzeigt. Gerling [GSH+20] und Gröger [Grö22] beschreiben Anwendungsfälle, die mithilfe der oben genannten Daten bereits in frühen Schritten des Produktionsprozesses die finale Produktqualität vorhersagen, die am Ende des gesamten Prozesses zu erwarten ist. Dies erfolgt über Klassifikationsverfahren oder über Regressionen, die die Wahrscheinlichkeit eines Produktfehlers vorhersagen.

Hirsch et al. [HRKM18; HRM19] betrachten analog eine finale Qualitätskontrolle des Produkts am Ende des gesamten Produktionsprozesses (sog. *End-of-Line-(EoL-)Prüffeld*). Dabei werden die Produkte auf Prüfständen hinsichtlich ihrer Funktionstüchtigkeit getestet (siehe Abbildung 3.1). Diese Prüfstände liefern eine Vielzahl an Sensorsignalen, um bestimmte Produkteigenschaften zu testen. Diese Sensordaten werden zusammen mit dem Label 'i.O.' ('in Ordnung') für funktionstüchtige Produkte oder 'n.i.O.' ('nicht in Ordnung') für defekte Produkte in einer Diagnosedatenbank abgelegt. Auf Basis dieser Diagnosedatenbank kann ein binäres Klassifikationsmodell trainiert werden, das für neue Produkte vorhersagt, ob diese defekt sind oder nicht.

Hirsch et al. [HRKM18; HRM19] betrachten zudem die Fehlerdiagnose anhand eines Beispiels für Fahrzeugmotoren. Jeder dieser Motoren besteht aus

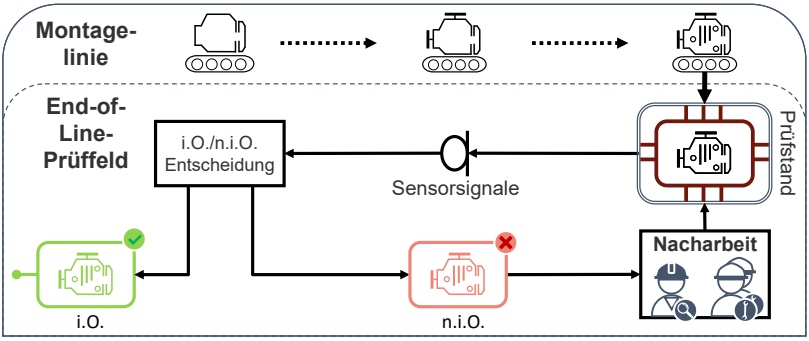


Abbildung 3.1: End-of-Line-Prüffeld (EoL-Prüffeld), Fehlerdiagnose und Nacharbeit für Motoren, vgl. Hirsch et al. [Hir21; HRM19].

ca. 100 Bauteilen, z. B. einem Turbolader oder mehreren Zylindern. Wird auf dem Prüfstand erkannt, dass ein Motor defekt ist, kommt dieser in einen Nacharbeitsbereich, in dem die Fehlerdiagnose durchgeführt wird. Hier untersuchen Qualitätsingenieure den defekten Motor und die Daten vom Prüfstand, um das Bauteil des Motors zu identifizieren, das den Fehler verursacht. Anschließend wird dieses Bauteil durch ein Ersatzteil ersetzt. Da die Fehlerdiagnose sehr komplex ist, treffen selbst erfahrene Qualitätsingenieure häufig Fehlentscheidungen. Hirsch et al. [HRM19] untersuchen daher, wie Qualitätsingenieure bei der Fehlerdiagnose über eine datengetriebene Klassifikation unterstützt werden können. Die Sensordaten der Prüfstände sowie weitere Eigenschaften der Motoren aus den Produktstammdaten dienen als Features für die Klassifikation. Die Klassenlabels sind die ca. 100 Bauteile eines Motors. Hierbei handelt es sich folglich um eine Multi-Klassen-Klassifikation, da das trainierte Klassifikationsmodell für jeden defekten Motor eines der ca. 100 Bauteile vorhersagen muss, das mit der höchsten Wahrscheinlichkeit den Defekt verursacht.

Kiefer et al. [KRM19a] betrachten weiterhin eine Diagnose bzw. Ursachenanalyse von Maschinenfehlern in einem Produktionsprozess. Die Annahme ist, dass bereits über eine Analyse von Sensordaten herausgefunden wurde, dass ein Fehler vorliegt. Der Ansatz von Kiefer et al. extrahiert zusätzlich noch Informationen aus Textdaten, die zuvor von Menschen zu bestimmten Fehlerfällen dokumentiert wurden. Damit kann der Ansatz sowohl die Ursachen als auch entsprechende Korrekturmaßnahmen für Maschinenfehler vorschlagen.

3.1.3 Anwendungsfälle im Vertrieb

Sun et al. [SRYD14] stellen einen Anwendungsfall für Datenanalysen vor, der in der Phase Vertrieb des Produktlebenszyklus verortet ist. Bezogen auf die Daten ist dieser Anwendungsfall eher dem Bereich Web Analytics zuzuordnen. Allerdings zeigen die Daten dieses konkreten Beispiels ähnliche Charakteristika wie bei den Anwendungsfällen aus dem Bereich Industrial Analytics. Daher ist auch dieses Beispiel Teil des Untersuchungsgegenstands dieser Arbeit.

Die Daten sind semi-strukturierte JSON-Dokumente aus Informationssystemen eines Webshops. In diesen JSON-Dokumenten werden für die im Webshop angebotenen Produkte eine textuelle Produktbeschreibung sowie weitere Produktattribute gespeichert. In über zehn Millionen der JSON-Dokumente fehlt allerdings die Angabe des Produkttyps, z.B. ob das Produkt ein Laptop oder ein Gartenstuhl ist. Das Ziel der Autoren ist, anhand der textuellen Produktbeschreibung und weiterer Attribute in den JSON-Dokumenten diese Dokumente einem Produkttypen zuzuordnen [SRYD14]. Dazu nutzen sie als Grundlage ein Klassifikationsverfahren. Da jedes Dokument zu einem von mehr als 5000 Produkttypen gehören kann, stellt dies eine Multi-Klassen-Klassifikation dar.

3.1.4 Anwendungsfälle in der Produktnutzung

Spieß et al. [SR19; SRWM22] untersuchen einen Anwendungsfall der Qualitätssicherung während der Produktnutzung. Dabei soll über eine Datenanalyse vorhergesagt werden, ob ein Fahrzeug während der Nutzung durch den Kunden innerhalb eines bestimmten Zeitraums, wie bspw. drei Jahre, ausfallen wird. Dies ist eine binäre Klassifikation, bei der die Klasse 'positiv' besagt, dass ein Fahrzeug wahrscheinlich ausfallen wird, während die Klasse 'negativ' für nicht ausfallgefährdete Fahrzeuge vorhergesagt wird. Das Klassifikationsmodell wird mit tabellarisch strukturierten Produktstammdaten trainiert, die Ausstattungsmerkmale der Fahrzeuge umfassen. Ebenso beinhalten die Daten ein Klassenlabel, das aussagt, ob ein Fahrzeug bereits ausgefallen ist. Das trainierte Modell wird dann auf die Daten von Fahrzeugen angewendet, die noch nicht ausgefallen sind, um somit ausfallgefährdete Fahrzeuge zu identifizieren. Die als ausfallgefährdet klassifizierten Fahrzeuge können bspw. in die Werkstätten zurückgerufen werden, bevor sie tatsächlich ausfallen.

Als weitere Möglichkeit kann das mit Daten aus der Produktnutzung trainierte Klassifikationsmodell auf Fahrzeuge angewendet werden, die aktuell noch zu produzieren sind [SRWM22]. Damit werden aus der Phase der Produktnutzung über Datenanalysen Erkenntnisse generiert, die wiederum in die Phase der Produktion zurück transferiert werden Es kann somit noch früher festgestellt werden, welche noch zu produzierenden Fahrzeuge später während der Nutzung durch den Kunden mit hoher Wahrscheinlichkeit ausfallen werden. Diese Fahrzeuge können schon während der Produktion mit robusteren Bauteilen ausgestattet und somit ihr Ausfall weitgehend verhindert werden [SR19].

Kassner et al. [KM16] betrachten einen weiteren Anwendungsfall, der Fahrzeuge betrifft, die wegen bestimmter Qualitätsprobleme bereits in die Werkstätten zurückgerufen und bei denen defekte Bauteile ausgetauscht wurden. Anschließend untersuchen Mechaniker der Werkstatt, der Automobilhersteller selbst sowie dessen Lieferanten die ausgebauten, defekten Fahrzeugbauteile, um die wahrscheinlichste Ursache des Qualitätsproblems zu bestimmen. Alle drei Beteiligten dokumentieren ihre Befunde in textuellen Berichten. Kassner et al. [KM16] entwickelten ein problemspezifisch angepasstes Klassifikationsverfahren auf Basis des K-nächster-Nachbar-Ansatzes [GWB+06], der die verschiedenen textuellen Berichte auswertet. Das Ziel ist, zu jedem Bericht den wahrscheinlichsten Fehlercode aus einem Pool von mehreren hundert Codes vorherzusagen, der die Ursache des jeweiligen Qualitätsproblems klassifiziert. Dieser Anwendungsfall ist eine Textanalyse für eine Multi-Klassen-Klassifikation, bei der die Fehlercodes die Klassenlabels repräsentieren.

3.1.5 Anwendungsfälle zur Prozessanalyse

Weiterer Untersuchungsgegenstand dieser Arbeit sind Anwendungsfälle zur Prozessanalyse und Prozessoptimierung, die Prozesse in verschiedenen Phasen des Produktlebenszyklus betrachten [BWD+21; DRG21; ZRS+22]. Dabei werden Verfahren des *Process Mining* auf den in den jeweiligen Phasen zur Verfügung stehenden Daten angewendet. Process Mining wird in die drei Typen *Process Discovery*, *Process Conformance Checking* und *Process Enhancement* eingeteilt [Aal+11; GMF+19]. Gängige Algorithmen nutzen sogenannte *Event Logs* als Datengrundlage. Diese Event Logs protokollieren relevante Ereignisse in den ausgeführten Prozessen, z. B. welche Arbeitsschritte zu welchen Zeit-

punkten von welchen Personen bzw. mit welchen Ressourcen durchgeführt wurden. Algorithmen der Process Discovery leiten aus solchen Event Logs ein vorher unbekanntes Prozessmodell ab, das möglichst alle in den Event Logs protokollierten Folgen von Arbeitsschritten abbildet. Conformance Checking wird angewendet, wenn bereits vorher ein Prozessmodell existiert. Dieses Prozessmodell wird mit dem Event Log abgeglichen, um festzustellen, ob die im Log protokollierten tatsächlichen Prozessausführungen mit dem Prozessmodell übereinstimmen bzw. an welchen Stellen sie sich unterscheiden. Beim Process Enhancement soll durch einen Vergleich zwischen Prozessmodell und den Protokollen im Event Log Einsichten gewonnen werden, wie das Prozessmodell verbessert werden kann, z. B. um Engpässe im Prozessmodell zu beheben.

Dreher et al. [DRG21] fassen verschiedene industrielle Anwendungsfälle für Process Mining in mehreren Phasen des Produktlebenszyklus zusammen. Ziegler et al. [ZRS+22] und Birk et al. [BWD+21] untersuchen spezifische Anwendungsfälle, die neben klassischen Event Logs noch andere Daten für die Prozessanalyse verwenden. Ziegler et al. [ZRS+22] betrachten eine Process Discovery für Projekte im Bereich der Produktentwicklung. In diesem Anwendungsfall werden auch Daten aus einem PLM-System analysiert. Birk et al. [BWD+21] untersuchen die ganze Bandbreite des Process Mining, von Verfahren zur Process Discovery über Conformance Checking bis zu Enhancment, für einen Bestell- und Produktionsprozess zur Herstellung pneumatischer Ventile. Neben Protokolleinträgen aus Event Logs von Maschinen des Produktionsprozesses verwenden die Autoren für die Prozessanalyse auch Daten aus einem MES sowie aus einem ERP-System.

3.2 Datencharakteristika in industriellen Anwendungsfällen

In diesem Abschnitt werden die in industriellen Awendungsfällen häufig vorkommenden Datencharakteristika beschrieben, die gängige Verfahren zur Datenakquise, Datenvorbereitung und Datenanalyse vor Probleme stellen. Dies ist eingeteilt in drei Gruppen von Datencharakteristika: Heterogene Datenformate und isolierte Datensilos (Abschnitt 3.2.1), verschiedene Ursachen für Bias in Daten [MMS+21; SG21] (3.2.2) sowie Datencharakteristika, die sich aus nichtstationären Produkt- oder Produktionsumgebungen ergeben (3.2.3).

3.2.1 Heterogene Datenformate und isolierte Datensilos

In vielen industriellen Anwendungsfällen sind die Formate der verfügbaren Daten sehr heterogen [Grö18; Grö22; WWIT16]. Wie Tabelle 3.1 zeigt, werden sogar innerhalb einzelner Phasen des Produktlebenszyklus unterschiedliche Arten von Daten erzeugt und verwaltet. Dies reicht u. a. von strukturierten Daten aus MES-, ERP- oder PLM-Informationssystemen eines Unternehmens, über semi-strukturierte Dokumente bis hin zu Sensor- oder Zeitreihendaten, Textdaten oder proprietären Dateiformaten. Nach Gröger [Grö18; Grö22] sind sogar die Datenmodelle der strukturierten MES-, ERP- oder PLM-Informationssysteme in verschiedenen Unternehmen oder Organisationseinheiten sehr unterschiedlich. Zum Beispiel sind die Datenmodelle eines MES auf die Eigenarten einer Produktionslinie zugeschnitten und unterscheiden sich, wenn entweder Serienfertigung, Sortenfertigung oder Einzelfertigung unterstützt werden soll.

Zudem liegen die Daten innerhalb eines Unternehmens häufig in voneinander isolierten Datensilos vor. Dies bedeutet, dass bestimmte Daten in Informationssystemen oder anderen Datenquellen gespeichert sind, auf die nur eine einzelne oder sehr wenige Organisationseinheiten eines Unternehmens Zugriff haben [ZRKM20]. Diese Aufteilung in isolierte Datensilos ist häufig historisch gewachsen, wenn einzelne Organisationseinheiten sich eigenständig um ihre Datenlandschaft kümmern, es aber keine ganzheitliche und unternehmensweite Strategie für die Umsetzung der Informationssysteme oder entsprechender Datenplattformen gibt [GH23; Grö18; Grö21]. In einigen Fällen ist die strikte Unterteilung in isolorierte Datensilos aber auch explizit gewollt, z. B. wenn besonders sensible bzw. schützenswerte Informationen nur von bestimmten Organisationseinheiten verwaltet und genutzt werden dürfen.

Die Heterogenität der Datenformate sowie deren Aufteilung in isolierte Silos haben insbesondere einen Einfluss auf den Schritt Datenakquise und Datenexploration des in Abschnitt 2.3.1 vorgestellten generischen Vorgehensmodells eines Datenanalyseprozesses. Konkret erschweren die geschilderten Probleme Domänenexperten die Suche nach und Auswahl der für ihre Analysen relevanten Daten [ZRKM21]. Insgesamt steht dies einer sog. Datendemokratisierung im Weg, d. h. dass verschiedene Daten jeder Person in einer Organisation zur Verfügung gestellt werden sollen, sofern dies nicht dem Datenschutz oder anderen Richtlinien widerspricht [Grö18; Grö21].

Tabelle 3.2: Datencharakteristika industrieller Anwendungsfälle und die jeweils primär verursachten Ausprägungen von Bias in Daten.

Datencharakteristika:	**Bias in Daten [MMS+21; SG21; WY12]:**
Klassenungleichverteilung	Bias bzgl. Mehrheitsklassen
Gruppenheterogenität	Aggregations-Bias bzgl. domänenspez. Gruppen
Gruppenungleichverteilung	Repräsentations-Bias bzgl. domänenspez. Gruppen
Falsch dokumentierte Labels	Mess- bzw. Dokumentations-Bias bzgl. Labels
Kleine Datenmenge	Überanpassung, Stichprobenverzerrung

3.2.2 Datencharakteristika als Ursachen für Bias in Daten

Die Daten industrieller Anwendungsfälle weisen häufig Datencharakteristika auf, die zu verschiedenen Ausprägungen von *Bias in Daten* führen [MMS+21; SG21; WY12]. Im Folgenden werden die wichtigsten Datencharakteristika beschrieben, die ursächlich für Bias in Daten sind. Dabei werden die Datencharakteristika anhand des Anwendungsfalls der Fehlerdiagnose im EoL-Prüffeld von Hirsch et al. [HRM19] illustriert. Tabelle 3.2 ordnet jeder der Datencharakteristika die jeweils verursachten Ausprägungen von Bias in Daten zu.

Klassenungleichverteilung Ein in der Literatur häufig genanntes Problem für Klassifikationsalgorithmen ist eine *Klassenungleichverteilung* in Daten [GFB+12; HG09; WP03; WY12]. Dies ist auch ein gängiges Problem in Daten industrieller Anwendungfälle [Grö22; HRM19; KM16; LLE16; WSR+20; WWIT16]. Abbildung 3.2 zeigt dies exemplarisch für zwei Features sowie zwei Klassen des Datensatzes aus dem EoL-Prüffeld. Die beiden Features entsprechen den Sensorsignalen 'Öltemperatur' und 'Kraftsfoffverbrauch', die am Prüfstand für Motoren gemessen werden. 'Kreis' und 'Stern' repräsentieren zwei der vielen Klassen im Datensatz, d. h. Bauteile der Motoren, wie ein Turbolader oder ein Injektor, die ursächlich für ein Qualitätsproblem sein können. Für diese beiden Klassen sind innerhalb der Abbildung jeweils mehrere Dateninstanzen im zweidimensionalen Feature-Raum verortet. Die Rechtecke in diesem Feature-Raum zeigen Entscheidungsgrenzen für die Muster M_1 und M_2 der Klassen 'Kreis' und 'Stern'. Die Klasse 'Kreis' kommt in insgesamt 38 Instanzen vor (Muster M_1). Die Klasse 'Stern' findet sich hingegen in lediglich sechs und damit in deutlich weniger Instanzen (Muster M_2). Häufig vorkommende Klassen wie die Klasse

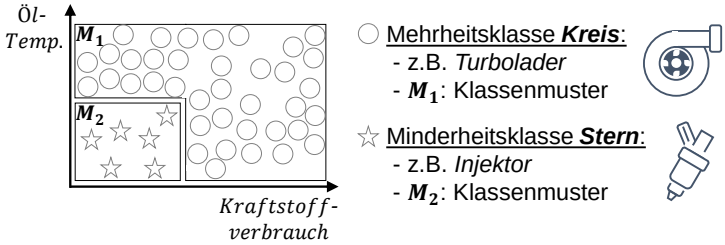


Abbildung 3.2: Illustration der Klassenungleichverteilung anhand zweier Features und zweier Klassen der Daten vom EoL-Prüffeld, vgl. Hirsch [Hir21].

'Kreis' werden als *Mehrheitsklassen* und ihre Instanzen *Mehrheitsinstanzen* bezeichnet. Die Klasse 'Stern' und alle seltenen Klassen werden *Minderheitsklassen* und ihre Instanzen *Minderheitsinstanzen* genannt [HG09].

In realen Multi-Klassen- und Multi-Label-Klassifikationen ist diese Klassenungleichverteilung noch viel stärker ausgeprägt [HRM19; KM16; SRYD14]. Zum Beispiel besteht der Datensatz für den Anwendungsfall im EoL-Prüffeld insgesamt aus 1050 Instanzen und 84 Klassen [HRM19]. Die zehn häufigsten Klassen kommen zusammen in ca. 44 % aller 1050 Instanzen vor, wobei jede einzelne dieser fünf Mehrheitsklassen zwischen 3 % und 10 % aller Instanzen ausmacht. Jede der verbleibenden 74 Klassen hat einzeln betrachtet einen deutlich geringeren Anteil mit weniger als 1 % der 1050 Instanzen. Allerdings kommen diese 74 Minderheitsklassen in der Summe in 56 % aller Instanzen vor. Die meisten Klassifikationsalgorithmen versuchen, die Genauigkeiten der Vorhersagen primär für die Mehrheitsklassen zu erhöhen, da sie davon ausgehen, dass somit auch die Vorhersageleistung für den gesamten Datensatz maximiert wird [GFB+12; HG09]. Daher lernen die Algorithmen vor allem die Muster der Mehrheitsklassen, ignorieren aber einen Teil der Muster der Minderheitsklassen. Die gelernten Klassifikationsmodelle besitzen also einen Bias in Bezug auf die Klassen, die sie vorhersagen können. Im schlimmsten Fall können sie im Beispiel aus dem EoL-Prüffeld [HRM19] nur die zehn Mehrheitsklassen richtig vorhersagen, während dies für die 74 Minderheitsklassen nicht möglich ist. Dies bedeutet jedoch, dass für bis zu 56 % der gesamten 1050 Instanzen die falsche Klasse vorhergesagt werden kann.

Eine Klassenungleichverteilung ist allerdings auch in Anwendungsfällen für eine binäre Klassifikation ein Problem [DB21; Grö22; GSH+20; LLE16; SRWM22; WSR+20; WWIT16]. Dies kann z. B. anhand von Anwendungsfällen der Fehlererkennung in einem Produktionsprozess aufgezeigt werden (siehe Abschnitt 3.1.2). Hier wird ein Fehlerfall durch die positive Klasse und ein Fall ohne Fehler durch die negative Klasse repräsentiert. Durch hohe Standards in der Qualitätssicherung sind speziell in Industrieländern diese Fehlerfälle auf wenige Prozent aller Fälle begrenzt [KBT11; Sch19]. Folglich kommt auch die positive Klasse in ebenso wenig Prozent der Dateninstanzen vor. Sie stellt damit eine Minderheitsklasse dar, die häufig von Klassifikationsalgorithmen weniger stark berücksichtigt wird als die negative Mehrheitsklasse. Dabei ist es in diesen Anwendungsfällen aber wichtiger, die positive Klasse richtig vorherzusagen, da ansonsten fehlerhafte Produkte an die Kunden ausgeliefert werden können.

Gruppenheterogenität Dateninstanzen speichern Informationen zu den jeweiligen Beobachtungen oder Messungen häufig bzgl. heterogener domänenspezifischer Entitätsgruppen. In industriellen Anwendungsfällen können diese Gruppen bspw. unterschiedliche Maschinentypen oder Produktgruppen sein, für die jeweils spezifische Informationen zu Maschinenfehlern oder zu Problemen mit der Produktqualität gespeichert werden. In anderen Anwendungsdomänen, z. B. in medizinischen Diagnosen, bilden häufig Personen verschiedener Geschlechter, Hautfarben oder unterschiedlichen Alters diese domänenspezifischen Gruppen [CRM+20; MGM20] (siehe auch Abschnitt 3.3.2).

Im Folgenden wird dieser Aspekt der heterogenen Gruppen sowie deren Auswirkungen auf Datenanalysen primär anhand von Produktgruppen aus industriellen Anwendungsfällen näher erläutert. Produktgruppen werden typischerweise in einer unternehmensweiten Dokumentation von Produktfamilien beschrieben und dabei meist in Form einer Produkthierarchie strukturiert [AK04]. Abbildung 3.3 zeigt eine abstrahierte Produkthierarchie für Verbrennungsmotoren [HRT+23]. Die erste Hierarchieebene unterscheidet Motoren nach ihrer Baureihe, z. B. ob es Diesel- oder Ottomotoren sind ('DM', 'OM'). Ebene 2 unterteilt die Motoren noch weiter in Motortypen. Hier umfasst bspw. die Gruppe 'OM54' Ottomotoren mit vier Zylindern, während die Gruppe 'OM56' Ottomotoren mit sechs Zylindern beschreibt. Die unterste Ebene unter-

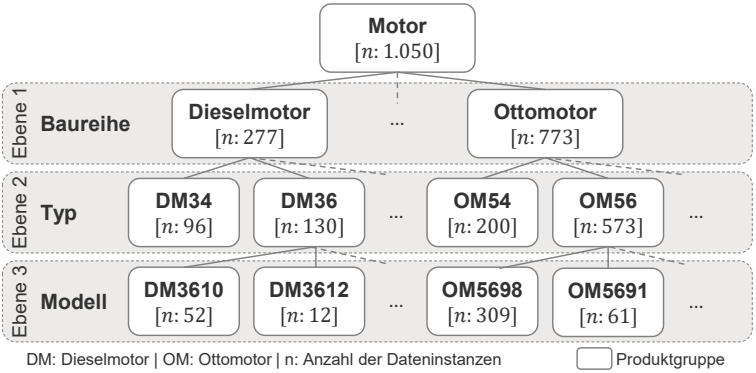


Abbildung 3.3: Abstrahierte Produkthierarchie zur Einteilung von Verbrennungsmotoren in Produktgruppen, vgl. Hirsch et al. [HRT+23].

scheidet verschiedene Motorenmodelle, die die Bauteile eines Motors konkret spezifizieren, z. B. welche Art von Einspritzdüse ein Motor hat.

Wie in Abschnitt 2.1 erläutert, lernen ML-Algorithmen aus den Trainingsdaten $\mathcal{D}$ eine Funktion $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$, welche die Muster zwischen Eingabedaten $x_i \in \mathcal{X}$ und Ausgabedaten $y_i \in \mathcal{Y}$ beschreibt. Viele Lernalgorithmen gehen dabei von der Annahme aus, dass diese zu lernende Funktion $\mathcal{M}$ über alle domänenspezifischen Entitätsgruppen hinweg nahezu identisch ist. Dies ist nach Suresh et al. [SG21] und Mehrabi et al. [MMS+21] in realen Anwendungsfällen aber häufig nicht der Fall, was auch auf zahlreiche industrielle Anwendungsfälle zutrifft [GSH+20; HRM19; HRM20a; KM16; SRYD14; TGS+19; WKR+21; WSR+20; WWIT16]. Die Ursache dieses Problems ist meist in der Vielfalt und *Heterogenität der realen domänenspezifischen Gruppen* begründet. In industriellen Anwendungsfällen entspricht dies der hohen Variantenvielfalt von Produkten wie auch der hohen Heterogenität von Produktionsanlagen und der dort eingesetzten Maschinen [HZWK08]. So unterscheiden sich bspw. die Produktvarianten verschiedener Produktgruppen hinsichtlich ihrer technischen Eigenschaften und Zusammensetzung aus Bauteilen [HRM20a].

In einer Klassifikation unterscheiden sich beispielsweise die zu lernenden Muster für ein und dieselbe Klasse zwischen verschiedenen domänenspezifischen Gruppen. Abbildung 3.4 illustriert dies exemplarisch für den Datensatz aus dem EoL-Prüffeld. Hier sind zusätzlich zu den in Abbildung 3.2 darge-

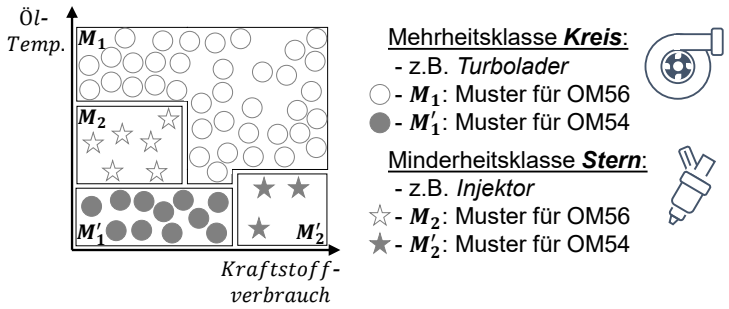


Abbildung 3.4: Illustration der Gruppenheterogenität und Gruppenungleichverteilung anhand zweier Features und zweier Klassen der Daten vom EoL-Prüffeld, vgl. Hirsch [Hir21].

stellten Mustern M_1 und M_2 je Klasse noch ein weiteres Muster dargestellt: M_1' für Klasse 'Kreis' und M_2' für Klasse 'Stern'. Dabei repräsentieren die beiden Muster einer Klasse jeweils einen anderen Motortyp und werden über unterschiedliche Wertebereiche der dargestellten Features beschrieben. Das Muster M_1 beschreibt die Klasse 'Kreis' für den Motortyp 'OM56' über eine mittlere bis hohe Öltemperatur, während Muster M_1' die gleiche Klasse für Motortyp 'OM54' über niedrige Werte dieses Features repräsentiert. Die Klasse 'Stern' wird einmal in Muster M_2 für Motortyp 'OM56' durch mittlere Werte in der Öltemperatur und niedrige Werte im Kraftstoffverbrauch gekennzeichnet. Für Motortyp 'OM54' beschreibt Muster M_2' diese Klasse allerdings über eine niedrige Öltemperatur sowie einen hohen Kraftstoffverbrauch.

Insgesamt kann diese Heterogenität der domänenspezifischen Entitätsgruppen bei einer Datenanalyse zu einem sog. *Aggregations-Bias* führen [MMS+21; SG21]. Dies bedeutet, dass sich die Ergebnisse der Datenanalyse, z. B. die trainierten Klassifikationsmodelle, unterscheiden, je nachdem ob die gleiche Analyse (a) einmal für die gesamte Datenmenge oder (b) jeweils separat für Datenteilmengen einzelner Gruppen durchgeführt wird. Häufig ist bei einer Analyse auf der gesamten Datenmenge auch die Qualität der Analyseergebnisse schlechter. In einer Klassifikation liegt dies primär daran, dass die Anzahl der von einem ML-Algorithmus zu lernenden Muster auf der gesamten Datenmenge höher ist, da diese Datenmenge selbst für eine einzelne Klasse mehrere Muster für die einzelnen domänenspezifischen Gruppen beinhalten kann [SRYD14].

Ebenso können sich Klassenmuster zwischen verschiedenen Gruppen überlappen. Beide Aspekte machen es Algorithmen schwer, die Muster klar voneinander abzugrenzen. Die Datenteilmengen der einzelnen Gruppen sind hingegen spezifischer und beinhalten insgesamt weniger Klassenmuster, die sich innerhalb der Datenteilmengen auch weniger stark überlappen [HRT+23; TRSM23].

Gruppenungleichverteilung Neben den Klassen sind in vielen industriellen Anwendungsfällen auch die *domänenspezifischen Gruppen* innerhalb der Daten *ungleichmäßig verteilt* [DB21; Grö22; HRM19; HRM20a; KM16; SRWM22; SWK09; WSR+20; WWIT16]. In Abbildung 3.3 sind bei jeder Produktgruppe in der Hierarchie die jeweilige Anzahl der Dateninstanzen im Datensatz des EoL-Prüffelds angegeben. Auf jeder Ebene der Hierarchie sind die einzelnen Produktgruppen unterschiedlich häufig vertreten. Die größte Ungleichverteilung besteht auf der untersten Ebene der Motorenmodelle. Hier reicht die Anzahl der Dateninstanzen von zwölf für die Gruppe 'DM3612' bis 309 für 'OM5698'. Zusammen mit der Heterogenität der Gruppen führt dies dazu, dass auch die für die jeweiligen Gruppen spezifischen Muster in den Daten ungleichmäßig verteilt sind. Hier neigen ML-Algorithmen analog zur Klassenungleichverteilung dazu, primär die Muster für häufig vorkommende Gruppen zu lernen. Dabei ignorieren die Algorithmen aber oft die spezifischen Muster für in den Daten selten repräsentierte Gruppen, sodass die trainierten ML-Modelle für diese Minderheitsgruppen fehlerhafte Vorhersagen treffen. Sind einzelne Gruppen in den Daten unterrepräsentiert, wird dies in der Literatur als eine Variante eines *Repräsentations-Bias* bezeichnet [MMS+21; SG21].

Die damit einhergehende Problematik kann anhand der in Abbildung 3.4 dargestellten Muster M_2 und M_2' der Klasse 'Stern' für die Motorengruppen 'OM54' und 'OM56' verdeutlicht werden. Insbesondere das Muster M_2' der Gruppe 'OM54' kommt im Datensatz mit nur drei Instanzen sehr selten vor. Da M_2' nur durch so wenig Instanzen repräsentiert wird, neigen ML-ALgorithmen dazu, dieses selten vorkommende Muster zu ignorieren. Die Algorithmen ordnen dann den entsprechenden Bereich des Feature-Raums für eine niedrige Öltemperatur und einen hohen Kraftstoffverbrauch den umgebenden Mustern M_1 bzw. M_1' zu. Damit wird für die Instanzen des eigentlichen Musters M_2' aber auch die falsche Klasse 'Kreis' anstelle der Klasse 'Stern' vorhergesagt.

Falsch dokumentierte Labels In vielen industriellen Anwendungsfällen werden die Labels y_i in den Instanzen (x_i, y_i) des Datensatzes $\mathcal{D}$ von Menschen dokumentiert [GSH+20; KM16; KRM19a; SRYD14; WSR+20; WWIT16]. In dem in Abbildung 3.1 aufgezeigten EoL-Prüffeld dokumentieren z. B. die Mitarbeiter im Bereich der Nacharbeit in den Daten, welche Bauteile sie in den defekten Motoren ersetzt haben [HRKM18]. Die entsprechenden Entscheidungsprobleme sind häufig aus technischer Sicht sehr komplex und benötigen ein hohes Maß an Domänenexpertise [BRL+19; HRM19; RBE+20; SRYD14]. Da die menschlichen Akteure hierfür nur begrenzt Zeit zur Verfügung haben, machen sie nicht nur bei den Entscheidungen selbst, sondern auch bei deren Dokumentation immer wieder Fehler. Dies bedeutet, dass eine gewisse Zahl an *Labels* y_i in den Daten *falsch dokumentiert werden*. Nach Suresh et al. [SG21] ist dies eine spezielle Form eines *Mess- bzw. Dokumentations-Bias*.

Dabei können die falsch dokumentierten Labels in unterschiedlichen Ausprägungen vorkommen. Eine Ausprägung sind unbrauchbare Labels wie "verschiedenes", die bei einer Klassifikation keinen Mehrwert bieten, da sie keins der gültigen Klassenlabels in der Menge $\mathcal{Y}$ darstellen. Die meisten unbrauchbaren Labels können relativ einfach in den Daten identifiziert werden [HRM19; KRM19a]. Die zweite Ausprägung sind solche, die zwar in der Menge $\mathcal{Y}$ der gültigen Labels vorkommen, aber dennoch bei einer konkreten Dateninstanz falsch dokumentiert sind. Im Beispiel des EoL-Prüffelds ist dies der Fall, wenn Mitarbeiter bspw. einen Turbolader im Motor ersetzen, aber fälschlicherweise einen Injektor in den Daten dokumentieren. Solche Ausprägungen falsch dokumentierter Labels führen dazu, dass in den Daten innerhalb der Entscheidungsgrenzen bestimmter Muster verrauschte Instanzen auftreten.

In Abbildung 3.5 sind im Vergleich zur Abbildung 3.4 noch zusätzlich vier verrauschte Instanzen mit falsch dokumentierten Klassenlabels eingetragen. Im Muster M_1 der Klasse 'Kreis' gibt es drei verrauschte Instanzen der Klasse 'Stern'. Ebenso gibt es im Muster M_2' eine verrauschte Instanz, die fälschlicherweise der Klasse 'Kreis' zugeordnet ist. Treten solche verrauschten Instanzen relativ häufig mit dem gleichen Klassenlabel auf, können ML-Algorithmen irrtümlicherweise davon ausgehen, dass die Instanzen und ihr Klassenlabel richtig sind. Die Algorithmen lernen für sie dann ein weiteres Muster und ordnen dieses Muster der falsch dokumentierten Klasse zu [HG09].

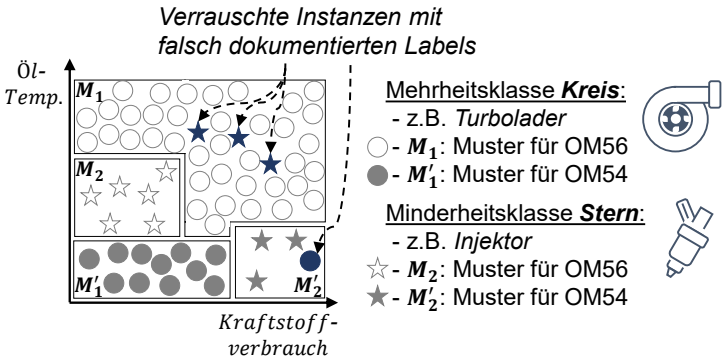


Abbildung 3.5: Illustration von falsch dokumentierten Klassenlabels anhand zweier Features und zweier Klassen der Daten vom EoL-Prüffeld, vgl. Hirsch [Hir21].

Weisen die Daten zudem eine Klassen- oder Gruppenungleichverteilung auf, variiert die Fehlerhäufigkeit bei der Dokumentation der Labels für Dateninstanzen zusätzlich zwischen verschiedenen Klassen und domänenspezifischen Entitätsgruppen [SG21]. Es ist davon auszugehen, dass die menschlichen Akteure sowohl mit seltenen Klassen als auch mit seltenen Gruppen weniger oft in Berührung kommen und daher mit diesen auch weniger Erfahrung haben. Folglich machen sie gerade für seltene Klassen und Gruppen häufiger Fehler bei der Dokumentation der Klassenlabels. Da speziell die Muster seltener Gruppen durch eine geringe Anzahl an Dateninstanzen beschrieben werden, nehmen bei diesen Mustern die verrauschten Instanzen relativ gesehen einen höheren Anteil ein. In Muster M'_2 in Abbildung 3.5 ist die verrauschte Instanz bspw. eine von insgesamt nur vier Instanzen. Da sie mit 25 % aller Instanzen relativ häufig in dem Muster vorkommt, ist es für Klassifikationsalgorithmen sehr schwer zu erkennen, dass dieser Instanz das falsche Klassenlabel zugeordnet ist.

Verrauschte Instanzen entstehen nicht nur durch eine falsche Dokumentation der Labels. Auch wenn einer Instanz das richtige Label zugeordnet wird, können Fehler bei der Messung oder Dokumentation der Feature-Werte gemacht werden [HG09]. Dies hat zur Folge, dass sich die Instanz im Feature-Raum verschiebt. Speziell an den Entscheidungsgrenzen der Muster kann es vorkommen, dass sich eine solche Instanz durch die Verschiebung im Feature-Raum

nicht mehr im Muster ihrer tatsächlichen Klasse, sondern einer anderen Klasse befindet. Diese Form des Bias muss aber von den falsch dokumentierten Labels unterschieden werden können, da zur Korrektur der Instanzen im einen Fall die Feature-Werte und im anderen Fall die Labels angepasst werden müssen.

Kleine Menge an qualitativ ausreichenden Daten Viele Übersichtsartikel ordnen industrielle Anwendungsfälle des maschinellen Lernens primär in den Kontext Big Data ein und betonen, dass aus den Datenquellen große Mengen an Daten extrahiert und analysiert werden müssen [CCS+18; CKC20; DB21; KBT11; SAH18; WWIT16]. Dies ist auch richtig, wenn man direkt die Daten betrachtet, die in den Quellsystemen gespeichert sind. Allerdings werden in den Schritten der Datenakquise und der Datenvorbereitung eines Datenanalyseprozesses für jeden Anwendungsfall nur eine für diesen Fall relevante Teilmenge der Daten aus den Quellsystemen extrahiert und integriert. In vielen industriellen Anwendungsfällen reduziert sich dabei die für die Analyse verwendbare Datenmenge deutlich. Speziell für Anwendungsfälle des überwachten Lernens stehen schließlich nur eine *kleine Menge an qualitativ ausreichenden Daten* zur Verfügung [Grö22; HRM19; KM16; KRM19a; SRYD14; WSR+20].

Im Beispiel der Fehlerdiagnose im EoL-Prüffeld sind im Trainingsdatensatz $\mathcal{D}$ lediglich 1050 Dateninstanzen mit 84 Klassenlabels und 115 Features verfügbar [HRM19]. Diese eher kleine Menge an Dateninstanzen bei einer gleichzeitig hohen Anzahl an Klassen und Features führt häufig zu dem Problem der Überanpassung der trainierten Klassifikationsmodelle. Dies bedeutet, dass die Modelle nahezu perfekt die in den Trainingsdaten vorhandenen Klassenmuster beschreiben. Allerdings sind die Modelle zu spezifisch an diese Muster angepasst und generalisieren sehr schlecht für andere Muster, die sich in neuen Daten wiederfinden können. Ebenso kann die während der Datenintegration aus den Quellsystemen durchgeführte Reduktion der Datenmenge nach Suresh et al. [SG21] und Mehrabi et al. [MMS+21] zu einem weiteren *Repräsentations-Bias* bzw. zu einer Stichprobenverzerrung führen. Die Verteilungen von Feature-Werten, Klassen, Mustern oder auch von domänenspezifischen Gruppen können sich während der Datenintegration deutlich verändern. Folglich können sich auch die Verteilungen in den dann zur Verfügung stehenden Trainingsdaten von denen in den ursprünglichen Quellsystemen unterscheiden.

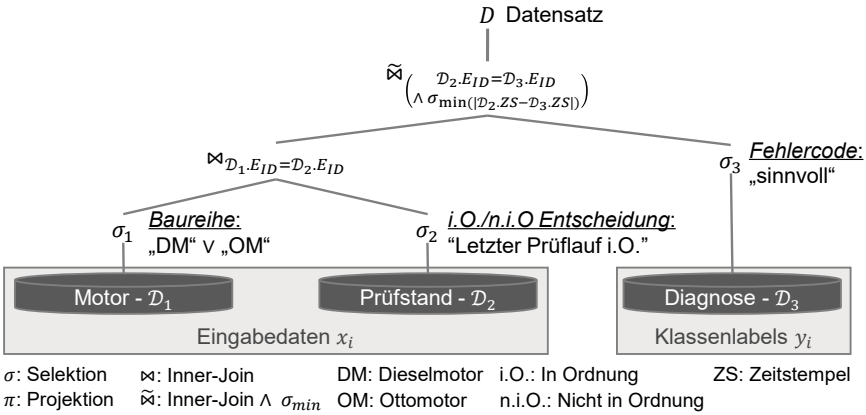


Abbildung 3.6: Ausschnitt aus dem Datenintegrationsprozesses für den Anwendungsfall im EoL-Prüffeld, vgl. Hirsch [Hir21].

Ein wesentlicher Grund für diese Reduktion der Datenmenge ist, dass für jeden Anwendungsfall einer Datenanalyse nur eine spezifische echte Teilmenge aus den Datenquellen relevant ist [Grö21; ZRKM21]. Weiterhin werden aufgrund der Heterogenität der Datenformate und der damit einhergehenden Komplexität der Datenakquise nicht immer alle relevanten Daten in den Quellsystemen gefunden [Grö21; HRM19; WSR+20] (siehe auch Abschnitt 3.2.1). Zusätzlich ist häufig auch die Qualität der Daten sowie der Datenmodelle in den Quellsystemen selbst nicht ausreichend [Grö22; KM16; KRM19a; WSR+20].

Um diese Aspekte zu veranschaulichen, zeigt Abbildung 3.6 die wesentlichen Schritte des Datenintegrationsprozesses für die Daten im EoL-Prüffeld [Hir21]. Dieser integriert Daten aus drei Datenquellen. Quelle D_1 speichert die Produktspezifikationen der Motoren, während D_2 die Messwerte der Prüfstände beinhaltet. Beide Quellen zusammen liefern die Features, also die Eingabedaten x_i der Trainingsdaten $\mathcal{D}$. Die Klassenlabels y_i sind in den Diagnosedaten D_3 gespeichert, in denen Mitarbeiter dokumentieren, welche Bauteile sie bei einzelnen defekten Motoren ausgetauscht haben.

Diese drei Quellen beinhalten Daten zu mehr als 500 000 Motoren und noch mehr Prüfläufen [Hir21]. Allerdings werden diese Daten in mehreren Selektionen und relationalen Verbünden auf die 1050 Instanzen im Datensatz $\mathcal{D}$ reduziert. Die erste Selektion σ_1 extrahiert aus Quelle D_1 nur Daten zu den

Motorbaureihen bestimmter Diesel- und Ottomotoren, die für den betrachteten Anwendungfall relevant sind. Weiterhin werden aus Quelle D_2 nur Daten zu Prüfläufen selektiert (σ_2), nach denen das jeweilige Qualitätsproblem auch wirklich gelöst wurde, also letzlich zu einer 'i.O.'-Entscheidung geführt hat. Die von σ_1 und σ_2 selektierten Datenmengen werden anschließend über einen Verbund ($\bowtie$) anhand der ID des Motors verknüpft, die in D_1 der Primärschlüssel und in D_2 ein Fremdschlüssel ist. Durch diese drei Operationen reduziert sich die Datenmenge bereits auf nur noch ca. 13 500 Instanzen [HRM19].

Die nachfolgende Selektion σ_3 ist ein Beispiel, warum sich die für die Analyse verwendbare Datenmenge aufgrund einer schlechten Qualität der Daten in den Quellsystemen noch weiter reduziert [HRM19]. Datenquelle D_3 beinhalten sehr viele Daten mit unbrauchbaren Klassenlabels wie 'verschiedenes' oder 'Motor kaputt'. Diese Daten werden in Selektion σ_3 herausgefiltert. Im anschließenden Verbund, müssen die Diagnosedaten aus Quelle D_3 sowohl mit den Motoren aus D_1 als auch mit den passenden Prüfläufen aus D_2 verknüpft werden. Quelle D_3 beinhaltet die ID eines Motors als Fremdschlüssel, sodass diese ID für den Verbund zwischen D_1 und D_3 verwendet werden kann. Allerdings gibt es keine passende Schlüsselbeziehung zwischen D_2 und D_3. Diese fehlende Schlüsselbeziehung ist ein Beispiel für unzureichende Datenmodelle der Quellsysteme, die zu einer weiteren Reduktion der verwendbaren Datenmenge führen [WSR+20]. Hier kann lediglich eine Heuristik zur Realisierung des Verbunds zwischen D_2 und D_3 verwendet werden, die in beiden Quellen vorkommende Zeitstempel ausnutzt [HRM19]. Der Zeitstempel in D_2 beschreibt den Zeitpunkt, wann der jeweilige Prüflauf stattgefunden hat, während es in D_3 entsprechend der Zeitpunkt der danach stattfindenden Nacharbeit an Motoren ist. Die in σ_{min} verwendete Heuristik geht davon aus, dass ein Prüflauf aus D_2 mit der Nacharbeit in D_3 zusammengehört, die zeitlich am nächsten beieinander liegen, bei denen also die Differenz zwischen beiden Zeitstempeln minimal ist. Diese nicht exakte Heuristik führt aber dazu, dass nicht zu jeder Nacharbeit aus D_3 ein Verbundpartner aus D_2 gefunden werden kann. Insgesamt reduzieren die Selektion σ_3 und der danach mit der anwendungsspezifischen Heuristik durchgeführte Verbund die Datenmenge auf 1050 Instanzen.

Diese Beispiele für unbrauchbare Klassenlabels oder fehlende Schlüsselbeziehung in Datenquellen werden auch in der Literatur häufig als zentrale Probleme

bei der Datenintegration genannt, die letztlich negative Auswirkungen auf die Qualität der mit den integrierten Daten gewonnenen Analyseergebnisse und Erkenntnisse haben [BWK+22; Grö21; Grö22; WSR+20; WWIT16]. Weitere Probleme sind Duplikate in den Quellsystemen oder nicht synchronisierte Zeitstempel bei Daten, die über verschiedene Schritte eines Produktionsprozesses akquiriert werden. Ebenso werden bei der Akquise und Speicherung von Zeitreihendaten Sampling-Verfahren eingesetzt, die die Abtastrate der Zeitreihen erheblich reduzieren, um somit den Kommunikations- und Speicheraufwand zu reduzieren [BWK+22; PMR19; PMRG21; WSR+20]. Dies reduziert allerdings auch den Informationsgehalt in den Daten. Im Beispiel des EoL-Prüffelds werden die Zeitreihendaten der Sensoren vom Prüfstand (Quelle D_2) so stark verdichtet, dass pro Sensor nur noch ein numerischer Mittelwert pro Zeitreihe gespeichert wird [HRM19; HRM20a]. Kiefer et al. [KRM19a; KRM19b] und Kassner et al. [KHW+17; KM16] beschreiben weitere Datenqualitätsprobleme, die speziell bei Textdaten auftreten. Beispielsweise werden viele Textdaten von Menschen dokumentiert und weisen daher typische Qualitätsprobleme wie Rechtschreib- und Grammatikfehler oder zu lange Sätze auf.

3.2.3 Datencharakteristika in nichtstationären Umgebungen

Nichtstationäre Umgebungen ändern ihren Zustand mit die Zeit [MRA+12]. In industriellen Anwendungsfällen ist insbesondere die Produktionsumgebung mit all ihren Sensoren und Informationssystemen eine nichtstationäre Umgebung [CCS+18; HB11]. Aber auch die im Kundeneinsatz befindlichen Produkte ändern ihre Eigenschaften durch die Produktnutzung kontinuierlich über die Zeit [SRWM22]. Solche nichtstationären Umgebungen können sich abrupt oder graduell verändern [WHC+16]. Abrupte Veränderungen treten plötzlich zu einem bestimmten Zeitpunkt auf. Beispielsweise kann ein Sensor in einem Produktionsprozess durch eine plötzliche Überspannung des Eingangsstroms fortan keine oder nur fehlerhafte Werte liefern. Graduelle Veränderungen vollziehen sich allmählich über eine gewisse Zeitspanne. Ein Beispiel ist, wenn ein Sensor mit der Zeit mehr und mehr verschleißt und auch die Fehlerrate in den Daten, die der Sensor liefert, mit diesem fortschreitenden Verschleiß zunimmt. Im Folgenden werden zwei der wichtigsten Probleme näher erläutert, die solche nichtstationäre Umgebungen für Datenanalysen mit sich bringen.

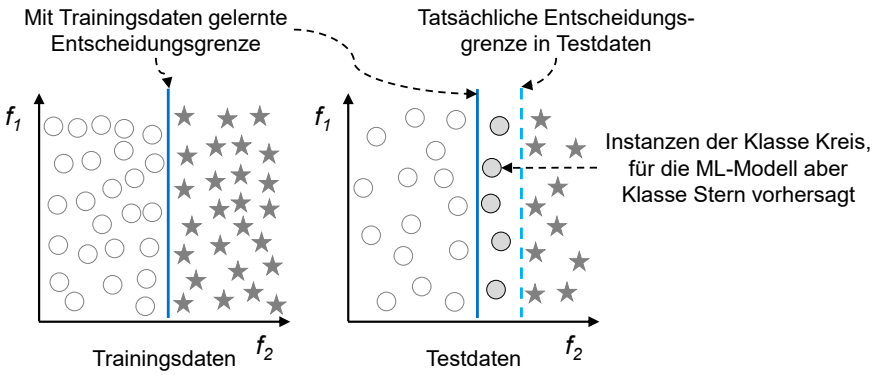


Abbildung 3.7: Schematische Darstellung einer Konzeptverschiebung anhand zweier Features f_1 und f_2 sowie anhand zweier Klassen.

Datensatzverschiebungen Nichtstationäre Umgebungen führen dazu, dass sich statistische Datenverteilungen ebenso über die Zeit ändern. Dies führt zu sog. *Datensatzverschiebungen* (engl. *Data Set Shifts*) [MRA+12; WHC+16]. Datensatzverschiebungen müssen beim Trainieren von ML-Modellen berücksichtigt werden, da die Vorhersagen dieser Modelle ansonsten aufgrund der sich über die Zeit verändernden Datenverteilungen immer fehlerhafter werden [BCN+17]. Allgemein wird ein ML-Modell auf einem Trainingsdatensatz $\mathcal{D}_{train}$ gelernt und auf einem Test- oder Anwendungsdatensatz $\mathcal{D}_{test}$ angewendet. In nichtstationären Umgebung unterscheiden sich häufig auch der Zeitpunkt des Trainings auf $\mathcal{D}_{train}$ von dem Zeitpunkt, wann das ML-Modell auf $\mathcal{D}_{test}$ angewendet wird. Folglich können sich auch die statistischen Datenverteilungen zwischen $\mathcal{D}_{train}$ und $\mathcal{D}_{test}$ unterscheiden. Das gelernte Modell ist dann an die Verteilungen in $\mathcal{D}_{train}$ gut angepasst, generalisiert jedoch schlecht für die in $\mathcal{D}_{test}$. In industriellen Anwendungsfällen kommen insbesondere zwei spezielle Varianten von Datensatzverschiebungen vor [CCS+18; SRWM22; WSR+20]: Konzeptverschiebungen (engl. *Concept Shifts* oder auch *Concept Drifts*) [EP11; LLD+19] sowie Kovariatenverschiebung (engl. *Covariate Shifts*) [YNSC19].

Bei Konzeptverschiebungen ändert sich die Funktion $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ von Eingabedaten $x_i \in \mathcal{X}$ zu Ausgabedaten $y_i \in \mathcal{Y}$ zwischen $\mathcal{D}_{train}$ und $\mathcal{D}_{test}$. So können sich z. B. auch die Entscheidungsgrenzen der Muster in den Daten über die Zeit ändern [SRWM22]. Ein ML-Algorithmus lernt also aus $\mathcal{D}_{train}$ für bestimmte

Klassen Entscheidungsgrenzen, die sich von den in $\mathcal{D}_{test}$ vorkommenden Entscheidungsgrenzen derselben Klassen unterscheiden. Hier kann das gelernte ML-Modell einzelnen Instanzen in $\mathcal{D}_{test}$ das falsche Muster und damit die falsche Klasse zuordnen. Abbildung 3.7 zeigt diesen Sachverhalt schematisch für zwei Klassen 'Kreis' und 'Stern', die jeweils unterschiedliche Verteilung in den Trainings- und Testdaten aufweisen. Die aus den Trainingsdaten gelernte Entscheidungsgrenze (dunkelblaue durchgezogene Linie) wird vom ML-Modell auch zur Vorhersage der Klassenlabels für die Instanzen der Testdaten verwendet. Die tatsächliche Entscheidungsgrenze in den Testdaten ist aber bzgl. Feature f_2 leicht nach rechts verschoben (hellblaue gestrichelte Linie). Folglich sagt das trainierte ML-Modell für sämtliche Dateninstanzen zwischen der gelernten und der tatsächlichen Grenze fälschlicherweise die Klasse Stern vorher, obwohl diese Instanzen in den Testdaten eigentlich zur Klasse Kreis gehören.

Bei Kovariatenverschiebungen ändert sich die Funktion $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ zwischen Trainingsdaten $\mathcal{D}_{train}$ und Testdaten $\mathcal{D}_{test}$ nicht. Allerdings verändert sich die Verteilung der Eingabedaten bzw. der Feature-Werte $x_i \in \mathcal{X}$. Dies kann zur Folge haben, dass verschiedene Muster unterschiedlich häufig in $\mathcal{D}_{train}$ und $\mathcal{D}_{test}$ vorkommen. Wie bei der Klassen- und Gruppenungleichverteilung in Abschnitt 3.2.2 erläutert, ignorieren ML-Algorithmen häufig diejenigen Muster, die selten in Daten vorkommen. Bei einer Kovariatenverschiebung können bestimmte Muster in $\mathcal{D}_{train}$ selten vorkommen, die aber häufige Muster in $\mathcal{D}_{test}$ sind. In so einem Fall lernen ML-Algorithmen die betroffenen Muster in $\mathcal{D}_{train}$ nicht oder nur ungenau. Da diese Muster aber in $\mathcal{D}_{test}$ häufig vorkommen, machen die gelernten ML-Modelle folglich für viele Instanzen aus $\mathcal{D}_{test}$ fehlerhafte Vorhersagen. Abbildung 3.8 zeigt dies schematisch für drei Klassen 'Kreis', 'Stern' und 'Quadrat'. In den Trainingsdaten kommt die Klasse 'Quadrat' mit drei Instanzen nur selten im Vergleich zu den anderen beiden Klassen vor. Der Klassifikationsalgorithmus ignoriert folglich alle drei Instanzen der Klasse 'Quadrat' und lernt nur die Entscheidungsgrenze zwischen den Klassen 'Kreis' und 'Stern' (dunkelblaue durchgezogene Linie). In den Testdaten kommt die Klasse 'Quadrat' aber mit 10 Instanzen genauso häufig vor wie die Klasse 'Stern'. Die tatsächliche Entscheidungsgrenze zwischen diesen beiden Klassen (hellblaue gestrichelte Linie) wurde vom Klassifikationsalgorithmus aber nicht gelernt. Das ML-Modell wendet auf diese Instanzen somit nur die gelernte

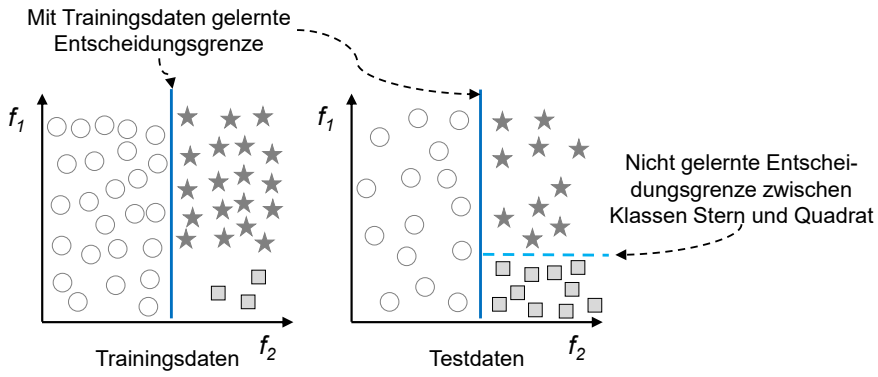


Abbildung 3.8: Schematische Darstellung einer Kovariatenverschiebung anhand zweier Features f_1 und f_2 sowie anhand dreier Klassen.

Entscheidungsgrenze zwischen den Klassen 'Kreis' und 'Stern' an und sagt für alle Instanzen der Klasse 'Quadrat' fälschlicherweise die Klasse 'Stern' vorher.

Nach Moreno-Torres et al. [MRA+12] können Methoden zur Adressierung von Datensatzverschiebungen entweder nur eine Konzeptverschiebung oder nur eine Kovariatenverschiebung behandeln. Es gibt also bisher keine Lösung, die mit Daten umgehen kann, bei denen beide Arten von Datensatzverschiebungen auftreten. In realen Anwendungsfällen treten Konzept- und Kovariatenverschiebungen aber häufiger gemeinsam auf, als es die wissenschaftliche Literatur annimmt. Beispiele finden sich u. a. in der Produktentwicklung [NLDD18], in medizinischen Anwendungsfällen [KU15; MGM20] sowie auch in dem in Abschnitt 3.1.4 beschriebenen Anwendungsfall von Spieß et al. [SRWM22] zur Klassifikation von Fahrzeugen, die während ihrer Nutzung ausfallen können. Hier fallen Fahrzeuge während ihrer Betriebszeit allmählich aus. Somit ändern sich in den von Ausfällen betroffenen Fahrzeugen auch graduell die Klassenlabels y_i in den Daten von 'negativ' (kein Ausfall) zu 'positiv' (Ausfall). Nachdem eine gewisse Anzahl an Fahrzeugen ausgefallen sind und sich deren Klassenlabels geändert haben, können sich auch die Entscheidungsgrenzen der Muster für negative und positive Dateninstanzen ändern (Konzeptverschiebung). Weiterhin ändern sich die Kundenwünsche über die Zeit, sodass in verschiedenen Zeiträumen Fahrzeuge mit unterschiedlichen Ausstattungsmerkmalen bestellt werden. Diese Ausstattungsmerkmale bilden im Anwendungsfall

die Features der Eingabedaten $x_i \in \mathcal{X}$. Die Verteilung der Feature-Werte ändert sich folglich ebenso über die Zeit (Kovariatenverschiebung). Treten wie in diesem Anwendungsfall beide Varianten einer Datensatzverschiebung auf, so kann es eine Trade-off-Beziehung zwischen diesen geben. Wenn z. B. Ansätze angewendet werden, die die Konzeptverschiebung adressieren, verschlimmert dies die Auswirkungen der Kovariatenverschiebung und umgekehrt [SRWM22].

Unstrukturierte und nichtlinear durchgeführte Prozesse Insbesondere in den Phasen Produktentwicklung und Produktion eines Produktlebenszyklus werden die Entwicklungs- bzw. Produktionsprozesse auf sehr unstrukturierte Weise ausgeführt [BWD+21; DRG21; WSR+20; ZRS+22]. Prozesse in der Produktentwicklung bestehen zwar aus einer Reihe von abstrakten Schritten, die von der Produktplanung, dem Entwurf, über die Computersimulation bis hin zu Produkttests reichen [DWGB04; VK11]. Allerdings werden in einem Entwicklungsprojekt typischerweise mehrere Instanzen dieser abstrakten Schritte ausgeführt, wobei sich die Reihenfolgen bzw. Kontrollflüsse dieser Schritte in verschiedenen Projekten deutlich unterscheiden können [ZRS+22]. Dies liegt hautpsächlich daran, dass Entwicklungsingenieure an verschiedenen Punkten des Projekts Entscheidungen treffen, die den weiteren Projektverlauf beeinflussen. Dies ist in der Produktentwicklung häufig notwendig, da jedes Entwicklungsprojekt durch individuelle Kunden- oder Marktanforderungen geprägt ist. Diese können sich dynamisch während des Projektverlaufs ändern, was eine Anpassung der Projektausführung zur Folge haben kann.

Die Ausführungen von Produktionsprozessen folgen häufig ebenso nicht exakt dem für diese Prozesse vordefinierten Kontrollfluss. Sobald Fehler auftreten, werden die involvierten Produkte oder Werkstücke aus dem vordefinierten Produktionsprozess entnommen [Ise06; Sch19; WSR+20]. Gerade teure Produkte oder Werkstücke werden dann häufig in einer Nacharbeit repariert und dann erneut im Produktionsprozess weiterverarbeitet. Dies kann im ursprünglichen Prozessschritt sein, aus dem sie auch vorher entnommen wurden, oder sogar in einem früheren Schritt des Produktionsprozesses [WSR+20]. Folglich werden die konkreten Ausführungen der eigentlich linear vordefinierten Produktionsprozesse dadurch häufig nichtlinear [CCS+18], d. h. die Ausführungsreihenfolge umfasst einige zuvor nicht definierte Schleifen.

Diese unstrukturierte und häufig nichtlineare Ausführung der Prozesse hat Auswirkungen auf das Datenmanagement und die Datenanalyse. Es erschwert insbesondere die Integration der im Rahmen der Prozessausführungen gesammelten Daten [BWD+21; DRG21; Grö22]. Da zum Beispiel ein Produkt einzelne Prozessschritte mehrmals durchlaufen kann, können für dieses auch mehrere Dateninstanzen in den Informationssystemen gespeichert werden, die jedoch jeweils mit anderen IDs oder Zeitstempeln versehen werden [WRGM21]. Dies führt zu Duplikaten in den Daten, die nur sehr schwer dem tatsächlichen Produkt zugeordnet werden können [Grö22; WWIT16].

Ebenso führen die Eigenschaften unstrukturierter und nichtlinearer Prozessausführungen zu Problemen bei einer Prozessanalyse mittels Verfahren des Process Mining [DRG21; ZRS+22]. Verfahren des Process Mining gehen davon aus, dass ein einzelnes, generisches Prozessmodell abgeleitet werden kann, das alle möglichen Varianten der Ausführungen eines Prozesses korrekt beschreibt. Die Variantenvielfalt der unstrukturiert ausgeführten Prozesse ist aber so hoch, dass z. B. Verfahren der Process Discovery ein sogenanntes 'Spaghetti'-Prozessmodell ableiten [Aal11]. Dies sind große und unübersichtliche Prozessmodelle, in denen die einzelnen Varianten der Prozessausführungen nicht mehr separierbar sind und daher auch nicht mehr analysiert werden können. Nach Günther und van der Aalst [GA07] sind unstrukturierte Prozessausführungen ein grundlegendes Problem im Bereich des Process Mining.

3.3 Datencharakteristika in anderen Anwendungsdomänen

Auch wenn sich diese Arbeit auf industrielle Anwendungsfälle fokussiert, existieren ein Großteil der in Abschnitt 3.2 beschriebenen Datencharakteristika auch in anderen Anwendungsdomänen [MMS+21; MRA+12; SG21]. Insbesondere in datengetriebenen medizinischen Diagnosen führen komplexe Datencharakteristika häufig dazu, dass maschinelle Lernverfahren nicht die gewünschte Vorhersageleistung erreichen [CRM+20; KU15; MGM20]. Die nachfolgenden Abschnitte diskutieren daher die Ausprägungen der in Abschnitt 3.2 beschriebenen Datencharakteristika in Anwendungsfällen aus der Medizin. Tabelle 3.3 fasst die primär untersuchten Anwendungsfälle hinsichtlich der betrachteten Zielsetzung, der am häufigsten genutzten Datenanalyseverfahren sowie der

Tabelle 3.3: Übersicht wesentlicher Merkmale der untersuchten Anwendungsfälle für datengetriebene medizinische Diagnosen.

Literaturquelle	Zielsetzung	Analyseverfahren	Typische Daten
Chan et al. [CRM+20]	Diagnose von Hautkrebs	Klassifikation (binär, Multi-Klassen)	Bilddaten, elektronische Krankenakten, Genomikdatenbanken, Textdaten, Sensordaten
Suresh et al. [SG21]	Diagnose kardiologischer Krankheiten	Klassifikation (meist binär)	Zeitreihendaten von EKG-Systemen, Genomikdatenbanken, Textdaten
Kahn et al. [KU15] und Maitín et al. [MGM20]	Diagnose neurologischer Krankheiten	Klassifikation (meist binär)	Zeitreihendaten von EEG-, ENG-, EMG-Systemen, Bilddaten aus Tomographien

typischerweise verwendeten Daten zusammen. Die in der Tabelle angegebenen Literaturquellen entsprechen Übersichtsartikeln, die jeweils mehrere Veröffentlichungen zu Anwendungsfällen für Datenanalysen untersuchen.

