Neuronale Modelle zur Offboard-Diagnostik in komplexen Fahrzeugsystemen

Von der Fakultät für Elektrotechnik, Informationstechnik, Physik
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

Dipl.-Ing. Tobias Carsten Müller

aus Coburg

Eingereicht am:	4. Februar 2011
Mündliche Prüfung am:	15. September 2011
Berichterstatter:	Prof. Dr.-Ing. Thomas Form
Mitberichterstatter:	Prof. Dr.-Ing. Hans-Christian Reuss

2011

Zusammenfassung

Neue innovative Diagnostikverfahren werden notwendig, um stetig komplexer werdende Fahrzeuge auch zukünftig noch effizient warten zu können. Zunehmend vernetzte Fahrzeugfunktionen, steigende Variantenvielfalt und kürzere Entwicklungszyklen führen dazu, dass etablierte Experten-basierte Verfahren an ihre Grenzen stoßen. Die Folge sind höhere Kosten durch längere Fehlersuchzeiten, unnötig getauschte Teile und überforderte Kundendienstmitarbeiter.

Bereits seit einigen Jahren werden umfangreich Daten von Reparaturfällen aus Servicewerkstätten aufgezeichnet, die bisher jedoch nicht automatisiert für Diagnostikverfahren genutzt werden. Im Rahmen eines Kooperationsprojektes mit der Volkswagen AG und der TU Braunschweig wurde ein Offboard-Diagnostik-System entwickelt, welches zuerst in einem automatisierten Prozess aus aufgezeichneten Reparaturfällen Diagnostik-Modelle erlernt. In den Servicewerkstätten erzeugen die Modelle auf Basis von Fahrzeuginformationen und Fehlersymptomen eines defekten Fahrzeugs Hypothesen über schadhafte Komponenten und sinnvolle Reparaturmaßnahmen. Das Ergebnis der Reparatur wird wiederum aufgezeichnet, wodurch ein iterativer Feedback- und Lösungsprozess zur automatischen kontinuierlichen Verbesserung der Modelle entsteht. Eine große Herausforderung stellen dabei die Art und die Qualität der zur Verfügung stehenden Reparaturfalldaten dar. Die Dualität zwischen Symptominformationen und Kontextinformationen, die hohe Merkmalsdimension, hohe Sparsity und mangelnde Qualität der Daten machen neue Modelle und Lernverfahren notwendig.

In dieser Arbeit werden zunächst deskriptive und inferenzielle statistische Diagnostik-Modelle analysiert. Diese können jedoch aufgrund der beschränkten Modellkomplexität bedingte Abhängigkeiten zwischen Fahrzeugsymptomen und Fahrzeugvarianten nicht modellieren, was zu einer reduzierten Leistungsfähigkeit führt. Eine Lösung hierfür stellen neuronale Netze dar, deren Komplexität durch Architekturmodifikationen flexibel modelliert werden kann. Daher wird das inferenzielle statistische Modell im ersten Schritt in ein einschichtiges neuronales Netz überführt, welches dann mit einer neuen Konstruktions- und Trainingsmethode um verdeckte Neuronen und Verbindungen derart erweitert wird, dass nur diejenigen Abhängigkeiten abgebildet werden können, die erforderlich bzw. in den Daten vorhanden sind. Das Verfahren ermöglicht weiterhin eine Interpretation der Verbindungsgewichte und Neuronen und wirkt damit dem typischen „*Black-Box*“-Charakter neuronaler Netze entgegen. Als Vergleich wird ein Standard-MLP-Netz herangezogen.

Die resultierenden Diagnostik-Modelle werden in drei Stufen, bestehend aus einer Konzeptbewertung, einer Evaluierung mit realen Falldaten und einem Prototypen zur Praxiserprobung am Fahrzeug evaluiert. Hierfür stehen u.a. Daten aus zahlreichen realen Reparaturfällen zur Verfügung. Die Ergebnisse zeigen, dass insbesondere mit dem in dieser Arbeit entwickelten, komplexitätserweiterten neuronalen Modell trotz der mangelnden Qualität und Anzahl der im Projekt zur Verfügung stehenden Reparaturfalldaten Ergebnisse erzielt werden, die die Erwartungen übertreffen. Die Arbeit wird mit einer Diskussion notwendiger Schritte zur Integration bzw. Migration des Verfahrens in heute eingesetzte etablierte Expertensysteme abgeschlossen.

Abstract

Well-established, expert-based diagnostics systems become more and more inadequate to cope with the steadily growing networked functions, increasing variability, and shortened development cycles of automotive systems. This leads to higher costs due to longer error identification times, an increasing number of erroneously replaced parts and high demands on customer service representatives. Novel and innovative diagnostics approaches are required in order to keep future systems efficiently maintainable and to handle the growing complexity.

Although car repair shops have recorded data of repair cases extensively for several years now, this data has not been used for automated diagnostics approaches so far. In a cooperation project of the Volkswagen AG and the TU Braunschweig, a system for Offboard Diagnostics was developed, that is able to auto-matically learn diagnostics models from previously recorded repair cases data. Considering error symptoms and vehicle data of a defective vehicle in a car repair shop, these models return hypotheses about defective components and appropriate repair actions. The repair results are recorded in turn, thus yielding an iterative solution process for automatically improving the models. The different kinds and the varying quality of available repair case data pose a major challenge. In particular, the duality between symptom information and context information, the high dimensionality of the feature space, the sparseness and the low quality of the data necessitate novel modeling and learning approaches. These approaches are derived in this work.

First, we analyze descriptive and inference-based statistical diagnostics models. The inference-based model is transferred into a single layer neural network. In contrast to inference-based statistical model, neural networks are able to model conditional dependencies as their complexity is adjustable in a flexible way. In the first step, a new construction and training technique is used to add hidden neurons and connections to the network. Thereby, only dependencies are modeled which are either necessary or exist in the real data. In addition, our approach allows an interpretation of connection weights and neurons, thus reducing the black-box nature of neural networks. A comparison to standard multilayer networks is given.

The resulting diagnostics models are evaluated in three ways: by evaluation of the concept, by an evaluation of the performance using real repair case data and by the implementation of a prototype for practical application to a vehicle. In spite of the low quality data available in this project, the performance of our complexity-enhanced neural model developed significantly exceeds expectations.

Finally, we propose a step-by-step process for the integration and/or migration of existent expert-based systems towards the introduced approach.

Danksagung

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter im Institut für Regelungstechnik an der Technischen Universität Braunschweig. Ich möchte mich an dieser Stelle herzlich bei Prof. Thomas Form bedanken, für die fortwährende Unterstützung, Förderung und das in mich gesetzte Vertrauen – auch über die Institutszeit hinaus.

Prof. Hans-Christian Reuss danke ich herzlich für seine Bereitschaft, die vorliegende Arbeit als Zweitgutachter zu beurteilen und sich durch manche längliche Formel zu quälen. Ebenso danke ich Prof. Markus Maurer für die freundliche Leitung des Promotionsverfahrens.

Mein inniger Dank geht an alle, die mich während der Entstehung der Arbeit unterstützt haben. Das gilt besonders für Andreas Breuer und Olaf Krieger, die mit den zahlreichen offenen und produktiven Diskussionen in so mancher stagnierender Phase die rettenden Ideen hervorbrachten. Ein besonderer Dank geht auch an meine Praktikanten und Diplomanden Phi Le, Andreas Mette, Daniel Wittling, Henning Dürkop und Phillip Kluge, die mit Ihrer Begeisterung und Ihrem Engagement wesentlich zur Entstehung der umfangreichen Implementierung beigetragen haben. Besonders möchte ich mich auch bei Anita, Falko, Jörn Marten, Malte, Monika und Peter für die manchmal sicherlich anstrengende Arbeit des Korrekturlesens und dem Ausmerzen so mancher schwäbischer grammatischer Eigenheiten bedanken.

Mein größter Dank schließlich gilt meinen Eltern, die mich stets in jeder Hinsicht unterstützt und gefördert haben und mir die Freiheit gegeben haben, meinen Weg zu gehen.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Herausforderung der Fahrzeugdiagnostik	1
1.2. Ausgangssituation der Arbeit	1
1.3. Lernen von diagnostischem Wissen	3
1.4. Gliederung	5
I. Theorie	7
2. Diagnostik	8
2.1. Einführung und Begriffsdefinition	8
2.2. Der diagnostische Prozess	11
2.2.1. Gegenstand der Diagnostik	11
2.2.2. Diagnostiksessions	12
2.2.3. Beobachtung von Systemen	12
2.2.4. Diagnostisches Schließen	15
2.2.5. Evaluation, Entscheidung und Reaktion auf Diagnosen . . .	20
2.2.6. Überlagerte diagnostische Prozesse	23
2.3. Diagnostik und Wissen	24
2.3.1. Diagnostik und wissensbasierte Systeme	24
2.3.2. Wissensrepräsentation und -Entstehung	25
2.3.3. Faktoren der Lösungsmethoden	26
2.4. Diagnostik im Kraftfahrzeug-Bereich	28
2.4.1. Anforderungen und Nutzer der Fahrzeugdiagnostik	30
2.4.2. Onboard-Diagnostik	31
2.4.3. Offboard-Diagnostik	34
2.4.4. Die Geführte Fehlersuche	36
3. Herausforderung der heutigen Fahrzeugdiagnostik	38
3.1. Beherrschung der Komplexität	38
3.2. Stand der Technik und relevante Arbeiten	42
3.2.1. Wissensmodellierung durch Experten	43
3.2.2. Automatische Wissensakquisition	56
3.3. Fokussierung und resultierende Forschungsaufgabe	59
4. Lernen aus Reparaturfällen	62
4.1. Einführung	62
4.2. Generalisierter Reparaturprozess	66

5. Analyse und Aufbereitung der Falldaten	69
5.1. Akquise und Analyse der Falldaten	70
5.2. Relevanz und Auswahl von Attributen	70
5.3. Wahrscheinlichkeitstheoretische Betrachtung von Merkmalen	74
5.4. Häufigkeitsverteilungen und Charakterisierung	75
6. Statistische Diagnostik-Modelle	80
6.1. Deskriptives Diagnostik-Modell	80
6.2. Der Bayes'sche Wahrscheinlichkeitsbegriff	83
6.3. Inferentielles Diagnostik-Modell	85
7. Neuronale Diagnostik-Modelle	90
7.1. Eine kleine Einführung in neuronale Netze	90
7.1.1. Architektur neuronaler Netze	91
7.1.2. Künstliche Neuronen	93
7.1.3. Berechnung neuronaler Netze	96
7.1.4. Training neuronaler Netze	97
7.2. Einschichtige neuronale Diagnostik-Modelle	103
7.2.1. Wahrscheinlichkeitstheoretische Interpretation neuronaler Diagnostik-Modelle	103
7.2.2. Interpretation der Verbindungsgewichte	108
7.3. Multi-Layer-Perceptrons als Diagnostik-Modelle	111
7.3.1. Unabhängigkeitsbetrachtung des Fahrzeugkontexts	111
7.3.2. Eine Modelldefinition	112
7.3.3. Eigenschaftsanalyse des MLP-Diagnostik-Modells	114
7.4. Komplexitätserweiterte neuronale Diagnostik-Modelle	118
7.4.1. Lösungsansätze zur gezielten Komplexitätserweiterung	118
7.4.2. Verfahren zur gezielten Komplexitätserweiterung neuronaler Netze	121
7.5. Adaptivität und Verlernen von altem Wissen	128
7.6. Beschleunigung durch Merkmals-Reduktion	132
7.7. Pruning von Diagnostik-Modellen	139
II. Evaluierung	145
8. Prototypische Umsetzung	146
8.1. Die Trainings- und Validierungsumgebung Neucus	146
8.2. Die Anwendungssoftware Neucus-Desktop	151
9. Evaluierung von Diagnostikverfahren	154
9.1. Entwicklungsbegleitende dreistufige Evaluierung	155
9.2. Konzeptbewertung	156
9.3. Evaluierung mit Reparaturfalldaten	159
9.3.1. Kriterien und praktische Umsetzung	159
9.3.2. Definition und Berechnung der Evaluierungsquote	160
9.3.3. Partitionierung der Evaluierungs- und Trainingsdaten	163

9.3.4.	Evaluierung der neuronalen Diagnostik-Modelle	163
9.3.5.	Evaluierung des Multi-Layer-Perceptron als Diagnostik-Modell	166
9.3.6.	Evaluierung des erweiterten neuronalen Diagnostik-Modells .	167
9.3.7.	Verlernen alter Reparaturfälle	170
9.3.8.	Evaluierung der Merkmals-Reduktion	171
9.3.9.	Effektivität von Pruning	174
9.3.10.	Zusammenfassung der Evaluierung mit Falldaten	176
9.4.	Praxiserprobung	178
9.4.1.	Manuelle Evaluierung der Leistungsfähigkeit	179
9.4.2.	Untersuchung exemplarischer, relevanter Fälle	181
9.4.3.	Praxiserprobung am Fahrzeug	183
III.	Abschluss	195
10.	Bewertung und Diskussion	196
10.1.	Ausblick	200
10.1.1.	Optimierung der Diagnostik-Modelle	200
10.1.2.	Integration von A-priori-Wissen in neuronale Modelle	202
10.1.3.	Interaktive Diagnostik mittels Bayes'scher Netze	203
10.2.	Integrationsmöglichkeiten	204
10.2.1.	Kombination verschiedener Diagnoseverfahren	204
10.2.2.	Kombination von Diagnostik-Modellen mit der GFS	206
11.	Zusammenfassung	209
Anhang		213
A.	Anforderungen und Analysen	214
A.1.	Allgemeine Anforderungen und Nutzer der Fahrzeugdiagnostik . . .	214
A.2.	Anforderungen an System und Konzept	216
A.3.	Zusätzliche Anforderungen an die Datenbasis	217
A.4.	Analyse diverser datenerzeugender Prozesse	219
B.	Zusatzinformationen zur Akquise und Aufbereitung der Falldaten	221
B.1.	Akquise realer Falldaten	221
B.2.	Attributbeschreibungen	222
B.3.	Kodierung von Merkmalen	224
B.3.1.	Nicht-dichotome nominale Merkmale	224
B.3.2.	Hierarchische Merkmale	225
B.4.	Analyse von drei repräsentativen Fallbeispielen	226
C.	Zusatzinformationen zur Evaluierung	232
C.1.	Diskussion und Begründung der Konzeptbewertung	232
C.1.1.	Anwendungsfälle und Einsatzbereich	232
C.1.2.	Technische Machbarkeit	233

C.1.3. Kosten	234
C.1.4. Leistungsfähigkeit	235
C.1.5. Anwendbarkeit und Nutzerakzeptanz	236
C.2. Weitere Evaluierungsergebnisse der neuronalen	
Diagnostik-Modelle	239
C.2.1. Bestimmung der Trainingsparameter	239
C.2.2. Varianz der Ergebnisse	240
C.2.3. Abhängigkeit der Symptom- und Maßnahmenmenge	
von der Fallzahl	240
C.3. Evaluierungsergebnisse exemplarischer Fälle	244
C.3.1. Fall 1: Tiguan – GRA	244
C.3.2. Fall 2: Tiguan – Funktionseinschränkung durch	
Unterspannung	245
C.3.3. Fall 3: Tiguan – K115 Innenraum	246
C.3.4. Fall 4: Golf – Ladedruckregelung, Regelgrenze unterschritten	247
D. Tabellen	248
D.1. Beispielhafte Vertrauenswerte für XOR-, AND- und	
OR-ähnliche Daten	248
D.2. Diagnostikservicefunktionen	249
D.3. Filterkriterien	249
D.4. Kodierungstabellen Symptome, Reparaturmaßnahmen, Fahrzeugkon-	
text	251
D.4.1. Symptome und Fahrzeugkontext	251
D.4.2. Reparaturmaßnahmen	251
E. Patente	253
F. Ausgewählte Publikationen	254
Tabellenverzeichnis	256
Abbildungsverzeichnis	258
Index Symbole	262
Literatur	264

1 Einleitung

Zur Einleitung führt dieses Kapitel kurz in die für diese Arbeit zentrale Herausforderung der Fahrzeugdiagnostik ein, um anschließend den Kontext der Arbeit zu umreißen. Begründend darauf wird die Forschungsaufgabe skizziert und der Aufbau der Arbeit vorgestellt.

1.1. Herausforderung der Fahrzeugdiagnostik

Eine große Herausforderung bei der Entwicklung neuer Fahrzeuge besteht darin, trotz steigender Komplexität die Zuverlässigkeit, Diagnostizierbarkeit und Wartbarkeit weiter zu verbessern. Betroffen davon sind insbesondere die Servicewerkstätten. In hoch vernetzten und verteilten Systemen, zu denen Kraftfahrzeug-Systeme zählen, können die Ursachen für viele Fehler nur mit Hilfe durchgängiger und systemübergreifender Diagnostikkonzepte zuverlässig identifiziert werden. Die unterschiedlichen im Fahrzeug verbauten Systeme besitzen viele Berührungspunkte untereinander. Dazu gehören vor allem die funktionalen Abhängigkeiten der einzelnen Systeme untereinander und die damit notwendige Vernetzung der Steuergeräte [ULPR05]. Die Auswirkungen von Störungen in solchen verteilten Systemen sind nur schwer zu überschauen.

Wartbarkeit von
vernetzten und
verteilten Systemen

Die Folge ist, dass nach [Abu03] eine in der Werkstatt durchgeführte Fehlerdiagnostik in über 60 Prozent der Fälle den Fehler nicht findet. Hierdurch steigen die Fehlersuchzeiten und die Fehltauskosten, die in Gewährleistungsfällen direkt auch den Fahrzeughersteller treffen. Zusätzlich wird die Diagnostik durch die Tatsache erschwert, dass ein Symptom nicht selten mehrere Ursachen hat, und die Ursachen durch die Symptomatik nur schwer differenziert werden können (vgl. [HDN06], [Rum06]).

Fehlersuchzeiten
und -Kosten in kom-
plexen Systemen

1.2. Ausgangssituation der Arbeit

Die meisten heute eingesetzten Steuergeräte können sich selbst diagnostizieren. Das bedeutet, sie besitzen Diagnostikroutinen innerhalb der Steuergerätesoftware, welche die Sensoren und Aktuatoren gegeneinander plausibilisieren. Überwiegend kommen hier modellbasierte Diagnostikverfahren zum Einsatz [Ise06]. Wird vom System eine Störung erkannt,

Onboard-Diagnostik

so wird ein Ereignisspeichereintrag¹²⁾ hinterlegt, der über eine Diagnostikchnittstelle³ aus dem Fahrzeug ausgelesen werden kann [ISO06].

Offboard-Diagnostik

Im Gegensatz zur Onboard-Diagnostik versteht man unter Offboard-Diagnostik-Systemen eine Klasse von Systemen, die in der Werkstatt an ein Fahrzeug angeschlossen werden und von außen auf die Systeme im Fahrzeug zugreifen. Sie haben die Möglichkeit, Ereignisspeichereinträge aller vorhandenen Fahrzeugsysteme auszulesen und dem Kundendienstmitarbeiter⁴ (KD-Mitarbeiter) in lesbarer Form anzuzeigen.

Geführte Fehlersuche

Leistungsfähige Diagnostiksysteme sollen dem KD-Mitarbeiter nicht nur alle Informationen aus den einzelnen Steuergeräten zur Verfügung stellen, sondern ihn zusätzlich aktiv bei der Fehlersuche unterstützen. Ein Beispiel hierfür ist im Kundendienst von Volkswagen die Geführte Fehlersuche (GFS). Die GFS verwendet Fehlersuchprogramme auf Basis von Entscheidungsbäumen, welche von Experten erstellt, in verschiedene Sprachen übersetzt und weltweit an die Anwender verteilt werden. Hier wird für jeden Ereignisspeichereintrag, der in einem Fahrzeug auftreten kann, ein Fehlersuchprogramm erstellt. Eine besondere Schwierigkeit besteht hierbei in der Abdeckung aller Varianten eines Fahrzeuges, welche sich durch die zahlreichen Ausstattungsmöglichkeiten ergeben. Dies ist bei der Erstellung der Fehlersuchprogramme zu berücksichtigen. So äußert sich derselbe Fehler in den unterschiedlichen Fahrzeugvarianten jeweils in anderer Art, und auch Prüf- und Reparaturanweisungen können stark variieren. Bereits im Voraus alle möglichen Fehlerquellen zu erkennen und abzudecken, ist eine weitere Schwachstelle solcher manuell erstellter Fehlersuchprogramme.

Herausforderung: Variantenvielfalt und Fehlerabdeckung

Komplexität nur mit großem Aufwand beherrschbar

Die genannten Punkte führen in der Praxis zwangsläufig zu Kompromissen hinsichtlich Qualität und Abdeckung, da das Berücksichtigen aller möglichen Fehlerquellen sowie aller Fahrzeugvarianten nur mit erheblichem Aufwand beherrschbar ist. Insbesondere der bereits genannte Anstieg an Komplexität als auch die Tendenz zu verteilten Funktionen führen damit vermehrt zu reduzierten und unspezifischeren Fehlersuchprogrammen (z.B. „*Suche nach Stromlaufplan*“).

Ein erfolgreich eingeführter wichtiger Schritt zur Verbesserung der Diagnostikqualität ist die Möglichkeit, die Reparaturfälle in den Werkstätten auszuwerten. So werden heute eine ganze Reihe Reparaturfall-relevanter Daten wie Ereignisspeichereinträge usw. in einer zentralen Datenbank aufgezeichnet. Mit den verfügbaren Informationen werden u.a. Auswertungen zur Verbesserung der Prüfabläufe durchgeführt. Dieser Vorgang

Feedback bisher manuell

¹ Früher wurde von *Fehlerspeichereinträgen* gesprochen. Allerdings bedeutet ein solcher Eintrag nicht zwangsläufig die Existenz eines Fehlers oder eines Defektes, so dass der Begriff Ereignisspeichereintrag eingeführt wurde.

² engl. DTC - Diagnostic Trouble Code

³ Im Kraftfahrzeug-Bereich ist der Terminus *Diagnose* an Stelle von *Diagnostik* weit verbreitet. Hier wird jedoch konsequent der Begriff *Diagnostik* verwendet. Eine Begriffsdefinition und -abgrenzung wird in Kapitel 2.1 gegeben.

⁴ Hiermit sind alle für die Fahrzeugreparatur beteiligten Mitarbeiter (KFZ-Mechatroniker, KFZ-Servicetechniker, usw.) in den Servicewerkstätten gemeint.

der Auswertung, wie auch die anschließende Verbesserung der Fehlersuchprogramme wird jedoch manuell vorgenommen und lässt sich nicht mit vertretbarem Aufwand automatisieren. Daher kann über diese Vorgehensweise nur eine geringe Anzahl an Prüfprogrammen verbessert werden.

1.3. Lernen von diagnostischem Wissen

Weltweit existieren für jeden großen Automobilhersteller mehrere zehntausend KFZ-Werkstätten mit insgesamt über hunderttausend Mitarbeitern⁵. Der Austausch an Erfahrung zwischen den einzelnen Werkstätten ist nicht nur eine große Chance sondern auch eine Herausforderung. Das Potential von kollektivem Erfahrungsaustausch soll mit einem Beispiel veranschaulicht werden: Zur Lösung eines komplexen Reparaturfalls muss ein KD-Mitarbeiter A einiges an Fehlersuchzeit aufwenden und weiterhin einige Komponenten auf Verdacht tauschen. Es handelt sich dabei nicht um einen Einzelfall, sondern der Fall tritt in einer anderen Werkstatt erneut auf. Dann muss der dortige KD-Mitarbeiter B mit demselben Problem denselben Aufwand investieren, wenn KD-Mitarbeiter B nicht von der Erfahrung des KD-Mitarbeiters A profitiert. Wäre es möglich, die einmal gewonnenen Erfahrungen allen Servicewerkstätten zur Verfügung zu stellen, bestünde ein großes Einsparpotenzial hinsichtlich Fehlersuchzeit und -kosten, für den Kunden wie auch für den Fahrzeughersteller. Das Ziel sollte daher sein, aus Reparaturfällen zu lernen und das erlangte Wissen für zukünftige Reparaturfälle zu nutzen. Da auch hier analog zur GFS die Menge der Daten und das daraus erlangbare Wissen groß ist, kann sowohl das Lernen als auch die Anwendung nur dann von gewinnbringender Effizienz sein, wenn der gesamte Prozess automatisch durchführbar ist. Gleichzeitig müssen die Algorithmen jedoch robust gegenüber fehlerhaften Daten bzw. Reparaturfällen sein, um Missbrauch zu verhindern und eine hohe Qualität zu gewährleisten.

Eine große Herausforderung für ein Lernsystem ist dabei die Art und die Qualität der zur Verfügung stehenden Daten. Typischerweise müssen nicht nur Symptome und Reparaturmaßnahmen, sondern auch Fahrzeuginformationen wie Modell, Ausstattung, Hardware- und Softwareversionen der Steuergeräte, etc. ausgewertet werden. Gleichzeitig jedoch existieren eine große Zahl verschiedener möglicher Symptome, Reparaturmaßnahmen und Fahrzeuginformationen. Die so entstehenden großen Datensätze mit verschiedenen Informationsarten verhindern den direkten Einsatz bewährter Lernverfahren und erfordern vielmehr eine flexible Adaption und intelligente Konstruktionsverfahren.

In dieser Arbeit wird ein neues Offboard-Diagnostikverfahren entwickelt, welches zunächst in einem automatisierten Prozess aus in Servicewerkstätten aufgezeichneten Reparaturfällen Diagnostikmodelle lernt.

Große Fallbasen
verfügbar

Ziel – Lernen aus
Reparaturfällen

Herausforderung:
Datenqualität
und -art

Forschungsaufgabe

⁵ Stand 2009

Diese Modelle werden in regelmäßigen Abständen als Update auf die Diagnostiktester verteilt und sind dort in der Lage, anhand den von einem defekten Fahrzeug eingelesenen Fehlersymptomen und Fahrzeuginformationen auf schadhafte Komponenten und hilfreiche Reparaturmaßnahmen zu schließen. Das Ergebnis der Reparatur wird wiederum aufgezeichnet und an ein zentrales Data Warehouse übermittelt. Hierdurch entsteht ein iterativer Feedback- und Lösungsprozess, der zu einer automatischen kontinuierlichen Verbesserung der Modelle führt.

Die Art und die Qualität der zur Verfügung stehenden Reparaturfalldaten weicht stark von den bekannten „*Standardproblemen*“ ab. Zu nennen sind hier vor allem die Dualität zwischen Symptominformationen und Kontextinformationen, die hohe Merkmalsdimension, hohe Sparsity und mangelnde Qualität, die damit auch den Einsatz von bekannten Modellen und Lernverfahren verhindern.

Zu den Forschungsaufgaben dieser Arbeit zählt neben der Konzeptentwicklung zunächst eine effiziente Datenvorverarbeitung wie Filter, Merkmalsextraktion und -Verdichtung. Für die letzteren beiden Punkte kann aufgrund der Randbedingungen nicht auf bekannte Methoden zurückgegriffen werden und es sind neue spezialisierte Verfahren notwendig. Darauf aufbauend müssen robuste und leistungsfähige Diagnostik-Modelle entwickelt werden, die insbesondere die Dualität in Sparsity und inhärenter Komplexität der Daten geeignet modellieren. Angepasste Konstruktions- und Lernverfahren müssen dafür sorgen, dass der „*Black-Box*“-Charakter der hier verwendeten neuronalen Netze umgangen wird und die Verbindungsgewichte interpretierbar bleiben, um somit u. a. wirksame Absicherungsmaßnahmen zu ermöglichen oder die Modelle zum weiteren Informationsgewinn verwenden zu können.

Die Analyse zum Stand der Technik zeigt, dass hier in vielen Bereichen Neuland betreten wird. Hierbei soll speziell der Anspruch hervorgehoben werden, neben einem auch in der Praxis funktionierenden Prototypen eine durchgehend mathematische Herleitung der Verfahren entwickeln zu wollen, die erst eine Interpretation der Verbindungsgewichte bzw. Parameter der Modelle ermöglicht. An manchen Stellen musste dabei auch ein Spagat zwischen theoretischen Algorithmen und praktischen Heuristiken gewagt werden.

1.4. Gliederung

Die Arbeit ist in drei Teile und 11 Kapitel strukturiert. Die Kapitel 2 – 7 bilden dabei den theoretischen Teil, der sich von der Idee über notwendige Grundlagen zur Lösung der Forschungsaufgabe erstreckt. Die akademischen Fragestellungen und entsprechenden Antworten dieser Arbeit werden hier vollständig zusammengestellt und können als eigenständige Einheit betrachtet werden.

Abbildung 1.1.: Gliederung

Der zweite Teil beschäftigt sich mit der Realisierung der Verfahren und ihrer Evaluierung, bevor im dritten Teil eine abschließende Bewertung und Diskussion stattfindet. Die nachfolgende Abbildung veranschaulicht die resultierende Gliederung.

Nach der Einleitung wird die Arbeit durch eine Einführung in die Grundlagen der technischen Diagnostik eröffnet. Dabei wird insbesondere auf den Begriff der Diagnostik und des diagnostischen Prozesses eingegangen.

Aufbauend darauf gibt Kapitel 3 eine Übersicht und Bewertung zum Stand der Technik. Bereits hier wird mit der Bewertung der Vorteil lernender Verfahren herausgearbeitet und eine Fokussierung hierauf formuliert. Diese beinhaltet auch eine Abgrenzung zu verwandten Arbeiten.

Kapitel 4 skizziert gemäß der Fokussierung konsequent das Konzept zum Lernen aus Reparaturfällen und erörtert die resultierenden Lösungs- und Feedbackprozesse. Für jedes lernende System ist die Charakterisierung und Aufbereitung der zur Verfügung stehenden Daten ein bedeutender Schritt, der in Kapitel 5 beschrieben wird. Aufbauend darauf werden verschiedene Diagnostik-Modelle entwickelt und bewertet. Hierzu diskutiert Kapitel 6 zunächst statistische

Modelle, die später in Kapitel 7 zu diversen neuronalen Modellen überführt und erweitert werden.

Kapitel 8 stellt eine prototypische Umsetzung des Konzepts vor, welche auch den praktischen Einsatz am Fahrzeug erlaubt. In Kapitel 9 wird eine umfassende 3-stufige Evaluierung, beginnend mit einer Konzeptbewertung, der Bewertung der Modelle und verschiedener Modellparameter hin bis zur praktischen Erprobung am Fahrzeug, beschrieben.