Chan et al. [CRM+20] geben einen Überblick über ca. 70 wissenschaftliche Artikel, die Anwendungsfälle des maschinellen Lernens in der Dermatologie untersuchen. In den meisten dieser Artikel werden Klassifikationsverfahren auf Bilddaten von Patienten angewendet, um Hautkrebs zu diagnostizieren. Einige nutzen als Daten noch elektronische Krankenakten, Genomikdatenbanken, Textdaten aus Versicherungsunterlagen oder personalisierte Sensordaten von Mobilgeräten. Ein erster Schritt bei der Diagnose von Hautkrebs ist zu erkennen, ob bei einem Patienten überhaupt Hautkrebs vorliegt. Dies kann über eine binäre Klassifikation umgesetzt werden. Im weiteren Verlauf soll herausgefunden werden, welche Art von Hautkrebs ein Patient hat. Hautkrebsarten können in Melanome oder Nicht-Melanome unterteilt werden. Melanome sind schwarze Hautkrebsarten, die Metastasen bilden und daher häufig zum Tod des Patienten führen, wenn sie nicht früh erkannt und behandelt werden. Nicht-Melanome sind weiße Hautkrebsarten, die weniger gefährlich sind, aber dennoch die Lebensqualität reduzieren können. Da es sowohl bei Melanomen als auch bei Nicht-Melanomen viele unterschiedliche Arten gibt [CRM+20], stellt die

Diagnose, welche Art von Hautkrebs ein Patienten hat, eine Multi-Klassen-Klassifikation dar, bei der die Klassen verschiedenen Krebsarten entsprechen.

Suresh et al. [SG21] betrachten einige Beispiele für die datengetriebene Diagnose kardiologischer Krankheiten wie z. B. Herzinfarkte oder Herzmuskelentzündungen. Dabei erwähnen die Autoren primär Beispiele für eine binäre Klassifikation, die für eine bestimmte Krankheit vorhersagt, ob ein Patient an dieser Krankheit leidet. Als Daten kommen i. d. R. Zeitreihendaten von EKG-Systemen in Betracht (EKG: Elektrokardiogramm). In wenigen Beispielen können auch noch Daten aus Genomikdatenbanken oder Textdaten aus elektronischen Krankenakten die zu analysierende Datenmenge anreichern.

Weitere in der Literatur prominente Beispiele für datengetriebene medizinische Diagnosen betreffen neurologische Krankheiten. Khan et al. [KU15] untersuchen elf wissenschaftliche Artikel für die Nutzung maschineller Lernverfahren zur Diagnose der Alzheimer-Krankheit. Maitín et al. [MGM20] betrachten analog neun Artikel für die Parkinson-Krankheit. In den meisten Fällen wird erneut eine binäre Klassifikation verwendet, die vorhersagt, ob die jeweils betrachtete Krankheit vorliegt oder nicht. Als Datenquellen werden häufig Zeitreihendaten einer Elektroenzephalographie (EEG), Elektroneurographie (ENG) oder Elektromyographie (EMG) verwendet. In einigen Beispielen werden auch mittels Tomographien des Gehirns aufgenommene Bilddaten analysiert.

3.3.1 Heterogene Datenformate und isolierte Datensilos

Wie in Tabelle 3.3 ersichtlich ist, müssen auch bei datengetriebenen medizinische Diagnosen Daten heterogener Formate analysiert werden. Dies schließt insbesondere unterschiedliche Ausprägungen von Bilddaten, Sensordaten, Zeitreihendaten, Genomikdatenbanken und Textdaten ein. Ebenso wird ein Großteil der Daten in einzelnen Studien erhoben. Da es sich i. d. R. um sensitive Patientendaten handelt, dürfen diese nicht beliebig an andere Personen oder Organisationen weitergegeben werden. Es handelt sich folglich analog zur Definition in Abschnitt 3.2.1 um einzelne isolierte Datensilos, auf die nahezu ausschließlich im Rahmen der jeweiligen Studie Zugriff gewährt werden darf. Lediglich stark aggregierte und anonymiserte Daten dürfen der Öffentlichkeit zur Verfügung gestellt werden. Dies erschwert allerdings den Vergleich von Daten und Ergebnissen mehrerer Studien [KU15; MGM20].

3.3.2 Datencharakteristika als Ursachen für Bias in Daten

Die in Abschnitt 3.2.2 diskutierten Datencharakteristika, die verschiedene Ausprägungen von Bias in Daten verursachen, treten ebenso in Anwendungsfällen datengetriebener medizinischer Diagnosen auf. Tabelle 3.4 gibt eine Übersicht, für welche Anwendungsfälle und in welchen Literaturquellen jeweils über die verschiedenen Ausprägungen von Bias in Daten berichtet wird.

Alle der betrachteten Übersichtsartikel [CRM+20; KU15; MGM20; SG21] berichten davon, dass die Daten eine **Klassenungleichverteilung** aufweisen. Bei Anwendungsfällen der Hautkrebsdiagnose stellt sich dies als besonders schweres Problem dar [CRM+20]. Im Allgemeinen haben weniger als 10 % der Patienten eine Variante von Melanomen, also von schwarzem Hautkrebs. Daher sind diese Krebsarten in den verfügbaren Daten im Vergleich zu den verschiedenen weißen Nicht-Melanomen meistens unterrepräsentiert. In diesem Fall versuchen Klassifikationsalgorithmen vor allem die Muster für Nicht-Melanome zu lernen, sodass die gelernten Klassifikationsmodelle einen Bias hinsichtlich dieser Mehrheitsklassen besitzen. Die Minderheitsklassen der Melanome werden dann von den Algorithmen teilweise ignoriert und anschließend häufig von den Klassifikationsmodellen falsch vorhergesagt. Melanome sind jedoch bösartig und können Metastasen bilden. Sie sind tödlicher als andere Krebsarten und müssen daher genauer vorhergesagt werden können.

Sowohl die **Gruppenheterogenität** als auch die **Gruppenungleichverteilung** werden von den Artikeln zur Diagnose von Hautkrebs [CRM+20], kardiologischer Krankheiten [SG21] und Parkinson-Erkrankungen [MGM20] als Problem diskutiert. Bei der Hautkrebsdiagnose nach Chan et al. [CRM+20] entsprechen bspw. Patienten verschiedener Ethnizitäten bzw. Hautfarben solchen heterogenen und in Daten ungleichmäßig verteilten Gruppen. Die Autoren stellen fest, dass sich die Vorhersageleistung von Klassifikationsmodellen für die Hautkrebsdiagnose zwischen Patienten verschiedener Hautfarben deutlich unterscheidet. Werden für die Klassifikation Bilddaten der Haut verwendet, werden als Features innerhalb der Klassenmuster häufig die Farbe der von Krebs befallenen Hautbereiche und der Kontrast zur umgebenden gesunden Haut herangezogen. Die Farbe und der Kontrast, die ein bildgebendes System für den von Krebs befallenen Hautbereich erkennt, können jedoch selbst bei der gleichen Krebsart unterschiedlich sein, wenn die Farbe der umgebenden

Tabelle 3.4: Auftreten verschiedener Ursachen für Bias in den Daten der untersuchten Anwendungsfälle medizinischer Diagnosen.

Art von Bias	**Anwendungsfälle und Literaturquellen**
Klassenungleichverteilung	Hautkrebs [CRM+20], kardiologische Krankheiten [SG21], Alzheimer-Erkrankung [KU15], Parkinson-Erkrankung [MGM20].
Gruppenheterogenität	Hautkrebs [CRM+20], kardiologische Krankheiten [SG21], Parkinson-Erkrankung [MGM20].
Gruppenungleichverteilung	Hautkrebs [CRM+20], kardiologische Krankheiten [SG21], Parkinson-Erkrankung [MGM20].
Falsch dokumentierte Labels	Kommen selten vor, nur bei Alzheimer-Erkrankungen berichten Khan et al. [KU15], dass Labels nicht überprüft werden.
Kleine Datenmenge	Hautkrebs [CRM+20], Alzheimer-Erkrankung [KU15], Parkinson-Erkrankung [MGM20] sowie allgemein bei seltenen Krankheiten [Hum12].

gesunden Haut unterschiedlich ist. Um diese Unterschiede abzudecken, müssen Klassifikationsalgorithmen für jede Hautfarbe unterschiedliche Klassenmuster lernen. Darüber hinaus betrachten sehr viele der 70 von Chan et al. [CRM+20] untersuchten Artikel Bilddaten von hellhäutigen Patienten. Für Patienten mit dunklerer Haut sind nur sehr wenige Bilddaten verfügbar. Daher sind auch die für diese Menschen mit dunklerer Haut spezifischen Klassenmuster in den Daten unterrepräsentiert. Dies bedeutet, dass ein mit diesen Daten gelerntes Klassifikationsmodell die für dunklere Hautfarben spezifischen Muster oft ignoriert und dann nicht in der Lage ist, die Hautkrebsarten korrekt für Patienten der entsprechenden Ethnizitäten vorherzusagen. Diese Problematik wird in der Literatur häufig auch als ethnischer Bias bezeichnet [MMS+21; SG21].

Eine weitere Ursache für eine Gruppenheterogenität und Gruppenungleichverteilung in Daten kann ein geschlechtsspezifischer Bias sein [SG21]. Ein in der Medizin diskutiertes Problem ist, dass sich die Symptome für bestimmte gefährliche Krankheiten zwischen Männern und Frauen unterscheiden. Dies ist ein häufiges Problem bei der Diagnose kardiologischer Krankheiten [SG21], das

nach Baggio et al. [BCF+13] ebenso in der Onkologie, bei Lebererkrankungen und Osteoporose auftritt. Beispielsweise haben Männer bei einem beginnenden Herzinfarkt starke Brustschmerzen als häufigstes Symptom, während Frauen eher unter Müdigkeit, Schwindel oder Magenschmerzen leiden [WS19]. Algorithmen des maschinellen Lernens können zwar die Diagnosedaten von Patienten nutzen, um Muster für typische Symptome eines Herzinfarkts zu lernen. Die Muster zur Diagnose eines Herzinfarkts können sich aber aufgrund der unterschiedlichen Symptomatik signifikant zwischen Frauen und Männern unterscheiden. Da für Frauen weniger Diagnosedaten zur Verfügung stehen als für Männer [WS19], lernen die Algorithmen in der Regel hauptsächlich die Muster für Männer, während sie die für Frauen als weniger wichtig ansehen.

Bei Parkinson-Erkrankungen unterscheiden sich die Symptome und Muster in den Daten je nach Fortschritt der Erkrankung und damit indirekt nach den Altersgruppen der Patienten [MGM20]. Betrouni et al. [BDC+19] berücksichtigen diesen Aspekt, indem sie die Patienten in fünf Gruppen mit unterschiedlich fortgeschrittenem Krankheitsverlauf einteilen. Allerdings weisen ihre Daten eine Gruppenungleichverteilung auf. Während in den drei Gruppen mit geringem bis mittlerem Krankheitsfortschritt 28, 33 und 43 Patienten enthalten sind, beinhalten die beiden Gruppen mit fortgeschrittenem Verlauf nur fünf bzw. neun Patienten. Die Autoren berichten, dass ihr Klassifikationsmodell eine schlechtere Vorhersageleistung für diese beiden seltenen Gruppen aufweist.

Das Problem der **falsch dokumentierten Klassenlabels** wird in den Artikeln zu datengetriebenen medizinischen Diagnosen nur selten erwähnt. In diesen Anwendungsfällen dokumentieren Ärzte ihre Diagnosen in der Regel erst nach umfassenden medizinischen Untersuchungen. Folglich kann angenommen werden, dass die Dokumentation der Klassenlabels in den Daten in einem Großteil der Dateninstanzen korrekt ist. Allerdings berichten Khan et al. [KU15], dass bei den meisten von ihnen untersuchten Studien zur datengetriebenen Diagnose von Alzheimer-Erkrankungen die Patientendaten nicht von unabhängigen Experten überprüft wurden. Hier kann nicht in jedem Fall sicher davon ausgegangen werden, dass jedes einzelne Klassenlabel, also jede einzelne dokumentierte medizinische Diagnose, korrekt ist.

Viele einzelne Studien, die datengetriebene Klassifikationsansätze für medizinische Diagnosen evaluieren, arbeiten mit einer vergleichsweise geringen

Anzahl an Studienteilnehmern und damit mit einer ebenso **kleinen Datenmenge** [CRM+20; KU15; MGM20]. Maitin et al. [MGM20] berichten bspw., dass an den meisten der von ihnen untersuchten Studien weniger als 50 Probanden beteiligt waren, von denen nur ca. die Hälfte eine Parkinson-Erkrankung hatten. Diese kleine Menge an Studienteilnehmern und Daten kann zu einer Überanpassung der ML-Modelle führen [KU15]. Ebenso kann sie einen Repräsentations-Bias bzw. eine Stichprobenverzerrung mit sich bringen [SG21]. Es ist nicht gesichert, ob die Verteilungen und Eigenschaften der jeweils verwendeten Daten mit denen der durch diese Stichproben repräsentierten Gesamtpopulation, also bspw. der Bevölkerung einer bestimmten Region bzw. sogar der Weltbevölkerung, übereinstimmen. Folglich ist auch unklar, ob die jeweiligen Studienergebnisse auf diese Gesamtpopulation übertragbar sind. Weiterhin sind im Allgemeinen nur sehr wenige Diagnosedaten für Patienten verfügbar, die an sog. seltenen Krankheiten leiden. So treten Krankheiten wie Mukoviszidose nur bei etwa einem von 15 000 Menschen auf [Hum12]. Allerdings wären gerade bei solchen Krankheiten eine größere Datenmenge und damit besser funktionierende datengetriebene Diagnosemethoden wünschenswert, da die meisten seltenen Krankheiten sehr gefährlich oder gar tödlich sind.

3.3.3 Datencharakteristika in nichtstationären Umgebungen

Bei datengetriebenen medizinischen Diagnosen führen insbesondere die zeitlichen Veränderungen der Erkrankungen von Patienten sowie die Veränderungen der damit einhergehenden Symptome zu nichtstationären Datenumgebungen. Dabei treten erneut Datensatzverschiebungen als Probleme für Datenanalysen auf. Bei allen in diesem Abschnitt diskutierten Anwendungsfällen für medizinische Diagnosen handelt es sich um graduell fortschreitende Krankheitsverläufe [CRM+20; KU15; MGM20; SG21]. Dies bedeutet, dass die Symptome für eine Krankheit zunächst sehr gering ausfallen und daher häufig nicht wahrgenommen werden. Tatsächlich wird die Krankheit erst nach einer gewissen Zeit, in der sich die Symptome nach und nach verschlimmern und deutlicher werden, durch eine medizinische Diagnose erkannt [CRM+20; KU15; MGM20]. In datengetriebenen Ansätzen wird in den entsprechenden Patientendaten erst mit der eindeutigen Diagnose das Klassenlabel zur jeweils diagnostizierten Krankheit geändert. Über den gesamten Datensatz hinweg ändern sich also

sowohl die Symptome als auch die Klassenlabels einzelner Dateninstanzen graduell mit der Zeit. Dies kann analog zu dem Anwendungsfall von Spieß et al. [SRWM22] nach und nach zu einer Änderung der Entscheidungsgrenzen für einzelne Klassenmuster sowie auch zu einer Änderung der Verteilung der Muster und damit zu Datensatzverschiebungen in den Daten führen.

Dabei können sich insbesondere die Klassenmuster in den Trainingsdaten $\mathcal{D}_{train}$ von denen in den Testdaten $\mathcal{D}_{test}$, auf die das trainierte Klassifikationsmodell angewendet wird, unterscheiden. Dies ist z. B. der Fall, wenn in Trainings- und Testdaten Patienten mit unterschiedlichen Krankheitsfortschritten beschrieben werden [MGM20]. Wird z. B. ein Klassifikationsmodell mit Trainingsdaten $\mathcal{D}_{train}$ von Patienten im späten Krankheitsstadium trainiert, ist dieses Modell typischerweise nicht gut auf Testdaten $\mathcal{D}_{test}$ von Patienten im frühem Stadium generalisierbar, da sich die für eine Krankheit charakteristischen Symptome und damit die Klassenmuster zwischen $\mathcal{D}_{train}$ und $\mathcal{D}_{test}$ unterscheiden. Somit trifft das Modell häufig fehlerhafte Vorhersagen für Patienten im frühen Krankheitsstadium. Allerdings sollten die in diesem Abschnitt betrachteten und typischerweise gefährlichen Krankheiten möglichst frühzeitig erkannt werden, was datengetriebene Ansätze aufgrund der graduellen Datensatzverschiebungen häufig nicht leisten können.

Kapitel 4

Effekte datengetriebener Ansätze

In dieser Arbeit werden als rein datengetriebene Ansätze solche betrachtet, die keine der in Abschnitt 2.2 beschriebenen Arten formal repräsentierten Domänenwissens nutzen. In der Literatur werden viele datengetriebene Ansätze zur Datenakquise, Datenvorbereitung und Datanenanalyse vorgeschlagen, die jeweils einzelne der in Abschnitt 3.2 beschriebenen Datencharakteristika adressieren können. Allerdings werden in den entsprechenden Artikeln immer nur isolierte Effekte auf einige wenige Datencharakteristika untersucht, sodass die Effekte auf jeweils andere Datencharakteristika unklar bleiben. Beispielsweise schlägt die Literatur eine Vielzahl an Techniken zur Balancierung eines Datensatzes vor [CBHK02; GFB+12; HG09; SWK09; WY12], um die in Abschnitt 3.2.2 erläuterte Klassenungleichverteilung zu adressieren. Hirsch et al. [HRM19] zeigen allerdings, dass diese Techniken die Probleme mit anderen Ursachen für Bias in Daten sogar noch verstärken.

In diesem Kapitel werden für verschiedene datengetriebene Ansätze deren Effekte auf die jeweils relevanten der in Abschnitt 3.2 beschriebenen Datencharakteristika diskutiert. Dies ist nachfolgend in die Schritte Datenakquise und Datenexploration (Abschnitt 4.1), Datenvorbereitung (4.2) sowie Datenanalyse und Modellbildung (4.3) des generischen Vorgehensmodells für

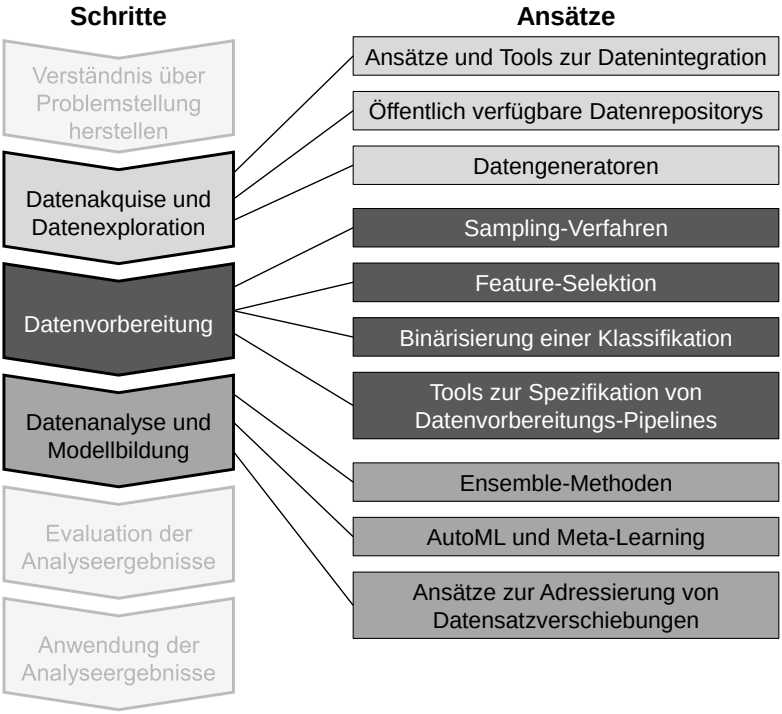


Abbildung 4.1: In diesem Kapitel diskutierte rein datengetriebene Ansätze für die Schritte Datenakquise und Datenexploration, Datenvorbereitung sowie Datenanalyse und Modellbildung.

Datenanalyseprozesse unterteilt. Abbildung 4.1 zeigt einen Überblick, welche datengetriebenen Ansätze jeweils in den drei Schritten des Vorgehensmodells diskutiert werden. Die Grundlage stellen mehrere vergleichende Studien dar, die in den vom Autor dieser Habilitationsschrift betreuten Projekten entstanden sind und in diesem Kapitel durch eigene Diskussionen ergänzt werden. Abschnitt 4.4 fasst schließlich die wichtigsten Erkenntnisse aus der Diskussion zusammen, wobei in diesem Kapitel insbesondere folgende Kernaussage belegt wird: Datengetriebene Ansätze, die kein Domänenwissen nutzen, reichen nicht aus, um alle für industrielle Anwendungsfälle spezifischen Datencharakterisika sowie deren Wechselwirkungen zu adressieren. In vielen Fällen führen sie sogar zu einer Verschlechterung der Vorhersageleistung der trainierten ML-Modelle.

4.1 Datenakquise und Datenexploration

Im Schritt Datenakquise und Datenexploration werden nachfolgend insbesondere Ansätze zur Datenintegration sowie öffentlich verfügbare Datenrepositorys und Datengeneratoren diskutiert.

4.1.1 Ansätze und Tools zur Datenintegration

Eine Vielzahl an Ansätzen und Software-Tools erlauben die Akquise von Daten aus heterogenen Datenquellen und die Integration dieser Daten in ein Datenformat [LN07]. Beispielhafte Ansätze sind föderierte Informationssysteme [BKLW99; SL90] sowie ETL-Tools [KRRT98; LN07] (Extraktion, Transformation, Laden von Daten). Weiterhin schlägt die Literatur ontologiebasierte Ansätze zur Datenintegration vor [She99; WD10; WVV+01]. Da diese aber das in Ontologien abgebildete Domänenwissen nutzen, sind sie nicht in diesem Abschnitt, sondern in Abschnitt 5.1 Gegenstand der Diskussion.

Die genannten Ansätze zu föderierten Systemen oder ETL-Tools sind grundlegend zur Integration einiger der in Tabelle 3.1 angegebenen Datenformate industrieller Anwendungsfälle geeignet. Insbesondere können die Ansätze strukturierte Daten aus MES-, SCADA-, ERP- oder PLM-Informationssystemen sowie Sensordaten und Zeitreihendaten integrieren. Auch eignen sie sich teilweise für semi-strukturierte Daten, wie JSON-Dokumente oder Maschinenlogs, sowie für unstrukturierte Textdaten. Allerdings sind die genannten Ansätze meistens nicht geeignet, um proprietäre Dateiformate zu integrieren [Rei17], wie sie z. B. im Bereich der Produktentwicklung vorkommen [ZRKM20; ZRKM21]. Ebenso erfordern die Ansätze und Tools zur Datenintegration immer noch Kenntnisse bzgl. IT und Datenverarbeitung, damit sie zur Akquise von Daten mit heterogenen Formaten genutzt werden können [Grö18; Grö21]. Hierbei kommen speziell Domänenexperten an ihre Grenzen, da sie i. d. R. nicht über das nötige Know-How bzgl. IT und Datenverarbeitung verfügen.

Ein weiterer Nachteil der Ansätze zur Datenintegration ist, dass sie nicht direkt das Problem lösen, dass Daten in voneinander isolierten Datensilos gespeichert sind. Dieses Problem ist in der Regel organisatorischer Natur, da sich einzelne Organisationseinheiten eines Unternehmens häufig eigenständig um ihre Informationssysteme kümmern [Grö18; Grö21]. Mit ihren eher informati-

onstechnischen Lösungen können Ansätze zur Datenintegration dieses Problem daher nicht lösen. Folglich können diese Ansätze in vielen Fällen nur Daten aus den Informationssystemen einzelner Organisationseinheiten integrieren.

4.1.2 Öffentlich verfügbare Datenrepositorys

Eine Reihe von Repositorys wie KEEL [AFL+11], Kaggle[1], openML [VRBT13], sowie das UCI ML Repository[2] stellen zusammen mehrere tausend Datensätze zur Verfügung. Diese Datensätze können für diverse Untersuchungen genutzt werden, z. B. zur Evaluation spezifischer Verfahren zur Datenvorbereitung oder Datenanalyse. Allerdings weisen sie nur einzelne der Datencharakteristika auf, die in realen industriellen Anwendungsfällen vorkommen. Folglich ist unklar, ob die Ergebnisse der Untersuchungen mit diesen Datensätzen auf die Daten industrieller Anwendungsfälle übertragbar sind. Bezüglich der in Abschnitt 3.2.1 beschriebenen heterogenen Datenformate finden sich in den Repositorys überwiegend tabellarisch strukturierte Daten sowie Zeitreihendaten oder Textdaten. Allerdings finden sich keine proprietären unstrukturierten Dateiformate und es sind auch kaum Daten verfügbar, die typische Eigenschaften und Strukturen gängiger MES-, ERP- oder PLM-Informationssysteme aufweisen.

Hinsichtlich der in Abschnitt 3.2.2 beschriebenen Datencharakteristika, die ursächlich für verschiedene Arten von Bias in Daten sind, finden sich in den Repositorys einige Datensätze mit einer Klassenungleichverteilung (siehe Tabelle 4.1). Beispielsweise betrachten ca. 40 der Datensätze aus openML eine Multi-Klassen-Klassifikation mit mehr als zehn Klassen und einem Gini-Koeffizient [Cow11; Gin21] von über 30 %, was zumindest auf eine moderate Klassenungleichverteilung hindeutet [HRT+23]. Weiterhin entsprechen ca. tausend Datensätze einer eher kleinen Datenmenge mit weniger als 1000 Instanzen sowie 100 oder mehr Features. Diese haben daher analog zum Anwendungsfall der Fehlerdiagnose im EoL-Prüffeld von Hirsch et al. [HRM19] ein Problem mit einer möglichen Überanpassung der trainierten ML-Modelle.

Allerdings wird nur bei sehr wenigen der Datensätze aus den Repositorys explizit beschrieben, dass sie heterogene domänenspezifische Entitätsgruppen

[1] Kaggle Datensätze: `https://www.kaggle.com/datasets`
[2] UCI ML Repository: `https://archive.ics.uci.edu/ml`

Tabelle 4.1: Datencharakteristika aus Tabelle 3.2, die Datensätze von öffentlich verfügbaren Repositorys oder von existierenden Datengeneratoren aufweisen oder nicht aufweisen können.

	Klassen-ungleich-verteilung	**Gruppen-hetero-genität**	**Gruppen-ungleich-verteilung**	**Falsche Labels**	**Kleine Daten-menge**
Datenrepositorys	ja	nein	nein	unklar	ja
Datengeneratoren	ja	nein	nein	ja	ja

repräsentieren [TRSM23]. Zwei Beispiele sind der Datensatz *adult*[1] sowie der Datensatz *COMPAS*[2], bei denen Menschen verschiedener Geschlechter bzw. Hautfarben die Gruppen darstellen. Diese und ähnliche Datensätze enthalten jedoch in der Regel nur zwischen zwei oder vier solcher Gruppen [ULP19; WLL21]. In industriellen Anwendungsfällen entsprechen die Gruppen häufig mindestens zehn bis sogar 100 Typen von Produkten [HRM20a; SRYD14]. Dabei ist sowohl die Anzahl als auch die Diversität der gruppenspezifischen Klassenmuster in den Datensätzen industrieller Anwendungsfälle größer als bei den Datensätzen aus öffentlich verfügbaren Repositorys [TRSM23]. Zudem sind die vielen Gruppen in Datensätzen industrieller Anwendungsfälle häufig deutlich ungleichmäßiger verteilt. Bei falsch dokumentierten Klassenlabels ist meistens unklar, inwieweit diese in den Datensätzen aus den Repositorys enthalten sind, da die Klassenlabels ähnlich wie in medizinischen Anwendungsfällen [KU15] selten von unabhängigen Experten überprüft wurden.

Bezüglich der in Abschnitt 3.2.3 beschriebenen Datensatzverschiebungen finden sich lediglich in openML 18 Datensätze [VRBT13], die explizit zur Untersuchung von Konzeptverschiebungen geeignet sind. Allerdings finden sich in den Repositorys keine Datensätze, die die in industriellen Anwendungsfällen häufig vorkommende Kombination von Konzept- und Kovariatenverschiebungen aufweisen [NLDD18; SRWM22]. Somit kann mit diesen Datensätzen auch nicht die in Abschnitt 3.2.3 angesprochene Trade-off-Beziehung zwischen beiden Datensatzverschiebungen untersucht werden.

[1] Datensatz *adult*: `https://archive.ics.uci.edu/ml/datasets/adult`
[2] Datensatz *COMPAS*: `https://www.kaggle.com/datasets/danofer/compass`

4.1.3 Datengeneratoren

Zusätzlich existieren einige Datengeneratoren, mit denen neue Datensätze erzeugt werden können [TRSM23]. Datengeneratoren aus dem Bereich der *Data Augmentation* [RHW21] wenden bspw. *Generative Adversarial Networks* [GPM+14; PWV16] auf einen bereits existierenden Datensatz an, um neue Dateninstanzen zu erzeugen, die eine ähnliche statistische Verteilung aufweisen wie der existierende Datensatz. Andere Datengeneratoren erzeugen hingegen neue Daten, ohne auf einen existierenden Datensatz aufbauen zu müssen. Bei einer Gruppe dieser Datengeneratoren muss zunächst ein Datenmodell spezifiziert werden, für das anschließend Daten erzeugt werden. Diese Generatoren können Daten für spezifische Anwendungsfälle erzeugen, z. B. zur Betrugserkennung [LKJ02] oder für das Gesundheitswesen [DC19]. Eine weitere Gruppe umfasst Datengeneratoren, die es erlauben, die Anzahl der Dateninstanzen, die Anzahl der Klassen und die Anzahl der Features für den zu erzeugenden Datensatz einzustellen [FLSB11; FS18; IZFZ19; SB21; SH05].

Sämtliche Datengeneratoren können lediglich Datensätze erzeugen, die nur einzelne der Datencharakteristika industrieller Anwendungsfälle aufweisen [TRSM23]. Die Generatoren sind insbesondere in der Lage, Datencharakteristika zu erzeugen, die rein auf statistischen Verteilungen beruhen. Damit ist hinsichtlich der in Abschnitt 3.2.2 beschriebenen Ursachen für Bias in Daten die Erzeugung einer kleinen Datenmenge sowie Daten mit verschiedenen Ausprägungen einer Klassenungleichverteilung oder mit unterschiedlichen Verteilungen falscher Klassenlabels möglich (siehe Tabelle 4.1). Um Daten mit anderen Charakteristika wie insbesondere die Heterogenität und Ungleichverteilung realer Entitätsgruppen zu generieren, wird Domänenwissen über die Eigenschaften dieser Gruppen benötigt [HRT+23]. Dieses Domänenwissen wird bei den existierenden Datengeneratoren allerdings nicht berücksichtigt, sodass sie auch entsprechende Datencharakteristika nicht erzeugen können.

Ebenso gibt es nur Generatoren, die jeweils eine der beiden in Abschnitt 3.2.3 beschriebenen Arten von Datensatzverschiebungen erzeugen können. Somit kann mit den Datensätzen erneut keine Trade-off-Beziehung zwischen gleichzeitig auftretenden Konzept- und Kovariatenverschiebungen untersucht werden.

Tabelle 4.2: Auswirkungen verschiedener Ansätze zur Datenvorbereitung auf die Ursachen für Bias in Daten: + Positiver Effekt; 0 kein bzw. neutraler Effekt; – negativer Effekt.

Ansatz	**Klassen-ungleich-verteilung**	**Gruppen-hetero-genität**	**Gruppen-ungleich-verteilung**	**Falsche Labels**	**Kleine Daten-menge**
Undersampling (RUS)	+	0	–	–	–
Oversampling (ROS, SMOTE)	+	0	0	–	0
Feature-Selektion	0	–	0	0	+
Binärisierung mit OvO	+	0	0	0	–
Binärisierung mit OvA	–	–	–	0	0
Tools für Pipelines zur Datenvorbereitung	–	0	–	0	–

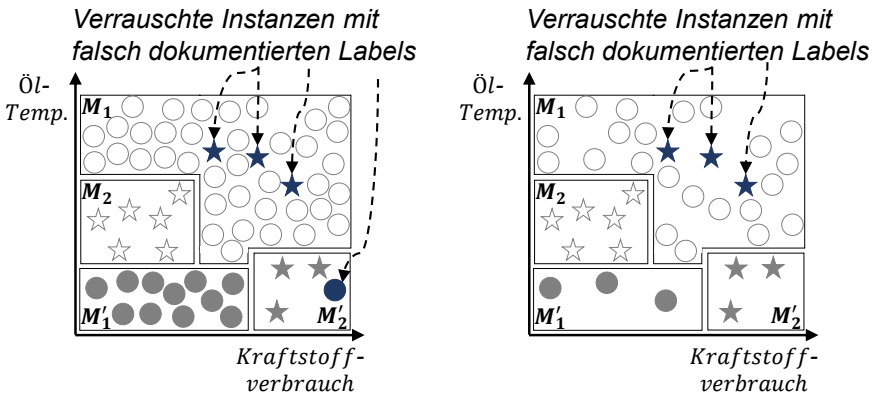
4.2 Datenvorbereitung

Im Rahmen dieser Habilitationsschrift wurden existierende Ansätze zur Datenvorbereitung untersucht, die insbesondere einzelne Ursachen für Bias in Daten adressieren. Auch hier ist die Regel, dass einzelne Ansätze zur Datenvorbereitung nur spezifische Ursachen für Bias adressieren, während sie aber die Probleme mit jeweils anderen Datencharakteristika sogar noch verschlimmern können. Dieser Aspekt wird nachfolgend für die gängigsten Ansätze zur Datenvorbereitung näher beleuchtet. Die Abschnitte 4.2.1 bis 4.2.3 diskutieren Verfahren für spezielle Datenvorbereitungsoperationen zum Sampling, zur Feature-Selektion und für die Binärisierung einer Multi-Klassen-Klassifikation. Abschnitt 4.2.4 betrachtet schließlich Ansätze und Software-Tools, mit denen über Datenvorbereitungs-Pipelines der Kontroll- oder Datenfluss mehrerer Datenvorbereitungsoperationen spezifiziert werden kann. Tabelle 4.2 fasst zusammen, welche Auswirkungen die jeweiligen Ansätze auf die verschiedenen in Abschnitt 3.2.2 beschriebenen Ursachen für Bias in Daten haben.

4.2.1 Sampling-Verfahren

Mehrere Übersichtsartikel [GFB+12; HG09; SWK09; WY12] diskutieren spezielle Sampling-Verfahren, mit denen ein Datensatz balanciert werden kann. Diese Verfahren können eine Klassenungleichverteilung adressieren, indem sie dafür sorgen, dass die Klassen gleichmäßiger verteilt sind. Repräsentative Verfahren sind *Random Undersampling* (RUS), *Random Oversampling* (ROS) und die *Synthetic Minority Oversampling Technique* (SMOTE) [CBHK02]. Nachfolgend wird auf Basis der Studie von Hirsch et al. [HRM19] diskutiert, welche Auswirkungen diese Verfahren auf weitere Ursaschen für Bias in Daten haben.

Random Undersampling (RUS) entfernt aus dem Datensatz eine zufällige Auswahl an Dateninstanzen der Mehrheitsklassen [HG09]. Abbildung 4.2 zeigt das Ergebnis einer Anwendung von RUS auf den in Abbildung 3.5 dargestellten Datensatz. Die Klassenungleichverteilung wird insofern adressiert, dass durch RUS die relativen Häufigkeiten der Mehrheitsklassen sinken, während die der Minderheitsklassen steigen. Um einen einigermaßen balancierten Datensatz zu erhalten, müssen aber speziell bei einer hohen Klassenungleichverteilung sehr viele Dateninstanzen der Mehrheitsklassen aus dem ursprünglichen Datensatz entfernt werden. Dadurch wird die in industriellen Anwendungsfällen zur Verfügung stehende kleine Datenmenge noch kleiner, was somit die Wahrscheinlichkeit der Überanpassung trainierter ML-Modelle erhöht. Es kann sogar das Problem der Gruppenungleichverteilung verschärfen, da gruppenspezifische Muster, die im ursprünglichen Datensatz viele Dateninstanzen enthalten, im reduzierten Datensatz aber selten vorkommen können und daher von Lernalgorithmen nach der Anwendung von RUS ignoriert werden. In Abbildung 4.2 ist dies anhand des Musters M_1' ersichtlich, das im ursprünglichen Datensatz durch elf Instanzen vertreten ist. Nach der Anwendung von RUS wird M_1' nur noch durch drei Instanzen repräsentiert, sodass Lernalgorithmen einzelne dieser drei Instanzen sogar den Mustern M_2 oder M_2' und damit der falschen Klasse 'Stern' zuordnen können. Weiterhin erhöht die Anwendung von RUS häufig auch innerhalb einzelner Muster die relative Häufigkeit verrauschter Instanzen mit falsch dokumentierten Labels. Die drei verrauschten Instanzen der Klasse 'Stern' sind im Muster M_1 nun z. B. durch deutlich weniger Instanzen der Klasse 'Kreis' umgeben. Dadurch steigt die Wahrscheinlichkeit, dass Lernalgorithmen für diese verrauschten Instanzen dedizierte Muster lernen.



(a) Datensatz vor RUS.

(b) Datensatz nach RUS.

Abbildung 4.2: Beispielhafte Anwendung von *Random Undersampling* (RUS) auf den Datensatz aus Abbildung 3.5. RUS entfernt zufällig Instanzen der Mehrheitsklasse 'Kreis', bis diese nur noch doppelt so häufig vorkommt wie die Minderheitsklasse 'Stern'.

Die Parametrisierung von RUS erlaubt es grundsätzlich, auch eine andere als die in Abbildung 4.2b dargestellte Verteilung der Klassen zu erzeugen. Hirsch et al. [HRM19] haben verschiedene Verteilungen untersucht und dabei festgestellt, dass bei allen einstellbaren Verteilungen die negativen Effekte bzgl. der Gruppenungleichverteilung, der falsch dokumentieren Labels und der Reduzierung der Datenmenge die positiven Effekte auf die Klassenungleichverteilung überwiegen. Sie untersuchten dazu für verschiedene Klassifikationsalgorithmen, inwieweit die Anwendung von RUS die Korrektklassifikationsrate (engl. *Accuracy*) für ihren Anwendungsfall der Fehlerdiagnose im EoL-Prüffeld verändern. Je nach Klassifikationsalgorithmus reduziert RUS die Korrektklassifikationsrate deutlich zwischen 27 und 39 %-Punkten [HRM19].

Random Oversampling (ROS) addressiert die Klassenungleichverteilung, indem es Kopien einer zufälligen Auswahl der Dateninstanzen von Minderheitsklasssen erzeugt [HG09], sodass die relative Häufigkeit dieser Minderheitsklassen steigt. Die *Synthetic Minority Oversampling Technique* (SMOTE) [CBHK02] erzeugt analog synthetische Dateninstanzen der Minderheitsklassen. Wie Abbildung 4.3 darstellt, entsprechen diese synthetischen Dateninstanzen Inter-

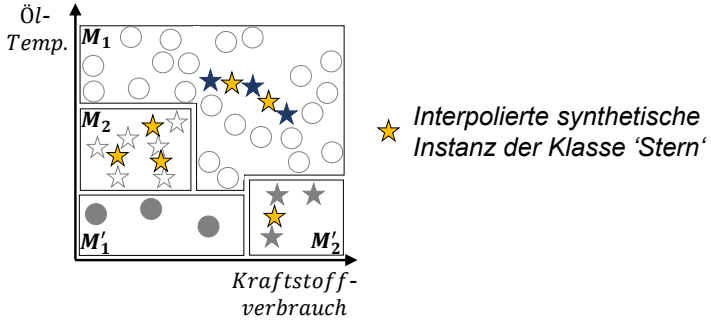


Abbildung 4.3: Beispielhafte Anwendung von SMOTE auf den in Abbildung 4.2b dargestellten Datensatz. SMOTE erzeugt hierbei mehrere synthetische Instanzen der Minderheitsklasse 'Stern' als Interpolationen zwischen existierenden Instanzen.

polationen zwischen bereits im ursprünglichen Datensatz enthaltenen Minderheitsinstanzen. Allerdings können ROS und SMOTE Kopien von Instanzen bzw. synthetische Instanzen erzeugen, die einem falsch dokumentierten Klassenlabel zugeordnet sind. Dies kann die relative Häufigkeit von verrauschten Instanzen in bestimmten Bereichen des Datensatzes erhöhen. In Abbildung 4.3 erhöht SMOTE z. B. die Anzahl der mit dem falschem Klassenlabel 'Stern' versehenen Instanzen im Muster M_1 von drei auf fünf. Dadurch werden sowohl die relative Häufigkeit als auch die Dichte dieser verrauschten Instanzen größer. Aus diesem Grund neigen Lernalgorithmen mit einer noch höheren Wahrscheinlichkeit dazu, dedizierte Muster für verrauschte Instanzen zu lernen. Hirsch et al. [HRM19] zeigen in ihrer Evaluation, dass die beiden Verfahren zum Oversampling aufgrund des geschilderten Problems auch die Korrektklassifikationsraten der mit den Daten trainierten ML-Modelle je nach Klassifikationsalgorithmus um Werte zwischen 2 und 45 %-Punkte verringern.

4.2.2 Statistische Feature-Selektion

Methoden zur statistischen Feature-Selektion bestimmen ein statistisches Maß, das angibt, wie gut einzelne Eingabe-Features $f_j \in \mathcal{F}$ eines Datensatzes zur Differenzierung zwischen den Labels $y_i \in \mathcal{Y}$ beitragen. Features, die nur geringfügig zur Differenzierung zwischen Labels beitragen, können vor der Anwendung

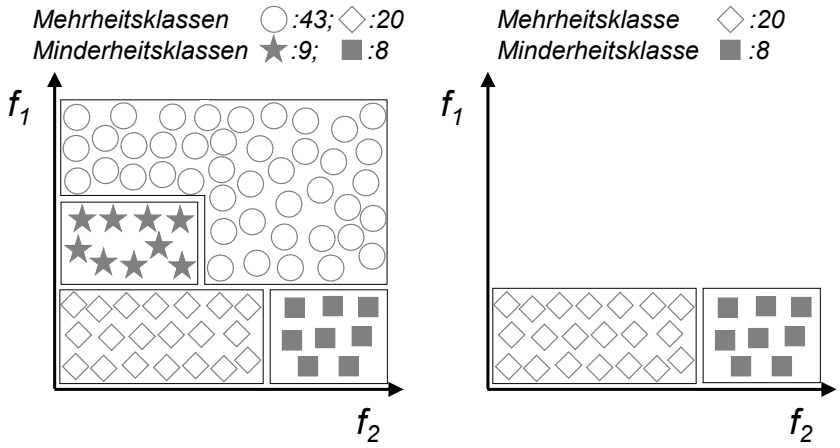
eines Lernalgorithmus aus dem Datensatz entfernt werden. Insbesondere wenn nur eine kleine Datenmenge zur Verfügung steht, kann eine Feature-Selektion durch die Reduktion der Anzahl an Features das Risiko der Überanpassung der trainierten ML-Modelle an die Daten reduzieren [HRM19].

Aufgrund der Gruppenheterogenität enthalten Daten industrieller Anwendungsfälle häufig viele komplexe Muster zur Charakterisierung der einzelnen Labels. Hirsch et al. [HRM19] zeigen, dass Methoden zur statistischen Feature-Selektion in solchen Fällen nur eine sehr kleine Teilmenge der Features $f_j \in \mathcal{F}$ mit ausreichender statistischer Sicherheit als irrelevant für die Klassifikation einstufen. So konnten Hirsch et al. [HRM19] mit dem gängigen Boruta-Algorithmus [KR10] lediglich zehn der insgesamt 115 Features ihres Datensatzes als irrelevant identifizieren und aus dem Datensatz entfernen. In der Folge führt Boruta zwar beim Klassifikationsalgorithmus Random Forest zu einer Verbesserung der Korrektklassifikationsrate von 5 %-Punkten, verringert den Wert dieser Metrik bei anderen Algorithmen aber um bis zu 19 %-Punkte.

4.2.3 Binärisierung einer Multi-Klassen-Klassifikation

Binärisierungstechniken reduzieren die Komplexität des Lernproblems einer Multi-Klassen-Klassifikation, indem sie die Trainingsdaten für mehrere binäre Klassifikationen aufteilen [FLG+13]. Die gängigsten Verfahren sind *One-vs-One* (OvO) und *One-vs-All* (OvA) [GFB+11]. Das Verfahren *OvO* trainiert für jedes mögliche Paar der Klassen, die im Datensatz vorkommen, ein binäres Klassifikationsmodell. Dabei wird je Klassenpaar eine echte Teilmenge des Datensatzes als Trainingsdaten verwendet. Diese Teilmenge enthält nur die Dateninstanzen, die zu den beiden betrachteten Klassen gehören. Abbildung 4.4 zeigt beispielhaft die Anwendung von OvO für einen Datensatz mit zwei künstlichen Features f_1 und f_2. In Abbildung 4.4a ist der Datensatz vor Anwendung von OvO mit zwei Mehrheitsklassen 'Kreis' und 'Raute' sowie zwei Minderheitsklassen 'Stern' und 'Viereck' zu sehen. Abbildung 4.4b zeigt exemplarisch für das Klassenpaar 'Raute' und 'Viereck', wie OvO den Datensatz zu einem binären Klassifikationsproblem zwischen diesen beiden Klassen aufteilt.

Da nach der Anwendung von OvO in jeder einzelnen Datenteilmenge nur zwei Klassen enthalten sind, gleicht das OvO-Verfahren teilweise die ursprünglich vorhandene Klassenungleichverteilung aus [HRM19]. In Abbildung 4.4b

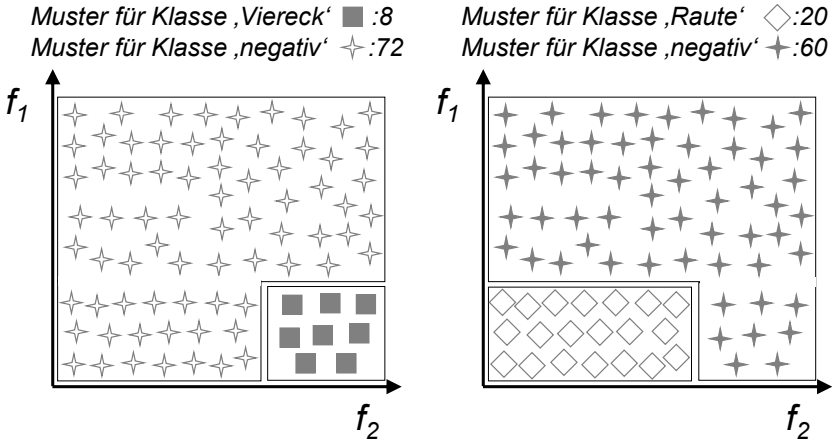


(a) Datensatz vor OvO. (b) OvO für Klassen 'Raute' und 'Viereck'.

Abbildung 4.4: Anwendung der Binärisierungstechnik *One-vs-One* (OvO) auf einen Datensatz mit zwei Features f_1, f_2 sowie vier Klassen 'Kreis', 'Stern', Raute' und 'Viereck'. Abbildung 4.4a zeigt den ursprünglichen Datensatz, während Abbildung 4.4b die Anwendung von OvO für die Klassen 'Raute' und 'Viereck' darstellt.

stehen den acht Instanzen der Minderheitsklasse 'Viereck' nur noch 20 Instanzen der Klasse 'Raute' gegenüber. Ebenso sind die 43 Instanzen der ursprünglichen Mehrheitsklasse 'Kreis' nicht mehr in der Datenteilmenge enthalten, sodass die Klassenungleichverteilung reduziert wurde. Ein wesentlicher Nachteil des OvO-Verfahrens ist, dass die jeweiligen Teilmengen der Trainingsdaten noch weniger Instanzen enthalten. Damit steigt das Risiko, dass die binären Klassifikationsmodelle an diese wenigen Dateninstanzen überangepasst sind.

Beim Verfahren *OvA* wird für jede im Datensatz enthaltene Klasse c_i ein Klassifikationsmodell mit dem gesamten ursprünglichen Datensatz trainiert. Dieser Datensatz wird zuvor aber so transformiert, dass alle Instanzen der anderen Klassen $c_j \neq c_i$ zu einer künstlichen Klasse 'negativ' zusammengefasst werden. Das mit diesem transformierten Datensatz trainierte Klassifikationsmodell trifft bei neuen Daten die binäre Entscheidung, ob diese neuen Daten der Klasse c_i angehören oder nicht [GFB+11]. Abbildung 4.5 illustriert die Anwendung von OvA auf den Datensatz aus Abbildung 4.4a. Während Abbildung 4.5a zeigt, wie



(a) OvA für Klasse 'Viereck'. (b) OvA für Klasse 'Raute'.

Abbildung 4.5: Beispielhafte Anwendung der Binärisierungstechnik *One-vs-All* (OvA) auf den Datensatz aus Abbildung 4.4a. Abbildung 4.5a zeigt die Anwendung von OvA für die Klasse 'Viereck' und Abbildung 4.5b für die Klasse 'Raute'.

OvA den Datensatz für ein binäres Klassifikationsproblem für die ursprüngliche Minderheitsklasse 'Viereck' transformiert, stellt Abbildung 4.5b dies für die ursprüngliche Mehrheitsklasse 'Raute' dar.

Im Gegensatz zu OvO verstärkt das Verfahren OvA häufig sogar das Problem der Klassenungleichverteilung [HRM19]. Dies liegt daran, dass jeder Klasse c_i in dem für sie transformierten Datensatz eine sehr große Anzahl an Instanzen der Klasse 'negativ' gegenübersteht. Diese Anzahl an Instanzen der Klasse 'negativ' entspricht der aufsummierten Anzahl der Instanzen aller anderen Klassen $c_j \neq c_i$. Insbesondere wenn die Klasse c_i eine Minderheitsklasse ist, ist die Anzahl an Instanzen der Klasse 'negativ' signifikant größer als die der Klasse c_i. Abbildung 4.5a verdeutlicht dieses Problem, da hier den acht Instanzen der Minderheitsklasse 'Viereck' nach der Anwendung von OvA insgesamt 72 Instanzen der Klasse 'negativ' gegenüberstehen. Allerdings können auch Klassen, die vor der Anwendung von OvA im Datensatz eine Mehrheitsklasse darstellen, nach OvA zu einer Minderheitsklasse werden. Ein entsprechendes Beispiel ist in Abbildung 4.5b dargestellt. Hier stehen der ursprünglichen Mehrheitsklasse

'Raute' mit ihren 20 Instanzen im transformierten Datensatz 60 Instanzen der Klasse 'negativ' gegenüber, sodass die Klasse 'Raute' nach OvA zur Minderheitsklasse wird. Im echten Datensatz von Hirsch et al. [HRM19] stellt sich dieser Umstand sogar noch schwerwiegender dar. Die häufigste Mehrheitsklasse wird in diesem Datensatz durch 95 der insgesamt 1050 Instanzen repräsentiert. Wird für diese häufigste Klasse OvA angewendet, stehen ihren 95 Instanzen folglich 955 und damit ca. zehnmal so viele Instanzen der Klasse 'negativ' gegenüber. In diesem Fall kann es passieren, dass Lernalgorithmen sogar diese im ursprünglichen Datensatz am häufigsten vorkommende Klasse ignorieren.

Analog bringt OvA weitere Probleme mit sich, wenn der ursprüngliche Datensatz eine Gruppenheterogenität und eine Gruppenungleichverteilung aufweist [HRM19]. Wie in Abschnitt 3.2.2 beschrieben enthält der Datensatz in diesem Fall pro Klasse mehrere Muster, die ungleichmäßig verteilt sind. Nach der Anwendung von OvA können insbesondere die Muster für seltene Gruppen von sehr vielen Instanzen der Klasse 'negativ' umgeben sein. Dies macht es Lernalgorithmen noch schwerer, die genauen Entscheidungsgrenzen für die spezifischen Muster der seltenen Gruppen zu lernen. Hirsch et al. [HRM19] zeigen, dass die diskutierten Probleme zu einer deutlichen Verschlechterung der Vorhersageleistung der trainierten Klassifikationsmodelle führen, wenn OvA auf die Daten angewendet wird. So reduziert sich die Korrektklassifikationsrate je nach Klassifikationsalgorithmus um Werte zwischen 10 und 32 %-Punkte.

4.2.4 Tools zur Spezifikation von Datenvorbereitungs-Pipelines

Viele verfügbare Software-Tools erlauben es, Datenvorbereitungs-Pipelines zu spezifizieren, die den Kontroll- oder Datenfluss mehrerer Datenvorbereitungsoperationen festlegen, um einen Datensatz $\mathcal{D}$ in einen Datensatz $\mathcal{D}'$ mit veränderten Datencharakteristika zu transformieren (siehe Abbildung 4.6). Gängige Datenvorbereitungsoperationen, die in solchen Pipelines genutzt werden können, unterstützen bspw. die oben genannten Verfahren zum Sampling, zur Feature-Selektion bzw. zum Feature-Engineering oder zur Binärisierung eines Datensatzes. Andere Beispiele solcher Operationen führen Datenformatkonvertierungen durch, bereinigen Daten hinsichtlich der Datenqualität [KRM19b], generieren neue Daten (siehe Abschnitt 4.1.3) oder filtern bzw. verschleiern Features oder Dateninstanzen, um den Datenschutz zu gewährleisten [Bor+03].

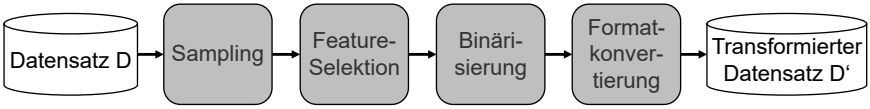


Abbildung 4.6: Datenvorbereitungs-Pipeline mit beispielhaften Datenvorbereitungsoperationen.

Beispiele entsprechender Software-Tools sind die in Abschnitt 4.1.1 genannten ETL-Tools [KRRT98; LN07]. Ebenso bieten Ansätze und Software-Tools zu sog. Data Mashups grafische Oberflächen, mit denen explorativ verschiedene Pipelines bzw. Datenflüsse von Datenquellen über Datenvorbereitungsoperationen bis hin zu Datensenken spezifiziert sowie direkt ausgeführt werden können [HB17; HBM18; HRWM15]. Auch Software-Tools wie KNIME[1], deren primärer Zweck eigentlich die Unterstützung von Datenanalysen ist, bieten Nutzern Möglichkeiten, um neben Analyseverfahren auch Datenvorbereitungsoperatonen in Pipelines oder Workflows zu spezifizieren. In einigen dieser Tools können zudem Ansätze zum Schema-Mapping [ABLM14; Kol05] und Schema-Matching [RB01] die Spezifikation von Operationen in Datenvorbereitungs-Pipelines unterstützen, indem sie Abhängigkeiten zwischen Features oder Dateninstanzen aus verschiedenen Datenquellen abbilden.

Auch wenn diese Software-Tools grafische Oberflächen bieten, ist in vielen industriellen Anwendungsfällen der Aufwand, um Datenvorbereitungs-Pipelines und die darin enthaltenen Datenvorbereitungsoperationen zu spezifizieren, immer noch sehr hoch [Grö18; Grö21]. Dies liegt primär daran, dass in diesen Anwendungsfällen eine Vielzahl an heterogenen Datenformaten integriert werden müssen (siehe Abschnitt 3.2.1) und dabei komplexe Datencharakteristika wie verschiedene Ursachen für Bias in Daten adressiert werden müssen (Abschnitt 3.2.2). Dies stellt insbesondere ein Problem für Domänenexperten dar, die nicht über das nötige Know-How bzgl. IT und Datenverarbeitung verfügen, um solche komplexen Datenvorbereitungs-Pipelines zu spezifizieren [BHM18; RSM14b; SBE+21]. Allerdings sollten Domänenexperten dennoch an der Datenvorbereitung beteiligt werden, da insbesondere nur diese Experten die resultierenden Ergebnisse hinsichtlich der domänenspezifischen Zielsetzung interpretieren können [BHM17; RS13; Via13].

[1] Konstanz Information Miner (KNIME): `https://www.knime.com/`

Darüber hinaus können Datenvorbereitungs-Pipelines und darin enthaltenen Datenvorbereitungsoperationen einige der Probleme verschlimmern, die verschiedene Ursachen für Bias in Daten mit sich bringen (siehe Wertung in Tabelle 4.2). Dies wird z. B. anhand des in Abbildung 3.6 dargestellten Datenintegrationsprozesses für den Anwendungsfall der Fehlerdiagnose im EoL-Prüffeld deutlich [Hir21; HRM19]. Dieser nutzt gängige Operationen, die von ETL-Tools oder anderen Software-Tools für die Datenvorbereitung angeboten werden, um Daten aus drei Quellsystemen in einen Trainingsdatensatz zu integrieren. Wie in Abschnitt 3.2.2 diskutiert wird, weisen die Datenmodelle und die Daten in den Quellsystemen verschiedene Qualitätsprobleme auf, wie bspw. unbrauchbare Klassenlabels in den Daten [HRM19; KM16] oder fehlende Schlüsselbeziehungen zwischen Datenquellen [Grö21; Grö22; WSR+20; WWIT16]. Diese führen häufig dazu, dass bei der Datenvorbereitung die aus den Quellsystemen extrahierte Datenmenge deutlich reduziert wird und somit für die Datenanalyse eine noch kleinere Datenmenge zur Verfügung steht. Weiterhin können sich bei der Datenvorbereitung die Verteilungen von Klassen und domänenspezifischen Gruppen in den Daten ändern. Dabei kann es passieren, dass gerade die relative Häufigkeit von seltenen Klassen oder seltenen Gruppen in den Daten noch geringer wird. Dies verschärft die Probleme mit der Klassenungleichverteilung oder Gruppenungleichverteilung.

4.3 Datenanalyse und Modellbildung

In diesem Abschnitt werden die zentralen Ergebnisse von Studien diskutiert, die datengetriebene Ansätze zur Datenanalyse und Modellbildung untersuchen. Zunächst werden mit Ensemble-Methoden [WGC14] sowie mit AutoML [ZH21] und Meta-Learning [Van19] Ansätze betrachtet, die einen positiven Effekt auf einzelne, aber ebenso einen negativen Effekt auf andere Ursachen für Bias in Daten haben. Anschließend steht die Untersuchung datengetriebener Ansätze zur Adressierung von Datensatzverschiebungen im Fokus [SRWM22].

4.3.1 Ensemble-Methoden für die Klassifikation

Ensemble-Methoden zielen speziell darauf ab, bei einer kleinen Menge an Trainingsdaten das Risiko einer Überanpassung der trainierten Klassifikations-

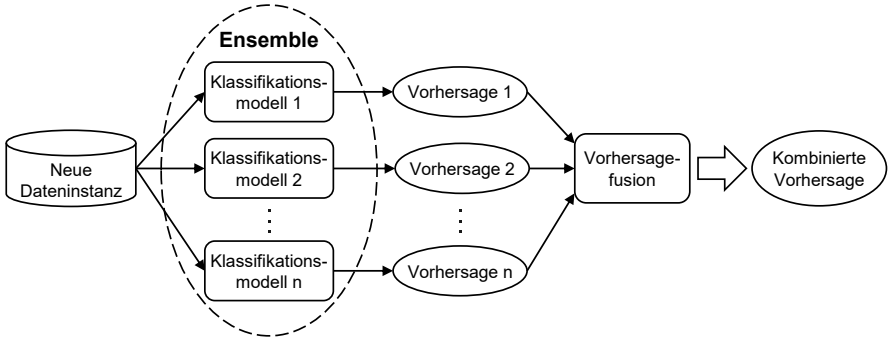


Abbildung 4.7: Anwendung eines Ensembles mit n Klassifikationsmodellen auf eine neue Dateninstanz. Die einzelnen Vorhersagen der Modelle werden durch eine Vorhersagefusion zu einer kombinierten Vorhersage zusammengeführt, vgl. [Hir21; WGC14].

modelle an die Daten zu reduzieren [Die00; Kun04; WGC14]. Dies geht häufig mit einer besseren Generalisierbarkeit und einer höheren Vorhersageleistung für neue Dateninstanzen einher. Ensemble-Methoden trainieren mehrere Klassifikationsmodelle, die divers bezüglich ihrer Vorhersagen sind. Dies bedeutet, dass die Modelle fehlerhafte und damit auch korrekte Vorhersagen jeweils für unterschiedliche Teilmengen der Daten treffen. Abbildung 4.7 verdeutlicht, wie ein aus n Klassifikationsmodellen bestehendes Ensemble auf eine neue Dateninstanz angewendet wird. Dabei trifft zunächst jedes der n Modelle eine Vorhersage. Wenn die Klassifikationsmodelle wie oben beschrieben divers sind, sind i. d. R. einzelne der Vorhersagen fehlerhaft, während andere wiederum korrekt sind. Anschließend kombinieren Algorithmen zur Vorhersagefusion – auch Entscheidungsfusion im Fall der Klassifikation – die einzelnen Vorhersagen zu einer finalen Vorhersage für die neue Dateninstanz. Die Idee ist, über eine Konsensentscheidung der n Klassifikationsmodelle möglichst nur die korrekten Vorhersagen als kombinierte Vorhersage zu übernehmen.

Verschiedene Ensemble-Methoden unterscheiden sich primär in den beiden Aspekten, wie eine Diversität der Klassifikationsmodelle erzeugt wird und wie die einzelnen Vorhersagen dieser Modelle zur kombinerten Vorhersage fusioniert werden. Gängige Methoden sind *Bootstrap Aggregation* (*Bagging*) [Bre96] und *Boosting* [Sch90]. Tabelle 4.3 fasst die Auswirkungen dieser Ensemble-

Tabelle 4.3: Auswirkungen verschiedener Ensemble-Methoden auf die Ursachen für Bias in Daten: + Positiver Effekt; 0 kein bzw. neutraler Effekt; − negativer Effekt.

Ansatz	**Klassen-ungleich-verteilung**	**Gruppen-hetero-genität**	**Gruppen-ungleich-verteilung**	**Falsche Labels**	**Kleine Daten-menge**
Bagging (z. B. Random Forest)	0	0	0	+	+
Boosting (z. B. AdaBoost)	+	0	+	−	+

Methoden auf die in Abschnitt 3.2.2 diskutierten Ursachen für Bias in Daten zusammen. Wie oben beschrieben adressieren beide Methoden das Problem einer Überanpassung der Klassifikationsmodelle an eine kleine Datenmenge. Im Folgenden werden die Auswirkungen auf die anderen Ursachen für Bias in Daten diskutiert und anhand zwei gängiger Ensemble-Methoden verdeutlicht: Random Forest [Bre01] für Bagging sowie AdaBoost für Boosting [FS96].

Random Forest [Bre01] trainiert n Entscheidungsbäume als Klassifikationsmodelle, die insofern divers sind, dass jeder Baum mit einer anderen zufälligen Teilmenge der Dateninstanzen trainiert wird. Zusätzlich wird bei jedem Knoten eines Entscheidungsbaums das für den Split des Knotens zu verwendende Feature aus einer zufälligen Teilmenge aller Features ausgewählt. Da jeder Entscheidungsbaum auf eine andere zufällige Teilmenge der Daten angepasst ist, ergänzen sich die Entscheidungsbäume gegenseitig hinsichtlich der Vorhersageleistung für unterschiedliche Dateninstanzen. Der Konsens der Entscheidungsbäume wird typischerweise über eine Mehrheitsentscheidung hergestellt. Dies bedeutet, dass Random Forest als finale Vorhersage die Klasse auswählt, die der Mehrheit der Vorhersagen der n Entscheidungsbäume entspricht.

Verschiedene Studien zeigen, dass Bagging-Verfahren wie Random Forest besser als andere Klassifikationsalgorithmen mit Daten umgehen können, die verrauschte Instanzen mit falschen Klassenlabels beinhalten [Bre01; Die00; Pol06]. Dies liegt insbesondere daran, dass mehrere der n Entscheidungsbäume mit echten Datenteilmengen trainiert werden, die weniger bis sogar keine der ursprünglich verrauschten Instanzen enthalten [HRM19]. Dadurch machen in den meisten Fällen die Mehrheit der Entscheidungsbäume bzgl. der

falsch dokumentierten Labels keine fehlerhaften Vorhersagen. Somit steigt die Wahrscheinlichkeit, dass die Konsensentscheidung aller Entscheidungsbäume ebenso weniger fehlerhaft ist. Auf die weiteren in Abschnitt 3.2.2 beschriebenen Ursachen für Bias in Daten haben Bagging-Verfahren nur geringfügige Auswirkungen [HRM19], sodass diese in Tabelle 4.3 neutral bewertet werden.

Das Boosting-Verfahren AdaBoost [FS96] trainiert iterativ mehrere diverse Entscheidungsbäume mit unterschiedlichen Variationen der ursprünglichen Trainingsdaten. In jeder Iteration verwendet AdaBoost eine Strichprobenziehung mit Ersetzung, um die jeweilige Menge an Trainingsdaten zu bestimmen. Dies bedeutet, dass in jeder Iteration einige Dateninstanzen ausgelassen werden, während andere Dateninstanzen repliziert werden, sodass sie mehrmals in der jeweiligen Datenmenge vorkommen. Die Wahrscheinlichkeiten, dass eine Dateninstanz in die Trainingsdaten übernommen wird oder dass sie sogar repliziert wird, hängen von einer Gewichtung der Dateninstanz ab. Bei der ersten Iteration sind alle Dateninstanzen gleich gewichtet. Danach erhöht AdaBoost in jeder Iteration die Gewichte derjenigen Instanzen, die in der vorherigen Iteration in einer Kreuzvalidierung vom trainierten Entscheidungsbaum falsch klassifiziert wurden. Die Gewichte der korrekt klassifizierten Instanzen werden hingegen reduziert. Nachdem AdaBoost n Iterationen durchgeführt hat, wird in der Vorhersagefusion die finale Vorhersage über einen Mittelwert aller Vorhersagen der n Entscheidungsbäume bestimmt.

Da AdaBoost in jeder Iteration unterschiedliche Variationen der Trainingsdaten verwendet, beugt es dem Problem einer Überanpassung des trainierten Klassifikationsmodells an eine einzelne Datenmenge vor [Die00; FS96; Pol06] (siehe Tabelle 4.3). Das dafür verwendete Verfahren der Strichprobenziehung mit Ersetzung hat zudem positive Effekte auf die Klassenungleichverteilung und auf die Gruppenungleichverteilung. In der ersten Iteration kommen Dateninstanzen von Minderheitsklassen und Minderheitsgruppen noch relativ selten in den verwendeten Trainingsdaten vor, sodass der Entscheidungsbaum für diese seltenen Instanzen tendenziell fehlerhafte Vorhersagen trifft. Dadurch werden diese seltenen Dateninstanzen aber in den folgenden Iterationen immer höher gewichtet. Folglich werden die Dateninstanzen von Minderheitsklassen und -gruppen in den folgenden Iterationen immer häufiger in den verwendeten Trainingsdaten repliziert, sodass die Entscheidungsbäume mit zunehmender

Anzahl an Iterationen auch mit einer höheren Wahrscheinlichkeit korrekte Vorhersagen für diese Minderheitsklassen und -gruppen treffen.

Allerdings führt genau dieses Verfahren der Stichprobenziehung mit Ersetzung zu Problemen, wenn die Daten verrauschte Instanzen mit falsch dokumentierten Klassenlabels enthalten. Viele dieser verrauschten Instanzen kommen in der ersten Iteration noch relativ selten in den Daten vor, sodass sie häufig falsch klassifiziert werden und AdaBoost anschließend deren Gewichte erhöht. Durch die Replikation der Dateninstanzen mit höheren Gewichten erhöht sich in den folgenden Iterationen von AdaBoost die Anzahl der verrauschten Instanzen in den Trainingsdaten. Daher tendieren Lernalgorithmen mehr und mehr dazu, die falsch dokumentierten Klassenlabels irrtümlich als korrekt anzusehen und für sie entsprechende Muster zu erzeugen und diese in den Entscheidungsbäumen abzubilden [Die00].

Die Diskussionsergebnisse werden von der durch Hirsch et al. [HRM19] durchgeführten Evaluation mehrere Klassifikationsalgorithmen für die Daten aus dem EoL-Prüffeld bestätigt. Sowohl Random Forest als auch AdaBoost weisen eine höhere Korrektklassifikationsrate auf als Klassifikationsverfahren, die keine Ensembles, sondern einzelne Klassifikationsmodelle verwenden. Dies liegt primär daran, dass Ensemble-Methoden besser mit dem kleinen Datensatz im Anwendungsfall umgehen können. Da aber keine der beiden Ensemble-Methoden alle Ursachen für Bias in Daten vollumfänglich adressieren, sind die Korrektklassifikationsraten mit 54 % für Random Forest sowie 50 % für AdaBoost immer noch relativ gering.

4.3.2 AutoML und Meta-Learning

AutoML-Systeme (*Automated Machine Learning*) [HZC21; XWL+21; ZH21] erzeugen automatisch optimierte ML-Modelle, indem sie das folgende Optimierungsproblem lösen [THHL13]:

$$\lambda^* \in \arg\min_{\lambda^j \in \Lambda} \mathcal{L}(\lambda^j, \mathcal{D}_{train}, \mathcal{D}_{valid}) \tag{4.1}$$

Dabei ist Λ ein Suchraum, der aus Konfigurationen λ^j von ML-Algorithmen und deren Hyperparametern besteht. AutoML-Systeme suchen in diesem Such-

Tabelle 4.4: Auswirkungen von AutoML-Systemen und Ansätzen zum Meta-Learning auf die Ursachen für Bias in Daten: + Positiver Effekt; 0 kein bzw. neutraler Effekt; − negativer Effekt.

Ansatz	**Klassen-ungleich-verteilung**	**Gruppen-hetero-genität**	**Gruppen-ungleich-verteilung**	**Falsche Labels**	**Kleine Daten-menge**
AutoML	−	0	−	0	−
Meta-Learning	+	−	−	0	+

raum diejenige Konfiguration λ^*, d. h. einen Lernalgorithmus mit einer spezifischen Hyper-Parametrisierung, die die Verlustfunktion $\mathcal{L}$ minimiert. Dabei werden für jede betrachtete Konfiguration $\lambda^j \in \Lambda$ mit Trainingsdaten $\mathcal{D}_{train}$ ein ML-Modell trainiert, für das anschließend mit Validierungsdaten $\mathcal{D}_{valid}$ die Verlustfunktion $\mathcal{L}$ berechnet wird. Diese Verlustfunktion kann beispielsweise die Fehlklassifikationsrate sein. Beispiele für AutoML-Systeme sind Autosklearn [FKE+19], AutoWEKA [THHL13], TPOT [OUA+16] und H2O [LP20].

AutoML-Systeme weisen Nachteile hinsichtlich einiger der in Abschnitt 3.2.2 beschriebenen Ursachen für Bias in Daten auf (siehe Tabelle 4.4). Die von AutoML-Systemen erzeugten ML-Modelle sind aufgrund der umfassenden Optimierung sehr stark an die Trainingsdaten $\mathcal{D}_{train}$ und an die Validierungsdaten $\mathcal{D}_{valid}$ angepasst. Wenn diese Datenmengen sehr klein sind, kann dies zu einer Überanpassung der ML-Modelle führen, die somit sehr schlecht generalisieren und eine geringe Vorhersageleistung für neue Dateninstanzen aufweisen. Außerdem minimieren AutoML-Systeme die Verlustfunktion $\mathcal{L}$ für den ganzen Datensatz. Dies führt dazu, dass sich die Optimierung meistens auf Mehrheitsklassen und Mehrheitsgruppen fokussiert, da diese einen größeren Anteil an den Dateninstanzen in $\mathcal{D}_{train}$ und $\mathcal{D}_{valid}$ einnehmen. Folglich werden Minderheitsklassen und Minderheitsgruppen auch von AutoML-Systemen tendenziell ignoriert, sodass die resultierenden ML-Modelle für entsprechend selten vorkommende Dateninstanzen häufig fehlerhafte Vorhersagen machen.

Ansätze zum *Meta-Learning* schlagen ML-Modelle oder Trainingskonfigurationen für ML-Modelle vor, die für eine bestimmte Lernaufgabe wie bspw. eine Klassifikation sowie für einen gegebenen Datensatz geeignet sind [PY10]. Sie nutzen dabei Metadaten, die bereits trainierte ML-Modelle beschreiben [Van19].

Diese Metadaten können z. B. aus Modell-Repositorys oder Modellverwaltungsplattformen stammen [VRBT13; VRWM23; WHR20; ZCD+18]. Die Metadaten beschreiben u. a. den für ein ML-Modell verwendeten Lernalgorithmus sowie dessen Hyperparameter, Ergebnisse verschiedener Evaluationsmetriken sowie bestimmte Charakteristika der verwendeten Trainingsdaten. Ansätze zum Meta-Learning identifizieren Muster innerhalb dieser Metadaten. Diese Muster können bspw. bestimmte Charakteristika von Trainingsdaten auf diejenigen Hyperparameter von Lernalgorithmen abbilden, mit denen für die jeweiligen Datencharakteristika gute Ergebnisse in den Evaluationsmetriken erzielt werden können. Für einen neuen Datensatz können mithilfe dieser Muster Kandidaten für Lernalgorithmen und Hyperparameter vorgeschlagen werden, die bei ähnlichen Datencharakteristika wie die des neuen Datensatzes eine gute Vorhersageleistung liefern [VWRM21].

Viele Ansätze zum Meta-Learning suchen für einen neuen Datensatz zunächst nach Datensätzen im Modell-Repository mit ähnlichen Datencharakteristika. Sie schätzen die Ähnlichkeiten der Datensätze bspw. über statistische oder informationstheoretische Metriken ab [RGS+22; TFSM23]. Dabei können auch Metriken hinsichtlich der Anzahlen und Verteilungen von Dateninstanzen, Klassen und Features genutzt werden [VWRM21]. Ist der neue Datensatz relativ klein oder weist er eine hohe Klassenungleichverteilung auf, so können über Ansätze zum Meta-Learning ML-Modelle gesucht werden, die mit Daten einer ähnlichen Datengröße oder einer ähnlichen Klassenverteilung trainiert wurden. Um Datensätze hinsichtlich ihrer Gruppenheterogenität und Gruppenungleichverteilung zu vergleichen, müssen die Metadaten Domänenwissen über die Eigenschaften dieser Gruppen berücksichtigen. Ansätze zum Meta-Learning betrachten solche domänenspezifischen Eigenschaften von Daten allerdings bisher kaum, sodass sie Nachteile bzgl. dieser beiden Ursachen für Bias haben.

4.3.3 Ansätze zur Adressierung von Datensatzverschiebungen

Eine weitere vom Autor dieser Habilitationsschrift betreutete Studie untersucht datengetriebene Ansätze zur Adressierung von Datensatzverschiebungen [SRWM22]. Dabei steht der in Abschnitt 3.1.4 beschriebene Anwendungsfall zur Qualitätssicherung während der Produktnutzung im Vordergrund. Der Trainingsdatensatz $\mathcal{D}_{train}$ umfasst 80 948 Dateninstanzen, von denen jede ein

Fahrzeug mittels 3000 Features von Produktstammdaten, wie bspw. Ausstattungsmerkmale des Fahrzeugs, beschreibt. Die Fahrzeuge in $\mathcal{D}_{train}$ wurden in den Jahren 2015 und 2016 produziert. Das binäre Klassenlabel beschreibt, ob ein Fahrzeug seit dessen Produktion innerhalb eines bestimmten Zeitraums, wie bspw. drei Jahren, bereits ausgefallen ist (Klasse 'positiv') oder nicht ('negativ'). Mit diesem Datensatz wird ein Klassifikationsmodell trainiert, das für zukünftig noch zu produzierende Fahrzeuge vorhersagt, ob diese während der Produktnuzung ausfallgefährdet sind. Als Testdaten $\mathcal{D}_{test}$ werden in der Studie Produktstammdaten zu Fahrzeugen verwendet, die im Jahr 2017 produziert wurden. Dies hat den Vorteil, dass zum Zeitpunkt der Veröffentlichung der Studie im Jahr 2022 [SRWM22] sicher gesagt werden kann, ob die Fahrzeuge in den Testdaten innerhalb des oben angesprochenen Zeitraums von drei Jahren ausgefallen sind oder nicht. Damit können die Vorhersagen des Klassifikationsmodells mit den tatsächlichen Klassenlabels in $\mathcal{D}_{test}$ abgeglichen werden. Als Evaluationsmetrik eignet sich die Wahr-Positiv-Rate, die den Anteil der in $\mathcal{D}_{test}$ enthaltenen tatsächlich ausgefallenen Fahrzeuge angibt, für die das Klassifikationsmodell die Klasse 'positiv' für einen Ausfall vorhergesagt hat.

Wie in Abschnitt 3.2.3 beschrieben weisen dieser Anwendungsfall und die Daten sowohl eine Konzeptverschiebung als auch eine Kovariatenverschiebung auf. Fahrzeuge fallen während ihrer Betriebszeit allmählich aus. Da sich somit auch die Klassenlabels der Dateninstanzen graduell von 'negativ' zu 'positiv' ändern, kann sich mit der Zeit eine Konzeptverschiebung zwischen diesen beiden Klassenlabels einstellen. Ebenso ändern sich über die Zeit auch die von Kunden bestellten Ausstattungsmerkmalen der Fahrzeuge. Dies führt letztlich zu einer Kovariatenverschiebung, da sich damit auch die Verteilungen der Feature-Werte zwischen Trainingsdaten $\mathcal{D}_{train}$ und Testdaten $\mathcal{D}_{test}$ unterscheiden. Beide Datensatzverschiebungen stehen zudem in einer Trade-off-Beziehung zueinander. Fahrzeuge, deren Produktion länger zurück liegt, weisen weniger gemeinsame Ausstattungsmerkmale mit den im Jahr 2017 produzierten Fahrzeugen in den Testdaten $\mathcal{D}_{test}$ auf. Folglich ergibt sich für diese älteren Fahrzeuge, wenn sie als Trainingsdaten verwendet werden, tendenziell eher eine Kovariatenverschiebung zu den Testdaten. Allerdings haben jüngere Fahrzeuge, die später produziert wurden und daher noch nicht so lange im Kundeneinsatz sind, eine höhere Wahrscheinlichkeit, dass sich ihr Klassenlabel noch von 'negativ' zu

’positiv’ ändert. Daher können jüngere Fahrzeuge auch mit ebenso höherer Wahrscheinlichkeit eine Konzeptverschiebung verursachen.

Andere Studien in der Literatur betrachten nur jeweils eine der beiden Arten von Datensatzverschiebungen [LLD+19; MRA+12; WHC+16; YNSC19]. Die Studie von Spieß et al. [SRWM22] untersucht hingegen, inwieweit datengetriebene Verfahren mit einem Datensatz umgehen können, der sowohl eine Konzeptverschiebung als auch eine Kovariatenverschiebung aufweist. Ebenso untersuchen die Autoren, welche Auswirkungen die Trade-off-Beziehung zwischen beiden Datensatzverschiebungen auf die Ergebnisse der datengetriebenen Verfahren hat. Dabei werden in der Studie von Spieß et al. [SRWM22] sowohl passive als auch aktive Verfahren zur Behandlung von Datensatzverschiebungen auf die Daten des Anwendungsfalls angepasst. Als passives Verfahren kommt ein inkrementeller Lernansatz [EP11; LHW18; Utg89] zum Einsatz, während das aktive Verfahren einem Windowing-Ansatz [AA19; BG07; LLD+19] entspricht. Für weitere Details zur Anpassung dieser Verfahren auf den Anwendungsfall sei auf die Publikation von Spieß et al. [SRWM22] verwiesen. Die angepassten Verfahren zum inkrementellen Lernen und Windowing werden in der Studie mit einer Baseline verglichen, die Datensatzverschiebungen nicht berücksichtigt. Angewendet auf die Testdaten $\mathcal{D}_{test}$ ergibt das Klassifikationsmodell dieser Baseline eine Wahr-Positiv-Rate von lediglich 55,9 % [SRWM22].

Das angepasste inkrementelle Verfahren erzielt eine Wahr-Positiv-Rate von 57,9 %, die um 2 %-Punkte besser ist als die der Baseline. Das inkrementelle Verfahren erreicht dies, indem es im Gegensatz zur Baseline die Konzeptverschiebung in den Daten adressiert und in monatlichen Zeitintervallen die jeweils neu ausgefallenen Fahrzeuge inkrementell mitberücksichtigt. Allerdings ist die Verbesserung der Wahr-Positiv-Rate mit lediglich 2 %-Punkten nicht so hoch, wie es die Literatur für andere Anwendungsfälle berichtet [EP11; LHW18; Utg89]. Dies liegt daran, dass in den Daten von Spieß et al. [SRWM22] die Kovariatenverschiebung stark zum Tragen kommt, da sich die Verteilung der Feature-Werte zwischen Trainingsdaten $\mathcal{D}_{train}$ und Testdaten $\mathcal{D}_{test}$ unterscheidet. Das inkrementelle Verfahren kann jedoch nur die Konzeptverschiebung adressieren. Die durch die Kovariatenverschiebung hervorgerufene Verschlechterung der Wahr-Positiv-Rate kann dieses Verfahren aber nicht ausgleichen.

Tabelle 4.5: Wahr-Positiv-Raten im Windowing-Verfahren für den Anwendungsfall von Spieß et al. [SRWM22].

Quartal für Teilmenge der Trainingsdaten $\mathcal{D}_{train}$	**Wahr-Positiv-Rate auf Testdaten $\mathcal{D}_{test}$**	**Veränderung bzgl. Baseline (55,9 %)**
Quartal 1, 2015	44,8 %	- 11,1 %-Punkte
Quartal 2, 2015	51,2 %	- 4,7 %-Punkte
Quartal 3, 2015	55,7 %	- 0,2 %-Punkte
Quartal 4, 2015	61,5 %	+5,6 %-Punkte
Quartal 1, 2016	69,3 %	+13,4 %-Punkte
Quartal 2, 2016	56,3 %	+0,4 %-Punkte

Beim auf den Anwendungsfall angepassten Windowing-Verfahren werden die Trainingsdaten $\mathcal{D}_{train}$ in mehrere disjunkte Teilmengen aufgeteilt [SRWM22]. Die Teilmengen enthalten jeweils die Fahrzeuge, die in einzelnen Quartalen der Jahre 2015 und 2016 hergestellt wurden. Für jede der Teilmengen von $\mathcal{D}_{train}$ wird ein Klassifikationsmodell trainiert und auf alle Testdaten $\mathcal{D}_{test}$ angewendet, um jeweils die Wahr-Positiv-Rate zu bestimmen. Tabelle 4.5 zeigt die Wahr-Positiv-Raten, die hierbei jeweils erzielt werden können.

Die Ergebnisse des Windowing-Verfahrens verdeutlichen den Trade-off zwischen der Konzeptverschiebung und der Kovariatenverschiebung. Die Fahrzeuge, die in den ersten Quartalen 2015 produziert wurden, weisen nur wenige gemeinsame Ausstattungsmerkmale mit den Fahrzeugen aus 2017 auf. Daher unterscheiden sich auch die Verteilungen der Feature-Werte zwischen den ersten quartalsspezifischen Teilmengen der Trainingsdaten von 2015 und den Testdaten von 2017, sodass die Kovariatenverschiebung stärker zum Tragen kommt. Dies führt dazu, dass die Wahr-Positiv-Rate mit den Teilmengen der Trainingsdaten aus den ersten drei Quartalen des Jahres 2015 sogar unter der Rate der Baseline liegt [SRWM22]. Erst im Quartal 4 des Jahres 2015 und im Quartal 1 des Jahres 2016 ist das Windowing-Verfahren um 5,6 %-Punkte bzw. um 13,4 %-Punkte besser als die Baseline. Wird eine Datenteilmenge ab Quartal 2 des Jahres 2016 verwendet, fällt die Wahr-Positiv-Rate aber wieder nahezu auf den Wert der Baseline ab. Bei den in diesen Teilmengen enthaltenen jüngeren Fahrzeugen spielt zwar die Kovariatenverschiebung eine geringere Rolle, allerdings stellt sich hier der negative Effekt der Konzeptverschiebung vollumfänglich ein. Jüngere Fahrzeuge haben grundsätzlich eine

höhere Wahrscheinlichkeit, dass sie noch ausfallen werden [SRWM22]. Daher ist ebenso die Wahrscheinlichkeit erhöht, dass mit Daten jüngerer Fahrzeuge eine Konzeptverschiebung verursacht wird.

Zusätzlich zu der Studie von Spieß et al. [SRWM22] werden in der Literatur spezifische Anpassungen von Ensemble-Methoden vorgeschlagen, die mit Datensatzverschiebungen umgehen können [BGBO19; GBR+17]. Diese Methoden überwachen Datenströme, um in ihnen eine Datensatzverschiebung erkennen zu können. Danach bestimmen sie die Entscheidungsbäume im Ensemble, die von der Datensatzverschiebung betroffen sind. Diese Entscheidungsbäume werden anschließend durch neue Bäume ersetzt, die mit neuen Daten im Datenstrom trainiert wurden und daher besser an die neue statistische Datenverteilung nach der Datensatzverschiebung angepasst sind. Allerdings können diese Ansätze immer nur eine Art von Datensatzverschiebungen adressieren, also entweder Konzeptverschiebungen oder Kovariatenverschiebungen. Somit bieten sie ebenso keine Lösung, die mit einer Trade-off-Beziehung zwischen beiden Arten von Datensatzverschiebungen umgehen können.

4.4 Zusammenfassung

Die Diskussionen in diesem Kapitel zeigen, dass rein datengetriebene Ansätze zur Datenakquise, Datenvorbereitung und Datenanalyse jeweils nur einzelne der in Abschnitt 3.2 beschriebenen Datencharakteristika industrieller Anwendungsfälle adressieren können. Tatsächlich haben die untersuchten datengetriebenen Ansätze negative Auswirkungen auf jeweils nicht adressierte Datencharakteristika, was in vielen Fällen zu einer Verschlechterung der Vorhersageleistung von Klassifikationsmodellen führt [HRM19; WSR+20].

In Schritt Datenakquise und Datenexploration des in Abbildung 2.4 dargestellten generischen Vorgehensmodells für Datenanalyseprozesse können zahlreiche Ansätze zur Datenintegration genutzt werden, um strukturierte oder semi-strukturierte Daten aus mehreren Datenquellen zu integrieren. Allerdings sind diese Ansätze nicht geeignet, um proprietäre Dateiformate zu integrieren [Rei17; ZRKM20; ZRKM21]. Hinsichtlich der in Abschnitt 3.2 beschriebenen Datencharakteristika bieten öffentlich verfügbare Datenrepositorys bzw. erzeugen existierende Datengeneratoren nur Datensätze, die eine spezifi-

sche Klassenungleichverteilung aufweisen und relativ klein sind. Allerdings bieten sie keine Datensätze, die viele heterogene und ungleichmäßig verteilte domänenspezifische Entitätsgruppen beschreiben.

Die meisten untersuchten Ansätze zur Datenvorbereitung können ausschließlich eine Klassenungleichverteilung adressieren. Allerdings verschlimmern entsprechende Ansätze die Probleme mit jeweils anderen Ursachen für Bias in Daten. In vielen Fällen führt dies zu einer deutlichen Verschlechterung der Vorhersageleistung von anschließend trainierten Klassifikationsmodellen [HRM19]. Einzig eine statistische Feature-Selektion kann zu einer Steigerung der Vorhersageleistung führen, indem diese Technik das Risiko einer Überanpassung der ML-Modelle an eine zu kleine Datenmenge reduziert. Allerdings kommen eine Feature-Selektion bei Datensätzen, die eine hohe Gruppenheterogenität aufweisen, an ihre Grenzen. In diesem Fall können die Datensätze sehr viele komplexe Muster enthalten, sodass schwer herauszufinden ist, welche Features als relevant oder irrelevant zur Abgrenzung dieser Muster einzustufen sind.

Im Schritt Datenanalyse und Modellbildung eignen sich Ensemble-Methoden als Lernverfahren, da sie das Risiko einer Überanpassung der ML-Modelle an eine kleine Datenmenge reduzieren [HRM19; WGC14]. Insbesondere stellen Bagging-Verfahren, wie bspw. Random Forest [Bre01], eine gute Wahl dar, die zudem gut mit falsch dokumentierten, verrauschten Klassenlabels umgehen können. AutoML-Systeme weisen Nachteile hinsichtlich einer Klassenungleichverteilung, einer Gruppenungleichverteilung sowie einer Überanpassung der optimierten ML-Modelle an kleine Datenmengen auf. Ansätze zum Meta-Learning berücksichtigen bei der Suche nach ähnlichen Datensätzen zwar die Verteilung von Dateninstanzen, Klassen und Features, allerdings keine Gruppenheterogenität oder Gruppenungleichverteilung.

Die Studie zu Datensatzverschiebungen von Spieß et al. [SRWM22] zeigt, dass sowohl passive als auch aktive Verfahren jeweils nur entweder eine Konzeptverschiebung oder eine Kovariatenverschiebungen adressieren können. Dies kann zu einer Verschlechterung der Vorhersageleistung von Klassifikationsmodellen führen, wenn der zugrundeliegende Datensatz beide Arten von Datensatzverschiebungen aufweist. Dies gilt insbesondere, wenn die Konzeptverschiebung und Kovariatenverschiebung in einer Trade-off-Beziehung stehen, was in vielen realen Anwendungsfällen der Fall ist [KU15; MGM20; NLDD18].

Kapitel 5

Nutzung von Domänenwissen zur Datenakquise und -exploration

Die in Abschnitt 4.1 diskutierten Ansätze zur Datenakquise und Datenexploration adressieren nur wenige der in Abschnitt 3.2 beschriebenen Datencharakteristika. Zum Beispiel sind Ansätze und Software-Tools zur Datenintegration nicht geeignet, um proprietäre Dateiformate zu integrieren [Rei17; ZRKM20; ZRKM21]. Ein weiteres Problem ergibt sich dadurch, dass die Datensätze, die von öffentlich verfügbaren Repositorys bereitgestellt werden bzw. mit existierenden Datengeneratoren erzeugt werden können, nicht alle Ursachen für Bias in Daten aufweisen, wie sie in realen Anwendungsfällen vorkommen, insbesondere keine Gruppenheterogenität und Gruppenungleichverteilung.

In diesem Kapitel werden Ansätze diskutiert, die im Schritt der Datenakquise und Datenexploration formal repräsentiertes Domänenwissen nutzen, um diese Probleme zu adressieren. Abbildung 5.1 kategorisiert die Ansätze hinsichtlich der jeweils durchzuführenden Aufgabe sowie hinsichtlich der jeweils genutzten Repräsentationsform des Domänenwissens. Abschnitt 5.1 diskutiert, wie Domänenwissen in Form semantischer Informationen in Metadatenmodellen verwendet werden kann, um heterogene Datenformate zu abstrahieren und semantisch zu beschreiben sowie damit den Aufwand für Domänenexperten zur Datenakquise zu verringern. Dieser Aspekt wird in Abschnitt 5.2 für ein

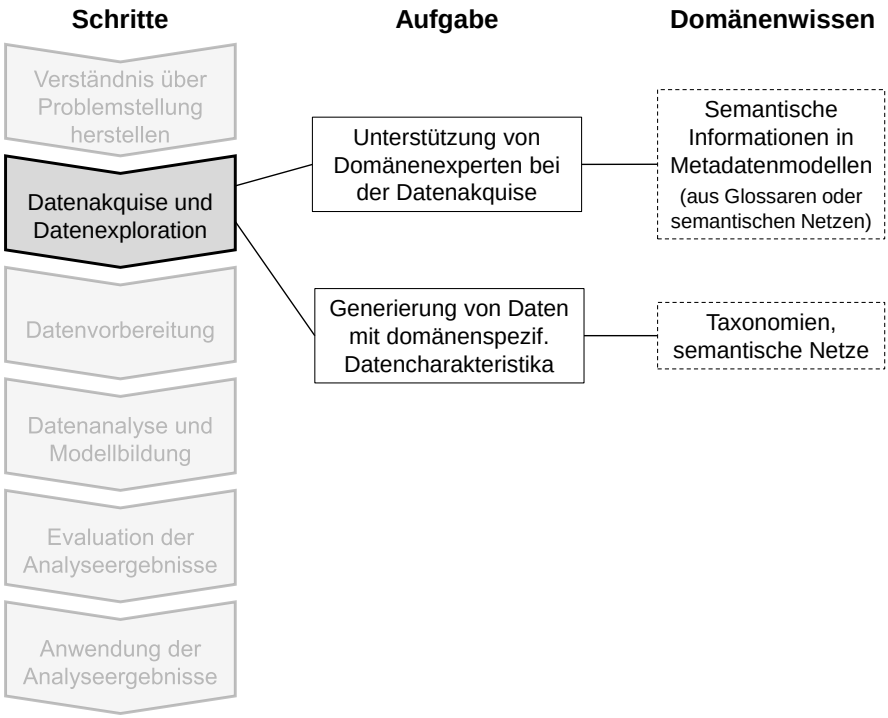


Abbildung 5.1: Durch Domänenwissen unterstützte Aufgaben in der Datenakquise und Datenexploration sowie Kategorisierung der Aufgaben und entsprechender Ansätze hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens.

konkretes Metadatenmodell vertieft, das im Rahmen eines vom Autor dieser Habilitationsschrift betreuten Projekts entstanden ist und speziell die Anwendungsdomäne der Produktentwicklung unterstützt [ZRKM21]. Sind in einem Anwendungsfall nicht ausreichend Daten vorhanden, können Datengeneratoren zusätzliche Daten erzeugen. Abschnitt 5.3 stellt einen neuen Ansatz für einen Datengenerator vor, der als Domänenwissen eine Taxonomie nutzt, um neue Daten mit spezifisch konfigurierbaren Ausprägungen einer Gruppenheterogenität und einer Gruppenungleichverteilung zu erzeugen [TRSM23].

5.1 Semantische Informationen in Metadatenmodellen zur Unterstützung von Domänenexperten bei der Datenakquise

Metadaten können insbesondere das Format, die Struktur, aber auch den Inhalt und die semantische Bedeutung von Daten innerhalb einer Anwendungsdomäne beschreiben [GH19]. Die Metadaten selbst werden dabei über Metadatenmodelle strukturiert. Metadatenmodelle und ihre Metadaten eignen sich, um heterogene Datenformate zu abstrahieren und so einen einheitlichen Zugriff auf die Daten zu ermöglichen [EGG+21a; ZRKM21]. Wird in den Metadaten der Inhalt und die semantische Bedeutung der Daten hinreichend gut beschrieben, erleichtert dies Domänenexperten, die für ihre Analyse passende Daten zu finden und zu akquirieren [GLN+05]. Zu diesem Zweck kann Domänenwissen in den Metadatenmodellen abgebildet werden, indem diese mit semantischen Informationen angereichert werden, die insbesondere aus Glossaren oder semantischen Netzen stammen [She99; WWR18; ZRKM21].

Nachfolgend wird in Abschnitt 5.1.1 zunächst beschrieben, auf welche Weise Metadaten allgemein den Schritt der Datenakquise und Datenexploration im generischen Vorgehensmodell für Datenanalyseprozesse unterstützen können. Abschnitt 5.1.2 diskutiert, wie Domänenwissen in Form semantischer Informationen in Metadatenmodelle abgebildet werden kann.

5.1.1 Metadaten zur Unterstützung der Datenakquise und Datenexploration

Gröger und Hoos [GH19] definieren mehrere Arten von Metadaten zur Beschreibung von Daten und der Datenverarbeitung (siehe Tabelle 5.1). *Technische Metadaten* beschreiben strukturelle und technische Aspekte der Daten und Datenhaltungssysteme, z. B. das Datenschema oder Zugriffsrechte bestimmter Nutzer oder Nutzergruppen. *Operative Metadaten* beschreiben Details, wie Daten verarbeitet wurden. Dies umfasst Spezifikationen der durchgeführten Datenzugriffe und Datentransformationen sowie Informationen über Datenqualität und Datenherkunft. Technische und operative Metadaten sind nach Gröger und Hoos [GH19] die zentrale Grundlage zur Sicherstellung von Governance-Richtlinien bei der Verarbeitung von Daten. Sie können genutzt werden, um die regel- und gesetzeskonforme Nutzung von Daten zu gewährleisten, z. B. hinsichtlich Anforderungen aus der Datenschutz-Grundverordnung (DSGVO).

Tabelle 5.1: Die wichtigsten Arten von Metadaten, was sie hauptsächlich beschreiben sowie welchen Einsatzzweck sie jeweils unterstützen; nach Gröger und Hoos [GH19].

Art der Metadaten	Beschreibung	primärer Einsatzzweck
Technische Metadaten	Strukturelle und technische Aspekte der Daten, z. B. Datenschemata, Zugriffsrechte	Daten-Governance
Operative Metadaten	Details zur Datenverarbeitung, z. B. zu Datentransformationen, Datenqualität, Datenherkunft	Daten-Governance
Fachliche Metadaten	Inhaltliche Bedeutung von Daten, Zusammenhänge in Daten	Datenakquise und Datenexploration

Fachliche Metadaten entsprechen einer semantischen bzw. konzeptionellen Beschreibung der inhaltlichen Bedeutung von Daten sowie von Zusammenhängen in Daten. Sie unterstützen somit speziell die Datenakquise und Datenexploration. Unter diese Metadaten fallen u. a. konzeptionelle Datenmodelle oder auch die oben angesprochene Abbildung von Domänenwissen und semantischer Informationen in Metadatenmodellen. Sie eignen sich insbesondere, um verschiedene heterogene Datenformate zu abstrahieren. Dies gilt auch für die in Tabelle 3.1 gelisteten strukturierten, semi-strukturierten und unstrukturierten Datenformate, die in industriellen Anwendungsfällen vorkommen. Die Metadaten können zuvor isolierte Datensilos verschiedener Organisationseinheiten aufbrechen, indem sie transparent beschreiben, welche Daten in welchen IT-Systemen eines Unternehmens vorliegen [EGG+21a].

Durch die konzeptionelle und semantische Abstraktion der Daten unterstützen fachliche Metadaten nicht nur Dateningenieure oder Datenwissenschaftlicher bei der Datenakquise, sondern auch Domänenexperten mit geringeren IT-Kenntnissen, wie bspw. Prozess- und Produktentwicklungsingenieure oder Betriebswirte [GH19; TSRC15; ZRKM20]. Ebenso unterstützen fachliche Metadaten Domänenexperten bei der Datenexploration, indem sie die zugrundeliegenden Daten bereits auf einer Meta-Ebene beschreiben, z. B. hinsichtlich der semantischen Bedeutung einzelner Attribute [GH19]. Somit muss bei der Umsetzung der Datenexploration diese Beschreibung auf der Meta-Ebene nicht mehr durch die Domänenexperten oder andere Fachexperten durchgeführt

werden. Zudem können bestimmte Zusammenhänge in den Daten bereits in Metadaten abgebildet werden, sodass Domänenexperten auf Basis dieser Zusammenhänge erste Hypothesen für ihre Datenanalysen erstellen können.

Die Literatur schlägt mehrere Metadatenmodelle vor, die geeignet sind, verschiedene Daten mit heterogenen Formaten zu abstrahieren. Das *MEtadata model for DAta Lakes* (MEDAL) [SSF+19] sowie Ground [HSG+17] repräsentieren jeweils Datenobjekte als Knoten in Graphen oder Hypergraphen, während Abhängigkeiten zwischen diesen Datenobjekten als Kanten oder Hyperkanten dargestellt werden. Das *Core Scientific Metadata Model* (CSMM) von Matthews et al. [MSF+10] nutzt Metadaten, um durch wissenschaftliche Studien erzeugte Daten zu beschreiben. Eichler et al. [EGG+21b] stellen mit HANDLE (*Handling metAdata maNagement in Data LakEs*) ein generisches Metadatenmodell vor, dem je nach Bedarf neue Metadatenstrukturen zur Beschreibung von Daten hinzugefügt werden können. Außerdem unterstützt HANDLE verschiedene Nutzergruppen, indem die Metadaten an die Bedürfnisse der Nutzergruppen angepasst bereitgestellt werden können. Mit Datenkatalogen gibt es spezielle Software-Lösungen, die eine Erfassung, Speicherung, Verarbeitung und Bereitstellung von Metadaten ermöglichen [EGH+22; JO23; LLEF20].

5.1.2 Domänenwissen und semantische Informationen in Metadatenmodellen

Um speziell Domänenexperten die Datenakquise und Datenexploration mittels Metadaten noch weiter zu vereinfachen, können Metadatenmodelle Domänenwissen in Form semantischer Informationen abbilden. Beschreiben diese semantischen Informationen Aspekte und Begriffe, die den Domänenexperten aus ihren jeweiligen Anwendungsfällen geläufig sind, können diese Experten über die Metadaten die für sie relevanten Daten mit deutlich geringerem Suchaufwand finden. Dabei können die Metadaten insbesondere mit Begriffen aus Glossaren oder kontrollierten Vokabularen verschlagwortet werden [EGG+21a; WWR18]. Ebenso ist es möglich, Begriffe oder weitere semantische Information aus semantischen Netzen wie Taxonomien oder Ontologien innerhalb der Metadatenmodelle abzubilden [WD10; WVV+01]. Die Abbildung der in einer Anwendungsdomäne oder einem Unternehmen geläufigen Begriffe in Metadatenmodelle ermöglicht eine kohärente, einheitliche und widerspruchsfreie Beschreibung der Metadaten [EGG+21a].

Velle et al. [VFP+05] stellen mit dem *Scientific Data Bag* (SDB) ein Speicherformat vor, mit dem semantische Informationen als Metadaten zur Beschreibung von Dateien heterogener Dateiformate verwaltet werden können. Dieser Ansatz ist allerdings auf Dateien beschränkt und unterstützt damit nicht alle der in Tabelle 3.1 gelisteten Datenformate industrieller Anwendungsfällen [ZRKM21]. Ebenso evaluieren die Autoren ihren Ansatz primär nur für die Anwendungsdomäne der numerischen Strömungsmechanik.

Andere Ansätze nutzen semantische Netze, die die Bedeutung von Daten und die semantischen Beziehungen zwischen Daten verschiedener Datenquellen beschreiben. Nargesian et al. [NPZ+20] schlagen einen entsprechenden Ansatz vor, mit dem die Daten eines Data Lake über eine Taxonomie hierarchisch organisiert und mit domänenspezifischen Begriffen aus der Taxonomie verschlagwortet werden können. Domänenexperten können durch die in der Taxonomie definierten hierarchischen Relationen zwischen Ober- und Unterbegriffen navigieren, um die für sie relevanten Daten zu finden. Weiterhin können semantisch reichhaltigere Informationen aus Ontologien oder Wissensgraphen verwendet werden, um Daten zu organisieren sowie zu verschlagworten und somit eine semantische Suche nach Daten zu ermöglichen [She99; WD10; XDCC19]. Schmid et al. [SHT19] zeigen beispielhaft, wie Ontologien und Wissensgraphen die Suche nach Daten in einem Data Lake aus dem Automobilbereich unterstützen können. Einige dieser Ansätze nutzen eine einzige Ontologie, die eine globale Sicht auf alle relevanten Datenquellen bietet [She99] (siehe Abbildung 5.2). Bei einer Vielzahl an heterogenen Datenformaten und unterschiedlichen Bedeutungen verschiedener Daten ist der Entwurf einer solchen einzigen globalen Ontologie allerdings schwierig [Rei17]. Hierfür eignen sich wiederum Ansätze, die mehrere Ontologien verwenden [WVV+01]. Dabei wird jede Datenquelle durch eine quellenspezifische Ontologie beschrieben, die dadurch deutlich weniger Datenformate abstrahieren muss. Beziehungen zwischen mehreren Datenquellen können über Relationen zwischen den Begriffen der jeweiligen Ontologien spezifiziert werden [WVV+01]).

Diese auf semantischen Netzen wie Taxonomien und Ontologien basierenden Ansätze unterstützen eine große Bandbreite der in Tabelle 3.1 gelisteten Datenformate. Manche eignen sich sogar zur Extraktion von Informationen aus unstrukturierten Textdaten oder aus Audio-, Bild- und Videodaten [ARR13;

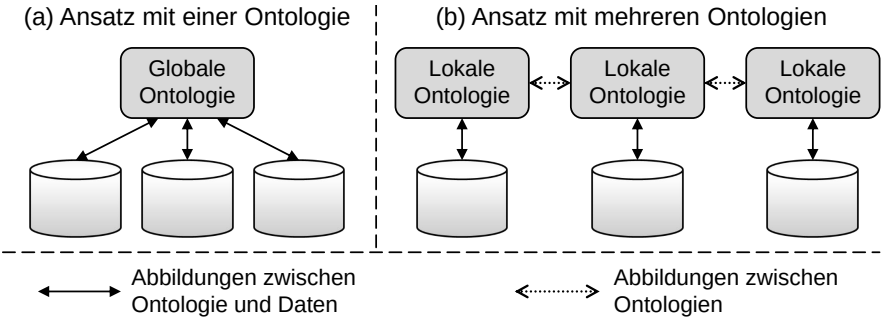


Abbildung 5.2: Grundlegende Ansätze zur Nutzung von Ontologien, um verschiedene Datenquellen zu abstrahieren. Die linke Seite zeigt, wie eine einzige Ontologie eine globale Sicht auf alle Datenquellen bietet, während auf der rechten Seite für jede Datenquelle eine separate Ontologie genutzt wird, vgl. Reimann [Rei17].

KFPI11; WD10]. Allerdings ist für solche Ansätze die Definition eines oder mehrerer semantischer Netze nötig, die alle in der Anwendungsdomäne relevanten Begriffe und deren Beziehungen spezifizieren. Da eine Definition semantischer Netze mit einem gewissen Aufwand verbunden ist, sollten für entsprechende Ansätze geeignete semantische Netze bereits vorliegen [Rei17]. Für industrielle Anwendungsfälle kommt dabei bspw. die Taxonomie ECLASS[1] zur Kategorisierung und semantischen Beschreibung von Produkten und Dienstleistungen infrage (siehe Abschnitt 2.2). Ebenso kann mittels der Automation Markup Language (AutomationML) [Dra10] eine Ontologie genutzt werden, die die Topologie und Struktur von Produktionsanlagen und Produkten abbildet.

Sämtliche in diesem Abschnitt beschriebenen Ansätze zur Abbildung von Domänenwissen und semantischer Informationen in Metadatenmodellen wurden allerdings noch nicht hinsichtlich ihres Potenzials untersucht, inwieweit sie die Datenakquise und Datenexploration speziell für Domänenexperten in den in Abschnitt 3.1 genannten industriellen Anwendungsfällen vereinfachen. Ebenso wurden auch die Taxonomie ECLASS und eine mit AutomationML beschriebene Ontologie, die aus dieser Anwendungsdomäne stammen, noch nicht hinsichtlich ihrer Eignung für Metadatenmodelle untersucht.

[1] ECLASS: `https://eclass.eu/`

5.2 Metadatenmodell zur Unterstützung der Datenakquise in der virtuellen Produktentwicklung

Produktentwicklungsprojekte werden mittels verschiedener CAx-Systeme, wie Computer-Aided Design (CAD), Computer-Aided Engineering (CAE), and Computer-Aided Testing (CAT) [DWGB04; VK11], zunehmend virtuell durchgeführt. Diese CAx-Systeme erzeugen und verarbeiten große Mengen an Daten, die prinzipiell viele Möglichkeiten für Datenanalysen bieten. Aber schon alleine die Datenakquise wird in der virtuellen Produktentwicklung für Domänenexperten wie bspw. Produktentwicklungsingenieure erschwert, da CAx-Systeme ihre Daten in verschiedenen heterogenen Formaten abspeichern [ZRKM21]. Neben strukturierten CSV-Dateien finden sich hier viele Formate für proprietäre 2D- oder 3D-Geometriedateien oder auch Text- oder Binärdateien (siehe Tabelle 3.1). Darüber hinaus sind die einzelnen CAD-, CAE- und CAT-Anwendungen meist unterschiedlichen Organisationseinheiten eines Unternehmens zugeordnet, die ihre Daten in isolierten Datensilos verwalten [DWGB04].

Im Rahmen eines vom Autor dieser Habilitationsschrift betreuten Projekts schlagen Ziegler et al. [ZRKM21] ein Metadatenmodell vor, mit dem Daten aus diversen CAx-Anwendungen beschrieben werden können. Dieses Metadatenmodell unterstützt sämtliche heterogenen Datenformate dieser Anwendungsdomäne. Es bietet eine einheitliche und zusammenhängende Sicht auf alle Daten der zuvor isolierten Datensilos in den Bereichen CAD, CAE und CAT, um somit die Datenakquise und folglich auch Datenanalysen über verschiedene CAx-Anwendungen hinweg zu ermöglichen. In diesem Abschnitt werden die zentralen Aspekte dieses Metadatenmodells diskutiert und dabei in den Gesamtkontext dieses Kapitels gesetzt. Für weitere Details sei auf die Publikation von Ziegler et al. [ZRKM21] verwiesen. Abschnitt 5.2.1 beschreibt die Bestandteile des Metadatenmodells sowie welche semantischen Informationen hierbei verwendet werden, um die Datenakquise für Domänenexperten zu unterstützen. Abschnitt 5.2.2 fasst die zentralen Ergebnisse einer Validierung zusammen, inwieweit das Metadatenmodell die Datenakquise für unterschiedliche Datenanalysen in der virtuellen Produkentwicklung tatsächlich vereinfacht. Schließlich diskutiert Abschnitt 5.2.3 die Übertragbarkeit des Metadatenmodells auf andere Anwendungsdomänen und deren Daten.

5.2.1 Bestandteile des Metadatenmodells

Abbildung 5.3 zeigt den Ausschnitt einer Instanz dieses Metadatenmodells für ein beispielhaftes virtuelles Produktentwicklungsprojekt [Rei23; ZRKM21]. Die Metadaten entsprechen einer graphbasierten Struktur mit mehreren Arten von Knoten. Die erste Art von Knoten entspricht *Datencontainern* (blau in der Abbildung). Ein Datencontainer ist eine in einem Datenhaltungssystem eindeutig identifizierbare Sammlung von Daten, z. B. eine Datei in einem Dateisystem oder eine Tabelle in einem Datenbanksystem [Rei17; RRS+11]. Datencontainer sind ein abstraktes Konzept, sodass sie beliebige Daten beschreiben können, unabhängig davon, welches Format diese Daten haben. Dies ermöglicht die einheitliche Beschreibung von Daten heterogener Formate im Metadatenmodell. Jeder Datencontainer wird zudem mit einem URI versehen, der auf den jeweiligen Speicherort im zugrundeliegenden Datenhaltungssystem verweist und mit dem die Daten von diesem Quellsystem akquiriert werden können.

Darüber hinaus beschreiben weitere Knoten des Metadatenmodells explizit die *Arbeitsaktivitäten* (gelb in Abbildung 5.3), die von Entwicklungsingenieuren mittels Software in virtuellen Produktentwicklungsprojekten durchgeführt werden [ZRKM21]. Weiterhin werden die Knoten der Arbeitsaktivitäten über gerichtete Kanten mit den Knoten der Datencontainer verknüpft, deren Daten die Arbeitsaktivitäten bzw. die jeweils genutzte Software als Eingaben benötigen oder die sie als Ausgaben produzieren. Damit wird in einer Instanz des Metadatenmodells der Datenfluss zwischen den Datencontainern und Arbeitsaktivitäten eines Produktentwicklungsprojekts abgebildet.

In dem in Abbildung 5.3 dargestellten Beispiel startet der Projektworkflow mit den Daten einer Produktspezifikation [ZRKM21] (Knoten 1). Diese Produktspezifikation dient als Grundlage zur Produktplanung, bei der Entwicklungsingenieure einen virtuellen Prototyp in Form eines 2D- oder 3D-Produktdesigns entwerfen (Knoten 2+3). In der nächsten Arbeitsaktivität wird der virtuelle Prototyp mit einer CAE-Software simuliert, um bestimmte Eigenschaften des Produktdesigns zu testen (Knoten 4). Die CAE-Software liefert Simulationsdaten (Knoten 5), auf deren Grundlage Ingenieure entscheiden, welches Produktdesign als physischer Prototyp gebaut werden soll (Knoten 6+7). Dieser physische Prototyp wird bspw. auf einem Prüfstand getestet (Knoten 8), wodurch finale Testergebnisse als Daten geliefert werden (Knoten 9).

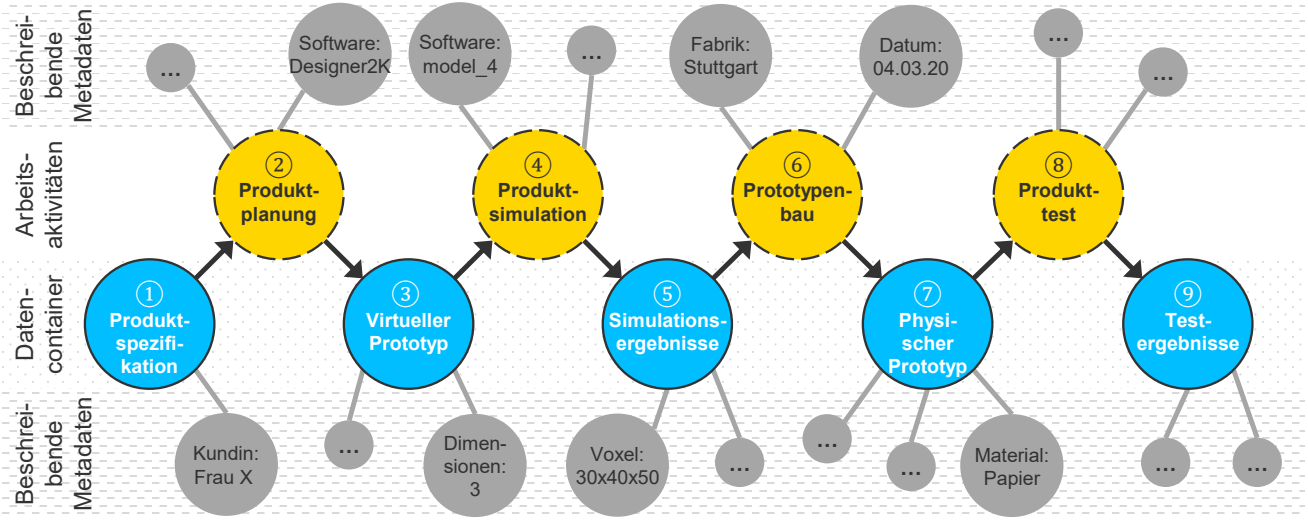


Abbildung 5.3: Beispiel einer Instanz des Metadatenmodels, das sowohl Daten, Metadaten als auch Arbeitsaktivitäten von virtuellen Produktentwicklungsprojekten beschreibt, vgl. Ziegler et al. [ZRKM21].

Sowohl Datencontainer als auch Arbeitsaktvitäten können im Graphmodell jeweils mit beliebig vielen Knoten für *beschreibende Metadaten* verknüpft werden (grau in Abbildung 5.3). Diese Metadatenknoten beinhalten Attribute und deren Werte, mit denen die Datencontainer oder Arbeitsaktivitäten näher beschrieben werden können. Zum Beispiel können solche Attribute in den Metadaten beschreiben, von welcher Kundin eine Produktspezifikation kommt, mit welcher CAD-Software die Produktplanung sowie mit welcher CAE-Software die Produktsimulation durchgeführt wurden oder in welcher Fabrik und mit welchem Material der physische Prototyp gebaut wurde.

Diese beschreibenden Metadaten stellen Wissen in Form semantischer Informationen aus der Anwendungsdomäne der virtuellen Produktentwicklung dar. Aber auch die graphbasierte Struktur selbst stellt in gewisser Weise Domänenwissen dar, da sie explizit die Arbeitsaktivitäten als Knoten sowie deren Datenfluss als Kanten darstellen, die die Produktentwicklungsingenieure in ihrem Arbeitsalltag durchführen und die ihnen daher geläufig sind. Daher ist das Metadatenmodell für diese Domänenexperten leicht verständlich. Die semantischen Informationen in den Metadatenknoten sowie die geläufige projektspezifische Struktur des graphbasierten Metadatenmodells erleichtern für diese Experten die Datenakquise und Datenexploration in erheblichem Maße. Sie können damit die für ihre gewünschten Analysen benötigten Daten einfacher finden. So können sie z. B. die Metadatenstruktur durchnavigieren, um Daten zu suchen, die Ein- oder Ausgabedaten bestimmter Arbeitsaktivitäten darstellen und dabei mit bestimmten Attributwerten in den beschreibenden Metadaten assoziiert sind. Diese vereinfachte Datanakquise ermöglicht auch eine von Experten geführte Datenanalyse, um so detaillierte Erkenntnisse aus den zugrundeliegenden Daten gewinnen zu können [Via13; ZRKM21].

Weiterhin erlaubt das Metadatenmodell, zusätzliches Domänenwissen aus externen Glossaren, kontrollierten Vokabularen oder semantischen Netzen wie Ontologien einzubinden [ZRKM21]. Jeder Knoten für Datencontainer, Arbeitsaktivitäten und Metadaten kann mit einem konkreten *Typ* assoziiert werden. Dieser Typ kann sich dabei auf die in den externen Glossaren, kontrollierten Vokabularen oder semantischen Netzen definierten Begriffe beziehen. Dies ist als Alternative zu beschreibenden Metadaten insbesondere dann sinnvoll, wenn es bereits externe Glossare, kontrollierte Vokabulare oder

semantische Netze gibt, die eine einheitliche und widerspruchsfreie Bedeutung relevanter domänenspezifischer Begriffe vorgeben. Damit wird diese einheitliche und widerspruchsfreie Begriffsbedeutung auch im Metadatenmodell übernommen. Ein gängiges Beispiel einer dazu passenden Ontologie ist die *Product Design Ontology* [CCF+09]. Diese Ontologie ist speziell zur Beschreibung von CAD- und CAE-Daten konzipiert und definiert dabei die Formate der in diesen CAx-Anwendungen typischerweise verwendeten Dateien sowie mit welchen Metadaten diese Dateien beschrieben werden sollten. Solch eine einheitliche und widerspruchsfreie Begriffsbedeutung und Begriffsverwendung im Metadatenmodell kann auch den Datenaustausch zwischen unterschiedlichen Organisationseinheiten und CAx-Anwendungen eines Unternehmens erleichtern [EGG+21a]. Damit kann sich die Zusammenarbeit verschiedener Organisationseinheiten in einem Produktentwicklungsprojekt verbessern, was sich positiv auf die Projektergebnisse auswirken kann [DWGB04].

Abbildung 5.3 stellt eine idealisierte lineare Durchführung eines Produktentwicklungsprojekts dar. Wie in Abschnitt 3.2.3 beschrieben, kann sich die tatsächliche Reihenfolge, in der diese Schritte in realen Projekten durchgeführt werden, je nach Projekt unterscheiden. Ebenso müssen einzelne Arbeitsaktivitäten häufig iteriert werden, wenn z. B. die erzielten Ergebnisse nicht den Erwartungen entsprechen [ZRKM21; ZRS+22]. Die realen Instanzen des von Ziegler et al. [ZRKM21] vorgestellten Metadatenmodells weisen mehrere Verzweigungen sowie Mehrfachausführungen derselben abstrakten Arbeitsaktivitäten auf. Durch den Ansatz der graphbasierten Struktur des Datenflusses im Metadatenmodell ist es allerdings möglich, beliebig strukturierte und verzweigte Projektabläufe und damit auch die in Abschnitt 3.2.3 beschriebenen unstrukturierten und nichtlinear durchgeführten Prozesse abzubilden [ZRS+22].

5.2.2 Zentrale Ergebnisse der Validierung des Metadatenmodells

Das Metadatenmodell wurde mit verschiedenen Anwendungsfällen für Datenanalysen validiert. Dabei stand im Fokus, inwieweit es die Datenakquise speziell für Domänenexperten wie Produktentwicklungsingenieure erleichtert. Abbildung 5.4 zeigt die Systemarchitektur, mit der das Metadatenmodell umgesetzt und validiert wurde [ZRKM21]. Die hierfür genutzten Daten, die von CAD-, CAE- und CAT-Anwendungen stammen, werden in ihrem Rohformat

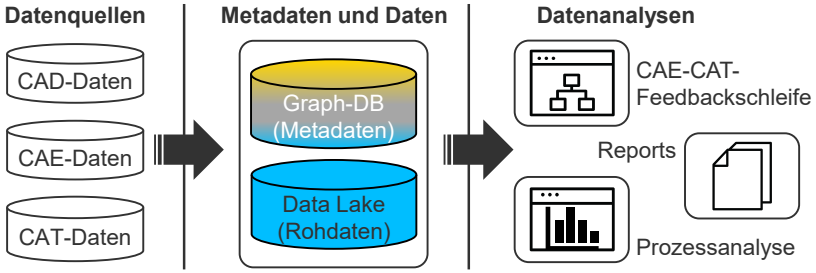


Abbildung 5.4: Systemarchitektur zur Umsetzung des graphbasierten Metadatenmodells sowie die Datenalysen, die für die Validierung genutzt wurden, vgl. Ziegler et al. [ZRKM21].

in einem Data Lake gespeichert. Die Instanzen des graphbasierten Metadatenmodells werden als Property-Graph-Struktur in einer Graph-Datenbank verwaltet. Wie oben beschrieben verweisen die einzelnen Metadatenknoten für Datencontainer mittels URIs auf die dazugehörigen Daten im Data Lake. Nutzer und Software-Anwendungen können auf die Metadaten und Daten zugreifen, indem sie durch die Graphstruktur der Metadaten durchnavigieren oder Anfragen in einer Anfragesprache wie bspw. Cypher [FGG+18] stellen.

Auf der rechten Seite zeigt Abbildung 5.4 die beispielhaften Anwendungsfälle für Datenanalysen, mit denen das Metadatenmodell validiert wurde [ZRKM20; ZRKM21; ZRS+22]. Das gesamte System und das Metadatenmodell wird aktiv von Ingenieuren des Projektpartners genutzt, um z. B. Daten für Reports zu akquirieren, die Ergebnisse von Produktentwicklungsprojekten zusammenfassen [ZRKM21]. Dabei können die Ingenieure bspw. mittels einer Analyse von Simulations- und Testergebnissen die von ihnen erstellten Produktdesigns verifizieren und Verbesserung der Designs ableiten [ZRK+24]. In einer weiteren Publikation diskutieren Ziegler et al. [ZRKM20], wie das Metadatenmodell Daten von abgeschlossenen Produktentwicklungsprojekten bereitstellen kann, um damit ein Empfehlungssystem für Produktdesigns zu ermöglichen.

Ein Anwendungsfall entspricht einer Feedbackschleife von physischen Produktests (CAT) zu computerbasierten Simulationen (CAE) [ZRKM21]. Dabei kann das Metadatenmodell zur Akquise einer spezifischen Teilmenge aller im Data Lake befindlichen CAT-Daten genutzt werden. Zum Beispiel kann eine Selektion von CAT-Daten durchgeführt werden, die zu Produkten gehören, die

aus einem spezifischen Material wie Zellulose bestehen. Die Analyse solcher spezifischen CAT-Daten kann Hinweise liefern, wie die vor den physischen Produktests durchgeführte computerbasierte Simulation bzw. bestimmte Simulationsparameter angepasst werden können, damit die Simulation genauere Ergebnisse liefert. Ist die Simulation genauer, können häufiger bereits nach der Simulation Erkenntnisse zur Verbesserung der Produktdesigns erzielt werden, sodass hierfür weniger physische Produkttests durchgeführt werden müssen.

In einer weiteren Publikation schlagen Ziegler et al. [ZRS+22] einen Ansatz vor, der die graphbasierten Metadaten für eine Prozessanalyse auswertet. Gängige Verfahren zur Prozessanalyse, z. B. zum Process Mining, sind nicht in der Lage, die Variantenvielfalt der unstrukturiert und nicht linear ausgeführten Prozesse von Produktentwicklungsprojekten zu erfassen [Aal11; DRG21; GA07] (siehe Abschnitt 3.2.3). Der Ansatz von Ziegler et al. [ZRS+22] nutzt nicht die normalerweise im Process Mining verwendete Datenstruktur eines *Event Logs*. Stattdessen nutzt er das graphbasierte Metadatenmodell, um Ausführungsdaten zu Produktentwicklugsprojekten aus einem PLM-System zu akquirieren und in eine Graphstruktur zu überführen, die den Kontroll- und Datenfluss der Prozesse abbilden kann. Anschließend können mit Verfahren zur Graphanalyse [JCZ13] Teilgraphen identifiziert werden, die Muster für häufig durchgeführte Teilprozesse der vielen Ausführungsvarianten von Produktentwicklungsprojekten repräsentieren. Dieser Ansatz kann deutlich besser mit unstrukturiert und nicht linear durchgeführten Prozessen umgehen, da nicht ein großes 'Spaghetti'-Prozessmodell [Aal11], sondern mehrere Muster für häufig vorkommende Teilprozesse abgeleitet werden.

5.2.3 Übertragbarkeit des Metadatenmodells auf andere Anwendungsdomänen

Das in diesem Abschnitt vorgestellte Metadatenmodell sowie die Datenakquise und Datenalysen, die es unterstützt, wurden von Ziegler et al. [ZRKM20; ZRKM21; ZRS+22] speziell für die Anwendungsdomäne der virtuellen Produktentwicklung implementiert und validiert. Allerdings ist das grundlegende Konzept auch auf Anwendungsfälle anderer Bereiche übertragbar. Das Metadatenmodell eignet sich insbesondere zur Beschreibung von Daten, wenn in diesen Anwendungsfällen Prozesse bzw. Projekte durchgeführt werden, in denen in

bestimmten Arbeitsaktivitäten mittels Software Daten als Eingaben eingelesen und als Ausgaben produziert werden. Dabei kann das Metadatenmodell nicht nur die Daten mit beschreibenden Informationen anreichern, sondern auch explizit die Arbeitsaktivitäten sowie den Datenfluss der Datenverarbeitung abbilden. Diese Charakterisierung der Prozesse bzw. Projekte trifft nicht nur auf viele industrielle Anwendungsfälle zu [Grö22; WWIT16], sondern auch auf andere Anwendungsdomänen wie bspw. medizinische Diagnosen [CRM+20; MGM20], da in deren Prozessen aufgrund der fortschreitenden Digitalisierung mehr und mehr Daten mittels Software generiert und ausgewertet werden. Dies bietet eine gute Ausgangsbasis zur Übertragung des Metadatenmodells sowie der Ergebnisse der Validierung von Ziegler et al. [ZRKM20; ZRKM21; ZRS+22] auf weitere Anwendungsdomänen.

5.3 Nutzung einer Taxonomie zur Generierung von Daten mit einer Gruppenheterogenität und Gruppenungleichverteilung

Öffentlich verfügbare Repositorys sowie existierende Datengeneratoren bieten keine Datensätze, die eine Gruppenheterogenität und Gruppenungleichverteilung in dem Maße aufweisen, wie sie in Daten realer industrieller Anwendungsfälle häufig existent sind (siehe Abschnitt 4.1). Dies liegt insbesondere daran, dass diese Repositorys und Datengeneratoren kein formal repräsentiertes Domänenwissen berücksichtigen, das aber zur realitätsnahen Erzeugung einer Gruppenheterogenität und Gruppenungleichverteilung benötigt wird. Insbesondere muss das Domänenwissen spezifische Eigenschaften und Zusammenhänge der domänenspezifischen Entitätsgruppen darstellen, wie bspw. zu verschiedenen Produktgruppen in industriellen Anwendungsfällen oder zu unterschiedlichen Personengruppen in medizinischen Anwendungsfällen.

Treder-Tschechlov et al. [TRSM23] schlagen einen Ansatz zur Datengenerierung vor, der diese Probleme adressiert. Dieser kann genutzt werden, um für einen Anwendungsfall realitätsnahe Daten zu erzeugen, sofern verfügbare Repositorys oder andere Datenquellen nicht ausreichend Daten anbieten. Der Ansatz zur Datengenierung nutzt zur realitätsnahen Erzeugung von Daten Domänenwissen in Form einer Taxonomie, die die jeweiligen domänenspezifischen Entitätsgruppen hierarchisch organisiert. Er kann Daten mit numerischen

und kategorischen Features sowie mit konfigurierbaren Anzahlen an Dateninstanzen, Klassen und Features und unterschiedlichen Klassenverteilungen erzeugen. Ebenso können mit dem Datengenerator aufgrund der Nutzung einer Taxonomie die jeweiligen Ausprägungen einer Gruppenheterogenität und Gruppenungleichverteilung in den erzeugten Daten eingestellt werden.

Nachfolgend werden in Abschnitt 5.3.1 zunächst die Vorgehensweise und die wesentlichen Schritte des Ansatzes zur Generierung synthetischer Daten erläutert. Anschließend fasst Abschnitt 5.3.2 die zentralen Evaluationsergebnisse von Treder-Tschechlov et al. [TRSM23] zusammen. Abschnitt 5.3.3 schließt mit einer Diskussion ab, inwieweit der Ansatz zur Datengenerierung auf andere Anwendungsdomänen und deren Datencharakteristika übertragbar ist.

5.3.1 Vorgehensweise des Ansatzes zur Datengenerierung

Abbildung 5.5 zeigt die wesentlichen Schritte des Ansatzes zur Generierung realitätsnaher Daten. Neben der Taxonomie werden dem Ansatz bestimmte Wahrscheinlichkeitsverteilungen zur Erzeugung einer Klassen- und Gruppenungleichverteilung als Eingabe übergeben, z. B. eine Zipf-Verteilung aus der Familie der Exponential-Verteilungen [Sch12]. Weitere Eingaben des Ansatzes zur Datengenerierung sind bestimmte Konfigurationsparameter, mit denen konkrete Eigenschaften und Verteilungen der zu erzeugenden Daten festgelegt werden können [TRSM23]. Zum Beispiel können die Anzahlen an Dateninstanzen, Features und Klassen festgelegt werden. Zwei weitere Parameter steuern die konkreten Verteilungen der Klassen bzw. Gruppen in den Daten, um somit die Ausprägungen einer Klassenungleichverteilung und Gruppenungleichverteilung zu spezifizieren. Mit weiteren Parametern kann eingestellt werden, inwieweit sich die Wertebereiche der Features für einzelne Klassen zwischen den domänenspezifischen Gruppen unterscheiden bzw. überlappen. Damit kann der Grad der Gruppenheterogenität in den Daten sowie die Komplexität der gruppenspezifischen Klassenmuster eingestellt werden.