Im letzten Teil wird die Arbeit mit einer abschließenden Bewertung und Diskussion in Kapitel 10 sowie der Vorstellung verschiedener weitergehender Ansätze inklusive einem Migrations- und Integrationskonzept in Kapitel 11 abgeschlossen. Am Ende der Dissertationsschrift finden sich nötige Verzeichnisse und Anhänge.

TEIL I: THEORIE

Der theoretische Teil dieser Dissertation stellt in den folgenden Kapiteln zunächst die Motivation der durchgeführten Arbeiten dar, erfasst die zur Bearbeitung nötigen Grundlagen und stellt weltweit herausragende Arbeiten im Forschungsbereich vor. Darauf aufbauend wird sukzessive die Lösung der Forschungsaufgabe analytisch entwickelt und diskutiert.

2 Diagnostik

Diese Arbeit konstituiert sich aus wesentlichen Bestandteilen der Diagnostik und des maschinellen Lernens, die eine ausführliche Behandlung der Gebiete notwendig macht. In Ermangelung einer kongruenten Definition des Begriffs *Diagnostik* und *diagnostischer Prozess* wird zunächst eine allgemeine Definitionen gegeben und darauf aufbauend ein Prozessmodell definiert. Anschließend werden Aspekte der Wissensrepräsentation und -entstehung sowie wichtige Faktoren der Lösungsmethode aufgezeigt.

Die folgende Spezialisierung auf den Kraftfahrzeug-Bereich zeigt zunächst Anforderungen der Forschung, Entwicklung, Produktion und des Service auf, die in alle weiteren Betrachtungen einfließen. Allgemein wird eine Übersicht über Onboard- und Offboard-Verfahren sowie Expertensysteme wie die Geführte Fehlersuche mit Überlegungen zur Fehlerfortpflanzung gegeben.

2.1. Einführung und Begriffsdefinition

Diagnostik vs.
Diagnose

Die **Diagnostik** (griechisch: *diágnossi*: *diá* = „(hin-)durch“, *gnóssi* = „die Erkenntnis“, „das Urteil“) ist allgemein der Prozess der Identifikation des Zustands eines Objekts auf Basis von Beobachtungen über das Objekt selbst. Am Beispiel der Fehlerdiagnostik umfasst das den Prozess der Identifikation der Fehlerursache eines Systems auf Basis von Messwerten, Prüfungen oder allgemein Symptomen. Das Ergebnis der Diagnostik ist eine **Diagnose**, meint allgemein den interessierenden Zustand des Objekts oder konkret eine Fehlerursache, Krankheit, etc. Viele Beobachtungen sind nur durch vorhergehende Stimulationen bzw. Aktionen (z. B. die Verwendung eines Stethoskops oder Messgeräts) möglich, so dass sich die Diagnostik häufig als iterativer Prozess konstituiert, bis eine hinreichend präzise Diagnose erreicht wurde. Mit dem Beispiel der Fehlerdiagnostik zählt somit nicht nur die Bereitstellung von Mess-Aktionen, sondern auch Reparatur-Aktionen (d. h. Aktionen allgemein) zur Diagnostik, wenn das Ermitteln der *Nulldiagnose* bzw. der Fehlerfreiheit inhärent ist.

Diagnostik & Diagnose – eine Begriffsdefinition

Keine einheitliche
Begriffsdefinition

Auf der Suche nach der Bedeutung der Begriffe *Diagnostik* und *Diagnose* fällt auf, dass (nicht nur) in der technischen Domäne einer genauen Definition oft aus dem Weg gegangen wird und anstatt dessen je nach

Domäne und Forscher unscharfe, unvollständige und teilweise widersprüchliche Erläuterungen gegeben werden. Darum soll im Folgenden eine für diese Arbeit geltende Definition erarbeitet werden. Abbildung 2.1 zeigt eine Aufstellung einiger Definitionen des Begriffs.

Diagnostik & Diagnose

Griechisch: *diágnossi*: *diá* = „(hin-)durch“, *gnóssi* = „die Erkenntnis“, „das Urteil“)

1. (Diagnose) Im Brockhaus nachgeschlagen lautet die Auskunft zum Eintrag: „(Medizin) vorgenommene Zuordnung der gefundenen Symptome zu einem Krankheitsbegriff; (allg.) das Feststellen, Prüfen und Klassifizieren von Merkmalen mit dem Ziel der Einordnung zur Gewinnung eines Gesamtbildes“, [Bro06, Bd. 6; S. 758]
2. (Diagnostik) Hier findet sich „(Medizin) alle zur Erkennung einer Krankheit getroffenen Maßnahmen.“, [Bro06, Bd. 6; S. 758]
3. (Diagnose) „Bestimmung und Klassifizierung von Merkmalen mit dem Ziel der Einordnung eines (Natur-)wissenschaftlichen Betrachtungsgegenstandes [...]“ und „[...] als Bezeichnung für die Tätigkeit des Arztes sowie deren Ergebnis für das Erkennen, Feststellen, Einordnen, [...] aus bestimmten Symptomen und mit bestimmten Hilfsmitteln [...]“, Deutsches Fremdwörterbuch [SSB99, Bd. 4; S. 479ff]
4. (Diagnostik) „Wissenschaft, Lehre von der Erkennung und Benennung von Krankheiten. [...] Gesamtheit der zur Krankheitserkennung getroffenen Maßnahmen“, [SSB99, Bd. 4; S. 478ff]
5. (Diagnose) „*Diagnosis is the task of identifying which component of a device are not functioning according to their behavioral specifications*“, [FK93, S. 619]
6. (Diagnose) „*The task of assigning a given input (e.g., a pattern, image, set of observations, etc.) to some category (or class)*“, [BH98, S. 1]
7. (Diagnostik) „[...] soll die Diagnostik als die Fähigkeit verstanden werden, durch Beobachtung und Untersuchung eine Beurteilung über den Zustand eines technischen Systems geben zu können“, [Mic02, S. 30]

Abbildung 2.1.: Zusammenstellung einiger Definitionen für den Begriff *Diagnostik* und *Diagnose*

Die begründende Disziplin der Diagnostik war die Medizin. Entsprechend existieren in dieser Domäne allgemein akzeptierte, gut abgegrenzte Terminologien. In [Wes72] beispielsweise werden für die Domäne der Psychodiagnostik (mathematisch beschriebene) Regeln definiert, nach denen

Begriffsdefinition
in der Medizin

ein diagnostischer Prozess abzulaufen hat. Die beschriebenen Regeln und Inferenzverfahren¹ haben starke Ähnlichkeit mit den z.B. in [Rei92] oder [KMR90] beschriebenen modellbasierten Diagnostikverfahren. Eine *Diagnose* ist nach [Wes72, S. 25] letztlich eine Hypothese zur Erklärung der Beobachtungen und damit ein Ergebnis des diagnostischen Prozesses². Die Definitionen des Brockhaus oder des deutschen Fremdwörterbuchs (siehe Abbildung 2.1) bestätigen diese Sichtweise, wenngleich sie darauf hinweisen, dass teilweise unter Diagnose auch der Prozess selbst, also die Diagnostik verstanden wird. In manchen Publikationen wie z.B. [FK93, S. 620], [KMR90] oder [BH98], die von namhaften Forschern der Domäne verfasst wurden, wird Diagnostik und Diagnose auch als Synonym verwendet³. Mit Begriffen wie *Diagnose-System*, *Diagnose-Verfahren* oder *Onboard-Diagnose* wird deutlich, dass sich in der technischen Domäne der Begriff Diagnose als Sammelbegriff für alles damit zusammenhängende, also auch der Diagnostik, eingebürgert hat.

Widersprüchliche
Verwendung
von Diagnos-
tik & Diagnose

Dies ist in sofern unglücklich, wie der Begriff *Diagnose* damit von den im medizinischen Bereich existierenden Definitionen abweicht, weiterhin immer eine kontextabhängige unter Umständen nicht eindeutige Interpretation zwischen dem Prozess (dem Diagnostizieren) oder dem Ergebnis (der Diagnose) notwendig ist und damit Missverständnisse und Verständnisschwierigkeiten zwischen den Domänen und letztlich Ungenauigkeiten provoziert werden⁴⁵. Eine positive Ausnahme ist jedoch die renommierte Arbeitsgruppe um Frank Puppe, die in ihren Büchern und Publikationen die Begriffe konsequent verwendet (vgl. z. B. [Pup91, ABP06, BSP04])⁶. Auch in dieser Arbeit werden die Begriffe *Diagnostik* und *Diagnose* gemäß ihren Bedeutungen konsequent verwendet, um eine eindeutige Terminologie zu prägen.

Hier verwendete
Terminologie

Eine aus Sicht des Autors treffende Definition der Diagnostik, die auch für diese Arbeit gelten soll, wurde in [DP99] eingeführt⁷. Die offene Formulierung wird hier als vorteilhaft angesehen und lässt die Definition so auf verschiedenste Domänen anwenden.

¹ Inferenzverfahren sind Verfahren des logischen Schließens auf Basis von Wissen.

² Die Begründung dieser Definition geht hier auf [Duc26, S. 541] zurück: „*The word which exactly designates this third sort of inference in common usage, from which there is no occasion to depart, is Diagnosis, or inference from Circumstantial Evidence.*“ (mit „*third sort of inference*“ sind Hypothesen gemeint).

³ Auszüge aus [FK93, S. 620]: „[...] *the diagnostic task.*“. Der übernächste Satz beginnt mit „*The diagnosis task [...]*“.

⁴ Nur wenige (technische) Arbeiten nehmen Bezug auf diese Terminologieproblematik. Für den Kraftfahrzeug-Bereich finden sich beispielsweise in [Mic02, S. 30ff] entsprechende Anmerkungen.

⁵ Für weiterführende Literatur wird auf [FB89], [JL04], [Büh34], [Who63] und [HH88, S.133ff.] verwiesen.

⁶ Weitere Arbeiten die speziell den Begriff *Diagnostiksystem* verwenden sind z. B. [Kub06, Har03, Str02a].

⁷ Die Definition deckt sich u. a. mit der Charakterisierung von Diagnostik in [BH98, S. 2] und [Pup91, S. 3].

Definition 2.1 (Diagnostik) *Diagnostik ist allgemein der Vorgang der Identifizierung (Klassifizierung) eines Objekts bei gegebenen Beobachtungen über das Objekt selbst in einem erwünschten Ausmaß und gegebenenfalls die Bereitstellung von verfügbaren Aktionen, die auf das Objekt angewendet werden können.*

Die verschiedenen Aspekte der Definition werden in den nachfolgenden Abschnitten näher betrachtet.

2.2. Der diagnostische Prozess

Die Diagnostik, vorhergehend als Vorgang/Prozess erklärt, kann nur in einem bestimmten umgebenden Rahmen definiert sein. Dieser kann je nach Domäne sehr unterschiedliche Ausprägungen annehmen und bestimmt das Ausmaß, die Objekte sowie die verfügbaren Aktionen. In Ermangelung einer kongruenten Definition des Terminus ist es nicht verwunderlich, dass hierfür kein einheitliches Prozessmodell, weder allgemeingültig noch domänenspezifisch (ausgenommen der Medizin), existiert. Nachfolgend wird ein solches definiert, das nicht nur als Grundlage für diese Arbeit dienen soll, sondern auch den Anspruch einer Allgemeingültigkeit erhebt.

Kontext der Diagnostik

2.2.1. Gegenstand der Diagnostik

Es scheint evident, dass der diagnostische Prozess nicht gegenstandslos existieren kann. Er bezieht sich immer auf ein System und ist selbst in einem System verortet, wobei sich sinnvolle Systemgrenzen zwischen dem diagnostizierenden und dem zu diagnostizierenden System definieren lassen sollten (vgl. Kapitel *Gegenstand und Methode* in [Wes72, S. 17]). Zum besseren Verständnis sollen die folgenden Betrachtungen mit einem typischen Anwendungsbereich der Diagnostik, der zugleich auch einer der ersten war, der Medizin, veranschaulicht werden. Als Szenario wird ein typischer Arztbesuch gewählt.

Beispiel –
der Arztbesuch

Im Beispiel ist der Gegenstand der Diagnostik der Mensch (Patient), der nach [Ber68] konsequenterweise auch als (offenes) System begriffen werden kann⁸. Weitere analytische Betrachtungen machen an dieser Stelle eine Definition des Systembegriffs notwendig. Eine allgemein anerkannte Definition, die auch hier verwendet werden soll, wird in [Kli01] gegeben:

Definition 2.2 (Offenes System) *Ein System Σ ist definiert durch ein Tupel, bestehend aus der abzählbaren Menge seiner Komponenten K_Σ und der abzählbaren Menge an Beziehungen Ξ_Σ , definiert auf K_Σ , so dass gilt*

Was sind Systeme?
Eine Definition

$$\Sigma = (K_\Sigma, \Xi_\Sigma).$$

⁸ Vergleiche [Mat02] für eine weiterführende Definition des Systembegriffs.

Jede Komponente $k \in K_\Sigma$ ist dabei ein Zustand z zugeordnet mit $z : K_\Sigma \rightarrow \mathbb{R}^n$. Eine Beziehung $\Xi \in \Xi_\Sigma$ definiert, wie die Zustände $z \in Z_\Sigma$ der Komponenten k zusammenhängen. Ein System ist offen, wenn eine zusätzlich abzählbare Menge an Beziehungen Ξ_{Σ_A} zwischen den Komponenten K_Σ des Systems Σ und einer abzählbaren Menge an Komponenten K_{Σ_A} anderer, außerhalb verorteter Systeme Σ_A existiert.

Welcher Art die Beziehungen Ξ sind, wird durch die Definition nicht festgelegt. Für mechanische Systeme kann dies z.B. eine Menge von Differentialgleichungen, für digitale Systeme Softwarestrukturen sein. Für offene Systeme gilt zusätzlich, dass Beziehungen zu Komponenten anderer Systeme Σ_A möglich sind, bezeichnet mit K_A .

2.2.2. Diagnostiksessions

Damit eine *Diagnostiksession*, im Beispiel ein Arztbesuch, begonnen wird, bedarf es einem Ereignis, das die Motivation auf eine genügende Stärke hebt, die den Aufwand zumindest subjektiv zu rechtfertigen scheint (vgl. auch [KW65]). Ereignisse können im Beispiel Befindlichkeitsstörungen oder allgemein festgestellte Abweichungen vom Wunschzustand sein (meistens „gesund“). Übertragen auf technische Domänen werden solche Abweichungen auch als Fehler⁹ bezeichnet. Auch Routineuntersuchungen, also eine regelmäßige Überprüfung und Sicherstellung des Wunschzustands (Fehlerfreiheit) sind häufig Anlässe für eine Diagnostiksession (z.B. die Prozessdiagnostik in der Automatisierungstechnik). Das treibende Ereignis könnte hier dann der Ablauf einer bestimmten Zeitspanne sein.

Anlässe sind Fehler oder Routineüberprüfungen

Definition 2.3 (Ereignisse für Diagnostiksessions) *Eine Diagnostiksession wird ausschließlich durch die Existenz eines Ereignisses angestoßen, das den Aufwand der Diagnostiksession selbst rechtfertigt. Ereignisse können beispielsweise das Auftreten von Indizien für einen Fehler bzw. die Abweichung vom Sollzustand ($Z_{ist} \neq Z_{soll}$ für $Z \in \{z_1, z_2, \dots, z_n\}$) oder die Sicherstellung der Fehlerfreiheit bzw. des Sollzustands ($Z_{ist} \stackrel{?}{=} Z_{soll}$) durch regelmäßige Überprüfung sein.*

2.2.3. Beobachtung von Systemen

Typischerweise klagt der Patient zunächst sein Leiden, oftmals in der Form, dass Wahrnehmungen über Störungen/Unannehmlichkeiten formuliert werden. Der Arzt nimmt diese auf und versucht sie zu ihm bekannten Symptomen zuzuordnen. In der Medizin existiert hierfür eine

⁹ Ein Fehler ist nach DIN 55350 eine unzulässige Abweichung eines Merkmals von einer vorgegebenen Forderung [DIN08].

eigene formale (Fach-)Terminologie¹⁰. Gegebenenfalls werden einzelne Symptomzuordnungen am Patienten überprüft, wenn hierüber Unsicherheit besteht. Die Summe aller bekannten Symptome wird dann als *Symptomatik* bezeichnet.

Symptomatik –
Menge aller bekannten Symptome

Dieser erste Schritt kann im Sinne der Definition 2.1 allgemein als *Beobachtung* bezeichnet werden¹¹. Viele Zustände eines Systems sind jedoch nicht direkt *beobachtbar* bzw. *messbar*. Der aktuelle Drehmomentverlauf eines Verbrennungsmotors zählt beispielsweise dazu (ein entsprechender Sensor ist zumeist nicht verbaut). Diejenigen Größen eines Systems, die jedoch beobachtbar bzw. messbar sind, werden im Folgenden *Observablen* genannt¹². Der Wert einer Observablen wird durch Messung bestimmt (beobachten durch Messen). Als Beispiele für Observablen in technischen Systeme seien hier z. B. Temperatur oder Spannung genannt. Observablen geben nun die Möglichkeit die wahren Zustände $z \in Z_\Sigma$ des Systems Σ zu schätzen. Im Beispiel des Motordrehmoments wird dieses aus Observablen wie Einspritzmenge, Luftmenge, Lambdawert, Temperatur, etc. mit Hilfe eines Modells geschätzt.

Auch wenn die Systemdefinition 2.2 die Möglichkeit der Beobachtung und damit Observablen mit einschließt, ist es für weitere Betrachtungen sinnvoll, eine Erweiterung einzuführen, die diese als Teil des Systems explizit definiert. Ein System Σ mit explizit vorhandenen Observablen $o \in \mathcal{O}_\Sigma$ wird dann definiert als ein Tripel mit

Erweiterung der
Systemdefinition um
Observablen

$$\Sigma = (K_\Sigma, \Xi_\Sigma, \mathcal{O}_\Sigma) , \quad (2.1)$$

wobei

$$\begin{aligned} K_\Sigma &= \text{Menge aller Komponenten,} \\ \Xi_\Sigma &= \text{Menge aller Beziehungen in } K_\Sigma, \\ \mathcal{O}_\Sigma &= \text{Menge aller Observablen für } \Sigma. \end{aligned}$$

Manche Messungen zur Bestimmung einer Observablen erfordern jedoch eine vorhergehende *Stimulation* des Systems. Eine ganze Reihe an Messungen sind ausschließlich durch externe Stimulationen mit vertretbarem Aufwand möglich. Hierzu zählt z. B. das Überprüfen des Patellarsehnenreflexes¹³, der nur durch einen kurzen Schlag auf die Patellarsehne

Observablen werden
durch Stimulati-
on und Messung
bestimmt

¹⁰ Anamnese genannt. Im Rahmen dieser wird die Vorgeschichte eines Patienten in Bezug auf seine aktuellen Beschwerden durch Befragung erhoben.

¹¹ Eine allgemeine Terminologie ist hier schwer zu definieren. In [Ise06, S. 17] wird dieser Schritt wahlweise „*measurements*“, „*monitoring*“ und „*feature generation*“ genannt, [PGPB96, S. 5] verwendet die Begriffe „*Merkmalsabstraktion*“ und „*Merkmalsvorverarbeitung*“. Die hier gewählte Terminologie *Beobachtung* schließt jedoch die letzt genannten Begriffe mit ein.

¹² Dieser Begriff wurde ursprünglich in der Quantenphysik durch Paul A. M. Dirac eingeführt um das Messbarkeitsproblem von Zuständen von Quantenobjekten zu beschreiben. Der Begriff wurde später zunehmend in andere Domänen übertragen (vgl. [Wun00, S. 84]).

¹³ Kniereflex

unterhalb der Kniescheibe ausgelöst werden kann. Technische Systeme erlauben teilweise die Modifizierung der externen Bedingungen, um so gezielt Stimulationen einzubringen. Ein Beispiel hierfür sind die weit verbreiteten HIL-Systeme, die eine sehr umfangreiche Stimulation und Beobachtung von Eigenschaften des Systems zulassen. Aber auch „nur“ das Abziehen eines Steckers an einem Sensor, um bestimmte Messungen durchführen zu können, kann als Stimulation betrachtet werden.

Aktionen

Allgemein kann jedoch festgestellt werden, dass jede Bestimmung einer Observablen eine Aktion darstellt, und wenn es nur die Messung selbst ist (auch eine Stimulation ist eine Aktion). Stimulationen und Messung werden im Folgenden als *Aktion* $\mathcal{A}$ zur Bestimmung einer Observablen $\mathcal{O}$ zusammengefasst. Formal kann dies durch eine surjektive Zuordnungsfunktion $Z_{\mathcal{A},\mathcal{O}}$ beschrieben werden, die jeder Aktion $\mathcal{A} \in \mathcal{A}_\Sigma$ eine oder mehrere Observablen $\mathcal{O} \subseteq \mathcal{O}_\Sigma$ zuordnet, die durch diese Aktion bestimmt werden können:

$$Z_{\mathcal{A},\mathcal{O}} : \mathcal{A}_\Sigma \twoheadrightarrow \mathcal{O}_\Sigma, \mathcal{A} \mapsto \mathcal{O}, \quad (2.2)$$

wobei

$$\mathcal{A}_\Sigma = \text{Menge aller für } \Sigma \text{ möglichen Aktionen.}$$

Informationsverdichtung zu Merkmalen

Bedingt durch z.B. alterungsbedingte Veränderungen im Fahrzeug müssen Messwerte gegebenenfalls durch Verfahren wie Anomalie- oder Fehlererkennung zu Informationen verdichtet werden, d.h. es werden zuerst mit Wissen über das System eine Reihe von Merkmalen (engl. Features) extrahiert, die eine sehr viel kompaktere Information darstellen¹⁴. Ein Beispiel hierfür ist der Klopfsensor. Durch komplexe Signalverarbeitungsalgorithmen werden die Sensordaten (Messungen) zum Merkmal „Klopfen“ verdichtet.

Es müssen folglich zuerst Abbildungsfunktionen $f_{\mathcal{O},m}$ zwischen den (zeitlich veränderlichen) Observablen des Systems $\mathcal{O} \subseteq \mathcal{O}_\Sigma$ und den Merkmalen $M \subseteq M_\Sigma$ existieren, wobei einzelne Abbildungsfunktionen komplexere Algorithmen zur Fehlererkennung oder Informationsverdichtungen enthalten können¹⁵:

$$f_{\mathcal{O},m} : \mathcal{O}_\Sigma \rightarrow M_\Sigma, \mathcal{O} \mapsto m \text{ mit } \mathcal{O} \subseteq \mathcal{O}_\Sigma, m \in M_\Sigma. \quad (2.3)$$

Aufbauend auf den Merkmalen können nun Symptome ermittelt werden. Mit dem Beispiel des Klopfensors wird deutlich, dass allein das Merkmal „Klopfen“ nicht schon ein Symptom darstellt. Erst ein Klopfen über einen längeren Zeitraum hinweg weist auf eine fehlerhafte Verbrennung

¹⁴ für eine ausführliche Behandlung von Merkmalsextraktion und Fehlererkennungsmechanismen in mechatronischen Systemen wird auf [Ise06, Teil II] verwiesen, eine Taxonomie wird in Abbildung 2.10 gegeben.

¹⁵ [Ise06, S. 17] nennt dies „feature generation“.

hin und rechtfertigt das Erzeugen eines Symptoms hierfür. Auch hier können teilweise komplexe Algorithmen zum Einsatz kommen. Formal handelt es sich hier um eine surjektive Zuordnungsfunktion $f_{M,s}$ zwischen einer Menge von Merkmalen $M \subseteq M_\Sigma$ und einem dem Diagnostiksystem bekannten Symptom $s \in S_\Sigma$ ¹⁶:

$$f_{M,s} : M_\Sigma \rightarrow S_\Sigma, M \mapsto s, \quad (2.4)$$

wobei

M_Σ = Menge aller aus $\mathcal{O}_\Sigma$ ableitbaren Merkmale und

S_Σ = Menge aller für Σ möglichen Symptome.

Damit ergibt sich nun die Zuordnungsfunktion $f_{\mathcal{O},s}$ zwischen Observablen $\mathcal{O}$ des Systems und bekannten Symptomen $S \subseteq S_\Sigma$ durch eine Verkettung von $\{f_{\mathcal{O},m}\}$ und $f_{M,s}$ mit $f_{\mathcal{O},s} := \{f_{\mathcal{O},m}\} \circ f_{M,s}$. Ein Symptom s kann dann bestimmt werden mit

$$s = f_{M,s}(\{f_{\mathcal{O},m}(\mathcal{O})\}) \quad (2.5)$$

Es ist evident, dass Observablen $\mathcal{O}$, die keinem bekannten Symptom s zugeordnet werden können, d.h. $f_{\mathcal{O},s}(s) = \emptyset$, für den Vorgang der Diagnostik nicht relevant sind.

Definition 2.4 (Beobachtung) *Jeder diagnostische Prozess enthält inhärent einen Schritt der Beobachtung. Dabei erfordert die Bestimmung der Observablen $\mathcal{O} \subseteq \mathcal{O}_\Sigma$ des zu diagnostizierenden Systems Σ eine vorhergehende Aktion $\mathcal{A} \in \mathcal{A}_\Sigma$, die mindestens eine Messung oder zusätzlich eine vorhergehende Stimulationen umfasst. Die Observablen werden mit $f_{\mathcal{O},m}$ zu Merkmalen $M \subseteq M_\Sigma$ verdichtet und mit $f_{M,s}$ den dem diagnostizierenden System bekannten Symptomen $S \subseteq S_\Sigma$ zugeordnet, so dass*

$$s = f_{M,s}(\{f_{\mathcal{O},m}(\mathcal{O})\}) = f_{\mathcal{O},s},$$

wobei $f_{\mathcal{O},s}$ die direkte Bestimmung eines Symptoms s aus den Observablen $\mathcal{O}$ zulässt.

2.2.4. Diagnostisches Schließen

Diagnostisches Schließen kann auf zwei Arten geschehen: Klassifikation und Inferenz. In [PGPB96, S. 3] wird der Begriff *Klassifikation* beispielsweise als Synonym für die Klasse *Diagnostik* betrachtet¹⁷. Dies trifft

¹⁶ Diese Zuordnung von Merkmalen nach Symptomen entspricht nach [Ise06, S. 17] der „*fault dedection*“. Jedoch kann dieser Schritt dort bereits intelligente diagnostische Funktionen enthalten, das für die hier vorgestellte Zuordnungsfunktion nicht gelten soll. Der in [Ise06] vorgestellte Schritt entspricht vielmehr einem diagnostischen Subprozess, wie er hier in Abschnitt 2.2.6 vorgestellt wird.

¹⁷ Die Begriffsverwendung zur Klassifikation ist in der Literatur leider nicht eindeutig (vgl. auch [Pup90], S.23, Fußnote). In [Ste95] bedeutet „*classification*“

Symptom-
generierung

Relevante
Observablen

Begriffsbestimmung
– *diagnostisches
Schließen*

zweifelsfrei für viele diagnostische Lösungsmethoden zu, wie auch im hier gegebenen Arztbeispiel, lässt aber Inferenzverfahren zunächst außen vor. Inferenzverfahren sind jedoch insofern mit Klassifikationsverfahren verwandt, wie die Lösung ebenfalls nur aus einem bekannten Element bestehen kann und aus diesem Element ausgewählt wird. Abbildung 2.2 veranschaulicht die beiden Kategorien. Als Oberbegriff wird darum im Folgenden der Begriff *diagnostisches Schließen* verwendet, der beide Kategorien umfasst.

Abbildung 2.2.: Methoden des diagnostischen Schließens. Rechts: Schließen durch Klassifikation. Links: Schließen durch Inferenz in einer gegebenen Systemstruktur. S : Symptome, K : Komponenten, E : Ereignisse

Warum geht es dem Patienten schlecht?

Um zum begleitenden Beispiel zurückzukommen, ermittelt der Arzt auf Basis der ihm bekannten Symptome und des Wissens über den konkreten Patienten (spezielles Wissen über den Patienten, beispielsweise die Krankenakte/Patientenhistorie) sowie des medizinischen Wissens (allgemeines Wissen über das System Mensch) mögliche oder besser wahrscheinliche Ursachen (Diagnosen), die zu den bekannten Symptomen führen können. Oftmals steht nach dieser ersten Ursachenermittlung noch nicht die eindeutige Diagnose fest. Vielmehr sind es eine Reihe von mehr oder weniger wahrscheinlichen Diagnosen, später Diagnose-Hypothesen genannt¹⁸. Der Arzt muss nun unter Verwendung seines Wissens eine Reihe weiterer Symptome ermitteln, die zu einer Bestätigung oder aber einer Ablehnung einzelner Diagnose-Hypothesen führen können.

beispielsweise Problemlösen durch Auswahl, während „*diagnosis*“ die Fehlersuche in technischen oder biologischen Systemen umfasst. [LAB⁺96] grenzt die Klassifikation zur Diagnostik dadurch ab, dass bei ersterer alle Problemmerkmale bereits zu Beginn bekannt sein müssen, während bei der Diagnostik auch ein Nachfragen möglich ist.

¹⁸ Die Medizin prägt hier auch den Begriff der Differentialdiagnose. Damit sind Gruppen von Erkrankungen mit ähnlichen Symptomen gemeint, die nach den Untersuchungsergebnissen wahrscheinlich sind und aus denen die eigentliche Diagnose ausgewählt werden muss [Bro06, Bd. 7;S. 7].

Dieser iterative Prozess zwischen Stimulation, Beobachtung und diagnostischen Schließens muss nun solange fortgesetzt werden, bis hinreichend Gewissheit über mögliche Ursachen bzw. der Lösung des Falls besteht. Das kann auch bedeuten, dass eine weitere Eingrenzung keinen zusätzlichen Nutzen bringt, wenn für die verbleibenden wahrscheinlichen Ursachen ohnehin dieselben Medikamente oder dieselbe Therapie verordnet werden würde. Der Detaillierungsgrad hängt folglich vom umgebenden Kontext ab, der auch das *Ziel der Diagnostik* bestimmt. Die Identifizierung eines defekten Bauteils innerhalb einer Komponente ist für eine Reparatur nutzlos, wenn die Komponente ohnehin die kleinste austauschbare Einheit darstellt [Rei92, S. 29ff.]. Die weitere Untersuchung eines Patienten bringt keinen weiteren Nutzen, wenn die vorhandenen Symptome eindeutig auf eine Gruppe von Erkrankungen hinweisen, für die dasselbe Medikament verschrieben wird, es sei denn für wissenschaftliche Zwecke.