Die Daten werden vom Generator in drei Schritten erzeugt. Im ersten Schritt traversiert der Datengenerator vom Wurzelknoten bis zu den Blattknoten durch die Taxonomie und legt für jeden Knoten und damit für jede domänenspezifische Gruppe auf jeder Ebene der Taxonomie fest, wie viele Dateninstanzen und welche Klassen für diese Gruppe erzeugt werden sollen. Durch diesen Schritt

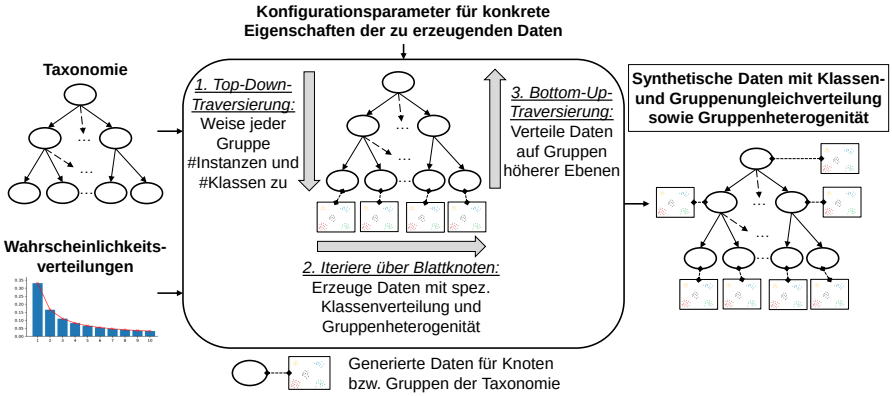


Abbildung 5.5: Vorgehensweise des Ansatzes zur Nutzung einer Taxonomie zur Generierung von Daten mit spezifischen Ausprägungen einer Gruppenheterogenität, Gruppenungleichverteilung und Klassenungleichverteilung, vgl. Treder-Tschechlov et al. [TRSM23].

wird primär die konkrete Ausprägung der Gruppenungleichverteilung in den Daten festgelegt. Dazu nutzt der Datengenerator die Wahrscheinlichkeitsverteilung und den entsprechenden Konfigurationsparameter für die Gruppenverteilung, um für jeden Knoten bzw. für jede Gruppe die konkrete Anzahl an Dateninstanzen und Anzahl an Klassen zu spezifizieren.

Im zweiten Schritt iteriert der Datengenerator über die Blattknoten der Taxonomie, um für jeden Blattknoten die spezifische Klassenverteilung zu erzeugen sowie die Daten zu generieren. Zunächst werden anhand der Wahrscheinlichkeitsverteilung und des Konfigurationsparameters für die Klassenverteilung pro Blattknoten die Anzahl an Dateninstanzen pro Klasse festgelegt. Dadurch wird die konkrete Ausprägung der Klassenungleichverteilung festgelegt. Weiterhin werden in jedem Blattknoten anhand des Ansatzes von Guyon [Guy03] die konkreten Feature-Werte jeder Dateninstanz erzeugt. Dabei wird der Konfigurationsparameter zur Einstellung des Grads der Gruppenheterogenität genutzt, um über die Blattknoten hinweg sicherzustellen, dass sich die Feature-Werte von Instanzen unterschiedlicher Gruppen in bestimmten charakteristischen Features unterscheiden. In der Folge unterscheiden sich auch die Muster derselben Klassen zwischen den domänenspezifischen Gruppen [TRSM23].

Im dritten Schritt werden die erzeugten Daten schließlich von den Blattknoten auf die weiteren Knoten der höheren Taxonomieebenen verteilt. Hierbei wird für jeden Elternknoten die Vereinigungsmenge aller Daten der Kindknoten übernommen. Schließlich liefert der Ansatz als Ergebnis die Taxonomie, bei der jedem Knoten die erzeugten Daten zugeordnet sind, die dann insgesamt die entsprechend konfigurierten Ausprägungen einer Klassenungleichverteilung, Gruppenheterogenität und Gruppenungleichverteilungen aufweisen.

5.3.2 Zentrale Evaluationsergebnisse

Treder-Tschechlov et al. [TRSM23] haben in ihrer Evaluation nachgewiesen, dass ihr Datengenerator in der Lage ist, Datensätze mit einem breiten Spektrum an Ausprägungen einer Klassenungleichverteilung, Gruppenheterogenität und Gruppenungleichverteilung zu erzeugen. Dazu haben sie mit unterschiedlichen Konfigurationsparametern mehrere Datensätze erzeugt und diese mit Metriken untersucht, die Aussagen über die Verteilungen von Klassen (Klassenungleichverteilung), Verteilungen von Klassenmustern (Gruppenungleichverteilung) sowie über die Komplexität und Separierbarkeit der Klassenmuster [HB02] (Gruppenheterogenität) in den Daten erlauben. Weiterhin haben sie Korrektklassifikationsraten von mit den Daten trainierten Klassifikationsmodellen evaluiert, um die Auswirkungen der verschiedenen Ursachen für Bias in Daten auf die Vorhersageleistung der Klassifikationsmodelle zu überprüfen. Nachfolgend werden die zentralen Ergebnisse dieser Evaluation hinsichtlich der drei relevanten Ursachen für Bias in Daten zusammengefasst:

- **Klassenungleichverteilung:** Hierfür wurde der Gini-Koeffizient [Cow11; Gin21] für die jeweiligen Klassenverteilungen der erzeugten Datensätze berechnet. Je nach Wert des zugehörigen Konfigurationsparameters des Datengenerators variieren die Gini-Koeffizienten zwischen 28 %, was einer relativ gleichmäßigen Klassenverteilung entspricht, und 73 %, was aussagt, dass die Klassen sehr ungleichmäßig verteilt sind.

  Weiterhin haben Treder-Tschechlov et al. [TRSM23] noch die Korrektklassifikationsraten von mit Random Forest trainierten Klassifikationsmodellen untersucht und dabei insbesondere überprüft, wie hoch die Vorhersageleistung für Mehrheitsklassen und Minderheitsklassen sind.

Je ungleichmäßiger die Klassen in den Datensätzen verteilt sind, desto geringer ist die Korrektklassifikationsrate speziell für Minderheitsklassen. Dies entspricht den Ergebnissen anderer Autoren, die Vorhersageleistungen von Klassifikationsmodellen bei ungleichmäßig verteilten Klassen untersuchen [GFB+12; HG09; HRM19; HRT+23; WP03; WY12].

- **Gruppenungleichverteilung:** Treder-Tschechlov et al. [TRSM23] haben analog bei jedem erzeugten Datensatz den Gini-Koeffizienten für die jeweilige Verteilung von Dateninstanzen auf domänenspezifische Gruppen der untersten Ebene der Taxonomie berechnet. Auch hierbei konnten sie je nach Einstellung des zugehörigen Konfigurationsparameter des Datengenerators unterschiedliche Gruppenverteilungen mit Gini-Koeffizienten zwischen ca. 20 % und 70 % erzeugen.

- **Gruppenheterogenität:** Hierfür haben die Autoren den mit dieser Datencharakteristika einhergehenden Aggregations-Bias untersucht, also inwieweit sich statistische Eigenschaften der Daten sowie die Ergebnisse einer Datenanalyse unterscheiden, wenn man entweder den ganzen Datensatz oder die Datenteilmengen für einzelne Gruppen auf der untersten Taxonomieebene betrachtet [TRSM23]. Dabei haben sie für den ganzen Datensatz und für die einzelnen Teilmengen einige der von Ho und Basu [HB02] vorgeschlagenen Komplexitätsmetriken für Klassifikationsprobleme berechnet, insbesondere Metriken auf Basis der Fisher'schen Diskriminanzfunktion sowie auf Basis des statistischen Anteils der Dateninstanzen, deren nächstgelegene Instanz im Feature-Raum einer anderen Klasse angehört (sog. *Border Points*). Diese Metriken sagen unabhängig vom Klassifikationsalgorithmus aus, wie schwierig es ist, in einem Datensatz die einzelnen Klassenmuster voneinander separieren zu können. Die von Treder-Tschechlov et al. [TRSM23] berechneten Werte dieser Komplexitätsmetriken sind deutlich höher, wenn die Metriken auf den ganzen Datensatz berechnet werden als nur auf den Teilmengen der einzelnen domänenspezifischen Gruppen. Demnach können die Klassenmuster im ganzen Datensatz deutlich schlechter voneinander separiert werden als innerhalb der einzelnen Teilmengen, was den wesentlichen Aspekt der Problematik einer Gruppenheterogenität ausmacht (siehe Abschnitt 3.2.2). Zudem haben Treder-Tschechlov et al. [TRSM23] gezeigt,

dass die Werte dieser Komplexitätsmetriken und damit die Separierbarkeit der Klassenmuster über die zugehörigen Konfigurationsparameter des Datengenerators gezielt eingestellt werden können.

Analog haben die Autoren auch die Korrektklassifikationsraten mit Random Forest für den ganzen Datensatz und für die einzelnen Teilmengen verglichen [TRSM23]. Bei Daten, die gemäß der Konfigurationsparameter des Datengenerators mit einer geringen Gruppenheterogenität erzeugt werden, ist der Unterschied zwischen den Korrektklassifikationsraten des ganzen Datensatzes und der Teilmengen nur marginal. Bei Daten mit einer höheren Gruppenheterogenität ist die Korrektklassifikationsrate für den ganzen Datensatz um mehr als 30 %-Punkte geringer als der Durchschnitt über alle gruppenspezifischen Teilmengen. Dies zeigt deutlich den mit der Gruppenheterogentiät einhergehenden Aggregations-Bias.

5.3.3 Übertragbarkeit des Ansatzes zur Datengenerierung auf andere Anwendungsdomänen und Datencharakteristika

Grundsätzlich ist der Datengenerator unabhängig von der konkreten Anwendungsdomäne nutzbar, um synthetische Daten mit einer Klassenungleichverteilung, Gruppenheterogenität und Gruppenungleichverteilung zu erzeugen. Die einzige Voraussetzung ist, dass in der jeweiligen Anwendungsdomäne eine Taxonomie oder ein anderes semantisches Netz mit hierarchischen Relationen zu Ober- und Unterbegriffen vorliegt, die die domänenspezifischen Entitätsgruppen hierarchisch organisieren. Da eine Taxonomie die einfachste Form eines semantischen Netzes ist, liegt sie in vielen Anwendungsdomänen bereits vor. Dies gilt insbesondere für industrielle Anwendungsfälle [AK04; HRT+23; SRYD14] sowie für medizinische Diagnosen [CRM+20; Fit88; Jab04].

Allerdings kann der Datengenerator bisher nur Daten mit numerischen oder kategorischen Features sowie mit kategorischen Klassenlabels erzeugen [TRSM23]. In Zukunft kann er folglich noch für andere Arten von Features, wie bspw. Textdaten, Zeitreihendaten oder Bilddaten, sowie für andere Arten von Labels, z. B. numerische Labels für eine Regression, erweitert werden.

Im Datengenerator können über Konfigurationsparameter beliebig die Anzahlen der zu erzeugenden Dateninstanzen, Klassen und Features festgelegt

werden. Damit können bzgl. weiterer Ursachen für Bias in Daten relativ einfach neben großen auch kleine Datenmengen mit einer geringen Anzahl an Dateninstanzen sowie relativ vielen Klassen und Features erzeugt werden, bei denen trainierte Klassifikationsmodelle tendenziell zur Überanpassung an die Daten neigen. Bisher ist es allerdings noch nicht möglich, falsch dokumentierte Klassenlabels im Datensatz zu erzeugen und die Anzahl und Verteilung dieser fehlerhaften Klassenlabels gezielt zu steuern. Um Datensätze mit dieser Ursache für Bias erzeugen zu können, muss der Datengenerator von Treder-Tschechlov et al. [TRSM23] entsprechend erweitert werden.

Ebenso muss noch untersucht werden, ob mit dem Datengenerator zwei oder mehr Datensätze erzeugt werden können, zwischen denen die in Abschnitt 3.2.3 beschriebenen Datensatzverschiebungen auftreten. Dabei ist insbesondere noch offen, ob der aktuelle Ansatz zur Datengenerierung ausreicht, um über eine gezielte Festlegung der Konfigurationsparameter unterschiedliche Ausprägungen der Konzept- und Kovariatenverschiebungen zu erzeugen.

5.4 Zusammenfassung

Im Schritt Datenakquise und Datenexploration des in Abbildung 2.4 dargestellten Vorgehensmodells für Datenanalyseprozesse wird formal repräsentiertes Domänenwissen insbesondere in Metadatenmodellen genutzt, um verschiedene Daten heterogener Formate zu abstrahieren und semantisch zu beschreiben. Dabei werden als Domänenwissen meist Begriffe aus Glossaren oder semantischen Netzen zur Verschlagwortung der Metadaten verwendet. Mit dieser Verschlagwortung der Metadaten wird speziell Domänenexperten mit geringen IT-Kenntnissen die Suche nach und Akquise von Daten aus unterschiedlichen Datenquellen vereinfacht [GLN+05; ZRKM21].

Im Rahmen eines vom Autor dieser Habilitationsschrift betreuten Projekts ist ein solches Metadatenmodell für die virtuelle Produktentwicklung entstanden [ZRKM20; ZRKM21]. Dieses Metadatenmodell beschreibt nicht nur Daten, sondern auch explizit die meist mit Software durchgeführten Arbeitsaktivitäten in Produktentwicklungsprojekten. Darüber hinaus bildet es den Datenfluss von Produktentwicklungsprojekten ab, indem es Arbeitsaktivitäten mit den Daten verknüpft, die von den Aktivitäten als Eingaben gelesen bzw.

als Ausgaben produziert werden. Da die Arbeitsaktivitäten sowie der durch diesen Datenfluss abgebildete Projektablauf die tägliche Arbeit der Produktenwicklungsingenieure widerspiegelt, sind diese Domänenexperten mit dieser Form des Metadatenmodells vertraut. Die prototypische Implementierung und Validierung des Metadatenmodells von Ziegler et al. [ZRKM21] zeigt, dass es die Domänenexperten in die Lage versetzt, die für verschiedene Datenanalysen passenden Daten zu finden und aus heterogenen und zuvor isolierten Quellsystemen zu akquirieren.

In einem weiteren Projekt ist der Ansatz zur Datengenerierung von Treder-Tschechlov et al. [TRSM23] entstanden. Dieser verwendet eine Taxonomie, um synthetische Daten mit möglichst realitätsnahen Datencharakteristika zu erzeugen. Er kann unterschiedlich große Datenmengen sowie konfigurierbare Ausprägungen einer Klassenungleichverteilung, Gruppenheterogenität und Gruppenungleichverteilung erzeugen. Damit schließt dieser Datengenerator eine Lücke im Stand der Forschung, da öffentlich verfügbare Repositorys bzw. andere Datengeneratoren keine Datensätze bieten, die eine Gruppenheterogenität und Gruppenungleichverteilung aufweisen.

Kapitel 6

Nutzung von Domänenwissen zur Datenvorbereitung

Auch im Schritt der Datenvorbereitung eines Datenanalyseprozesses adressieren die in Abschnitt 4.2 diskutierten datengetriebenen Techniken nur einzelne der Datencharakteristika industrieller Anwendungsfälle. Insbesondere gibt es viele, die hinsichtlich der Ursachen für Bias in Daten eine Klassenungleichverteilung adressieren. Allerdings berücksichtigt keine der datengetriebenen Techniken eine Gruppenheterogenität oder Gruppenungleichverteilung.

In diesem Kapitel werden daher Ansätze zur Nutzung formal repräsentierten Domänenwissens im Schritt der Datenvorbereitung diskutiert und aufgezeigt, inwieweit diese Ansätze die Lücke bei der Adressierung spezifischer Datencharakteristika schließen können. Abbildung 6.1 kategorisiert diese Ansätze jeweils hinsichtlich der für die Datenvorbereitung durchzuführenden Aufgabe sowie hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens. Ebenso wird nachfolgend diskutiert, welche der in Abschnitt 3.2 beschriebenen Datencharakteristika die Ansätze erfolgreich adressieren und auf welche der Datencharakteristika sie keine oder sogar negative Effekte haben. Dabei weisen die diskutierten Ansätze insbesondere positive oder negative Effekte auf verschiedene Ursachen für Bias in Daten auf. Tabelle 6.1 fasst die Ergebnisse dieser Diskussion zusammen.

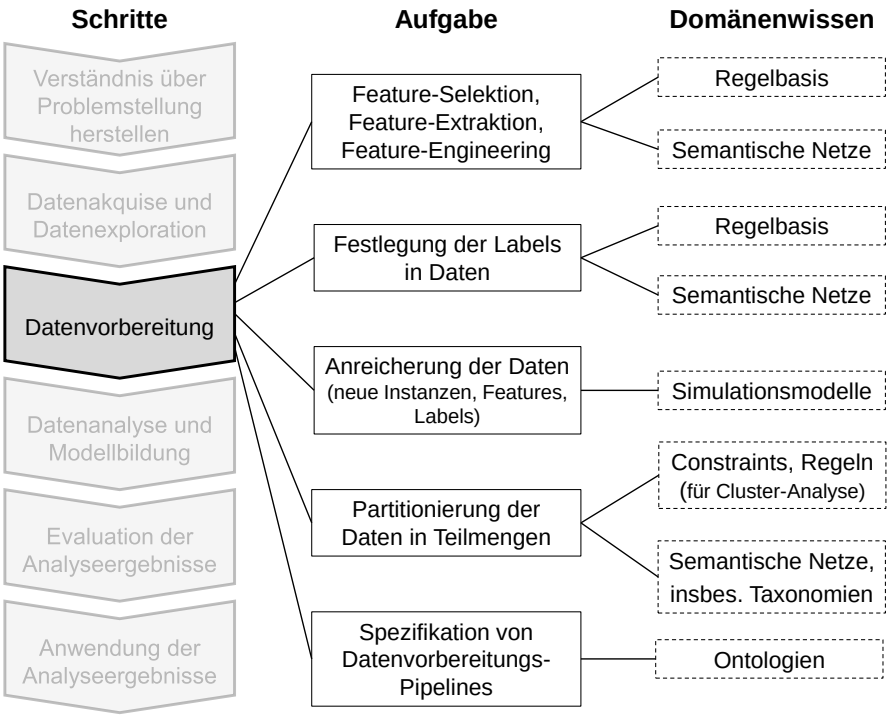


Abbildung 6.1: Durch Domänenwissen unterstützte Aufgaben in der Datenvorbereitung sowie Kategorisierung der Aufgaben und entsprechender Ansätze hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens.

6.1 Feature-Engineering mit Regeln oder semantischen Netzen

Eine Gruppe von Ansätzen in der Datenvorbereitung nutzt Domänenwissen, um eine Feature-Selektion, eine Feature-Konstruktion oder ein Feature-Engineering zu unterstützen. Von den in Abschnitt 2.2 beschriebenen Repräsentationsformen für Domänenwissen kommen in diesen Ansätzen primär eine Regelbasis [VLT10; ZRK+24] oder kausale Relationen aus semantischen Netzen zum Einsatz [BWK+22; KM16; Men20; MVS+19]. In Regeln oder kausalen Relationen können bekannte logische oder kausale Abhängigkeiten zwischen Features $f_1, \ldots, f_F$ im Feature-Raum $\mathcal{F}$ dargestellt werden. Diese können zur

Tabelle 6.1: Ansätze zur Nutzung von Domänenwissen in der Datenvorbereitung sowie Auswirkungen auf die Ursachen für Bias in Daten: + Positiver Effekt; 0 kein bzw. neutraler Effekt; – negativer Effekt.

Ansatz	**Klassen-ungleich-verteilung**	**Gruppen-hetero-genität**	**Gruppen-ungleich-verteilung**	**Falsche Labels**	**Kleine Daten-menge**
Feature-Engineering mit Regeln, semant. Netzen	0	–	–	0	+
Labeling mit Regeln, semant. Netzen	–	–	–	+	+
Anreicherung der Daten mit Simulationsergebnissen	–	0	–	+	+
Constraint-basierte Clusteranalyse	0	+	–	0	–
Datenpartitionierung mit Taxonomie	+	+	+	0	–
Ontologiebasierte Datenvorbereitungs-Pipelines	0	0	0	0	0

Selektion relevanter Features, zur Dimensionsreduktion in einem Datensatz oder auch zur Erhöhung des Informationsgehalts durch eine Extraktion oder Konstruktion aussagekräftigerer Features verwendet werden [KM16; ZRK+24]. Müller et al. [MVS+19] sowie Mende [Men20] beschreiben bspw. mit dem Merkmalentstehungsbaum (MEB) einen Graphen mit kausalen Relationen, mit denen Abhängigkeiten zwischen Merkmalen von Produkten und Produktionsprozessen dargestellt werden können. Solch ein MEB kann nach Blum et al. [BWK+22] zur Selektion relevanter Features aus diesen Merkmalen verwendet werden.

Die Nutzung semantischer Netze ist darüber hinaus ein gängiges Vorgehen bei der Auswertung von Textdaten, um semantisch reichhaltigere Features aus diesen Daten zu extrahieren [ITW05; KM16]. Ohne die Nutzung von Domänenwissen werden meistens die im Text häufig vorkommenden Wörter als Features für nachfolgende Datenanalysen verwendet (sog. *bag-of-words*). Alternativ können die Wörter über semantische Netze auf domänenspezifische Begriffe

abgebildet werden, sodass dann diese Begriffe als Features verwendet werden (sog. *bag-of-concept*) [ITW05]. Diese domänenspezifischen Begriffe sowie deren Häufigkeit in den Textdaten bieten in den meisten Anwendungsfällen aussagekräftigere Informationen. Kassner et al. [KM16] zeigen beispielhaft, wie ein semantisches Netz genutzt werden kann, um verschiedene Synonyme von Produktkomponenten und Fehlersymptomen aus den Textdaten zu extrahieren und auf einheitliche Komponenten-Fehler-Paare abzubilden. Diese einheitlichen Komponenten-Fehler-Paare dienen schließlich als aussagekräftigere Features für eine Klassifikation. Die Evaluationsergebnisse von Kassner et al. [KM16] zeigen, dass das verwendete semantische Netz auf den jeweiligen Anwendungsbereich zugeschnitten sein muss, damit die Nutzung dieses Domänenwissens zu einer Steigerung der Vorhersageleistung der mit den extrahierten Features trainierten Klassifikationsmodelle führt. Die in Textanalysen häufig verwendeten semantischen linguistischen Netze, wie bspw. WordNet[1] oder GermaNet[2], sind aber eher für Textdaten aus dem Bereich Web Analytics (siehe Abschnitt 2.1) und weniger für Industrial Analytics sowie für die in industriellen Anwendungsfällen vorherrschenden Datencharakteristika geeignet. Um dieses Problem anzugehen, bieten Kiefer et al. [KRM19b] einen Ansatz, der zu bestimmten Textdaten passende semantische Netze vorschlägt.

Sowohl eine mit diesen Ansätzen durchführbare Selektion relevanter Features als auch die Extraktion oder Konstruktion aussagekräftigerer Features kann bei einer kleinen Menge an zur Verfügung stehenden Daten das Risiko einer Überanpassung der ML-Modelle verringern. Bei näherer Betrachtung zeigt sich aber, dass die in industriellen Datensätzen häufig vorkommende Gruppenheterogenität zu einer Erhöhung der Anzahl und Komplexität der für solche Ansätze zu spezifizierenden Relationen zwischen Features führt [HRM19]. Dies liegt primär daran, dass sich analog zu den Klassifikationsmustern auch die Abhängigkeiten zwischen Features bei verschiedenen Gruppen deutlich unterscheiden können [HRT+23]. Für Domänenexperten ist es kaum möglich, alle in einer Anwendungsdomäne relevanten Abhängigkeiten zwischen Features voneinander zu differenzieren und vollständig innerhalb der Regeln bzw. innerhalb der semantischen Netzen zu spezifizieren. In vielen Fällen beherr-

[1] WordNet: `https://wordnet.princeton.edu/`
[2] GermaNet: `https://uni-tuebingen.de/en/142806`

schen Domänenexperten nur die Spezifikation der Feature-Abhängigkeiten, die für häufig vorkommende Gruppen gültig sind, weil sie sich mit diesen häufigen Gruppen noch auskennen. Demzufolge werden bei entsprechenden Ansätzen die Feature-Abhängigkeiten, die für selten vorkommende Gruppen gelten, meist ignoriert. Insgesamt weisen die Ansätze folglich Nachteile bzgl. einer Gruppenheterogeniät und einer Gruppenungleichverteilung auf.

Chen et al. [CLPC17] zeigen darüber hinaus, wie Ontologien genutzt werden können, um aus Datenströmen aussagekräftige Features zu extrahieren, die bei der Erkennung von Datensatzverschiebungen helfen können. Dabei müssen einzelne Dateninstanzen in den Datenströmen bereits mit Begriffen aus der Ontologie annotiert sein [LP13]. Weiterhin müssen in den Ontologien Relationen oder Axiome modelliert sein, die in der jeweiligen Anwendungsdomäne häufig auftretende Muster von Datensatzverschiebungen abbilden und semantisch beschreiben. Werden diese Relationen oder Axiome auf einen annotierten Datenstrom angewendet, können sog. semantische *Feature Embeddings* aus den Daten extrahiert werden, die die jeweils im Datenstrom auftretenden Muster von Datensatzverschiebungen repräsentieren [CLPC17]. Diese semantischen Embeddings können anschließend als Eingabefeatures beim Training eines ML-Modells verwendet werden, das in der Folge robuster gegenüber auftretende Datensatzverschiebungen ist. Allerdings behandelt dieser Ansatz nur Konzeptverschiebungen, aber keine Kovariatenverschiebungen, wie sie ebenso in vielen industriellen Anwendungsfällen vorkommen (siehe Abschnitt 3.2.3). Folglich adressiert der Ansatz auch nicht die typische Trade-off-Beziehung zwischen beiden Arten von Datensatzverschiebungen [SRWM22].

6.2 Festlegung der Labels in Daten mit Regeln oder semantischen Netzen

Eine Regelbasis oder kausale Relationen aus semantischen Netzen können ebenso Abhängigkeiten zwischen Features $f_1, \ldots, f_F \in \mathcal{F}$ und Labels $y_i \in \mathcal{Y}$ darstellen. Chimera [SRYD14] sowie Snorkel [BRL+19; RBE+20] sind entsprechende Ansätze, bei denen Domänenexperten ihr Wissen über die Abhängigkeiten zwischen Features und Labels nutzen können, um sogenannte Labeling-Regeln zu spezifizieren. Diese Labeling-Regeln werden anschließend genutzt, um bei

einzelnen Eingabedaten x_i des Datensatzes $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ das zuvor unbekannte Label y_i festzulegen [BRL+19; RBE+20] bzw. die Eingabedaten direkt zu klassifizieren [SRYD14]. Blum et al. [BWK+22] beschreiben wiederum Beispiele für Graphen mit kausalen Relationen aus der Domäne der Produktion, die u. a. die Festlegung der Labels in Daten unterstützen können. Die bekanntesten Beispiele solcher kausalen Graphen sind das Ursache-Wirkungs-Diagramm (UWD) von Ishikawa [Ish86] oder der *Cause & Effect Graph (CEG)* von Vogel-Heuser et al. [VKFK17]. Die in diesen Graphen modellierten kausalen Relationen können beispielsweise den Einfluss von Sensormesswerten aus Produktionsprozessen auf die Parameter der resultierenden Produktqualität bzw. auf mögliche Produkt- oder Produktionsfehler abbilden [HJX16; VKFK17]. In diesem Fall repräsentieren die Sensormesswerte die Features eines Datensatzes, während die Parameter der Produktqualität bzw. die Produkt- oder Produktionsfehler die Labels darstellen [BWK+22; HRM19]. Somit stellen die Graphen kausale Zusammhänge zwischen Features und Labels dar.

Speziell für Textdaten gibt es zudem Ansätze, die semantische Netze für eine sog. *Distant Supervision* nutzen [MBSJ09; YL19]. Dabei werden einzelne Wörter oder Sätze aus den Textdaten automatisch oder semi-automatisch mit Begriffen aus den semantischen Netzen annotiert. Bei geeigneter Wahl können diese annotierten Begriffe geeignete Labels für die Wörter oder Sätze darstellen.

Mit diesen Ansätzen müssen Domänenexperten nicht mehr für jede Dateninstanz aus $\mathcal{D}$ einzeln das Label y_i festlegen, sondern können mit der Spezifikation einer Labeling-Regel oder einer Relation in einem semantischen Netz mehrere Dateninstanzen abdecken. Dadurch reduziert sich der Aufwand zur Festlegung der Labels der Dateninstanzen, bzw. es kann bei gleichem Aufwand eine größere Datenmenge mit Labels versehen und als Trainingsdaten bereitgestellt werden. Dies reduziert das Risiko einer Überanpassung der trainierten ML-Modelle [RBE+20]. Bei geeigneter Wahl der Labeling-Regeln oder semantischen Netze können diese auch die Erkennung und Korrektur falsch dokumentierter Labels unterstützen [SRYD14; YL19].

Allerdings weisen diese Ansätze Nachteile hinsichtlich anderer Ursachen für Bias in Daten auf. Um alle möglichen Labels abzudecken, müssen die von Domänenexperten spezifizierten Labeling-Regeln oder Relationen in semantischen Netzen alle im Datensatz $\mathcal{D}$ vorkommenden Muster der einzelnen Labels

abbilden. Die Gruppenheterogenität führt aber dazu, dass pro Label mehrere, sich ggf. überlappende und schwer zu differenzierende Muster im Datensatz existieren [HRT+23; TRSM23]. Speziell bei Multi-Klassen- oder Multi-Label-Klassifikationen müssen Domänenexperten einen nicht vertretbaren Aufwand erbringen, um viele Labeling-Regeln oder Relationen in semantischen Netzen für die große Anzahl an Klassenlabels und die noch größere Anzahl an Klassenmustern zu spezifizieren [HRT+23; SRYD14]. Da die Klassenmuster verschiedener Klassen sich häufig noch im Datensatz überlappen [TRSM23], ist es für Domänenexperten zusätzlich noch schwer, diese Muster in diesen Wissensrepräsentationsformen korrekt zu differenzieren. Analog zur Diskussion in Abschnitt 6.1 kennen sich Domänenexperten primär mit den Zusammenhängen bzw. Mustern für häufig vorkommende Klassen oder häufige domänenspezifische Gruppen aus. Folglich spezifizieren sie vor allem korrekte Labeling-Regeln oder Relationen für die Muster dieser Mehrheitsklassen und Mehrheitsgruppen. Mit den Ansätzen werden also Minderheitsklasssen sowie Minderheitsgruppen vernachlässigt, sodass die Ansätze negative Effekte bzgl. einer Klassenungleichverteilung und einer Gruppenungleichverteilung aufweisen können.

6.3 Anreicherung der Daten mit Simulationsergebnissen

Die zur Verfügung stehenden Daten können mit weiteren simulierten Daten angereichert werden [GNH22; RMB+23; SFMP21], um spezifische Ursachen für Bias in Daten zu adressieren (siehe Tabelle 6.1). Dabei erzeugt eine Simulation Daten, die die Trainingsdaten $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ auf drei verschiedene Weisen ergänzen können (siehe Abbildung 6.2):

1. Simulationen erzeugen neue Dateninstanzen (x_i, y_i) und fügen diese dem Datensatz $\mathcal{D}$ hinzu. Analog können Simulationen auch für bereits existierende Eingabedaten x_i die noch unbekannten Labels y_i vorhersagen und die resultierenden Dateninstanzen (x_i, y_i) dem Datensatz $\mathcal{D}$ hinzufügen [PZL+18]. Beide Varianten erhöhen die Menge an mit Labels versehenen Trainingsdaten und reduzieren damit das Risiko einer Überanpassung der trainierten ML-Modelle an eine zu kleine Datenmenge.
2. Simulationen erzeugen keine neuen Dateninstanzen, sondern neue Features, die in den Trainingsdaten noch nicht vorhanden waren [DPW+19].

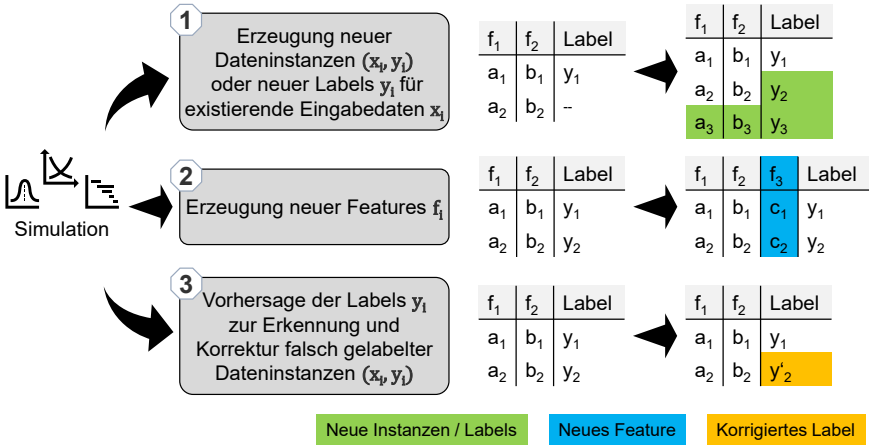


Abbildung 6.2: Drei gängige Vorgehensweisen zur Anreicherung eines Datensatzes mit Ergebnissen einer Simulation.

Werden geeignete neue Features erzeugt, erhöht dieses Vorgehen den Informationsgehalt des Feature-Raums $\mathcal{F}$. Dies kann erneut das Risiko einer Überanpassung der ML-Modelle an den Datensatz $\mathcal{D}$ reduzieren.

3. Simulationen sagen die Labels y_i von bereits gelabelten Dateninstanzen (x_i, y_i) vor. Damit können die im Datensatz $\mathcal{D}$ enthaltenen Labels überprüft werden [DKW+22]. Stimmt für eine Dateninstanz (x_i, y_i) das in $\mathcal{D}$ zuvor enthaltene Label y_i nicht mit dem durch die Simulation vorhergesagten Label überein, ist die entsprechende Dateninstanz möglicherweise mit einem falschen Label versehen. Dieses falsche Label kann dann mit dem von der Simulation vorhergesagten Label korrigiert werden.

Die erste oben gelistete Vorgehensweise kann prinzipiell auch genutzt werden, um wie in Abschnitt 5.3 beschrieben vollständig neue Daten zu generieren, wenn also zuvor noch keine Daten zur Verfügung standen. Da sie aber auch zur Anreicherung eines bereits existierenden Datensatzes und damit zu einem Teil des Schritts der Datenvorbereiterung verwendet werden kann und da alle weiteren oben gelisteten Vorgehensweisen ebenso zur Datenanreicherung genutzt werden, werden die Effekte entsprechender Ansätze auf relevante Datencharakteristika gemeinsam in diesem Abschnitt diskutiert.

Alle drei Vorgehensweisen setzen voraus, dass im jeweiligen Anwendungsfall ein Simulationsmodell vorliegt, das präzise und umfassend genug ist, um möglichst viele korrekte Dateninstanzen, Features und Labels erzeugen zu können. Allerdings ist der Aufwand zur Entwicklung solcher Simulationsmodelle in der Praxis sehr hoch, sodass die meisten verfügbaren Simulationsmodelle approximiert und idealisiert sind und einzelne im jeweiligen Anwendungsfall zentrale Aspekte, wie bspw. bestimmte physikalische Wechselwirkungen, nicht adäquat berücksichtigen. Dies führt dazu, dass Simulationen häufig nicht die in der Realität tatsächlich vorkommendenen Datenverteilungen erzeugen können. Konkret bezogen auf die in Abschnitt 3.2.2 beschriebenen Ursachen für Bias in Daten berücksichtigen Simulationen selten die Klassenverteilungen und Verteilungen domänenspezifischer Gruppen, wie sie in industriellen Anwendungsfällen vorkommen [HRM19; HRT+23; TRSM23]. Insbesondere erzeugen Simulationen häufig keine Daten für Minderheitsklassen oder Minderheitsgruppen. Ansätze zur Anreicherung von Daten mit Simulationsergebnissen adressieren folglich weder die Klassenungleichverteilung noch die Gruppenungleichverteilung in ausreichendem Maße (siehe Tabelle 6.1).

Diese Fehlanpassung, dass die Verteilungen der Simulationsdaten und realen Daten nicht übereinstimmen, wird als *Sim-to-Real-Gap* bezeichnet [SFMP21; TFR+17]. Es existieren einige Ansätze, um den Sim-to-Real-Gap zu schließen und simulierte Daten bzw. die mit ihnen trainierten ML-Modelle an die Verteilungen realer Daten anzupassen [GSSG12; HBH+21; SFMP21; TFR+17; ZQW20]. Diese Ansätze finden sich vor allem in Bereichen des Bestärkenden Lernens (engl. Reinforcement Learning) und der Computer Vision. Dabei werden sie primär für Bilddaten aus Anwendungsfällen der Robotik verwendet [RMB+23], sodass ihre Effekte für andere in industriellen Anwendungsfällen verwendeten Datenformate (siehe Tabelle 3.1) kaum erforscht sind.

Grundsätzlich lassen sich Ansätze zum Schließen des Sim-to-Real-Gap in drei Kategorien einteilen: Domänenrandomisierung, Domänenanpassung und Systemidentifikation [GSSG12; TFR+17; ZQW20]. Bei der Domänenrandomisierung werden die Parameter der Simulation zufällig variiert, sodass eine Variabilität in den erzeugten Simulationsdaten entsteht [TFR+17]. Aufgrund dieser soll das auf diesen Daten trainierte ML-Modell besser auf reale Daten generalisieren können. Durch die zufällige Wahl der Simulationsparameter

kann aber nicht garantiert werden, dass explizit die Datenverteilungen erzeugt werden, die im jeweiligen Anwendungsfall auftreten.

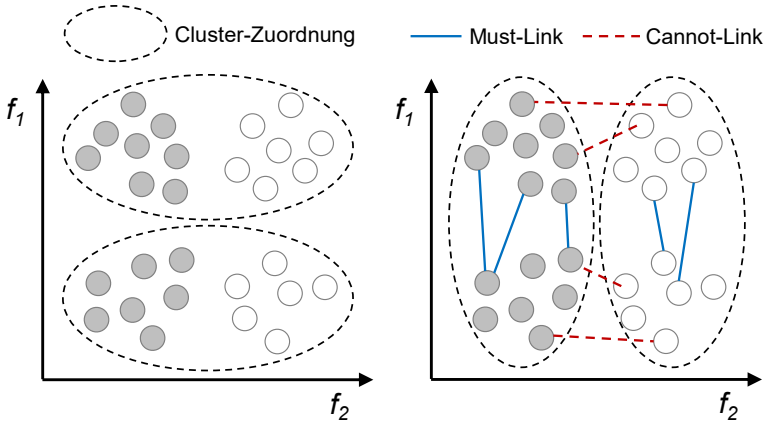
Bei Ansätzen der Domänenanpassung sollen die Datendomänen, also die jeweiligen Datenverteilungen der Simulationsdaten und realen Daten ineinander überführt werden. Dabei kann das mit den Simulationsdaten trainierte ML-Modell selbst an die realen Daten angepasst werden [ZQW20], oder es können beide Datenverteilungen in eine gemeinsame Datenverteilung überführt werden [GSSG12]. Eine Systemidentifikation bezeichnet die Einstellung der Parameter einer Simulation anhand von realen Daten [ZQW20]. Somit soll eine möglichst genaue, der Realität entsprechende numerische Modellierung einer Simulation erstellt werden. Damit eine Domänenanpassung oder eine Systemidentifikation zuverlässig funktionieren, müssen insbesondere ausreichend reale Daten zur Verfügung stehen. Dies ist in vielen Anwendungsfällen aber speziell bei Minderheitsklassen und Minderheitsgruppen nicht der Fall.

6.4 Datenpartitionierung mit Constraint-basierter Clusteranalyse

Bei einer Clusteranalyse werden Dateninstanzen eines Datensatzes so in Gruppen bzw. Datenteilmengen partitioniert, dass alle Instanzen einer Gruppe bzgl. der Werte bestimmter Features möglichst ähnlich zueinander sind [HKP12]. Die Datenteilmengen mit ähnlichen Instanzen werden als Cluster bezeichnet, die Zuordnung von Instanzen zu Clustern als Clustering. Eine Clusteranalyse kann sowohl als Technik zur Datenanalyse als auch zur Datenvorbereitung vor einer anderen Analyse verwendet werden [HKP12]. In diesem Abschnitt wird sie als Technik zur Datenvorbereitung betrachtet, um eine spezifische Ursache für Bias in Daten zu adressieren. Wird eine Clusteranalyse bspw. vor der Anwendung von Klassifikationsalgorithmen auf die Trainingsdaten angewendet, kann somit unter Umständen die Gruppenheterogenität adressiert werden. Die Idee ist, dass in jedem resultierenden Cluster nur Daten von einer oder von sehr wenigen domänenspezifischen Gruppen enthalten sind [Jai10; KRM19a]. Dies reduziert die Anzahl und Komplexität der in einem Cluster enthalten Klassenmuster. Anschließend können Klassifikationsalgorithmen einzeln auf die spezifischen Daten der Cluster angewendet werden und haben dabei weniger Probleme, die verschiedenen Klassenmuster zu erkennen [HRT+23].

Die Clusteranalyse muss dabei aber gezielt durchgeführt werden, sodass jedes resultierende Cluster nur noch Daten einer domänenspezifischen Gruppe oder zumindest Daten ähnlicher Gruppen enthält. Standardverfahren zur Clusteranalyse wenden statistische Metriken auf Features an, um die Ähnlichkeit von Dateninstanzen zu bestimmen [HKP12]. Diese statistischen Metriken bzw. die verwendeten Features berücksichtigen aber nicht zwingend die Zugehörigkeit der Dateninstanzen zu domänenspezifischen Gruppen. Hierbei ist die Einbeziehung von Domänenwissen über diese Gruppen in die Clusteranalyse und Datenpartitionierung hilfreich [HRT+23; KRM19a]. Verfahren zur Constraint-basierten Clusteranalyse [LJJ07; WCRS01] nutzen von Domänenexperten spezifizierte Constraints oder Regeln als Domänenwissen. Beispielsweise können Constraints zwischen zwei Dateninstanzen besagen, dass diese Instanzen zwingend demselben Cluster angehören müssen (sog. *Must-Link*) oder dass sie unterschiedlichen Clustern zugeordnet werden sollen (sog. *Cannot-Link*). Algorithmen der Constraint-basierten Clusteranalyse teilen die Dateninstanzen anhand ihrer statistischen Ähnlichkeit in Cluster auf und stellen dabei zusätzlich sicher, dass alle spezifizierten Constraints erfüllt werden.

Abbildung 6.3 zeigt den Unterschied zwischen dem Ergebnis einer Clusteranalyse, wenn diese ohne oder mit Constraints durchgeführt wird. Dabei sollen innerhalb der Features f_1 und f_2 ein Cluster für die grau eingefärbten Dateninstanzen und ein Cluster für die weißen Instanzen gebildet werden, die in einem konkreten Anwendungsfall bspw. zwei verschiedenen domänenspezifischen Gruppen entsprechen. Abbildung 6.3a zeigt das Ergebnis einer Clusteranalyse, bei der keine Constraints genutzt werden und die Dateninstanzen lediglich anhand ihrer statistischen Ähnlichkeit bzgl. der Features f_1 und f_2 in Cluster eingeteilt werden. Die somit gebildeten Cluster entsprechen aber nicht der dargestellten Färbung der Dateninstanzen und damit auch nicht den domänenspezifischen Gruppen. In Abbildung 6.3b ist das Clustering zu sehen, das durch eine Constraint-basierte Clusteranalyse erzielt werden kann. Hierbei haben Domänenexperten fünf Must-Links zwischen je zwei gleich gefärbten Dateninstanzen und vier Cannot-Links zwischen je zwei unterschiedlich gefärbten Instanzen spezifiziert. Durch die Ausnutzung dieser Constraints während der Clusteranalyse wird sichergestellt, dass zwei korrekte Cluster für grau bzw. weiß gefärbte Instanzen und deren Gruppen gebildet werden.



(a) Clustering ohne Constraints. (b) Clustering mit Constraints.

Abbildung 6.3: Schematische Darstellung einer Constraint-basierten Clusteranalyse. Die linke Teilabbildung zeigt das Ergebnis der Clusteranalyse ohne Nutzung von Constraints, während rechts das Ergebnis einer Constraint-basierten Clusteranalyse mit fünf Must-Links und vier Cannot-Links gezeigt wird.

Ansätze zur Constraint-basierten Clusteranalyse haben allerdings den gleichen Nachteil wie die Ansätze, die Regeln zum Feature-Engineering oder zur Festlegung der Labels in Daten nutzen [HRT+23]. Damit bei der Clusteranalyse ein Großteil der Dateninstanzen den richtigen domänenspezifischen Gruppen zugeordnet werden, müssen Domänenexperten eine Vielzahl an Constraints spezifizieren. Dabei kommt erschwerend hinzu, dass bei gängigen Verfahren zur Constraint-basierten Clusteranalyse die Constraints sogar paarweise für Dateninstanzen spezifiziert werden müssen [LJJ07; WCRS01]. Somit kann die Anzahl der für ein gutes Clustering-Ergebnis benötigten Constraints je nach Anwendungsfall sogar noch höher sein als die Anzahl an Regeln beim Feature-Engineering bzw. bei der Festlegung von Labels in Daten.

Insgesamt müssen Domänenexperten einen hohen Aufwand aufbringen, um eine Vielzahl an Constraints zu spezifizieren. Da dies eine komplexe Aufgabe ist, beschränken sie sich oft auf eine kleine Teilmenge der eigentlich benötigten Constraints. Dabei ist es wahrscheinlicher, dass sie Constraints für Dateninstanzen spezifizieren, die den Mehrheitsgruppen angehören, da sie sich mit diesen

Mehrheitsgruppen besser auskennen. Folglich werden bei der Anwendung von Constraint-basierten Clusteranalysen Minderheitsgruppen tendenziell vernachlässigt, sodass diese Ansätze negative Effekt auf die Gruppenungleichverteilung aufweisen. Konkret werden bei den Verfahren für die meisten Minderheitsgruppen keine dedizierten Cluster gebildet, sodass entsprechende Dateninstanzen fälschlicherweise den Clustern von Mehrheitsgruppen zugeordnet werden. Innerhalb dieser Cluster sind die spezifischen Klassenmuster der Minderheitsgruppen im Vergleich zu den Mustern der Mehrheitsgruppen unterrepräsentiert und werden anschließend häufig von Klassifikationsalgorithmen ignoriert.

Ein weiterer Nachteil von Clusteranalysen im Allgemeinen ist, dass die in industriellen Anwendungsfällen meist schon kleine Datenmenge pro Cluster noch kleiner wird. Bei diesen Verfahren muss also beachtet werden, dass das Risiko einer Überanpassung der ML-Modelle noch höher werden kann.

6.5 Verfahren zur Datenpartitionierung mittels einer Taxonomie

Die meisten der oben beschriebenen Verfahren adressieren keine Klassenungleichverteilung, Gruppenheterogenität und Gruppenungleichverteilung bzw. haben sogar negative Effekte auf diese Ursachen für Bias in Daten (siehe Tabelle 6.1). Im Rahmen eines vom Autor dieser Habilitationsschrift betreuten Projekts ist ein neuartiges Verfahren zur Datenvorbereitung entstanden, das diese wesentliche Lücke im Stand der Forschung schließt. Dieses Verfahren nutzt Domänenwissen in Form eines semantischen Netzes, d. h. in Form einer Taxonomie für eine Datenpartitionierung, um neben einer Klassenungleichverteilung auch effektiv eine Gruppenheterogenität und Gruppenungleichverteilung in Daten zu adressieren. Es ist in Autorenpublikationen beschrieben [HRM20a; HRT+23; Rei23], deren zentrale Inhalte in diesem Abschnitt erläutert und in den Gesamtkontext des Themas der Habilitationsschrift gesetzt werden.

In Abschnitt 6.5.1 werden die wesentlichen Schritte dieses Verfahrens zur Partitionierung von Daten anhand einer Taxonomie erläutert, die einer Produkthierarchie aus der Anwendungsdomäne der Produktion entspricht. Anschließend werden in Abschnitt 6.5.2 die zentralen Ergebnisse einer Evaluation des Verfahrens mittels der realen Daten eines Anwendungsfalls der Fehlerdiagnose im EoL-Prüffeld [HRM19; HRM20a] sowie mittels mehrerer synthetisch

erzeugter Daten mit unterschiedlichen Datenverteilungen [HRT+23; TRSM23] zusammengefasst. Schließlich diskutiert Abschnitt 6.5.3 die Übertragbarkeit des neuen Verfahrens auf andere Anwendungsdomänen und deren Daten.

6.5.1 Nutzung einer Taxonomie zur Datenpartitionierung

Abbildung 6.4 zeigt die wesentlichen Schritte des neuen Verfahrens. Die grundsätzliche Idee ist, einen Datensatz $\mathcal{D}$ so in einzelne Teilmengen $\mathcal{D}_j \subseteq \mathcal{D}$ aufzuteilen, dass die Gruppenheterogenität, Gruppenungleichverteilung und ebenso die Klassenungleichverteilung innerhalb der Teilmengen $\mathcal{D}_j$ weniger stark ausgeprägt sind wie im ganzen Datensatz $\mathcal{D}$. Anschließend können auf den Teilmengen $\mathcal{D}_j$ Klassifikationsmodelle trainiert werden, die dann nicht mehr so stark von den negativen Auswirkungen dieser drei Ursachen für Bias in Daten betroffen sind. Der erste Schritt *Segmentierung gemäß Produkthierarchie* (engl. *Segmentation according to Product Hierarchy (SPH)*) adressiert primär die Gruppenheterogenität und Gruppenungleichverteilung und nutzt eine Taxonomie, bspw. in Form einer Produkthierarchie, um den Datensatz $\mathcal{D}$ in mehrere gruppenspezifische Teilmengen $\mathcal{D}_j$ aufzuteilen. Der zweite Schritt *Klassenaufteilung gemäß Ungleichgewicht* (engl. *Class Partitioning according to Imbalance (CPI)*) adressiert die Klassenungleichverteilung, indem er einzelne der Teilmengen $\mathcal{D}_j$ weiter aufteilt, um Mehrheitsklassen von Minderheitsklassen zu separieren.

Segmentierung gemäß Produkthierarchie (SPH): Die grundlegende Motivation des Schritts SPH ist, spezifisches Domänenwissen zu nutzen, das direkt die domänenspezifische Ursache der Gruppenheterogenität und Gruppenungleichverteilung abbildet. In industriellen Anwendungsfällen liegt die Ursache für die Gruppenheterogenität und Gruppenungleichverteilung primär in der hohen Variantenvielfalt der Produkte eines Unternehmens [HZWK08; WSR+20]. Unternehmen pflegen i. d. R. eine systematische Beschreibung von Produktfamilien, die die vielen Produktvarianten des Unternehmens in Produktgruppen strukturiert [AK04]. Diese Beschreibung von Produktfamilien liegt häufig in Form einer Produkthierarchie vor, wie z. B. die in Abbildung 3.3 dargestellte Produkthierarchie, die verschiedene Varianten von Verbrennungsmotoren in hierarchisch strukturierte Produktgruppen organisiert. Eine Produkthierarchie kann in Form einer Taxonomie als semantisches Netz abgebildet werden. Das in

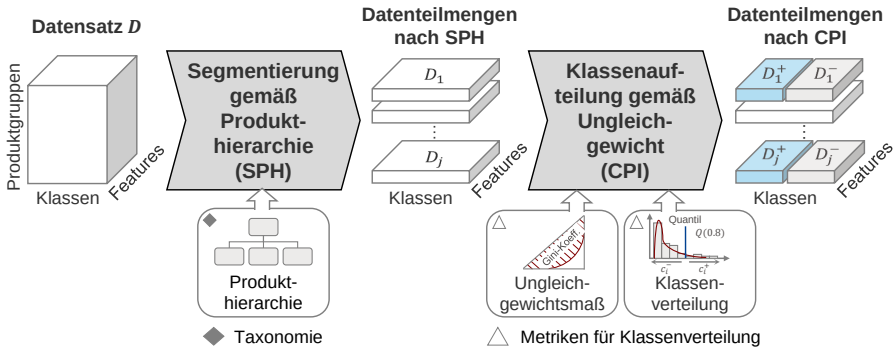


Abbildung 6.4: Wesentliche Schritte des Ansatzes, der eine Taxonomie und Metriken über Klassenverteilungen nutzt, um eine Gruppenheterogenität, Gruppenungleichverteilung und Klassenungleichverteilung zu adressieren, vgl. Hirsch et al. [HRM20a]

diesem Abschnitt vorgestellte Verfahren nutzt das in einer solchen Taxonomie repräsentierte Domänenwissen, um die zugrundeliegenden Daten systematisch aufzubereiten [HRT+23]. Grundsätzlich ist das Verfahren aber nicht auf Produkthierarchien beschränkt, sondern generell auf Taxonomien anwendbar, deren Knoten den domänenspezifischen Entitätsgruppen entsprechen, die somit in der Taxonomie hierarchisch organisiert werden (siehe Abschnitt 6.5.3).

Der Schritt SPH bildet für jeden Knoten j auf einer spezifischen Ebene der Taxonomie eine Datenteilmenge $\mathcal{D}_j \subseteq \mathcal{D}$. Werden bspw. mit der in Abbildung 3.3 auf Seite 61 dargestellten Produkthierarchie aus dem Anwendungsfall der Fehlerdiagnose im EoL-Prüffeld Datenteilmengen auf der dritten Hierarchieebene „Modell“ gebildet, so gibt es nach der Anwendung von SPH für jede Produktgruppe $DM3610$ bis $OM5691$ usw. jeweils eine Datenteilmenge $\mathcal{D}_j$. Da die Knoten der verwendeteten Taxonomie den jeweiligen domänenspezifischen Gruppen entsprechen, enthält jede der resultierenden Datenteilmengen $\mathcal{D}_j$ nur noch Daten einer einzelnen domänenspezifischen Gruppe. SPH reduziert somit in jeder Teilmenge $\mathcal{D}_j$ die Anzahl und Komplexität der Klassenmuster [HRT+23] und adressiert auf diese Weise die Gruppenheterogenität sowie den damit einhergehenden Aggregations-Bias in Daten (siehe Tabelle 6.1). Werden anschließend Klassifikationsalgorithmen einzeln auf die Teilmengen $\mathcal{D}_j$ angewendet, können diese Algorithmen die Klassenmuster besser erkennen

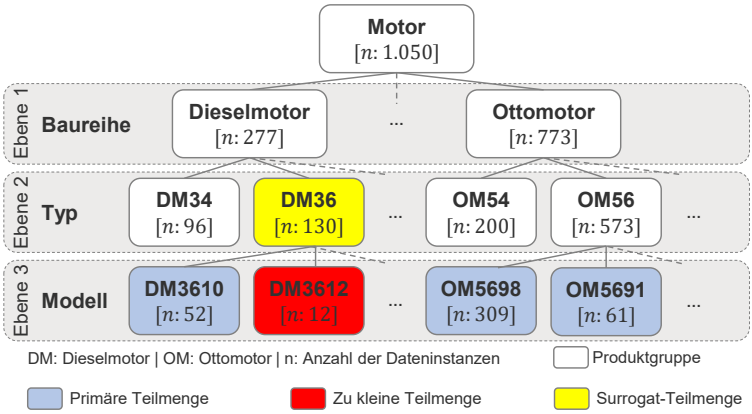


Abbildung 6.5: Produkthierarchie aus Abbildung 3.3 mit beispielhafter Darstellung, für welche Produktgruppen SPH typischerweise primäre Datenteilmengen oder Surrogat-Teilmengen bildet.

und voneinander abgrenzen, als wenn die Algorithmen auf den ganzen Datensatz $\mathcal{D}$ angewendet werden. Ebenso kann es keine Gruppenungleichverteilung und damit auch keine Vernachlässigung von Minderheitsgruppen mehr geben, da jede Teilmenge $\mathcal{D}_j$ nur noch Daten einer einzelnen Gruppe enthält.

Die Datenteilmengen $\mathcal{D}_j$ können prinzipiell für Gruppen auf unterschiedlichen Ebenen der verwendeten Taxonomie gebildet werden. Je tiefer die Ebene der Taxonomie gewählt wird, desto spezifischer sind die resultierenden Teilmengen $\mathcal{D}_j$ hinsichtlich der Klassifikationsmuster und desto stärker werden die Gruppenheterogenität und Gruppenungleichverteilung adressiert. Ist die gewählte Ebene aber zu tief, dann können in einzelnen Teilmengen $\mathcal{D}_j$ zu wenig Dateninstanzen enthalten sein, sodass das Risiko einer Überanpassung der Klassifikationsmodelle für diese kleinen Teilmengen sehr groß sein kann. Um solche Fälle zu berücksichtigen, führt SPH *primäre Datenteilmengen* und *Surrogat-Datenteilmengen* ein [HRT+23].

Abbildung 6.5 zeigt beispielhaft, für welche Produktgruppen aus der Produkthierarchie SPH typischerweise primäre Datenteilmengen und Surrogat-Teilmengen bildet. Primäre Datenteilmengen enthalten Daten von domänenspezifischen Gruppen, die sich auf einer tieferen Ebene der Taxonomie befinden, auf der die Gruppenheterogenität am meisten adressiert wird. In der Produkt-

hierarchie ist dies die dritte Ebene „Modell“. Die meisten der auf dieser Ebene befindlichen Gruppen werden durch eine ausreichende Anzahl an Dateninstanzen repräsentiert, sodass SPH für diese Gruppen primäre Datenteilmengen auf dieser Ebene bildet, mit denen anschließend Klassifikationsmodelle trainiert werden. Für sehr kleine Gruppen, wie z. B. die Gruppe $DM3612$ mit lediglich 12 Dateninstanzen, führt SPH allerdings Surrogat-Teilmengen ein, die mindestens eine Hierarchieebene höher angesiedelt sind. Die Gruppe $DM36$ auf der zweiten Ebene „Typ“ enthält beispielsweise 130 Dateninstanzen. SPH verwendet die Datenteilmenge der Gruppe $DM36$ als Surrogat, um die Gruppe $DM3612$ eine Ebene tiefer zu repräsentieren. Das bedeutet, dass ein Klassifikationsmodell mit den Daten der Gruppe $DM36$ trainiert wird und dieses Klassifikationsmodell verwendet wird, um die Klasse für neue Dateninstanzen, die zur Gruppe $DM3612$ gehören, vorherzusagen. Gruppe $DM36$ wird dabei aber nur als Surrogat für Gruppe $DM3612$ verwendet. Die benachbarte Gruppe $DM3610$ enthält mit 52 genug Dateninstanzen, die daher für diese Gruppe als primäre Datenteilmenge verwendet werden können.

Folglich enthält eine Surrogat-Teilmenge mehr Instanzen, sodass bei ihr das Risiko einer Überanpassung der Klassifikationsmodelle geringer ist. Die Surrogate sind aber weniger spezifisch, d. h. die Anzahl und Komplexität der Klassenmuster ist in einer Surrogat-Teilmenge höher als in einer primären Teilmenge. Dennoch sind die Anzahl und Komplexität der Klassenmuster in einer Surrogat-Teilmenge auf einer Ebene unterhalb des Wurzelknotens der Taxonomie immer noch geringer als im gesamten Datensatz $\mathcal{D}$. Surrogat-Teilmengen adressieren also immer noch die Gruppenheterogenität und Gruppenungleichverteilung, allerdings nicht in demselben Maße wie primäre Teilmengen. Sie stellen damit eine Art Kompromiss zwischen dem gesamten Datensatz mit mehr Daten, aber vielen komplexen Klassenmustern sowie einzelnen primären Teilmengen mit weniger Daten, aber ebenso weniger Klassenmustern dar.

Hirsch et al. [HRT+23] schlagen einen Algorithmus für SPH vor, der für einen gegebenen Datensatz $\mathcal{D}$ und eine Taxonomie die Partitionierung von $\mathcal{D}$ in Teilmengen $\mathcal{D}_j$ durchführt. Für weitere Details zu diesem Algorithmus sowie zu den Bedingungen, mit denen der Algorithmus entscheidet, wann eine primäre Teilmenge und wann eine Surrogat-Teilmenge gebildet wird, sei an dieser Stelle auf die Autorenpublikation verwiesen [HRT+23].

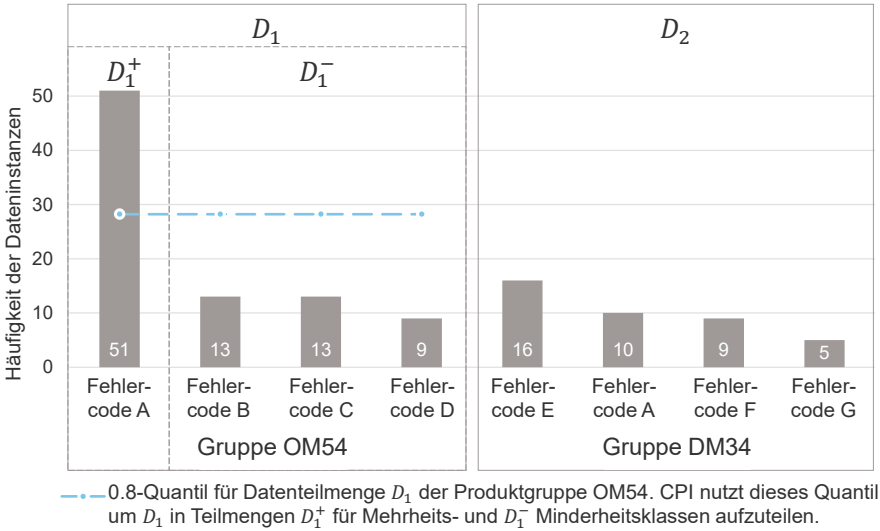


Abbildung 6.6: Beispielhafte Klassenverteilung für die Gruppen $OM54$ und $DM34$ von Verbrennungsmotoren nach der Anwendung von SPH ($\mathcal{D}_1$ und $\mathcal{D}_2$) und nach der Anwendung von CPI ($\mathcal{D}_1^+$ und $\mathcal{D}_1^-$). Die Fehlercodes A bis G stehen für einzelne Klassen in den Datenteilmengen, vgl. Hirsch et al. [HRT+23].

Klassenaufteilung gemäß Ungleichgewicht (CPI): Abbildung 6.6 zeigt beispielhaft die Klassenverteilung nach der Anwendung von SPH und der Partitionierung eines Datensatzes in Teilmengen $\mathcal{D}_1$ und $\mathcal{D}_2$ für zwei Produktgruppen $OM54$ und $DM34$ des Anwendungsfalls der Fehlerdiagnose im EoL-Prüffeld. Dabei werden als Klassenbezeichnungen abstrakte Fehlercodes A bis G verwendet, die auf ein fehlerhaftes Bauteil eines Motors verweisen [HRM20a]. In der rechten Datenteilmenge $\mathcal{D}_2$ für Gruppe $DM34$ sind die enthaltenen Fehlercodes bzw. Klassen einigermaßen ausgewogen verteilt. In der Datenteilmenge $\mathcal{D}_1$ für Gruppe $OM54$ kommt Fehlercode A mit 51 Dateninstanzen allerdings immer noch deutlich häufiger vor als alle anderen Fehlercodes. Für solche Gruppen mit einer hohen Klassenungleichverteilung wendet das in diesem Abschnitt beschriebene Verfahren noch den zweiten Schritt der *Klassenaufteilung gemäß deren Ungleichgewicht* an (engl. *Class Partitioning according to Imbalance (CPI)*). In diesem Schritt CPI werden einige der Datenteilmengen von SPH in zwei

weitere disjunkte Teilmengen aufgeteilt: Eine Teilmenge $\mathcal{D}_j^+$ für Mehrheitsklassen und eine Teilmenge $\mathcal{D}_j^-$ für Minderheitsklassen. Für die in Abbildung 6.6 links dargestellte Teilmenge $\mathcal{D}_1$ werden bspw. eine Teilmenge $\mathcal{D}_1^+$ mit allen Dateninstanzen der Mehrheitsklasse Fehlercode A gebildet sowie eine Teilmenge $\mathcal{D}_1^-$ mit allen Instanzen der Minderheitsklassen, d. h. für Fehlercodes B, C und D. Auf diese Weise werden in den einzelnen Teilmengen $\mathcal{D}_1^+$ und $\mathcal{D}_1^-$ jeweils die Klassenungleichverteilung deutlich reduziert. Werden später Klassifikationsalgorithmen separat auf die Teilmengen $\mathcal{D}_1^+$ und $\mathcal{D}_1^-$ angewendet, ist sichergestellt, dass die Algorithmen die zuvor seltenen Klassen B, C und D nicht mehr aufgrund der vorher dominanten Klasse A ignorieren.

Der Schritt CPI berechnet für jede Datenteilmenge $\mathcal{D}_j$, die SPH liefert, zunächst den Wert des Gini-Koeffizienten [Cow11; Gin21], um die Klassenungleichverteilung in $\mathcal{D}_j$ zu quantifizieren. Dieser Gini-Koeffizient liefert einen Wert zwischen 0 und 1, wobei der Wert 0 bedeutet, dass alle Klassen gleich häufig vorkommen, während höhere Werte eine stärkere Klassenungleichverteilung ausdrücken. Auf Basis eines bestimmten Schwellenwerts für den Gini-Koeffizienten entscheidet CPI, ob eine Teilmenge $\mathcal{D}_j$ weiter aufgeteilt werden soll oder nicht. Der Gini-Koeffizient der Teilmengen in Abbildung 6.6 beträgt etwa 0,48 für $\mathcal{D}_1$ und etwa 0,28 für $\mathcal{D}_2$ [HRM20a]. Für einen Schwellenwert von 0,3 wird beispielsweise die Teilmenge $\mathcal{D}_1$, die eine hohe Klassenungleichverteilung aufweist, weiter in die Teilmengen $\mathcal{D}_1^+$ und $\mathcal{D}_1^-$ aufgeteilt. Die Teilmenge $\mathcal{D}_2$ der Gruppe $DM34$, deren Klassen weitgehend gleichmäßig verteilt sind, wird mit diesem Schwellenwert hingegen nicht weiter aufgeteilt.