Diagnostik als iterativer Prozess

Kontext und Granularität einer Diagnostik

Lösungen des diagnostischen Schließens

Bisher wurden im Rahmen des verwendeten Beispiels sehr konkret wahrscheinliche Ursachen bzw. Diagnose-Hypothesen als Ergebnis diagnostischen Schließens vorgestellt. Allgemeiner formuliert ist das Ergebnis eines Diagnostikfalls eine Art Beschreibung des Systems, das diagnostiziert werden soll, d. h. es werden Eigenschaften oder Zustandswerte des Systems ermittelt. Aufgrund der Komplexität des zu diagnostizierenden Systems besteht die Lösung selten aus einer vollständigen Beschreibung (das würde die Bestimmung aller Zustände Z_Σ voraussetzen). Normalerweise ist diese vereinfacht auf eine Liste von defekten Komponenten, Krankheiten, etc. mit der impliziten Annahme, dass die gegebenen Elemente charakteristisch für die Beschreibung sind und alle relevanten Details implizieren. Ist dies nicht möglich, z.B. aufgrund einer unvollständigen Kenntnis des Normalzustands des Systems, könnte die Lösung lediglich eine partielle Beschreibung des Systems enthalten oder noch einfacher einen Hinweis angeben, was sinnvoll zu tun wäre.

Partielle Lösungsbeschreibung

Im medizinischen Bereich ist, wie auch das Beispiel verdeutlicht, eine vollständige Beschreibung aufgrund der Komplexität des menschlichen Körpers unmöglich. Auch in vielen technischen Domänen ist das zugrunde liegende System und deren Prozesse viel zu komplex für eine detaillierte Beschreibung. Das Finden des richtigen Detaillierungsgrads ist daher auch ein notwendiger Schritt zum Erfolg eines guten Diagnostiksystems.

Keine vollständige Lösung möglich

Formale Definition des diagnostischen Schließens

Als Ausgangspunkt wird eine bekannte initiale Symptomatik $S_I \subseteq S_\Sigma$ vorausgesetzt. Die Aufgabe des diagnostischen Schließens ist es nun, Diagnose-Hypothesen $H_D \subseteq D_\Sigma$ über Lösungselemente oder hier Kom-

Diagnose- Hypothesen

ponenten $K \subseteq K_\Sigma$ zu identifizieren, die die bekannte Symptomatik S_I befriedigend erklären. Das notwendige Wissen kann dabei als eine auf diagnostische Zwecke angepasste Systembeschreibung, auch (System-)Modelle genannt, angesehen werden. Für ein diagnostisches (System-)Modell $\Pi_{\mathfrak{D}}$ gilt damit

$$\Pi_{\mathfrak{D}} : \mathcal{P}(S_\Sigma) \rightarrow \mathcal{P}(D_\Sigma), \quad S_I \mapsto H_D \quad \text{mit} \quad H_D \subseteq D_\Sigma, \quad S_I \subseteq S_\Sigma, \quad (2.6)$$

wobei $\mathcal{P}(S_\Sigma)$ für die Potenzmenge von S_Σ steht¹⁹. Ferner bezeichnet eine Diagnose-Hypothese $h \in D_\Sigma$ ein Tupel aus der (als defekt vermuteten) Komponente k und einem zugehörigen Vertrauenswert $v \in [0,1]$, so dass

$$h = (k, v). \quad (2.7)$$

Dabei ist D_Σ die Menge aller (Diagnose-)Hypothesen. Der mit $\Pi_{\mathfrak{D}}$ aufgespannte Hypothesenraum aller möglichen Hypothesenmengen ist dann gegeben durch die Potenzmenge $\mathcal{P}(D_\Sigma)$. Der Vorteil einer Lösungsmenge H_D , bestehend aus Hypothesen mit Vertrauenswerten, äußert sich im Vergleich zu den weit verbreiteten Algorithmen²⁰, die lediglich eine Lösung/Diagnose vorschlagen, darin, dass eine anschließende Risiko- oder Nutzen-Bewertung möglich wird. Hierfür ist der Vertrauenswert v hilfreich, so dass es für eine Reparatur z.B. sinnvoll sein kann, zunächst die Hypothese mit dem zweithöchsten Vertrauenswert zu verfolgen, wenn dieses Vorgehen aufgrund des geringeren Aufwands sehr viel günstiger ist.

Treffend zusammengefasst, bezeichnet [PGPB96, S.3] die Diagnostik als „eine Abbildung aller möglichen Kombinationen der Symptomwerte in die Potenzmenge der Lösungen“, was dem Kreis mit der Menge an Komponenten K in der Mitte der Abbildung 2.2 entspricht. Eine solche Abbildung wird allgemein als Klassifikation bezeichnet, die damit den Fokus dieser Arbeit darstellt.

Herausforderung – Mehrfachlösungen

Abbildung 2.2 (linke Seite) veranschaulicht eine nicht zu vernachlässigende Herausforderung der Diagnostik - Mehrfachlösungen. Eine Betrachtung der Komplexität des Lösungsraums verdeutlicht, dass im Fall von Einfachlösungen nur zwischen n möglichen Lösungen ausgewählt werden muss, während bei Mehrfachlösungen die Komplexität auf 2^n steigt, folglich n Einfachlösungen $+ \frac{n \cdot (n-1)}{2}$ Doppellösungen $+ \frac{n \cdot (n-1) \cdot (n-2)}{1 \cdot 2 \cdot 3}$ Dreifachlösungen $+ \binom{n}{k}$ k-fachlösungen usw. existieren. Von Bedeutung ist auch, ob sich die Symptome von Mehrfachlösungen wechselseitig beeinflussen bzw. maskieren können oder unabhängig von-

¹⁹ Die Potenzmenge bezeichnet die Menge aller Teilmengen einer gegebenen Grundmenge.

²⁰ Gemeint sind hier vor allem Klassifikationsverfahren, die keinen Sicherheitswert für die Klassifikation berechnen (können). Dazu zählen z.B. klassische Entscheidungs-bäume, viele Clusterverfahren oder regelbasierte Verfahren.

einander auftreten. Erstere Kategorie gehört wohl zu den schwierigsten Diagnostikaufgaben.

Zur Lösung existiert eine große Bandbreite an verschiedenen Verfahren, die alle auf die Form in (2.6) überführt werden können. Eine umfangreiche Einführung wird beispielsweise in [Ise06] oder [PGPB96] gegeben.

Eine weitere, bereits erläuterte Aufgabe des diagnostischen Schließens besteht darin, neue Hypothesen darüber zu identifizieren, welche zusätzlichen Symptome nützlich sein könnten, um den Vertrauenswert v der mit Π_D ermittelten Hypothesen H_D zu erhöhen. Unter Umständen wird hier ein von Π_D verschiedenes Modell benötigt, das zur besseren Unterscheidung im Folgenden mit Π_{DS} sowie die dazugehörigen Hypothesen mit $H_{DS} \subseteq H_{DS_\Sigma}$ benannt werden. Damit folgt

Symptom-Hypothesen

$$\Pi_{DS} : \mathcal{P}(S_\Sigma) \rightarrow \mathcal{P}(H_{DS_\Sigma}), S_I \mapsto H_{DS} \text{ mit } H_{DS} \subseteq H_{DS_\Sigma}, S_I \subseteq S_\Sigma, \quad (2.8)$$

wobei eine Symptom-Hypothese h_{DS} definiert ist durch ein Tupel aus dem (als nützlich vermuteten) Symptom $s \in S_\Sigma$ und einem zugehörigen Vertrauenswert $v \in [0,1]$, so dass

Vertrauenswerte für Symptom-Hypothesen

$$h_{DS} = (s, v). \quad (2.9)$$

Auch hier stellt H_{DS_Σ} analog zu H_{D_Σ} die Menge aller möglichen Hypothesen h_{DS} dar.

Im vorhergehenden Abschnitt wurde der diagnostische Prozess als iterativer Prozess bestimmt. Sind die Vertrauenswerte der Hypothesen H_D nicht hoch genug oder die einzelnen Hypothesen $h_D \in H_D$ nicht separierbar genug, ist eine Erweiterung der bekannten (initialen) Symptomatik S_I notwendig. Es werden somit gemäß den Hypothesen H_{DS} neue Beobachtungen und ggf. Stimulationen durchgeführt, um die bekannte initiale Symptomatik S_I um ausgewählte Symptome $S \subset S_\Sigma \mid s \in h_{DS}$ mit $s \in S, h_{DS} \in H_{DS}$ zu erweitern²¹. Die dann bekannte Symptomatik kann anschließend allerdings nicht mehr als initial bezeichnet werden. Konsequenterweise wird sie dann mit S_R benannt, wobei der Index R für Reparaturfall steht²².

Weitere Symptome

Welche Symptome aus den Symptom-Hypothesen H_{DS} zur weiteren Auswertung nun ausgewählt werden und in welcher Reihenfolge, ist jedoch nicht Aufgabe des diagnostischen Schließens, sondern zählt bereits zum nachfolgenden Schritt, der Bestimmung weiterer Aktionen $\mathcal{A}$. Hier ist eine Nutzenfunktion $N_{H_{DS}}$ notwendig, die jeder Symptom-Hypothese h_{DS} einen Nutzwert, abhängig von den Kosten der notwendigen Aktionen

Nutzenfunktion für weitere Auswertungen

²¹ In [PGPB96, S.6] wird dieser Vorgang die „Hypothesize-and-Test-Strategie“ genannt. Siehe auch Abbildung 2.3.

²² Eine ausführliche Einführung von S_R wird im Rahmen der Fallmodellierung in Kapitel 4.2 gegeben.

$\mathcal{A}$ zur Ermittlung des Symptoms $s \in h_{DS}$ und dem Vertrauenswert $v \in h_{DS}$, zuordnet. Es sollte dann diejenige Symptom-Hypothese h_{DS} zuerst ausgewertet werden, die den größten Nutzen zu bringen scheint. So ist

$$N_{H_{DS}} : H_{DS_{\Sigma}} \rightarrow \mathbb{R}_+, \quad h_{DS} \mapsto n \quad \text{mit } h_{DS} \in H_{DS} . \quad (2.10)$$

Daraus geht auch hervor, dass durch die Nutzenfunktion keine Symptom-Hypothese ausgewählt werden könnte, die nicht auch in der Symptom-Hypothesenmenge $H_{DS_{\Sigma}}$ enthalten ist. Das beschriebene Vorgehen kann nach [PGPB96, S. 5] auch als *Hypothesize-and-Test-Strategie* bezeichnet werden, dargestellt in Abbildung 2.3.

Abbildung 2.3.: Die Hypothesize-and-Test-Strategie (angelehnt an [PGPB96, S. 5])

2.2.5. Evaluation, Entscheidung und Reaktion auf Diagnosen

Welche Therapie?

Einem Patienten, dem der Arzt nach einer Diagnose die Ursachen für sein Leiden erklärt, wird dieses Wissen nicht genügen. Er wird vom Arzt eine Therapie zur Genesung erwarten. Für die Auswahl einer geeigneten Therapie ist zuvor jedoch eine Bewertung der aufgestellten Hypothesen notwendig. Der Arzt muss beispielsweise bewerten, wie folgenschwer einzelne Krankheits-/Ursachen-Hypothesen sind und entsprechend abschätzen, welche Therapie, aufwändig oder weniger aufwändig, durchgeführt werden soll.

Fehlermanagement

In technischen Systemen und der Einschränkung auf Fehlerdiagnostik, wird die Reaktion auf Fehler üblicherweise als Fehlermanagement bezeichnet. Das umfasst zunächst eine Bewertung (Evaluierung) der Diagnose-Hypothesen. Üblich ist eine Einordnung in Fehlerklassen und Fehlergewichtsklassen (eine Definition hierfür wird z.B. in der aktuell gültigen Norm [DIN08] aber auch in [Ise96] gegeben). Auf Basis der Bewertung kann nun eine Entscheidung zu einer entsprechenden Reaktion getroffen werden. Für den Patienten könnte das die Verordnung von Medikamenten, eine Operation oder einfach nur Ruhe sein.

Unabhängig vom gegebenen Beispiel gibt es allgemein zwei Arten der Reaktion. Zum einen die Fehlerkorrektur, für technische Systeme also eine Reparatur oder Wartung, und zum anderen eine Fehlerkompensation. Unter Letzterem werden z.B. Maßnahmen zur Vermeidung größeren Schadens verstanden, bis die Beseitigung des Fehlers im Rahmen einer Wartung oder Reparatur möglich ist. Weiterhin zählt zur Fehlerkompensation z.B. das Wechseln in einen sicheren Betriebszustand, eine Betriebsänderung oder die Rekonfiguration des Systems. Von Bedeutung im Fehlermanagement ist immer auch eine Kosten-Nutzen-Risiko-Abschätzung, wie die vorhergehenden Beispiele illustrierten. Für eine ausführliche Behandlung der Thematik Fehlermanagement wird auf [Ise06, S. 18] verwiesen²³.

Nun könnte es sein, dass die vom Arzt verordnete Therapie nicht die gewünschte Wirkung zeigt. Bereits vorhergehend wurde angemerkt, dass oftmals mehrere mögliche Diagnosen existieren. Ursachen für das Fehlschlagen der Therapie könnten beispielsweise die vorrangige Behandlung der wahrscheinlichsten Ursache (z.B. die Entfernung des Blinddarms) oder aber die Verordnung der kostengünstigsten Therapie sein. So wird es nötig, neue Symptome gemäß den Symptom-Hypothesen H_{DS} zu überprüfen, also neue Beobachtungen anzustrengen, um anschließend neue diagnostische Schlussfolgerungen zu ziehen, wobei das Fehlschlagen der Therapie ebenfalls eine (wertvolle) Information bzw. Beobachtung hierfür darstellt.

Wenn die Therapie nicht wirkt.

Letzteres zeigt auf, dass Reparaturaktionen oder allgemein Aktionen $\mathcal{A}$ des Fehlermanagements zwangsläufig zur Diagnostik zählen müssen. Der Erfolg oder das Fehlschlagen einer Aktion zählt immer auch als Aktion zum Informationsgewinn bzw. zur Erweiterung der Symptomatik. Ein anschauliches Beispiel hierfür ist das Tauschen einer Komponente „auf Verdacht“, also zur Überprüfung, ob diese defekt ist. Gleichzeitig ist damit auch das Bereitstellen von möglichen Aktionen, die auf das

Fehlermanagement als Teil der Diagnostik

²³ Die Abgrenzung der Diagnostik ist in der Literatur leider auch in diesem Punkt nicht eindeutig. Während in [Ise06] der hier beschriebene Schritt als Fehlermanagement bezeichnet und der Diagnostik zugeordnet wird, schreibt [BCF⁺08] diesen Schritt nicht mehr explizit der Diagnostik sondern der Fehlerbehandlung zu, was dort das Ermitteln von geeigneten Schritten zur Fehlerbehebung (im Service) meint.

System Σ angewendet werden können, Teil der Diagnostik, konsistent mit der Definition 2.1. Eine scharfe Trennung zwischen Aktionen zur reinen Symptombestimmung und Aktionen zum Fehlermanagement ist weder möglich noch sinnvoll. Vielmehr handelt es sich um zwei Pole eines Spektrums.

Zusammenfassung des diagnostischen Prozesses

Der eingeführte diagnostische Prozess kann zusammenfassend mit Abbildung 2.4 dargestellt werden. Die einzelnen Schritte sowie die Informationsflüsse sind hier anschaulich dargestellt. Vom zu diagnostizie-

Abbildung 2.4.: Der diagnostische Prozess

renden System Σ , bestehend aus einer Menge an Komponenten K_Σ , der darauf definierten Beziehungen Ξ_Σ und beobachtbaren Zuständen Z , müssen zunächst durch *Beobachtung* Observablen $o \in \mathcal{O} \in \mathcal{O}_\Sigma$ bestimmt, mit intelligenten Methoden $f_{\mathcal{O},m}$ zu Merkmalen m verdichtet und bekannten Symptomen $s \in S_I$ zugeordnet werden. Zur Bestimmung von Observablen sind Aktionen $\mathcal{A}$ wie z.B. Messungen notwendig.

Mit der so verfügbaren initialen Symptomatik S_I werden im Schritt des *diagnostischen Schließens* mittels eines diagnostischen Modells Π_D Diagnose-Hypothesen $H_D = (k, v)$ abgeleitet. Eine Diagnose-Hypothese $h_D \in H_D$ nimmt den Defekt einer Komponente k mit dem zugehörigen Vertrauenswert v an. Gegebenenfalls sollen die Diagnose-Hypothesen präzisiert werden. Hierfür werden aus dem diagnostischen Modell Symptom-Hypothesen $H_{DS} = (s, v)$ abgeleitet. Unter Berücksichtigung einer Nutzenfunktion $N_{H_{DS}}$ werden iterativ durch weitere Aktionen $\mathcal{A}$ weitere Observablen $o \in \mathcal{O}$ und damit Symptome $s \in h_{DS} \in H_{DS}$ bestimmt und zur Symptomatik S_R , nun nicht mehr *initial*, hinzugefügt, bis eine hinreichend präzise Differentialdiagnose H_D erreicht ist. Dieses Vorgehen wird auch *Hypothesize-and-Test* genannt.

Darauf aufbauend werden in der *Aktionsplanung* nun primär Entscheidungen zu geeigneten Aktionen $\mathcal{A}$ zum Fehler-Management getroffen.

Dies kann zum einen eine Fehlerkorrektur wie Reparatur oder Wartung und zum anderen eine Fehlerkompensation wie z. B. der Wechsel in einen sicheren Betriebszustand sein. Die Reaktion beeinflusst/modifiziert also auch hier das diagnostizierte System selbst. Erzielen die Aktionen $\mathcal{A}$ nicht den gewünschten Effekt, stellt dies auch eine Information im Sinne eines Symptoms dar. Es sind dann gegebenenfalls wieder weitere Informationen und Aktionen $\mathcal{A}$ zur Präzisierung der Diagnose-Hypothesen H_D notwendig.

2.2.6. Überlagerte diagnostische Prozesse

Der diagnostische Prozess wurde bewusst sehr allgemein definiert. Angewendet auf die Kraftfahrzeug-Domäne gilt er gleichermaßen für die Onboard-Diagnostik in Steuergeräten wie auch für die Offboard-Diagnostik in Werkstatttestern oder das Vorgehen von Kundendienstmitarbeitern. Am Beispiel des Zusammenspiels von Onboard- und Offboard-Diagnose wird deutlich, dass diagnostische Prozesse teilweise voneinander anhängig sind oder sich als Subprozesse konstituieren. Letzteres ist in Abbildung 2.5 skizziert. Die Informationsermittlung des überlagerten Prozesses interpretiert die Diagnosen und das (Fehler-)Management des Subprozesses explizit als Symptome. Beispiele hierfür sind DTCs²⁴, die ein Ergebnis der jeweiligen Onboard-Diagnostik-Methoden der Steuergeräte darstellen. Der überlagerte Prozess ist in diesem Beispiel folglich die Offboard-Diagnostik in den Service-Werkstätten. Auch der überlagerte Prozess kann direkt das zu diagnostizierende System beobachten und Aktionen darauf anwenden, oder aber die Onboard-Diagnostik-Methoden der Subprozesse bedienen. Weitere Konstellationen sind denkbar.

Abbildung 2.5.: Überlagerte diagnostische Prozesse

²⁴ DTC = Diagnostic Trouble Code (siehe Abschnitt 2.4.2 – Onboard-Diagnostik).

Modulare
Wissensbasen

Interessant hierbei ist die Betrachtung des zur Verfügung stehenden Wissens. Die einzelnen (Sub-)Prozesse können dabei jeweils unterschiedliche Komponenten und Beobachtungen eines Systems erschließen, bedienen sich entsprechend unterschiedlicher Methoden und Modelle zur Merkmalsextraktion und Diagnostik. Entsprechend ist Wissensart und -umfang je Prozess verschieden, so dass hier von modularen Wissensbasen gesprochen werden kann (dieser Aspekt wird u. a. in [PGPB96, S. 165] näher behandelt). Bezogen auf die Summe aller Prozesse stehen damit disjunkte Mengen von Symptomen und Diagnosen durch typischerweise unsicheres mehrstufiges Wissen zwischen Symptome und Diagnosen in Beziehung (vgl. [Mic02, S. 33]). Eine ähnliche Sichtweise wird durch [PGPB96, S. 5] mit dem „*diagnostischen Mittelbau*“ vertreten, hier jedoch nicht so konsequent und abgegrenzt definiert. Weiterhin interessant ist die aus der KI bekannte Agententheorie, welche auf die hier beschriebene Aufgabenstellung adaptiert werden kann. Dann entspricht jeder diagnostische Prozess einem intelligenten (Diagnostik-)Agenten mit unterschiedlichem Wissen und unterschiedlichen Fähigkeiten, die miteinander kooperieren. Weiterführende Literatur zu Agenten in wissensbasierten Systemen ist beispielsweise in [BKI08, Kap. 11], Adaptionen auf Diagnostik und Instanthaltung in [BP99, S. 135ff.], [ARB⁺07, S. 7ff.] oder [BRRSA08] zu finden.

Kooperierende
(Diagnostik-)
Agenten

2.3. Diagnostik und Wissen

Unternehmensspe-
zifische Aspekte
von Wissensbasen

Ein Diagnostiksystem leitet aus Beobachtungen Hypothesen ab. Die Qualität der Diagnostik hängt dabei maßgeblich vom verfügbaren Wissen ab. Die Existenz des Wissens wurde bisher vorausgesetzt, jedoch ist der Wissenserwerb und die Wissenspflege in der Praxis einer der Hauptschwierigkeiten (vgl. [PGPB96, Kap. 1] und [DP99, S. 30]). Wichtig sind hier zwangsläufig unternehmensspezifische Aspekte, die berücksichtigt werden müssen, gerade wenn es um den Aufbau, die Wartung und die Pflege einer Wissensbasis geht. Das beginnt bereits mit der Bestimmung der Verantwortlichen oder der Auswahl der Systeme, die für die Wissensakquisition eingesetzt werden sollen. Manchmal ist in großen Unternehmen die „*organisatorische Distanz*“ sehr viel größer als die Distanz in Raum und Zeit. Auch welche weiteren Aufgaben mit der Verwendung des wertvollen Wissens gelöst werden können, ist eine wichtige Frage. Erst nachdem diese und ähnliche organisatorischen Beschränkungen adressiert wurden, macht es Sinn, über technische Details zu diskutieren.

2.3.1. Diagnostik und wissensbasierte Systeme

Diagnostik setzt Wissen voraus, in der Medizin Wissen über den Menschen, in technischen Domänen Wissen über das System. So ist es nicht verwunderlich, dass diagnostisches Schließen eines der ersten Anwendun-

gen von wissensbasierten Systemen war. Bereits in den späten 60ern wurden erste erfolgreiche Systeme, überwiegend im medizinischen Bereich, entwickelt. Beispielsweise erreichte ein auf Bayesschen Netzen basierendes System zur Diagnostik starker Unterleibsschmerzen [DLS⁺72] oder ein heuristisches System, genannt MYCIN, zur Ermittlung des wahrscheinlichsten Bakteriums und die adäquate Antibiotika-Therapie bei Blutinfektionen [Sho76] eine Leistungsfähigkeit, die mit den Experten in diesen (speziellen) Gebieten vergleichbar war. Intensive Forschung in diesem Bereich brachte eine Vielzahl an Lösungsmethoden und Wissensrepräsentationsformen (Formalismen) hervor, die die Diagnostik zu den bestverstandenen Disziplinen im Gebiet der wissensbasierten Systeme machen. So werden intelligente Diagnostiksysteme heute neben der Medizin in verschiedensten Domänen eingesetzt. Dennoch ist der Markt nur wenig mit solchen Systemen durchdrungen, verglichen mit dem möglichen Potenzial solcher Systeme. Gründe hierfür sind u. a. die notwendigen Infrastrukturen zur (verteilten) Datenerfassung und Datenverarbeitung und der Aufbau und die Wartung einer Wissensbasis ausreichend guter Qualität. Nichtsdestotrotz waren die Voraussetzungen hierfür noch nie so gut wie im Informationszeitalter des 21. Jahrhunderts. Die breit und günstig verfügbare Vernetzung auch von kleinen und preiswerten technischen Geräten erschließt neue Potenziale diagnostischer Anwendungen und wissensbasierter Systeme.

Diagnostisches
Schließen als erste
Anwendung für
wissensbasierte
Systeme

Hohes Potential im
Informationszeitalter

2.3.2. Wissensrepräsentation und -Entstehung

Dieser Abschnitt soll sich nicht damit beschäftigen, wie Wissen definiert werden kann²⁵, oder ob Maschinen in der Lage sind, Wissen zu besitzen, zu erzeugen oder anzuwenden - dazu wird auf teilweise philosophisch angehauchte Diskussionen z. B. in [Row07] verwiesen. Im Kontext dieser Arbeit sind vielmehr die für wissensbasierte Systeme, Data Mining und maschinelles Lernen relevanten Wissensarten bzw. Repräsentationsformen interessant. Auch hier existiert eine große Zahl diverser Formen. Die Bandbreite reicht von rein deskriptivem Wissen zu prozeduralem Wissen, vielfältigen Mischformen und Hybriden²⁶. Sie können im Wesentlichen jedoch nach [DP99, S.30] mit den Oberbegriffen

- „Fallbasiert“,
- „Erfahrungsbasiert“ und
- „Modellbasiert“

zusammengefasst werden²⁷. In einer *fallbasierten* Wissensrepräsentation wird Wissen in Form von (*Referenz-*)*Fällen* repräsentiert. Dabei wer-

Wissensrepräsentati-
onsformen

²⁵ Mit *Wissen definieren* ist hier die Frage gemeint: *Was ist Wissen?*

²⁶ Für eine ausführliche Behandlung des Themas Wissensrepräsentationsformen wird beispielsweise auf [GRS03, S.154ff.;S.599ff.] oder [Kur89, S.39ff.] und für wissensbasierte Systeme auf [BKI08] verwiesen

²⁷ Leider ist diese Einteilung in der Literatur nicht eindeutig. In [PGPB96, S.7] wird beispielsweise in sechs verschiedene Oberkategorien, orientiert an der Anwen-

den Beobachtungen über das vergangene Verhalten des Systems direkt verknüpft mit von Menschen bereitgestellten, bekannten erfolgreichen Lösungen bei diesen Beobachtungen.

Eine *erfahrungsbasierte* Wissensrepräsentation beherbergt Wissen in Form von Regeln, Modellen, Ontologien, etc., die über einen langen Zeitraum hinweg durch Experten bzw. Spezialisten durch das Lösen von Fällen (es kann also von *Erfahrung* gesprochen werden) entstehen. Erstaunlicherweise werden hier Beobachtungen und Lösungen in abstrakter Form verbunden, wie es insbesondere für Ontologien gilt.

Unter *modellbasierter* Wissensrepräsentation wird insbesondere Wissen in Form von explizit modellierten Systemen verstanden. Diese wird ebenfalls durch Experten durchgeführt, ist jedoch im Gegensatz zu „*erfahrungsbasierten Modellen*“ sehr viel konkreter beziehungsweise detaillierter. Beobachtungen und Lösungen sind hier nicht direkt verknüpft, sondern müssen beispielsweise durch Inferenzmethoden hergeleitet werden.

Wissens-
entstehungsarten

Wissensrepräsentationsformen sollten nicht unabhängig von der Wissensentstehung betrachtet werden. Generell kann hier nach [BH98] in

- *Wissensmodellierung* (engl. Knowledge Engineering) und
- *Maschinelles Lernen* (engl. Machine Learning)

unterschieden werden. Während Wissensmodellierung den sukzessiven Aufbau von Wissen durch Experten meint und somit vornehmlich durch sogenannte *Expertensysteme* erfahrungsbasiertes und modellbasiertes Wissen entsteht, wird unter *maschinellern Lernen* die Fähigkeit des Systems verstanden, aus (aufbereiteten) Daten selbstständig Wissen zu generieren.

2.3.3. Faktoren der Lösungsmethoden

Für viele Domänen sind nicht alle Wissensrepräsentationsformen verfügbar, z. B. kann Wissen in Form von (Referenz-)Fällen für die Diagnostik neuer technischer Produkte noch nicht verfügbar sein, während für viele biologische Systeme das kausale Verständnis oft beschränkt ist. Wenn Falldatenbanken existieren, hängt deren Verwendbarkeit stark von der Quantität, Qualität und Formalisierung ab. Auch erfahrungsbasiertes Wissen kann nicht verfügbar sein, wenn Experten unwillig oder zu ausgelastet sind, um es zu formalisieren. Neben den unternehmensspezifischen Aspekten gibt es noch eine ganze Reihe weiterer Fragestellungen, die die Wahl der Lösungsmethoden bzw. des Diagnostikverfahrens beeinflussen:

dungsmethode (sichere Klassifikation, heuristische Klassifikation, modellbasierte Klassifikation, statistische Klassifikation, neuronale Klassifikation und fallbasierte Klassifikation), eingeteilt. In [BH98] wird erfahrungsbasiertes Wissen hier als heuristisches Wissen bezeichnet, ansonsten jedoch eine ähnliche Einteilung gegeben.

Aufwand für die initiale Wissensakquisition Wie viel Zeit wird benötigt, um eine verwendbare initiale Wissensbasis aufzubauen?

Wartungs-/Pflegeaufwand Welche Ressourcen - speziell menschliche Ressourcen - sind notwendig, um die Wissensbasis in einem sich verändernden Umfeld verwendbar zu halten?

Schleichende Degradierung durch fehlerhafte, unvollständige oder unpräzise Beobachtungen Wie empfindlich sind die verwendeten Methoden gegenüber der Qualität der bereitgestellten Beobachtungen?

Schleichende Degradierung durch fehlerhaftes oder unvollständiges Wissen Wenn es nicht möglich oder ökonomisch sinnvoll ist, vollständige und korrekte Wissensbasen zu erzeugen, inwieweit degradiert die Qualität der Lösung mit der Qualität des zur Verfügung stehenden Wissens?

Objektivität und Korrektheit Kann die Korrektheit der Lösungen bewiesen werden? Die Korrektheit hängt von Annahmen über die verfügbaren Beobachtungen und das bereitgestellte Wissen ab.

Erklärbarkeit Wie gut kann die Lösung erklärt werden? Können Anwender durch die Überprüfung der Erklärung die Lösung beurteilen? Das ist besonders dann bedeutend, wenn Menschen für die Qualität der Lösung auch im Nachhinein verantwortlich bleiben.

Einzelfehlerannahme Ist es theoretisch wie praktisch akzeptierbar, dass lediglich ein Fehler gleichzeitig diagnostiziert werden kann? Bzw. sind Interaktionen typisch, wenn Mehrfachfehler auftreten? Werden in diesem Fall kritische Fehldiagnosen erzeugt, wenn nur von einem Einfachfehler ausgegangen wird?

Varianten Wenn die Komponenten, die durch die Diagnostik identifiziert werden sollen, variieren, ohne dabei aber zu vollständig anderen Komponentenklassen zu gehören, ist es dann wünschenswert, diese durch kleine Änderungen in der Wissensbasis zu handhaben? Welcher Aufwand zur Variantenmodellierung ist dann erforderlich?

Wiederverwendung von Wissen Ist das Wissen, das für die Diagnostik verwendet wird, auch für andere Aufgaben verwendbar?

Unvorhergesehene Lösungen Bis zu welchem Grad ist es möglich, Fehler zu diagnostizieren, die nicht explizit von den Entwicklern der Wissensbasis vorgesehen wurden?

Zeitabhängigkeit Wie wichtig ist die zeitliche Veränderung von Beobachtungen und Fehlern zum Finden von Lösungen?

Variable Granularität von Beobachtungen/Lösungen Wie detailliert muss eine Beobachtung/Lösung sein? Ist es notwendig, dass diese während des Diagnostikprozesses variiert? Würden präzisere Beobachtungen in präzisere Lösungen übersetzt werden?

2.4. Diagnostik im Kraftfahrzeug-Bereich

Die Diagnostik im Kraftfahrzeug-Bereich unterliegt einigen domänenspezifischen Besonderheiten. Es haben sich viele „*Best Practice*“-Methoden, Standards und Normen sowie eine eigene Terminologie etabliert. Weiterhin gibt es eine ganze Reihe an für die Domäne typischen Herausforderungen. Einen guten Überblick über Kraftfahrzeug-Systeme und Kraftfahrzeug-Diagnostik geben beispielsweise [WR06], [Rei09], [Bor08], [Krü06] und [Gmb07]. Ein grundlegendes Verständnis wird für die nachfolgenden Kapitel vorausgesetzt. Der folgende Abschnitt gibt lediglich einen kleinen Abriss und konkretisiert die vorhergehenden Ausführungen zur Diagnostik für den Kraftfahrzeug-Bereich.

Historischer Abriss

Die Diagnostik im Kraftfahrzeug-Bereich existiert schon so lange, wie es Kraftfahrzeuge gibt. Während vor einigen Jahrzehnten die Diagnostik überwiegend aus Sichtprüfungen bestand, unterzog sie sich aufgrund der stark zunehmenden elektronischen Komponenten und der damit einhergehenden Komplexität einem starken Wandel. Mit der Einführung der elektronischen Zündanlage beispielsweise reichte eine Sichtprüfung nicht mehr aus, und es waren neue Messgeräte zur Überprüfung notwendig. Die Interpretation der Messwerte übernahm typischerweise aber weiterhin der KD-Mitarbeiter. Mit der breiten Einführung von elektronischen Komponenten, insbesondere Steuergeräten (ECUs)²⁸, war eine Änderung der Diagnostik notwendig. Während lange Zeit lediglich ein Steuergerät, das Motorsteuergerät, im Fahrzeug verbaut war, kamen jetzt Steuergeräte für Sicherheitsfunktionen wie ABS, ESP, Airbag oder Komfortfunktionen wie Navigation, Klimatronik, usw. hinzu. Die Steuergeräte arbeiteten jedoch nicht unabhängig voneinander, sondern benötigten oftmals Sensordaten bzw. Informationen aus anderen Steuergeräten, so dass die Einführung einer Buskommunikation notwendig wurde. Als Beispiel bezieht die Funktion des Blinkens, je nach Fahrzeug bis zu 10 Steuergeräte mit ein. Abbildung 2.6 zeigt eine typische Vernetzung zwischen Steuergeräten in einem modernen Kraftfahrzeug. In diesem Beispiel mit über 30 Steuergeräten werden drei verschiedene CAN-Busse für die Kommunikation verwendet. Die Zahl der Busse ist u. a. aufgrund der sonst zu hohen Buslast notwendig. Typisch ist auch die Gruppierung der Steuergeräte in Antrieb, Komfort und Infotainment.

Verteilte Funktionen — Beispiel Blinken

²⁸ Steuergeräte werden hier allgemein als Systeme im Sinne von Definition 2.2 betrachtet.

Während in der hier schon für viele Beispiele herangezogenen Medizin Ärzte für jedes Spezialgebiet existieren, müssen Fahrzeuge nach wie vor von *einem* KD-Mitarbeiter diagnostiziert und repariert werden. Eine komponentenspezifische Aufteilung der Fachkräfte existiert also für Fahrzeuge (noch) nicht. Eine Lösung ist es, den KD-Mitarbeitern eine entsprechende Wissensquelle zur Verfügung zu stellen, mit deren Hilfe sie die enorme Komplexität beherrschen können: das Diagnostiksystem.

Keine/wenig spezialisierte KD-Mitarbeiter

Die Komplexität der heutigen Fahrzeuge verändert die Anforderungen an geeignete Diagnostiksysteme. Generell können nach [Pet04] Fehler, die im Service auftreten, in drei Kategorien eingeteilt werden (siehe auch Abbildung 2.7):

Fehlerkategorien und deren Beherrschbarkeit

- 1. Schwierigkeitsgrad einfach bis mittel** Der Fehler kann noch von gut ausgebildeten KD-Mitarbeitern diagnostiziert und behoben werden.
- 2. Schwierigkeitsgrad schwer** Der Fehler kann nur noch von KD-Mitarbeitern mit einer speziellen Ausbildung diagnostiziert und behoben werden.
- 3. Schwierigkeitsgrad sehr schwer** Der Fehler kann nur noch durch den OEM²⁹ diagnostiziert und behoben werden.

Das Ziel eines guten Diagnostiksystems ist es nun, die Grenzen der Kategorie zwei und drei in Abbildung 2.7 nach rechts zu verschieben, so

Ziel — Beherrschbarkeit komplexer Fehler ohne spezielle Ausbildung

²⁹ Original Equipment Manufacturer (KFZ-Hersteller)

Abbildung 2.6.: Typische Vernetzung von Steuergeräten in einem modernen Fahrzeug [Pet04]

dass auch Fehler höheren Schwierigkeitsgrades in Zukunft durch den Service und weniger spezialisiert ausgebildete KD-Mitarbeiter diagnostiziert und instant gesetzt werden können.

2.4.1. Anforderungen und Nutzer der Fahrzeugdiagnostik

Bevor nun weiter die technischen Details und Verfahren der Fahrzeugdiagnostik behandelt werden, soll zunächst der unternehmensspezifische Kontext umrissen werden. In einem Automobilkonzern sind die Nutzer der Fahrzeugdiagnostik keineswegs, wie aufgrund der bisherigen Beispiele vermutet werden könnte, nur die KD-Mitarbeiter in den Werkstätten. Eine ganze Reihe an Nutzern und damit auch Anforderungen ergeben sich bereits, bevor das erste Fahrzeug ausgeliefert wird, in der Entwicklung und später in der Produktion. Abbildung 2.8 stellt die einzelnen Nutzer und deren Anforderungen anschaulich dar³⁰. Zusammengefasst können die Anforderungen Nutzerübergreifend konkretisiert werden zu:

1. Wirtschaftliche Reparatur und Instandsetzung von Fahrzeugen
2. Gewährleistung der Sicherheit und Zuverlässigkeit während des Betriebs
3. Qualitätssicherung und Fehlersuche in der Produktion
4. Unterstützung der Fahrzeugentwicklung

Die technische Realisierung dieser Anforderungen ist mit ihren wesentlichen Elementen in Abbildung 2.9 skizziert. Es kann dabei grundsätzlich

³⁰ Eine detaillierte Beschreibung der einzelnen Anforderungen wird im Anhang A.1 gegeben.

Abbildung 2.7.: Auswirkungen der Komplexität auf den Service (qualitative Darstellung) [Pet04]

in die zwei Anwendungskategorien Onboard(Eigen)-Diagnostik, realisiert durch die Steuergeräte im Fahrzeug, und der Fremd- oder Offboard-Diagnostik, bereitgestellt durch den Werkstatttester bzw. Diagnostiktester, unterschieden werden. Während als Onboard-Diagnostik all diejenigen Funktionen bezeichnet werden, die im Fahrzeug bzw. in den einzelnen Steuergeräten verortet sind, bezieht sich die Offboard-Diagnostik auf Funktionen außerhalb des Fahrzeugs, die jedoch auf die Onboard-Diagnostik aufbauen (vgl. auch Beispiel des diagnostischen Subprozesses in Abschnitt 2.2.6).

2.4.2. Onboard-Diagnostik

In Bezug auf die eingeführte Systematik der Diagnostik (vgl. Abbildung 2.4 und Definition 2.4 – Beobachtung) liegt der methodische Schwerpunkt der Onboard-Diagnostik im Schritt der Beobachtung und damit inhärent der Merkmalsextraktion und Informationsverdichtung. Für mechatronische Systeme und vor dem Hintergrund der Anforderung der Sicherheit während des Betriebs, konstituiert sich dies vermehrt in Methoden der Fehlererkennung, kombiniert mit geeigneten Sensoren.

Nach Abbildung 2.9 kann für elektronische Steuerungs-/Regelungs- und Überwachungssysteme im Fahrzeug zwischen folgenden Komponenten unterschieden werden: Sollwertgeber, Sensoren, Aktuatoren, elektronische Steuergeräte und die Strecke³¹. Die Vernetzung der Steuergeräte untereinander wurde bereits angesprochen (vgl. Abbildung 2.6). Das Verhalten des Fahrzeugs wird durch den Fahrer und die Umwelt beeinflusst bzw. durch die Komponenten des Systems Fahrzeug–Fahrer–Umwelt.

Technische
Realisierung der
Fahrzeugdiagnostik

³¹ Gemeint ist hier die Regelstrecke, die den (elektro-)mechanischen Zusammenhängen im Fahrzeug entspricht.

Abbildung 2.8.: Verschiedene Anforderungen und Nutzer der Fahrzeugdiagnostik [Sup05]

Erst ein vollständiges elektronisches System aus Steuergeräten, Sollwertgebern, Sensoren und Aktuatoren beeinflusst oder überwacht die Strecke und erfüllt so die (Benutzer-)Anforderungen.

Zur Überwachung der korrekten Funktion bzw. Fehlerfreiheit existieren eine ganze Reihe an verschiedenen Verfahren. Eine Taxonomie einiger hierfür gängigen Verfahren ist in Abbildung 2.10 dargestellt³².

Es kann grundsätzlich in die Auswertung von Einfach- oder Mehrfachsignalen unterteilt werden. Entsprechend einfach bzw. komplex können die darauf aufsetzenden Verfahren bestimmt sein. Aufgrund der fast immer begrenzten und dafür aber hoch okkupierten Rechenleistung der Mikrocontroller werden jedoch zumeist einfache (recheneffiziente) Verfahren wie die Gruppe der Grenzwert- oder Trendüberwachungen

³² Diese und weitere Verfahren werden weiterführend in [Ise06], [WR06], [Rei09], [IHSK03] und [Bor08] behandelt.

Abbildung 2.9.: Prinzipielle Darstellung der Fahrzeugdiagnostik (angelehnt an [SZ04, S. 104ff])

Abbildung 2.10.: Taxonomie von Verfahren zur Fehlererkennung in mechatronischen Systemen (angelehnt an [Ise06])

genutzt. Hierfür werden, wie Abbildung 2.9 verdeutlicht, Sensorsignale digitalisiert bzw. digital erfasst (A/D-Wandlung) und neben der funktionsrelevanten Nutzung auf zulässige Werte überwacht. Analog gilt dies für Aktuatoren, wobei hier weitere Mechanismen/Sensoren zur Detektion von Kurzschlüssen, Unterbrechungen, Überhitzung, etc. integriert werden. Aufwändige Signalmodelle werden sehr viel seltener eingesetzt, während Prozessmodelle nur noch in rechenstarken Komponenten wie z. B. dem Motorsteuergerät verwendet werden. Oftmals können aufwändige Prozessmodelle auch durch statische Kennlinienfelder kombiniert mit einfachen Toleranzüberwachungen ersetzt werden³³.

Wurde mit einem der Verfahren ein abnormes Verhalten, somit ein Fehlverhalten detektiert, wird ein entsprechendes Fehlermanagement (vgl. Abschnitt 2.2.5) durchgeführt. Je nach betroffener Funktionalität wird dann ein sogenannter Ereignisspeichereintrag (engl. Diagnostic Trouble Code - DTC) erzeugt und im Speicher des Steuergeräts abgelegt (in Abbildung 2.9 als Ereignisspeichereintrag bezeichnet). Weiterhin werden für sicherheitsrelevante Funktionen wie dem ABS oder ESP geeignete Schutzmaßnahmen, zumeist eine Deaktivierung des Systems, gekoppelt mit einem Warnsignal, eingeleitet. Aber auch Änderungen des Betriebszustands wie z. B. Motornotlauf sind mögliche Reaktionen im Fehlermanagement.

2.4.3. Offboard-Diagnostik

Die Offboard-Diagnostik vereint die Diagnostikfunktionalität außerhalb des Fahrzeugs, die zur Realisierung zunächst einen Informationsaustausch mit den Komponenten des Fahrzeuges voraussetzt (vgl. diagnostischer Subprozess). Hierfür wurde die Diagnostikschnittstelle über einen genormten Stecker, derzeit OBD-2 nach ISO 15031 [ISO06], eingeführt, mit Hilfe dessen fahrzeugseitig über das Gatewaysteuergerät mit allen Steuergeräten kommuniziert werden kann (vgl. auch Abbildung 2.6). Ein externes Diagnostiksystem wie beispielsweise der in Abbildung 2.12 dargestellte VAS5051 der Volkswagen AG kann nun auf verschiedene Diagnostikdienste der Steuergeräte zugreifen. Dazu gehört u. a. das Abrufen der Ereignisspeichereinträge (DTCs) der Steuergeräte (genannt *Eigendiagnose*). Diese liefern im Fall einer Kundenbeanstandung, d. h. eine vom Kunden beanstandete Fehlfunktion, erste (wertvolle) Hinweise auf mögliche Ursachen. Es muss hier nicht weiter ausgeführt werden, dass Ereignisspeichereinträge wesentlich dazu beitragen, die Fehlersuchzeiten zu verkürzen³⁴. Weiterhin zählen zu den Funktionen auch das Lesen der vom Steuergerät gemessenen Sensormesswerte (die sogenannten *Messwertblöcke*) oder das Durchführen von Stellgliedtests (fest vorgegebene

Kommunikation
mit diagnostischen
Subprozessen

Werkstatttester-
funktionen:

- Eigendiagnose
- Messtechnik
- Geführte
Fehlersuche

³³ Einen Überblick über Modellierungsarten gibt beispielsweise [Lük03, S. 9ff].

³⁴ Weitere in Steuergeräten häufig verfügbare Diagnostikdienste sind in Tabelle D.4 im Anhang D.2 aufgeführt, wovon einige bereits diskutiert wurden. Für weiterführende Literatur wird auf [Gmb07] verwiesen.

Abbildung 2.11.: Übersicht von Klassifikationsverfahren nach [PGPB96, S. 7]

Abbildung 2.12:
Diagnostiksystem
VAS5051

Abläufe für die Aktuatoren, um deren Funktion zu testen). Weitere Messungen können mit den integrierten Messgeräten manuell durchgeführt werden (dazu gehören Multimeter- oder Oszilloskopfunktionen).

Wird die Onboard-Diagnostik in Bezug zum in Kapitel 2.1 eingeführten Diagnostikbegriff gesetzt, fällt auf, dass dort aus der Perspektive der Offboard-Diagnostik lediglich der Schritt der Beobachtung und Informationsverdichtung verortet ist. Das Ergebnis der Beobachtung sind Symptome, die hier konkret Ereignisspeichereinträge oder aber direkte Sensormesswerte über die Diagnostikservicedienste *Ereignisspeicher abfragen* und *Messwertblock lesen* bereit gestellt werden (vgl. Abbildung 2.9 unten).

Das eigentliche diagnostische Schließen bzw. die Ursachenfindung wurde bisher dem KD-Mitarbeiter überlassen. Leistungsfähige Diagnostiksysteme (im Service *Werkstatttester* oder *Diagnosetester* genannt) zeichnen sich dadurch aus, dass sie dem KD-Mitarbeiter nicht nur alle Informationen aus den einzelnen Steuergeräten anzeigen, sondern ihn zusätzlich bei der Fehlersuche aktiv unterstützen. So wird auch der Bezug zu Abbildung 2.7 und dem Ziel, auch anspruchsvollere Diagnostikfälle durch nicht spezialisierte KD-Mitarbeiter lösen zu können, hergestellt. Hierfür existieren eine Vielzahl verschiedener Verfahren, wovon einige angelehnt an [PGPB96] in Abbildung 2.11 dargestellt sind³⁵. Hier soll angemerkt werden, dass sich diese Arbeit auf Klassifikation fokussiert. Eine derzeit am häufigsten eingesetzten Methodenklasse sind Entscheidungsbäume. Eine Ausprägung davon, die auch im Service der Volkswagen AG und der BMW AG eingesetzt wird, ist die sogenannte *Geführte Fehlersuche (GFS)* mit SIDIS³⁶ von Siemens.

2.4.4. Die Geführte Fehlersuche

GFS durch Entscheidungsbäume

Die Geführte Fehlersuche (GFS) verwendet Fehlersuchprogramme auf Basis von Entscheidungsbäumen, welche von Experten erstellt, in verschiedene Sprachen übersetzt und weltweit an die Anwender verteilt werden. Das Wissen wird dabei von einem Team durch eine FMEA (Failure-Modes-and Effects Analysis), d. h. eine Untersuchung der Auswirkungen von Komponentenfehlern auf die Funktion des Systems, manuell erarbeitet. Es wird für jeden Ereignisspeichereintrag, der in einem Fahrzeug auftreten kann, ein Fehlersuchprogramm erstellt. Ein Fehlersuchprogramm überprüft über den Diagnostikservicedienst *Messwertblock lesen* oder über die Anleitung des KD-Mitarbeiters, eine oder mehrere Komponenten. Dieser Zusammenhang zwischen Ereignisspeichereinträge (DTCs), Fehlersuchprogrammen der GFS und Komponenten (K) in der Geführten Fehlersuche wird durch Abbildung 2.13 veranschaulicht. Der prinzipielle

³⁵ Weitere Taxonomien und Diskussionen verschiedener Verfahren hierfür sind beispielsweise in [Hei99, S. 13], [Ise06, S. 290], [PGPB96, S. 7], [PZMH00, S. 11ff], [DP99, S. 33ff] und [BH98, S. 4] zu finden.

³⁶ Siemens Diagnose- und Informationssystem

Abbildung 2.13.: Zusammenhang zwischen Ereignisspeichereinträgen (DTCs), Fehlersuchprogrammen (GFS) und Komponenten (K) in der Geführten Fehlersuche

Aufbau eines Fehlersuchprogramms ist in Abbildung 2.14 dargestellt. Der Start eines Prüfprogramms setzt die Existenz eines bestimmten DTCs voraus³⁷. Beginnend wird die erste Prüfanweisung an den KD-Mitarbeiter ausgegeben. Nach der Durchführung der Prüfung und Rückmeldung an den Diagnostiktester, wird nun die nächste Prüfanweisung generiert oder aber die als defekte vermutete Komponente angezeigt. Eine ausführliche Beschreibung der Geführten Fehlersuche mit dem System SIDIS wird u. a. in [Tap04, S. 44ff] gegeben. Auf Schwachstellen und Herausforderungen der GFS wird im Rahmen der Analyse näher eingegangen werden.

Fehlersuchprogram-
me

³⁷ Der Startpunkt wurde teilweise auch als *Einsprungpunkt* bezeichnet.

Abbildung 2.14.: Links: Beispiel für ein Fehlersuchprogramm der Geführten Fehlersuche. Rechts: Einige Bildschirmausschnitte

3 Herausforderung der heutigen Fahrzeugdiagnostik

Die Diagnostik im Kraftfahrzeug-Bereich ist heute im Wesentlichen durch drei Herausforderungen geprägt: Dazu gehört zum Ersten die steigende Komplexität durch immer mehr, teilweise stark verteilte Funktionen. Zum Zweiten der Trend zur Individualisierung des Fahrzeugs und der damit verbundenen Diversität und zum Dritten der Paradigmenwechsel von der bauteilorientierten hin zur funktionsorientierten Entwicklung.

Dieses Kapitel gibt anhand der aufgezeigten Trends eine Analyse von heute eingesetzten und zum Stand der Forschung gehörenden Verfahren zur Offboard-Diagnostik, mit dem Ziel, die jeweiligen Vor- und Nachteile zu identifizieren und zu bewerten. Darauf basierend wird die resultierende Forschungsaufgabe abgeleitet und definiert.

3.1. Beherrschung der Komplexität

Bauteilorientierung

Im Automobilbau wie auch in vielen weiteren technischen Domänen wird *bauteilorientiert* entwickelt. Das bedeutet, das Gesamtsystem wird zerlegt in möglichst abgeschlossene und schnittstellenarme Teilsysteme, bestehend aus einem oder mehreren Komponenten. Eine Komponente wird im Kraftfahrzeug-Bereich oftmals auch als *Bauteil* bezeichnet und stellt die kleinste (im Service) tauschbare Einheit dar. Es ist nicht gleichzusetzen mit den *Einzelteilen*, aus denen ein Bauteil besteht¹. Die Dekomposition in Teilsysteme wird in der Hardwarearchitektur durchgeführt, d. h. sie führt gleichzeitig auch zu räumlich abgegrenzten Teilsystemen, deren Kern zumeist aus jeweils einem Steuergerät (der Komponente) besteht. Aus technischer Sicht ergeben sich durch die Trennung auf Hardwareebene Vorteile für das Teilsystem, wie eine bessere Überschaubarkeit, einfachere Integration, einfachere Austauschbarkeit oder vereinfachtes Variantenmanagement, alles nach dem Prinzip „*teile und herrsche*“.

Onboard

Aus diagnostischer Sicht generiert dies insofern Vorteile, da Diagnostikwissen, Onboard wie Offboard, durch Experten manuell erstellt wird und somit auch von solchen überschaubar sein muss. Hier ist eine Bauteilorientierung auf Hardwarearchitekturebene insoweit hilfreich, wie sie die Überschaubarkeit des Teilsystems gewährleistet oder veranschaulicht. Ein vom Experten nicht vorausgesehener Fehler wird durch die Eigendiagnose höchstwahrscheinlich gar nicht oder nicht korrekt erkannt und

¹ Beispiele für Komponenten bzw. Bauteile sind ein ABS-Sensor, das Kombiinstrument oder das Motorsteuergerät.

folglich auch nicht diagnostiziert. Modellbasiertes Wissen, angewendet durch entsprechende Algorithmen in den Steuergeräten, könnte hier zwar bedingt durch die verschiedene Art der Wissensrepräsentation eine Verbesserung darstellen, verursacht aber neue Herausforderungen wie die notwendige Detailtiefe des Modells und insbesondere die notwendige Rechenleistung, so dass diese Methoden selten zum Einsatz kommen.

Für die Offboard-Diagnostik ergibt sich ein ähnliches Bild. Hier hat sich als Standard die GFS etabliert, die in die Kategorie der Expertensysteme, d. h. der Wissensmodellierung von erfahrungsbasiertem Wissen eingeteilt werden kann. Systeme aus dieser Klasse können nur durch die Überschaubarkeit der (Teil-)Systeme befriedigende Ergebnisse liefern. Auch hier gilt analog, dass ein Fehler, der durch den Experten vergessen wurde, also nicht in der Wissensbasis modelliert wurde, durch kein Expertensystem diagnostiziert werden kann.

Offboard

Der Sachverhalt lässt sich für die Onboard- wie auch Offboard-Diagnostik folgendermaßen formalisieren: Sei

Vollständigkeit und Diagnostizierbarkeit

- K_Σ die Menge aller Komponenten eines Systems,
- K^* die Menge aller durch das Diagnostiksystem als defekt diagnostizierbaren Komponenten mit $K^* \subseteq K_\Sigma$ und
- $\mathcal{O}^*$ die Menge aller durch das Diagnostiksystem bestimmbar Observablen mit $\mathcal{O}^* \subseteq \mathcal{O}_\Sigma$ und damit
- S^* die Menge aller dem Diagnostiksystem bekannten Symptome mit $S^* \subseteq S_\Sigma$,

dann muss zum einen eine hinreichend gut definierte Diagnostikfunktion $\mathfrak{D}$ existieren, d. h. die Abbildung

$$\mathcal{P}(S^*) \rightarrow \mathcal{P}(H_D^*) \text{ mit } H_D^* = \bigcup_{h^* \in H_D^*} (\exists k_i^* \in h^* \mid k_i^* \in K^*) \quad (3.1)$$

bildet die bekannten Symptome S^* *geeignet* auf die Hypothesen H_D^* ab, und zum anderen müssen *geeignete* Abbildungsfunktionen $f_{\mathcal{O},s}$ existieren, so dass für jedes $h^* \in H_D^*$ eine Observable $o^* \in \mathcal{O}^*$ existiert, die ein oder mehrere Symptome $S \subseteq S^*$ erzeugt, die h^* hinreichend begründen. Damit muss mindestens

$$\forall h^* \in H_D^* \exists_+ o^* \in \mathcal{O}^* \mid h^* \in Z_{\mathcal{O},H}(o^*) \text{ mit } Z_{\mathcal{O},H} : Z_{\mathcal{O},s} \circ \mathfrak{D} \quad (3.2)$$

erfüllt sein.

Die Formalisierungen machen zweierlei deutlich: Zum einen können Fehlerarten und damit zwangsläufig Observablen $\mathcal{O}^- \subseteq \mathcal{O}$, die bei der Entwicklung der Wissensbasis nicht berücksichtigt wurden, unter Umständen zu keinen, wenn $\mathcal{O}^- \not\subseteq \mathcal{O}^*$, oder aber falschen Hypothesen H_D

führen, und zum anderen können Komponenten, die bei der Erstellung der Wissensbasis nicht berücksichtigt wurden, nicht in der Hypothesenmenge H_D^* enthalten sein und wegen $\nexists h \in H_D^- \mid h \in H_D^*$ mit $H_D^- \cap H_D^* = \emptyset$ nie als Komponente verdächtigt werden.

Fehlerfortpflanzung – Beispiel Sensorfehler

Ein anderes Problem der Komplexität ist die Fehlerfortpflanzung. Zur Veranschaulichung des Gegenstands kann folgendes Beispiel nach Abbildung 3.1 dienen. Das Motorsteuergerät erkennt einen Sensorfehler. Als Folge erzeugt die Software routine, die den Sensor überprüft, einen Ereignisspeichereintrag (DTC). Weitere Funktionen, die ebenfalls diesen Sensor benötigen, erzeugen durch deren Selbsttests ebenfalls Ereignisspeichereinträge. Gleichzeitig sind weitere Steuergeräte, z. B. das Getriebe- und Kombisteuergerät auf die Ergebnisse der Funktionen des Motorsteuergeräts angewiesen. Als Folge erzeugen diese ebenfalls Ereignisspeichereinträge. So werden für nur einen realen Fehler eine ganze

Abbildung 3.1.: Beispiel für die Auswirkungen eines Sensorfehlers (SG = Steuergerät) [WR06]

Reihe an Ereignisspeichereinträgen in verschiedenen Steuergeräten und zu verschiedenen Zeiten eingetragen, die weder zeitlich noch inhaltlich in irgendeiner Weise korrelieren müssen.

Beispiel – Steckfehler am Gatewaysteuergerät

Ein weiteres Beispiel der Fehlerfortpflanzung wird in Abbildung 3.2 gegeben, das besonders Fehler in der Produktion adressiert. Für nur einen Steckfehler am Gatewaysteuergerät werden, bedingt durch die hierdurch gestörte Kommunikation, 38 Ereignisspeichereinträge in 5 verschiedenen Steuergeräten erzeugt.

Die Beispiele verdeutlichen, warum gerade in vernetzten Fahrzeugsystemen Symptommatiken auftreten können, die bei einer rein komponentenorientierten Betrachtung, wie es für die Onboard-Diagnostik implizit

Abbildung 3.2.: Fehlerfortpflanzung für einen Steckfehler am Gateway [Hud04]

gilt, nicht aufgelöst werden können.

Der Trend in der Fahrzeugentwicklung geht nun insbesondere mit der Gründung und Migration der AUTOSAR-Initiative² zunehmend zu einer funktionsorientierten Architektur. Das heißt, das System Fahrzeug wird zuerst hinsichtlich der zu realisierenden Funktionalität dekomponiert, bevor dann ein Mapping der (Teil-)Funktionen auf (Hardware-)Komponenten durchgeführt wird. Damit stehen nicht die Komponenten sondern die Funktion im Vordergrund. Dieser Gesinnungswandel ist insofern sinnvoll, wie immer mehr hochvernetzte *Funktionen* aus dem Bereich Fahrerassistenz oder Infotainment hinzukommen, für die eine hardwareorientierte Komponentensicht zunehmend hinderlich wird. Es sind primär die Funktionen, nicht die Hardwarekomponenten, die den Kundennutzen erzeugen. Angemerkt sei hier, dass der Begriff *Komponente* in dieser Arbeit sehr allgemein verstanden wird, also nicht auf Hardware wie Steuergeräte o. ä. beschränkt ist, sondern Softwarekomponenten oder Funktionen ebenfalls als Komponenten im Sinn der Definition 2.1 angesehen werden können.