Mit dem p-Quantil $Q(p)$ verwendet der Schritt CPI eine weitere Metrik, deren Ergebnis als Schwellenwert zur Aufteilung einer Datenteilmenge $\mathcal{D}_j$ in eine Teilmenge $\mathcal{D}_j^+$ für Mehrheitsklassen und eine Teilmenge $\mathcal{D}_j^-$ für Minderheitsklassen dient. Die Idee ist, dass ein Wert für p festlegt wird und das p-Quantil $Q(p)$ für jede aufzuteilende Teilmenge $\mathcal{D}_j$ berechnet wird. Klassen mit mehr Dateninstanzen als der durch $Q(p)$ bestimmte Schwellenwert kommen in die Teilmenge $\mathcal{D}_j^+$ und die anderen Klassen in $\mathcal{D}_j^-$. Wie in Abbildung 6.6 angegeben ist, liegt bspw. das 0,8-Quantil ($p = 0,8$) für die Gruppe $OM54$ bei etwa 28. Dies wird dabei als Schwellenwert zur Aufteilung von $\mathcal{D}_1$ verwendet. Damit ist der Fehlercode A, der mehr als 28 Dateninstanzen hat, die einzige Mehrheitsklasse und seine Dateninstanzen werden in die Teilmenge $\mathcal{D}_1^+$ übernommen.

Die Fehlercodes B, C und D sind gemäß dieses Schwellenwerts Minderheitsklassen, sodass deren Instanzen in $\mathcal{D}_1^-$ übernommen werden. Schließlich haben alle resultierenden Teilmengen $\mathcal{D}_1^+$ und $\mathcal{D}_1^-$ eine ausgewogenere Klassenverteilung. Der Schritt CPI reduziert den Gini-Koeffizienten von 0,48 für die Teilmenge $\mathcal{D}_1$ auf Werte von 0,19 für $\mathcal{D}_1^+$ und 0,11 für $\mathcal{D}_1^-$ [HRM20a]. Weitere Details zur Berechnung von $Q(p)$ sowie zu einem Algorithmus, der den Schritt CPI formalisiert, werden von Hirsch et al. [HRM20a; HRT+23] beschrieben.

Nach der Anwendung von SPH und CPI wird pro Datenteilmenge $\mathcal{D}_j$ bzw. $\mathcal{D}_j^+$ und $\mathcal{D}_j^-$ ein separates Klassifikationsmodell trainiert. Da durch die Datenpartitionierung die Anzahl der Dateninstanzen in den einzelnen Teilmengen reduziert wird, kann das Risiko einer Überanpassung der trainierten Klassifikationsmodelle an eine zu kleine Datenmenge erhöht sein (siehe Wertung in Tabelle 6.1). Dieses Problem kann durch die Anwendung von Ensemble-Methoden für die Klassifikation, wie bspw. Random Forest, angegangen werden [Bre01; HRM19] (siehe Abschnitt 4.3.1). Aus diesem Grund und da kaum Ansätze existieren, die eine Gruppenheterogenität und Gruppenungleichverteilung adressieren, lässt das in diesem Abschnitt beschriebene Verfahren explizit zu, dass die Datenmenge pro Klassifikationsmodell geringer wird, aber dass dafür die Gruppenheterogenität und Gruppenungleichverteilung adressiert werden.

6.5.2 Zentrale Evaluationsergebnisse

In diesem Abschnitt werden die Ergebnisse einer Evaluation des Verfahrens zur Nutzung einer Taxonomie für die Datenpartitionierung zusammengefasst. Diese Evaluation wurde von Hirsch et al. [HRM20a; HRT+23] mit mehreren Datensätzen für Multi-Klassen-Klassifikationen durchgeführt. Tabelle 6.2 fasst die wesentlichen Eigenschaften dieser Datensätze zusammen. Der erste Datensatz umfasst reale Daten des Anwendungsfalls der Fehlerdiagnose im EoL-Prüffeld [HRM19; HRM20a]. Dieser Datensatz besteht aus 1050 Dateninstanzen sowie 84 Klassen und 115 Features. Der Anteil der fehlenden Feature-Werte („NA“-Werte) im gesamten Datensatz ist mit 17 % relativ hoch. Dies kann nach Hirsch et al. [HRT+23] als Indikator für ein hohes Ausmaß einer Gruppenheterogenität und Gruppenungleichverteilung gesehen werden. Zudem weist der Datensatz aus dem EoL-Prüffeld eine hohe Klassenungleichverteilung auf, wie der hohe Wert des Gini-Koeffizienten von 55 % zeigt.

Tabelle 6.2: Wesentliche Eigenschaften der für die Evaluation des Verfahrens von Hirsch et al. [HRT+23] verwendeten Datensätze.

Datensatz	Daten-instanzen	Klassen	Features	„NA“ in Features	Gini-Koeff.
aus EoL-Prüffeld	1050	84	115	17 %	55 %
Synth.: sehr ausgewogen	1050	84	100	24 %	32 %
Synth.: ausgewogen	1050	84	100	24 %	42 %
Synth.: mittel	1050	84	100	24 %	51 %
Synth.: unausgewogen	1050	84	100	24 %	54 %
Synth.: sehr unausgewogen	1050	84	100	24 %	57 %

Zudem haben Hirsch et al. [HRT+23] das Verfahren mit fünf synthetisch generierten Datensätzen mit unterschiedlichen Datenverteilungen evaluiert. Jeder synthetische Datensatz beinhaltet 1050 Dateninstanzen, 84 Klassen und 100 Features. Der Anteil der fehlenden Feature-Werte lässt mit 24 % auf ein etwas höheres Ausmaß einer Gruppenheterogenität und Gruppenungleichverteilung schließen als beim Datensatz aus dem EoL-Prüffeld. Der wesentliche Unterschied zwischen den einzelnen fünf synthethischen Datensätzen besteht darin, dass die 84 Klassen unterschiedlich verteilt sind, sodass sich auch jeweils das Verhältnis zwischen Mehrheits- und Minderheitsklassen unterscheidet. Die Datensätze werden entsprechend ihrer Klassenverteilungen als *sehr ausgewogen*, *ausgewogen*, *mittel*, *unausgewogen* und *sehr unausgewogen* bezeichnet. Die Gini-Koeffizienten bewegen sich dabei zwischen 32 % und 57 %.

Hirsch et al. [HRM20a; HRT+23] haben für alle sechs Datensätze jeweils eine theoretische und eine experimentelle Evaluation durchgeführt. In der theoretischen Evaluation untersuchen sie die Eigenschaften der Datensätze vor der Anwendung von SPH und CPI sowie die Eigenschaften der jeweiligen Datenteilmengen nach der Anwendung dieser beiden Schritte. Damit überprüfen sie, welche Auswirkungen SPH und CPI auf verschiedene Ursachen für Bias in den Daten haben, insbesondere auf die Gruppenheterogenität, Gruppenungleichverteilung und Klassenungleichverteilung.

In der experimentellen Evaluation untersuchen sie insbesondere, inwieweit SPH und CPI die Vorhersageleistung der mit den Daten bzw. Datenteilmengen trainierten Klassifikationsmodelle verbessern. Als Klassifikationsalgorithmus

wenden sie Random Forest auf die Datenteilmengen an, die SPH und CPI liefern. Die Autoren vergleichen die entsprechenden Klassifikationsmodelle mit einer Baseline, bei der Random Forest auf den jeweiligen gesamten Datensatz angewendet wird, also ohne diesen mit SPH und CPI in Datenteilmengen aufzuteilen. Hirsch et al. [HRM20a; HRT+23] vergleichen das neue Verfahren mit der Baseline, indem sie die resultierenden Klassifikationsmodelle mit zwei Evaluationsmetriken bewerten. Die erste Metrik entspricht der Korrektklassifikationsrate (engl. *Accuracy*) auf einzelnen Positionen einer Empfehlungsliste, die die wahrscheinlichsten Vorhersagen der Klassifikationsmodelle auflistet. Die zweite Metrik gibt im Anwendungsfall der Fehlerdiagnose im EoL-Prüffeld die Anzahl der Nacharbeitsversuche an (engl. *Rework Attempts*: *RA*), die im Durchschnitt benötigt werden, um ein Qualitätsproblem durch Abarbeiten dieser Empfehlungsliste zu lösen. Während der Wert der Korrektklassifikationsrate möglichst hoch sein sollte, ist bzgl. der Metrik *RA* die Reduktion dieser Anzahl an Nacharbeitsversuchen das Ziel. Für Details zur verwendeten Hard- und Software, zur Parametrisierung des neuen Verfahrens und der Klassifikationsalgorithmen sowie zur Berechnung der Evaluationsmetriken sei auf Hirsch et al. [HRM20a; HRT+23] verwiesen.

Im Folgenden werden die wichtigsten Ergebnisse dieser umfassenden Evaluation zusammengefasst, wobei Tabelle 6.3 wesentliche Kennzahlen für die ersten vier der folgenden Punkte darstellt:

- Das neue Verfahren, insbesondere der Schritt SPH, ist sehr gut geeignet, um die Gruppenheterogentität und Gruppenungleichverteilung in Daten zu adressieren. Ein Indikator dafür ist, dass das neue Verfahren den Anteil fehlender Feature-Werte („NA“-Werte) signifikant um 12 %-Punkte für den realen Datensatz aus dem EoL-Prüffeld sowie um Werte zwischen 15 und 17 %-Punkte für die synthetischen Datensätze reduziert.
- Die beiden Schritte SPH und CPI adressieren gemeinsam auch eine Klassenungleichverteilung in erheblichem Maße. So reduzieren sie den Gini-Koffizienten um 34 %-Punkte für den Datensatz aus dem EoL-Prüffeld sowie um Werte zwischen 18 und 35 %-Punkte für die synthetischen Datensätze. Dabei wird der Gini-Koeffizient umso stärker reduziert je höher die ursprüngliche Klassenungleichverteilung der Datensätze ist.

Tabelle 6.3: Zentrale Ergebnisse der Evaluation des neuen Verfahrens zur Nutzung einer Taxonomie für die Datenpartitionierung [HRT+23]. Die Tabelle zeigt für alle sechs evaluierten Datensätze, um wie viele %-Punkte das neue Verfahren den Anteil fehlender Feature-Werte („NA“-Werte) und den Wert des Gini-Koffizienten reduziert. Ebenso zeigt sie, um wie viele %-Punkte im Vergleich zur Baseline die durchschnittliche Korrektklassifikationsrate ($AvgAcc$) erhöht wird und um wie viele Punkte die Metrik *RA* reduziert wird.

Datensatz	Δ„NA“ in Features *(%-Punkte)*	ΔGini-Koeff. *(%-Punkte)*	$\Delta AvgAcc$ *(%-Punkte)*	ΔRA *(einfache Punkte)*
aus EoL-Prüffeld	-12	-34	+6,3	-0,2
Synth.: sehr ausgewogen	-17	-18	+12,9	-1,25
Synth.: ausgewogen	-17	-22	+10,5	-0,28
Synth.: mittel	-16	-26	+7,7	-0,53
Synth.: unausgewogen	-16	-29	+4,6	-0,25
Synth.: sehr unausgewogen	-15	-35	+6,8	-0,32

- Das neue Verfahren mit den Schritten SPH und CPI übertrifft die Baseline für alle sechs Datensätze hinsichtlich der Vorhersageleistung der Klassifikationsmodelle. Es erhöht die durchschnittliche Korrektklassifikationsrate um Werte zwischen 4,6 und 12,9 %-Punkte.
- Darüber hinaus reduzieren SPH und CPI die Metrik *RA* um Werte zwischen 0,2 und 1,25 Punkte. Dies bringt erhebliche praktische Vorteile für reale Anwendungsfälle mit sich. Es reduziert zum Beispiel die Anzahl der Nacharbeitsversuche im EoL-Prüffeld und damit auch die Kosten für solche Nacharbeitsversuche [HRM20b]. Hierbei ist noch zu beachten, dass in Anwendungsfällen datengetriebener medizinischer Diagnosen (siehe Abschnitt 3.3) die Metrik *RA* sinngemäß der Anzahl an zusätzlichen medizinischen Untersuchungen entspricht, die ein Arzt zur Verifizierung der Klassenvorhersagen durchführt und die aber für Patienten unangenehme Nebenwirkungen mit sich bringen können. Folglich haben SPH und CPI auch in anderen Anwendungsdomänen Vorteile, da sie z. B. die Anzahl dieser zusätzlichen für Patienten unangenehmen medizinischen Untersuchungen reduzieren können.

- Eine detaillierte Untersuchung der Evaluationsergebnisse zeigt, dass der Schritt SPH insbesondere für kleinere domänenspezifische Entitätsgruppen, die in den ursprünglichen Datensätzen unterrepräsentiert sind, die Korrektklassifikationsrate und den Wert der Metrik *RA* deutlich stärker verbessert als für größere Gruppen. Dies zeigt erneut, dass dieser Schritt sehr gut geeignet ist, um die Gruppenungleichverteilung zu adressieren.
- Der Schritt CPI erzielt sehr gute Ergebnisse speziell für größere Gruppen, wenn in deren gruppenspezifischen Datenteilmengen nach SPH eine einzige Mehrheitsklasse einen sehr großen Anteil aller Dateninstanzen ausmacht. In solchen Fällen erhöht CPI durch die weitere Aufteilung der Datenteilmengen die Korrektklassifikationsrate speziell für alle anderen Klassen, die zuvor von dieser einen Mehrheitsklasse dominiert wurden.
- Weiterhin haben Hirsch et al. [HRT+23] noch die Rechenlaufzeit und Skalierbarkeit des neuen Verfahrens hinsichtlich größerer Datenmengen mit 10 000, 100 000, 500 000 und 1 000 000 Dateninstanzen evaluiert. Die entsprechenden Ergebnisse untermauern die positiven Effekte des neuen Verfahrens. Tatsächlich steigern SPH und CPI für Anwendungsfälle mit größeren Datensätzen nicht nur die Vorhersageleistung der Klassifikationsmodelle, sondern reduzieren auch die Rechenlaufzeit der gesamten Datenpipeline mit der Datenvorbereitung und dem Modelltraining, um bis zu 40 %. Dies liegt insbesondere daran, dass das Modelltraining nach der Anwendung von SPH und CPI auf kleineren Datenteilmengen und damit mit reduziertem Rechenaufwand durchgeführt wird.

6.5.3 Übertragbarkeit des Verfahrens auf andere Anwendungsdomänen

In diesem Abschnitt werden zentrale Aspekte hinsichtlich der Übertragbarkeit des Verfahrens und seiner Schritte SPH und CPI auf verschiedene Anwendungsdomänen und Daten diskutiert. Zunächst werden die Voraussetzungen erörtert, die eine formale Repräsentation des Domänenwissens erfüllen muss, damit SPH grundsätzlich in anderen Anwendungsfällen anwendbar ist. Dem schließt sich eine Diskussion an, ob diese Voraussetzungen für das Domänenwissen gelten, das in verschiedenen der in Kapitel 3 untersuchten Anwendungsfälle verfügbar ist. Dies betrifft die in Abschnitt 3.1 beschriebenen industriellen An-

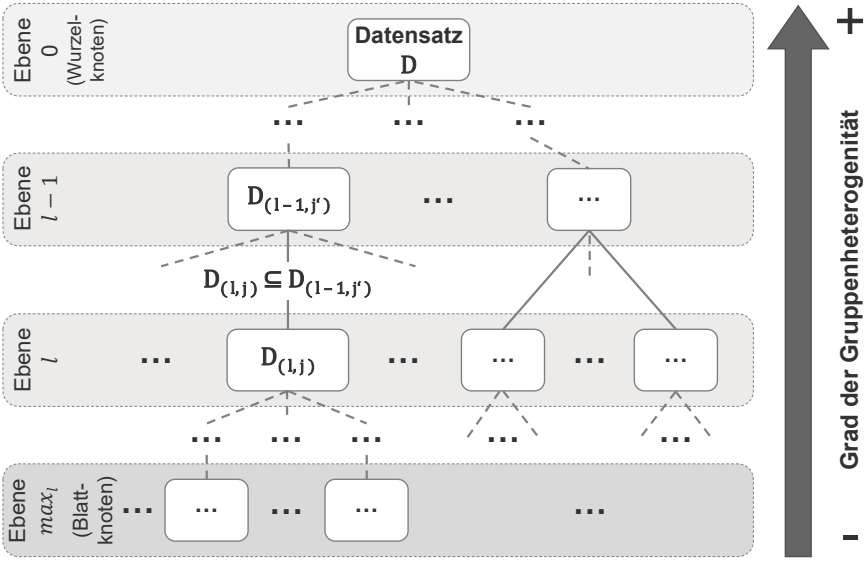


Abbildung 6.7: Abstraktes hierarchisches Modell zur Illustration der Voraussetzungen von SPH für das Domänenwissen, vgl. [HRT+23].

wendungsfälle, aber auch die in Abschnitt 3.3 untersuchten Anwendungsfälle für datengetriebe medizinische Diagnosen. Abschließend folgt eine Diskussion, welche Eigenschaften die jeweiligen Daten aufweisen müssen, damit die Schritte SPH und CPI auf diese Daten anwendbar sind [HRT+23].

Voraussetzungen für die Taxonomie beim Schritt SPH: Der Schritt SPH benötigt als formal repräsentiertes Domänenwissen ein Modell, das in einer hierarchischen Baumstruktur aufgebaut ist, bspw. als semantisches Netz mit hierarchischen Relationen. Abbildung 6.7 veranschaulicht die wichtigsten Voraussetzungen von SPH auf Basis eines abstrakten hierarchischen Modells:

1. Der einzige Wurzelknoten auf der obersten Ebene $l = 0$ umfasst den gesamten Datensatz $\mathcal{D}$.
2. Jeder Kindknoten (l, j), d. h. der j-te Knoten auf einer Ebene $l \geq 1$ unterhalb des Wurzelknotens beinhaltet eine Teilmenge der Daten des jeweiligen Elternknotens $(l-1, j')$ eine Ebene darüber, d. h., $\mathcal{D}_{(l,j)} \subseteq \mathcal{D}_{(l-1,j')}$

3. Alle Blattknoten befinden sich auf derselben tiefsten Ebene $l = max_l$.
4. Damit SPH die Gruppenheterogenität und Gruppenungleichverteilung adressieren kann, muss das hierarchische Modell die Partitionierung des Datensatzes $\mathcal{D}$ in Teilmengen ermöglichen, in denen jeweils nur noch Daten einer oder zumindest weniger domänenspezifischen Gruppen enthalten sind. Dies wird dadurch erreicht, dass das Modell die domänenspezifischen Gruppen des jeweiligen Anwendungsfalls hierarchisch organisiert. Dabei ist jedem Knoten eine bestimmte Menge der Gruppen sowie deren Daten zugeordnet, wobei jeder Kindknoten (l, j) eine Untermenge der Gruppen und deren Daten des jeweiligen Elternknotens $(l-1, j')$ umfasst. Somit nimmt die Anzahl der Gruppen und daher auch der Grad der Gruppenheterogenität und Gruppenungleichverteilung in den Teilmengen $\mathcal{D}_{(l,j)}$ mit jeder tieferen Ebene l ab.

Vorhandenes Domänenwissen in anderen Anwendungsdomänen: Gängige Modelle zur Beschreibung domänenspezifischer Gruppen und ihrer Beziehungen sind Wissensgraphen und semantische Netze, z. B. Taxonomien oder Ontologien [Qui67; Rei10; Sow91] (siehe auch Abschnitt 2.2). Solche Modelle beschreiben u. a. hierarchische Beziehungen zwischen über- und untergeordneten Begriffen, wobei jeder Begriff für eine bestimmte domänenspezifische Entitätsgruppe steht. Diese hierarchischen Beziehungen zwischen Begriffen bzw. Entitätsgruppen erfüllen in den meisten Anwendungsdomänen die oben diskutierten Voraussetzungen für das zur Anwendung von SPH benötigte Domänenwissen. In den meisten Fällen können anhand der Begriffe einer Hierarchieebene die jeweiligen Daten bzw. Datenteilmengen identifiziert werden, die zu den entsprechenden domänenspezifischen Entitätsgruppen gehören. Weiterhin sind untergeordnete Begriffe, d. h. Kindknoten in der Hierarchie, spezifischer als die ihnen zugeordneten übergeordneten Begriffe, d. h. die Elternknoten [Rei10; Sow91]. Das bedeutet auch, dass die Daten eines untergeordneten Kindknotens in der Regel eine Teilmenge von spezifischeren Gruppen repräsentieren, als die Daten, die durch den jeweiligen übergeordneten Elternknoten repräsentiert werden. Insgesamt nimmt somit der Grad der Gruppenheterogenität in den Daten entlang des Pfads vom Wurzelknoten zu den Blattknoten des hierarchischen Modells ab.

Wie in Abschnitt 3.2.2 erwähnt, verfügt nahezu jedes produzierende Unternehmen über eine Dokumentation der unternehmenseigenen Produktfamilie [AK04]. Diese Dokumentation ist häufig als eine Art hierarchisches semantisches Netz strukturiert, z. B. als Produkttaxonomie oder Thesaurus, das alle Voraussetzungen zur Anwendung von SPH erfüllt [HRT+23]. Insbesondere strukturiert solch eine Dokumentation der Produktfamilie verschiedene Produktvarianten in hierarchisch organisierten Produktgruppen. Als Beispiel beschreiben Sun et al. [SRYD14] eine solche Produktaxonomie, die für SPH geeignet ist. Diese Taxonomie gliedert Produkte und ihre Daten in drei Ebenen: (1) Produktkategorien, (2) Produktunterkategorien und (3) Produkttypen. Hier kann SPH einen Datensatz $\mathcal{D}$, der alle Produkte umfasst, in mehrere primäre Datenteilmengen aufteilen, die nur noch Daten der Produkttypen auf der untersten Ebene der Taxonomie enthalten. Auf diese Weise verringert SPH die Anzahl und Komplexität der Klassenmuster in den resultierenden Datenteilmengen und reduziert so die negativen Auswirkungen der Variantenvielfalt von Produktgruppen und der damit einhergehenden Gruppenheterogenität. Nur wenn eine Datenteilmenge für einen Produkttyp zu wenig Dateninstanzen enthält, bildet SPH Surrogat-Teilmengen auf einer höheren Ebene der Produkttaxonomie, also für Produktunterkategorien oder sogar für Produktkategorien.

In Anwendungsfällen datengetriebener medizinischer Diagnosen entsprechen die durch das Domänenwissen zu modellierenden Entitätsgruppen je nach Anwendungsfall bspw. verschiedenen Ethnizitäten bzw. Hautfarben [CRM+20], unterschiedlichen Geschlechter [SG21] oder spezifischen Altersgruppen der jeweiligen Patienten [MGM20] (siehe Abschnitt 3.3.2). Semantische Modelle zur hierarchischen Strukturierung dieser Entitätsgruppen beschreiben häufig eine flache Hierarchie mit nur einer einzigen Ebene für Blattknoten direkt unter dem Wurzelknoten. Abbildung 6.8 zeigt eine solche flache Taxonomie, die verschiedene Typen menschlicher Hautfarben anhand der Fitzpatrick-Skala [Fit88] organisiert. Diese Fitzpatrick-Skala ist eine der prominentesten Einteilungen von Hautfarben, die auch im Bereich der Dermatologie weitgehend akzeptiert ist [Jab04]. Sie teilt Hautfarben in sechs verschiedene Typen hinsichtlich ihrer Reaktion auf ultraviolettes Licht ein.

Die in Abbildung 6.8 dargestellte Taxonomie erfüllt im Beispiel der datengetriebenen Diagnose bestimmter Krebsarten [CRM+20] (siehe Abschnitt 3.3.2)

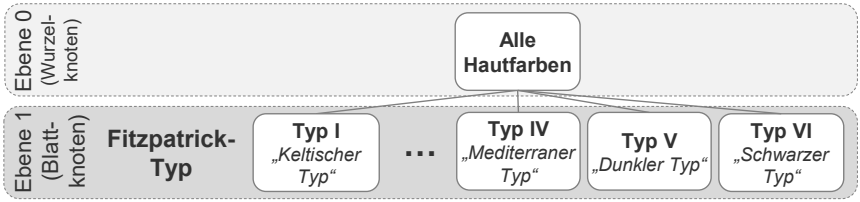


Abbildung 6.8: Flache Taxonomie zur Kategorisierung von Hautfarben gemäß des Fitzpatrick-Typs, vgl. [HRT+23].

alle der oben definierten Voraussetzung für die Anwendung von SPH. Die Ebene 0 umfasst einen Wurzelknoten, der den gesamten Datensatz $\mathcal{D}$ zu allen Hautfarben enthält. Die Ebene 1 darunter unterteilt die Daten in einzelne Teilmengen $\mathcal{D}_j \subseteq \mathcal{D}$ für jeden der Fitzpatrick-Typen I bis VI [Fit88]. Werden Klassifikationsmodelle anstatt mit dem ganzen Datensatz $\mathcal{D}$ mit diesen Teilmengen $\mathcal{D}_j$ für spezifische Typen von Hautfarben trainiert, adressiert dies die Gruppenheterogenität. Insbesondere können die Klassenmuster für Krebsarten besser differenziert werden, wenn nur Trainingsdaten eines einzelnen Hautfarbtyps verwendet werden [CRM+20]. Ebenso adressiert diese eine Gruppenungleichverteilung, da z. B. in einer Teilmenge $\mathcal{D}_j$ für eine dunklere Hautfarbe die für diese Hautfarbe spezifischen Klassenmuster nicht mehr wie im ganzen Datensatz $\mathcal{D}$ unterrepräsentiert sind bzw. nicht mehr von den häufiger vorkommenden Mustern für hellere Hautfarben überlagert werden.

Dieses Beispiel zeigt, dass SPH auch anwendbar ist, wenn das Domänenwissen durch ein flaches hierarchisches Modell mit nur einer Blattebene unterhalb des Wurzelknotens repräsentiert wird. SPH erzeugt dann primäre Teilmengen $\mathcal{D}_j$ für die Gruppen dieser Blattebene. Der Unterschied zu einer nicht flachen Hierarchie besteht darin, dass SPH den gesamten Datensatz $\mathcal{D}$ als Surrogat-Datenmenge verwendet, falls für einen spezifischen Blattknoten zu wenig Daten zur Verfügung stehen. In diesem Fall adressiert SPH für diese einzelnen Surrogat-Datenmengen weder die Gruppenheterogenität noch die Gruppenungleichverteilung. Dies kann durch Hinzufügen einer Ebene zwischen der Wurzel und der Blattebene der Taxonomie behoben werden. In der in Abbildung 6.8 dargestellten Taxonomie können die Knoten auf dieser Ebene bspw. jeweils zwei oder drei der Fitzpatrick-Hauttypen zusammenfassen.

Notwendige Eigenschaften der Daten: Das in diesem Abschnitt vorgestellte Verfahren zur Nutzung einer Taxonomie für die Partitionierung von Daten ist so generisch, dass es grundsätzlich auf alle möglichen Arten von Daten $\mathcal{D}$ anwendbar ist, die sich in einzelne Teilmengen $\mathcal{D}_j \subseteq \mathcal{D}$ aufteilen lassen. Für den Schritt SPH gibt es darüber hinaus keine weiteren Einschränkungen. Der nächste Schritt CPI nutzt als statistische Metriken den Gini-Koeffizienten und das p-Quantil, um zu entscheiden, ob und wie die Datenteilmengen noch weiter für Mehrheits- und Minderheitsklassen aufgeteilt werden sollen. Diese beiden Metriken können auf alle Daten mit einer diskreten oder kontinuierlichen Klassenverteilung angewendet werden [Cow11; FLG+13].

Auch wenn beide Schritte SPH und CPI auf nahezu beliebige Daten anwendbar sind, wurden ihre Effekte von Hirsch et al [HRM20a; HRT+23] bisher nur für tabellarische Daten mit numerischen oder kategorischen Features evaluiert. Für andere Daten, etwa für Zeitreihendaten oder Bilddaten, muss noch untersucht werden, ob die Evaluationsergebnisse übertragbar sind.

6.6 Ontologiebasierte Ansätze zur Spezifikation von Datenvorbereitungs-Pipelines

Um den hohen Aufwand zu reduzieren, den Domänenexperten mit gängigen Software-Tools für die Spezifikation von Datenvorbereitungs-Pipelines aufbringen müssen (siehe Abschnitt 4.2.4), existieren Ansätze, die eine Spezifikation der Pipelines auf Basis von Ontologien vereinfachen. Einige Ansätze ermöglichen es, Datenvorbereitungs-Pipelines und deren Datenvorbereitungsoperationen mit gängigen Software-Tools zu spezifizieren und dabei mit Begriffen aus einer domänenspezifischen Ontologie zu annotieren [MCD+05; OOD+12]. Alternativ existieren Ansätze, mit denen die Pipelines bzw. Operationen selbst in einer ontologiebasierten Sprache definiert werden können [SBE+21; SES23].

Ansätze der ersten Kategorie, mit denen Datenvorbereitungs-Pipelines mit Begriffen aus einer domänenspezifischen Ontologie annotiert werden, finden sich insbesondere im Bereich der wissenschaftlichen Workflows [MCD+05; OOD+12; WAH+07]. Hierbei werden existierende Software-Tools für wissenschaftliche Workflows genutzt, um zunächst den Datenfluss zwischen den Workflow-Aktivitäten zu spezifizieren [LWMB09]. Dabei müssen Domänenex-

perten die Daten, die von den Workflow-Aktivitäten gelesen und ausgegeben werden, nicht im Detail spezifizieren, sondern lediglich mit Begriffen aus der jeweiligen Ontologie abstrakt beschreiben [MCD+05; OOD+12; WAH+07]. Ebenso kann der Datenfluss mit Relationen einer Ontologie annotiert werden, um somit semantische Abhängigkeiten zwischen den Ausgabedaten einer Aktivität und den Eingabedaten einer anderen Aktivität zu beschreiben [BL05]. Die abstrakten, semantischen Annotationen der Workflow-Aktivitäten und des Datenflusses können über regelbasierte Ansätze auf konkrete ausführbare Workflow-Spezifikationen abgebildet werden [RLS+06; RRS+11].

Stach et al. [SBE+21] schlagen mit BARENTS (*tailorable data preparation zone for Data Lakes*) einen Ansatz der oben zweitgenannten Kategorie vor. BARENTS nutzt keine domänenspezifische Ontologie, wie sie in Abschnitt 2.2 beschrieben ist, die also relevante Begriffe einer Anwendungsdomäne sowie die Beziehungen zwischen diesen Begriffen formalisiert [Rei10; Sow91]. Stattdessen bietet BARENTS eine ontologiebasierte Sprache, mit der kleine, relativ einfache Regeln für die Datenvorbereitung als Ontologie spezifiziert werden können. Jede dieser Regeln besteht aus drei Teilen: (1) Der Datenquelle, aus der die Daten stammen, (2) einer Datenvorbereitungsoperation, die auf die Daten angewendet wird, und (3) die Datensenke, in der die Ergebnisdaten gespeichert werden sollen. Mehrere solcher einzelnen Regeln können dabei zu einer größeren Datenvorbereitungs-Pipeline zusammengesetzt werden [SES23]. Alle zu einem konkreten Anwendungsfall definierten Regeln zur Datenvorbereitung werden in einer sog. BARENTS-Ontologie zusammengefasst [SBE+21].

Abbildung 6.9 zeigt eine schematische Darstellung einer Ontologie von BARENTS, die eine Regel für die Datenvorbereitung spezifiziert [SES23]. Die Begriffe dieser Ontologie können dabei entweder über partitative Relationen (*ist-teil-von*, durchgezogene Linien in Abbildung 6.9) oder über eigens von BARENTS neu definierte Relationen (BARENTS-spezifische Relationen, gestrichtelte Linien) verknüpft werden. Der erste Begriff beschreibt das entsprechende Quellsystem, z. B. ein Datenbanksystem, aus dem die Daten geladen werden. Die aus dieser Quelle zu extrahierenden Dateninstanzen werden über weitere Begriffe spezifiziert, die jeweils mit einer BARENTS-spezifischen Relation *'Quelle'* dem Quellsystem zugeordnet werden. Jede Dateninstanz wird mit einer partitativen Relation mit einem Begriff für die Operation verknüpft, die auf

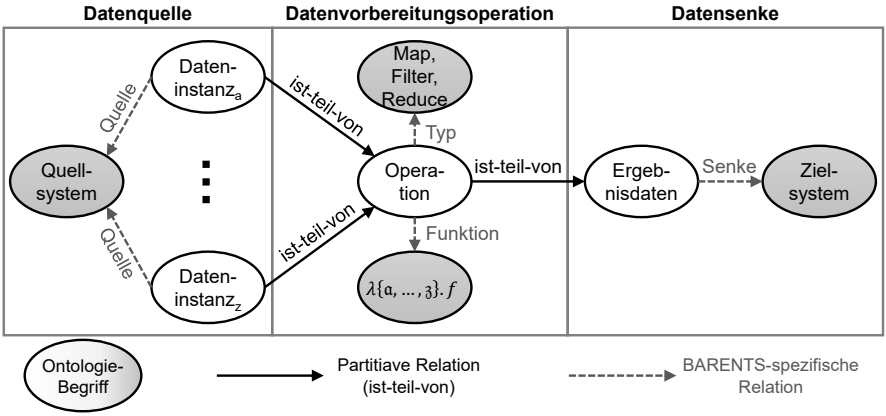


Abbildung 6.9: Schematische Darstellung einer in der BARENTS-Ontologie spezifizierten Regel zur Datenvorbereitung, vgl. Stach et al. [SES23].

die Dateninstanzen angewendet werden soll. Eine Operation wird dabei mit einer BARENTS-spezifischen Relation *'Typ'* auf eine von von drei in BARENTS zulässigen Operatoren aus dem Bereich der funktionalen Programmierung abgebildet [SBE+21]: (i) Map-Operator, der eine Funktion zur Transformation aller Dateninstanzen anwendet, (ii) Filter-Operator, der Dateninstanzen gemäß eines Selektionsprädikats filtert oder (iii) Reduce-Operator, der z. B. mehrere Dateninstanzen bzw. deren Features zu einem Wert aggregiert. Weiterhin können noch beliebige benutzerdefinierte Typen von Funktionen in einer BARENTS-Ontologie spezifiziert werden. Zusätzlich wird jede Operation noch mittels der Relation *'Funktion'* mit einer auszuführenden Funktion verknüpft, die in der formalen Sprachen des Lamda-Kalküls [Chu41] spezifiziert wird. Schließlich wird die Operation mittels einer weiteren partitativen Relation auf die Ergebnisdaten der Datenvorbereitung abgebildet. Diese Ergebnisdaten werden mit einer BARENTS-spezifischen Relation *'Senke'* mit dem Zielsystem assoziiert, in das sie gespeichert werden sollen.

Die Definition der Datenquelle, der extrahierten Dateninstanzen sowie der Ergebnisdaten und Datensenke kann für Domänenexperten weiter vereinfacht werden, indem diese in BARENTS mittels domänenspezifischer Begriffe aus einem externen semantischen Netz verschlagwortet werden können [SBE+21].

Im Gegensatz zur BARENTS-Ontologie muss dieses externe semantische Netz, wie in Abschnitt 2.2 beschrieben, die relevanten Begriffe einer Anwendungsdomäne sowie die Beziehungen zwischen diesen Begriffen darstellen. Stach et al. [SBE+21] beschreiben ein Beispiel, in der die Daten in einer BARENTS-Ontologie über Begriffe aus einer anderen Ontologie der Lebensmittelchemie verschlagwortet werden [BB18]. Diese Ontologie beschreibt Begriffe und deren Beziehungen für Bestandteile von Lebensmitteln, die Allergien auslösen können. Beispiele für solche Begriffe sind verschiedene Arten von Nüssen sowie allergieauslösende Peptide.

Bzgl. der Definition der Datenvorbereitungsoperation reduziert die Einschränkung auf drei zentrale Operatoren der funktionalen Programmierung zwar die Auswahlmöglichkeiten und damit auch die Komplexität für Domänenexperten. Sie müssen jedoch nach wie vor mit der funktionalen Programmierung und den zugrundeliegenden formalen Sprachen des Lamda-Kalküls [Chu41] vertraut sein. Hier kann nicht davon ausgegangen werden, dass Domänenexperten aus jeder Anwendungsdomäne diese formalen Sprachen beherrschen. Aus diesem Grund haben Stach et al. [SES23] noch ein Empfehlungs- bzw. Expertensystem entworfen, das mit BARENTS spezifizierte Datenvorbereitungsoperationen vorschlägt. Dabei gehen die Autoren davon aus, dass mehrere Domänenexperten oder andere Personen für einen bestimmten Anwendungsfall bereits eine gewisse Anzahl an Regeln für Datenvorbereitungsoperationen in einer BARENTS-Ontologie spezifiziert haben. Diese bereits spezifizierte BARENTS-Ontologie dient als Fallbasis für das Expertensystem. Andere Domänenexperten müssen mit BARENTS lediglich die ihnen zur Verfügung stehenden Daten als Datenquelle spezifizieren. Das Expertensystem von Stach et al. [SES23] fragt die gesamte BARENTS-Ontologie nach bereits existierenden Regeln für Datenvorbereitungsoperationen ab, die auf Datenquellen ausgeführt wurden, die der von den Nutzern spezifizierten neuen Datenquelle ähnlich sind. Die Nutzer können anschließend die vorgeschlagenen BARENTS-Spezifikationen von Datenvorbereitungsoperationen auf ihre eigenen Daten anpassen.

Die in diesem Abschnitt diskutierten Ansätze zielen primär darauf ab, die Komplexität der Spezifikation von Datenvorbereitungs-Pipelines zu reduzieren, indem sie mit einer Ontologie von den zugrundeliegenden heterogenen Datenformaten und von weiteren komplexen Datencharakteristika abstrahie-

ren [Rei17; RSM14a; SBE+21]. Sie ermöglichen somit Domänenexperten, auf relativ einfache Weise Datenvorbereitungs-Pipelines bzw. Datenvorbereitungsoperationen zu spezifizieren und adressieren damit eines der zentralen Probleme entsprechender Software-Tools. Allerdings wurde bei keinem der Ansätze bisher untersucht, welche Auswirkungen ihre Nutzung auf die verschiedenen Ursachen für Bias in Daten haben. Somit ist unklar, ob sie die in Abschnitt 4.2.4 diskutierten Probleme existierender Software-Tools bzgl. der Ursachen für Bias in Daten lösen können. Daher werden die in diesem Abschnitt diskutierten Ansätze in Tabelle 6.1 weder positiv noch negativ bewertet.

6.7 Zusammenfassung

Gängige Ansätze zur Datenvorbereitung nutzen als Domänenwissen eine Regelbasis, semantische Netze oder Simulationen, um entweder das Feature-Engineering zu verbessern, Labels in Daten festzulegen oder Daten mit neuen Instanzen, Features oder Labels anzureichern. Diese Ansätze helfen insbesondere das Risiko einer Überanpassung der trainierten ML-Modelle an eine kleine Datenmenge zu reduzieren bzw. falsch dokumentierte Labels zu identifizieren und zu korrigieren (siehe Tabelle 6.1). Allerdings haben sie negative Effekte auf eine Klassenungleichverteilung, Gruppenheterogenität und Gruppenungleichverteilung in Daten. Eine Datenpartitionierung in einzelne Teilmengen mit einer Constraint-basierten Clusteranalyse ist einer der wenigen Ansätze, die prinzipiell eine Gruppenheterogenität adressieren können. Der Ansatz kommt aber schnell an seine Grenzen, da viele Constraints auf Ebene der Dateninstanzen spezifiziert werden müssen. Dies führt dazu, dass Domänenexperten lediglich Constraints zur Abdeckung der Dateninstanzen von Mehrheitsgruppen beschreiben, sodass Minderheitsgruppen vernachlässigt werden.

Aufgrund der Defizite dieser Ansätze wurde in einem im Rahmen des vom Autor dieser Habilitationsschrift betreuten Projekts ein neues Verfahren zur Nutzung von Domänenwissen in der Datenvorbereitung entworfen [HRT+23] (siehe Abschnitt 6.5). Analog zur Constraint-basierten Clusteranalyse partitioniert dieses Verfahren die Daten in einzelne Teilmengen. Dabei nutzt es mit einer Taxononomie aber eine andere Form von Domänenwissen, das nicht für einzelne Dateninstanzen, sondern nur einmal in der jeweiligen Anwen-

dungsdomäne definiert werden muss und in vielen Anwendungsdomänen bereits vorliegt. Diese Taxonomie muss die domänenspezifischen Gruppen der jeweiligen Anwendungsdomäne hierarchisch organisieren. Das neue Verfahren partitioniert mit dem ersten Schritt SPH die Trainingsdaten in mehrere Teilmengen, sodass in jeder Teilmenge nur Daten einer domänenspezifischen Gruppe enthalten sind. Einzelne der Datenteilmengen werden im zweiten Schritt CPI jeweils noch weiter in eine Teilmenge für Mehrheitsklassen und eine Teilmenge für Minderheitsklassen aufgeteilt. Insgesamt adressieren beide Schritte SPH und CPI drei zentrale Ursachen für Bias in Daten industrieller Anwendungsfälle: Gruppenheterogenität, Gruppenungleichverteilung und Klassenungleichverteilung. Dadurch steigert das Verfahren die Vorhersageleistung der trainierten Klassifikationsmodelle erheblich. Dies wird durch eine umfassende Evaluation von Hirsch et al. [HRM20a; HRT+23] bestätigt.

Die in Abschnitt 6.6 diskutierten Ansätze reduzieren zwar auf Basis von Ontologien bzw. ontologiebasierten Sprachen die Komplexität der Spezifikation von Datenvorbereitungs-Pipelines und Datenvorbereitungsoperationen. Allerdings sind bei diesen Ansätzen die Auswirkungen auf die verschiedenen Ursachen für Bias in Daten noch nicht erforscht und daher unbekannt.

Kapitel 7

Nutzung von Domänenwissen zur Datenanalyse und Modellbildung

In diesem Kapitel werden schließlich Ansätze zur Nutzung formal repräsentierten Domänenwissens im Schritt der Datenanalyse und Modellbildung von Datenanalyseprozessesen diskutiert. Abbildung 7.1 kategorisiert diese Ansätze hinsichtlich der für die Datenanalyse durchzuführenden Aufgabe sowie hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens. Ebenso wird in diesem Kapitel diskutiert, welche der in Abschnitt 3.2 beschriebenen Datencharakteristika die Ansätze erfolgreich adressieren und auf welche sie keine oder sogar negative Effekte haben. Dabei weisen die diskutierten Ansätze erneut primär Effekte auf verschiedene Ursachen für Bias in Daten auf. Tabelle 7.1 fasst die Ergebnisse dieser Diskussion zusammen.

7.1 Modellverwaltungsplattformen als Expertensysteme zum Vorschlag von ML-Modellen

Modellverwaltungsplattformen nutzen ein Repository, das bereits erstellte ML-Modelle sowie die zugehörigen Daten verwaltet und mit Metadaten beschreibt [VRBT13; VSL+16; ZCD+18]. Abschnitt 7.1.1 fasst zusammen, welche Metadaten typischerweise in Modellverwaltungsplattformen verwendet

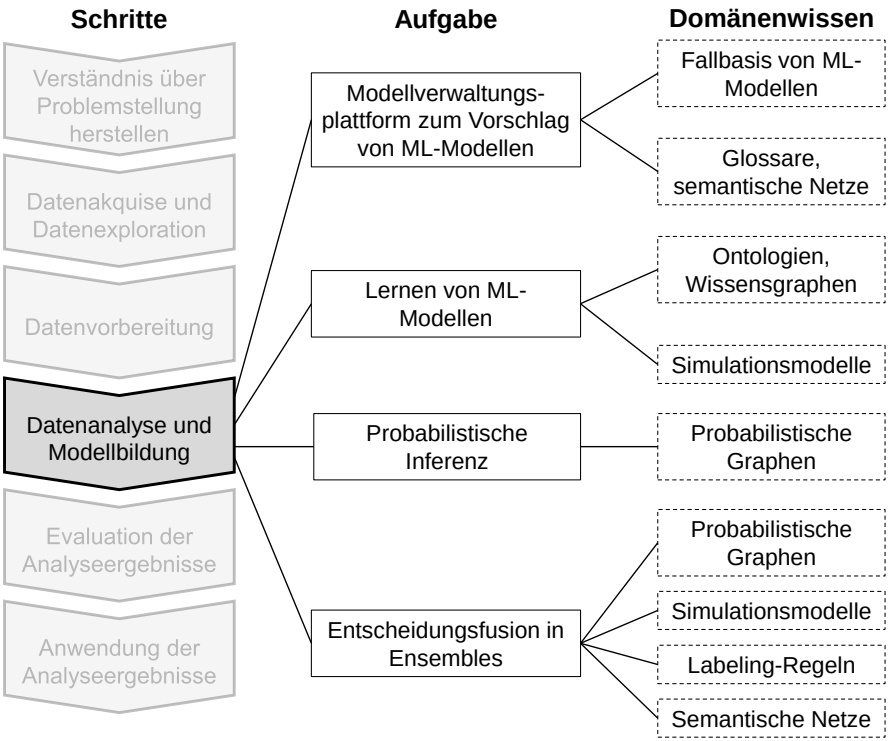


Abbildung 7.1: Durch Domänenwissen unterstützte Aufgaben in der Datenanalyse und Modellbildung sowie Kategorisierung der Aufgaben und entsprechender Ansätze hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens.

werden. Die Abschnitte 7.1.2 und 7.1.3 stellen zwei Ansätze für Modellverwaltungsplattformen vor, die im Rahmen von Projekten entstanden sind, die vom Autor dieser Habilitationsschrift betreut wurden [VRWM23; WHR20]. Diese Ansätze bieten auf Basis eines Repositorys von ML-Modellen ein Experten- oder Empfehlungssystem, das für bestimmte Datensätze und Analyseszenarien passende ML-Modelle vorschlägt. Der erste, AssistML [VRWM23; VWRM21], bietet einen automatisierten Ansatz zur Suche nach ML-Modellen an, der speziell für Nutzer mit geringen Kenntnissen zum maschinellen Lernen geeignet ist. Der zweite Ansatz, die *Model Management Platform* (MMP) [WHR20; WR20], bettet in Metadaten für ML-Modelle Domänenwissen aus Glossaren oder se-

Tabelle 7.1: Ansätze zur Nutzung von Domänenwissen in der Datenanalyse sowie Auswirkungen auf die Ursachen für Bias in Daten: + Positiver Effekt; 0 kein bzw. neutraler Effekt; − negativer Effekt.

Ansatz	**Klassen-ungleich-verteilung**	**Gruppen-hetero-genität**	**Gruppen-ungleich-verteilung**	**Falsche Labels**	**Kleine Daten-menge**
Modellverwaltungs-plattformen (AssistML, MMP)	0	+	+	0	0
Modelltraining mit Ontologien, Wissensgraphen, Simulations-modellen	−	0	−	+	+
Inferenz mit probabilistischen Graphen	−	0	−	+	+
Entscheidungsfusion in Ensembles (PUSION)	0	0	0	+	+

mantischen Netzen ein, um die Suche nach passenden ML-Modellen weiter zu vereinfachen. Abschnitt 7.1.4 fasst die zentralen Ergebnisse der Evaluation dieser beiden Ansätze zusammen. Dem folgt eine zusätzliche Diskussion in Abschnitt 7.1.5, inwieweit die beiden Ansätze die in Abschnitt 3.2 beschriebenen Datencharakteristika industrieller Anwendungsfälle adressieren können. Abschnitt 7.1.6 schließt mit einer Diskussion, ob und wie die Ansätze auf verschiedene Anwendungsdomänen und deren Daten übertragbar sind.

7.1.1 Metadaten zur Beschreibung von ML-Modellen

Gängige Beispiele für Modellverwaltungsplattformen, die ML-Modelle in ihren Repositorys mit Metadaten beschreiben, sind OpenML [VRBT13], ModelDB [VSL+16] und MLflow [ZCD+18]. Die Metadaten können prinzipiell direkt in Feldern oder Annotationen der Datenformate gespeichert werden [Web21], die auch zur Serialisierung der ML-Modelle genutzt werden, wie

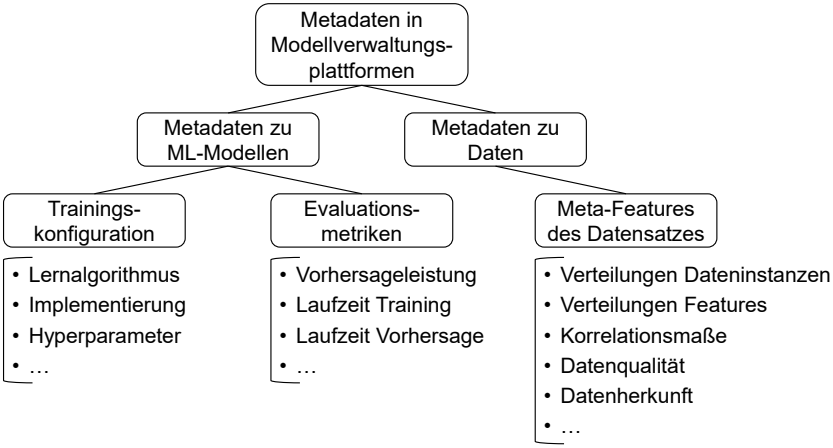


Abbildung 7.2: Zentrale Beispiele von Metadaten zur Beschreibung von ML-Modellen und Daten in Modellverwaltungsplattformen.

bspw. die XML-basierte *Predictive Model Markup Language* (PMML[1]) oder das *Standard Format for Packaging Models* von MLflow [ZCD+18]. Viele Modellverwaltungsplattformen nutzen in ihren Repositorys jedoch Datenbankysteme, um die Metadaten in spezifischen Metadatenschemata abzubilden und zu speichern [EMN+15; KŁd+15; PSD14; VRBT13; VS10; VSL+16; WHR20]. Mit dem W3C ML-Schema [PEŁ+18] existiert eine Empfehlung für ein Metadatenschema in Modellverwaltungsplattformen.

Abbildung 7.2 zeigt eine Kategorisierung zentraler Beispiele solcher Metadaten zur Beschreibung von ML-Modellen und von zugehörigen Daten. Die Metadaten beschreiben zum einen die für ein ML-Modell verwendete Trainingskonfiguration, d. h. den jeweiligen Lernalgorithmus, dessen Implementierung, z. B. mittels Bibliotheken einer Programmiersprache, sowie die genutzten Werte der Hyperparameter des Algorithmus [PEŁ+18]. Darüber hinaus können in den Metadaten die Ergebnisse verschiedener Evaluationsmetriken erfasst werden. In überwachten Lernverfahren sind dies insbesondere Metriken zur Bestimmung der Vorhersageleistung eines ML-Modells, z. B. die Korrektklassifikationsrate, die auf Test- oder Validierungsdaten berechnet wurden. Zusätzlich können die Laufzeit oder Rechenkomplexität sowohl des Trainings des ML-

[1] PMML: https://dmg.org/pmml/v4-4-1/GeneralStructure.html

Modells mit einem Trainingsdatensatz als auch einer mit dem Modell erfolgten Vorhersage für neue Dateninstanzen erfasst werden [VRWM23].

Modellverwaltungsplattformen verwalten in ihren Repositorys auch Metadaten zu den Datensätzen, mit denen die ML-Modelle erzeugt wurden [PEŁ+18; VRBT13; VRWM23]. Ein wesentlicher Teil dieser Metadaten umfasst sog. Meta-Features [RGS+22], die spezifische Charakteristika eines Datensatzes auf einer Meta-Ebene zusammenfassen. Beispiele solcher Meta-Features, die häufig in Modellverwaltungsplattformen genutzt werden, sind die Anzahl und Verteilung von Dateninstanzen, Verteilungen oder Varianzen von Feature-Typen bzw. Feature-Werten [TFSM23; VRWM23] sowie Korrelationsmaße zwischen Features oder Labels [PEŁ+18]. Weiterhin können Meta-Features die Datenqualität quantifizieren oder Informationen zur Datenherkunft beschreiben [SBK+17].

7.1.2 AssistML-Ansatz als Expertensystem zur Empfehlung von ML-Lösungen

Villanueva et al. [VRWM23; VWRM21] schlagen mit AssistML einen automatisierten Ansatz vor, der ML-Lösungen empfiehlt, die für eine bestimmte überwachte Lernaufgabe, z. B. für eine Klassifikation oder eine Regression, sowie für einen bestimmten Datensatz geeignet sind. Eine ML-Lösung umfasst dabei nicht nur ein ML-Modell, sondern eine vollständige Pipeline, die sämtliche Algorithmen und Software-Tools zum Feature-Engineering, zum Modelltraining und zur Modellanwendung kombiniert und konfiguriert. AsisstML basiert auf einem Modell-Repository, in dem bereits existierende ML-Lösungen sowie die dazugehörigen Daten verwaltet und mit Metadaten beschrieben werden. Es nutzt dieses Repository als Fallbasis für ML-Modelle, um ein Experten- oder Empfehlungssystem zu realisieren, das Nutzern eine Liste von ML-Lösungen vorschlägt, die für ihre Daten geeignet sind und zu guten Vorhersageleistungen der ML-Modelle führen. Abbildung 7.3 illustriert die Vorgehensweise von AssistML, die nachfolgend überblicksmäßig erläutert wird. Für weitere Details sei auf die Publikationen von Villanueva et al. [VRWM23; VWRM21] verwiesen.

Im Modell-Repository von AssistML wird zu jeder ML-Lösung der gesamte Qellcode dieser Lösung gespeichert, d. h. die jeweiligen Skripte und Konfigurationsdateien, die zur Ausführung des Feature-Engineerings, des Modelltrainings und der Modellanwendung benötigt werden. Zusätzlich werden auch die Datensätze, auf die die ML-Lösungen angewendet wurden, im Repository

vorgehalten. Sowohl die ML-Lösungen als auch die Datensätze werden in AssistML mit umfassenden Metadaten beschrieben. So beschreiben die Metadaten zu den ML-Lösungen u. a. die jeweilige Lernaufgabe, d. h. ob es sich bspw. um eine bestimmte Regressionsaufgabe oder um eine binäre Klassifikation, Multi-Klassen-Klassifikation oder Multi-Label-Klassifikation handelt. Weiterhin werden die in Abschnitt 7.1.1 erläuterte Trainingskonfiguration der ML-Modelle sowie Ergebnisse bestimmter Evaluationsmetriken hinterlegt.

Als Metadaten für Daten bietet AssistML einerseits Meta-Features, die einen Datensatz als Ganzes charakterisieren. Dies sind bspw. die Anzahl an Dateninstanzen, Anzahl an Features sowie die relativen Anteile bestimmter Feature-Typen im Datensatz. Dabei unterscheidet AssistML zwischen numerischen, kategorischen und textuellen Feature-Typen, Typen für Datumsangaben sowie Zeitreihendaten oder Bilddaten. Darüber hinaus liefert AssistML umfassende Meta-Features, mit denen jedes einzelne Feature eines Datensatzes näher beschrieben wird. Diese Meta-Features unterscheiden sich je nach konkretem Feature-Typ. Zum Beispiel können numerische Features durch die Anzahl an Ausreißern oder durch den Anteil fehlender Feature-Werte charakterisiert werden, während für kategorische Features die Anzahl der verwendbaren Kategorien oder Verteilungs- bzw. Ungleichverteilungsmaße bzgl. dieser Kategorien relevant sind. Villanueva et al. [VRWM23] beschreiben für alle der oben genannten Feature-Typen noch weitere Beispiele solcher Meta-Features.

Nutzer, die für neue Daten eine ML-Lösung benötigen, können AssistML eine Anfrage mit Informationen zu ihren Daten und zur gewünschten ML-Lösung stellen. AssistML nutzt das Modell-Repository als Fallbasis für ML-Lösungen, um den Nutzern diejenigen ML-Lösungen vorzuschlagen, die für ihre Daten geeignet sind und zu guten Vorhersageleistungen führen. Dabei fokussiert AssistML mit sog. *Citizen Data Scientists* eine Nutzergruppe mit grundlegenden, aber begrenzten Kenntnissen zum maschinellen Lernen [Grö18]. Dies sind typischerweise Domänenexperten, wie bspw. Ingenieure, die im Laufe ihrer Tätigkeiten Grundkenntnisse zu Datenanalysen erlernt haben. Daher ist AssistML gezielt so entwickelt worden, dass diese Nutzer in einer Anfrage nur ein Minimum an abstrakten Informationen angeben müssen. Die erste Eingabe spezifiziert die Lernaufgabe, die von der gewünschten ML-Lösung umgesetzt werden soll, also z. B. eine binäre Klassifikation oder eine Multi-

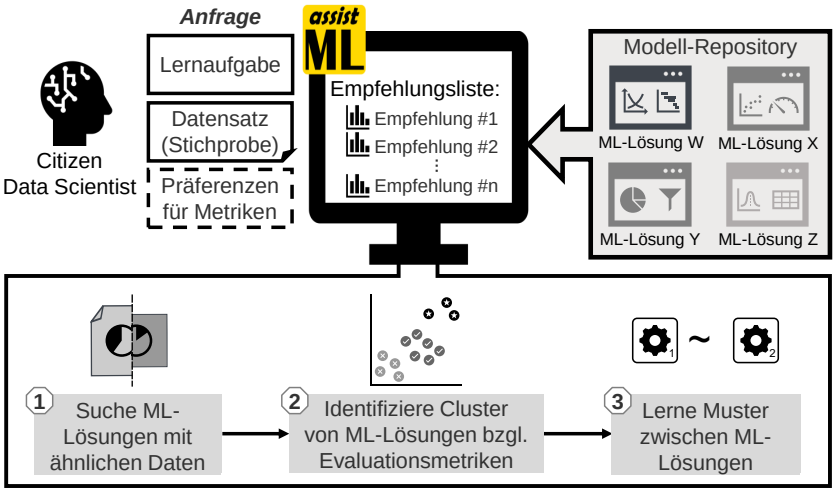


Abbildung 7.3: Vorgehensweise des AsssistML-Ansatz zum Vorschlag von ML-Lösungen, vgl. Villanueva [Vil21].

Klassen-Klassifikation. Als zweite Eingabe geben Nutzer den Datensatz oder eine Stichprobe dieses Datensatzes an, für den sie eine ML-Lösung suchen. Die dritte, optionale Eingabe spezifiziert Präferenzen bzgl. bestimmter Evaluationsmetriken, insbesondere in welchen Bereichen die Werte dieser Metriken liegen müssen, damit die Nutzer eine ML-Lösung als akzeptabel ansehen.

Abbildung 7.3 zeigt unten die drei zentralen Schritte, die AssistML durchführt, um im Repository ML-Lösungen zu finden, die zur Nutzeranfrage passen. Im ersten Schritt sucht AssistML nach ML-Lösungen, die (a) die in der Anfrage spezifizierte Lernaufgabe unterstützen und die (b) auf Datensätze angewendet wurden, die ähnliche Eigenschaften wie der in der Anfrage spezifizierte Datensatz haben. Um die Ähnlichkeit zwischen den Datensätzen zu bestimmen, extrahiert AssistML einige Meta-Features aus dem Datensatz der Nutzeranfrage und vergleicht diese Meta-Features mit denen der Datensätze im Repository [VRWM23]. Zum Beispiel überprüft AssistML, ob der Datensatz des Nutzers sowie die Datensätze im Repository ähnliche Verteilungen der Feature-Typen oder auch ähnliche Werteverteilungen einzelner Features aufweisen.

Im zweiten Schritt reduziert AssistML die ML-Lösungen aus dem vorherigen Schrit auf diejenigen, die gemäß den Metadaten im Modell-Repository

ausreichend gute Werte in den vom Nutzer spezifizierten Evaluationsmetriken aufweisen. Zuvor teilt der Schritt die ML-Lösungen mittels einer Clusteranalyse in mehrere Cluster auf, sodass in jedem Cluster ML-Lösungen enthalten sind, die ähnliche Werte in den Evaluationsmetriken haben. Die Clusteranalyse ist insbesondere notwendig, da Modell-Repositorys wie die von AssisML oder von anderen Modellverwaltungsplattformen tausende ML-Lösungen enthalten können [VRBT13; VRWM23; VSL+16]. Durch die Clusteranalyse müssen nicht mehr die vielen ML-Lösungen, sondern wenige Cluster von Lösungen bzgl. ihrer Evaluationsmetriken mit den Präferenzen in der Nutzeranfrage verglichen werden. Schließlich übergibt AssistML dem nächsten Schritt jedes Cluster von ML-Lösungen, bei dem die Mehrheit der Lösungen die vom Nutzer spezifizierten Anforderungen bzgl. der Evaluationsmetriken erfüllen [VRWM23].

Im dritten Schritt lernt AssistML Muster innerhalb der Metadaten der vom Schritt zuvor übergebenen ML-Lösungen, indem der Ansatz Assoziationsregeln zwischen den Metadaten und den Werten der Evaluationsmetriken erzeugt. Diese Assoziationsregeln bieten den Nutzern einfach zu interpretierende Erklärungen, welche Eigenschaften einer ML-Lösungen bei welchen Meta-Features der Datensätze zu welchen Ergebnissen in den Evaluationsmetriken führen. Ein Beispiel für eine so gelernte Regel ist *„WENN ML-Lösung neuronale Netze mit 10-20 Hyperparametern nutzt DANN ist Trainingszeit sehr hoch“*. Diese Regel bildet den genutzten Algorithmus und dessen Trainingskonfiguration auf Wertebereiche einer Evaluationsmetrik ab.

Auf Basis der Ergebnisse dieser drei Schritte erstellt AssistML eine Empfehlungsliste mit den besten ML-Lösungen für die in der Nutzeranfrage spezifizierte Lernaufgabe sowie für dessen Datensatz und Präferenzen bzgl. der Evaluationsmetriken. Die ML-Lösungen werden in dieser Liste absteigend anhand des Durchschnitts der normierten Werte aller in der Anfrage angegebenen Evaluationsmetriken sortiert [VRWM23]. Zusätzlich erzeugt AssistML noch eine detaillierte Beschreibung jeder einzelnen ML-Lösung, die sich der Nutzer bei Bedarf ansehen kann. Dabei wird jeder Wert der spezifizierten Evaluationsmetriken im Detail dargestellt. Zusätzlich zeigt die detaillierte Beschreibung weitere intuitive Erklärungen bzgl. der Funktionsweise einer ML-Lösung, z. B. welche Features des jeweiligen Datensatzes primär dazu beigetragen haben, die dokumentierten Werte in den Evaluationsmetriken zu erzielen. Weitere

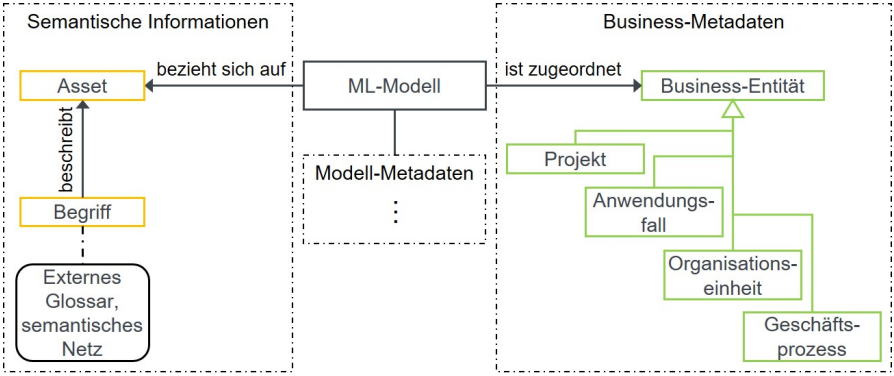


Abbildung 7.4: Abstrakte Sicht auf zentrale Teile des Metadatenmodells der Modellverwaltungsplattform von Weber [Web21].

Erklärungen umfassen insbesondere die im dritten Schritt erzeugten Assoziationsregeln in den Metadaten der ML-Lösung. Auf Basis der Rangfolge in der Empfehlungsliste sowie auf Basis der detaillierten Beschreibungen der ML-Lösungen kann der Nutzer einzelne ML-Lösungen auswählen und auf den eigenen Datensatz anwenden, um für diesen ML-Modelle zu erzeugen.

7.1.3 Plattform mit semantischen Informationen in Modell-Metadaten

Als Teil der *Model Management Platform* (MMP) schlagen Weber et al. [WHR20] ein Metadatenmodell vor, das u. a. semantische Informationen aus Glossaren oder semantischen Netzen sowie Informationen aus dem Business-Kontext eines Unternehmens zur Beschreibung von ML-Modellen verwendet. Abbildung 7.4 zeigt die zentralen Teile dieses Metadatenmodells. Für die Modell-Metadaten, die eine Erweiterung des W3C ML-Schemas [PEŁ+18] sind, sei an dieser Stelle auf Weber [Web21] verwiesen. Nachfolgend werden die semantischen Informationen sowie die Business-Metadaten näher beschrieben und wie diese Domänenexperten bei der Suche nach passenden ML-Modellen unterstützen.