Funktions-
orientierung

In der funktionsorientierten Sicht kann eine Funktion nicht mehr eindeutig einem Steuergerät oder einem (Hardware-)Teilsystem zugeordnet werden. Vielmehr verteilen sich Funktionen auf verschiedene Steuergeräte. Ein prominentes Beispiel hierfür ist die Funktion *Blinken*, die sich schon seit mehreren Fahrzeuggenerationen auf mehrere Steuergeräte verteilt. Ist nun die Funktion Blinken fehlerhaft, genügt es nicht, den Fehler im Kontext des verantwortlichen *Bauteils* zu suchen, da dieses per se nicht existiert. Es entsteht also ein Kontrast zwischen der funktionsorientierten Sichtweise, die (Software-)Funktionen von Hardwarearchitektur und letztlich Bauteilen abstrahiert, und zu der notwendigerweise bauteilori-

Konsequenzen
aus der Funktions-
orientierung

² Für weiterführende Details zu AUTOSAR wird auf [KF08] verwiesen.

entierten Fehlerdiagnostik. Die Bauteilorientierung in der Diagnostik ist notwendig, da Software und damit Softwarefunktionen keiner Alterung im Sinne von Veränderungen unterliegt, ausgehend von einem ursprünglich funktionierenden System. Folglich sind es die Hardwarekomponenten, die Änderungen durch Alterung oder anderen äußeren Einflüssen unterliegen und daher notwendige und einzige Elemente der Hypothesen H_D sein müssen. Natürlich können, wird von einer an sich nie fehlerfreien Software ausgegangen, auch fehlerhafte Softwarefunktionen als Ursache eines Fehlers identifiziert werden, dennoch muss letztlich im Kontext der Offboard-Diagnostik im Service immer ein Steuergerät, also eine Komponente einem Softwareupdate unterzogen werden.

Konsequenzen für
die Fehlerdiagnostik

Zusammenfassend kann festgestellt werden, dass verteilte Funktionen, auch bedingt durch die damit erhöhte Fehlerfortpflanzung, nur noch schwer insbesondere von Einzelnen überschaubar und die heutigen bauteilorientierten Methoden ab einer gewissen Komplexität nicht mehr beherrschbar sind. Als Konsequenz sinkt die Diagnostikqualität mit der Anzahl der nicht erfassten Ursachen, nicht erfassten Fehler und der mangelnden Modellierung (Unvollständigkeit). Mit den angestrengten Betrachtungen wird deutlich, warum die heute eingesetzten Diagnostikmethoden zwangsläufig an ihre Grenzen stoßen müssen.

3.2. Stand der Technik und relevante Arbeiten

Im Folgenden werden verschiedene, zum Stand der Technik und der Forschung gehörende Verfahren analysiert und bewertet. Hierfür wurden Publikationen einer ganzen Reihe an Konferenzen untersucht. Eine Auswahl davon, die sich intensiver mit dem Thema Diagnostik in Fahrzeugsystemen beschäftigen, sind³

- SAE World Congress, Session Vehicle Diagnostics, Detroit [SAEhr]
- SAE Transactions Journal of Passenger Cars: Electronic and Electrical Systems [Mahch]
- Diagnose in mechatronischen Fahrzeugsystemen, München [Diahra]
- Diagnosesysteme im Automobil, Stuttgart [Diahrb]
- IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems (CIVVS)⁴, Nashville, TN, USA [CIVhr]

³ Eine Sortierung nach Relevanz wird hier nicht gegeben.

⁴ Der Workshop wurde 2009 zum ersten Mal abgehalten. On- und Offboard-Diagnostik sind explizit behandelte Themen.

- AUTOREG - Steuerung und Regelung von Fahrzeugen und Motoren⁵, Baden-Baden [AUThra]
- Elektronik im Kraftfahrzeug [EIKre], Baden-Baden⁶
- Mechatronik - Innovative Produktentwicklung⁷, Wiesloch bei Heidelberg [Mec]
- IEEE AUTOTESTCON - Systems Readiness Technology Conference⁸, Baltimore [AUThrb]

Weiterhin wurden auch eine große Zahl an Beiträgen von Konferenzen und Journals anderer Domänen, insbesondere im Data Mining und maschinellem Lernen ausgewertet. Im Folgenden werden die in Tabelle 3.1 gelisteten und aus Sicht des Autors als besonders relevant eingestuften Beiträge vorgestellt und bewertet. Die Auswahl erfolgte dabei anhand der in Kapitel 2.3 vorgestellten Wissensrepräsentationsformen und Wissensentstehungsarten. Zu jeder Wissensrepräsentationsform werden zunächst allgemeine Vor- und Nachteile diskutiert und dann die jeweiligen Beiträge vorgestellt.

3.2.1. Wissensmodellierung durch Experten

Analyse von erfahrungsbasiertem Wissen

Die Modellierung von erfahrungsbasiertem Wissen durch Experten trifft die klassische Definition von Expertensystemen. Die älteste Form ist dabei die manuelle Definition einer Regelbasis, wie z.B. das System *MYCIN* aus den 80ern [BS84]. Auch noch heute werden fast ausnahmslos Expertensysteme mit manuell erstellten Wissensbasen verwendet [Nau04, Tabelle 7.1]. Am Beispiel der heute in vielen OEMs eingesetzten GFS, einem regelbasierten Expertensystem, werden nachfolgend die Stärken und Schwächen solcher Systeme diskutiert.

Wie die Erörterungen in Kapitel 3.1 zeigen, liegen die Stärken von Expertensystemen im Bereich von überschaubaren, separierbaren Teilsystemen bzw. Komponenten, wie das lange Zeit für Fahrzeuge auch gegeben war. Eine ausführliche Untersuchung und Bewertung der GFS wurde bereits in [Tap04, K.7, S.94] in Zusammenarbeit mit der Audi AG durchgeführt, deren Ergebnisse die nachfolgenden Betrachtungen

Expertensystem –
Geführte Fehler-
suche

⁵ Die meisten diagnoserelevanten Vorträge beziehen sich jedoch auf die Onboard-Diagnostik.

⁶ Die Tagung deckt einen großen Themenbereich der Fahrzeugelektronik ab. Daher keine Spezialisierung auf Diagnostik.

⁷ Die Tagung deckt einen großen Themenbereich der Mechatronik ab. Daher keine Spezialisierung auf Diagnostikthemen.

⁸ Der Fokus der Konferenz liegt überwiegend auf militärischen Anwendungen im Luftfahrtbereich. In mehreren Sessions zum Thema Diagnostik werden jedoch gezielt Diagnostikthemen behandelt. Der Untertitel der Konferenz 2007 war "Transforming Maintenance: Closing the Loop Between ATE and Integrated Diagnostics"

<i>Arbeit/ Publikation</i>	<i>Wissens- repräsentation</i>	<i>Wissens- entstehung</i>	<i>Bemerkung / Stichpunkte</i>
[Nau04], [Tap04]	erfahrungsbasiert/ regelbasiert	manuell	Geführte Fehlersuche
[AD05a], [AD05b]	erfahrungsbasiert/ regelbasiert	manuell/ automatisierte Transformation	Dynamischer Fehlerbaum, Zuverlässigkeitsanalyse, diagnostische Relevanz
[FJ05]	erfahrungsbasiert/ modellbasiert	manuell/ automatisierte Transformation	Modellgenerierung aus FMEA-Analyse
[ULPR05]	erfahrungsbasiert/ modellbasiert	manuell	Funktionsorientierte Dia- gnostik & Prozessüberwa- chung
[SRB ⁺ 02, Str02b, SP04]	modellbasiert	manuell	Produkt OCC'M der Firma Raz'r. Es werden qualita- tative Modelle eingesetzt
[HS00, LW07]	modellbasiert	manuell	RODON: Verwendung von quantitativen Modellen
[Abu03]	fallbasiert	manuell	e:Diagnostic Life Cycle Suite: CBR-System
[BCF ⁺ 08]	fallbasiert/ erfahrungsbasiert/ modellbasiert	manuell	EU-Forschungsprojekt SM- MART: hybrider Ansatz
[KBL ⁺ 07], [KBM08], [KLFM09]	modellbasiert	automatisiert	Bayes'sche Netze aus Ent- wicklungsdaten (überwie- gend Bordnetz) & Aus- fallwahrscheinlichkeiten

Tabelle 3.1.: Ausgewählte Arbeiten der Domäne Fahrzeugdiagnostik und deren Klassifikation.

stützen. Die enge Zusammenarbeit mit der Audi AG garantiert dabei die Qualität und die Relevanz der Bewertung.

Diversität und
Variantenvielfalt

Zu den besonderen Schwierigkeiten heute eingesetzter regelbasierter Expertensysteme, wie z. B. die GFS, zählt die Abdeckung aller Varianten eines Fahrzeuges, welche sich durch die zahlreichen Ausstattungsmöglichkeiten ergeben [Tap04, S.28]. Dies ist bei der Erstellung der Fehlersuchprogramme zu berücksichtigen. So äußert sich derselbe Fehler in den unterschiedlichen Fahrzeugvarianten jeweils in anderer Art und auch Prüf- und Reparaturanweisungen können stark variieren. Bereits im Voraus alle möglichen Fehlerquellen zu erkennen und abzudecken, ist eine weitere Schwachstelle solcher manuell erstellter Fehlersuchprogramme. Auch wenn von erfahrenen Experten eine ausführliche FMEA durchgeführt wird, werden nie alle möglichen Fehlerarten und -quellen identifiziert werden können. Tappe sagt in [Tap04, S.20] hierzu „Durch

die heutige Fahrzeugsystemvernetzung kann eine eindeutige Aussage zur Korrelation von Fehlerauswirkung und Fehlerursache nicht mehr sofort getroffen werden.“ Naumann formuliert in [Nau04, S.105] noch etwas deutlicher: „[...] sind die Autoren, [...], in Zukunft kaum mehr in der Lage, die Komplexität auf herkömmlichen Weg in den Griff zu bekommen.“

Ein wichtiger Schritt zur Verbesserung der Diagnostikqualität ist die Möglichkeit, die Vorgänge im Feld auszuwerten, wie z.B. durch die Einführung einer Datenbank für Felddaten. So werden z. B. die für einen Reparaturfall relevanten Daten wie Ereignisspeichereinträge usw. aufgezeichnet. Mit den heute verfügbaren Informationen werden bereits Auswertungen zur Verbesserung der Prüfabläufe durchgeführt. Dieser Vorgang der Auswertung, wie auch die anschließende Verbesserung der Fehlersuchprogramme, wird jedoch manuell vorgenommen und lässt sich nicht mit vertretbarem Aufwand automatisieren. Daher kann über diese Vorgehensweise nur eine geringe Anzahl an Prüfprogrammen verbessert werden. Weiterhin vergeht einige Zeit, bis ein in den Werkstätten festgestellter Fehler eines GFS-Programms korrigiert und die korrigierte Version wieder verbreitet wird [Tap04, S.54]. Es kann sich hier durchaus eine Latenzzeit von einigen Monaten ergeben.

Rückführung von
Felderfahrung auf-
wändig

Damit verbunden bildet sich ein weiterer Nachteil im enormen (auch personellen) Aufwand, die Wissensbasis zu warten bzw. auf dem aktuellen Stand zu halten [Tap04, S.44], heraus. Als Beispiel sollen hier neue Ausstattungsvarianten genannt werden, für die theoretisch alle Prüfprogramme dieses Fahrzeugs überprüft werden müssen, um sicherzustellen, dass keine neuen oder veränderten Abhängigkeiten entstehen. In der Praxis ist das in dem Umfang nicht realisierbar, sondern wird auf die Untersuchung einiger weniger als relevant vermuteter Programme beschränkt. Ein weiteres Beispiel sind sogenannte *Fehlerabstellmaßnahmen*, die im Fall von Qualitätsmängeln eines Bauteils eingeleitet werden und häufig in einem veränderten Bauteil mit einer neuen Teilenummer bestehen. Auch hier können neue und veränderte Abhängigkeiten entstehen, so dass eine Anpassung von Prüfprogrammen notwendig werden kann.

Wartungsaufwand

Die genannten Punkte führen in der Praxis zwangsläufig zu Kompromissen, da das Berücksichtigen aller möglichen Fehlerquellen sowie aller Fahrzeugvarianten nur mit erheblichem Aufwand beherrschbar ist. Es soll hier angemerkt werden, dass bereits jetzt schon je nach OEM ca. 30 Experten für die Erstellung, Aktualisierung und Wartung der Wissensbasis eingesetzt werden [Nau04, S.112]. Insbesondere der bereits genannte Anstieg an Komplexität als auch die Tendenz zu verteilten Funktionen führen damit vermehrt zu reduzierten und unspezifischeren Fehlersuchprogrammen (z.B. „Suche nach Stromlaufplan“). Weiterhin können die Effekte einer umgreifenden Fehlerfortpflanzung zu einer ganzen Reihe an Prüfprogrammen führen (für jeden DTC existiert ein eigenes Fehlersuchprogramm), von denen nicht zwangsläufig eines zu der Identifizierung

Kompromisse

der tatsächlich defekten Komponente führt.

Stand der Forschung

Eine Untersuchung aktueller Entwicklungen im Bereich der regelbasierten Verfahren zeigt etliche neue innovative Ansätze. Einige aus Sicht des Autors relevanten werden nachfolgend vorgestellt.

Zuverlässigkeitsanalyse und diagnostische Relevanz

In [AD05a] und [AD05b] wird ein *dynamischer Fehlerbaum* (*Diagnostic Fault Tree (DFT)*) mit Hilfe einer Zuverlässigkeitsanalyse (reliability analysis) der Komponenten erweitert. Anschließend wird mittels eines „*Diagnostic Importance Factor (DIF)*“ die diagnostische Relevanz der Komponenten berechnet, auf Basis derer letztlich automatisiert ein hinsichtlich der DIF optimaler Diagnostikentscheidungsbaum (*Diagnostic Decision Tree (DDT)*), analog einem Prüfprogramm der GFS, erzeugt wird. Als Beispiel wurde eine Einspritzanlage gewählt. Während in [AD05b] der DIF mittels der Sensitivität⁹ berechnet wird, wird in [AD05a] hierfür der DFT zusammen mit der Zuverlässigkeitsanalyse zunächst in ein Markov Modell überführt und daraus die DIFs der Komponenten bestimmt. Anzumerken ist hierbei, dass die Kosten für die Überprüfung einer Komponente nicht mit einbezogen werden, genauso wenig wie auf die Durchführung/Anleitung der Überprüfung selbst eingegangen wird. Weiterhin muss auch hier der dynamische Fehlerbaum sowie die Zuverlässigkeitsanalyse von Experten manuell erstellt werden. Nichtsdestotrotz erscheinen die Einbeziehung von Zuverlässigkeitsinformationen, der diagnostischen Relevanz sowie die automatisierte Generierung von DDTs als innovativer Ansatz. Abbildung 3.3 zeigt in a) den DFT der Einspritzanlage (hier mit CRFS abgekürzt), in b) die daraus resultierenden DIFs und in c) den letztlich erzeugten DDT.

Modellgenerierung aus FMEA

In [FJ05] wird von einer ohnehin in der Entwicklung notwendigen FMEA ausgegangen, die um die Attribute *Node No.* (entspricht der Steuergerätadresse), *Fault Code* (entspricht dem DTC) und *Corrective Action* erweitert wurde. Als Medium wird hier, Kraftfahrzeug-konform pragmatisch, von einer großen Microsoft Excel-Tabelle ausgegangen (vgl. Abbildung 3.4). Aus der FMEA wird anschließend automatisiert ein (abduktives) Modell generiert, das im Wesentlichen aus den identifizierten Abhängigkeiten zwischen Fehler, DTC und Fehlerbehebung besteht. Der Prozess des diagnostischen Schließens besteht vereinfacht daraus, bei gegebenen DTCs alle assoziierten Fehler in einer nach DTC-Überdeckung sortierten Liste zu ermitteln.

Anzumerken ist hierbei, dass im Gegensatz zu [AD05a, AD05b] in die Gewichtung der Fehler keinerlei Ausfallwahrscheinlichkeiten o. ä. eingehen. Dennoch scheint es innovativ, aus ohnehin bestehenden FMEA-Analysen, die weiterhin Bestandteile der Diagnostiklastenhefte sind, automatisiert Modelle für die Fehleridentifizierung zu erstellen, so dass auch mit multiplen Symptombildern umgegangen werden kann.

⁹ Die Sensitivität einer Komponente wird ebenfalls aus der Zuverlässigkeitsanalyse ermittelt.

Abbildung 3.3.: Ausschnitte aus [AD05b]: a) Der DFT der Einspritzanlage (hier mit CRFS abgekürzt). b) Die daraus resultierenden DIFs. c) Der erzeugte DDT.

FMEA Process														Action Results				
Item and Function	Potential Failure Mode	Potential Effects of Failure	Node No.	Fault Code	Corrective Action	Severity	Potential Cause(s) of Failure	Occurrence	Current Controls	Detection	RPN	Recommended Action	Resp.bility	Action Taken	Severity	Occurrence	Detection	RPN
Power Circuit	High Resistance in FL Headlight circuit	Reduced FL Headlight	1	001	Check local wiring circuit and repair as necessary	6	Wear of wire	2		2	24							
	High Resistance in FR Headlight circuit	Reduced FR Headlight	1	002	Check local wiring circuit and repair as necessary	6	Wear of wire	2		2	24							
Power Circuit (contd.)	Low Battery at Front ECU	Front ECU not operating correctly	1	001	Check for clean contacts with battery else replace battery	6	Age of battery	3		2	36							
			1	002		6	Age of battery	3		2	36							
			1	003		6	Age of battery	3		2	36							
			1	004		6	Age of battery	3		2	36							
			1	005		6	Age of battery	3		2	36							

Abbildung 3.4.: Ausschnitt aus einer erweiterten FMEA [FJ05].

Ein weiterer innovativer Ansatz wurde in [ULPR05] entwickelt. Ausgehend von der diskutierten Herausforderung der Wandlung von der bauteilorientierten zur funktionsorientierten Sicht, wird hier konsequent eine funktionsorientierte Prozessüberwachung in den Steuergeräten vorgeschlagen. Funktionen werden dazu in Funktionsszenarien dekomponiert, für die jeweils alle beteiligten Komponenten sowie die hierfür nötigen Anfangs- und Endbedingungen der beteiligten Signale bekannt sein müssen. Im Fall einer Abweichung in einem Funktionsszenario werden alle damit assoziierten Komponenten verdächtigt. Durch die Überdeckung der Komponenten mit als für gut befundenen Funktionsszenarien wird eine Einschränkung auf wenige verbleibende Komponenten vorgenommen.

Funktionsorientierte
Diagnostik

Der Ansatz ist damit geeignet, um verteilte Funktionsnetzwerke zu modellieren und zu diagnostizieren. Anzumerken bleibt jedoch, dass die Modellierung der Funktionsabhängigkeiten, der Funktionsszenarien und deren Komponentenzuordnungen sowie Anfangs- und Endzustände nicht zu komplex sein dürfen und daher manuell durch den Entwickler durchgeführt werden muss. Weiterhin ist die funktionsorientierte Prozessüberwachung nicht kompatibel mit heute eingesetzten Steuergeräten, die mit einer komponentenorientierten Prozessüberwachung/Fehlererkennung ausgestattet sind. Dennoch könnte diese Sichtweise gerade hinsichtlich des kommenden Standards AUTOSAR interessant werden.

Die Diskussion der Verfahren zeigt, dass zwar innovative Ansätze existieren, die durchaus die Qualität der heute eingesetzten Verfahren übertreffen könnten, diese aber nach wie vor auf von Experten generierten Fehlersuchbäumen, FMEAs, etc. aufsetzen und damit die diskutierte Herausforderung der Überschaubarkeit der Komplexität durch die Experten nicht gelöst haben. Der Sachverhalt wird in [BH98] treffend formuliert: „*The efficacy of this approach obviously depends on the faithfulness of the encoding as well as the quality of experts' domain experience*“.

Zusammenfassung
Expertensysteme

Analyse modellbasierter Verfahren

Modellbasierte Diagnostikverfahren wurden bereits in einer Vielzahl an Anwendungen verschiedenster Domänen erprobt¹⁰. Die wesentlichen Nachteile modellbasierter Verfahren bestehen zum einen in der manuellen Erstellung der Wissensbasis, für die die Systemzusammenhänge und -Strukturen bekannt sein und geeignet modelliert werden müssen und gleichzeitig ein geeigneter Detaillierungsgrad gefunden werden muss¹¹. Gerade bei mechatronischen (Teil-)Systemen wie z. B. Motor und Abgas-system (Verbrennungsprozesse) sind oftmals nicht alle Zusammenhänge bis zu einem beliebigen Detaillierungsgrad bekannt. Zum anderen ent-

Herausforderung
Modellierungs- und
Rechenaufwand

¹⁰ Ein grundlegendes Verständnis modellbasierter Verfahren wird für diesen Abschnitt vorausgesetzt. Eine ausführliche Behandlung der Thematik sowie der verfügbaren Algorithmen wird beispielsweise in [Ise06] gegeben.

¹¹ Diese Wissensrepräsentationsform wird in [Hus01] als („*tiefes*“) kausales Wissen bezeichnet.

Modellierungsun- genauigkeiten- und -Fehler

stehen Nachteile in der rechenintensiven Anwendung größerer Modelle. Ein vollständiges Fahrzeug mit allen Systemen kann letztlich weder modelliert noch gerechnet werden. Auch kleinere Modelle sind teilweise nur mit hohem Ressourceneinsatz rechenbar. Hinsichtlich der diagnostischen Leistungsfähigkeit ist es bereits bei kleineren Fehlern oder Ungenauigkeiten im Modell möglich, dass nicht reale Fehler, sondern Fehler im Modell erkannt werden. Weiterhin können Modelle dieselben Schwachstellen besitzen wie das reale System, das modelliert wurde, und dadurch Fehler nicht erkannt werden.

In [Tap04] wird der Sachverhalt wie folgt treffend formuliert:

Allerdings wird an dieser Stelle auch schnell ersichtlich, dass der Aufwand extrem zunimmt und die Modellierung selbst von einfachen Schaltungen hier schwer möglich wird. Zum Beispiel kommt der modellbasierte qualitative Ansatz bei einer komplexeren Parallelschaltung wie der Wheatstoneschen Messbrücke bereits an ihre Grenzen. Da die modellbasierte qualitative Modellierung keine kontextfreie Modellierung ist, kann sie derzeit mit den vorhandenen Mitteln deshalb nicht verwendet werden. [Tap04, S.44]

OCC'M und RO- DON

Dennoch wurde beispielsweise in [SRB⁺02, Str02b, SP04] durch die Firma *OCC'M* aus München mit *Raz'r* ein Ansatz zur modellbasierten Fehlersuche mit qualitativen Modellen verfolgt, während in [HS00, LW07] durch die Firma *R.O.S.E. Informatik* mit *RODON* ein System mit quantitativen Modellen entwickelt wurde. Beide Ansätze wurden bereits in [Tap04] untersucht und bewertet. Beiden Verfahren wurde die Herausforderung der Datenbereitstellung und -Modellierung sowie der notwendige Rechenaufwand für die Anwendung bescheinigt. Im Vergleich zur GFS, wurden beide Verfahren schlechter bewertet.

Zusammenfassung Modellbasiert

Verallgemeinert kann etwas plakativ formuliert werden, dass genau mit dem Beginn von Software (in Steuergeräten) die Modellierungsfähigkeit an ihre Grenzen stößt, nicht nur aus zulieferer-/unternehmenspolitischer Sicht, sondern auch aufgrund der erörterten Komplexitäts-, Modellierungs- und Anwendungsherausforderungen. Inwieweit eine vollständig modellbasierte Diagnostik einmal möglich sein wird, ist hier noch nicht absehbar. Mit der zunehmend modellbasierten Entwicklung von Software und der nach wie vor rapide steigenden Rechenleistung könnte dies jedoch in ferner Zukunft möglich sein, sofern noch einige weitere unternehmenspolitische Herausforderungen zu IP-Fragen¹² usw. ebenfalls gelöst werden.

¹² IP steht für Intellectual Property.

Analyse fallbasierter Verfahren

Während in modellbasierten Verfahren Wissen explizit aus Konstruktions-/Entwicklungsdaten erstellt wird, wird *fallbasiertes* Wissen implizit mittels Musterfällen modelliert¹³. Unter der Prämisse der Wissensmodellierung durch Experten führt dies zu der Klasse des sogenannten fallbasierten Schließens (engl. case-based reasoning (CBR)). Ein Musterfall ist dann definiert als „*a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner*“ [Kol93]. Ein Fall (engl. case) kann damit betrachtet werden als eine vergangene Situation, die aus mindestens einer Problembeschreibung, also einer Menge an Symptomen S , und einer zugehörigen Lösung, nach Gleichung 2.7 also eine Menge an instantgesetzten Komponenten K , besteht. Die Fälle sind in einer Fallbasis gespeichert und geeignet indiziert, um einen effizienten Zugriff zu gewährleisten. Gegeben einer neuen Situation mit bekannter initialer Symptommenge S_I , werden durch einen „Reasoner“ und einer geeigneten Matchingfunktion passende bekannte Fälle aus der Fallbasis identifiziert. Ein Fall wird aus dieser Menge dann als die „Lösung“ ausgewählt, wobei die Auswahl auf verschiedenste Weisen erfolgen kann (die beste Übereinstimmung, größte generelle Ähnlichkeit, etc.). Falls die neue Situation nicht exakt mit dem (ausgewählten) Fall übereinstimmt, kann der ausgewählte Fall auch entsprechend verändert werden, ein Vorgehen, das Adaption genannt wird¹⁴.

Ein gerade für große Fallsammlungen bedeutender Faktor ist die Organisation der Daten. Die einfachste, eine „*flat memory organisation*“, ist eine flache Auflistung aller Fälle (z. B. eine Tabelle in einer relationalen Datenbank), wobei die Matchingfunktion seriell alle verfügbaren Fälle vergleicht, um die passenden zu identifizieren. Weil serielle Suchfunktionen für große Fallsammlungen sehr zeitaufwändig sein können, werden in der Praxis oft andere Organisationsweisen basierend auf Indizierung oder Hash-Tabellen verwendet. Alternativ können auch parallele Suchfunktionen, die passende Hardware vorausgesetzt, eingesetzt werden. Weiterhin sind Clustermethoden möglich, die ähnliche Fälle gruppieren (wobei Ähnlichkeit durch eine passenden, möglicherweise domänenabhängige Metrik definiert ist) und in einer hierarchischen Struktur organisieren. Dies kann zu deutlichen Beschleunigungen führen, besitzt aber Nachteile in der Erstellung oder Reorganisation der Fallbasis.

Die Matchingfunktion bestimmt entscheidend, wie das gespeicherte Wissen genutzt wird, und ist damit die vermutlich kritischste Komponente eines CBR-Systems. In die nötigen Matchingregeln fließt häufig heuristisches Expertenwissen über die Domäne mit ein, so kann z. B.

Datenorganisation

Matchingfunktion

¹³ Diese Wissensrepräsentationsform wird auch in [Hus01] konsistent als implizites Wissen deklariert.

¹⁴ Für eine detaillierte Einführung in CBR-Systeme wird auf [Kol93] oder [BAB⁺03] verwiesen.

der Vergleich zweier unbekannter Attribute zu einem schwachen Match, der Vergleich von bekannten, gleichen Attributwerten zu einem starken Match und der Vergleich konträrer Attributwerte zu einem negativen Match bestimmt werden. Aufgrund der entscheidenden Rolle der Matchingfunktion werden in der Praxis durchaus sehr viel komplexere Matchingregeln verwendet, die nicht nur jeweils einen Fall, sondern eine ganze Reihe ähnlicher Fälle betrachten und dabei den Informationsgehalt, statistische Wahrscheinlichkeiten, weitere Heuristiken usw. mit einbeziehen, wie z. B. auch [Zsc08], [BAP02], [PMT04] oder [BAB⁺03] zeigen.

Akquisitions-
methode

Bisher scheint die Einordnung in den Abschnitt *Wissensmodellierung durch Experten* zu suggerieren, dass Fallbasen von CBR-Systemen zu eben dieser Wissensart gehören. Tatsächlich ist das jedoch abhängig von der Akquisitionsmethode der Fallbasis. Zum einen können Fälle durch Expertenteams erdacht oder aber zum anderen reale Fälle in die Fallbasis aufgenommen werden. Auch Kombinationen aus beiden Methoden oder eine Art „*Bereinigung*“ von realen Fällen durch Experten, wie sie z. B. durch [Tap04] vorgeschlagen wird, sind möglich. So kann eine Fallbasis, wenn sie rein aus realen Fällen besteht, durchaus auch in die Kategorie des „*maschinellen Lernens*“ eingeordnet werden. Der Standard-CBR-Prozess sieht hier vor, dass für jedes unbekannte Problem ein neuer Fall in die Fallbasis aufgenommen wird, so dass ein solches Problem in Zukunft gelöst werden kann.

Stand der For-
schung

Konferenzen wie die ECCBR¹⁵ [ABMH08] zeigen, dass CBR-Systeme nach wie vor einen aktiven Forschungsbereich darstellen. Daneben wird auch in [GHP05] eine gute Übersicht über relevante Literatur zum Thema gegeben. Um den aktuellen Stand der Forschung kurz aufzuzeigen, sollen im Folgenden einige ausgewählte Ansätze und Realisierungen von CBR-Systemen vorgestellt werden.

e:Diagnostic Li-
fecycle Suite

Ein bekanntes CBR-System wird mit der *e:Diagnostic Lifecycle Suite* (*e:DLS*) der *empolis GmbH* in [Abu03] vorgestellt. Ein Derivat davon ist bereits bei der Robert Bosch GmbH im Einsatz. Das System deckt alle Phasen des CBR-Zyklus, angefangen bei der Erstellung der Wissens- und Fallbasis, der Verbreitung in die Werkstätten, der Anwendung und des Feedbacks ab (siehe Abbildung 3.5). Die Wissensakquisition erfolgt zum einen durch das Erstellen von Prüfprogrammen für Komponenten und zum anderen durch manuell erstellte (erdachte) Fälle, die als Lösungen die zuvor erstellen Prüfprogramme enthalten. In der Anwendung wird je nach Matchingscore nacheinander ein nach Kosten, Verfügbarkeit, Komplexität und Erfolgsaussicht günstiges Prüfprogramm vorgeschlagen, solange bis der Fehler identifiziert wurde. Das Feedback erfolgt dabei manuell durch Experten und wird ebenfalls durch Experten in die Fallbasis eingepflegt.

¹⁵ ECCBR: Advances in Case-Based Reasoning, European Conference, Trier

Damit handelt es sich hier um einen Hybrid, bestehend aus einem CBR- und Expertensystem. Hierbei sei angemerkt, dass der Erstellprozess der Prüfprogramme sehr an die Vorgehensweise der GFS erinnert. Neu ist, dass anstelle der direkten Verknüpfung von DTC zu Prüfprogramm (vgl. Abbildung 2.13) ein CBR-System gesetzt wurde.

Fallbasen von CBR-System enthalten häufig nur einen (Muster-)Fall pro Problem, um die Anzahl an Fällen gering zu halten. In Fällen uneindeutiger Diagnostizierbarkeit, d. h. für ein und dieselbe Symptommenge existieren mehrere mögliche Lösungen, fehlen dann wertvolle Informationen über die Relevanz bzw. Wahrscheinlichkeit der Lösung, so dass eine im Mittel optimierte Fehlersuche nicht möglich ist. In [BAP02] wird hierfür das Konzept der *Diagnostic Profiles* eingeführt: Ein Diagnostic Profile P_d für eine Diagnose $d \in D_\Sigma$ ist dann definiert als ein Tupel $P_d = (F_{P_d}, D_{P_d}^{corr})$, wobei F_{P_d} ein Häufigkeitsprofil für die Diagnose d ist und die Menge $D_{P_d}^{corr}$ aller Diagnosen d' enthält, die zusammen mit d auftreten. Zusammen mit einer modifizierten Matchingfunktion und Distanzmetrik kann so nicht nur besser mit einer mangelnden Diagnostizierbarkeit umgegangen werden, sondern auch die Herausforderung mit Mehrfachfehlern zumindest entschärft werden, insofern wie für die auftretenden Fehler unterschiedliche Häufigkeiten bestehen. Das Verfahren wurde anhand einer Fallbasis aus dem medizinischen Bereich untersucht

Diagnostic Profiles
und multiple Fehler

Abbildung 3.5.: Das Diagnostic Lifecycle Modell nach [Abu03]

und brachte deutliche Verbesserungen in der Qualität der Auswahl der Fälle.

SMMART Projekt

Im Rahmen des EU-Forschungsprojektes SMMART¹⁶ wird in Zusammenarbeit mit VOLVO Trucks ein Hybrid-Ansatz verfolgt, bestehend aus einem CBR-, Fehlerbaum (FT)- und modellbasierten (MBR) System [BCF⁺08]. Dabei werden, wie in Abbildung 3.6 illustriert, mittels CBR zunächst anhand gegebener Symptome des vorliegenden Falls verdächtige „*Highlevelkomponenten*“ (HLCs) identifiziert. Anschließend kann der Servicetechniker selbst entscheiden, ob die Fehlerquelle innerhalb der verdächtigten HLC mittels FT oder MBR (hier wird das bereits erwähnte Rodon verwendet) erkannt werden soll. Ein Fall der Fallbasis wird dabei mit geeignet kodierten subjektiven Symptomen, DTCs, Fahrzeugdaten (Modell etc.) und Umgebungsbedingungen (Höhe, Temperatur etc.) sowie den dazugehörigen HLCs (es können mehrere sein, um einer evtl. mangelnden Diagnostizierbarkeit zu begegnen), Fehlerquelle(n) und verwendete Methode (FT oder MBR) modelliert. Nach einer erfolgreichen Reparatur kann der Servicetechniker selbst entscheiden, ob er den Fall der Fallbasis hinzufügen möchte oder nicht.

Anzumerken ist hierbei, dass die verwendeten Matchingscores nicht immer rechnerisch, sondern im Fall der DTCs manuell für jedes DTC-Paar durch Experten (Entwickler) erstellt werden sollen. Ferner ist die Kombination von CBR und einer „*Komponentenüberprüfungsmethode*“ bereits aus dem vorhergehend beschriebenen e:DLS der empolis GmbH ([Abu03]) bekannt, hier jedoch noch um ein MBR-System erweitert. Zwischen den beiden Methoden FT und MBR findet jedoch keine echte Integration statt. Die Entscheidung der geeigneten Methode dem Servicetechniker zu überlassen ist zumindest fraglich. Weiterhin bemerkenswert ist es, die Entscheidung des Hinzufügens neuer Fälle ebenfalls dem Servicetechniker zu überlassen. In wieweit sich dieses Vorgehen bewährt, muss die angekündigte Praxiserprobung in den Werkstätten von VOLVO zeigen.

Zusammenfassung CBR-Systeme

CBR-Systeme unterscheiden sich signifikant von den vorhergehend diskutierten wissensbasierten Experten- und modellbasierten Systemen. Das Diagnostikwissen ist weder als explizites Modell der Domäne noch in der Form von qualitativen, heuristischen Regeln, sondern implizit in Form von Fällen repräsentiert. In die Matchingfunktionen fließen wiederum Heuristiken aus problemspezifischem Wissen mit ein (z.B. für Ähnlichkeitsmetriken usw.). CBR-Systeme haben insofern die Fähigkeit (wenngleich begrenzt), Änderungen innerhalb einer Domäne zu adaptieren, wie solche Adaptierungen einfach aus einer Erweiterung/Änderungen der Fallbasis resultieren können. Hier sei angemerkt, dass für hierarchisch organisierte Fallbasen Änderungen in einzelnen Fällen eine vollständige Reorganisation der Fallbasis notwendig machen kann.

¹⁶ SMMART: System for Mobile Maintenance Accessible in Real Time, [SMM09]

Die Nachteile von CBR-Systemen sind u.a. der hohe Rechenaufwand verbunden mit der Such-/Matchingfunktion und der hohe Speicherbedarf bedingt durch die Organisation der Fallbasis. Eine Möglichkeit diese Nachteile zu umgehen ist, wie sie auch in [BH98] vorgeschlagen wird, beispielsweise die Verwendung von induktiven Lernverfahren, die aus der Fallbasis verallgemeinerte Regeln extrahieren, die die meisten Fälle der Fallbasis abdecken. Dies kann dann die Größe der Fallbasis soweit reduzieren, dass lediglich die bekannten Ausnahmefälle (der gelernten Regeln) verbleiben. Die Anwendbarkeit solcher Verfahren hängt letztlich von der Anwendungsdomäne und der Repräsentationsart der Fälle ab. Für den Kraftfahrzeug-Bereich ist z. B. die direkte (oder auch gefilterte) Verwendung von Reparaturfällen nicht sinnvoll, solange die Datenqualität und Zuverlässigkeit nicht stark steigt. Wie auch die Falldatenanalyse in Kapitel 5 zeigt, sind die Prozesse in den Servicewerkstätten nicht primär auf die Datenqualität, sondern vielmehr auf die Kundenzufriedenheit ausgelegt. Ein Ausweg wäre hier sicherlich die manuelle Nachbearbeitung der Fälle durch Experten, die jedoch aufgrund der großen Menge aufwändig sein würde. Mit zunehmender Filterung könnte zwar die Fallzahl reduziert werden, dafür gehen jedoch wichtige statistische Informationen verloren, die aufgrund einer teilweise mangelnden Diagnostizierbarkeit zwangsläufig von Bedeutung für eine im Mittel kostenminimierte Fehlersuche und -reparatur ist. Ein weiterer zentraler Aspekt von CBR-Systemen ist die Rückführung von Fällen, für die noch keine geeigne-

Nachteile:

- Rechenaufwand
- Speicherbedarf
- Robustheit

Hohe Qualität der Fallbasis notwendig

Abbildung 3.6.: Der von [BCF⁺08] vorgeschlagene Hybrid-Ansatz zur Diagnostik und Reparatur in Servicewerkstätten.

te Lösung bzw. kein geeigneter Fall existiert. Diese Rückführung kann aufgrund der bereits genannten Gründe nicht, wie in [BCF⁺08] vorgeschlagen, zuverlässig von Servicetechnikern realisiert werden. Die in [Tap04] oder [Zsc08] vorgeschlagene manuelle Modellierung von Fällen (durch Experten erdachte Fälle) umgehen zwar die Herausforderung der Datenqualität, bringen aber alle bereits für Expertensysteme und modellbasierte Verfahren diskutierten Herausforderung der Unüberschaubarkeit der Komplexität mit sich. Wurde ein Zusammenhang zwischen Symptomen und defekter Komponente nicht in Form eines Falls modelliert, kann dieser Fehler nicht gelöst werden. Eine Rückführung mit allen bereits aufgezeigten Problemen wäre zwangsläufig notwendig.

Herausforderung
Mehrfachfehler

Eine weitere bekannte Herausforderung ist der Umgang mit Mehrfachfehlern [PGPB96, S.136], da hier die Anzahl der möglichen Kombinationen stark steigt und damit die Wahrscheinlichkeit groß wird, dass kein geeigneter Vergleichsfall existiert (das hierzu in [BAP02] beschriebene Verfahren hat für die Bedingungen im Kraftfahrzeug-Bereich Grenzen). Auch wenn für die beschriebenen Nachteile sicher mehr oder weniger geeignete Lösungen gefunden werden könnten, zeigen die folgenden Abschnitte weitere auf Falldaten basierende innovative Lösungsmöglichkeiten auf, die CBR-Systemen überlegen sind.

3.2.2. Automatische Wissensakquisition

Die aufgezeigten Nachteile der manuellen Wissensmodellierung waren ein Grund, warum in den vergangenen Jahren ein Großteil der KI-Forschung in die automatisierte Wissensgewinnung insbesondere unter Verwendung von Verfahren des maschinellen Lernens fokussierte (vgl. auch [Gal93], [Mit97] und [Mac03]).

Ein entscheidender Faktor für den Einsatz solcher Methoden ist die Qualität und Verfügbarkeit maschinell verarbeitbarer Daten. Im Kraftfahrzeug-Bereich können diese unterteilt werden in

1. Konstruktions-/Entwicklungsdaten und
2. Reparaturfalldaten¹⁷.

Ausgewählte Verfahren beider Varianten werden nachfolgend vorgestellt.

Wissensakquisition aus Konstruktions-/Entwicklungsdaten

Konstruktions-/
Entwicklungsdaten

Bereits [ULPR05] geht ausführlich auf die Schwierigkeiten der automatisierten Verwendung von Daten ersterer Kategorie ein. Auch in [Tap04] werden Entwicklungsdaten verwendet. Beide Arbeiten machen jedoch

¹⁷ Eine Einteilung der Verfahren könnte hier anhand der Lernstrategie, der Wissensrepräsentation oder dem Anwendungsbereich gegeben werden [BKI08, S.98ff], [CMM84, S.3ff]. Wie jedoch schon in [BKI08, S.98] bemerkt, kann dies unter verschiedenen Blickwinkeln erfolgen. Auf eine ausführliche Diskussion hierüber wird auf denselben verwiesen und nachfolgend nur auf die für diese Arbeit relevanten Verfahren eingegangen.

deutlich, dass die Art der Daten eine automatisierte Verwendung für diagnostisches Schließen nicht erlaubt. Nicht unbedeutend ist hierbei die Tatsache, dass ein Großteil der Steuergerätesoftware von Zulieferbetrieben entwickelt wird, die den Quellcode aus nachvollziehbaren Gründen nicht zur Verfügung stellen wollen [KKL⁺08, S.192]. Eine Ausnahme bildet hierbei das Bordnetz, das nachwievor vom OEM selbst entwickelt wird. Die domänenspezifischen Besonderheiten erlauben eine automatisierte Verwendung der Daten, beschränkt jedoch auf das Bordnetz. Ein Projekt hierzu wird nachfolgend exemplarisch vorgestellt.

Im Rahmen des Volkswagen-Projekts „*Bereichsübergreifende Diagnostikkonzeptoptimierung*“, wurde die Verwendung von Konstruktions- und Entwicklungsdaten zur automatisierten Modellbildung untersucht. Mit der Einschränkung auf elektrische Fehler wurde hier ein Verfahren entwickelt, dass automatisiert aus einer fahrzeugindividuellen elektrischen Strukturbeschreibung (im Wesentlichen Bordnetz und Verbauliste) Bayes'sche Netze generiert, mit Hilfe derer eine dynamische Fehlersuche inklusive einer automatisierten Generierung von Prüfanweisungen möglich ist (Abbildung 3.7). Mit der Integration von Komponentenausfallwahrscheinlichkeiten, die ebenfalls automatisiert aus den Gewährleistungsdatensätzen ermittelt werden können, wurde ein quasimodellbasiertes Verfahren entwickelt, dass durch vollständige Automatisierung die Herausforderung der (manuellen) Modellerstellung umgeht. Die Anwendbarkeit mit heute verfügbaren Standardrechnern wird durch die Modellierung als Bayes'sches Netz in Kombination mit einigen Strukturoptimierungsverfahren gewährleistet. Für weitere Details wird auf [KBL⁺07], [KBM08] und [KLFM09] verwiesen.

Auch wenn in der Arbeit ein Verfahren mit Potenzial zur praktischen Anwendbarkeit entwickelt wurde, existiert die Einschränkung auf wohlbekannte automatisiert auswertbare, *elektrische* Strukturbeschreibungen. Nach wie vor wird die Funktionalität der Komponenten selbst nicht modelliert oder automatisiert erzeugt, sondern lediglich eine (manuelle) Zuordnung von DTC zum verantwortlichen Hardware-Pin verwendet. Weiterhin ist die Modellierung und Anwendbarkeit als Bayes'sches Netz nur aufgrund der vereinfachten Modellierung eines Kupferkabels als idealer Leiter möglich. Komplexere mechatronische Fehler können darum mit diesem Verfahren prinzipbedingt nicht identifiziert werden. Trotzdem erreicht das Verfahren für rein elektrische Fehler vielversprechende Ergebnisse und zeigt, dass eine automatisierte Verwendung von Konstruktions- und Entwicklungsdaten für diagnostisches Schließen sinnvoll möglich ist.

Bayes'sche Netze
aus Entwicklungsda-
ten

Einschränkung auf
elektrische Struktur-
beschreibungen

Abbildung 3.7.: Ausschnitte aus [KBM08]: **a)** Ausschnitt aus dem Stromlaufplan einer Klimaanlage. **b)** Das daraus resultierende Bayes'sche Netz.

Zusammenfassung:
Diagnostikwissen
aus Entwicklungsdaten

Es existieren einige Ansätze, wie aus Entwicklungsdaten automatisiert Diagnostikwissen generiert werden kann. Alle Ansätze haben jedoch harte Anforderungen an die Verfügbarkeit, Struktur und Art der Entwicklungsdaten, die heute allenfalls für Teilsysteme, jedoch nicht für ein Gesamtfahrzeug erfüllt werden können. Lösungsansätze hierfür könnten in ferner Zukunft ein zunehmend, hinsichtlich Automatisierung, durchgängigerer Entwicklungsprozess, wie er beispielsweise im AUTOSAR-Standard vorgeschlagen wird, sein. Jedoch wird die hierfür notwendige Umstrukturierung ihre Zeit brauchen. Zu bedenken ist weiterhin, dass heute gebaute Fahrzeuge über den gesamten Lebenszyklus diagnostiziert werden müssen und damit die Verwendung von Verfahren der ersten Kategorie für alle Fahrzeuge in ferne Zukunft rückt.

Automatische Wissensakquisition aus Reparaturfalldaten

Keine relevanten Verfahren vorhanden

Im Rahmen der Recherchen wurden keine Verfahren gefunden, welche im Kraftfahrzeugbereich eine automatisierte Wissensakquisition aus Reparaturfalldaten für diagnostische Zwecke realisiert, so dass hier kein Stand zur Forschung vorgestellt werden kann. Gründe hierfür könnten sein, dass Reparaturfalldaten erstens erst seit geraumer Zeit umfangreich aufgezeichnet werden und zweitens diese als hochsensibel und geheim eingestuft werden, so dass sie der breiten Forschergemeinde nicht zur Verfügung stehen (auch in diesem Projekt standen die Daten erst spät zur Verfügung).

3.3. Fokussierung und resultierende Forschungsaufgabe

Immer mehr OEMs (darunter auch Volkswagen) zeichnen die Vorgänge in den Servicewerkstätten in zunehmendem Umfang und steigender Qualität auf. Zu den Daten zählen nicht nur die Kundenbeanstandungen, wahrgenommene Symptome und DTCs sondern auch die durchgeführten Reparaturen (Arbeitspositionen, Ersatzteile, etc.)¹⁸, so dass sich daraus vollständige und für ein Lernsystem geeignete Datensätze ableiten lassen. Weltweit existieren beispielsweise für Volkswagen über 10.000 Werkstätten mit insgesamt mehr als 170.000 Mitarbeitern¹⁹, die allesamt Fahrzeuge des VW-Konzerns instantsetzen. Neben dem Potential, dass sich aus den Fakten vermuten lässt, existiert gleichzeitig bedingt durch die heute eingesetzten Diagnostiktester die notwendige Infrastruktur für die Aufzeichnung, Übermittlung der Daten und spätere Verbreitung/Anwendung des Diagnostikwissens.

Potential der Reparaturfalldaten

Fokussierung auf Reparaturfalldaten

Aus den skizzierten Gründen wird der Fokus in dieser Arbeit auf die zweite Kategorie, dem Lernen aus Reparaturfalldaten gesetzt.

Die Anforderungen an ein Diagnostiksystem mit dem Anspruch, bereits mit den heute existierenden Daten eingesetzt werden zu können, geben dabei auch die Rahmenbedingungen für das zu wählende Lernverfahren vor²⁰. Dessen Auswahl ist mit der Vielzahl an verschiedenen Algorithmen, die die KI-Forschung seit dem Beginn der Data-Mining-Ära hervorbrachte, nicht trivial und für den Erfolg entscheidend. Hier spielt die Art und Verfügbarkeit der Datenbasis eine wichtige Rolle. Mit der Falldatenanalyse in Kapitel 5 können die Daten zum einen in Symptom- und Lösungsinformationen und zum anderen in Kontextinformationen unterteilt werden²¹. Während Symptom- und Lösungsinformationen nominal dichotom (vorhanden/nicht vorhanden), unzuverlässig, teilweise fehlerhaft, uneindeutig (multilabel) und zur Kategorie „*sparse data*“ gehören (einzelne Symptome oder Komponenten treten mit $P(s_i) \approx 0,14\%$ sehr selten auf), sind Kontextinformationen multinominal dichotom ($\#\tilde{c}_{Version} \approx 32,8$), eindeutig und immer und zuverlässig verfügbar (Kategorie „*dense data*“). Dieser Dualismus der Falldaten stellt eine große Herausforderung für ein Lernsystem dar. Er verhindert nicht nur den direkten Einsatz bewährter Lernverfahren, sondern erfordert vielmehr neue flexible und adaptive Methoden.

Welches Lernverfahren?

Falldatencharakterisierung

¹⁸ Eine detaillierte Beschreibung und Analyse der Falldaten wird in Kapitel 5 gegeben.

¹⁹ Stand 2005

²⁰ vgl. Anhang A.2 für eine Zusammenfassung der Anforderungen.

²¹ Eine ausführliche formale Definition der Falldaten wird später in Kapitel 6 gegeben.

Lösungsansatz und verwandte Arbeiten

Es wurde bereits festgestellt, dass für den Bereich der Fahrzeugdiagnostik keine verwandten Arbeiten zum Lernen aus Reparaturfalldaten existieren. Wird die Domäne und der Anwendungsfall zunächst außen vor gelassen, eröffnet sich mit dem Bereich des maschinellen Lernens eine große Auswahl verschiedener Verfahren. Hier haben sich neuronale Netze als vielversprechender Lösungsansatz herausgestellt.

Künstliche neuronale Netze als Lösungsansatz

Künstliche neuronale Netze sind wegen ihren Eigenschaften wie Adaptivität, Lernen aus Beobachtungen oder Fehlertoleranz in diversen Anwendungen erfolgreich zum Einsatz gekommen. Auch wenn an dieser Stelle keine vollständige Liste gegeben werden kann und soll, so hilft die Kenntnis wichtiger Beispiele bei der späteren Abschätzung, wie sie zum Einsatz kommen können und welche Ergebnisse zu erwarten sind: Die Anwendungen erstrecken sich vom wissenschaftlichen bis zum kommerziellen Bereich, wie eine Recherche der Artikel in [CB09] oder [Pol09] sowie vorgestellte Anwendungen in [Dre05, S.55ff], [MAS03], [Smi02] und [GRF07] zeigen. Im Kraftfahrzeug-Bereich werden neuronale Netze bisher nur wenig eingesetzt (vgl. Ausführungen von [Pro08, S.101ff] hierzu). Gründe hierfür sind jedoch weniger die Leistungsfähigkeit sondern vielmehr der „*Black-Box*“-Charakter von neuronalen Netzen, ein gewisser Konservatismus der Branche und die Existenz vieler Legacy-Komponenten und -Verfahren, deren Modifikation oder Neuentwicklung mit Kosten und Risiken verbunden sind ([Chi97, S.94ff], [Kor05, S.110ff]).

Anwendung neuronaler Netze in vielen Bereichen

Neuronale Netze zur Klassifizierung

Für diagnostische Anwendungen, d. h. die Verwendung von neuronalen Netzen zur Klassifizierung, existieren eine Reihe von Untersuchungen und Anwendungen, für die neuronale Netze eine sehr gute oder die beste Performanz gezeigt haben. In [Kre01] wird beispielsweise systematisch die Eignung neuronaler Netze für Klassifikationsaufgaben untersucht, in [Pön05] findet ein Vergleich Bayes'scher und neuronaler Netze statt und in [Tag00, Tab.2.1] wird ein Vergleich symbolischer, evolutionärer und konnektionistischer Verfahren angestrengt, jeweils mit dem Ergebnis der Qualifikation neuronaler Netze. In [KS97], [Sun96] und [Pat08] werden neuronale Netze zur Klassifizierung für eine im Kraftfahrzeug-Kontext Onboard-ähnlichen Fehlerdiagnostik erfolgreich eingesetzt, wobei hier jedoch noch zusätzliche Signalverarbeitungsschritte und damit Klassifikationen metrischer Variablen beliebiger Verteilung erfolgen. Bezugnehmend auf die hier geforderte Offboard-Diagnostik bzw. der Klassifikation nominaler dichotomer Merkmalsvektoren, werden in [KKKC04, S.357ff] eine Reihe von Anwendungen vorgestellt, die auch hier die Eignung neuronaler Netze für die jeweilige Aufgabenstellung zeigen.

Leistungsfähigkeit der logistischen Regression

Mit der später in Kapitel 7.2 gezeigten Äquivalenz der logistischen Regression mit einstufigen neuronalen Netzen (vorausgesetzt bestimmter Randbedingungen) werden eine Reihe von Effizienz- und Leistungsfähigkeitsuntersuchungen interessant:

- In [LLS00] werden 33 verschiedene Klassifikationsalgorithmen aus den Bereichen entscheidungsbaumbasierte, regelbasierte, statistische und neuronale Verfahren hinsichtlich ihrer Rechenzeit und Klassifikationsleistung verglichen. Dabei erlangte die logistische Regression hinsichtlich Präzision Platz zwei.
- In [RK06] wurden die Klassifikationsleistung des naiven Bayes-Klassifikators und der logistischen Regression für verschieden große Datensätze (4.000 – 50.000) mit teilweise mehr als 100.000 Merkmalen und 8.500 Klassen bei gleichzeitig wenig Datensätzen verglichen. Die logistische Regression war besonders bei geringen Falldaten pro Klasse überlegen.
- In [PPS03] wurde eine ausführliche Analyse zur Lernkurve und Klassifikationspräzision von logistischer Regression und Entscheidungsbäumen für Falldaten verschiedener Größe durchgeführt, wobei Entscheidungsbäume zwar für kleinere Falldatengrößen, jedoch logistische Regression für größere Datensätze bessere Ergebnisse erzielte.
- In [EHN⁺98] wurde für medizinische Datensätze logistische Regression mit diversen adaptiven nicht linearen Lernverfahren verglichen. Keines der Verfahren erreichte bessere Ergebnisse als die logistische Regression.

Fokussierung auf neuronale Netze

Die Aufstellung zeigt, dass die logistische Regression und damit auch das neuronale Pendant zunächst für die Symptom- und Lösungsinformationen der zur Verfügung stehenden Falldaten sehr gut geeignet ist. Mit der Fähigkeit der flexiblen Adaption und Lösungsraumgestaltung konzentriert sich die resultierende Forschungsaufgabe dieser Arbeit damit neben einer tragfähigen Konzeptentwicklung auf die Adaption vorerst einstufiger logistischer neuronaler Netze auf die Aufgabenstellung der Fehlerdiagnostik im Kraftfahrzeug-Bereich nach dem in Abbildung 4.1 vorgestellten Konzept. Mit der Berücksichtigung der Kontextinformationen, die im Kontrast zur Sparsity der Symptomdaten stehen, wird dann eine Konstruktionsmethode zur schrittweisen Erweiterung des Lösungsraumes der neuronalen Netze erarbeitet, um so die Dualität der Falldaten gewinnbringend beherrschen zu können. Zur Forschungsaufgabe gehört weiterhin die Entwicklung und Anwendung einer geeigneten Evaluationsmethode zur Ermittlung der Leistungsfähigkeit des Konzepts und der neuronalen Modelle.

Resultierende Forschungsaufgabe –
Lernverfahren

4 Lernen aus Reparaturfällen

Das Kapitel versteht sich als Übersichtskapitel, in dem das Konzept zum Lernen aus Reparaturfällen grob skizziert wird, um später auf einzelne Bestandteile näher einzugehen. Vorbereitend dazu wird neben der Konzeptvorstellung eine verallgemeinerte Betrachtung von Reparaturfällen hinsichtlich den entstehenden Daten gegeben, bevor im nachfolgenden Kapitel die erste Vertiefung mit der Analyse und Aufbereitung der Falldaten erfolgt.

4.1. Einführung

Die Betrachtung und Bewertung von Verfahren zum Stand der Technik wie auch von ausgewählten Forschungsarbeiten hat aufzeigt, warum der Fokus in dieser Arbeit auf das Lernen aus Reparaturfällen gesetzt wird. Das Lernen aus Reparaturfällen wurde als eine Lösung auf die Herausforderung der steigenden Komplexität in der Offboard-Diagnostik ermittelt. Eine Charakterisierung der zu erwartenden Reparaturfalldaten und eine Ableitung geeigneter Verfahren wurde durchgeführt. An dieser Stelle soll nun das Konzept zum Lernen aus Reparaturfällen in den Gesamtkontext gestellt und ein ganzheitlicher Prozess hierfür beschrieben werden, bevor in den nachfolgenden Kapiteln detaillierter auf einzelne Bestandteile eingegangen wird.

Geschlossener
Feedbackprozess

Gelernt werden sollen Diagnostik-Modelle zur Offboard-Diagnostik in den Servicewerkstätten. Die Voraussetzung zum Lernen solcher Modelle sind Reparaturfalldaten aus den Werkstätten. Hierzu werden in Abbildung 4.1 formal alle Prozesse innerhalb der Werkstätten unter *Reparatur-Prozesse* zusammengefasst. Eine Auswahl der dabei entstehenden Daten, gemeint sind hier vor allem Reparaturfalldaten, werden in ein Data Warehouse übertragen. Aus dem Datenbestand wird nun ein Lernsystem gespeist, welches aus der Vielzahl der Daten ein diagnostisches Modell Π_D nach (2.6) erzeugt. Diese Modelle werden anschließend als Diagnostikwissen in einer Datenbank gespeichert, welche die Daten an die Werkstätten zur Verwendung in den Diagnostiktestern verteilt. Wie Abbildung 4.2 veranschaulicht, werden dort in einem konkreten Reparaturfall zunächst Symptome und Fahrzeuginformationen über die Diagnostik-Schnittstelle des zu diagnostizierenden Fahrzeugs ausgelesen und zu Merkmalen aufbereitet. Die Modelle generieren anschließend Diagnostik-Hypothesen, die nach einer Aufbereitung bzw. Anreicherung mit weiteren Informationen den KD-Mitarbeiter in einer geeigneten Weise zur Verfügung gestellt werden, um ihn bei der Fehlersuche zu

unterstützen. Die Ergebnisse bzw. Daten der Reparatur werden wieder zurück an das Data Warehouse übertragen.

Es handelt sich somit um einen geschlossenen Prozess, dessen Rückkopplung weitere Eigenschaften ausbildet. Hierdurch entsteht für bisher unbekannte Reparaturprobleme durch die Gesamtheit der KD-Mitarbeiter in tausenden von Servicewerkstätten ein automatischer, iterativer Lösungsprozess. Dieser soll nachfolgend näher beschrieben werden.

Abbildung 4.1.: Datenflussdiagramm des Feedbackprozess

Angenommen, in einer Werkstatt tritt eine neue, bisher in den Datenbeständen unbekannte Reparaturaufgabe (z. B. der Defekt der Komponente k_x) auf, so besteht für dieses erste Auftreten im erlernten Modell kein Diagnostikwissen. Eine zielgerichtete Diagnostik würde wegen $k_x \notin h_D \forall h_D \in H_D$ nicht möglich sein. Der betreffende KD-Mitarbeiter müsste den Fall dann mit den ihm vorhandenen Mitteln lösen (z. B. Fehlersuchprogramme etc.). Weiterhin wird davon ausgegangen, dass in dieser ersten Lösung mehrere, auch z. T. nicht erfolgreiche Reparaturmaßnahmen durchgeführt werden, die anschließend in den Datenbestand einfließen. Im folgenden Schritt würde das Lernsystem erneut aus den Datenbeständen des Data Warehouse lernen und das Diagnostikwissen optimieren. Würde nun dieses Problem in einer anderen Werkstatt erneut auftreten, so könnte das Diagnostiksystem aufgrund der veränderten Modelle bereits eine zielgerichtete Diagnose stellen, d. h. $\exists h_D$ mit $k_x \in h_D, h_D \in H_D$. Diese könnte aufgrund der Datenlage jedoch noch unscharf sein. Der betreffende KD-Mitarbeiter wird also im Regelfall eine Auswahl von den vorgeschlagenen Diagnose-Hypothesen h_D treffen müssen. Wäre die „Reparaturlösung“ dabei Teil der Diagnosen, so würde die Menge der durchgeführten, nicht erfolgreichen Reparaturmaßnahmen im Mittel kleiner. Bereits nach ein paar wenigen Iterationen würde so

Automatischer,
iterativer Lösungs-
prozess

automatisch die beste Reparaturlösung für das neue Problem gefunden. Nach dem gleichen Funktionsprinzip würden auf diese Weise nicht nur neue Lösungen gefunden, sondern auch Veränderungen automatisch adaptiert werden.