In den semantischen Informationen wird ein ML-Modell mit einem domänenspezifischen Asset verknüpft, in dessen Kontext das ML-Modell für Vorhersagen genutzt wird [Web21]. Dies kann bspw. eine Anlage, eine Maschine, ein Produkt oder ein Produktionsprozesss sein. Handelt es sich z. B. um eine Maschine,

dann bezieht sich das Asset im Metadatenmodell auch auf die an der Maschine verbauten Sensoren, die Daten liefern, um damit ein ML-Modell zu erzeugen. Zur detaillierteren Beschreibung wird ein Asset mit domänenspezifischen Begriffen verschlagwortet. Weber [Web21] diskutiert dazu insbesondere Verschlagwortungen mit Begriffen aus externen Glossaren oder semantischen Netzen aus dem Kontext Industrie 4.0 [BHV14]. Dabei betrachtet der Autor insbesondere Begriffe aus den OPC UA Companion Specifications[1] für die Verschlagwortung von Maschinen oder Anlagen. Ebenso können Begriffe aus der Taxonomie ECLASS[2] genutzt werden, um Produkte oder Dienstleistungen eines Unternehmens als Assets im Metadatenmodell zu beschreiben. Darüber hinaus ist noch eine Verschlagwortung der Assets mit Begriffen aus einer mit AutomationML [Dra10] spezifizierten Ontologie zur Anlagenplanung oder mit Begriffen aus Produkthierarchien [HRT+23; SRYD14] möglich.

Diese Verknüpfung der ML-Modelle mit domänenspezifischen Assets sowie die Verschlagwortung dieser Assets mit Begriffen aus Glossaren oder semantischen Netzen unterstützen speziell Domänenexperten mit geringen Kenntnissen zur Datenanalyse bei der Suche nach ML-Modellen. Es ermöglicht bspw. die einfache Suche nach einem ML-Modell, das für eine neue Maschine benötigt wird. Dazu kann die Plattform von Weber et al. [WR20] nach ML-Modellen abgefragt werden, die laut der verknüpften Assets für eine andere Maschine entwickelt wurde, die den gleichen Maschinentyp wie die neue Maschine hat. Maschinen des gleichen Typs liefern häufig die gleichen Sensordaten, sodass das entsprechende ML-Modell oder dessen Trainingskonfiguration auch auf die Daten der neuen Maschine anwendbar ist [Web21].

Das Metadatenmodell von Weber [Web21] unterstützt zudem die Verknüpfung von ML-Modellen mit zugeordneten Business-Entitäten, d. h. mit Projekten, Anwendungsfällen, Organisationseinheiten und Geschäftsprozessen. Das Projekt beschreibt das Entwicklungsprojekt, in dem die zugrundeliegende ML-Lösung entwickelt wurde [VGR20]. Weiterhin werden in den Metadaten zusätzlich zum Asset noch weitere Details des jeweiligen Anwendungsfalls beschrieben, für den das ML-Modell entwickelt wurde [Web21]. Dies kann bspw. eine Beschreibung des zugrundeliegenden Produktionsprozesses sowie der mit

[1] OPC UA Companion Specifications: https://opcfoundation.org/about/opc-technologies/opc-ua/ua-companion-specifications/

[2] ECLASS: https://eclass.eu/

dem ML-Modell in diesem Prozess durchzuführende Aufgabe sein, wie bspw. die Vorhersage der Produktqualität. Diese Beschreibung des Anwendungsfalls im Metadatenmodell ermöglicht Domänenexperten neben der Suche nach ML-Modellen für konkrete Assets auch die einfache Suche nach ML-Modellen, die bestimmte Anwendungsfälle unterstützen.

Mit Metadaten zu Organisationseinheiten und Geschäftsprozessen können die ML-Modelle innerhalb einer Unternehmensstruktur verortet werden. Auf dieser Grundlage hat Weber [Web21] eine Modelllandschaftskarte entwickelt, die u. a. für Business-Analysten eine Übersicht bietet, in welchen Organisationseinheiten oder Geschäftsprozessen welche ML-Modelle entwickelt wurden oder zum Einsatz kommen. Damit können Business-Analysten Optimierungsmöglichkeiten aufdecken, wenn z. B. mehrere ML-Modelle aus verschiedenen Organisationseinheiten für den gleichen Geschäftsprozess eingesetzt werden.

7.1.4 Zentrale Evaluationsergebnisse für AssistML und die MMP

In diesem Abschnitt werden die zentralen Ergebnisse der Evaluation bzw. der Validierung von AssistML und der MMP zusammengefasst, die in den Publikationen von Villanueva et al. [VRWM23; VWRM21] und Weber et al. [WHR20] ausführlich diskutiert werden.

Ergebnisse der Evaluation von AssistML: Villanueva et al. [VRWM23] haben zur Evaluation von AssistML 228 ML-Lösungen für insgesamt fünf Datensätze umgesetzt und diese in ihrem Modell-Repository mit Metadaten beschrieben. Tabelle 7.2 fasst wesentliche Eigenschaften dieser ML-Lösungen und Datensätze zusammen. Für die fünf Datensätze werden in der Tabelle jeweils in Klammern die verwendeten Feature-Typen aufgelistet. Die Datensätze decken Lernaufgaben für binäre Klassifikationen und für Multi-Klassen-Klassifikationen ab. Die 228 ML-Lösungen nutzen jeweils einen von insgesamt neun Lernalgorithmen sowie verschiedene Hyperparameter. Die Autoren evaluieren AssistML, indem sie für zwei neue Datensätze[1] [2], die nicht im Repository enthalten sind, nach passenden ML-Lösungen für eine binäre Klassifikation suchen. AssistML liefert für beide Datensätze jeweils eine Empfehlungsliste von ML-Lösungen aus dem

[1] Datensatz 'steel-plates-fault': `https://www.openml.org/d/1504`
[2] Datensatz 'adult': `https://www.openml.org/d/1590`

Tabelle 7.2: Inhalte des zur Evaluation von AssistML verwendeten Modell-Repositorys, vgl. Villanueva et al. [VRWM23]. Die Datensätze sind mit den angegebenen Suchbegriffen auf Kaggle zu finden.

Elemente (Anzahl)	Inhalte im Repository
Datensätze (5)	'kick auction' (kategorisch, numerisch) 'bank marketing' (kategorisch, numerisch) 'human activity recognition' (numerisch) 'gas drift' (numerisch) 'amazon fine food reviews' (numerisch, textuell, kategorisch, Datum)
Lernaufgaben (2)	Binäre Klassifikation, Multi-Klassen-Klassifikation
Lernalgorithmen (9)	Entscheidungsbäume, Random Forest, neuronale Netze, logistische Regression, Naive Bayes, K-nächster-Nachbar, Gradient-Boosting-Maschinen, Support-Vektor-Maschinen, allgemeines lineares Modell
ML-Lösungen (228)	Einzelne ML-Lösungen für obige Datensätze, Lernaufgaben, Lernalgorithmen sowie verschiedene Hyperparameter

Repository. Diese ML-Lösungen werden anschließend auf die beiden neuen Datensätze angewendet, um für diese Daten ML-Modelle zu erzeugen.

Mittels einer Kreuzvalidierung bestimmen Villanueva et al. [VRWM23] die Werte mehrerer Evaluationsmetriken sowohl für die 228 ML-Lösungen im Repository als auch für die ML-Modelle, die mit den von AssistML vorgeschlagenen ML-Lösungen für die beiden neuen Datensätze erzeugt wurden. Die Vorhersageleistung der ML-Modelle berechnen sie über die Korrektklassifikationsrate (engl. *Accuracy*), die Wahr-Positiv-Rate (engl. *Recall*), den positiven Vorhersagewert (engl. *Precision*) sowie den F1-Score. Außerdem bestimmen sie jeweils die Zeit für die Ausführung von AssistML und zum Training des ML-Modells. Als zentrale Baseline vergleichen die Autoren die mit AssistML erzeugten ML-Modelle mit solchen, die mit dem AutoML-System H2O [LP20] erzeugt wurden. Neben den oben genannten Evaluationsmetriken untersuchen sie dabei auch die Komplexität der erzeugten ML-Modelle. Folgende zentrale Evaluationsergebnisse lassen sich hierbei zusammenfassen [VRWM23]:

- Für den Datensatz 'steel-plates-fault' schlägt AssistML ML-Lösungen vor, die zu einer besseren Vorhersageleistung führen als die ML-Modelle, die vom AutoML-System H2O erzeugt werden. Das beste von H2O erzeug-

te ML-Modell liefert bspw. eine Korrektklassifikationsrate von 79,7 %. AssistML schlägt dem Nutzer insgesamt elf ML-Lösungen vor, die zu Korrektklassifikationsraten von 85,3 bis 100 % führen. Damit besitzt das schlechteste von AssistML vorgeschlagene Modell immer noch eine bessere Vorhersageleistung als das beste von H2O. Beim Datensatz 'adult' liegen AssistML und das AutoML-System H2O weitgehend gleichauf. Das beste von H2O erzeugte ML-Modell liefert eine Korrektklassifikationsrate von 83,6 %, während der beste Vorschlag von AssistML 84,9 % erreicht.

- Ein wesentlicher Vorteil des AssistML-Ansatzes ist seine deutlich geringere Laufzeit. Das AutoML-System H2O benötigt bei jedem der beiden Datensätze etwas mehr als 41 Minuten, um optimierte ML-Modelle zu erzeugen. Die Ausführungszeit von AssistML zum Vorschlag von ML-Lösungen und zum anschließenden Training der ML-Modelle ist hingegen bei keinem der Datensätze höher als 42 Sekunden.
- Weiterhin erzeugt das AutoML-System H2O deutlich komplexere ML-Modelle als AssistML. Für den Datensatz 'steel-plates-fault' besteht das beste ML-Modell von H2O aus einem Ensemble von insgesamt fünf Teilmodellen, die mit vier verschiedenen Lernalgorithmen erzeugt wurden. Beim Datensatz 'adult' werden im Ensemble sogar 54 Teilmodelle kombiniert. Speziell für Nutzer mit geringen Kenntnissen zum maschinellen Lernen, wie Citizen Data Scientists, ist es schwer, die Funktionsweise dieser mit H2O erzeugten ML-Modelle zu verstehen. AssistML schlägt hingegen einfachere ML-Modelle vor, die jeweils nur mit einem Lernalgorithmus erzeugt wurden. Weiterhin bietet AssistML für jede vorgeschlagene ML-Lösung eine Beschreibung, die für Citizen Data Scientist intuitive Erklärungen der Funktionsweise der ML-Lösung liefert [VRWM23].

Fallbeispiel zur Validierung der *Model Management Platform* (MMP): Weber et al. [WHR20] diskutieren die Funktionsweise sowie die Vorteile ihrer *Model Management Platform* (MMP) anhand eines Fallbeispiels, das sie von dem auch von Villanueva et al. [VRWM23] verwendeten Datensatz 'steel-plates-fault' von OpenML ableiten. In diesem Fallbeispiel entwickeln Datenwissenschaftler eines Unternehmens zunächst ein neues ML-Modell, das für einen bestimmten Maschinentyp in der Stahlblechfertigung die Ausfallzeiten der Maschinen vor-

hersagt. Als Datengrundlage verwenden sie die Daten von Sensoren, die in den Maschinen dieses Typs verbaut wurden. Die Datenwissenschaftler laden das neue ML-Modell in die MMP hoch und annotieren es dort mit beschreibenden Metadaten. Unter anderem verknüpfen sie es mit einem Asset für den konkreten Maschinentyp und verschlagworten dieses Asset mit geeigneten Begriffen aus einem OPC UA Informationsmodell, das Maschinen dieses Typs sowie die darin verbauten Sensoren beschreibt. Weiterhin ordnen sie das ML-Modell in den Business-Metadaten der Organisationseinheit 'Maschineninstandhaltung' und dem Geschäftsprozess 'Maschinenwartung' des Unternehmens sowie dem konkreten Anwendungsfall 'Vorhersage von Maschinenausfallzeiten' zu.

Die Autoren diskutieren, wie diese Metadaten der MMP speziell Domänenexperten mit geringen Kenntnissen zur Datenanalyse helfen, geeignete ML-Modelle für ihre Anwendungsfälle zu finden [WHR20]. Dazu betrachten sie im Fallbeispiel eine Domänenexpertin, die ein ML-Modell zur Vorhersage von Ausfallzeiten für eine andere Maschine des gleichen Maschinentyps benötigt. Diese Domänenexpertin kann hierzu eine Anfrage an die MMP stellen, in der sie nach ML-Modellen dieses Maschinentyps sowie des Anwendungsfalls 'Vorhersage von Maschinenausfallzeiten' sucht. Die MMP findet daraufhin direkt das zuvor von den Datenwissenschaftlern erzeugte ML-Modell, da dieses mit zur Anfrage passenden Metadaten beschrieben ist. In diesem Fall müssen weder die Domänenexpertin noch die Datenwissenschaftler einen größeren Aufwand betreiben, um ein neues ML-Modell zu erzeugen.

Weiterhin erörtern Weber et al. [WHR20], wie die MMP Business-Analysten bei der Erstellung eines Berichts unterstützt, der zusammenfasst, welche ML-Modelle im Unternehmen für Anwendungsfälle einer vorausschauenden Maschinenwartung verwendet werden. Außerdem soll der Bericht die ML-Modelle den jeweiligen Organisationseinheiten und Geschäftsprozessen des Unternehmens zuordnen. Business-Analysten können die ML-Modelle in der MMP anhand einer Stichwortsuche bzgl. der gewünschten Anwendungsfälle suchen. Die MMP unterstützt die Erstellung des Berichts bspw. mit der Visualisierung einer Modelllandschaftskarte, in der die zu den gewünschten Anwendungsfällen gefundenen ML-Modelle direkt den jeweiligen Organisationseinheiten und Geschäftsprozessen zugeordnet werden. Folglich müssen die Business-Analysten selbst keinen Aufwand mehr betreiben, um diese Zuordnung zu erstellen.

Tabelle 7.3: Zusammenfassung der Meta-Features oder Metadaten, mit denen AssistML bzw. die MMP die jeweiligen Ursachen für Bias in Daten adressieren, sowie entsprechende allgemeine Meta-Features, die weder von AssistML noch der MMP genutzt werden.

	Meta-Features in AssistML	**Metadaten in der MMP**	**allgemeine Meta-Features**
Klassen-ungleich-verteilung	—	—	Verteilungsmaße für Klassenlabels (z. B. Gini-Koeffizient)
Gruppen-heterogenität	Anteil fehlender Feature-Werte	Mit Assets verknüpftes Domänenwissen	Komplexitätsmaße
Gruppen-ungleich-verteilung	Anteil fehlender Feature-Werte	Mit Assets verknüpftes Domänenwissen	Verteilungsmaße für Gruppen
Falsche Labels	—	—	Anteil an Ausreißer in Klassenlabels
Kleine Datenmenge	—	—	Anzahlen an Instanzen, Klassen, Features

7.1.5 Adressierung der Datencharakteristika industrieller Anwendungsfälle

In diesem Abschnitt wird diskutiert, inwieweit AssistML [VRWM23] und die MMP [WHR20] die in Abschnitt 3.2 beschriebenen Datencharakteristika industrieller Anwendungsfälle adressieren können. Der primäre Fokus dieser Diskussion liegt, wie bei den anderen in diesem Kapitel erörterten Ansätzen, auf verschiedenen Ursachen für Bias in Daten (siehe Tabelle 7.1). Dabei stellt sich im Kontext von Modellverwaltungsplattformen insbesondere die Frage, ob AssistML und die MMP mit ihren Metadaten die zielgerichtete Suche nach ML-Modellen bzw. nach ML-Lösungen ermöglichen, die mit den jeweiligen Ursachen für Bias in Daten umgehen können. Tabelle 7.3 fasst in der zweiten und dritten Spalte zusammen, mit welchen Metadaten bzw. Meta-Features die beiden Ansätze die jeweiligen Datencharakteristika adressieren. In der vierten Spalte nennt die Tabelle Meta-Features, die allgemein zur Charakterisierung der Ursachen für Bias in Daten genutzt werden können, die aber weder von AssistML noch von der MMP unterstüzt werden. Dies zeigt auf, inwieweit die beiden Ansätze noch um entsprechende Meta-Features ergänzt werden können.

AssistML: Für einen neuen Datensatz sucht AssistML im Repository nach ML-Lösungen, die für Datensätze mit ähnlichen Charakteristika wie die des neuen Datensatzes entwickelt wurden [VRWM23]. Dabei bestimmt AssistML die Ähnlichkeit der Datensätze über spezifische Meta-Features. Die Evaluationsergebnisse von Villanueva et al. [VRWM23] zeigen, dass die von AssistML gefundenen ML-Lösungen, deren Datensätze gemäß der verwendeten Meta-Features eine hohe Ähnlichkeit zum neuen Datensatz aufweisen, auch zu einer hohen Vorhersageleistung für diesen neuen Datensatz führen. Die in diesem Abschnitt zu diskutierende Frage kann also darauf reduziert werden, ob die von AssistML zur Bestimmung der Ähnlichkeit von Datensätzen verwendeten Meta-Features ausreichen, um die Ursachen für Bias in Daten zu charakterisieren. Ist dies der Fall, kann AssistML prinzipiell ML-Lösungen vorschlagen, die diese komplexen Datencharakteristika adressieren können und bei entsprechenden Datensätzen eine gute Vorhersageleistung erzielen.

Eine Klassenungleichverteilung kann generell durch Meta-Features wie die Anzahl an Dateninstanzen, die pro Klasse im Datensatz enthalten sind, oder durch daraus berechnete Verteilungsmaße wie den Gini-Koeffizienten [Cow11; Gin21] quantifiziert werden. Als Indikator für falsch dokumentierte bzw. verrauschte Klassenlabels können bspw. die Anzahl oder der Durchschnitt der im Datensatz enthaltenen Ausreißer in den Klassenlabels herangezogen werden. Analog ist bei Datensätzen mit einer geringen Anzahl an Dateninstanzen, einer hohen Anzahl an Klassen sowie einer hohen Anzahl an Features abschätzbar, dass das Risiko einer Überanpassung der ML-Modelle an die zu kleine Datenmenge relativ hoch ist. Bei diesen drei Ursachen für Bias in Daten müssen also u. a. Meta-Features berücksichtigt werden, die spezifische Aussagen über die Anzahl und Verteilungen der Klassenlabels im Datensatz erlauben. AssistML berücksichtigt zur Bestimmung der Ähnlichkeit von Datensätzen allerdings nur Meta-Features, die die Verteilungen von Feature-Typen in den Datensätzen quantifizieren oder die einzelne Features näher charakterisieren. Beispielsweise umfassen die Meta-Features von AssistML zwar Verteilungsmaße für kategorische Features, aber keine Verteilungsmaße für Klassenlabels [VRWM23]. AssistML ist daher weder in der Lage, eine Klassenungleichverteilung noch falsch dokumentierte Klassenlabels oder das Risiko einer Überanpassung der ML-Modelle an eine kleine Datenmenge zu adressieren (siehe Tabelle 7.1).

AssistML nutzt mit der Anzahl fehlender Feature-Werte ein Meta-Feature, das nach Hirsch et al. [HRT+23] als Indikator für das Ausmaß einer Gruppenheterogenität und einer Gruppenungleichverteilung verwendet werden kann. Es handelt sich bei diesem Meta-Feature jedoch nur um einen Indikator, da noch nicht umfassend untersucht wurde, welche konkreten Zusammenhänge zwischen dem Anteil fehlender Feature-Werte und dem Ausmaß einer Gruppenheterogenität und Gruppenungleichverteilung bestehen [HRT+23]. Ebenso können mit diesem Indikator nur beide Ursachen für Bias gemeinsam, aber nicht unabhängig voneinander charakterisiert werden.

Zur isolierten Charakterisierung der Gruppenheterogenität können nach Treder-Tschechlov et al. [TRSM23] die von Ho und Basu [HB02] vorgeschlagenen Komplexitätsmaße herangezogen werden. Diese sagen aus, wie schwierig es ist, die Klassenmuster in einem Datensatz voneinander zu separieren. Ein hohes Ausmaß einer Gruppenheterogenität hat zur Folge, dass Klassenmuster schwer voneinander separierbar sind. Folglich kann mit diesen Komplexitätsmaßen häufig im Umkehrschluss auf das Ausmaß einer Gruppenheterogenität geschlossen werden. Allerdings unterstützt AssistML keine Komplexitätsmaße als Meta-Features, sodass lediglich der Indikator bzgl. fehlender Feature-Werte verwendet werden kann. Um die Gruppenungleichverteilung isoliert zu charakterisieren, können allgemein Verteilungsmaße, wie bspw. der Gini-Koeffizient, auf die Gruppen in Datensätzen angewendet werden. Dazu muss allerdings bekannt sein, nach welchen Merkmalen die Dateninstanzen in den Datensätzen den jeweiligen domänenspezifischen Gruppen zugeordnet werden können. AssistML bietet hierzu keine Meta-Features oder andere Möglichkeiten.

Auch wenn AssistML die Ursachen für Bias in Daten nicht vollumfassend adressiert, zeigen die Evaluationsergebnisse von Villanueva et al. [VRWM23] dennoch, dass ihr Ansatz ML-Lösungen vorschlägt, die zu sehr guten Vorhersageleistungen der ML-Modelle führen. Allerdings haben die Autoren bei der Evaluation nur die Datensätze 'steel-plates-fault' sowie 'adult' betrachtet, die die diskutierten Ursachen für Bias in Daten nur geringfügig aufweisen. Daher ist in weiteren Evaluationen zu untersuchen, inwieweit mit dem grundlegenden Ansatz von AsisstML diese Datencharakteristika adressiert werden können und ob dazu weitere Meta-Features sowie eine Anpassung des Ansatzes zur Bestimmung der Ähnlichkeit von Datensätzen notwendig sind.

MMP: Die MMP von Weber et al. [WHR20] verknüpft ML-Modelle in den Modell-Metadaten zwar auch mit den Datensätzen, mit denen die ML-Modelle erzeugt wurden. Allerdings unterstützt die MMP ebensowenig die in der rechten Spalte von Tabelle 7.3 aufgeführten Meta-Features, die prinzipiell zur Charakterisierung einer Klassenungleichverteilung, von falsch dokumentierten bzw. verrauschten Klassenlabels oder der Größe der Datensätze geeignet sind.

Nichtsdestoweniger adressiert die MMP eine Gruppenheterogenität und eine Gruppenungleichverteilung, indem die Plattform in ihren Metadaten Domänenwissen einbindet, das zentrale Eigenschaften der domänenspezifischen Gruppen beschreibt. Dies wird über die semantischen Informationen und Verschlagwortungen der Assets im Metadatenmodell ermöglicht. Nach Weber et al. [WHR20] beschreiben diese Assets primär Maschinentypen oder Produktgruppen und damit die beiden zentralen Arten domänenspezifischer Entitätsgruppen in industriellen Anwendungsfällen (siehe Abschnitt 3.2.2). Maschinentypen können in den Assets bspw. mit Begriffen aus OPC UA Informationsmodellen verschlagwortet werden. Zur semantischen Beschreibungen von Produktgruppen eignen sich die branchenübergreifende Taxonomie ECLASS oder unternehmensspezifische Produkttaxonomien [AK04; HRT+23; SRYD14].

Nutzer der MMP können dieses in den Assets eingebunde Domänenwissen nutzen, um in der Plattform nach ML-Modellen zu suchen, die Vorhersagen für bestimmte domänenspezifische Gruppen, z. B. für bestimmte Maschinentypen oder Produktgruppen unterstützen. Die Annahme ist, dass solche für bestimmte domänenspezifische Gruppen geeignete ML-Modelle auch nur mit Daten einer einzelnen oder von wenigen Gruppen erzeugt wurden [WHR20]. Folglich sind in diesen ML-Modellen und deren Daten die Auswirkungen einer Gruppenheterogenität und einer Gruppenungleichverteilung kaum mehr vorhanden [HRT+23]. Die MMP von Weber et al. [WHR20] kann also diese beiden Ursachen für Bias in Daten adressieren. Allerdings setzt dies voraus, dass solche gruppenspezifischen ML-Modelle mit dazu passenden Daten erzeugt und in der MMP erfasst wurden. Hierfür kann bspw. der in Abschnitt 6.5 beschriebene Ansatz von Hirsch et al. [HRT+23] verwendet werden.

Darüber hinaus kann mit den Metadaten der MMP auch der Fall abgebildet werden, dass ML-Modelle aufgrund von Datensatzverschiebungen veraltet sind und an eine sich geänderte statistische Verteilung der Daten angepasst werden

müssen. Dies wird in Statusinformationen innerhalb der Modell-Metadaten abgebildet [Web21; WHRS19]. Hierbei können Statusinformationen u. a. angeben, ob sich ein ML-Modell bspw. in der Trainingsphase befindet oder bereits genutzt wird. Wenn das ML-Modell während der Nutzungsphase z. B. aufgrund von Datensatzverschiebungen immer häufiger falsche Vorhersagen trifft, wird es in der MMP auf den Status 'inaktuell' gesetzt [Web21]. Während das ML-Modell anschließend an neuen Daten angepasst wird, bekommt es den Status 'in Wartung' zugewiesen. Allerdings müssen hierzu auch Methoden zur Anpassung eines ML-Modells an sich geänderte statistische Verteilungen der Daten implementiert werden [LHW18; MRA+12].

7.1.6 Übertragbarkeit von AssistML und der MMP auf andere Anwendungsdomänen

Villanueva et al. [VRWM23] diskutieren Meta-Features, die in AssistML zur Charakterisierung numerischer und kategorischer Features sowie zu Datumsangaben, Textdaten, Zeitreihendaten und Bilddaten verwendet werden können. Damit können in den Metadaten von AssistML grundsätzlich viele der Features beschrieben werden, die auch ein Großteil der in Tabellen 3.1 und 3.3 gezeigten Datenformate aufweisen, die für industrielle oder für medizinische Anwendungsfälle typisch sind [VRWM23]. Lediglich für proprietäre Datenformate, wie z. B. für Geometriedaten aus Anwendungsfällen der Produktentwicklung [ZRKM20; ZRKM21], bietet AssistML keine Meta-Features an, die zur Bestimmung der Ähnlichkeit zwischen Datensätzen verwendet werden können. Für solche proprietären Datenformate werden aber auch sonst in der Literatur keine geeigneten Meta-Features diskutiert [RGS+22]. Darüber hinaus haben Villanueva et al. [VRWM23] ihren Ansatz bisher primär für numerische, kategorische und textuelle Features evaluiert. Folglich muss noch untersucht werden, ob die Evaluationsergebnisse auch für andere Arten von Daten, insbesondere für Zeitreihendaten oder Bilddaten übertragbar sind.

Weiterhin kann AssistML verhältnismäßig einfach mit dem von Kiefer et al. [KRM20] vorgeschlagenen Ansatz kombiniert werden, um somit auch Textanalysen und unstrukturierte Textdaten zu unterstützen. Kiefer et al. [KRM20] nutzen als Grundlage ihres Ansatzes ein Repository, das verschiedene Trainingsdaten für Textanalysen verwaltet. Für einen neuen Datensatz und eine neue

Textanalyse, sucht der Ansatz in diesem Repository nach denjenigen Trainingsdaten, die hinsichtlich Metriken wie bspw. der Kosinus-Ähnlichkeit [Sin01] die größte Ähnlichkeit zum neuen Datensatz aufweisen. Diese Idee kann grundsätzlich auch im ersten Schritt der in Abbildung 7.3 dargestellten Vorgehensweise von AssistML eingebunden werden, um nach ML-Lösungen für Textanalysen zu suchen, die mit geeigneten textuellen Trainingsdaten erzeugt wurden.

Der Großteil des Metadatenmodells der MMP von Weber et al. [WHR20] ist so generisch, dass es zur Beschreibung von ML-Modellen für beliebige Datenformate aus verschiedenen Anwendungsdomänen verwendet werden kann. Die einzige Einschränkung ist das zur Verschlagwortung der domänenspezifischen Assets genutzte Domänenwissen, das in der jeweiligen Anwendungsdomäne bereits vorliegen muss. Hierfür sind allerdings Taxonomien oder sogar einfache Glossare ausreichend. Diese existieren bereits in den meisten industriellen Anwendungsfällen [AK04; HRT+23; SRYD14] sowie auch in vielen Anwendungsfällen zur datengetriebenen medizinischen Diagnose [CRM+20; Fit88; Jab04]. Folglich kann die Idee der MMP zur Verknüpfung von Modell-Metadaten mit Domänenwissen aus Glossaren oder semantischen Netzen auch auf viele dieser Anwendungsdomänen übertragen werden.

7.2 Explizite Nutzung von Ontologien, Wissensgraphen oder Simulationsmodellen im Lernprozess von ML-Modellen

Die Literatur schlägt mehrere Ansätze vor, die formal repräsentiertes Domänenwissen explizit im Lernprozess von ML-Modellen mitverwenden [RMB+23; TZH07]. Dabei kommen als Domänenwissen hauptsächlich Ontologien bzw. Wissensgraphen oder algebraische Gleichungen bzw. Differenzialgleichungen aus Simulationsmodellen zum Einsatz. Die Ansätze können in primär drei Arten eingeteilt werden (siehe Abbildung 7.5): (1) Einbindung des Domänenwissens in Teilen der Struktur eines ML-Modells, z. B. als Layer eines neuronalen Netzes, (2) Nutzung des Domänenwissens innerhalb der Lernalgorithmen und (3) Nutzung des Domänenwissens bei der Bewertung und Auswahl von ML-Modellen [RMB+23]. Nachfolgend werden sie kurz beschrieben, ehe am Ende dieses Abschnitts erörtert wird, inwieweit die Datencharakteristika in industriellen Anwendungsfällen von den Ansätzen adressiert werden.

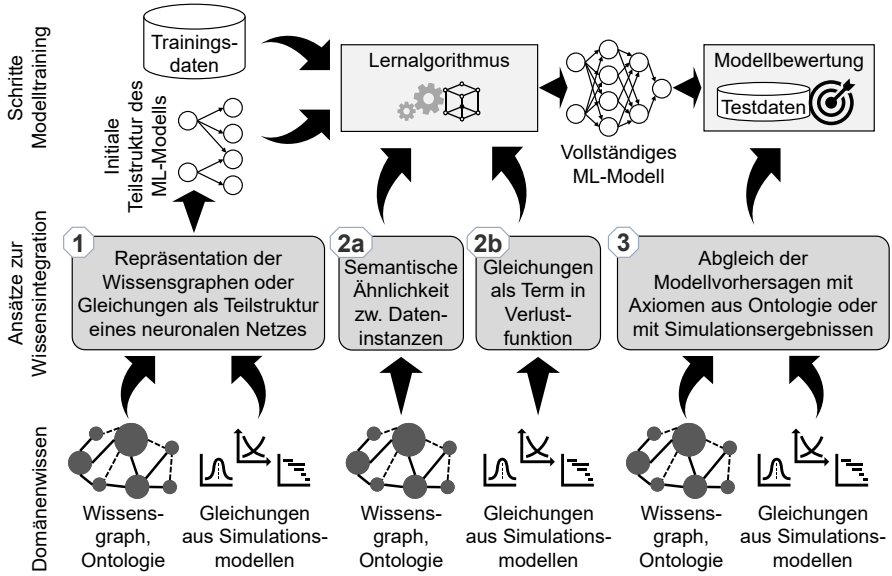


Abbildung 7.5: Schematische Darstellung der wichtigsten Ansätze zur Nutzung von Ontologien, Wissensgraphen oder Gleichungen aus Simulationsmodellen beim Training von ML-Modellen.

7.2.1 Einbindung des Domänenwissens als Teil der Struktur eines ML-Modells

Da semantische Netze wie Ontologien oder Wissensgraphen eine graphbasierte Struktur aufweisen, können mit Ansätzen zu *Graph Neural Networks* [MSG17; SGT+09] aus den Ontologien oder Wissensgraphen direkt neuronale Netze oder Teilstrukturen von neuronalen Netzen erzeugt werden. Dieses Vorgehen wird beispielsweise im Bereich der Objekterkennung in Bilddaten eingesetzt [MSG17]. Hier kann eine Ontologie oder ein Wissensgraph semantische Beziehungen zwischen den zu erkennenden Objekten abbilden. Dabei wird die Ontologie oder der Wissensgraph im Lernverfahren meist als zweite Informations- bzw. Wissensquelle neben anderen Trainingsdaten, bspw. der entsprechenden Bilddaten, genutzt [RMB+23]. Ansätze zu Graph Neural Networks erzeugen zunächst aus der Ontologie oder dem Wissensgraph ein spezielles Layer im neuronalen Netz [LHZ+18]. Dabei werden typischerweise aus den Relationen, Axiomen oder Inferenzregeln, die in der Ontologie

bzw. im Wissensgraph abgebildet sind, Adjazenzmatrizen gebildet, die die Verknüpfungen und Gewichte des entsprechenden Layers im neuronalen Netz vorgeben [MSG17]. Solch ein spezielles Layer repräsentiert das in der Ontologie oder im Wissensgraphen modellierte Domänenwissen direkt im neuronalen Netz. Die Verknüpfungen und Gewichte der restlichen Layer des Netzes können schließlich mittels der anderen Trainingsdaten gelernt werden.

Analog können algebraische Gleichungen oder Differenzialgleichungen aus Simulationsmodellen Einschränkungen für die Strukturen von neuronalen Netzen vorgeben [GNH22; RMB+23]. Die Gleichungen können als initiale lineare oder nichtlineare Verknüpfungen zwischen den Neuronen einzelner Layers eines neuronalen Netzes kodiert werden [LRZL17; PU92]. Ein Simulationsmodell spiegelt physikalische Einschränkungen wider, die in der jeweiligen Anwendungsdomäne gelten müssen. Durch einen solchen Ansatz werden diese physikalischen Einschränkungen direkt im neuronalen Netz abgebildet. Einige verwandte Ansätze bieten auch die Möglichkeit, domänenspezifische Regeln in initialen Verknüpfungen von neuronalen Netzen abzubilden [FZD14; TS94].

7.2.2 Nutzung des Domänenwissens in Lernalgorithmen

Algorithmen für eine Klassifikation wie auch für eine Clusteranalyse gehen allgemein davon aus, dass zwei Dateninstanzen mit höherer Wahrscheinlichkeit zur selben Klasse bzw. zum selben Cluster gehören, je höher die statistische Ähnlichkeit dieser Instanzen bezüglich der Werte ihrer Features ist [HKP12]. Tsymbal et al. [TZH07] schlagen hingegen einen Ansatz vor, der mittels Ontologien die semantische Ähnlichkeit von Dateninstanzen für deren Zuordnung zu Klassen bzw. Clustern verwendet. Zur Bestimmung der semantischen Ähnlichkeit nutzen die Autoren insbesondere Relationen oder Axiome aus einer Ontologie, die semantische Abhängigkeiten zwischen Features beschreiben. Sie nutzen hierbei also ein ähnliches Domänenwissen sowie eine ähnliche Grundidee wie die in Abschnitt 6.1 beschriebenen Ansätze zum Feature-Engineering. Allerdings verwenden sie das Domänmenwissen nicht in einem vor der Datenanalyse durchzuführenden Schritt der Datenvorbereitung, sondern innerhalb der Datenanalyse, genauer innerhalb des Lernalgorithmus.

Andere Ansätze verwenden algebraische Gleichungen oder numerische Lösungen von Differenzialgleichungen aus Simulationsmodellen innerhalb von

Lernalgorithmen [GNH22; RMB+23], um das Optimierungsproblem zur Suche nach den besten Hyperparametern eines Algorithmus einzuschränken. Dabei werden die Gleichungen der Simulationsmodelle über zusätzliche Terme in die Verlustfunktion dieses Optimierungsproblems integriert [DKW+22; HWM+19; KSL+21; MIM+18; MW08]. Diese zusätzlichen Terme quantifizieren die Abweichung der Vorhersagen eines ML-Modells von den Ergebnissen des Simulationsmodells und addieren diese Abweichung mit einer Gewichtung auf die restlichen Terme der Verlustfunktion [DKW+22; DRG17; SE17]. Diese erweiterte Verlustfunktion bestraft beim Modelltraining Inkonsistenzen zwischen den Vorhersagen des ML-Modells und des Simulationsmodells.

7.2.3 Nutzung des Domänenwissens zur Bewertung von ML-Modellen

Neben der Auswertung statistischer Evaluationsmetriken können auch Axiome aus Ontologien oder Wissensgraphen zur finalen Bewertung trainierter ML-Modelle herangezogen werden [RMB+23]. So kann überprüft werden, ob die Vorhersagen, die ein ML-Modell auf Testdaten trifft, die in der Ontologie bzw. im Wissensgraph modellierten Axiome erfüllen oder diesen widersprechen. Damit kann untersucht werden, ob die ML-Modelle domänenspezifische Anforderungen erfüllen, die innerhalb der Axiome abgebildet werden. Dieses Vorgehen wird insbesondere im Bereich der Objekterkennung oder semantischen Segmentierung in Bilddaten eingesetzt [FKL+17; RWH+21]. Analog können die Vorhersagen eines ML-Modells auf Testdaten auch mit algebraischen Gleichungen von Simulationsmodellen evaluiert und hinsichtlich ihrer Konsistenz überprüft werden [GNH22]. Dabei kann bspw. die Abweichung zwischen den Vorhersagen eines ML-Modells und den Simulationsergebnissen quantifiziert und dies als weitere Evaluationsmetrik verwendet werden [DKW+22].

7.2.4 Adressierung der Datencharakteristika industrieller Anwendungsfälle

Die Grundannahme aller in diesem Abschnitt diskutierten Ansätze ist, dass bereits geeignete Ontologien, Wissensgraphen oder Simulationsmodelle vorliegen. Da diese meist zuvor von menschlichen Domänenxperten entwickelt wurden, ist für deren Genauigkeit die Größe der Datenmenge unerheblich [GNH22; RMB+23; SE17]. Werden Ontologien, Wissensgraphen oder Simulationsmo-

delle mittels der in diesem Abschnitt diskutierten Ansätze beim Training von ML-Modellen mitverwendet, reduziert dies folglich insbesondere das Risiko einer Überanpassung von ML-Modellen, wenn zum Modelltraining nur eine kleine Datenmenge zur Verfügung steht (siehe Tabelle 7.1).

Ebenso können einige der Ansätze auch das Problem adressieren, dass in Datensätzen falsch dokumentierte Klassenlabels auftreten [GNH22; KAF+17]. Dies trifft insbesondere auf Ansätze zu, die Gleichungen aus Simulationsmodellen als zusätzlichen Term in der Verlustfunktion eines Lernalgorithmus oder zur Bewertung trainierter ML-Modelle verwenden. Lernalgorithmen, die kein Domänenwissen mitverwenden, lernen für einige falsch dokumentierte Labels dedizierte Muster. Für neue Dateninstanzen, die in diese Muster passen, sagen die ML-Modelle anschließend das falsche Label vorher. Bei Gleichungen aus Simulationsmodellen kann davon ausgegangen werden, dass diese weitgehend korrekt sind und daher kaum falsche Labels vorhersagen [GNH22; KAF+17]. Werden diese Gleichungen in die Verlustfunktion eines Lernalgorithmus integriert [DRG17; SE17], so bestraft diese erweiterte Verlustfunktion den Fall, dass der Lernalgorithmus aufgrund der falsch dokumentierten Labels falsche Muster lernt. Sofern die Gleichungen als weitere Evaluationsmetrik beim Modelltest verwendet werden [DKW+22], können vom ML-Modell falsch vorhergesagte Labels dahingehend überprüft werden, ob sie von den tendenziell richtigen Vorhersagen des Simulationsmodells abweichen.

Allerdings wurde für die Ansätze noch nicht untersucht, inwieweit sie die anderen der in Abschnitt 3.2.2 erläuterten Ursachen für Bias in Daten adressieren. Die Ansätze wurden bisher nur in sehr wenigen industriellen Anwendungsfällen evaluiert, in denen diese Datencharakteristika vorkommen [GNH22; KSL+21; LRZL17]. Grundlegend können sich ähnliche Probleme ergeben, wie sie für semantische Netze in Abschnitt 6.1 und 6.2 sowie für Simulationsmodelle in Abschnitt 6.3 diskutiert werden. In Ontologien oder Wissensgraphen modellieren Domänenexperten primär Relationen und Axiome, die für häufig vorkommende Klassen oder Gruppen gelten, sodass seltene Klassen oder Gruppen nicht adäquat berücksichtigt werden. Ebenso sind viele Simulationsmodelle idealisiert, sodass sie physikalische Zusammenhänge, die ggf. für seltene Klassen oder Gruppen spezifisch sind, kaum erfassen. Folglich adressieren die in diesem Abschnitt diskutierten Ansätze weder eine Klassenungleichverteilung noch eine

Gruppenungleichverteilung bzw. haben tendenziell eher negative Effekte auf diese beiden Datencharakteristika. Ob durch die Ansätze eine Gruppenheterogenität adressiert wird, ist anhand des bisherigen Stands der Forschung noch nicht ersichtlich, sodass die Auswirkung bzgl. dieser Ursache für Bias in Tabelle 7.1 weder positiv noch negativ bewertet werden.

7.3 Inferenz und Vorhersagen mit probabilistischen Graphen

Wie in Abschnitt 2.2 beschrieben können in probabilistischen Graphen, bspw. in Bayesschen Netzen, folgende Komponenten modelliert werden [JN07; KF09]:

- Die Knoten der Graphstruktur, die bestimmte Ereignisse oder Zustände repräsentieren, die in einer Anwendungsdomäne häufig auftreten.
- Gerichtete Kanten, die i. d. R. kausale Abhängigkeiten zwischen den Ereignissen oder Zuständen beschreiben.
- Bedingte Wahrscheinlichkeitsverteilungen oder Wahrscheinlichkeitstabellen an jedem Knoten bzw. an dessen Eingangskanten, die aussagen, mit welcher Wahrscheinlichkeit das dem Knoten zugeordnete Ereignis durch ein anderes mit der Kante verknüpftes Ereignis verursacht wird.

Probabilistische Graphen erlauben zusammen mit einer Inferenzmaschine, bspw. auf Basis der Bayesschen Inferenz [BS00], auch eine Vorhersage von Ereignissen oder Zuständen. In einer Klassifikation entsprechen die Klassen häufig ebenso konkreten Ereignissen, z. B. bestimmte Fehlerfälle in der Produktion [HRM19; WSR+20]. So können probabilistische Graphen als Alternative zu ML-Modellen verwendet werden, um für neue Dateninstanzen die Klassen vorherzusagen. Auf Basis der gegebenen Feature-Werte der neuen Dateninstanz kann festgestellt werden, welche der in einem probabilistischen Graphen modellierten Ereignisse sicher eingetreten sind. Die Wahrscheinlichkeit der Knoten dieser sicheren Ereignisse wird dann auf 100 % gesetzt. Anschließend können rekursiv über alle Kanten des Graphen sowie über die bedingten Wahrscheinlichkeitstabellen die Wahrscheinlichkeiten aller verknüpften Knoten und deren Ereignisse berechnet werden. Entsprechen einzelne Knoten den Klassen eines Anwendungsfalls, kann durch dieses Vorgehen für eine neue Dateninstanz die Wahrscheinlichkeit aller möglichen Klassenlabels bestimmt werden.

Nachfolgend wird zunächst erläutert, wie probabilistische Graphen durch Domänenexperten erzeugt und über datengetriebene Lernverfahren ergänzt werden können. Dem folgt eine Erörterung, inwieweit solch ein Ansatz die Datencharakteristika in industriellen Anwendungsfällen adressieren kann.

7.3.1 Erzeugung probabilistischer Graphen durch Domänenexperten sowie Ergänzung der Graphen durch datengetriebene Lernverfahren

Grundsätzlich können alle Komponenten eines probabilistischen Graphen, d. h. die Knoten, Kanten und bedingten Wahrscheinlichkeitstabellen, manuell durch Domänenexperten spezifiziert werden. Diese Experten können somit ihr Wissen über Ereignisse, die in der Anwendungsdomäne häufig auftreten, sowie über die kausalen Abhängigkeiten zwischen diesen Ereignissen in die Modellierung eines probabilistischen Graphen einfließen lassen. Allerdings ist die manuelle Erstellung eines probabilistischen Graphen in vielen Anwendungsfällen nicht trivial und mit großem Aufwand für Domänenexperten verbunden [WRGM21]. Ein Vorteil von probabilistischen Graphen wie Bayeschen Netzen ist, dass sie alternativ auch anhand von Daten gelernt werden können [Nea03]. Dabei können sowohl die graphbasierte Struktur gelernt werden (engl. *Structure Learning*) als auch die bedingten Wahrscheinlichkeitstabellen (engl. *Parameter Learning*). Es ist auch möglich, dass Domänenexperten zunächst nur einen Teil aller relevanten Knoten und Kanten modellieren. Die noch fehlenden Knoten, Kanten und insbesondere Wahrscheinlichkeitstabellen können anschließend über Daten gelernt und im Graphen ergänzt werden [RD03; YPR+14]. Abbildung 7.6 stellt schematisch die zentralen Schritte dar, mit denen probabilistischen Graphen unter Einbeziehung von Domänenexperten sowie mittels der beschriebenen datengetriebenen Lernverfahren erzeugt werden können.

Sadhasivan [Sad20] beschreibt, wie Bayessche Netze für eine Fehlerdiagnose und Fehlerkorrektur in einer Produktionslinie verwendet werden können. Zentrale Knoten des Netzes beschreiben häufige Fehlerarten. Die Kanten ordnen jeder Fehlerart andere Knoten für mögliche Fehlerursachen und Maßnahmen zur Fehlerbehebung zu. Durch die bedingten Wahrscheinlichkeitstabellen wird festgelegt, welche Ursachen wie wahrscheinlich einen Fehler verursachen und mit welcher Wahrscheinlichkeit bestimmte Fehlerbehebungsmaßnahmen zur Korrektur dieses Fehlers geeignet sind. Sadhasivan [Sad20] beschreibt zudem,

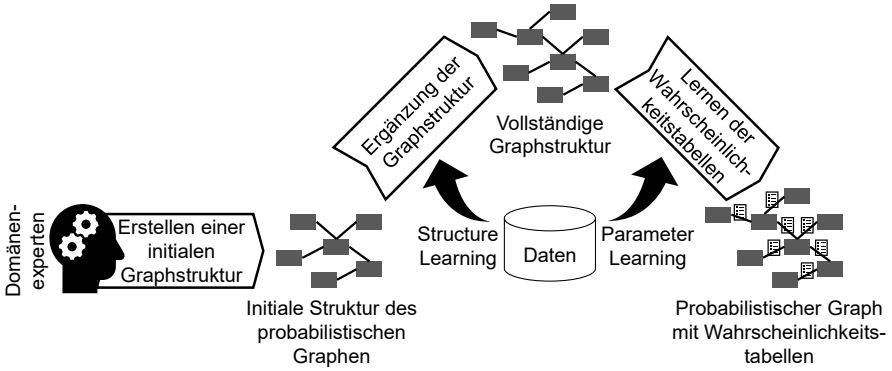


Abbildung 7.6: Zentrale Schritte zur Erzeugung eines probabilistischen Graphen unter Einbeziehung von Domänenexperten sowie durch Kombination mit datengetriebenen Lernverfahren.

wie sowohl die Graphstruktur als auch die bedingten Wahrscheinlichkeitstabellen über Daten aus dem zugrundeliegenden Anwendungsfall erzeugt werden können. Eine der Datenquellen ist eine FMEA-Datenbank [DIN06], die von Ingenieuren als Fallbasis für in der Produktionslinie erwartbare Fehler sowie zugehörige Fehlerursachen und Fehlerbehebungsmaßnahmen entwickelt wurde. Anhand dieser FMEA-Datenbank können zunächst eine initiale Graphstruktur des Bayesschen Netzes sowie initiale Wahrscheinlichkeitstabellen erzeugt werden. In einer weiteren Datenbank werden die in der Produktionslinie tatsächlich aufgetretenen Fehler und ihre Häufigkeiten sowie die durchgeführten Korrekturmaßnahmen dokumentiert. Mit diesen Daten können die Wahrscheinlichkeitstabellen im Bayesschen Netz an die tatsächlich aufgetretenen Fehlersituationen angepasst werden. Ebenso kann die Graphstruktur des Netzes um weitere Fehlerarten, Ursachen und Behebungsmaßnehmen erweitert werden, die noch nicht in der FMEA-Datenbank erfasst wurden.

7.3.2 Adressierung der Datencharakteristika industrieller Anwendungsfälle

Wie oben beschrieben können probabilistische Graphen teilweise durch menschliche Experten erstellt und durch datengetriebene Lernverfahren um noch fehlende Knoten, Kanten und Wahrscheinlichkeitstabellen ergänzt werden. Für die datengetriebenen Lernverfahren liegt folglich bereits eine initiale Graph-

struktur vor, sodass für sie weniger Daten zur Verfügung stehen müssen als dies für herkömmliche maschinelle Lernverfahren ohne Einbeziehung des Expertenwissens der Fall ist. Somit können probablistische Graphen das Risiko einer Überanpassung an eine kleine Datenmenge reduzieren (siehe Tabelle 7.1).

Für ein Klassifikationsproblem können Domänenexperten zudem innerhalb der initialen Graphstruktur die ihnen bekannten kausalen Abhängigkeiten zwischen Features und Klassenlabels spezifizieren. Die Literatur schlägt bereits Beispiele für entsprechende Graphen mit passenden kausalen Relationen aus der Anwendungsdomäne der Produktion vor [BWK+22; HJX16; Ish86; VKFK17] (siehe Abschnitt 6.2). In den meisten Fällen kann davon ausgegangen werden, dass Domänenexperten die Abhängigkeiten zwischen Features und Labels für die betrachteten Klassen korrekt spezifizieren. Folglich kann eine Spezifikation dieser Abhängigkeiten in probabilistischen Graphen auch zum Teil vermeiden, dass die Graphen falsche Klassenlabels vorhersagen.

Allerdings wurde für die anderen Ursachen für Bias in Daten noch nicht untersucht, inwieweit Ansätze zu probabilistischen Graphen diese Datencharakteristika adressieren. Hier können sich ähnliche Probleme ergeben, wie sie in Abschnitt 6.2 für kausale Relationen in semantischen Netzen diskutiert werden. Zum Beispiel müssen Domänenexperten einen gewissen Aufwand betreiben, um alle relevanten kausalen Abhängigkeiten zwischen Features und Klassen in der Graphstruktur festzulegen [WRGM21]. Dabei kann es passieren, dass sie speziell für Minderheitsklassen und Minderheitsgruppen keine kausalen Abhängigkeiten spezifizieren. Prinzipiell können die Graphen durch das anschließende datengetriebene Lernverfahren um weitere kausale Abhängigkeiten ergänzt werden. Allerdings sind Minderheitsklassen und Minderheitsgruppen auch in den Daten unterrepräsentiert, sodass entsprechende Ansätze in den meisten Fällen weder eine Klassenungleichverteilung noch eine Gruppenungleichverteilung adressieren bzw. sogar negative Effekte auf diese beiden Ursachen für Bias in Daten haben können. Ob durch die Kombination der Nutzung des Expertenwissens und des datengetriebenen Lernverfahrens die Probleme bzgl. einer Gruppenheterogenität adressiert werden können, ist bisher noch nicht evaluiert worden. Die Auswirkungen bzgl. dieser Ursache für Bias in Daten werden in Tabelle 7.1 daher weder positiv noch negativ bewertet.

7.4 Entscheidungsfusion in Ensembles aus datengetriebenen ML-Modellen und wissensbasierten Modellen

Ensemble-Methoden fusionieren für eine Dateninstanz die Vorhersagen mehrerer diverser Vorhersagemodelle zu einer kombinierten Vorhersage [Die00; Kun04; WGC14] (siehe Abschnitt 4.3.1). Die einzelnen Vorhersagemodelle sind insbesondere dann divers, wenn sie korrekte Vorhersagen für unterschiedliche Teilmengen der Daten treffen und sich dementsprechend bzgl. dieser korrekten Vorhersagen komplementär ergänzen. In der Vorhersagefusion sollen für eine konkrete Dateninstanz über eine Konsensentscheidung möglichst nur die korrekten Vorhersagen übernommen werden. Die Diversität der einzelnen Vorhersagemodelle sowie die Konsensentscheidung in der Vorhersagefusion führen i. d. R. zu einer höheren Vorhersageleistung des gesamten Ensembles im Vergleich zu einzelnen Vorhersagemodellen [Pol06; VRM23].

Die Diversität der Vorhersagemodelle kann über unterschiedliche Vorgehensweisen erzeugt werden [BWHY05; SIM+21]. So kann jedes Vorhersagemodell bspw. mit einer anderen Teilmenge der Dateninstanzen oder der Features sowie mit unterschiedlichen Lernalgorithmen oder Hyperparametern trainiert werden. Grundsätzlich können auch ML-Modelle, die mit Trainingsdaten gelernt werden, sowie wissensbasierte Modelle, in denen explizit Domänenwissen abgebildet wird, in einem Ensemble kombiniert werden [WRGM21]. Abbildung 7.7 zeigt die wichtigsten Arten von wissensbasierten Modellen, die auf diese Weise mit datengetriebenen ML-Modellen kombiniert werden können. So können als wissensbasierte Modelle insbesondere probabilistische Graphen oder Simulationsmodelle in einem Ensemble verwendet werden. Außerdem können mittels Verfahren der sog. *Distant Supervision* [MBSJ09; YL19] auch semantische Netze oder Wissensgraphen als zusätzliche wissensbasierte Modelle im Ensemble einbezogen werden. Weiterhin sind für Klassifikationsprobleme die in Abschnitt 6.2 beschriebenen Labeling-Regeln geeignet, die Abhängigkeiten zwischen Features und Klassenlabels darstellen [RBE+20; SRYD14]. Solch eine Kombination von rein datengetriebenen ML-Modellen mit einem oder mehreren wissensbasierten Modellen kann prinzipiell zu einer noch höheren Diversität der einzelnen Vorhersagen und damit auch zu einer noch besseren Vorhersageleistung des gesamten Ensembles führen [WRGM21].

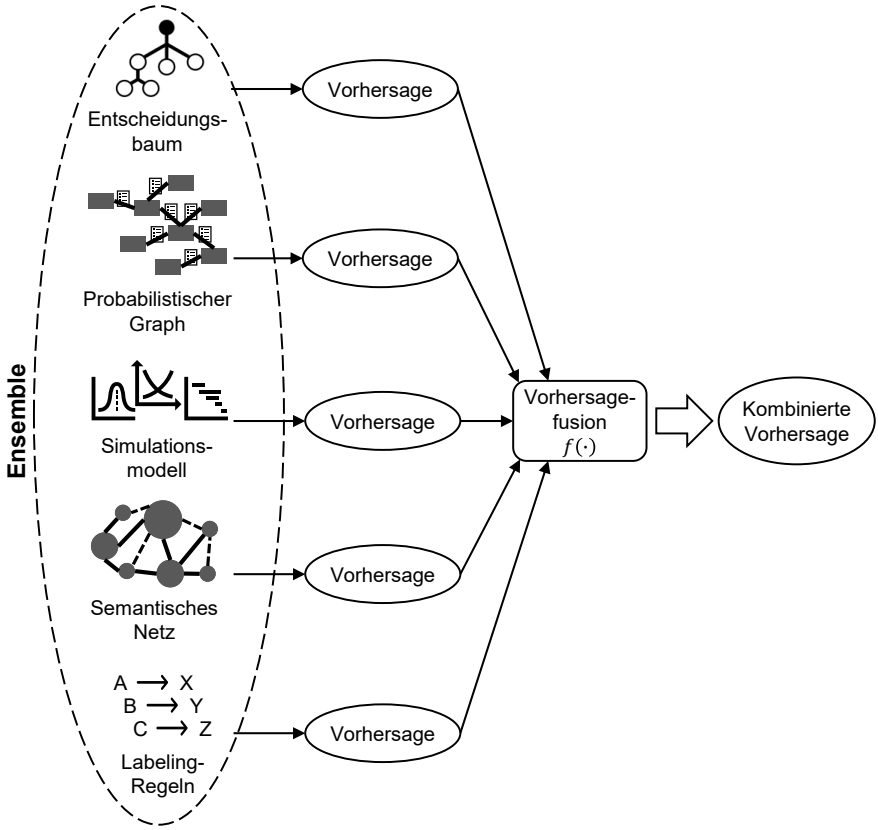


Abbildung 7.7: Darstellung eines beispielhaften Ensembles, das neben Vorhersagen eines datengetrieben erzeugten Entscheidungsbaums die Vorhersagen mehrerer wissensbasierter Modelle kombiniert.

Je diverser die einzelnen Vorhersagemodelle in einem Ensemble sind, desto komplexer wird zumeist auch die Vorhersagefusion. Dies gilt insbesondere, wenn neben datengetriebenen ML-Modellen auch wissensbasierte Modelle Bestandteil des Ensembles sind. Der nachfolgende Abschnitt beschreibt daher ein Rahmenwerk, das in einem vom Autor dieser Habilitationsschrift betreuten Projekt entstanden ist und das die Komplexität für eine Entscheidungsfusion in Klassifikationen beherrschbar macht [WRG+23].

7.4.1 PUSION: Generisches Rahmenwerk zur Optimierung der Entscheidungsfusion in Ensembles

PUSION (Python Universal Fusion) [WRG+23] ist ein generisches Rahmenwerk zur Erstellung und Optimierung von Ensembles für Klassifikationsprobleme. Mit PUSION können unterschiedliche datengetriebene ML-Modelle sowie verschiedene wissensbasierte Modelle in einem Ensemble kombiniert werden. Abbildung 7.8 zeigt die zentralen Systemkomponenten von PUSION. In diesem Abschnitt werden dessen grundlegenden Aspekte beschrieben. Für weitere Details sei auf die Publikation von Wilhelm et al. [WRG+23] verwiesen.

Diverse Klassifikationsmodelle können sich auch bzgl. der Ausgabeform ihrer Vorhersagen unterscheiden. Dabei sind primär folgende Unterscheidungsmerkmale möglich [Kun04; MRT18; WRG+23]:

- Jedes Klassifikationsmodell kann eine binäre Klassifikation zwischen zwei Klassenlabels oder eine Multi-Klassen-Klassifikation bzw. eine Multi-Label-Klassifikation zwischen mehr als zwei Labels umsetzen [MRT18].
- Weiterhin kann ein Klassifikationsmodell bei jeder Vorhersage entweder eine sog. *crisp*-Klassenvorhersage oder eine *soft*-Klassenvorhersage machen [Kun04]. Bei der *crisp*-Vorhersage entscheidet sich das Modell für exakt ein Klassenlabel aus der Menge $\mathcal{Y}$ der möglichen Labels. Unterstützt das Modell eine *soft*-Vorhersage, bestimmt es für jedes Label aus $\mathcal{Y}$ eine Wahrscheinlichkeit bzw. einen Konfidenzwert, wie sicher das Modell ist, dass der jeweiligen Dateninstanz das Label zugeordnet werden soll.
- Schließlich kann jedes Klassifikationsmodell im Ensemble mit Daten einer anderen Teilmenge der Klassenlabels $\mathcal{Y}$ trainiert worden sein, sodass jedes Modell auch nur für die jeweilige Teilmenge der Labels Vorhersagen machen kann. Für zwei Modelle $\mathcal{M}_i$ und $\mathcal{M}_j$ können die entsprechenden Teilmengen $Y_i, Y_j \subseteq \mathcal{Y}$ der vorhersagbaren Labels (1) vollständig redundant sein, d. h. $Y_i = Y_j$, (2) komplementär sein, d. h. $Y_i \cap Y_j = \emptyset$ oder (3) komplementär-redundant sein, d. h. $Y_i \cap Y_j \neq \emptyset \wedge Y_i \neq Y_j$.

Für alle Klassifikationsmodelle in einem Ensemble ergeben sich anhand dieser Unterscheidungsmerkmale mehrere Kombinationsszenarien [WRG+23]. Viele Algorithmen zur Entscheidungsfusion unterstützen aber nur eine einge-

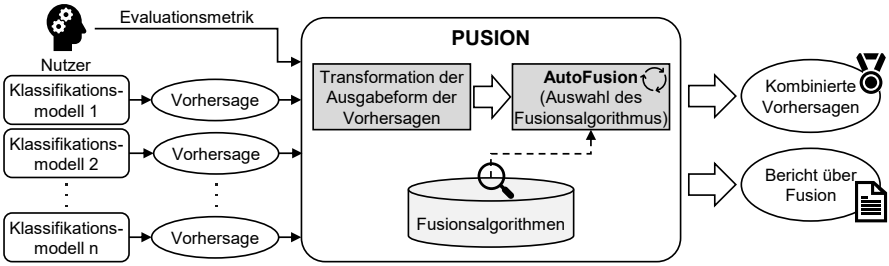


Abbildung 7.8: Zentrale Komponenten des PUSION-Rahmenwerks zur Optimierung der Entscheidungsfusion in Klassifikations-Ensembles.

schränkte Menge an Kombinationsszenarien, z. B. nur für eine Multi-Klassen-Klassifikation mit einer *crisp*-Klassenvorhersage und einer vollständig redundaten Überdeckung der vorhersagbaren Klassen. Daher beinhaltet das PUSION-Rahmenwerk eine Systemkomponente zur Transformation der Ausgabeform der einzelnen Klassenvorhersagen, um die Ausgabeform jeweils an die Anforderungen der Fusionsalgorithmen anzupassen und um somit eine größere Anzahl dieser Algorithmen zu unterstützen. Diese Systemkomponente kann bspw. genutzt werden, um (a) *soft*-Klassenvorhersagen in *crisp*-Vorhersagen, (b) Multi-Label- in Multi-Klassen-Vorhersagen sowie (c) komplementär-redudante in redundante Überdeckungen der vorhersagbaren Klassenlabels umzuwandeln [WRG+23].

Eine weitere Problematik für die Entscheidungsfusion in einem Ensemble ergibt sich daraus, dass eine Vielzahl an Algorithmen zur Entscheidungsfusion existieren, die jeweils ein anderes Vorgehen umsetzen, um aus den einzelnen Klassenvorhersagen eine Konsensentscheidung zu einer kombinierten Vorhersage abzuleiten [GNS11; Kun04; Pol06; Wol92; ZG15]. Eine erste Gruppe von Fusionsalgorithmen kombiniert die Vorhersagen über Heuristiken, z. B. indem sie eine Mehrheitsentscheidung durchführen. Die nächste Gruppe umfasst evidenzbasierte Algorithmen, die bspw. die Konfidenzwerte der einzelnen Klassifikationsmodelle mit in die Entscheidungsfusion einbeziehen. So können z. B. in einer gewichteten Mehrheitsentscheidung die Vorhersagen anhand der Konfidenzen der einzelnen Modelle gewichtet werden. Außerdem existieren trainierbare Algorithmen zur Entscheidungsfusion, die mit den Vorhersagen auf einem Testdatensatz trainiert werden. Für jedes Ensemble sowie für die jeweiligen Daten muss aus diesen zahlreichen Algorithmen zur Entscheidungsfusion

derjenige Algorithmus ausgewählt werden, der bei der Konsensentscheidung primär die korrekten Vorhersagen der einzelnen Klassifikationsmodelle bevorzugt und somit zu einer höheren Vorhersageleistung des gesamten Ensembles führt. Bei den evidenzbasierten und trainierbaren Fusionsalgorithmen müssen zudem noch jeweils eine spezifische Menge an Hyperparametern optimiert werden. Dies führt zu einem weiteren komplexen Optimierungsproblem.

PUSION bietet hierfür mit AutoFusion einen automatisierten Optimierungsansatz, der für ein konkretes Ensemble und einen Datensatz den besten Algorithmus zur Entscheidungsfusion auswählt. Der AutoFusion-Ansatz geht davon aus, dass für den Datensatz bereits mehrere diverse Klassifikationsmodelle trainiert wurden. Weiterhin kann ein Nutzer eine Evaluationsmetrik, z. B. die Korrektklassifikationsrate, auswählen, mit denen die Fusionsalgorithmen bzw. die kombinierten Vorhersagen des Ensembles getestet werden. PUSION unterstützt insgesamt 14 Fusionsalgorithmen aller oben genannten Gruppen dieser Algorithmen. AutoFusion wählt für ein konkretes Ensemble zunächst diejenigen Fusionsalgorithmen aus, die für das jeweils zu unterstützende Kombinationsszenario bzgl. der oben beschriebenen Möglichkeiten für die Ausgabeform der Klassenvorhersagen geeignet sind. Anschließend wendet das Optimierungskonzept jeden der geeigneten Fusionsalgorithmen auf die Vorhersagen der einzelnen Klassifikationsmodelle für einen zusätzlichen Testdatensatz an und evaluiert die kombinierten Vorhersagen hinsichtlich der vom Nutzer spezifizierten Metrik. AutoFusion wählt den Fusionsalgorithmus aus, der hinsichtlich dieser Metrik am besten abgeschnitten hat, und wendet ihn auf die Vorhersagen für den gesamten Datensatz an. Das gesamte Rahmenwerk PUSION liefert als Ergebnis sowohl die mit dem besten Fusionsalgorithmus kombinierten Vorhersagen als auch einen Bericht über die durchgeführte Optimierung und Entscheidungsfusion. Dieser Bericht umfasst alle Informationen, die ein Nutzer zur Bewertung des Gesamtergebnisses benötigt, z. B. alle evaluierten Fusionsalgorithmen sowie das jeweilige Ergebnis mehrerer Evaluationsmetriken.

7.4.2 Zentrale Evaluationsergebnisse für PUSION

Wilhelm et al. [WRG+23] evaluierten das PUSION-Rahmenwerk mit zwei Anwendungsfällen aus dem Bereich der Fehlererkennung und Fehlerdiagnose für Komponenten von Produktionssystemen. Die Anwendungsfälle umfassen je

Tabelle 7.4: Wesentliche Aspekte der Anwendungsfälle für die Evaluation des PUSION-Rahmenwerks [WRG+23].