Abbildung 4.2.: Anwendung der Diagnostik-Modelle

Anforderungen an das Konzept

Im Anschluss an die Konzeptidee wurde schon sehr früh im Projekt eine Anforderungsanalyse durchgeführt, um eine leistungsfähige Realisierung zu gewährleisten. Diese wurde bereits im Projektbericht [Mül08, S. 12ff] vorgestellt sowie später in [Kriar, Kap. 4.1 u. 5.1] diskutiert und soll hier nicht wiederholt werden. Eine Zusammenfassung ist im Anhang A.2 zu finden.

Gliederung der Konzeptbeschreibung

Im Folgenden werden die einzelnen Punkte des Konzepts, beginnend mit der Betrachtung der Reparaturprozesse in den Service-Werkstätten zusammen mit der Modellierung der Reparaturfalldaten in Abschnitt 4.2 und einer Analyse der datenerzeugenden Prozesse in Abschnitt A.4 ausführlich behandelt. In Kapitel 5 erfolgt die für Lernverfahren wichtige Analyse und Aufbereitung der Falldaten. In Kapitel 6 werden zunächst statistische diagnostische Modelle vorgestellt und darauf aufbauend drei neuronale Diagnostik-Modelle entwickelt. Hierzu wird in Abschnitt 7.1 zunächst eine kurze, für das weitere Verständnis notwendige Einführung in neuronale Netze gegeben, bevor in Abschnitt 7.2 das erste einschichtige neuronale Diagnostik-Modell hergeleitet wird. Die besondere Struktur der Falldaten erfordert zur Berücksichtigung von Fahrzeuginformationen wie Steuergeräteversionen etc. spezielle Konstruktions- und Trainingsverfahren, die mit dem MLP-Diagnostik-Modell in Abschnitt 7.3 und schließlich mit dem komplexitätserweiterten Diagnostik-Modell in Abschnitt 7.4 beschrieben werden. Um die Adaptivität der Modelle weiter

zu erhöhen, wird in Abschnitt 7.5 ein Verfahren zum Verlernen alter Reparaturfälle vorgestellt und ein entsprechender Vergleich gegeben. Aufgrund der großen Datenmengen und der hochdimensionalen Merkmalsräume werden für die praktische Umsetzung Optimierungsverfahren hinsichtlich des Ressourcenbedarfs notwendig, die in den darauffolgenden Abschnitten mit der Merkmals-Reduktion (beschrieben in Kapitel 7.6) und dann Pruning (Kapitel 7.7) beschrieben werden.

Damit wurden die theoretischen Betrachtungen abgeschlossen, so dass in Kapitel 8 die Umsetzung und Anwendung in den Werkstätten beschrieben werden kann. Eine mehrstufige, ausführliche Evaluierung des Gesamtsystems hinsichtlich Konzept, Leistungsfähigkeit und Praxistauglichkeit, wurde durchgeführt und wird in Kapitel 9 vorgestellt und diskutiert.

Gliederung der
Umsetzung und
Evaluierung

Abbildung 4.3.: Schematische Darstellung der Gliederung

4.2. Generalisierter Reparaturprozess

Für ein lernendes System ist die Art und die Qualität der vorhandenen Daten von entscheidender Bedeutung. Gelernt werden soll aus den in den Werkstätten durchgeführten Reparaturfällen. Hierfür muss zunächst ein allgemeiner Diagnostik- und Reparaturprozess näher betrachtet werden. Dieser ist in Abbildung 4.4 als iterativer Prozess dargestellt.

Abbildung 4.4.: Generalisierter Reparaturprozess

Prozessbeschreibung

Im ersten Schritt erfasst der Werkstatttester Symptome, die entweder aus dem Fahrzeug (automatisiert) ausgelesen oder vom KD-Mitarbeiter manuell eingegeben werden. Automatisiert erfasste Symptome sind vor allem Ereignisspeichereinträge. Vom KD-Mitarbeiter wahrgenommene Symptome, hier Expertensymptome genannt, sind z. B. Geräusche, Vibrationen, Messwerte oder gestörte Funktionen, welche in einer geeigneten Weise kodiert wurden. Nach diesem ersten Schritt wird das System Handlungsanweisungen an den KD-Mitarbeiter ausgeben. Dies können entweder Reparaturmaßnahmen oder mögliche Fehlerursachen sein, deren Überprüfung vom System vorgeschlagen wird. Führt die vorgeschlagene Handlungsanweisung nicht zur Identifikation der Fehlerursache oder zur Behebung des Defektes, wird der KD-Mitarbeiter im nächsten Schritt eine Rückmeldung an das System geben, das daraufhin erneut, analog zu Schritt zwei, Handlungsanweisungen ausgibt. Diese Rückmeldung kann dabei in Form von (vom System als Handlungsanweisung verlangten) Messwerten, Beobachtungen oder zusätzlichen Symptomen gestaltet sein. Dieser Prozess endet mit einer vom KD-Mitarbeiter als erfolgreich vermuteten Reparatur oder mit einem im ungünstigsten Fall nicht lokalisierbaren Defekt. Wurde die Fehlerursache dabei nicht durch einen direkten oder indirekten Hinweis des Diagnostiksystems ermittelt, wird dies als „*No Trouble Found*“ (NTF) bezeichnet. Das Diagnostik-

system war in diesem Fall nicht hilfreich. Diesem Prozess ist ein zeitlich ausgedehnter Prozess überlagert. Wurde der Defekt nur vermeintlich gelöst, wird davon ausgegangen, dass der Kunde, zumindest im Gewährleistungsfall erneut die Werkstatt aufsuchen wird und eine Reparatur verlangt.

Verschiedene Fahrzeugvarianten

Die System-Definition 2.2 berücksichtigt bisher nicht die Tatsache, dass Fahrzeuge je nach Modell, Variante, aber auch Alterserscheinungen sehr verschieden sein können, es sich also um (leicht) verschiedene Systeme $\Sigma_1, \dots, \Sigma_n$ mit $\Sigma_i \neq \Sigma_j, i, j \in [1, n]$ handelt. Mit der Annahme, dass dies einen Einfluss auf das resultierende Diagnostik-Modell hat, muss die Varianz der Fahrzeugsysteme ebenfalls berücksichtigt und modelliert werden. Nicht alle Varianz-bestimmenden Parameter können beobachtet bzw. aufgezeichnet werden. Zur Lösung werden die nicht beobachtbaren Parameter C_{nb} den ebenfalls existierenden nicht beobachtbaren Größen des Systems Σ zugeordnet und die beobachtbaren Parameter C mit in die Modellbestimmung aufgenommen. Die Parameter C werden im Folgenden als *Fahrzeugkontext* zum Fahrzeugsystem Σ bezeichnet. Eine Analyse zur Herleitung des Sachverhalts ist im Anhang A.4 zu finden.

Der Fahrzeugkontext

Modellierung von Reparaturfällen

Möglich wird eine umfassende Datenerfassung durch eine Onlineanbindung vieler Werkstatttester zum OEM und den Zwang der Werkstätten, reparaturspezifische Informationen zumindest für Gewährleistungsfälle zu übermitteln. Die Daten werden durch den OEM in einem Data Warehouse gesammelt und für eine Reihe verschiedener (meist statistischer) Auswertungen genutzt.

Aufbauend auf der generalisierten Prozessbeschreibung soll eine formale Modellierung der daraus entstehenden und aufgezeichneten Reparaturfalldaten erfolgen. Eine solche ist für alle weiteren Betrachtungen und Modellbildungen zwingend notwendig. So soll ein einzelner Reparaturfall r gemäß Definition 4.1 auf ein Tupel mit den vorhandenen bzw. gezielt ermittelten Symptomen S_R , den als defekt vermuteten Komponenten K_R , dem zugehörigen Fahrzeugkontext C_R und einigen zusätzlichen (Meta-)Informationen $\mathcal{I}_R$, den Reparaturfall betreffend, zusammengefasst werden.

Formale Definition

Definition 4.1 (Reparaturfall) *Ein Reparaturfall r ist definiert als ein Tupel*

$$r = (S_R, K_R, C_R, \mathcal{I}_R),$$

wobei $S_R \subseteq S_\Sigma$ die Menge aller vorhandenen Symptome, $K_R \subseteq K_\Sigma$ die Menge der defekten Komponenten, $C_R \subseteq C_\Sigma$ die Menge aller vorhandenen Kontextinformationen und $\mathcal{I}_R$ eine Menge zusätzlicher (Meta-)Informationen, wie z.B. Datum, Werkstatt, Verlauf der Reparatur darstellt. Die Menge aller für die Domäne möglichen Fälle wird mit R_Ω

bezeichnet. Dann sei $R_{FB} \subseteq R_{\Omega}, r \in R_{FB}$ die Menge aller in einer Fallbasis verfügbaren Fälle.

In der Symptomatik S_R werden alle während der Reparatur r festgestellten Symptome $s \in S_R$ zusammengefasst. Die defekt vermuteten Komponenten K_R stehen je nach Art der Kodierung nicht zwangsläufig direkt zur Verfügung und müssen gegebenenfalls aus den Reparaturmaßnahmen abgeleitet werden. Zu Reparaturmaßnahmen werden Aktionen des KD-Mitarbeiters wie getauschte Teile oder Arbeitspositionen zusammengefasst. Konsequenter muss die Zerlegung des Systems in Komponenten hinreichend feingranular erfolgen, dass beispielsweise auch Instantsetzungen von Abschnitten in Leitungssträngen geeignet erfasst werden können. Letztlich werden diese Reparaturmaßnahmen jedoch auf Komponenten K_R abgebildet¹. Abbildung 4.5 veranschaulicht die in einem Reparaturfall erfassten Daten.

Abbildung 4.5.: Reparaturprozess als eine Abbildung von Symptombild S_R auf Komponenten K_R , unter Berücksichtigung des Fahrzeugkontexts C_R und einiger Meta-Informationen $\mathcal{I}_R$

¹ Die Abbildung von Reparaturmaßnahmen auf Komponenten K_R wird detaillierter im Rahmen der Merkmalskodierung im Anhang B.1 beschrieben.

5 Analyse und Aufbereitung der Falldaten

Einer der ersten und zugleich wichtigsten Schritte im KDD-Prozess¹ ist es, zu Beginn ein Verständnis der vorhandenen Daten zu gewinnen und darauf aufbauend eine geeignete Merkmalsauswahl² und Datenvorverarbeitung zu entwickeln. Die Entwicklung der Modelle und deren Lernverfahren hängt maßgeblich von diesem Schritt ab.

Beginnend mit einem kurzen Überblick über die Akquise umfangreicher realer Falldaten folgt eine Analyse zur Relevanz verschiedener Attribute für die hier gestellte Lernaufgabe. Nach der Aggregation zu Merkmalen folgt eine wahrscheinlichkeitstheoretische Betrachtung, auf die später die Beschreibung der Diagnostik-Modelle aufsetzt. Abschließend werden verschiedene, für die Modellentwicklung und Lernverfahren relevante Häufigkeitsverteilungen vorgestellt, die eine Charakterisierung der resultierenden Datenbasis ermöglichen.

Die Gründe für die Bedeutung der Falldatenanalyse liegen im Einfluss derer auf die Modellauswahl und -parametrierung. Kriterien hierbei sind z. B. die Komplexität hinsichtlich Abhängigkeiten in den Daten, *Sparse Data* oder *Dense Data*, die Korrektheit bzw. Fehlerhäufigkeit (Rauschen), Single- oder Multilabel, die Skalierung der Attribute (nominal, ordinal, intervall, etc.), zu erwartende Fallzahl pro Modellparameter und viele mehr³. Werden die Kriterien falsch bestimmt, würde das zwangsläufig zur Entwicklung eines ungünstigen Modells oder zumindest zu einer ungünstigen Modellparametrierung führen. So sind z. B. Bayes'sche Netze nicht mehr berechenbar, wenn zu viele Abhängigkeiten in Kombination mit vielen Merkmalen vorhanden sind, oder SVMs liefern ungünstige Ergebnisse, wenn die Fallzahl pro Merkmal zu gering ist. Logistische Regressionsmodelle können wiederum keine bedingten Abhängigkeiten abbilden usw. Auch die Datenvorverarbeitung hat einen Einfluss auf die Entwicklung des Modells und umgekehrt, wie z. B. durch eine entsprechende Merkmalskodierung Abhängigkeiten geschaffen oder entfernt werden können oder die Komplexität erhöht oder reduziert werden kann.

Bedeutung der
Datenanalyse und -
aufbereitung

¹ KDD steht für *Knowledge Discovery in Databases*, ein in der KI eingeführter Begriff für den Gesamtprozess der Wissensgewinnung aus Datenbeständen (vgl. [Fay96, S. 39]).

² Die Merkmalsauswahl wird im englischen als *Feature Selection* bezeichnet und ist der Vorgang der Auswahl der für eine Lernaufgabe geeigneten Merkmale (Features). In der Statistik wird hierfür der Terminus „*unabhängige Variablen*“ verwendet.

³ Die einzelnen Begriffe werden später in den einzelnen Abschnitten näher erläutert.

Ein erheblicher Entwicklungsaufwand des Prototyps entfiel entsprechend auf die Analyse und Aufbereitung der Falldaten.

5.1. Akquise und Analyse der Falldaten

Mangelnde Qualität der Daten

Für die Datenakquisition standen u. a. Gewährleistungsanträge, Prüfprotokolle der Diagnostik-Tester, Kundenwahrnehmung und Werkstattwahrnehmung zur Verfügung. Diese Daten wurden derzeit jedoch nur für statistische Auswertungen verwendet und waren daher nicht in der für ein Lernsystem notwendigen Qualität erfasst und abgelegt. Entsprechend aufwändig gestaltete sich der Akquiseprozess. Die Summe der in den Datenbanksystemen enthaltenen Datensätze beinhaltet mehr als 1.000 Attribute. Im Rahmen der Datenexploration und -verknüpfung wurden Stück für Stück Attribute hinsichtlich ihrer Eignung analysiert und selektiert. Einige davon wurden zur weiteren Attributreduktion zu aggregierten Attributen zusammengefasst. So wurde letztlich durch komplexe SQL-Skripte, einige heuristische Filterregeln und Plausibilisierungen eine Datenbasis mit 56.184 Datensätzen von Reparaturfällen mit Fahrzeugen vom Modell Golf, Modelljahr 2005 - 2007 erzeugt⁴.

Resultierender Datensatz

Kennzahlen der Datenbasis

Die Struktur und die Verteilungen der Daten sind für das weitere Verständnis von Bedeutung und werden daher im Folgenden näher betrachtet. In Tabelle 5.1 sind einige Kennzahlen dargestellt, auf die später, speziell in der Charakterisierung der Merkmale, noch häufiger referenziert wird⁵.

5.2. Relevanz und Auswahl von Attributen

Manuelle Untersuchung

Ein entscheidender Aspekt eines Lernsystems ist die Auswahl der interessierenden Attribute, hier also, welche Klassen von Symptomen und Maßnahmen betrachtet werden müssen. Eine Vorauswahl wurde im Rahmen der Analyse der Fallbasis bereits getroffen, es bleibt jedoch noch offen, welchen Nutzen ein einzelnes Attribut erzielt bzw. auf welche Attribute verzichtet werden kann. Zu diesem Zweck sowie zur Untersuchung der Zuverlässigkeit wurde eine manuelle Untersuchung und Bewertung von 100 realen Reparaturfalldatensätzen vorgenommen. Jeder einzelne Datensatz wurde durch 48 Kriterien bewertet. Die Ergebnisse der wichtigsten, teilweise aggregierten Kriterien sind in Tabelle 5.2 dargestellt. Von den untersuchten Datensätzen waren 84 % auswertbar. Unauswertbar war ein Fall beispielsweise, wenn nicht eindeutig bestimmt werden konnte, was der Inhalt bzw. Sinn und Zweck der Reparatur war. Von den

Kongruenz und Inkongruenz

⁴ Für eine detailliertere Beschreibung des Akquiseprozesses wird auf Anhang B.1 verwiesen, eine Aufstellung der Filterkriterien ist im Anhang D.3 zu finden.

⁵ Eine detaillierte Beschreibung der Attribute DTC, S-DISS, ET, ETRN, AP, M-DISS und KDNR ist im Anhang B.2 zu finden.

	<i>Absolut</i>	<i>Anteilig</i>
Reparaturfallzahl	56184	
Anz. verschiedener Symptome	7100	
davon Ereignisspeichereinträge (DTCs)	5006	70 % ^a
davon Kundenwahrnehmung (S-DISS)	2094	30 % ^a
Anz. verschiedener Reparaturmaßnahmen	15242	
davon Ersatzteile (ET)	4087	27 % ^b
davon Rumpfnummern (ETRN)	2565	63 % ^c
davon Arbeitspositionen (AP)	6494	43 % ^b
davon Kundendienstnummern (KDNR)	895	6 % ^b
davon Werkstattwahrnehmung (M-DISS)	3624	24 % ^b
davon Sonstige (FM, FL)	142	> 1 % ^b
Anz. verschiedener Fahrzeugkontextvarianten	1474	

^a Bezogen auf Symptome

^b Bezogen auf Reparaturmaßnahmen

^c Bezogen auf Ersatzteile

Tabelle 5.1.: Verschiedene Kennzahlen der Datenbasis

auswertbaren Fällen waren 77 % kongruent⁶ (vgl. Abbildung 5.1), wobei die Zeilen 4 bis 9 jeweils den Anteil der einzelnen Attribute zur Kongruenz angeben. Die verbleibenden Zeilen geben an, welche Gründe zur Inkongruenz eines Falls geführt haben oder welche Aspekte zumindest als problematisch einzustufen sind⁷.

Symptom-Attribute

Aufbauend auf der Untersuchung soll zuerst der Einfluss der Symptom-Attribute auf die Kongruenz eines Falls betrachtet werden. Abbildung 5.2 (linke Seite) zeigt die Verteilung der Symptom-Attribute als Ergebnis der manuellen Untersuchung. Betrachtet wurden hierbei nur als (insgesamt) kongruent klassifizierte Fälle. Die Abbildung zeigt, dass in lediglich 23 % der Fälle beide Symptom-Attribute für die Lösung relevant waren, also von einer Redundanz gesprochen werden kann. Es sei an dieser Stelle angemerkt, dass auch wenn beide Symptom-Attribute relevant waren, keine gesicherte Aussage über die tatsächliche Redundanz gemacht werden kann (vgl. Beispiele im Anhang B.4). Damit wird deutlich, dass beide Symptom-Attribute, DTC und S-DISS, als Attribute verwendet werden müssen.

DTC und S-DISS
als Attribute relevant

⁶ Unter kongruent wird hier die plausible Übereinstimmung von Symptomen und durchgeführten Maßnahmen verstanden (vgl. dazu auch Anhang B.4).

⁷ Dem geneigten Leser wird die Analyse von 3 repräsentativen Fallbeispielen im Anhang B.4 empfohlen.

Abbildung 5.1.: Anteil der auswertbaren, kongruenten und inkongruenten Reparaturfälle in der manuellen Untersuchung der Datenbasis

Maßnahmen-Attribute

AP, ET und M-DISS als Attribute relevant

Abbildung 5.2 (rechte Seite) zeigt die Relevanz der Maßnahmen-Attribute zur Kongruenz eines Falls. Betrachtet wurden auch hier nur als kongruent klassifizierte Fälle. Es zeigt sich, dass alle drei Maßnahmen-Attribute betrachtet werden müssen, und kein Attribut redundant vertreten ist. Würde beispielsweise das M-DISS-Attribut ignoriert, könnten 8 % der Fälle, zumindest in der Evaluierung, in keinem Fall gelöst werden. In wie weit beispielsweise das M-DISS-Attribut in Kombination mit anderen ähnlichen Fällen dennoch redundant wäre und vernachlässigt werden könnte, kann nur eine manuelle Evaluierung des Diagnostikergebnisses zeigen.

Abbildung 5.2.: Relevanz von Symptom-Attributen (linke Seite) und Maßnahme-Attributen (rechte Seite) zur Kongruenz eines Datensatzes

Fahrzeugkontext-Attribute

In der Auswahl der Attribute des Fahrzeugkontexts erscheint die ausschließliche Betrachtung der Attribute Steuergeräte-Hardwareteilenummer, -Softwareteilenummer und -Versionsnummer, wie sie in der Analyse der Datenbasis (Abschnitt 5.1) vorgestellt wurden, zunächst unvollständig. Tatsächlich jedoch ist die Kombination der drei Attribute nur für sehr wenig verschiedene Hardwarekonfigurationen gültig (häufig gibt es tatsächlich nur eine mögliche Konfiguration). Mit Hardwarekonfiguration sind hier die für ein Steuergerät umgebenden Elemente (Stecker, Sensoren, Kabelstrang, etc.) gemeint. Der Zusammenhang wird deutlich, wenn berücksichtigt wird, dass jeder Sensor eine leicht verschiedene Kennlinie bzw. Charakteristik besitzt. Selbiges gilt für Aktuatoren. Auch Steckerbelegungen müssen exakt übereinstimmen usw. Konsequenterweise müssen daher keine weiteren differenzierenden Attribute

Steuergeräte-Hardwareteilenummer, -Softwareteile- und -Versionsnummer relevant

		<i>Absolut</i>	<i>Anteilig</i>
	Betrachtete Fälle insgesamt	100	
	Auswertbare Fälle	84	
	davon kongruente Fälle	65	77 %^a
Kongruent	DISS-Symptom relevant	27	42 % ^b
	DTC-Symptom relevant	53	82 % ^b
	ET aufschlussreich	58	89 % ^b
	AP aufschlussreich	48	74 % ^b
	M-DISS aufschlussreich	23	35 % ^b
	Ursachenkennzeichnung korrekt	60	92 % ^b
Inkongruent oder problematisch	Als Lsg. wurde die GFS angegeben	8	10 % ^a
	M-DISS - Kodierung falsch	3	4 % ^a
	M-DISS - Kodierung zu ungenau	5	6 % ^a
	Arbeitsposition zu ungenau	2	2 % ^a
	Die Lösung wird im Werkstatt-Freitext beschrieben	8	10 % ^a
	S-DISS zu ungenau	15	18 % ^a
	S-DISS falsch kodiert	4	5 % ^a
	Nicht relevanter DTC vorhanden	40	4 % ^a
	Sympt. - Maßn. widersprüchlich	6	7 % ^a
	2 od. mehr verschied- Fälle gleichzeitig	3	4 % ^a

^a Bezogen auf auswertbare Fälle

^b Bezogen auf kongruente Fälle

Tabelle 5.2.: Ergebnisse der manuellen Untersuchung der Datenbasis

berücksichtigt werden, sondern es muss eher noch über eine Einschränkung der Differenzierung durch die ausgewählten Attribute nachgedacht werden.

Aggregation zu Merkmalen

Merkmale als aggregierte Attribute

In der Reparaturfall-Definition 4.1 wird allgemein von Merkmalen (engl. Features) S_R, K_R, C_R und nicht von Attributen gesprochen. In den Domänen des Data Mining wie auch des maschinellen Lernens ist dies die gängige Nomenklatur, deren Abgrenzung zur Nomenklatur von Datenbanksystemen insofern sinnvoll ist, wie sich Merkmale nicht zwangsläufig 1:1 auf Attribute abbilden lassen oder dies aus Effizienzgründen nicht zweckmäßig ist. Die Symptom-Attribute sollen daher durch Mengensaddition auf das Merkmal S_R , die Maßnahmen-Attribute auf das Merkmal K_R und die Kontext-Attribute auf das Merkmal C_R abgebildet werden.

5.3. Wahrscheinlichkeitstheoretische Betrachtung von Merkmalen

Die Definition eines statistischen und später neuronalen Modells erfordert eine wahrscheinlichkeitstheoretische Betrachtung der nun aus Attributen aggregierten Merkmale S_R, K_R und C_R .

Symptome

Definition 5.1 (Symptom) *Ein Symptom $\mathcal{S}$ sei ein Ensemble, bestehend aus dem Tripel $(s, \mathcal{A}_\mathcal{S}, \mathcal{P}_\mathcal{S})$, wobei s den Wert einer Zufallsvariable darstellt, welcher eines der Elemente aus der Menge $\mathcal{A}_\mathcal{S} = \{s_0, s_1\}$ mit den zugehörigen Wahrscheinlichkeiten für jedes Symbol mit $\mathcal{P} = \{p_0, p_1\}$ annimmt.*

Dabei steht der Name $\mathcal{A}$ für *Alphabet* von $\mathcal{S}$, dessen Elemente im Folgenden dichotom nominalskaliert definiert sind mit⁸

$$\mathcal{A}_\mathcal{S} : \begin{cases} s_0 = 0 : \text{Symptom nicht vorhanden} \\ s_1 = 1 : \text{Symptom vorhanden} \end{cases} \quad (5.1)$$

Komponenten

Die Definition gilt analog für Komponenten $\mathcal{K}$ mit

$$\mathcal{K} = (k, \mathcal{A}_\mathcal{K}, \mathcal{P}_\mathcal{K}) \text{ mit } \mathcal{A}_\mathcal{K} : \begin{cases} k_0 = 0 : \text{Komponente i.O.} \\ k_1 = 1 : \text{Komponente defekt} \end{cases} \quad (5.2)$$

Als kürzere Schreibweise wird im Folgenden $P(s = s_0)$ auch mit $P(s_0)$ und $P(s = s_1)$ mit $P(s)$ notiert.

⁸ Mit der Definition gilt damit z. B. $P(s = s_0) = p_0$.

Es ist naheliegend, dass sich bei unterschiedlichen Systemen Σ_i für ein und dieselbe defekte Komponente unterschiedliche Fehlerbilder bzw. Symptomaten ausbilden. Diesem Umstand wurde mit der Einführung des Fahrzeugkontext C Rechnung getragen. Nun können aufgrund einer mangelnden Beobachtbarkeit nicht alle Systemunterschiede erfasst werden, so dass eine Beschränkung stattfinden muss. Betrachtet werden darum folgende in der Fallbasis vorhandenen, nominalen Eigenschaften:

- Fahrzeugmodell $c_{Modell} \in Modell_{\Sigma}$
- Verbauliste (verbaute ECUs) $C_{ECU} = \{(c_{ecu}, c_{Version})\} \subseteq ECU_{\Sigma}$ mit den dazu gehörigen
- Hardware- und Softwareversion einer ECU $c_{Version} \in Version_{\Sigma}$

Eine ECU c_{ecu} wird dabei mit der dazugehörigen Hardware- und Softwareversion $c_{Version}$ zum Tupel $c_{ECU} = (c_{ecu}, c_{Version})$ zusammengefasst. Die Menge aller Kontextinformationen für ein Fahrzeug bildet sich letztlich aus der Verbauliste C_{ECU} und dem Fahrzeugmodell c_{Modell} mit

$$C = c_{Modell} \cup C_{ECU} \quad (5.3)$$

Definition 5.2 (Fahrzeugkontext) *Ein Fahrzeugkontext $\mathcal{C}$ sei ein Ensemble, bestehend aus dem Tripel $(c, \mathcal{A}_{\mathcal{C}}, \mathcal{P}_{\mathcal{C}})$, wobei c den Wert einer Zufallsvariablen darstellt, welcher eines der Elemente aus der Menge $\mathcal{A}_{\mathcal{C}} = \{C_{\Sigma}\} = \{Modell_{\Sigma}, ECU_{\Sigma}\}$ mit den zugehörigen Wahrscheinlichkeiten $\mathcal{P}\{p_{Modell}, p_{ECU}\}$ annimmt.*

Dabei sind die Elemente des *Alphabets* von C im Folgenden nominalskaliert definiert mit

$$\mathcal{A}_{\mathcal{C}} : \begin{cases} c_{Modell} \in Modell_{\Sigma} & : \text{für } C \in Modell_{\Sigma} \\ c_{ECU} \in ECU_{\Sigma} & : \text{für } C \in ECU_{\Sigma} \end{cases} . \quad (5.4)$$

Weiterhin ist $Modell_{\Sigma}$ die Menge aller existierenden Fahrzeugmodelle, ECU_{Σ} die Menge aller existierenden Steuergeräte mit den dazugehörigen Hardware- und Softwareversionen aus $Version_{\Sigma}$. Auch hier wird als kürzere Schreibweise im Folgenden $P(c = c_{Modell_1})$ mit $P(c_{Modell_1})$ und $P(c = c_x)$ mit $P(c_x)$ notiert.

5.4. Häufigkeitsverteilungen und Charakterisierung

Die Charakterisierung nimmt direkten Einfluss auf die Modellauswahl und -auslegung. Wird irrtümlich von abweichenden Eigenschaften der

Daten ausgegangen, führt dies zwangsläufig zu einer verringerten Leistungsfähigkeit des Modells, wie das auch in diesem Projekt durch die anfangs fehlenden realen Datensätze der Fall war⁹.

Häufigkeitsverteilungen

Häufigkeitsverteilung von Symptomen

Ein bedeutender Aspekt sind dabei Häufigkeitsverteilungen. Für Symptome gilt z.B., das pro Reparaturfall ca. 3,2 Symptome vorhanden sind und damit ein bestimmtes Symptom S_i durchschnittlich in 25 Reparaturfällen der Datenbasis (das entspricht 0,44 % oder alle 2.200 Fälle) einmal vorkommt. Die Häufigkeitsverteilung in Abbildung 5.3 zeigt jedoch, dass keinesfalls von einer Gleichverteilung ausgegangen werden kann. Über 41 % der 7.100 Symptome sind in der Fallbasis nur einmal (= 0,018 % bzw. alle 56.000 Fälle) vorhanden. In Kontrast dazu existieren (wenige) Symptome, die dafür in über 5,3 % aller Fälle vorkommen. Diese Verteilungsinformation ist insofern von Bedeutung, als viele Algorithmen nicht mit Merkmalen umgehen können, die nur einmal in der Fallbasis existieren.