	Anwendungsfall 1: Hydraulisches System	**Anwendungsfall 2: Servopneumatisches Antriebssystem**
Eigenschaften der Zeitreihendaten	17 Sensoren, 1449 Zyklen, 144 Klassen	7 Sensoren, 21 699 Zyklen, 11 Klassen
Im Ensemble verwendete Modelle	Entscheidungsbaum, Support-Vektor-Maschine, 2 x K-nächster-Nachbar-Ansatz, neuronales Netz	Neuronales Netz, Simulationsmodell, Entscheidungsbaum, Support-Vektor-Maschine, K-nächster-Nachbar-Ansatz
Ausgewählter Fusionsalgorithmus	Evidenzbasierter Naive-Bayes-Algorithmus	Trainierbarer Meta-Klassifikator (neuronales Netz)

einen Datensatz mit Zeitreihendaten von Sensoren der Systemkomponenten. Tabelle 7.4 fasst wesentliche Aspekte der Datensätze sowie die jeweils von Wilhelm et al. [WRG+23] im Ensemble verwendeten Klassifikationsmodelle und den von PUSION ausgewählten Fusionsalgorithmus zusammen.

Im ersten Anwendungsfall nutzen die Autoren einen Datensatz eines hydraulischen Systems, der von Helwig et al. [HPS15] auf dem UCI Machine Learning Repository zur Verfügung gestellt wird[1]. Der Zeitreihendatensatz wurde von 17 Sensoren des hydraulischen Systems erfasst und enthält 1449 Zeitreihenzykeln sowie 144 Klassen. Als einzelne Klassifikationsmodelle, die anschließend im Ensemble kombiniert wurden, trainierten die Autoren fünf datengetriebene ML-Modelle für eine Multi-Klassen-Klassifikation. Sie stellten dabei eine implizite Diversität zwischen diesen ML-Modellen her, indem sie unterschiedliche Lernalgorithmen bzw. Hyperparameter verwendeten. Die einzelnen Klassifikationsmodelle sind ein Entscheidungsbaum, eine Support-Vektor-Maschine, ein K-nächster-Nachbar-Ansatz (KNN-Ansatz) mit $K = 3$, ein KNN-Ansatz mit $K = 5$ sowie ein neuronales Netz.

Im zweiten Anwendungsfall untersuchten Wilhelm et al. [WRG+23] einen Datensatz zu einem servopneumatischen Antriebssystem einer Schweißzange.

[1] Datensatz zur Zustandsüberwachung eines hydraulischen Systems: `https://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems`

Dieser Zeitreihendatensatz wurde von sieben Sensoren erfasst und enthält 21 699 Zeitreihenzyklen und 11 Klassen. Die Autoren nutzen als Klassifikationsmodell zunächst ein neuronales Netz für eine Multi-Label-Klassifikation, dessen Netzarchitektur auf den Arbeiten von He et al. [HZRS16] und Hu et al. [HSS18] basiert. Wilhelm et al. [WRG+23] haben dieses neuronale Netz im Ensemble u. a. mit einem wissensbasierten Modell kombiniert, d. h. mit einem mathematischen Simulationsmodell, das den zeitabhängigen Systemzustand des servopneumatischen Antriebssystems simuliert. Mit diesem Simulationsmodell berechnen sie Residuen zwischen den simulierten und den tatsächlichen Systemzuständen und nutzen diese Residuen als Teil der Eingabe-Features für drei einfache Klassifikationsmodelle [GNH22]. Diese Klassifikationsmodelle sind ein Entscheidungsbaum, eine Support-Vektor-Maschine und ein K-nächster-Nachbar-Ansatz. Wilhelm et al. [WRG+23] führen mit PUSION eine Entscheidungsfusion zwischem dem neuronalen Netz und den drei mit den simulationsbasierten Residuen erzeugten Klassifikationsmodellen durch.

Zur Evaluation von PUSION berechneten Wilhelm et al. [WRG+23] mehrere Evaluationsmetriken sowohl für die einzelnen Klassifikationsmodelle als auch für die mit PUSION und mit unterschiedlichen Algorithmen zur Entscheidungsfusion erzeugten Ensembles. Im Folgenden werden die zentralen Ergebnisse hinsichtlich der Korrektklassifikationsrate der Modelle sowie der Rechenlaufzeit von PUSION zusammengefasst [WRG+23]:

- Die mit PUSION erzeugten Ensembles erzielen eine höhere Vorhersageleistung als jedes der einzelnen Klassifikationsmodelle, die im Ensemble kombiniert werden. Im ersten Anwendungsfall ist das beste einzelne Klassifikationsmodell und damit die Baseline für PUSION der erste KNN-Ansatz mit $K = 3$, der eine Korrektklassifikationsrate von 87,7 % erreicht. PUSION wählt als besten Fusionsalgorithmus einen evidenzbasierten Naive-Bayes-Algorithmus aus. Dieser erhöht im Vergleich zur Baseline, d. h. zum einzelnen KNN-Ansatz, die Korrektklassifikationsrate auf 92,6 %. Im zweiten Anwendungsfall ist das beste einzelne Klassifikationsmodell das neuronale Netz, das eine Korrektklassifikationsrate von 91,3 % erreicht. PUSION wählt als Fusionsalgorithmus mit einem weiteren neuronalen Netz einen trainierbaren Meta-Klassifikator aus. Dieser erhöht die Korrektklassifikationsrate auf 96,8 %.

- Weiterhin zeigen die Ergebnisse von Wilhelm et al. [WRG+23], dass die Wahl des Fusionsalgorithmus einen erheblichen Einfluss auf die Vorhersageleistung des Ensembles hat. Wird bspw. der trainierbare Meta-Klassifikator, der im zweiten Anwendungsfall zum besten Ergebnis führt, auch im ersten Anwendungsfall als Fusionsalgorithmus verwendet, so reduziert dieser sogar die Korrektklassifikationsrate im Vergleich zur besten Baseline um 1,5 %-Punkte. Der AutoFusion-Ansatz von PUSION evaluiert alle Fusionsalgorithmen, die für den jeweiligen Datensatz und Anwendungsfall anwendbar sind. Der Ansatz garantiert somit, dass jeweils der beste Fusionsalgorithmus ausgewählt und im Ensemble verwendet wird.
- PUSION und der AutoFusion-Ansatz haben eine geringe Rechenlaufzeit, die im Vergleich zur Steigerung der Vorhersageleistung des Ensembles vernachlässigbar ist. Selbst wenn AutoFusion sämtliche Fusionsalgorithmen sequentiell nacheinander evaluiert, benötigt PUSION für einen Anwendungsfall maximal 42 Sekunden für alle involvierten Schritte, d. h. für die Transformation der Ausgabeform der Vorhersagen, für die Auswahl des besten Fusionsalgorithmus mit dem AutoFusion-Ansatz und für die Kombination der einzelnen Vorhersagen im Ensemble [WRG+23].

7.4.3 Adressierung der Datencharakteristika industrieller Anwendungsfälle

Wie in Abschnitt 4.3.1 beschrieben wird, sind Ensemble-Methoden insbesondere geeignet, das Risiko einer Überanpassung der Vorhersagemodelle an eine zu kleine Datenmenge zu reduzieren [Die00; Kun04; WGC14]. Dies trifft in besonderem Maße auch auf die mit dem PUSION-Rahmenwerk erzeugten Ensembles zu [VRM23; WRG+23]. PUSION bietet mit dem AutoFusion-Ansatz sogar eine Auswahl desjenigen Fusionsalgorithmus, der für einen konkreten Datensatz zu der höchsten Vorhersageleistung und damit zu einer besseren Generalisierbarkeit des Ensembles führt. Analog zu Bagging-Verfahren [Bre96], sind die mit PUSION erzeugten Ensembles durch die optimierte Entscheidungsfusion i. d. R. auch weniger anfällig für verrauschte Dateninstanzen mit falschen Labels.

Werden zudem wissensbasierte Modelle ins Ensemble integriert, so ist der positive Effekt auf diese beiden Ursachen für Bias in Daten häufig noch größer. Dies liegt an ähnlichen Gründen, wie sie in den Abschnitten 7.2.4 und 7.3.2

erläutert werden. So werden die in Ensembles verwendbaren probabilistischen Graphen, Simulationsmodelle, semantische Netze bzw. Wissensgraphen oder Labeling-Regeln von menschlichen Domänenexperten erzeugt, sodass für deren Genauigkeit die Größe der Datensätze keine Rolle spielt [GNH22; RMB+23; SE17]. Ebenso kann prinzipiell davon ausgegangen werden, dass Domänenexperten Abhängigkeiten zwischen Features und Klassenlabels in den Wissensmodellen weitgehend korrekt spezifizieren. Folglich kann eine Einbeziehung der wissensbasierten Modelle in die Konsensentscheidung des Ensembles zum Teil vermeiden, dass das Ensemble falsche Klassenlabels vorhersagt.

Allerdings haben Wilhelm et al. [WRG+23] zur Evaluation des PUSION-Rahmenwerks bisher nur ein einzelnes Simulationsmodell in den Ensembles verwendet, mit dem zudem nur Residuen erzeugt werden, die als Features für einfache Klassifikationsmodelle dienen. Die Auswirkungen einer optimierten Entscheidungsfusion auf die verschiedenen Ursachen für Bias in Daten müssen daher noch für sämtliche der in Abbildung 7.7 dargestellten wissensbasierten Modelle sowie für deren direkte Integration im Ensemble überprüft werden.

Auf die weiteren in Abschnitt 3.2.2 beschriebenen Ursachen für Bias in Daten haben Ensemble-Methoden, bei denen die Entscheidungsfusion mit PUSION optimiert wird, nur geringfügige Auswirkungen, sodass diese in Tabelle 7.1 weder positiv noch negativ bewertet werden. Speziell wenn neben datengetriebenen ML-Modellen auch wissensbasierte Modelle in einem Ensemble miteinbezogen werden, sind diese Auswirkungen noch nicht ausreichend untersucht worden. Grundsätzlich können die einzelnen Klassifikationsmodelle, die mit PUSION im Ensemble kombiniert werden sollen, auch jeweils auf Daten unterschiedlicher Teilmengen der Klassenlabels oder domänenspezifischer Gruppen angepasst sein. So kann ein Teil der einzelnen Klassifikationsmodelle auch auf Minderheitsklassen oder Minderheitsgruppen spezialisiert sein, sodass im gesamten Ensemble tendenziell die Vorhersageleistung für diese seltenen Klassen oder Gruppen erhöht wird. Allerdings gehen Wilhelm et al. [WRG+23] davon aus, dass die einzelnen Klassifikationsmodelle bereits vorliegen. Folglich gibt es mit PUSION keine explizite Möglichkeit, um die Menge der Klassifikationsmodelle im Ensemble gezielt zu erzeugen bzw. auszuwählen, sodass diese Menge der Modelle ausgewogen hinsichtlich der Vorhersageleistung für unterschiedlich häufige Klassen oder domänenspezifische Gruppen ist.

Zur Erzeugung einer solch ausgewogenen Menge an Klassifikationsmodellen ist es denkbar, zunächst das in Abschnitt 6.5 beschriebene Verfahren zur Nutzung von Domänenwissen für die Partitionierung von Trainingsdaten [HRT+23] anzuwenden. Anschließend können mit den Teilmengen der Daten klassen- oder gruppenspezifische Klassifikationsmodelle trainiert und, ggf. zusammen mit weiteren wissensbasierten Modellen, mit dem PUSION-Rahmenwerk in einem Ensemble kombiniert werden. Diese Möglichkeit muss aber in Zukunft noch genauer erforscht und evaluiert werden [Vog23].

7.4.4 Übertragbarkeit von PUSION auf andere Anwendungsdomänen

Auch wenn Wilhelm et al. [WRG+23] das PUSION-Rahmenwerk nur mit Anwendungsfällen aus dem Bereich der Fehlererkennung und Fehlerdiagnose in Produktionssystemen sowie mit Zeitreihendaten evaluiert haben, ist PUSION grundsätzlich für beliebige Klassifikationsprobleme aus unterschiedlichen Anwendungsdomänen und für verschiedene Datensätze anwendbar. Dies liegt u. a. daran, dass PUSION die Entscheidungsfusion für eine beliebige Menge an Klassifikationsmodellen optimieren und durchführen kann. Unterscheiden sich diese einzelnen Klassifikationsmodelle, wie in Abschnitt 7.4.1 beschrieben, in der Ausgabeform ihrer Vorhersagen, so kann PUSION die Ausgabeformen aneinander angleichen, sodass viele diverse Klassifikationsmodelle in einem Ensemble kombiniert werden können. Der AutoFusion-Ansatz von PUSION wählt anschließend für jeden Anwendungsfall und Datensatz denjenigen Fusionsalgorithmus aus, der gemäß der vom Nutzer vorgegebenen Evaluationsmetrik zur besten Vorhersageleistung des Ensembles führt. Inwieweit diese Optimierung des Ensembles auch in anderen Anwendungsfällen und für andere Datensätze zu einer Verbesserung der Vorhersageleistung führt, muss noch durch weitere Evaluationen untersucht werden [Vog23]

Darüber hinaus unterstützt der AutoFusion-Ansatz von PUSION bisher noch keine Optimierung der Hyperparameter für evidenzbasierte und trainierbare Fusionsalgorithmen. In ihrer Evaluation haben Wilhelm et al. [WRG+23] empirisch die Hyperparameter bestimmt, die für die von ihnen betrachteten zwei Anwendungsfälle zum besten Ergebnis geführt haben. Diese empirisch bestimmten Hyperparameter haben die Autoren als Default in der Implementierung des AutoFusion-Ansatzes verwendet. Da solch ein Aufwand zur empirischen Suche

nach den besten Hyperparametern in neuen Anwendungsfällen unangemessen hoch ist, sollten PUSION und der AutoFusion-Ansatz in Zukunft noch um eine automatische Suche nach den besten Hyperparametern für evidenzbasierte und trainierbare Fusionsalgorithmen ergänzt werden [VRM23].

7.5 Zusammenfassung

Im Schritt der Datenanalyse und Modellbildung von Datenanalyseprozessen finden sich unterschiedliche Ansätze, die formal repräsentiertes Domänenwissen nutzen. Zum Beispiel nutzen Modellverwaltungsplattformen ein Repository als Fallbasis von bereits erstellten ML-Modellen und bieten damit ein Empfehlungssystem, das für neue Datensätze passende ML-Modelle vorschlägt [VRBT13; VSL+16; ZCD+18]. Die Plattformen beschreiben die ML-Modelle sowie die zugehörigen Daten mittels Metadaten. Sie adressieren die in Abschnitt 3.2 beschriebenen Datencharakteristika industrieller Anwendungsfälle insbesondere dann, wenn diese Datencharakteristika mittels der jeweils unterstützten Metadaten ausreichend genau spezifiziert werden können. AssistML von Villanueva et al. [VRWM23; VWRM21] bietet bspw. ein statistisches Meta-Feature, das den relativen Anteil fehlender Feature-Werte als Indikator zur Charakterisierung einer Gruppenheterogenität und Gruppenungleichverteilung in Daten nutzt. Die *Model Management Platform* (MMP) von Weber et al. [WHR20; WR20] beschreibt diese beiden Datencharakteristika explizit über Domänenwissen, indem es eine Verschlagwortung der ML-Modelle und Datensätze mit Begriffen aus Glossaren oder semantischen Netzen erlaubt, die domänenspezifische Gruppen, wie bspw. Produktgruppen oder Maschinentypen, beschreiben. Zur Charakterisierung einer Klassenungleichverteilung, von falsch dokumentierten Klassenlabels sowie des Risikos einer Überanpassung der ML-Modelle an eine kleine Datenmenge können in Modellverwaltungsplattformen statistische Meta-Features verwendet werden, die die Anzahlen und Verteilungen der Klassenlabels, Dateninstanzen oder Features quantifizieren.

Weiterhin existieren einige Ansätze, die formal repräsentiertes Domänenwissen explizit im Lernprozess von ML-Modellen mitverwenden [GNH22; RMB+23]. Dabei werden insbesondere Ontologien bzw. Wissensgraphen oder Gleichungen aus Simulationsmodellen genutzt, um bspw. Teile der Struktur

eines ML-Modells vorzugeben, um das Optimierungsproblems von Lernalgorithmen einzuschränken oder die trainierten ML-Modelle mithilfe des Domänenwissens zu bewerten. Andere Ansätze verwenden probabilistische Graphen, bei denen Domänenexperten eine initiale Graphstruktur sowie ggf. initiale Wahrscheinlichkeitstabellen vorgeben können [KF09]. Anschließend können die Graphen und die Wahrscheinlichkeitstabellen über datengetriebene Lernverfahren ergänzt und an konkrete Daten angepasst werden [Nea03]. All diese Ansätze helfen vor allem, das Risiko einer Überanpassung der trainierten ML-Modelle an eine kleine Datenmenge zu reduzieren bzw. falsch dokumentierte Labels zu identifizieren und zu korrigieren [GNH22; KAF+17; Nea03; RMB+23]. Allerdings können die Ansätze tendenziell negative Effekte auf eine Klassenungleichverteilung und Gruppenungleichverteilung haben, da das genutzte Domänenwissen meist nur umfassend genug für Mehrheitsklassen und Mehrheitsgruppen spezifiziert wurde.

PUSION bietet ein generisches Rahmenwerk, das die Entscheidungsfusion in Ensembles für Klassifikationsprobleme optimiert [WRG+23]. PUSION kann dabei prinzipiell mehrere datengetriebene ML-Modelle und unterschiedliche wissensbasierte Modelle, wie bspw. probabilistische Graphen, Simulationsmodelle, semantische Netze oder Labeling-Regeln, in einem Ensemble kombinieren. Für einen konkreten Datensatz und für eine konkrete Menge an einzelnen Vorhersagemodellen wählt PUSION den besten Algorithmus zur Entscheidungsfusion aus, der die Diversität der einzelnen Modelle möglichst optimal ausnutzt, um somit die Vorhersageleistung des gesamten Ensembles zu maximieren. Durch diese Optimierung der Entscheidungsfusion kann PUSION insbesondere das Risiko einer Überanpassung des gesamten Ensembles an eine zu kleine Datenmenge oder an einzelne falsche Klassenlabels im Vergleich zu herkömmlichen Ensemble-Methoden sogar noch weiter reduzieren.

Kapitel 8

Zusammenfassung und Ausblick

In diesem Kapitel werden die zentralen Ergebnisse der Diskussion zu Datencharakteristika in industriellen Anwendungsfällen für Datenanalysen sowie zu Ansätzen, die formal repräsentiertes Domänenwissen zur Adressierung dieser Datencharakteristika nutzen, zusammengefasst. Zudem werden Themen identifiziert, die in diesem Kontext weiter vertieft werden können.

8.1 Zentrale Ergebnisse der Diskussion

Den Untersuchungsgegenstand der vorliegenden Habilitationsschrift bilden industrielle Anwendungsfälle für Datenanalysen sowie die in dieser Anwendungsdomäne vorherrschenden und meist komplexen Datencharakteristika, die zu mäßigen oder nicht aussagekräftigen Analyseergebnissen führen können. Kapitel 3 fasst auf Basis einer Analyse mehrerer industrieller Anwendungsfälle die häufigsten solcher komplexen Datencharakteristika zusammen. Zum Beispiel liegen Daten oftmals in *heterogenen Formaten* und in *isolierten Datensilos* vor [Grö18], wodurch insbesondere die Datenakquise aus den Quellsystemen erschwert wird [ZRKM20]. Weitere komplexe Datencharakteristika entsprechen *Ursachen für verschiedene Ausprägungen von Bias in Daten* [MMS+21; SG21]. Zentrale Beispiele sind eine zu kleine Menge an verfügbaren Daten, eine Klassenungleichverteilung sowie eine Heterogenität und eine Ungleichver-

teilung der in den Daten repräsentierten domänenspezifischen Entitätsgruppen, wie bspw. Produktgruppen [HRM19; KM16; SRYD14]. Darüber hinaus ergeben sich komplexe Datencharakteristika daraus, dass die Produktionsumgebungen sowie die durch Kunden verwendeten Produkte *nichtstationäre Umgebungen* bilden [HB11; SRWM22]. Hierbei können sich durch Datensatzverschiebungen die Muster in Daten und deren statistische Verteilung über die Zeit ändern [MRA+12; SRWM22]. Weiterhin werden viele Prozesse in der Produktion oder der Produktentwicklung unstrukturiert durchgeführt, was ein Problem für gängige Verfahren zur Prozessanalyse darstellt [DRG21; GA07; ZRS+22].

Kapitel 4 untersucht für mehrere rein datengetriebene Ansätze zur Datenakquise, Datenvorbereitung und Datenanalyse, welche der komplexen Datencharakteristika industrieller Anwendungsfälle diese Ansätze adressieren können und auf welche sie negative Effekte haben. Unter rein datengetriebenen Ansätzen werden dabei solche verstanden, die kein oder nur geringfügig Domänenwissen einbeziehen, z. B. Sampling-Verfahren, eine statistische Feature-Selektion oder AutoML-Systeme. Zentrales Ergebnis dieser Untersuchung ist, dass jeder untersuchte Ansatz zwar einzelne der komplexen Datencharakteristika adressieren kann, aber auch negative Effekte auf einzelne der nicht berücksichtigten Datencharakteristika hat. Zum Beispiel gibt es hinsichtlich der Ursachen für Bias in Daten viele datengetriebene Ansätze, die eine Klassenungleichverteilung adressieren. Allerdings verstärken diese Ansätze die Probleme, die sich durch die Heterogenität und Ungleichverteilung der domänenspezifischen Entitätsgruppen ergeben. Einige vergleichende Studien, die in vom Autor dieser Schrift betreuten Projekten durchgeführt wurden, weisen nach, dass rein datengetriebene Ansätze aus diesen und ähnlichen Gründen die Qualität der Analyseergebnisse häufig sogar reduzieren [HRM19; SRWM22].

Einzelne Arbeiten zeigen, dass die komplexen Datencharakteristika industrieller Anwendungsfälle durch die Einbeziehung von Domänenwissen effektiver adressiert werden können (siehe bspw. [HRT+23; RMB+23; WRGM21]). Die vorliegende Habilitationsschrift schließt eine wesentliche Lücke im Stand der Wissenschaft, indem Kapitel 5 bis 7 die Ergebnisse einer systematischen und umfassenden Untersuchung verschiedener Ansätze zur Einbeziehung unterschiedlicher Arten formal repräsentierten Domänenwissens in bestimmte Aufgaben von Datenanalyseprozessen zusammenfassen. Dabei steht erneut

im Vordergrund, auf welche der komplexen Datencharakteristika die untersuchten Ansätze positive oder negative Effekte haben. Als formale Wissensrepräsentationsformen werden primär Fallbasen [SS09a], Regelbasen [Ros03], Glossare [ANSI/NISO10], semantische Netze wie Taxonomien, Ontologien oder Wissensgraphen [EW16; Sow91], probabilistische Graphen [KF09] sowie mathematische Simulationsmodelle [BZBP13] betrachtet. Die Ansätze zur Einbeziehung dieses Domänenwissens werden hinsichtlich der zentralen Schritte Datenakquise, Datenvorbereitung und Datenanalyse eines generischen Vorgehensmodells für Datenanalyseprozesse gruppiert.

Im Schritt der Datenakquise (Kapitel 5) wird Domänenwissen insbesondere in Metadatenmodellen eingebunden. Dabei werden die Metadaten mit Begriffen aus Glossaren oder semantischen Netzen verschlagwortet, um auf diese Weise Daten heterogener Formate zu abstrahieren [EGG+21a; GLN+05]. Im Rahmen eines vom Autor dieser Schrift betreuten Projekts haben Ziegler et al. [ZRKM21] ein entsprechendes Metadatenmodell für die virtuellen Produktentwicklung entworfen. Zusätzlich zur Verschlagwortung der Metadaten mit Begriffen aus Glossaren oder semantischen Netzen bildet dieses Metadatenmodell explizit den Datenfluss von Produktentwicklungsprojekten ab. Da diese Projekte die tägliche Arbeit der Produktentwicklungsingenieure widerspiegelt, erleichtert das Metadatenmodell ihnen die Akquise von Daten aus heterogenen und zuvor isolierten Datenquellen. In einem weiteren Projekt ist ein Datengenerator entstanden, der bei der Datengenerierung Domänenwissen in Form einer Taxonomie einbezieht [TRSM23]. Er kann daher Datensätze erzeugen, die bzgl. der Ursachen für Bias in Daten neben einer Klassenungleichverteilung auch eine Gruppenheterogenität und Gruppenungleichverteilung aufweisen.

Zur Unterstützung der Datenvorbereitung (Kapitel 6) schlägt die Literatur viele Ansätze vor, die eine Regelbasis, semantische Netze oder Simulationsmodelle zum Feature-Engineering, zur Festlegung der Labels in Daten oder zur Anreicherung von Daten verwenden [GNH22; ITW05; RBE+20; SRYD14]. Diese Ansätze eignen sich, um das Risiko einer Überanpassung der trainierten ML-Modelle an eine kleine Datenmenge zu reduzieren oder um falsch dokumentierte Labels zu identifizieren und zu korrigieren. Allerdings haben die Ansätze negative Effekte auf eine Klassenungleichverteilung, Gruppenheterogenität und Gruppenungleichverteilung in Daten. Aus diesen Grund wurde

in einem weiteren vom Autor betreuten Projekt ein Verfahren entwickelt, das auch diese Ursachen für Bias in Daten adressiert [HRM20a; HRT+23]. Dieses Verfahren nutzt Domänenwissen zur Partitionierung eines Datensatzes in einzelne Teilmengen. Im Gegensatz zu verwandten Ansätzen zur Constraint-basierten Clusteranalyse [LJJ07; WCRS01] nutzt es aber nicht Constraints, die mit hohem Aufwand zwischen einzelnen Dateninstanzen spezifiziert werden müssen. Stattdessen basiert das neue Verfahren auf Taxonomien, die die domänenspezifischen Entitätsgruppen hierarchisch organisieren. Die mit diesem Verfahren erzeugten Datenteilmengen enthalten nur noch Daten weniger domänenspezifischer Gruppen. Ebenso sind diese Gruppen sowie die Klassen innerhalb der Teilmengen gleichmäßiger verteilt. So steigert das Verfahren die Vorhersageleistung der trainierten Klassifikationsmodelle erheblich [HRT+23].

Im Kontext der Datenanalyse (Kapitel 7) bieten Modellverwaltungsplattformen Möglichkeiten, um bereits trainierte ML-Modelle mit Metadaten zu beschreiben und für neue Datensätze ML-Modelle vorzuschlagen [VSL+16; ZCD+18]. In vom Autor dieser Schrift betreuten Projekten sind die Plattform AssistML [VRWM23; VWRM21] sowie die *Model Management Platform* (MMP) [WHR20; WR20] entstanden. Insbesondere mit der MMP können die Metadaten für ML-Modelle mit Begriffen aus Glossaren oder semantischen Netzen verschlagwortet werden, die bspw. bestimmte Produktgruppen oder Maschinentypen beschreiben. Damit adressiert die MMP eine Gruppenheterogenität und Gruppengleichverteilung, indem sie die Suche nach ML-Modellen erlaubt, die für die Daten bestimmter domänenspezifischer Gruppen geeignet sind. Weitere in Kapitel 7 diskutierte Ansätze reduzieren insbesondere das Risiko einer Überanpassung der trainierten ML-Modelle an eine kleine Datenmenge oder an falsch dokumentierte Labels. Diese Ansätze nutzen bspw. Ontologien bzw. Wissensgraphen oder Simulationsmodelle explizit im Lernprozess von ML-Modellen [GNH22; RMB+23] oder führen eine Inferenz mit teilweise durch Domänenexperten vordefinierten probabilistischen Graphen durch [Nea03]. Schließlich wird mit PUSION ein Rahmenwerk diskutiert, das die Entscheidungsfusion in Ensembles für Klassifikationen optimiert [WRG+23]. PUSION kann prinzipiell mehrere datengetriebene ML-Modelle und wissensbasierte Modelle, wie bspw. probabilistische Graphen, Simulationsmodelle, semantische Netze oder Labeling-Regeln, in einem Ensemble kombinieren.

8.2 Ausblick

Ein bisher wenig erforschtes Themengebiet ist die Nutzung formal repräsentierten Domänenwissens zur Adressierung von Datensatzverschiebungen. Im Rahmen dieser Arbeit wurde primär nur der Ansatz von Chen et al. [CLPC17] identifiziert. Dieser verwendet Ontologien, um aus Datenströmen Features zu extrahieren, die bei der Erkennung von Datensatzverschiebungen helfen können. Weiterhin kann die Modellverwaltungsplattform von Weber et al. [WHR20] in ihren Metadaten abbilden, dass ML-Modelle aufgrund von Datensatzverschiebungen veraltet sind und an eine sich geänderte statistische Datenverteilung angepasst werden müssen [WHRS19]. Allerdings können diese Ansätze jeweils nur eine einzelne Art von Datensatzverschiebung adressieren und sind daher nicht in der Lage, sowohl eine Konzept- als auch eine Kovariatenverschiebung sowie eine mögliche Trade-off-Beziehung zwischen beiden Datensatzverschiebungen zu behandeln. Folglich sollte in Zukunft untersucht werden, wie formal repräsentiertes Domänenwissen genutzt werden kann, um die Auswirkungen mehrerer, gleichzeitig auftretender Datensatzverschiebungen sowie Trade-off-Beziehungen zwischen diesen auszugleichen. Entsprechende neu zu entwickelnde Ansätze können zudem im Forschungsumfeld des sog. *Machine Learning Operations* (MLOps) eingebettet werden [KKH23].

Außerdem sind gerade in industriellen Anwendungsfällen die Ursachen für konkrete Datensatzverschiebungen nicht immer ausschließlich auf die Daten zurückzuführen. So können z. B. in einer Produktionslinie auch andere Ursachen wie z. B. Sensorverschleiß, eine kurzfristige Verdeckung von Sensoren, Fehlparametrisierungen von Maschinen oder plötzlich auftretende Hitze in der Umgebung zu abrupten oder graduellen Datensatzverschiebungen führen [WHRS19]. Solche Ursachen können andere Lösungsansätze erfordern als die Anpassung eines ML-Modells an die statistische Verteilung neuer Daten. Beispielsweise sollte bei einem Sensorverschleiß eher der Sensor ausgetauscht werden. Da viele solcher Ursachen domänenspezifischer Natur sind, kann in Zukunft erforscht werden, wie formal repräsentiertes Domänenwissen zur Abgrenzung der Ursachen von Datensatzverschiebungen genutzt werden kann [Vij21]. Ebenso können mittels des Domänenwissens gezielt Lösungsansätze für jede Ursache von Datensatzverschiebungen abgeleitet werden.

Analog gibt es in der Literatur kaum Ansätze zur Prozessanalyse, die mit unstrukturierten und nichtlinear durchgeführten Prozessen umgehen können, wie sie in nichtstationären Produkt- und Produktionsumgebungen vorkommen [DRG21; ZRS+22]. Zum Beispiel sind Verfahren des Process Mining nicht in der Lage, verschiedene Ausführungsvarianten eines Prozesses zu analysieren [GA07]. In diesem Zusammenhang kann in Zukunft untersucht werden, ob bereits existierende oder auch neu zu entwickelnde Ansätze zur Nutzung formal repräsentierten Domänenwissens dabei helfen können, in einer Prozessanalyse besser mit unstrukturierten und nichtlinear durchgeführten Prozessen sowie mit einzelnen Prozessvarianten umzugehen.

Literaturverzeichnis

[AA19] E. Almalki, M. Abdullah. „Random Tree Data Stream Classifier with Sliding Window Estimator and Concept Drift“. In: *Bioscience Biotechnology Research Communications* 12.1 (2019) (Zitiert auf S. 106).

[Aal+11] W. M. van der Aalst et al. „Process Mining Manifesto“. In: *Workshopband der 9. International Conference on Business Process Management (BPM)*. Clermont-Ferrand, France: Springer, 2011, S. 169–194 (Zitiert auf S. 55).

[Aal11] W. M. P. van der Aalst. „Process Mining: Discovering and Improving Spaghetti and Lasagna Processes“. In: *Tagungsband des 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. Paris, France, 2011 (Zitiert auf S. 74, 124).

[ABLM14] M. Arenas, P. Barcel, L. Libkin, F. Murlak. *Foundations of Data Exchange*. New York, NY, USA: Cambridge University Press, 2014 (Zitiert auf S. 97).

[AFL+11] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García. „KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework“. In: *Multiple-Valued Logic and Soft Computing* 17.2-3 (2011), S. 255–287 (Zitiert auf S. 86).

[AK04] B. Agard, A. Kusiak. „Data-Mining-based Methodology for the Design of Product Families“. In: *International Journal of Production Research* 42.15 (2004), S. 2955–2969 (Zitiert auf S. 32, 60, 130, 146, 159, 184, 186).

[ANSI/NISO10] *ANSI/NISO Z39.19-2005 (R2010) Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*. Standard. American National Standards Institute (ANSI) & National Information Standards Organization (NISO), 2010 (Zitiert auf S. 18, 30, 209).

[AP94] A. Aamodt, E. Plaza. „Case-Based Reasoning; Foundational Issues“. In: *AI Communications* 7.101 (1994), S. 39–59 (Zitiert auf S. 28).

[ARR13] R. Anantharangachar, S. Ramani, S. Rajagopalan. „Ontology Guided Information Extraction from Unstructured Text“. In: *International Journal of Web & Semantic Technology (IJWesT)* 4.1 (2013), S. 19–36 (Zitiert auf S. 116).

[BB18] J. Bräcker, J. Brockmeyer. „Characterization and Detection of Food Allergens Using High-Resolution Mass Spectrometry: Current Status and Future Perspective“. In: *Journal of Agricultural and Food Chemistry* 66.34 (2018), S. 8935–8940 (Zitiert auf S. 164).

[BCF+13] G. Baggio, A. Corsini, A. Floreani, S. Giannini, V. Zagonel. „Gender Medicine: a Task for the Third Millennium“. In: *Clinical Chemistry and Laboratory Medicine (CCLM)* 51.4 (2013), S. 713–727 (Zitiert auf S. 79).

[BCM+07] F. Baader, D. Calvanese, D. L. McGuiness, D. Nardi, P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, Applications*. Cambridge, UK: Cambridge University Pres, 2007 (Zitiert auf S. 30).

[BCN+17] E. Breck, S. Cai, E. Nielsen, M. Salib, D. Sculley. „The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction“. In: *Tagungsband der IEEE International Conference on Big Data*. Boston, MA, USA, 2017, S. 1123–1132 (Zitiert auf S. 70).

[BDC+19] N. Betrouni, A. Delval, L. Chaton, L. Defebvre, A. Duits, A. Moonen, A. F. G. Leentjens, K. Dujardin. „Electroencephalography-based Machine Learning for Cognitive Profiling in Parkinson’s Disease: Preliminary Results“. In: *Movement Disorders Journal* 34.2 (2019), S. 210–217 (Zitiert auf S. 79).

[BDH11] P. Blanchard, R. L. Devaney, G. R. Hall. *Differential Equations*. Boston, MA, USA: Brooks/Cole Thomson Learning, 2011 (Zitiert auf S. 36).

[BE05] J. Barwise, J. Etchemendy. *Sprache, Beweis und Logik. Band 1: Aussagen- und Prädikatenlogik*. Paderborn: Mentis, 2005 (Zitiert auf S. 30).

[BFSM22] M. Behringer, M. Fritz, H. Schwarz, B. Mitschang. „DATA-IMP: An Interactive Approach to Specify Data Imputation Transformations on Large Datasets“. In: *Tagungsband der 28. International Conference on Cooperative Information Systems (CoopIS)*. Bozen-Bolzano, Italien, 2022, S. 55–74 (Zitiert auf S. 47).

[BG07] A. Bifet, R. Gavaldà. „Learning from Time-Changing Data with Adaptive Windowing“. In: *Tagungsband der SIAM International Conference on Data Mining (SDM)*. Minneapolis, MN, USA: Society for Industrial und Applied Mathematics, 2007 (Zitiert auf S. 106).

[BGBO19] L. E. Boiko Ferreira, H. Gomes, A. Bifet, L. S. Oliveira. „Adaptive Random Forests with Resampling for Imbalanced Data Streams“. In: *Tagungsband der 2019 International Joint Conference on Neural Networks (IJCNN)*. Budapest, Ungarn, 2019 (Zitiert auf S. 108).

[BH21] N. Burkart, M. F. Huber. „A Survey on the Explainability of Supervised Machine Learning“. In: *Journal of Artificial Intelligence Research* 70 (2021), S. 245–317 (Zitiert auf S. 44).

[BHFM20] M. Behringer, P. Hirmer, M. Fritz, B. Mitschang. „Empowering Domain Experts to Preprocess Massive Distributed Datasets“. In: *Tagungsband der 23. International Conference on Business Information Systems (BIS)*. Colorado Springs, CO, USA, 2020, S. 61–75 (Zitiert auf S. 47).

[BHM17] M. Behringer, P. Hirmer, B. Mitschang. „Towards Interactive Data Processing and Analytics – Putting the Human in the Center of the Loop“. In: *Tagungsband der 19. International Conference on Enterprise Information Systems (ICEIS)*. Porto, Portugal, 2017 (Zitiert auf S. 97).

[BHM18] M. Behringer, P. Hirmer, B. Mitschang. „A Human-Centered Approach for Interactive Data Processing and Analytics“. In: *Enterprise Information Systems* (2018). Hrsg. von S. Hammoudi, M. Śmiałek, O. Camp, J. Filipe, S. 498–514 (Zitiert auf S. 97).

[BHTM22] M. Behringer, P. Hirmer, D. Tschechlov, B. Mitschang. „Increasing Explainability of Clustering Results for Domain Experts by Identifying Meaningful Features“. In: *Tagungsband der 24. International Conference on Enterprise Information Systems (ICEIS)*. 2022, S. 364–373 (Zitiert auf S. 47).

[BHV+23] M. Behringer, P. Hirmer, A. G. Villanueva Zacarias, J. Rapp, B. Mitschang. „Unobtrusive Integration of Data Quality in Interactive Explorative Data Analysis“. In: *Tagungsband der 25. International Conference on Enterprise Information Systems (ICEIS)*. Prag, Tschechische Republik: SciTePress, 2023 (Zitiert auf S. 47).

[BHV14] T. Bauernhansl, M. ten Hompel, B. Vogel-Heuser. *Industrie 4.0 in Produktion, Automatisierung und Logistik – Anwendung, Technologien, Migration*. Wiesbaden, Deutschland: Springer Vieweg, 2014 (Zitiert auf S. 176).

[BKLW99] S. Busse, R.-D. Kutsche, U. Leser, H. Weber. *Federated Information Systems: Concepts, Terminology and Architectures*. Techn. Ber. Fakultät Informatik der Technischen Universität Berlin, 1999 (Zitiert auf S. 85).

[BL05] S. Bowers, B. Ludäscher. „Actor-Oriented Design of Scientific Workflows“. In: *Tagungsband der 24. International Conference on Conceptual Modeling (ER)*. Klagenfurt, Österreich: Springer, 2005, S. 369–384 (Zitiert auf S. 162).

[Bor+03] J. Borking et al. *Handbook of Privacy and Privacy-Enhancing Technologies – The Case of Intelligent Software Agents*. 2003 (Zitiert auf S. 96).

[Bre01] L. Breiman. „Random Forests“. In: *Machine Learning* 45.1 (2001), S. 5–32 (Zitiert auf S. 100, 109, 152).

[Bre96] L. Breiman. „Bagging Predictors“. In: *Machine Learning* 24.1 (1996), S. 123–140 (Zitiert auf S. 99, 202).

[BRL+19] S. H. Bach, D. Rodriguez, Y. Liu, C. Luo, H. Shao, C. Xia, S. Sen, A. Ratner, B. Hancock, H. Alborzi, R. Kuchhal, C. Ré, R. Malkin. „Snorkel Drybell: A Case Study in Deploying Weak Supervision at Industrial Scale“. In: *Tagungsband der 2019 ACM SIGMOD International Conference on Management of Data*. Amsterdam, Niederlande, 2019, S. 362–375 (Zitiert auf S. 64, 137, 138).

[BS00] J. M. Bernardo, A. F. M. Smith. *Bayesian Theory*. Chichester, UK: Wiley, 2000 (Zitiert auf S. 35, 191).

[BWD+21] A. Birk, Y. Wilhelm, S. Dreher, C. Flack, P. Reimann, C. Gröger. „A Real-World Application of Process Mining for Data-Driven Analysis of Multi-Level Interlinked Manufacturing Processes“. In: *Procedia CIRP* 104 (2021), S. 417–422 (Zitiert auf S. 16, 55, 56, 73, 74).

[BWHY05] G. Brown, J. Wyatt, R. Harris, X. Yao. „Diversity Creation Methods: A Survey and Categorisation“. In: *Information Fusion* 6.1 (2005), S. 5–20 (Zitiert auf S. 195).

[BWK+22] A. Blum, Y. Wilhelm, S. Klein, C. Schnur, P. Reimann, R. Müller, A. Schütze. „Ganzheitlicher Ablaufplan für wissensgetriebene Projekte des maschinellen Lernens in der Produktion“. In: *tm - Technisches Messen* 89.5 (2022), S. 363–383 (Zitiert auf S. 69, 134, 135, 138, 194).

[BZBP13] H.-J. Bungartz, S. Zimmer, M. Buchholz, D. Pflüger. *Modellbildung und Simulation: Eine anwendungsorientierte Einführung*. Berlin, Heidelberg: Springer Spektrum, 2013 (Zitiert auf S. 18, 27, 36, 209).

[Cao17] L. Cao. „Data Science: A Comprehensive Overview“. In: *ACM Computing Surveys* 50.3 (2017) (Zitiert auf S. 22).

[CBHK02] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer. „SMOTE: Synthetic Minority Over-sampling Technique“. In: *Journal Of Artificial Intelligence Research* 16.1 (2002), S. 321–357 (Zitiert auf S. 83, 90, 91).

[CCF+09] C. E. Catalano, E. Camossi, R. Ferrandes, V. Cheutet, N. Sevilmis. „A Product Design Ontology for Enhancing Shape Processing in Design Workflows“. In: *Journal of Intelligent Manufacturing* 20.5 (2009), S. 553–567 (Zitiert auf S. 122).

[CCK+00] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth. *CRISP-DM 1.0: Step-by-step Data Mining Guide*. SPSS Inc., CRISP-DM-Konsortium, 2000 (Zitiert auf S. 15, 37, 45, 47).

[CCS+18] Y. Cheng, K. Chen, H. Sun, Y. Zhang, F. Tao. „Data and Knowledge Mining with Big Data towards Smart Production“. In: *Journal of Industrial Information Integration* 9 (2018) (Zitiert auf S. 16, 24, 50, 66, 69, 70, 73).

[CGH97] E. Castillo, J. M. Gutiérrez, A. S. Hadi. *Expert Systems and Probabilistic Network Models*. New York City, NY, USA: Springer-Verlag, 1997 (Zitiert auf S. 35).

[Chu41] A. Church. *The Calculi of Lambda-Conversion*. Princeton University Press, 1941 (Zitiert auf S. 163, 164).

[CKC20] Y. Cui, S. Kara, K.-C. Chan. „Manufacturing Big Data Ecosystem: A Systematic Literature Review“. In: *Robotics and Computer Integrated Manufacturing* 62 (2020) (Zitiert auf S. 66).

[CLPC17] J. Chen, F. Lécué, J. Z. Pan, H. Chen. „Learning from Ontology Streams with Semantic Concept Drift“. In: *Tagungsband der 26. International Joint Conference on Artificial Intelligence (IJCAI)*. Melbourne, Australien, 2017, S. 957–963 (Zitiert auf S. 137, 211).

[Cow11] F. Cowell. *Measuring Inequality*. 3rd. Oxford Academic, 2011 (Zitiert auf S. 86, 128, 151, 161, 182).

[CRM+20] S. Chan, V. Reddy, B. Myers, Q. Thibodeaux, N. Brownstone, W. Liao. „Machine Learning in Dermatology: Current Applications, Opportunities, and Limitations“. In: *Dermatology and Therapy* 10.3 (2020), S. 365–386 (Zitiert auf S. 17, 20, 49, 60, 74, 75, 77, 78, 80, 125, 130, 159, 160, 186).

[DB21] A. Dogan, D. Birant. „Machine Learning and Data Mining in Manufacturing“. In: *Expert Systems with Applications* 166.2 (2021) (Zitiert auf S. 15, 16, 24, 50, 60, 63, 66).

[DC19] J. Dahmen, D. Cook. „SynSys: A Synthetic Data Generation System for Healthcare Applications“. In: *Sensors* 5 (2019) (Zitiert auf S. 88).

[DEF+24] S. Durnagöz, T. Eberhardt, F. Fuchs, P. Reimann, T. Ackermann, M. Fiedler, M. Mayer, M. F. Huber, A. Verl. „A Novel Approach for Unsupervised Sensor Error Classification“. In: *Procedia CIRP* 130 (2024), S. 206–213 (Zitiert auf S. 50).

[DHMR24] S. Durnagöz, M. F. Huber, M. Mayer, P. Reimann. „An Approach to Inline Monitoring of the Electrode State in Resistance Spot Welding“. In: *International Journal of Electrical and Electronic Engineering & Telecommunications* 13 (3 2024), S. 245–251 (Zitiert auf S. 50).

[Die00] T. G. Dietterich. „An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization“. In: *Machine Learning* 40.2 (2000), S. 139–157 (Zitiert auf S. 99–102, 195, 202).

[DIN06] DIN EN 60812. *Analysetechniken für die Funktionsfähigkeit von Systemen – Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA) (IEC 60812:2006); Deutsche Fassung EN 60812:2006*. Berlin, Deutschland: Deutsches Institut für Normung e.V., 2006 (Zitiert auf S. 29, 30, 193).

[DJR+20] C. Deng, X. Ji, C. Rainey, J. Zhang, W. Lu. „Integrating Machine Learning with Human Knowledge“. In: *iScience* 23.11 (2020) (Zitiert auf S. 26).

[DKW+22] A. Daw, A. Karpatne, W. D. Watkins, J. S. Read, V. Kumar. „Physics-Guided Neural Networks (PGNN): An Application in Lake Temperature Modeling“. In: Hrsg. von A. Karpatne, R. Kannan, V. Kumar. Knowledge Guided Machine Learning. New York City, NY, USA: Chapman und Hall/CRC, 2022 (Zitiert auf S. 140, 189, 190).

[DPW+19] T. Deist, A. Patti, Z. Wang, D. Krane, T. Sorenson, D. Craft. „Simulation-Assisted Machine Learning“. In: *Bioinformatics* 35.20 (2019), S. 4072–4080 (Zitiert auf S. 139).

[Dra10] R. Draht. *Datenaustausch in der Anlagenplanung mit AutomationML*. Berlin, Heidelberg: Springer, 2010 (Zitiert auf S. 34, 117, 176).

[DRG17] M. Diligenti, S. Roychowdhury, M. Gori. „Integrating Prior Knowledge into Deep Learning“. In: *Tagungsband der 16. IEEE International Conference on Machine Learning and Applications (ICMLA)*. Cancun, Mexico, 2017, S. 920–923 (Zitiert auf S. 189, 190).

[DRG21] S. Dreher, P. Reimann, C. Gröger. „Application Fields and Research Gaps of Process Mining in Manufacturing Companies“. In: *Tagungsband der INFORMATIK 2020*. Hrsg. von R. H. Reussner, A. Koziolek, R. Heinrich. Gesellschaft für Informatik (GI), 2021, S. 621–634 (Zitiert auf S. 16, 55, 56, 73, 74, 124, 208, 212).

[DWGB04] C. W. Dankwort, R. Weidlich, B. Guenther, J. E. Blaurock. „Engineers' CAx Education — it's not only CAD“. In: *Computer-Aided Design* 36.14 (2004), S. 1439–1450 (Zitiert auf S. 51, 73, 118, 122).

[EGG+21a] R. Eichler, C. Giebler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang. „Enterprise-Wide Metadata Management – An Industry Case on the Current State and Challenges“. In: *Tagungsband der 24. International Conference on Business Information Systems (BIS)*. Hannover, Deutschland, 2021, S. 269–279 (Zitiert auf S. 19, 31, 113–115, 122, 209).

[EGG+21b] R. Eichler, C. Giebler, C. Gröger, H. Schwarz, B. Mitschang. „Modeling Metadata in Data Lakes – A Generic Model“. In: *Data & Knowledge Engineering* 136 (2021) (Zitiert auf S. 115).

[EGH+22] R. Eichler, C. Gröger, E. Hoos, H. Schwarz, B. Mitschang. „Data Shopping — How an Enterprise Data Marketplace Supports Data Democratization in Companies“. In: *Tagungsband der 34. International Conference on Advanced Information Systems Engineering (CAiSE)*. Leuven, Belgien: Springer International Publishing, 2022, S. 19–26 (Zitiert auf S. 115).

[EMN+15] D. Esteves, D. Moussallem, C. B. Neto, T. Soru, R. Usbeck, M. Ackermann, J. Lehmann. „MEX Vocabulary: A Lightweight Interchange Format for Machine Learning Experiments“. In: *Tagungsband der 11. International Conference on Semantic Systems (SEMANTICS)*. Wien, Österreich: ACM, 2015, S. 169–176 (Zitiert auf S. 170).

[EP11] R. Elwell, R. Polikar. „Incremental Learning of Concept Drift in Nonstationary Environments“. In: *IEEE Transactions on Neural Networks* 22.10 (2011), S. 1517–1531 (Zitiert auf S. 70, 106).

[EW16] L. Ehrlinger, W. Wöß. „Towards a Definition of Knowledge Graphs“. In: *SEMANTiCS* 48.1-4 (2016) (Zitiert auf S. 18, 34, 209).

[FGG+18] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer, A. Taylor. „Cypher: An Evolving Query Language for Property Graphs“. In: *Tagungsband der 2018 ACM SIGMOD International Conference on Management of Data*. Houston, TX, USA: Association for Computing Machinery (ACM), 2018, S. 1433–1445 (Zitiert auf S. 123).

[Fin03] G. A. Fink. *Mustererkennung mit Markov-Modellen: Theorie, Praxis, Anwendungsgebiete*. Wiesbaden: Vieweg+Teubner Verlag, 2003 (Zitiert auf S. 35).

[Fit88] T. B. Fitzpatrick. „The Validity and Practicality of Sun-reactive Skin Types I Through VI“. In: *Archives of Dermatology* 124.6 (1988), S. 869–871 (Zitiert auf S. 130, 159, 160, 186).

[FKE+19] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, F. Hutter. „Auto-Sklearn: Efficient and Robust Automated Machine Learning“. In: *Automated Machine Learning*. Hrsg. von F. Hutter,

L. Kotthoff, J. Vanschoren. Cham: Springer International Publishing, 2019, S. 113–134 (Zitiert auf S. 103).

[FKL+17] Y. Fang, K. Kuan, J. Lin, C. Tan, V. Chandrasekhar. „Object Detection Meets Knowledge Graphs“. In: *Tagungsband der 26. International Joint Conference on Artificial Intelligence (IJCAI)*. Melbourne, Australien: AAAI Press, 2017, S. 1661–1667 (Zitiert auf S. 189).

[FLG+13] A. Fernández, V. López, M. Galar, M. J. del Jesus, F. Herrera. „Analysing the Classification of Imbalanced Data-Sets with Multiple Classes: Binarization Techniques and ad-hoc Approaches“. In: *Knowledge-Based Systems* 42 (2013), S. 97–110 (Zitiert auf S. 93, 161).

[FLSB11] J. V. Frasch, A. Lodwich, F. Shafait, T. M. Breuel. „A Bayes-true Data Generator for Evaluation of Supervised and Unsupervised Learning Methods“. In: *Pattern Recognition Letters* 32.11 (2011), S. 1523–1531 (Zitiert auf S. 88).

[FP08] J. H. Ferziger, M. Perić. *Numerische Strömungsmechanik*. Springer-Verlag Berlin Heidelberg, 2008 (Zitiert auf S. 51).

[FPS96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. „The KDD Process for Extracting Useful Knowledge from Volumes of Data“. In: *Communications of the ACM* 39.11 (1996), S. 27–34 (Zitiert auf S. 15, 37, 39, 45).

[FS18] P. Fränti, S. Sieranoja. „K-Means Properties on Six Clustering Benchmark Datasets“. In: *Applied Intelligence* 48.12 (2018), S. 4743–4759 (Zitiert auf S. 88).

[FS96] Y. Freund, R. E. Schapire. „Experiments with a New Boosting Algorithm“. In: *Tagungsband der 13. International Conference on Machine Learning (ICML)*. Hrsg. von L. Saitta. Bari, Italien, 1996, S. 148–156 (Zitiert auf S. 100, 101).

[FSC+20] E. C. Fernandes, L. I. dos Santos, J. A. Camatti, L. Brown, M. Borsato. „Flexible Production Data Generator for Manufacturing Companies“. In: *Procedia Manufacturing* 51 (2020), S. 1478–1484 (Zitiert auf S. 17, 41).

[FZD14] M. V. França, G. Zaverucha, A. S. D’avila Garcez. „Fast Relational Learning Using Bottom Clause Propositionalization with Artificial Neural Networks“. In: *Machine Learning* 94.1 (2014), S. 81–104 (Zitiert auf S. 188).

[GA07] C. W. Günther, W. M. P. van der Aalst. „Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics“. In: *Tagungsband der 5. International Conference on Business Process Management (BPM)*. Brisbane, Australia, 2007, S. 328–343 (Zitiert auf S. 74, 124, 208, 212).

[GA11] C. Grosan, A. Abraham. *Intelligent Systems: A Modern Approach*. Berlin, Heidelberg: Springer-Verlag, 2011 (Zitiert auf S. 27, 30).

[GBR+17] H. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, T. Abdessalem. „Adaptive Random Forests for Evolving Data Stream Classification“. In: *Machine Learning* 106.9–10 (2017), S. 1469–1495 (Zitiert auf S. 108).

[GFB+11] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera. „An Overview of Ensemble Methods for Binary Classifiers in Multi-Class Problems: Experimental Study on One-Vs-One and One-Vs-All Schemes“. In: *Pattern Recognition* 44.8 (2011), S. 1761–1776 (Zitiert auf S. 93, 94).

[GFB+12] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera. „A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches“. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.4 (2012), S. 463–484 (Zitiert auf S. 17, 58, 59, 83, 90, 129).

[GH19] C. Gröger, E. Hoos. „Ganzheitliches Metadatenmanagement im Data Lake: Anforderungen, IT-Werkzeuge und Herausforderungen in der Praxis“. In: *Tagungsband der 18. Fachtagung für Datenbanksysteme für Business, Technologie und Web (BTW)*. Rostock, Deutschland: Gesellschaft für Informatik (GI), 2019, S. 435–452 (Zitiert auf S. 113, 114).

[GH23] C. Giebler, E. Hoos. „A Systematic Approach to Consuming Data in Complex Data Management Landscapes Using Data Consumption Patterns“. In: *Datenbank-Spektrum* 23.2 (2023), S. 117–122 (Zitiert auf S. 57).

[Gin21] C. Gini. „Measurement of Inequality of Incomes“. In: *The Economic Journal* 31.121 (1921), S. 124–126 (Zitiert auf S. 86, 128, 151, 182).

[GLN+05] J. Gray, D. T. Liu, M. Nieto-Santisteban, A. Szalay, D. J. DeWitt, G. Heber. „Scientific Data Management in the Coming Decade“. In: *SIGMOD Record* 34.4 (2005), S. 34–41 (Zitiert auf S. 19, 113, 131, 209).

[GMF+19] C. d. S. Garcia, A. Meincheim, E. R. Faria Junior, M. R. Dallagassa, D. M. V. Sato, D. R. Carvalho, E. A. P. Santos, E. E. Scalabrin. „Process Mining Techniques and Applications – A Systematic Mapping Study“. In: *Expert Systems with Applications* 133.1 (2019), S. 260–295 (Zitiert auf S. 55).

[GNH22] I. Gauger, T. Nagel, M. Huber. „Hybrides Maschinelles Lernen im Kontext der Produktion“. In: Digitalisierung souverän gestalten II. Berlin, Heidelberg: Springer, 2022, S. 64–79 (Zitiert auf S. 19, 20, 139, 188–190, 201, 203, 205, 206, 209, 210).

[GNS11] K. Ghosh, Y. S. Ng, R. Srinivasan. „Evaluation of Decision Fusion Strategies for Effective Collaboration among Heterogeneous Fault Diagnostic Methods“. In: *Computers & Chemical Engineering* 35.2 (2011), S. 342–355 (Zitiert auf S. 198).

[GPM+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. „Generative Adversarial Nets“. In: *Tagungsband der 28. Annual Conference on Neural Information Processing Systems (NIPS)*. 2014 (Zitiert auf S. 41, 88).

[Grö15] C. Gröger. *Advanced Manufacturing Analytics - Datengetriebene Optimierung von Fertigungsprozessen*. Josef Eul Verlag, Dissertation, Universität Stuttgart, 2015 (Zitiert auf S. 50).

[Grö18] C. Gröger. „Building an Industry 4.0 Analytics Platform“. In: *Datenbank-Spektrum* 18.1 (2018), S. 5–15 (Zitiert auf S. 16, 22, 57, 85, 97, 172, 207).

[Grö21] C. Gröger. „There Is No AI Without Data“. In: *Communications of the ACM* 64.11 (2021), S. 98–108 (Zitiert auf S. 15, 57, 67, 69, 85, 97, 98).

[Grö22] C. Gröger. „Industrial Analytics — An Overview“. In: *it – Information Technology* 64.1–2 (2022), S. 55–65 (Zitiert auf S. 15, 21–24, 49, 50, 52, 57, 58, 60, 63, 66, 67, 69, 74, 98, 125).

[GRS07] C. Grossmann, H.-G. Roos, M. Stynes. *Numerical Treatment of Partial Differential Equations*. Berlin, Heidelberg, New York City: Springer-Verlag, 2007 (Zitiert auf S. 36).

[GSH+20] A. Gerling, U. Schreier, A. Hess, A. Saleh, H. Ziekow, D. Ould Abdeslam. „A Reference Process Model For Machine Learning Aided Production Quality Management“. In: *Tagungsband der 22. International Conference on Enterprise Information Systems (ICEIS)*. Prag, Tschechische Republik, 2020, S. 515–523 (Zitiert auf S. 15, 23, 25, 38, 39, 50, 52, 60, 61, 64).

[GSSG12] B. Gong, Y. Shi, F. Sha, K. Grauman. „Geodesic Flow Kernel for Unsupervised Domain Adaptation“. In: *Tagungsband der IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA, 2012, S. 2066–2073 (Zitiert auf S. 141, 142).

[Guy03] I. Guyon. „Design of Experiments for the NIPS 2003 Variable Selection Benchmark“. In: *Tagungsband der 16. Annual Conference on Neural Information Processing Systems (NIPS)*. Vancouver, BC, Kanada, 2003 (Zitiert auf S. 127).

[GWB+06] G. Guo, H. Wang, D. Bell, Y. Bi, K. Greer. „Using kNN Model for Automatic Text Categorization“. In: *Soft Computing* 10.5 (2006), S. 423–430 (Zitiert auf S. 55).

[Har96] S. Hartmann. „The World as a Process: Simulations in the Natural and Social Sciences“. In: *Simulation and Modelling in the Social Sciences from the Philosophy of Science Point of View, Theory and Decision Library*. Hrsg. von R. Hegselmann, U. Mueller, K. Troitzsch. Dordrecht, Niederlande: Kluwer, 1996, S. 77–100 (Zitiert auf S. 36).

[Hau81] E. Haug. „Engineering Safety Analysis via Destructive Numerical Experiments“. In: *Engineering Transactions* 29.1 (1981), S. 39–49 (Zitiert auf S. 36).

[HB02] T. K. Ho, M. Basu. „Complexity Measures of Supervised Classification Problems“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.3 (2002), S. 289–300 (Zitiert auf S. 128, 129, 183).

[HB11] H. M. Hashemian, W. C. Bean. „State-of-the-Art Predictive Maintenance Techniques“. In: *IEEE Transactions on Instrumentation and Measurement* 60.10 (2011), S. 3480–3492 (Zitiert auf S. 69, 208).

[HB17] P. Hirmer, M. Behringer. „FlexMash 2.0 – Flexible Modeling and Execution of Data Mashups“. In: *Rapid Mashup Development Tools*. Springer International Publishing, 2017, S. 10–29 (Zitiert auf S. 97).

[HBH+21] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard, C. Karen Liu, J. Peters, S. Song, P. Welinder, M. White. „Sim2Real in Robotics and Automation: Applications and Challenges“. In: *IEEE Transactions on Automation Science and Engineering* 18.2 (2021), S. 398–400 (Zitiert auf S. 141).

[HBM18] P. Hirmer, M. Behringer, B. Mitschang. „Partial Execution of Mashup Plans During Modeling Time“. In: *Computer Science - Research and Development* 33.3-4 (2018), S. 341–352 (Zitiert auf S. 97).

[HCG+22] F. Hermann, B. Chen, G. Ghasemi, V. Stegmaier, T. Ackermann, P. Reimann, S. Vogt, T. Graf, M. Weyrich. „A Digital Twin Approach for the Prediction of the Geometry of Single Tracks Produced by Laser Metal Deposition“. In: *Procedia CIRP* 107 (2022), S. 83–88 (Zitiert auf S. 50).

[HG09] H. He, E. A. Garcia. „Learning from Imbalanced Data“. In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), S. 1263–1284 (Zitiert auf S. 17, 58, 59, 64, 65, 83, 90, 91, 129).

[Hir21] V. Hirsch. *Datengetriebene Identifikation potenzieller Qualitätsprobleme in komplexen Montageprodukten*. Dissertation, Universität Stuttgart, 2021 (Zitiert auf S. 53, 59, 62, 65, 67, 98, 99).

[HJX16] K. He, M. Jia, Q. Xu. „Optimal Sensor Deployment for Manufacturing Process Monitoring Based on Quantitative Cause-Effect Graph“. In: *IEEE Transactions on Automation Science and Engineering* 13.2 (2016), S. 963–975 (Zitiert auf S. 138, 194).

[HKP12] J. Han, M. Kamber, J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2012 (Zitiert auf S. 25, 42, 142, 143, 188).

[HLS07] M. Hepp, J. Leukel, V. Schmitz. „A Quantitative Analysis of Product Categorization Standards: Content, Coverage, and Maintenance of eCl@ss, UNSPSC, eOTD, and the RosettaNet Technical Dictionary“. In: *Knowledge and Information Systems* 12.1 (2007), S. 77–114 (Zitiert auf S. 33).

[HMB+23] F. Hermann, A. Michalowski, T. Brünnette, P. Reimann, S. Vogt, T. Graf. „Data-Driven Prediction and Uncertainty Quantification of Process Parameters for Directed Energy Deposition“. In: *Materials* 16.23 (2023) (Zitiert auf S. 16, 50).

[HPS15] N. Helwig, E. Pignanelli, A. Schutze. „Condition Monitoring of a Complex Hydraulic System Using Multivariate Statistics“. In: *Tagungsband der IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. Pisa, Italien, 2015, S. 210–215 (Zitiert auf S. 50, 200).

[HRKM18] V. Hirsch, P. Reimann, O. Kirn, B. Mitschang. „Analytical Approach to Support Fault Diagnosis and Quality Control in End-Of-Line Testing“. In: *Procedia CIRP* 72 (2018), S. 1333–1338 (Zitiert auf S. 15, 16, 30, 52, 64).

[HRM19] V. Hirsch, P. Reimann, B. Mitschang. „Data-Driven Fault Diagnosis in End-of-Line Testing of Complex Products“. In: *Tagungsband der 6. IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Washington, D.C., USA: IEEE, 2019, S. 492–503 (Zitiert auf S. 16, 52, 53, 58, 59, 61, 63, 64, 66–69, 83, 86, 90–93, 95, 96, 98, 100–102, 108, 109, 129, 136, 138, 141, 145, 152, 191, 208).

[HRM20a] V. Hirsch, P. Reimann, B. Mitschang. „Exploiting Domain Knowledge to Address Multi-Class Imbalance and a Heterogeneous Feature Space in Classification Tasks for Manufacturing Data“. In: *PVLDB* 13.12 (2020), S. 3258–3271 (Zitiert auf S. 20, 61, 63, 69, 87, 145, 147, 150–154, 161, 166, 210).

[HRM20b] V. Hirsch, P. Reimann, B. Mitschang. „Incorporating Economic Aspects into Recommendation Ranking to Reduce Failure Costs“. In: *Procedia CIRP* 93 (2020), S. 747–752 (Zitiert auf S. 43, 155).

[HRT+23] V. Hirsch, P. Reimann, D. Treder-Tschechlov, H. Schwarz, B. Mitschang. „Exploiting Domain Knowledge to Address Class Imbalance and a Heterogeneous Feature Space in Multi-Class Classification“. In: *International Journal on Very Large Data Bases (VLDB-Journal)* 32.5 (2023), S. 1037–1064 (Zitiert auf S. 17, 19, 20, 32, 60, 61, 63, 86, 88, 129, 130, 136, 139, 141–150, 152–157, 159–161, 165, 166, 176, 183, 184, 186, 204, 208, 210).

[HRWM15] P. Hirmer, P. Reimann, M. Wieland, B. Mitschang. „Extended Techniques for Flexible Modeling and Execution of Data Mashups“. In: *Tagungsband der 4. International Conference on Data Management Technologies and Applications (DATA)*. Colmar, Frankreich: SciTePress, 2015, S. 111–122 (Zitiert auf S. 97).

[HSG+17] J. M. Hellerstein, V. Sreekanti, J. E. Gonzalez, J. Dalton, A. Dey, S. Nag, K. Ramachandran, S. Arora, A. Bhattacharyya, S. Das, M. Donsky, G. Fierro, C. She, C. Steinbach, V. Subramanian, E. Sun. „Ground: A Data Context Service“. In: *Tagungsband der 8. Conference on Innovative Data Systems Research (CIDR)*. Chaminade, CA, USA, 2017 (Zitiert auf S. 115).

[HSS18] J. Hu, L. Shen, G. Sun. „Squeeze-and-Excitation Networks“. In: *Tagungsband der 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA, 2018, S. 7132–7141 (Zitiert auf S. 201).

[Hum12] G. Humphreys. „Coming Together to Combat Rare Diseases“. In: *Bulletin of the World Health Organization* 90.6 (2012), S. 401–476 (Zitiert auf S. 78, 80).

[Hum90] P. Humphreys. „Computer Simulations“. In: *Tagungsband des Biennial Meeting of the Philosophy of Science Association (PSA)*. 1990, S. 497–506 (Zitiert auf S. 36).

[HWM+19] R. Heese, M. Walczak, L. Morand, D. Helm, M. Bortz. „The Good, the Bad and the Ugly: Augmenting a Black-Box Model with Expert Knowledge“. In: *Tagungsband der 28. International Conference on Artificial Neural Networks (ICANN)*. München, Deutschland: Springer-Verlag, 2019, S. 391–395 (Zitiert auf S. 189).

[HZC21] X. He, K. Zhao, X. Chu. „AutoML: A Survey of the State-of-the-Art“. In: *Knowledge-Based Systems* 212 (2021) (Zitiert auf S. 17, 102).

[HZRS16] K. He, X. Zhang, S. Ren, J. Sun. „Deep Residual Learning for Image Recognition“. In: *Tagungsband der 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, 2016, S. 770–778 (Zitiert auf S. 201).

[HZWK08] S. J. Hu, X. Zhu, H. Wang, Y. Koren. „Product Variety and Manufacturing Complexity in Assembly Systems and Supply Chains“. In: *CIRP Annals* 57.1 (2008), S. 45–48 (Zitiert auf S. 61, 146).

[Ise06] R. Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer-Verlag Berlin Heidelberg, 2006 (Zitiert auf S. 29, 30, 52, 73).

[Ish86] K. Ishikawa. *Guide to Quality Control*. Asian Productivity Organization, 1986 (Zitiert auf S. 138, 194).

[ISO15] *ISO 9000:2015: Quality Management Systems – Fundamentals and Vocabulary*. Standard. International Organization for Standardization, ISO/TC 176/SC 1 Concepts und Terminology, 2015 (Zitiert auf S. 52).

[ITW05] G. Ifrim, M. Theobald, G. Weikum. „Learning Word-to-Concept Mappings for Automatic Text Classification“. In: *Tagungsband der 22. International Conference on Machine Learning – Learning in Web Search (LWS 2005)*. Bonn, Deutschland, 2005, S. 18–26 (Zitiert auf S. 19, 135, 136, 209).

[IZFZ19] F. Iglesias, T. Zseby, D. Ferreira, A. Zimek. „MDCGen: Multidimensional Dataset Generator for Clustering“. In: *Journal of Classification* 36 (2019), S. 599–618 (Zitiert auf S. 88).

[Jab04] N. Jablonski. „The Evolution of Human Skin and Skin Color“. In: *Annual Review of Anthropology* 33 (2004), S. 585–623 (Zitiert auf S. 130, 159, 186).

[Jai10] A. K. Jain. „Data Clustering: 50 Years Beyond K-Means“. In: *Pattern Recognition Letters* 31.8 (2010), S. 651–666 (Zitiert auf S. 142).

[JCZ13] C. Jiang, F. Coenen, M. Zito. „A Survey of Frequent Subgraph Mining Algorithms“. In: *The Knowledge Engineering Review* 28.1 (2013), S. 75–105 (Zitiert auf S. 124).

[JN07] F. V. Jensen, T. D. Nielsen. *Bayesian Networks and Decision Graphs*. New York City, NY, USA: Springer-Verlag, 2007 (Zitiert auf S. 35, 191).

[JO23] N. Jahnke, B. Otto. „Data Catalogs in the Enterprise: Applications and Integration“. In: *Datenbank-Spektrum* 23.2 (2023), S. 89–96 (Zitiert auf S. 115).

[KAF+17] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar. „Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data“. In: *IEEE Transactions on Knowledge and Data Engineering* 29.10 (2017), S. 2318–2331 (Zitiert auf S. 190, 206).

[KBT11] G. Köksal, I. Batmaz, M. C. Testik. „A Review of Data Mining Applications for Quality Improvement in Manufacturing Industry“. In: *Expert Systems with Applications* 38.10 (2011), S. 13448–13467 (Zitiert auf S. 24, 50, 60, 66).

[KF09] D. Koller, N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Boston, MA, USA: The MIT Press, 2009 (Zitiert auf S. 18, 35, 191, 206, 209).

[KFPI11] V. Karkaletsis, P. Fragkou, G. Petasis, E. Iosif. „Ontology Based Information Extraction from Text“. In: *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution: Bridging the Semantic Gap*. Hrsg. von G. Paliouras, C. D. Spyropoulos, G. Tsatsaronis. Berlin, Heidelberg, Deutschland: Springer-Verlag, 2011, S. 89–109 (Zitiert auf S. 117).

[KHS+24] M. Klaiber, M. Hug, L. Schneller, Ö. Can, A. Jahn, A. Fehrenbacher, P. Reimann, A. Michalowski. „Automated Image-based Parameter Optimization for Single-Pulse Laser Drilling“. In: *Forum Bildverarbeitung* (2024) (Zitiert auf S. 50).

[KHW+17] L. Kassner, P. Hirmer, M. Wieland, F. Steimle, J. Königsberger, B. Mitschang. „The Social Factory: Connecting People, Machines and Data in Manufacturing for Context-Aware Exception Escalation“. In: *Tagungsband der 50. Hawaii International Conference on System Sciences (HICSS)*. Waikoloa Village, HI, USA, 2017, S. 1673–1682 (Zitiert auf S. 24, 69).

[KKH23] D. Kreuzberger, N. Kühl, S. Hirschl. „Machine Learning Operations (MLOps): Overview, Definition, and Architecture“. In: *IEEE Access* 11 (2023), S. 31866–31879 (Zitiert auf S. 211).

[KŁd+15] C. M. Keet, A. Ławrynowicz, C. d'Amato, A. Kalousis, P. Nguyen, R. Palma, R. Stevens, M. Hilario. „The Data Mining Optimization Ontology“. In: *Web Semantics* 32.C (2015), S. 43–53 (Zitiert auf S. 170).

[KM16] L. Kassner, B. Mitschang. „Exploring Text Classification for Messy Data: An Industry Use Case for Domain-Specific Analytics Technology“. In: *Tagungsband der 19. International Conference on Extending Database Technology (EDBT)*. Bordeaux, France: ACM, 2016, S. 491–502 (Zitiert auf S. 15, 16, 23, 55, 58, 59, 61, 63, 64, 66, 67, 69, 98, 134–136, 208).

[Kol05] P. G. Kolaitis. „Schema Mappings, Data Exchange, and Metadata Management“. In: *Tagungsband des 24. Symposium on Principles of Database Systems (PODS)*. Baltimore, MD, USA: ACM, 2005, S. 61–75 (Zitiert auf S. 97).

[KR10] M. B. Kursa, W. R. Rudnicki. „Feature Selection with the Boruta Package“. In: *Journal of Statistical Software* 36.11 (2010) (Zitiert auf S. 17, 93).

[KRM19a] C. Kiefer, P. Reimann, B. Mitschang. „A Hybrid Information Extraction Approach Exploiting Structured Data Within a Text Mining Process“. In: *Tagungsband der 18. Fachtagung für Datenbanksysteme für Business, Technologie und Web (BTW)*. Rostock, Deutschland: Gesellschaft für Informatik (GI), 2019, S. 149–168 (Zitiert auf S. 16, 23, 53, 64, 66, 67, 69, 142, 143).

[KRM19b] C. Kiefer, P. Reimann, B. Mitschang. „QUALM: Ganzheitliche Messung und Verbesserung der Datenqualität in der Textanalyse“. In: *Datenbank-Spektrum* 19.2 (2019), S. 137–148 (Zitiert auf S. 47, 69, 96, 136).

[KRM20] C. Kiefer, P. Reimann, B. Mitschang. „Prevent Low-Quality Analytics by Automatic Selection of the Best-Fitting Training Data“. In: *Tagungsband der 53. Hawaii International Conference on System Sciences (HICSS)*. Maui, HI, USA, 2020 (Zitiert auf S. 185).

[KRRT98] R. Kimball, L. Reeves, M. Ross, W. Thornthwaite. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses*. New York City, NY, USA: John Wiley & Sons, Inc., 1998 (Zitiert auf S. 85, 97).

[KSL+21] M. von Kurnatowski, J. Schmid, P. Link, R. Zache, L. Morand, T. Kraft, I. Schmidt, J. Schwientek, A. Stoll. „Compensating Data Shortages in Manufacturing with Monotonicity Knowledge“. In: *Algorithms* 14.12 (2021) (Zitiert auf S. 189, 190).

[KU15] A. Khan, M. Usman. „Early Diagnosis of Alzheimer’s Disease Using Machine Learning Techniques: A Review Paper“. In: *Tagungsband der 7. International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*. Lisbon, Portugal, 2015, S. 380–387 (Zitiert auf S. 17, 20, 49, 72, 74–80, 87, 109).

[Kun04] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: Wiley, 2004 (Zitiert auf S. 99, 195, 197, 198, 202).

[Leh21] F. Lehner. *Wissensmanagement: Grundlagen, Methoden und technische Unterstützung*. Carl Hanser Verlag GmbH & Co KG, 2021 (Zitiert auf S. 18, 26).

[LHW18] V. Losing, B. Hammer, H. Wersing. „Incremental on-line Learning: A Review and Comparison of State of the Art Algorithms“. In: *Neurocomputing* 275 (2018), S. 1261–1274 (Zitiert auf S. 44, 106, 185).

[LHZ+18] X. Liang, Z. Hu, H. Zhang, L. Lin, E. P. Xing. „Symbolic Graph Reasoning Meets Convolutions“. In: *Tagungsband der 32. Annual Conference on Neural Information Processing Systems (NIPS)*. Montréal, QC, Kanada, 2018, S. 1858–1868 (Zitiert auf S. 187).

[LJJ07] Y. Liu, R. Jin, A. Jain. „BoostCluster: Boosting Clustering by Pairwise Constraints“. In: *Tagungsband der 13. International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. San Jose, CA, USA, 2007, S. 450–459 (Zitiert auf S. 143, 144, 210).

[LKJ02] E. Lundin, H. Kvarnström, E. Jonsson. „A Synthetic Fraud Data Generation Methodology“. In: *Tagungsband der 2. International Conference on Information and Communications Security (ICICS)*. Singapore, 2002, S. 265–277 (Zitiert auf S. 88).

[LKS13] A. Linden, L. Kart, W. R. Schulte. *Extend Your Portfolio of Analytics Capabilities*. 2013. URL: https://www.gartner.com/en/documents/2594822 (Zitiert auf S. 22, 23, 25).

[LLD+19] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang. „Learning under Concept Drift: A Review“. In: *Transactions on Knowledge and Data Engineering* 31.12 (2019), S. 2346–2363 (Zitiert auf S. 70, 106).

[LLE16] L. Leitner, A. Lagrange, C. Endisch. „End-of-line Fault Detection for Combustion Engines Using One-Class Classification“. In: *Tagungsband der IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. Banff, AB, Kanada, 2016, S. 207–213 (Zitiert auf S. 16, 50, 58, 60).

[LLEF20] C. Labadie, C. Legner, M. Eurich, M. Fadler. „FAIR Enough? Enhancing the Usage of Enterprise Data with Data Catalogs“. In: *Tagungsband der 22. Conference on Business Informatics (CBI)*. 2020, S. 201–210 (Zitiert auf S. 115).

[LLL13] H.-C. Liu, L. Liu, N. Liu. „Risk Evaluation Approaches in Failure Mode and Effects Analysis: A Literature Review“. In: *Expert Systems with Applications* 40.2 (2013), S. 828–838 (Zitiert auf S. 30).

[LN07] U. Leser, F. Naumann. *Informationsintegration – Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen*. dpunkt.verlag, 2007 (Zitiert auf S. 17, 85, 97).

[LP13] F. Lécué, J. Z. Pan. „Predicting Knowledge in an Ontology Stream“. In: *Tagungsband der 23. International Joint Conference on Artificial Intelligence (IJCAI)*. Peking, China: AAAI Press, 2013, S. 2662–2669 (Zitiert auf S. 137).

[LP20] E. LeDell, S. Poirier. „H2O AutoML: Scalable Automatic Machine Learning“. In: *Tagungsband des 7. ICML Workshop on Automated Machine Learning (AutoML)*. 2020. URL: https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf (Zitiert auf S. 103, 178).

[LRZL17] Y. Lu, M. Rajora, P. Zou, S. Y. Liang. „Physics-Embedded Machine Learning: Case Study with Electrochemical Micro-Machining“. In: *Machines* 5.1 (2017) (Zitiert auf S. 188, 190).

[LWMB09] B. Ludäscher, M. Weske, T. McPhillips, S. Bowers. „Scientific Workflows: Business as Usual?“ In: *Tagungsband der 7. International Conference on Business Process Management (BPM)*. Ulm, Germany, 2009, S. 31–47 (Zitiert auf S. 161).

[MB08] K. McHenry, P. Bajcsy. *An Overview of 3D Data Content, File Formats and Viewers*. Techn. Ber. isda08-002. National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, 2008. URL: https://www.archives.gov/files/applied-research/ncsa/8-an-overview-of-3d-data-content-file-formats-and-viewers.pdf (Zitiert auf S. 51).

[MBSJ09] M. Mintz, S. Bills, R. Snow, D. Jurafsky. „Distant Supervision for Relation Extraction without Labeled Data“. In: *Tagungsband des 47. Annual Meeting of the ACL und der 4. International Joint Conference on Natural Language Processing of the AFNLP*. Suntec City, Singapore: Association for Computational Linguistics, 2009, S. 1003–1011 (Zitiert auf S. 138, 195).

[MCD+05] P. Maechling, H. Chalupsky, M. Dougherty, E. Deelman, Y. Gil, S. Gullapalli, V. Gupta, C. Kesselman, J. Kim, G. Mehta, B. Mendenhall, T. Russ, G. Singh, M. Spraragen, G. Staples, K. Vahi. „Simplifying Construction of Complex Workflows for Non-expert Users of the Southern

California Earthquake Center Community Modeling Environment“. In: *SIGMOD Record* 34.3 (2005), S. 24–30 (Zitiert auf S. 161, 162).

[Men20] L. Mende. *Merkmalentstehungs- und -wechselwirkungsanalyse (MEWA) für das prozessorientierte Toleranzmanagement in der Montage*. Dissertation, Universität des Saarlandes, 2020 (Zitiert auf S. 134, 135).

[MGM20] A. M. Maitín, A. J. García-Tejedor, J. P. R. Muñoz. „Machine Learning Approaches for Detecting Parkinson’s Disease from EEG Analysis: A Systematic Review“. In: *Applied Sciences* 10.23 (2020) (Zitiert auf S. 17, 20, 49, 60, 72, 74–81, 109, 125, 159).

[MIM+18] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, N. Ramakrishnan. „Incorporating Prior Domain Knowledge into Deep Neural Networks“. In: *Tagungsband der IEEE International Conference on Big Data*. Seattle, WA, USA, 2018, S. 36–45 (Zitiert auf S. 189).

[Mit97] T. M. Mitchel. *Machine Learning*. McGraw-Hill, 1997 (Zitiert auf S. 25, 42).

[MMR+24] T. Mayr, S. Marinescu, C.-A. Rusu, T. Ackermann, P. Reimann, R. Schulz, O. Riedel. „Application of Machine Learning for Ergonomic Workplace Assessments in Assembly and Logistics to Identify Potential Use Cases for Exoskeleton“. In: *Procedia CIRP* 130 (2024), S. 625–630 (Zitiert auf S. 50).

[MMS+21] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan. „A Survey on Bias and Fairness in Machine Learning“. In: *ACM Computing Surveys* 54.6 (2021) (Zitiert auf S. 56, 58, 61–63, 66, 74, 78, 207).

[MRA+12] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, F. Herrera. „A Unifying View on Dataset Shift in Classification“. In: *Pattern Recognition* 45.1 (2012), S. 521–530 (Zitiert auf S. 16, 44, 69, 70, 72, 74, 106, 185, 208).

[MRT18] M. Mohri, A. Rostamizadeh, A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018 (Zitiert auf S. 25, 26, 197).

[MSF+10] B. Matthews, S. Sufi, D. Flannery, L. Lerusse, T. Griffin, M. Gleaves, K. K. van Dam. „Using a Core Scientific Metadata Model in Large-Scale Facilities“. In: *International Journal of Digital Curation* 5.1 (2010), S. 106–118 (Zitiert auf S. 115).

[MSG17] K. Marino, R. Salakhutdinov, A. Gupta. „The More You Know: Using Knowledge Graphs for Image Classification“. In: *Tagungsband der IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA, 2017, S. 20–28 (Zitiert auf S. 187, 188).

[MVS+19] R. Müller, M. Vette-Steinkamp, M. Scholer, L. Schirmer, A. Blum. „Upgrading and Ensuring a Fully-Automated Assembly Process Using Tolerance Management Methods“. In: *Procedia CIRP* 81 (2019), S. 174–179 (Zitiert auf S. 134, 135).

[MW08] O. L. Mangasarian, E. W. Wild. „Nonlinear Knowledge-Based Classification“. In: *IEEE Transactions on Neural Networks* 19.10 (2008), S. 1826–1832 (Zitiert auf S. 189).

[Nea03] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003 (Zitiert auf S. 20, 192, 206, 210).

[NLDD18] O. Nalbach, C. Linn, M. Derouet, W. Dirk. „Predictive Quality: Towards a New Understanding of Quality Assurance Using Machine Learning Tools“. In: *Tagungsband der 21. International Conference on Business Information Systems (BIS)*. Berlin, Deutschland: Springer International Publishing, 2018, S. 30–42 (Zitiert auf S. 72, 87, 109).

[NPZ+20] F. Nargesian, K. Q. Pu, E. Zhu, B. Ghadiri Bashardoost, R. J. Miller. „Organizing Data Lakes for Navigation“. In: *Tagungsband der 2020 ACM SIGMOD International Conference on Management of Data*. Portland, OR, USA: ACM, 2020, S. 1939–1950 (Zitiert auf S. 116).

[OOD+12] D. de Oliveira, E. Ogasawara, J. Dias, F. Baiao, M. Mattoso. „Ontology-based Semi-automatic Workflow Composition“. In: *Journal of Information and Data Management* 3.1 (2012), S. 61–72 (Zitiert auf S. 161, 162).

[OR02] E. Otte, R. Rousseau. „Social Network Analysis: a Powerful Strategy, also for the Information Sciences“. In: *Journal of Information Science* 28.6 (2002), S. 441–453 (Zitiert auf S. 23).

[OUA+16] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, J. H. Moore et al. „Automating Biomedical Data Science Through Tree-Based Pipeline Optimization“. In: *Tagungsband der European Conference on the Applications of Evolutionary Computation*. Porto, Portugal, 2016, S. 123–137 (Zitiert auf S. 103).

[PEŁ+18] G. Publio, D. Esteves, A. Ławrynowicz, P. Panov, L. Soldatova, T. Soru, J. Vanschoren, H. Zafar. „ML-Schema: Exposing the Semantics of Machine Learning with Schemas and Ontologies“. In: *Tagungsband des Workshops Enabling Reproducibility in Machine Learning an der 35. International Conference on Machine Learning (ICML)*. Stockholm, Schweden, 2018 (Zitiert auf S. 170, 171, 175).

[PMR19] D. Petrik, M. Mormul, P. Reimann. „Anforderungen für Zeitreihendatenbanken in der industriellen Edge“. In: *HMD Praxis der Wirtschaftsinformatik* 56 (6 2019), S. 1282–1308 (Zitiert auf S. 69).

[PMRG21] D. Petrik, M. Mormul, P. Reimann, C. Gröger. „Anforderungen für Zeitreihendatenbanken im industriellen IoT“. In: Hrsg. von S. Meinhardt, F. Wortmann. IoT – Best Practices. Wiesbaden, Deutschland: Springer-Verlag, 2021 (Zitiert auf S. 69).

[Pol06] R. Polikar. „Ensemble Based Systems in Decision Making“. In: *IEEE Circuits and Systems Magazine* 6.3 (2006), S. 21–45 (Zitiert auf S. 100, 101, 195, 198).

[Pol09] M. Polanyi. *The Tacit Dimension*. University of Chicago Press, 2009 (Zitiert auf S. 18, 26).

[PSD14] P. Panov, L. Soldatova, S. Džeroski. „Ontology of Core Data Mining Entities“. In: *Data Mining and Knowledge Discovery* 28.5–6 (2014), S. 1222–1265 (Zitiert auf S. 170).

[PU92] D. C. Psichogios, L. H. Ungar. „A Hybrid Neural Network-First Principles Approach to Process Modeling“. In: *AIChE Journal* 38.10 (1992), S. 1499–1511 (Zitiert auf S. 188).

[Pup91] F. Puppe. *Einführung in Expertensystem*. Berlin, Heidelberg: Springer, 1991 (Zitiert auf S. 28).

[PWV16] N. Patki, R. Wedge, K. Veeramachaneni. „The Synthetic Data Vault“. In: *Tagungsband der 3. IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Montreal, Canada: IEEE, 2016 (Zitiert auf S. 88).

[PY10] S. J. Pan, Q. Yang. „A Survey on Transfer Learning“. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), S. 1345–1359 (Zitiert auf S. 103).

[PZL+18] J. Pfrommer, C. Zimmerling, J. Liu, L. Kärger, F. Henning, J. Beyerer. „Optimisation of Manufacturing Process Parameters Using Deep Neural Networks as Surrogate Models“. In: *Procedia CIRP* 72 (2018), S. 426–431 (Zitiert auf S. 51, 139).

[QSJ+05] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, C. F. Jeff Wu. „Building Surrogate Models Based on Detailed and Approximate Simulations“. In: *Journal of Mechanical Design* 128.4 (2005), S. 668–677 (Zitiert auf S. 51).

[Qui67] R. Quillian. „Word Concepts. A Theory and Simulation of Some Basic Semantic Capabilities“. In: *Behavioral Science* 12 (1967), S. 410–430 (Zitiert auf S. 158).

[RB01] E. Rahm, P. A. Bernstein. „A Survey of Approaches to Automatic Schema Matching“. In: *International Journal on Very Large Data Bases (VLDBJ)* 10.4 (2001), S. 334–350 (Zitiert auf S. 97).

[RBE+20] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, C. Ré. „Snorkel: Rapid Training Data Creation with Weak Supervision“. In: *VLDB Journal* 29 (2020), S. 709–730 (Zitiert auf S. 18, 19, 27, 47, 64, 137, 138, 195, 209).

[RD03] M. Richardson, P. Domingos. „Learning with Knowledge from Multiple Experts“. In: *Tagungsband der 20. International Conference on Machine Learning (ICML)*. Washington, D.C., USA: AAAI Press, 2003, S. 624–631 (Zitiert auf S. 192).

[Rei10] K. Reichenberger. *Kompendium semantische Netze: Konzepte, Technologie, Modellierung*. Springer, 2010 (Zitiert auf S. 27, 31, 32, 158, 162).

[Rei17] P. Reimann. *Data Provisioning in Simulation Workflows*. Dissertation, Universität Stuttgart, 2017 (Zitiert auf S. 85, 108, 111, 116, 117, 119, 165).

[Rei23] P. Reimann. „Metadata Modeling and Use of Domain Knowledge to Support Industrial Data Analytics“. In: *Tagungsband des 34. GI-Workshops Grundlagen von Datenbanken (GvDB)*. Hrsg. von H. Schwarz. Calw-Hirsau, Deutschland: CEUR Workshop Proceedings, 2023 (Zitiert auf S. 17, 119, 145).

[RGS+22] A. Rivolli, L. P. Garcia, C. Soares, J. Vanschoren, A. C. de Carvalho. „Meta-Features for Meta-Learning“. In: *Knowledge-Based Systems* 240 (2022) (Zitiert auf S. 104, 171, 185).

[RHW21] Y. Roh, G. Heo, S. E. Whang. „A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective“. In: *IEEE Transactions on Knowledge and Data Engineering* 33.4 (2021), S. 1328–1347 (Zitiert auf S. 17, 41, 88).

[Ric03] M. M. Richter. „Fallbasiertes Schließen“. In: *Informatik-Spektrum* 26 (2003), S. 180–190 (Zitiert auf S. 28).

[RLS+06] U. Radetzki, U. Leser, S. Schulze-Rauschenbach, J. Zimmermann, J. Lüssem, T. Bode, A. B. Cremers. „Adapters, Shims, and Glue – Service Interoperability for in Silico Experiments“. In: *Bioinformatics* 22.9 (2006), S. 1137–1143 (Zitiert auf S. 162).

[RMB+23] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, J. Schuecker. „Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems“. In: *IEEE Transactions on Knowledge and Data Engineering* 35 (1 2023), S. 614–633 (Zitiert auf S. 17, 19, 20, 139, 141, 186–189, 203, 205, 206, 208, 210).

[Ros03] R. G. Ross. *Principles of the Business Rule Approach*. Boston, MA, USA: Addison-Wesley, 2003 (Zitiert auf S. 18, 30, 209).

[Row07] J. Rowley. „The Wisdom Hierarchy: Representations of the DIKW Hierarchy“. In: *Journal of Information Science* 33.2 (2007) (Zitiert auf S. 26).

[RRS+11] P. Reimann, M. Reiter, H. Schwarz, D. Karastoyanova, F. Leymann. „SIMPL – A Framework for Accessing External Data in Simulation Workflows“. In: *Tagungsband der 14. Fachtagung für Datenbanksysteme für Business, Technologie und Web (BTW)*. Kaiserslautern, Deutschland: Gesellschaft für Informatik (GI), 2011, S. 534–553 (Zitiert auf S. 119, 162).

[RS13] P. Reimann, H. Schwarz. „Datenmanagementpatterns in Simulationsworkflows“. In: *Tagungsband der 15. Fachtagung für Datenbanksysteme für Business, Technologie und Web (BTW)*. Hrsg. von G. für In-

formatik (GI). Magdeburg, Deutschland, 2013, S. 279–293 (Zitiert auf S. 97).

[RSM14a] P. Reimann, H. Schwarz, B. Mitschang. „A Pattern Approach to Conquer the Data Complexity in Simulation Workflow Design“. In: *Tagungsband der 22. International Conference on Cooperative Information Systems (CoopIS)*. Hrsg. von R. Meersman et al. Amantea, Italien: Springer-Verlag, 2014, S. 21–38 (Zitiert auf S. 165).

[RSM14b] P. Reimann, H. Schwarz, B. Mitschang. „Data Patterns to Alleviate the Design of Scientific Workflows Exemplified by a Bone Simulation“. In: *Tagungsband der 26. International Conference on Scientific and Statistical Database Management (SSDBM)*. Aalborg, Dänemark, 2014 (Zitiert auf S. 97).

[RWH+21] L. von Rueden, T. Wirtz, F. Hueger, J. D. Schneider, N. Piatkowski, C. Bauckhage. „Street-Map Based Validation of Semantic Segmentation in Autonomous Driving“. In: *Tagungsband der 25. International Conference on Pattern Recognition (ICPR)*. Mailand, Italien, 2021, S. 10203–10210 (Zitiert auf S. 189).

[Sad20] H. Sadhasivan. *Knowledge Representation and Automated Reasoning for Decision Support on Fault Correction in Manufacturing*. Masterarbeit, Universität Stuttgart, 2020 (Zitiert auf S. 35, 36, 192).

[SAH18] M. Sharp, R. Ak, T. J. Hedberg. „A Survey of the Advancing Use and Development of Machine Learning in Smart Manufacturing“. In: *Journal of Manufacturing Systems* 48.C (2018), S. 170–179 (Zitiert auf S. 15, 24, 50, 66).

[SB21] G. Steinbuss, K. Böhm. „Benchmarking Unsupervised Outlier Detection with Realistic Synthetic Data“. In: *ACM Transaction on Knowledge Discovery from Data* 15.4 (2021), S. 1–20 (Zitiert auf S. 88).

[SBE+21] C. Stach, J. Bräcker, R. Eichler, C. Giebler, B. Mitschang. „Demand-Driven Data Provisioning in Data Lakes: BARENTS — A Tailorable Data Preparation Zone“. In: *Tagungsband der 23. International Conference on Information Integration and Web Intelligence (iiWAS ’21)*. Linz, Österreich: ACM, 2021, S. 191–202 (Zitiert auf S. 19, 97, 161–165).

[SBK+17] S. Schelter, J.-H. Böse, J. Kirschnick, T. Klein, S. Seufert. „Automatically Tracking Metadata and Provenance of Machine Learning Experiments“. In: *Tagungsband des Machine Learning Systems Workshops an der 31. Conference on Neural Information Processing Systems (NIPS)*. Long Beach, CA, USA, 2017 (Zitiert auf S. 171).

[Sch12] M. J. Schervish. *Theory of Statistics*. Springer Science & Business Media, 2012 (Zitiert auf S. 126).

[Sch19] R. Schmitt. „Quality Assurance“. In: *CIRP Encyclopedia of Production Engineering*. Hrsg. von S. Chatti, L. Laperrière, G. Reinhart, T. Tolio. Berlin, Heidelberg: Springer, 2019, S. 1402–1406 (Zitiert auf S. 60, 73).

[Sch90] R. E. Schapire. „The Strength of Weak Learnability“. In: *Machine Learning* 5.1 (1990), S. 197–227 (Zitiert auf S. 99).

[SE17] R. Stewart, S. Ermon. „Label-Free Supervision of Neural Networks with Physics and Domain Knowledge“. In: *Tagungsband der 31. AAAI Conference on Artificial Intelligence*. San Francisco, CA, USA: AAAI Press, 2017, S. 2576–2582 (Zitiert auf S. 189, 190, 203).

[SES23] C. Stach, R. Eichler, S. Schmidt. „A Recommender Approach to Enable Effective and Efficient Self-Service Analytics in Data Lakes“. In: *Datenbank-Spektrum* 23.2 (2023), S. 123–132 (Zitiert auf S. 161–164).

[SFMP21] E. Salvato, G. Fenu, E. Medvet, F. A. Pellegrino. „Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning“. In: *IEEE Access* 9 (2021), S. 153171–153187 (Zitiert auf S. 139, 141).

[SG21] H. Suresh, J. V. Guttag. „A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle“. In: *Tagungsband der 1. ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO)*. ACM, 2021 (Zitiert auf S. 56, 58, 61–66, 74–78, 80, 159, 207).

[SGT+09] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini. „The Graph Neural Network Model“. In: *IEEE Transactions on Neural Networks* 20.1 (2009), S. 61–80 (Zitiert auf S. 187).

[SH05] D. Steinley, R. Henson. „OCLUS: An Analytic Method for Generating Clusters with Known Overlap“. In: *Journal of Classification* 22.2 (2005), S. 221–250 (Zitiert auf S. 88).

[She99] A. P. Sheth. „Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics“. In: *Interoperating Geographic Information Systems*. Hrsg. von M. Goodchild, M. Egenhofer, R. Fegeas, C. Kottman. Boston, MA, USA: Springer US, 1999, S. 5–29 (Zitiert auf S. 85, 113, 116).

[SHT19] S. Schmid, C. Henson, T. Tran. „Using Knowledge Graphs to Search an Enterprise Data Lake“. In: *Tagungsband der 16. European Semantic Web Conference*. Portoroz, Slowenien: Springer-Verlag, 2019, S. 262–266 (Zitiert auf S. 116).

[SIM+21] M. P. Sesmero, J. A. Iglesias, E. Magán, A. Ledezma, A. Sanchis. „Impact of the Learners Diversity and Combination Method on the Generation of Heterogeneous Classifier Ensembles“. In: *Applied Soft Computing* 111 (2021) (Zitiert auf S. 195).

[Sin01] A. Singhal. „Modern Information Retrieval: A Brief Overview“. In: *IEEE Data Engineering Bulletin* 24.4 (2001), S. 35–44 (Zitiert auf S. 186).

[SL90] A. P. Sheth, J. A. Larson. „Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases“. In: *ACM Computing Surveys* 22.3 (1990), S. 183–236 (Zitiert auf S. 85).

[Sow91] J. F. Sowa. *Principles of Semantic Networks. Explorations in the Representation of Knowledge*. Representation and Reasoning. Morgan Kaufmann, 1991 (Zitiert auf S. 18, 31, 32, 158, 162, 209).

[SR19] M. Spieß, P. Reimann. „Angepasstes Item Set Mining zur gezielten Steuerung von Bauteilen in der Serienfertigung von Fahrzeugen“. In: *Workshop-Band der 18. Fachtagung für Datenbanksysteme für Business, Technologie und Web (BTW)*. Rostock, Deutschland: Gesellschaft für Informatik (GI), 2019, S. 119–128 (Zitiert auf S. 54, 55).

[SRM25] U. Schreier, P. Reimann, B. Mitschang. „A Kanban-based Approach to Manage Machine Learning Projects in Manufacturing“. In: *Procedia CIRP: 58. Conference on Manufacturing Systems (CIRP CMS)* (2025) (Zitiert auf S. 45).

[SRWM22] M. Spieß, P. Reimann, C. Weber, B. Mitschang. „Analysis of Incremental Learning and Windowing to handle Combined Dataset Shifts on Binary Classification for Product Failure Prediction“. In: *Tagungsband der 24. International Conference on on Enterprise Information Systems (ICEIS)*. SCITEPRESS, 2022, S. 394–405 (Zitiert auf S. 16, 54, 55, 60, 63, 69, 70, 72, 73, 81, 87, 98, 104–109, 137, 208).

[SRYD14] C. Sun, N. Rampalli, F. Yang, A. Doan. „Chimera: Large-Scale Classification Using Machine Learning, Rules, and Crowdsourcing“. In: *PVLDB* 7.13 (2014), S. 1529–1540 (Zitiert auf S. 16, 18, 19, 27, 32, 47, 54, 59, 61, 62, 64, 66, 87, 130, 137–139, 159, 176, 184, 186, 195, 208, 209).

[SS09a] C. Spreckelsen, K. Spitzer. *Wissensbasen und Expertensysteme in der Medizin*. Wiesbaden: Vieweg+Teubner Verlag, 2009 (Zitiert auf S. 18, 28, 209).

[SS09b] S. Staab, R. Studer. *Handbook on Ontologies*. Berlin, Heidelberg: Springer-Verlag, 2009 (Zitiert auf S. 33).

[SSF+19] P. Sawadogo, E. Scholly, C. Favre, E. Ferey, S. Loudcher, J. Darmont. „Metadata Systems for Data Lakes: Models and Features“. In: *Tagungsband des International Workshop on BI & Big Data Applications an der 23. European Conference on Advances in Databases and Information Systems (ADBIS)*. Bled, Slowenien, 2019 (Zitiert auf S. 115).

[Sta20] J. Stark. *Product Lifecycle Management (Volume 1) : 21st Century Paradigm for Product Realisation*. Springer International Publishing, 2020 (Zitiert auf S. 21).

[SWK09] Y. Sun, A. Wong, M. Kamel. „Classification of Imbalanced Data: A Review“. In: *International Journal of Pattern Recognition and Artificial Intelligence* 23.4 (2009), S. 687–719 (Zitiert auf S. 17, 63, 83, 90).

[SXF+24] A. Shoshi, Y. Xia, A. Fieschi, T. Ackermann, P. Reimann, M. Weyrich, B. Mitschang, T. Bauernhansl, R. Miehe. „A Flexible Digital Twin Framework for ATMP Production – Towards an Efficient CAR-T Cell Manufacturing“. In: *Procedia CIRP* 125 (2024), S. 124–129 (Zitiert auf S. 50).

[SXF+25] A. Shoshi, Y. Xia, A. Fieschi, Y. Baumgarten, A. Gaißler, T. Ackermann, P. Reimann, B. Mitschang, M. Weyrich, T. Bauernhansl, R. Miehe. „An Analysis of Monitoring Solutions for CAR T Cell Production“. In: *Healthcare Technology Letters* 12 (2025) (Zitiert auf S. 50).

[TFR+17] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel. „Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World“. In: *Tagungsband der IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, BC, Canada: IEEE Press, 2017, S. 23–30 (Zitiert auf S. 141).

[TFSM23] D. Treder-Tschechlov, M. Fritz, H. Schwarz, B. Mitschang. „ML2DAC: Meta-Learning to Democratize AutoML for Clustering Analysis“. In: *Tagungsband der 2023 ACM SIGMOD International Conference on Management of Data*. Seattle, WA, USA: Association for Computing Machinery, 2023 (Zitiert auf S. 104, 171).

[TGS+19] S. Thalmann, H. G. Gursch, J. Suschnigg, M. Gashi, H. Ennsbrunner, A. K. Fuchs, T. Schreck, B. Mutlu, J. Mangler, G. Kappl, C. Huemer, S. Lindstaedt. „Cognitive Decision Support for Industrial Product Life Cycles: A Position Paper“. In: *Tagungsband der 11. International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE)*. Venedig, Italien: IARIA, 2019, S. 3–9 (Zitiert auf S. 16, 50, 61).

[THHL13] C. Thornton, F. Hutter, H. H. Hoos, K. Leyton-Brown. „Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms“. In: *Tagungsband der 19. International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. Chicago, IL, USA: ACM, 2013, S. 847–855 (Zitiert auf S. 102, 103).

[TRSM23] D. Treder-Tschechlov, P. Reimann, H. Schwarz, B. Mitschang. „Approach to Synthetic Data Generation for Imbalanced Multi-class Problems with Heterogeneous Groups“. In: *Tagungsband der 20. Fachtagung für Datenbanksysteme für Business, Technologie und Web (BTW)*. Dresden, Deutschland: Gesellschaft für Informatik (GI), 2023 (Zitiert auf S. 16, 19, 20, 41, 63, 87, 88, 112, 125–132, 139, 141, 146, 183, 209).

[TS94] G. G. Towell, J. W. Shavlik. „Knowledge-Based Artificial Neural Networks“. In: *Artificial Intelligence* 70.1 (1994), S. 119–165 (Zitiert auf S. 188).

[TSRC15] I. G. Terrizzano, P. M. Schwarz, M. Roth, J. E. Colino. „Data Wrangling: The Challenging Journey from the Wild to the Lake“. In: *Tagungsband der 7. Conference on Innovative Data Systems Research (CIDR)*. Asilomar, CA, USA, 2015 (Zitiert auf S. 114).

[TZH07] A. Tsymbal, S. Zillner, M. Huber. „Ontology–Supported Machine Learning and Decision Support in Biomedicine“. In: *Tagungsband des 4. International Workshops on Data Integration in the Life Sciences (DILS)*. Philadelphia, PA, USA: Springer, 2007, S. 156–171 (Zitiert auf S. 186, 188).

[ULP19] B. Ustun, Y. Liu, D. Parkes. „Fairness without Harm: Decoupled Classifiers with Preference Guarantees“. In: *Tagungsband der 36. International Conference on Machine Learning*. 2019, S. 6373–6382 (Zitiert auf S. 87).

[Utg89] P. E. Utgoff. „Incremental Induction of Decision Trees“. In: *Machine Learning* 4.2 (1989), S. 161–186 (Zitiert auf S. 106).

[Van19] J. Vanschoren. „Meta-Learning“. In: *Automated Machine Learning - Methods, Systems, Challenges*. Hrsg. von F. Hutter, L. Kotthoff, J. Vanschoren. Cham, Switzerland: Springer, 2019, S. 35–61 (Zitiert auf S. 98, 103).

[VFP+05] M. Valle, J. Favre, E. Parkinson, A. Perrig, M. Fahrat. „Scientific Data Management for Visualization Implementation Experience“. In: *Tagungsband der Konferenz Simulation und Visualisierung 2005 (SimVis 2005)*. Magdeburg, Germany: SCS Publishing House e.V., 2005, S. 347–354 (Zitiert auf S. 116).

[VGR20] A. G. Villanueva Zacarias, R. Ghabri, P. Reimann. „AD4ML: Axiomatic Design to Specify Machine Learning Solutions for Manufacturing“. In: *Tagungsband der 21. International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2020, S. 148–155 (Zitiert auf S. 47, 176).

[Via13] S. Viaene. „Data Scientists Aren't Domain Experts“. In: *IT Professional* 15.6 (2013), S. 12–17 (Zitiert auf S. 97, 121).

[Vij21] M. Vijayaruban. *Erklärbarkeit von Data Set Shifts in Zeitreihendaten von Feinstaubmessungen*. Masterarbeit, Universität Stuttgart, 2021 (Zitiert auf S. 211).

[Vil21] A. G. Villanueva Zacarias. *Concepts and Methods for the Design, Configuration and Selection of Machine Learning Solutions in Manufacturing*. Dissertation, Universität Stuttgart, 2021 (Zitiert auf S. 173).

[VK11] S. Vinodh, D. Kuttalingam. „Computer-aided Design and Engineering as Enablers of Agile Manufacturing“. In: *Journal of Manufacturing Technology Management* 22.3 (2011), S. 405–418 (Zitiert auf S. 36, 51, 73, 118).

[VKFK17] B. Vogel-Heuser, V. Karaseva, J. Folmer, I. Kirchen. „Operator Knowledge Inclusion in Data-Mining Approaches for Product Quality Assurance using Cause-Effect Graphs“. In: *IFAC-PapersOnLine* 50.1 (2017), S. 1358–1365 (Zitiert auf S. 138, 194).

[VLT10] S. Verron, J. Li, T. Tiplica. „Fault Detection and Isolation of Faults in a Multivariate Process with Bayesian Network“. In: *Journal of Process Control* 20.8 (2010), S. 902–911 (Zitiert auf S. 47, 134).

[Vog23] J. Voggesberger. „Optimierung von Klassifikator-Ensembles mit AutoML“. In: *Tagungsband des 34. Workshops für Grundlagen von Datenbanken*. Hirsau-Calw, Deutschland: Gesellschaft für Informatik (GI), 2023 (Zitiert auf S. 204).

[VRBT13] J. Vanschoren, J. N. van Rijn, B. Bischl, L. Torgo. „OpenML: Networked Science in Machine Learning“. In: *SIGKDD Explorations* 15.2 (2013), S. 49–60 (Zitiert auf S. 86, 87, 104, 167, 169–171, 174, 205).

[VRM23] J. Voggesberger, P. Reimann, B. Mitschang. „Towards the Automatic Creation of Optimized Classifier Ensembles“. In: *Tagungsband der 25. International Conference on Enterprise Information Systems (ICEIS)*. Prag, Tschechische Republik: SciTePress, 2023, S. 614–621 (Zitiert auf S. 195, 202, 205).

[VRWM23] A. G. Villanueva Zacarias, P. Reimann, C. Weber, B. Mitschang. „AssistML: An Approach to Manage, Recommend and Reuse ML Solutions“. In: *International Journal of Data Science and Analytics (JDSA)* 16 (2023), S. 455–479 (Zitiert auf S. 20, 104, 168, 171–174, 177–179, 181–183, 185, 205, 210).

[VS10] J. Vanschoren, L. Soldatova. „Exposé: An Ontology for Data Mining Experiments“. In: *Tagungsband des Workshops on Third Generation Data Mining an der European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*. Barcelona, Spanien, 2010 (Zitiert auf S. 170).

[VSL+16] M. Vartak, H. Subramanyam, W.-E. Lee, S. Viswanathan, S. Husnoo, S. Madden, M. A. Zaharia. „ModelDB: A System for Machine Learning Model Management“. In: *Tagungsband des 1. Workshops on Human-In-the-Loop Data Analytics (HILDA)*. San Francisco, CA, USA, 2016 (Zitiert auf S. 19, 167, 169, 170, 174, 205, 210).

[VWRM21] A. G. Villanueva Zacarias, C. Weber, P. Reimann, B. Mitschang. „AssistML: A Concept to Recommend ML Solutions for Predictive Use Cases“. In: *Tagungsband der 8. International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2021, S. 148–155 (Zitiert auf S. 20, 104, 168, 171, 177, 205, 210).

[WAH+07] K. Wolstencroft, P. Alper, D. Hull, C. J. Wroe, P. W. Lord, R. D. Stevens, C. A. Goble. „The myGrid Ontology: Bioinformatics Service Discovery“. In: *International Journal of Bioinformatics Research and Applications* 3.3 (2007), S. 303–325 (Zitiert auf S. 161, 162).

[WCRS01] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl. „Constrained K-means Clustering with Background Knowledge“. In: *Tagungsband der 18. International Conference on Machine Learning (ICML)*. Williamstown, MA, USA, 2001, S. 577–584 (Zitiert auf S. 19, 143, 144, 210).

[WD10] D. C. Wimalasuriya, D. Dou. „Ontology-based Information Extraction: An Introduction and a Survey of Current Approaches“. In: *Journal of Information Science* 36.3 (2010), S. 306–323 (Zitiert auf S. 85, 115–117).

[Web21] C. Weber. *Konzeption und Realisierung einer Plattform zur Verwaltung von Modellen des maschinellen Lernens im Kontext von Industrie 4.0*. Dissertation, Universität Stuttgart, 2021 (Zitiert auf S. 169, 175–177, 185).

[WFH11] I. H. Witten, E. Frank, M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011 (Zitiert auf S. 42).

[WGC14] M. Woźniak, M. Graña, E. Corchado. „A Survey of Multiple Classifier Systems as Hybrid Systems“. In: *Information Fusion* 16 (2014), S. 3–17 (Zitiert auf S. 17, 98, 99, 109, 195, 202).

[WHC+16] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, F. Petitjean. „Characterizing Concept Drift“. In: *Data Mining and Knowledge Discovery* 30.4 (2016), S. 964–994 (Zitiert auf S. 16, 69, 70, 106).

[WHR20] C. Weber, P. Hirmer, P. Reimann. „A Model Management Platform for Industry 4.0 - Enabling Management of Machine Learning Models in Manufacturing Environments“. In: *Tagungsband der 23. International Conference on Business Information Systems (BIS)*. 2020, S. 403–417 (Zitiert auf S. 19, 20, 104, 168, 170, 175, 177, 179–181, 184, 186, 205, 210, 211).

[WHRS19] C. Weber, P. Hirmer, P. Reimann, H. Schwarz. „A New Process Model for the Comprehensive Management of Machine Learning Models“. In: *Tagungsband der 21. International Conference on Enterprise Information Systems (ICEIS)*. Heraklion, Kreta, Griechenland: SCITEPRESS, 2019, S. 415–422 (Zitiert auf S. 44, 185, 211).

[WKR+21] E. Wagner, B. Keller, P. Reimann, C. Gröger, D. Spath. „Advanced Analytics for Evaluating Critical Joining Technologies in Automotive Body Structures and Body Shops“. In: *Procedia CIRP* 112 (2021), S. 442–447 (Zitiert auf S. 50, 61).

[WLL21] J. Wang, Y. Liu, C. Levy. „Fair Classification with Group-Dependent Label Noise“. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2021, S. 526–536 (Zitiert auf S. 87).

[Wol92] D. H. Wolpert. „Stacked Generalization“. In: *Neural Networks* 5.2 (1992), S. 241–259 (Zitiert auf S. 198).

[WP03] G. M. Weiss, F. Provost. „Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction“. In: *Journal of Artificial Intelligence Research* 19 (2003), S. 315–354 (Zitiert auf S. 58, 129).

[WR20] C. Weber, P. Reimann. „A Platform to Manage Machine Learning Models in Industry 4.0 Environments“. In: *Tagungsband der 24. International Enterprise Distributed Object Computing Workshops (EDOCW)*. IEEE, 2020 (Zitiert auf S. 20, 168, 176, 205, 210).

[WRG+23] Y. Wilhelm, P. Reimann, W. Gauchel, S. Klein, B. Mitschang. „PUSION-A Generic and Automated Framework for Decision Fusion“. In: *Tagungsband der 39. International Conference on Data Engineering (ICDE)*. Anaheim, CA, USA: IEEE, 2023 (Zitiert auf S. 20, 196–204, 206, 210).

[WRGM21] Y. Wilhelm, P. Reimann, W. Gauchel, B. Mitschang. „Overview on Hybrid Approaches to Fault Detection and Diagnosis: Combining Data-driven, Physics-based and Knowledge-based Models“. In: *Procedia CIRP* 99 (2021), S. 278–283 (Zitiert auf S. 16, 17, 19, 20, 52, 74, 192, 194, 195, 208).

[WS19] H. P. Whitley, W. D. Smith. „Sex-based Differences in Medications for Heart Failure“. In: *The Lancet* 394.10205 (2019), S. 1210–1212 (Zitiert auf S. 79).

[WS96] R. Y. Wang, D. M. Strong. „Beyond Accuracy: What Data Quality Means to Data Consumers“. In: *Journal of Management of Information Systems* 12.4 (1996), S. 5–33 (Zitiert auf S. 39).

[WSR+20] Y. Wilhelm, U. Schreier, P. Reimann, B. Mitschang, H. Ziekow. „Data Science Approaches to Quality Control in Manufacturing: A Review of Problems, Challenges and Architecture“. In: *Tagungsband des 14. Symposium on Service-Oriented Computing (SummerSOC)*. Communications in Computer and Information Science (CCIS). Springer-Verlag, 2020, S. 45–65 (Zitiert auf S. 23, 37–39, 52, 58, 60, 61, 63, 64, 66–70, 73, 98, 108, 146, 191).

[WVV+01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hübner. „Ontology-Based Integration of Information – A Survey of Existing Approaches“. In: *Tagungsband des IJCAI-01 Workshops on Ontologies and Information Sharing*. Seattle, WA, USA, 2001, S. 108–117 (Zitiert auf S. 85, 115, 116).

[WWIT16] T. Wuest, D. Weimer, C. Irgens, K.-D. Thoben. „Machine Learning in Manufacturing: Advantages, Challenges, and Applications“. In: *Production & Manufacturing Research* 4.1 (2016), S. 23–45 (Zitiert auf S. 16, 24, 50, 57, 58, 60, 61, 63, 64, 66, 69, 74, 98, 125).

[WWR18] C. Weber, M. Wieland, P. Reimann. „Konzepte zur Datenverarbeitung in Referenzarchitekturen für Industrie 4.0: Konsequenzen bei der Umsetzung einer IT-Architektur“. In: *Datenbank-Spektrum* 18.1 (2018), S. 39–50 (Zitiert auf S. 31, 113, 115).

[WY12] S. Wang, X. Yao. „Multiclass Imbalance Problems: Analysis and Potential Solutions“. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.4 (2012), S. 1119–1130 (Zitiert auf S. 17, 58, 83, 90, 129).

[XDCC19] G. Xiao, L. Ding, B. Cogrel, D. Calvanese. „Virtual Knowledge Graphs: An Overview of Systems and Use Cases“. In: *Data Intelligence* 1 (2019), S. 201–223 (Zitiert auf S. 116).

[XWL+21] D. Xin, E. Y. Wu, D. J.-L. Lee, N. Salehi, A. Parameswaran. „Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows“. In: *Tagungsband der CHI Conference on Human Factors in Computing Systems*. Yokohama, Japan, 2021, S. 8–13 (Zitiert auf S. 17, 102).

[YL19] Z.-X. Ye, Z.-H. Ling. „Distant Supervision Relation Extraction with Intra-Bag and Inter-Bag Attentions“. In: *Tagungsband der 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, MN, USA: Association for Computational Linguistics, 2019, S. 2810–2819 (Zitiert auf S. 138, 195).

[YNSC19] G. D. Y, N. G. Nair, P. Satpathy, J. Christopher. „Covariate Shift: A Review and Analysis on Classifiers“. In: *2019 Global Conference for Advancement in Technology (GCAT)*. Bangalore, India, 2019 (Zitiert auf S. 70, 106).

[YPR+14] B. Yet, Z. B. Perkins, T. E. Rasmussen, N. R. Tai, W. R. Marsh. „Combining Data and Meta-Analysis to Build Bayesian Networks for Clinical Decision Support“. In: *Journal of Biomedical Informatics* 52 (2014), S. 373–385 (Zitiert auf S. 192).

[YQL11] Z. Yang, L. Qing, P. Lu. „Integration of Deep and Shallow Aircraft Fault Knowledge“. In: *Tagungsband der 3. IEEE International Conference on Communication Software and Networks*. Xi’an, China, 2011, S. 320–324 (Zitiert auf S. 27).

[ZCD+18] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, F. Xie, C. Zumar. „Accelerating the Machine Learning Lifecycle with MLflow“. In: *IEEE Data Engineering Bulletin* 41 (2018), S. 39–45 (Zitiert auf S. 19, 104, 167, 169, 170, 205, 210).

[ZG15] F. Zhang, Z. Ge. „Decision Fusion Systems for Fault Detection and Identification in Industrial Processes“. In: *Journal of Process Control* 31 (2015), S. 45–54 (Zitiert auf S. 198).

[ZH21] M.-A. Zöller, M. Huber. „Benchmark and Survey of Automated Machine Learning Frameworks“. In: *Journal of Artificial Intelligence Research* 70 (2021), S. 409–472 (Zitiert auf S. 17, 98, 102).

[ZP15] G. Zheng, S. Peltsverger. „Web Analytics Overview“. In: *Encyclopedia of Information Science and Technology* (2015), S. 7674–7683 (Zitiert auf S. 23).

[ZQW20] W. Zhao, J. P. Queralta, T. Westerlund. „Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey“. In: *IEEE Symposium Series on Computational Intelligence (SSCI)*. 2020, S. 737–744 (Zitiert auf S. 141, 142).

[ZRK+24] J. Ziegler, P. Reimann, A. Kilian, C. Schulz, F. Keller, B. Mitschang. „An Approach to Integrate Domain Knowledge into Feature Engineering to Enhance Data-Driven Surrogate Models of Simulations“. In: *Procedia CIRP* 130 (2024), S. 1276–1281 (Zitiert auf S. 123, 134, 135).

[ZRKM20] J. Ziegler, P. Reimann, F. Keller, B. Mitschang. „A Graph-based Approach to Manage CAE Data in a Data Lake“. In: *Procedia CIRP* 93 (2020), S. 496–501 (Zitiert auf S. 15, 16, 20, 36, 57, 85, 108, 111, 114, 123–125, 131, 185, 207).

[ZRKM21] J. Ziegler, P. Reimann, F. Keller, B. Mitschang. „A Metadata Model to Connect Isolated Data Silos and Activities of the CAE Domain“. In: *Tagungsband der 33. International Conference on Advanced Information Systems Engineering (CAiSE)*. Springer International Publishing, 2021, S. 213–228 (Zitiert auf S. 19, 20, 31, 51, 57, 67, 85, 108, 111–113, 116, 118–125, 131, 132, 185, 209).

[ZRS+22] J. Ziegler, P. Reimann, C. Schulz, F. Keller, B. Mitschang. „A Graph Structure to Discover Patterns in Unstructured Processes of Product Development“. In: *Tagungsband der 23. International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2022 (Zitiert auf S. 55, 56, 73, 74, 122–125, 208, 212).

[ZTZ13] O. C. Zienkiewicz, R. L. Taylor, J. Zhu. *The Finite Element Method: its Basis and Fundamentals*. 7. Aufl. Oxford, UK: Butterworth-Heinemann, Elsevier, 2013 (Zitiert auf S. 36, 51).

Abbildungsverzeichnis

2.1 *Industrial Analytics Cycle*: Kontinuierlicher Lernzyklus im Produktlebenszyklus, vgl. Gröger [Grö22]. 22
2.2 Im Rahmen dieser Habilitationsschrift primär untersuchte Arten formal repräsentierten Domänenwissens sowie die jeweils genutzte formale Struktur zur Abbildung des Domänenwissens. 29
2.3 Wichtigste Arten semantischer Netze sowie die jeweils zugelassenen Relationen zwischen domänenspezifischen Begriffen. . . 33
2.4 Schritte des generischen Vorgehensmodells zur Umsetzung von Datenanalysen sowie zentrale Aufgaben in den sechs Schritten. 38
2.5 Zuordnung der wichtigsten Hauptakteure, die ihr jeweiliges Fachwissen in die Aufgaben der einzelnen Schritte des generischen Vorgehensmodells aus Abbildung 2.4 einbeziehen. 46

3.1 End-of-Line-Prüffeld (EoL-Prüffeld), Fehlerdiagnose und Nacharbeit für Motoren, vgl. Hirsch et al. [Hir21; HRM19]. 53
3.2 Illustration der Klassenungleichverteilung anhand zwei Klassen der Daten vom EoL-Prüffeld, vgl. Hirsch [Hir21]. 59
3.3 Abstrahierte Produkthierarchie zur Einteilung von Verbrennungsmotoren in Produktgruppen, vgl. Hirsch et al. [HRT+23]. . . . 61
3.4 Illustration der Gruppenheterogenität und Gruppenungleichverteilung anhand zwei Klassen der Daten vom EoL-Prüffeld, vgl. Hirsch [Hir21]. 62

3.5 Illustration von falsch dokumentierten Klassenlabels anhand zwei Klassen der Daten vom EoL-Prüffeld, vgl. Hirsch [Hir21]. 65
3.6 Ausschnitt aus dem Datenintegrationsprozesses für den Anwendungsfall im EoL-Prüffeld, vgl. Hirsch [Hir21]. 67
3.7 Schematische Darstellung einer Konzeptverschiebung anhand zweier Features f_1 und f_2 sowie anhand zweier Klassen. 70
3.8 Schematische Darstellung einer Kovariatenverschiebung anhand zweier Features f_1 und f_2 sowie anhand dreier Klassen. 72

4.1 In diesem Kapitel diskutierte rein datengetriebene Ansätze für die Schritte Datenakquise und Datenexploration, Datenvorbereitung sowie Datenanalyse und Modellbildung. 84
4.2 Beispielhafte Anwendung von *Random Undersampling* (RUS) auf den Datensatz aus Abbildung 3.5. 91
4.3 Beispielhafte Anwendung von SMOTE auf den in Abbildung 4.2b dargestellten Datensatz. 92
4.4 Anwendung der Binärisierungstechnik *One-vs-One* (OvO) auf einen Datensatz mit zwei Features f_1, f_2 sowie vier Klassen 'Kreis', 'Stern', Raute' und 'Viereck'. 94
4.5 Beispielhafte Anwendung der Binärisierungstechnik *One-vs-All* (OvA) auf den Datensatz aus Abbildung 4.4a. 95
4.6 Datenvorbereitungs-Pipeline mit beispielhaften Datenvorbereitungsoperationen. 97
4.7 Anwendung eines Ensembles mit n Klassifikationsmodellen auf eine neue Dateninstanz, vgl. [Hir21; WGC14]. 99

5.1 Durch Domänenwissen unterstützte Aufgaben in der Datenakquise und Datenexploration sowie Kategorisierung der Aufgaben und entsprechender Ansätze hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens. 112
5.2 Grundlegende Ansätze, die Ontologien nutzen, um verschiedene Datenquellen mit semantischen Informationen zu abstrahieren, vgl. Reimann [Rei17]. 117

5.3 Beispiel einer Instanz des Metadatenmodels, das Daten, Metadaten und Arbeitsaktivitäten von virtuellen Produktentwicklungsprojekten beschreibt, vgl. Ziegler et al. [ZRKM21]. 120
5.4 Systemarchitektur zur Umsetzung des graphbasierten Metadatenmodells sowie die Datenalysen, die für die Validierung genutzt wurden, vgl. Ziegler et al. [ZRKM21]. 123
5.5 Vorgehensweise des Ansatzes zur Nutzung einer Taxonomie zur Generierung von Daten mit spezifischen Ausprägungen einer Gruppenheterogenität, Gruppenungleichverteilung und Klassenungleichverteilung, vgl. Treder-Tschechlov et al. [TRSM23]. 127

6.1 Durch Domänenwissen unterstützte Aufgaben in der Datenvorbereitung sowie Kategorisierung der Aufgaben und entsprechender Ansätze hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens. 134
6.2 Drei gängige Vorgehensweisen zur Anreicherung eines Datensatzes mit Ergebnissen einer Simulation. 140
6.3 Schematische Darstellung einer Constraint-basierten Clusteranalyse. 144
6.4 Wesentliche Schritte des Ansatzes, der eine Taxonomie und Metriken über Klassenverteilungen nutzt, um eine Gruppenheterogenität, Gruppenungleichverteilung und Klassenungleichverteilung zu adressieren, vgl. Hirsch et al. [HRM20a] 147
6.5 Produkthierarchie aus Abbildung 3.3 mit beispielhafter Darstellung, für welche Produktgruppen SPH typischerweise primäre Datenteilmengen oder Surrogat-Teilmengen bildet. 148
6.6 Beispielhafte Klassenverteilung für die Gruppen $OM54$ und $DM34$ von Verbrennungsmotoren nach der Anwendung von SPH und von CPI, vgl. Hirsch et al. [HRT+23]. 150
6.7 Abstraktes hierarchisches Modell zur Illustration der Voraussetzungen von SPH für das Domänenwissen, vgl. [HRT+23]. . . . 157
6.8 Flache Taxonomie zur Kategorisierung von Hautfarben gemäß des Fitzpatrick-Typs, vgl. [HRT+23]. 160

6.9 Schematische Darstellung einer in der BARENTS-Ontologie spezifizierten Regel zur Datenvorbereitung, vgl. Stach et al. [SES23]. 163

7.1 Durch Domänenwissen unterstützte Aufgaben in der Datenanalyse und Modellbildung sowie Kategorisierung der Aufgaben und entsprechender Ansätze hinsichtlich der formalen Repräsentationsform des genutzten Domänenwissens. 168
7.2 Zentrale Beispiele von Metadaten zur Beschreibung von ML-Modellen und Daten in Modellverwaltungsplattformen. 170
7.3 Vorgehensweise des AsssistML-Ansatz zum Vorschlag von ML-Lösungen, vgl. Villanueva [Vil21]. 173
7.4 Abstrakte Sicht auf zentrale Teile des Metadatenmodells der Modellverwaltungsplattform von Weber [Web21]. 175
7.5 Schematische Darstellung der wichtigsten Ansätze zur Nutzung von Ontologien, Wissensgraphen oder Gleichungen aus Simulationsmodellen beim Training von ML-Modellen. 187
7.6 Zentrale Schritte zur Erzeugung eines probabilistischen Graphen unter Einbeziehung von Domänenexperten sowie durch Kombination mit datengetriebenen Lernverfahren. 193
7.7 Darstellung eines beispielhaften Ensembles, das neben Vorhersagen eines datengetrieben erzeugten Entscheidungsbaums die Vorhersagen mehrerer wissensbasierter Modelle kombiniert. . 196
7.8 Zentrale Komponenten des PUSION-Rahmenwerks zur Optimierung der Entscheidungsfusion in Klassifikations-Ensembles. . . 198

Tabellenverzeichnis

1.1 Ansätze zur Einbeziehung formal repräsentierten Domänenwissens in Datenanalyseprozessen, die im Rahmen von Projekten entstanden sind, die vom Autor dieser Schrift betreut wurden. 20

2.1 Wesentliche Unterscheidungsmerkmale zwischen Web Analytics und Industrial Analytics, vgl. Gröger [Grö22]. 24
2.2 Wesentliche Unterschiede zwischen formal repräsentiertem Domänenwissen und informellem Wissen der Domänenexperten. 27
2.3 Ausprägungen und Bezeichnungen der Schritte des generischen Vorgehensmodells zur Umsetzung von Datenanalysen in CRISP-DM, KDD, Microsoft TDSP und SEMMA. 40

3.1 Übersicht wesentlicher Merkmale der untersuchten industriellen Anwendungsfälle für Datenanalysen. 50
3.2 Datencharakteristika industrieller Anwendungsfälle und die jeweils primär verursachten Ausprägungen von Bias in Daten. . . 58
3.3 Übersicht wesentlicher Merkmale der untersuchten Anwendungsfälle für datengetriebene medizinische Diagnosen. 75
3.4 Auftreten verschiedener Ursachen für Bias in den Daten der untersuchten Anwendungsfälle medizinischer Diagnosen. . . . 78

4.1 Datencharakteristika, die Datensätze von öffentlichen Repositorys oder von existierenden Datengeneratoren aufweisen 87
4.2 Auswirkungen verschiedener Ansätze zur Datenvorbereitung auf die Ursachen für Bias in Daten. 89
4.3 Auswirkungen verschiedener Ensemble-Methoden auf die Ursachen für Bias in Daten. 100
4.4 Auswirkungen von AutoML-Systemen und Ansätzen zum Meta-Learning auf die Ursachen für Bias in Daten. 103
4.5 Wahr-Positiv-Raten im Windowing-Verfahren für den Anwendungsfall von Spieß et al. [SRWM22]. 107

5.1 Die wichtigsten Arten von Metadaten, was sie hauptsächlich beschreiben sowie welchen Einsatzzweck sie jeweils unterstützen; nach Gröger und Hoos [GH19]. 114

6.1 Ansätze zur Nutzung von Domänenwissen in der Datenvorbereitung sowie Auswirkungen auf die Ursachen für Bias. 135
6.2 Wesentliche Eigenschaften der für die Evaluation des Verfahrens von Hirsch et al. [HRT+23] verwendeten Datensätze. 153
6.3 Zentrale Ergebnisse der Evaluation des neuen Verfahrens zur Nutzung einer Taxonomie für die Datenpartitionierung. 155

7.1 Ansätze zur Nutzung von Domänenwissen in der Datenanalyse sowie Auswirkungen auf die Ursachen für Bias in Daten. 169
7.2 Inhalte des zur Evaluation von AssistML verwendeten Modell-Repositorys, vgl. Villanueva et al. [VRWM23]. 178
7.3 Zusammenfassung der Metadaten oder Meta-Features, mit denen AssistML bzw. die MMP die jeweiligen Ursachen für Bias in Daten adressieren. 181
7.4 Wesentliche Aspekte der Anwendungsfälle für die Evaluation des PUSION-Rahmenwerks [WRG+23]. 200