Häufigkeitsverteilung von Maßnahmen

Dieselbe Analyse kann für Reparaturmaßnahmen durchgeführt werden. Hier enthält ein Reparaturfall ca. 5,8 Reparaturmaßnahmen¹⁰. Entsprechend kommt eine bestimmte Reparaturmaßnahme K_i durchschnittlich in 22 Reparaturfällen der Datenbasis vor (= 0,38 %). Wie Abbildung 5.3 (rechts) zeigt, ist auch hier die Verteilung äußerst einseitig. 44 % aller Reparaturmaßnahmen kommen in der gesamten Fallbasis nur einmal vor (= 0,018 %). Dagegen gibt es Maßnahmen, die in über 72 % aller

⁹ Es wurden in Ermangelung realer Falldaten künstliche Daten modelliert. Die Modelle lieferten dabei hervorragende Ergebnisse, jedoch nicht mehr für die später vorhandenen realen Daten.

¹⁰ Die zunächst überraschend hohe Zahl ist darauf zurückzuführen, das die vier Kategorien ET, AP, M-DISS und KDNR zur Menge der Merkmale K_R hinzugefügt werden.

Abbildung 5.3.: Häufigkeitsverteilung von Symptomen (links) und Maßnahmen (rechts): 41 % aller Symptome und 44 % aller Maßnahmen existieren nur einmal in der Datenbasis

Reparaturfälle durchgeführt wurden (z. B. die GFS oder auch Nachbesserungen/Rückrufaktionen). Analog zur Symptommhäufigkeitsbetrachtung gilt auch für die Reparaturmaßnahmen, dass viele Algorithmen mit nur einem Fall pro Klasse nicht umgehen können.

Eine Kontextinformation kann im Gegensatz zu Symptomen oder Maßnahmen nur einmal pro Steuergerät vorhanden sein bzw. jedes Steuergerät besitzt genau eine Kontextinformation (es kann nur eine Softwareversion gleichzeitig programmiert sein). Kontextinformationen sind daher multinominale Merkmale. Eine Betrachtung der Häufigkeitsverteilung, wie sie für Symptome oder Maßnahmen durchgeführt wurde, ist daher nicht sinnvoll. Dennoch von Interesse ist die Anzahl der möglichen Kontextvarianten pro Steuergerät, wie sie Abbildung 5.4 zeigt. Dabei kann festgestellt werden, dass die meisten Varianten im Motorsteuergerät existieren, viele Steuergeräte jedoch nur ein oder zwei Varianten besitzen. Die mittlere Variantenzahl pro Steuergerät beträgt 32,8. Hinsichtlich der Diagnostik-Modelle ist eine geringe Variantenzahl von Vorteil, da so mehr Fälle pro Variante existieren.

Häufigkeitsverteilung
von Kontextinfor-
mationen

Charakterisierung zur Modellentwicklung

Mit den Kennzahlen der Datenbasis (Tabelle 5.1), der manuellen Untersuchung zur Zuverlässigkeit und Attributauswahl (Tabelle 5.2, Abbildung 5.2 und 5.2) und der Betrachtung der Häufigkeitsverteilungen konnte ein umfassendes Bild der zur Verfügung stehenden Daten gegeben werden. Nachfolgend sollen die bezüglich der Auswahl des Modells und des Lernverfahrens besonders relevanten Eigenschaften zusammengestellt und bewertet werden.

Abbildung 5.4.: Anzahl verschiedener Kontextvarianten pro Steuergerät (bezogen auf die Datenbasis)

Wie Abbildung 4.5 anschaulich darstellt, können die Reparaturfalldaten zum einen in Symptom- und Lösungsinformationen und zum anderen in Kontext- und Meta-Informationen unterteilt werden. Symptom- und Lösungsinformationen können wie folgt charakterisiert werden:

Nominal dichotome Merkmale Die Merkmale $s_i \in S_R$ und $k_i \in K_R$ können genau zwei Zustände annehmen: Symptom vorhanden/nicht vorhanden, Komponente defekt/nicht defekt. Diese Tatsache erleichtert die Modellauswahl und -parametrierung enorm, da z. B. statistische Funktionen wie der Satz von Bayes, etc. angewandt werden können.

Geringe Symptommhäufigkeit Tabelle 5.1 verdeutlicht, dass zwar viele verschiedene Symptome existieren ($\#S \approx 7.100$), gleichzeitig aber ein einzelnes Symptom s_i selten auftritt ($\tilde{P}(s_i) \approx 0,14\%$) und dabei äußerst ungleich verteilt ist (vgl. Abbildung 5.3)¹¹.

Geringe Maßnahmenhäufigkeit Analog zur Symptommhäufigkeit existieren viele verschiedene Komponenten ($\#K \approx 15.000$), gleichzeitig ist der Defekt einer bestimmten Komponente k_i aber selten ($\tilde{P}(k_i) \approx 0,38\%$) und dabei äußerst ungleich verteilt (vgl. Abbildung 5.3).

Geringe Fallzahl pro Merkmal Gerade für neue Fahrzeugmodelle existieren anfangs wenig Falldaten (< 56.000) in Relation zum Lösungsraum (Anzahl der Parameter). Die später vorgestellten Modelle besitzen aber bis zu $1 \cdot 10^{11}$ Parameter.

Unschärfe, verrauschte Daten Wie die Ergebnisse der manuellen Untersuchung in Tabelle 5.2 verdeutlichen, muss mit fehlenden, falsch ausgewählten oder ungenügend kodierten S-DISS-Symptomen und irrelevanten DTCs etc. in hohem Umfang gerechnet werden.

Mehrfachklassifizierung (Multilabel) Es existieren gleichzeitig verschiedenartige Hinweise auf die defekte Komponente (z. B. Ersatzteil, Arbeitsposition, Werkstattwahrnehmung, etc.) und es können mehrere Komponenten defekt sein. Viele Algorithmen verlangen die Existenz genau einer Lösungsklasse, können also nicht mit Mehrfachklassifizierung umgehen.

Gerade die geringe Symptom- und Maßnahmenhäufigkeit ordnen die Fallbasis in die Kategorie der „*Sparse Data*“ ein [Tew73]. Damit sind Daten gemeint, die aufgespannt auf eine Matrix M mit der Dimension $\#S \times \#K$ nur *sparsam* mit Elementen $m_{ij} = true$ befüllt sind. Besonders ungünstig wird diese Eigenschaft in Kombination mit geringen Fallzahlen.

Im Kontrast dazu stehen die Kontextinformationen. Für diese gilt:

¹¹ $\#S$ steht für die Kardinalität der Menge S und $\tilde{P}(s_i)$ für den Median.

Multinomial dichotom Jedes verbaute Steuergerät besitzt genau eine Hardware- und Softwareversion und entsprechend eine Kontextinformation c_i , die daher nicht entweder *wahr* oder *falsch*, sondern multinominalen Charakter hat. Die mittlere Anzahl verschiedener Kontextversionen beträgt $\#\tilde{c}_{Version} \approx 32,8$.

Charakterisierung
von Kontextinfor-
mationen

Einfachklassifizierung Jedes Steuergerät besitzt exakt eine Software- und Hardwareversion $c_{Version}$.

Zuverlässigkeit Für neuere Fahrzeuge sind Hardware- und Softwareversion zuverlässig mittels der Steuergeräteidentifikation zu erfassen. Allerdings gilt dies vor allem für ältere Steuergeräte nicht immer.

Fallzahl pro Merkmal Dicht besetzte Daten (engl. „dense data“): Für jeden Reparaturfall ist die vollständige Verbauliste bekannt.

Die Charakterisierung zeigt, dass Symptom- und Lösungsinformationen einen Kontrast zu Kontextinformationen bilden, also von einem Dualismus der Falldaten gesprochen werden kann. Verstärkt wird dieser Dualismus durch die später in Abschnitt 7.3.1 festgestellte Tatsache, dass jeweils zwischen Symptomen mit $s_i \perp s_j$ und zwischen Kontextinformationen mit $c_i \perp c_j$ nur sehr geringe statistische Abhängigkeiten bestehen, dafür jedoch zwischen Symptomen und Kontextinformationen mit $s_i \perp c_j$. Damit wird der Einsatz von Modellen und Verfahren *Out of the Box* verhindert, und es sind zwangsläufig spezifische neue Verfahren notwendig. Die nachfolgenden Kapitel beschreiben den sukzessiven Aufbau eines solchen spezifischen Modells.

Dualismus der
Falldaten

6 Statistische Diagnostik-Modelle

Mit der in der Falldatenanalyse durchgeführten Charakterisierung, gemeint ist insbesondere die aufgezeigte Dualität, wurde deutlich, dass neue Modelle zur Lösung der Diagnostikaufgabe notwendig sind. Das Kapitel beschreibt dabei unter Verwendung rein deskriptiver statistischer Methoden die Entwicklung eines deskriptiven Diagnostik-Modells. Mit der anschließenden Bewertung des Modells als in der Praxis schwer umsetzbar, wird dieses mit dem Bayestheorem auf ein inferentielles Modell erweitert. Letzteres zeigt dabei deutlich bessere Eigenschaften. Primär dienen die hier hergeleiteten statistischen Modelle als Grundlage für die im nachfolgenden Kapitel entwickelten neuronalen Modelle.

Mit dem in Abbildung 4.1 beschriebenen Prozess besteht die resultierende Lernaufgabe darin, aus den vorhandenen Daten ein Modell Π abzuleiten, das die beobachteten Eingangsgrößen wie Symptome und Fahrzeugkontext bestmöglich auf die Ausgangsgrößen, Diagnose-Hypothesen über defekte Komponenten, abbildet. Im Kontext der Verwendung des Modells wird im Folgenden jedoch von einem *Diagnostik-Modell* gesprochen¹. Für die Modellbildung selbst können nur die in aus der Systembetrachtung in Abschnitt A.4 beschriebenen beobachtbaren Größen X und $C_{b,i}$ bzw. mit $X \equiv S$ und $C_{b,i} \equiv C$, die Größen S und C herangezogen werden, so dass sich die resultierende Abbildung des Modells Π ergibt mit

$$\Pi : S_{\Sigma} \times C_{\Sigma} \rightarrow D_{\Sigma}, \quad S_R \times C_R \mapsto H_D, \quad (6.1)$$

wobei

$S_{\Sigma} =$ Menge aller Symptome

$C_{\Sigma} =$ Menge aller Kontextinformationen

$D_{\Sigma} =$ Menge aller Diagnosehypothesen.

Aufgrund der nicht beobachtbaren Größen Y in Σ erzeugt das Modell Π anstelle von K die in (2.6) und (2.7) eingeführten Hypothesen H_D über defekte Komponenten K .

6.1. Deskriptives Diagnostik-Modell

Die in Definition 5.1 verwendete Notation $P(x)$ gibt bisher lediglich ein Maß für die Wahrscheinlichkeit des Auftretens einer Zufallsvariable

¹ Die Bezeichnung Diagnostik-Modell ist gerechtfertigt, solange die Definition 2.1 der Diagnostik als Klassifikation akzeptiert wird und die Abbildung $S_R \times C_R \mapsto H_D$ aus 6.1 eine solche Klassifikation darstellt, wie später gezeigt wird.

x in einer Grundgesamtheit wie z.B. die der Fallbasis R_{FB} an. Schon für eine rein deskriptive Beschreibung von Zufallsvariablen ist die Verbundwahrscheinlichkeit $P(x \cap y)$ oder kürzer $P(x, y)$ interessant. Angewandt auf einen gegebenen Diagnostikfall mit einer beliebigen initialen Symptomatik $S_{Ix} = \{s_{Ix1}, s_{Ix2}, \dots, s_{Ixn}\} \subseteq S_\Sigma$ kann damit die Verbundwahrscheinlichkeit für den Defekt einer Reihe von Komponenten $K_y = \{k_{y1}, k_{y2}, \dots, k_{ym}\}$ bei gegebener Symptomatik S_{Ix} bestimmt werden mit

$$P(K_y, S_{Ix}) = P(k_{y1}, k_{y2}, \dots, k_{ym}, s_{Ix1}, s_{Ix2}, \dots, s_{Ixn}), \quad (6.2)$$

wobei für die hier angestrebten Betrachtungen der Kontext C_R zur Vereinfachung vorerst vernachlässigt und erst ab Kapitel 7.3 näher behandelt wird. Unter der Betrachtung der mit (6.2) bestimmten Verbundwahrscheinlichkeit als Vertrauenswert v (vgl. auch (2.7)) kann die korrespondierende Diagnostik-Hypothese h zu K_y und S_{Ix} mit dem Tupel

$$h = (K_y, P(K_y, S_{Ix})) \quad (6.3)$$

gebildet werden. Der so definierte Vertrauenswert wird jedoch mit größer werdender Anzahl von möglichen Symptomen S_Σ und Komponenten K_Σ sehr klein, da die Randwahrscheinlichkeiten sowie Integritätsbedingungen nach Kolmogorov (vgl. z.B. citeRice2006) nach wie vor erfüllt sein müssen, d.h. es gilt

$$P(K_y) \equiv \sum_{S \in \mathcal{P}(S_\Sigma)} P(K_y, S) \quad (6.4)$$

$$P(S_{Ix}) \equiv \sum_{K \in \mathcal{P}(K_\Sigma)} P(K, S_{Ix}) \quad (6.5)$$

und

$$\sum_{y \in \Lambda} P(K_y) = \sum_{x \in \Lambda} P(S_{Ix}) \equiv 1 .$$

Dies macht eine Normierung des Vertrauenswertes v sinnvoll, derart dass bei einer gegebenen Symptomatik S_{Ix} die Summe der Vertrauenswerte v aller durch das Modell hierfür generierten Diagnostik-Hypothesen $H_D(S_{Ix})$ eins ergibt. Damit ist

Normierung des
Vertrauenswertes v

$$\sum_{v_i \in h_i \in H_D(S_{Ix})} v_i \equiv 1 . \quad (6.6)$$

Mit (6.2) und (6.3) folgt

$$\sum_{K_y \in h_i \in H_D(S_{Ix})} P(K_y, S_{Ix}) \equiv 1 , \quad (6.7)$$

das letztlich der Randwahrscheinlichkeit für S_{Ix} aus (6.5) entspricht. Der normierte Vertrauenswert v_{norm} wird dann definiert mit

$$v_{norm} = \frac{P(K_y, S_{Ix})}{P(S_{Ix})} . \quad (6.8)$$

Daraus folgt die Definition für ein normiertes deskriptives Diagnostik-Modell.

Definition 6.1 (Deskriptives Diagnostik-Modell) Sei S_Σ die Menge aller möglichen Symptome und K_Σ die Menge aller möglichen Komponenten, dann ist ein deskriptives Diagnostik-Modell Π_{deskr} , das unter Verwendung deskriptiver statistischer Verfahren eine Menge von Symptomen S_I auf eine Menge an Diagnostik-Hypothesen H_D abbildet, definiert mit

$$\Pi_{deskr} : S_\Sigma \rightarrow K_\Sigma, S_I \mapsto \Pi_{deskr}(S_I) := \bigcup_{K_y \in \mathcal{P}(K_\Sigma)} \left\{ \left(K_y, \frac{P(K_y, S_I)}{P(S_I)} \right) \right\} . \quad (6.9)$$

Auf eine Sortierung der Ergebnismenge oder der Anwendung von Grenzwerten wird an dieser Stelle verzichtet.

Nachteile des
deskriptiven
Diagnostik-Modells

Eine Betrachtung von anwendungsorientierten Aspekten der Definition zeigt einige Nachteile, die im Folgenden diskutiert werden müssen. Zunächst ist hierfür die quantitative Berechnung der Verbundwahrscheinlichkeit $P(K_y, S_R)$ von Bedeutung. Bei gegebener Fallbasis R_{FB} wird diese durch „Auszählen“ bestimmt mit

$$P(K_y, S_R) = \frac{\#\bigcup_{r \in R_{FB}} \{r \mid K_i = K_y \wedge S_I = S_R : K_i, S_I \in r\}}{\#R_{FB}} \quad (6.10)$$

wobei

$$\begin{aligned} \#M &= \text{Kardinalität einer Menge } M \\ \#R_{FB} &= \text{Anzahl Fälle in der Fallbasis.} \end{aligned}$$

Keine Ähnlich-
keitsbeziehung

Anschaulich betrachtet repräsentiert $P(K_y, S_R)$ die relative Häufigkeit des Auftretens exakt dieser Merkmalskombination in einer Grundgesamtheit bzw. Fallbasis. Von Bedeutung ist hierbei die geforderte Gleichheit der Mengen K_y und K_R sowie S_I und S_R . Als Konsequenz besteht zwischen zwei in der Grundgesamtheit ähnlichen Fällen², die beispielsweise in nur einem Symptom abweichen, kein Einfluss. Hierfür ist eine Ähnlichkeitsbeziehung zwischen Reparaturfällen oder im mindesten zwischen Symptomatiken notwendig, die jedoch über die Fallbasis hinausgehende Annahmen voraussetzt, die in der deskriptiven Statistik nicht zulässig sind. Das Modell kann daher auch als exaktes deskriptives Diagnostik-

² Ein Ähnlichkeitsmaß soll an dieser Stelle nicht definiert werden.

Modell ohne Interpolationsfähigkeit bezeichnet werden. Dies bringt eine Reihe von Nachteilen mit sich.

Rauschsensitivität Die Falldatenanalyse (vgl. Kapitel 5) zeigt, dass zum einen ein gewisses „*Rauschen*“ existiert, d. h. dass zu den typischen Symptomatiken mit einer bestimmten Wahrscheinlichkeit weitere beliebige Symptome hinzukommen. Ohne Ähnlichkeitsbeziehungen existieren in der Fallbasis damit entweder kein oder nur sehr wenige Fälle. Sinnvolle Diagnostik-Hypothesen können so nicht aufgestellt werden.

Multiple Fehler Ähnlich verhält es sich mit multiplen Fehlern (vgl. Abschnitt 2.2.4). Sinnvolle Ergebnisse werden nur mit häufig auftretenden Fehlerkombinationen erreicht.

Fallbasisgröße Die beiden vorhergehend angeführten Punkte implizieren für eine geforderte Diagnostikqualität eine immens große Fallbasis, damit auch für jede Fehlerkombination und „*Rauschvariante*“ hinreichend viele Fälle für statistische Berechnungen vorhanden sind.

Generalisierungsfähigkeit Aus den vorhergehend genannten Punkten, insbesondere der Rauschsensitivität, die sich reziprok proportional zur Generalisierung verhält, folgt letztlich, dass eine Generalisierung nur für sehr große Fallbasen geschieht.

Aus der Berechnung der Verbundwahrscheinlichkeit $P(K_y, S_R)$ folgt noch ein weiterer Nachteil. Durch die immensen Kombinationsmöglichkeiten ist die Bestimmung sowie Verbreitung aller möglichen $P(K_y, S_R)$ nicht sinnvoll. Anstatt dessen muss für jede Diagnostik-Aufgabe direkt die Fallbasis verwendet werden. Damit stellt die Fallbasis selbst das Modellwissen dar. Die resultierende Größe, mangelnde Distributierbarkeit und großer Rechenaufwand sind einige Nachteile, die zu nennen wären.

Fallbasis als Modellwissen

6.2. Der Bayes'sche Wahrscheinlichkeitsbegriff

In der deskriptiven Statistik, häufig auch als objektivistische oder frequentistische Wahrscheinlichkeitsauffassung bezeichnet, werden Wahrscheinlichkeiten als relative Häufigkeiten interpretiert, wie auch die Bestimmung der Verbundwahrscheinlichkeit $P(K_y, S_R)$ in (6.10) verdeutlicht. In Kontrast dazu wird in der Bayes'schen Wahrscheinlichkeitstheorie ein neuer Wahrscheinlichkeitsbegriff geprägt, indem Wahrscheinlichkeit als Plausibilität einer Aussage bzw. Grad einer (persönlichen) Überzeugung (engl. „*degree of belief*“) interpretiert wird. Nichtsdestotrotz können solche Überzeugungsgrade auf frequentistische Wahrscheinlichkeitsvariablen

Bayes'scher vs. frequentistischer Wahrscheinlichkeitsbegriff

abgebildet werden, solange sie eine Reihe von einfachen Konsistenzregeln, bekannt als die Cox Axiome [Cox46], erfüllen. Folglich können Wahrscheinlichkeiten verwendet werden, um zum einen Annahmen und zum anderen Schlussfolgerungen (Inferenzen), diese Annahmen vorausgesetzt, zu beschreiben. Das heißt, zwei Personen mit denselben Annahmen und denselben Daten ziehen die gleichen Schlussfolgerungen. Diese allgemeinere Verwendung von Wahrscheinlichkeiten zum Quantifizieren von Überzeugungen ist auch als die Bayes'sche Sichtweise bekannt. Es wird dann von einer subjektiven Interpretation von Wahrscheinlichkeiten gesprochen, weil die Wahrscheinlichkeiten selbst unter einer Reihe von Annahmen abgeleitet wurden. Vertreter dieser Sichtweise sehen diese Subjektivität jedoch nicht als Schwäche, denn

Schlussfolgerungen können nicht ohne Annahmen gemacht werden³.

Es soll jedoch angemerkt werden, dass diese Sichtweise keine vollständig anerkannte ist, sondern vielmehr ein Disput zwischen Vertretern der frequentistischen und Bayes'schen Sichtweise besteht. Dies ist insofern nicht verwunderlich, als die Statistik die meiste Zeit des 20. Jahrhunderts durch nicht-Bayes'sche Methoden dominiert wurde, in denen Wahrscheinlichkeiten ausschließlich durch Zufallsvariablen beschrieben werden.

Der Satz von Bayes

Ein wichtiges Hilfsmittel ist der Satz von Bayes, der es erlaubt, aus den direkten Wahrscheinlichkeiten die indirekten (inversen) Wahrscheinlichkeiten zu ermitteln. Da viele praxisrelevante Fragestellungen ein inverses Problem beinhalten, ist er ein essentielles Hilfsmittel, wie nachstehend verdeutlicht wird [MPI08].

Die bequemerweise oft verwendete Maximum-Likelihood-Methode liefert als Ergebnis einer Parameterschätzung die (Modell-)Parameter Ψ , die am wahrscheinlichsten zu den erfassten Daten D (beispielsweise aus der Fallbasis R_{FB}) führen. Eine nützliche Schreibweise wird mit $P(D|\Psi, \mathcal{H})$ eingeführt. Die Notation steht für die Wahrscheinlichkeit der Daten D , gegeben der Parameter Ψ und die für die Betrachtungen notwendigen Hypothesen (Annahmen) $\mathcal{H}$. Die Maximum-Likelihood-Methode sucht nun nach den Parametern Ψ , die $P(D|\Psi, \mathcal{H})$ maximieren.

In einer Lernaufgabe sind jedoch die Parameter, die für gegebene Daten am wahrscheinlichsten sind, interessant, also $P(\Psi|D, \mathcal{H})$. Nur in Spezialfällen gilt $P(\Psi|D, \mathcal{H}) = P(D|\Psi, \mathcal{H})$, im Allgemeinen sind die beiden Größen verschieden. Das soll mit einem Beispiel veranschaulicht werden.

Beispiel zu inversen Wahrscheinlichkeiten

Angenommen, es wird eine Pfütze auf der Straße beobachtet, wie groß ist dann die Wahrscheinlichkeit für Regen $P(\text{Regen}|\text{Fleck})$? Sie ist nicht 100 %, denn die Straße kann aus vielerlei Gründen nass sein (die Straße ist vom letzten Regen noch nicht trocken, ein Auto wurde gewaschen, etc.). Wie groß ist jedoch die Wahrscheinlichkeit für eine Pfütze auf der

³ Diese Aussage ist übereinstimmend mit [Mac07, S.26].

Straße, falls es regnet $P(\text{Fleck}|\text{Regen})$? Diese wird bei nahezu 100 % liegen. Der Satz von Bayes erlaubt nun die Bestimmung von indirekten (inversen) Wahrscheinlichkeiten aus den leichter zugänglichen direkten Wahrscheinlichkeiten:

$$P(\Psi|D, \mathcal{H}) = \frac{P(D|\Psi, \mathcal{H}) * P(\Psi|\mathcal{H})}{P(D|\mathcal{H})} \quad (6.11)$$

Dabei stellt der Divisor $P(D|\mathcal{H})$ eine Normierungskonstante dar. Diese wird auch, aufgrund der Verfügbarkeit, als *Evidenz* bezeichnet. Daneben ist $P(D|\Psi, \mathcal{H})$ die bereits beschriebene *Likelihood*, eine normalerweise leicht zugängliche Größe. In $P(\Psi|\mathcal{H})$, der Priorterm, wird das aus früheren Beobachtungen vorhandene Wissen kodiert, d.h. Schätzungen der Größe der Parameter und ihrer Unsicherheit. Damit lässt sich letztlich $P(\Psi|D, \mathcal{H})$, als *Posterior* bezeichnet, bestimmen. Mit

$$\text{Posterior} = \frac{\text{Likelihood} * \text{Prior}}{\text{Evidenz}} \quad (6.12)$$

kann der so interpretierte Satz von Bayes nach [Mac07, S.29] zusammengefasst werden.

6.3. Inferentielles Diagnostik-Modell

Die Gleichungen (6.11) und (6.12) verdeutlichen den Inferenzcharakter inverser Wahrscheinlichkeiten. Durch A-priori-Wissen, Beobachtungen (Daten) und eine Menge von (subjektiven) Annahmen $\mathcal{H}$ wird eine A-posteriori-Verteilung geschätzt. Im Folgenden soll das deskriptive Diagnostik-Modell aus Definition 6.1 um die beschriebenen Bayes'schen Methoden erweitert werden.

Wird wiederum eine konkrete Diagnostik-Aufgabe mit der initialen Symptomatik S_I betrachtet, ergibt sich mit dem Bayestheorem (6.11) die A-posteriori-Wahrscheinlichkeit für eine defekte Komponente k_i mit

$$P(k_i|S_I, \mathcal{H}) = \frac{P(S_I|k_i, \mathcal{H})P(k_i|\mathcal{H})}{P(S_I|\mathcal{H})}, \quad (6.13)$$

wobei auch hier $\mathcal{H}$ für eine Menge von Hypothesen/Annahmen steht. Eine Annahme daraus ist, dass lediglich eine Komponente k_i und nicht Komponentenkombinationen K_y (vgl. (6.2)) die Lösung der Diagnostik-Aufgabe darstellen. Das schließt nicht aus, dass für eine Reparatur zwangsläufig mehrere Komponenten instandgesetzt bzw. ausgetauscht werden müssen. Auf weitere Hypothesen wird später noch eingegangen werden.

Annahme: nur eine Komponente defekt.

Werden nun die einzelnen Terme von (6.13) näher betrachtet, fällt schnell auf, dass speziell der Term $P(S_I|k_i, \mathcal{H})$, wie auch schon im

Nachteil:
 $P(S_I|k_i, \mathcal{H})$

deskriptiven Diagnostik-Modell der Term $P(K_Y, S_R)$ (6.10), in der Praxis schwer zu bestimmen ist und ebenfalls einige der bereits aufgeführten Nachteile wie Rauschsensitivität und Fallbasisgröße mit sich bringt. Wird der Term $P(S_I|k_i, \mathcal{H})$ ausgeschrieben mit $P(s_{I1}, s_{I2}, \dots, s_{In}|k_i, \mathcal{H})$, wird die Ursache in der Anzahl der möglichen Kombinationen von S_I und der daraus resultierenden Anzahl notwendiger repräsentativer Fälle anschaulich. Die Anzahl theoretisch möglicher Kombinationen entspricht dabei der Kardinalität der Potenzmenge $\mathcal{P}(S_\Sigma)$ und kann für die in Kapitel 5, Tabelle 5.1 bestimmten Anzahl von 7.100 verschiedenen Symptomen mit $\#\mathcal{P}(S_\Sigma) \approx 2^{7100} \approx 2 \cdot 10^{2137}$ abgeschätzt werden.

Annahme: Unabhängigkeit
der Symptome

Nun ermöglicht die Bayes'sche Sichtweise die Integration verschiedener Annahmen $\mathcal{H}$. So wird im Folgenden angenommen, die Symptome seien untereinander unabhängig, d. h. es gilt

$$P\left(\bigcap_{j \in J} s_j\right) = \prod_{j \in J} P(s_j) , \quad (6.14)$$

Paarweise Unabhängigkeit $s_i \perp s_j$

wobei $P(s_i \cap s_j)$ eine äquivalente Schreibweise zu $P(s_i, s_j)$ ist. Anzumerken ist hierbei, dass eine rein paarweise Unabhängigkeit, d. h.

$$(s_i \perp s_j) \forall i \in I, j \in J \quad (6.15)$$

nicht genügt, denn sie bedeutet nicht, dass zwei Zufallsvariablen „*nichts miteinander zu tun haben*“. Für zwei konkrete paarweise unabhängige Symptome s_i, s_j kann durchaus eine Abhängigkeit bestehen, falls ein weiteres Symptom s_x bekannt wird (für ein anschauliches Beispiel hierfür wird auf [Oel06, Kap. 4.3.1] verwiesen). Mit dem Hintergrund ist eine Überprüfung der Unabhängigkeit nach (6.14) für eine gegebene Fallbasis praktisch nicht durchführbar, gerade weil die relativ einfache Überprüfung der paarweisen Unabhängigkeit nicht genügt.

Vereinfachungen

Mit der Kettenregel (vgl. [Koc00, S.13-Kap. 2.1.7]) kann der Term $P(S_I|k_i, \mathcal{H})$ aus (6.13) wie folgt umgeschrieben werden:

$$\begin{aligned} P(S_I|k = k_i, \mathcal{H}) &= P(s_{In}|s_{In-1}, \dots, s_{I1}, k_i, \mathcal{H}) \cdot \\ &\quad P(s_{In-1}|s_{In-2}, \dots, s_{I1}, k_i, \mathcal{H}) \cdot \\ &\quad \dots \\ &\quad P(s_{I2}|s_{I1}, k_i, \mathcal{H}) \cdot P(s_{I1}|k_i, \mathcal{H}) \end{aligned}$$

Bemerkung:
Für $s_i \perp s_j$ gilt
 $P(s_i|s_j)=P(s_i)$

mit der Bedingung der Unabhängigkeit nach (6.14) folgt schließlich

$$P(S_I|k = k_i, \mathcal{H}) = P(s_{I1}|k_i, \mathcal{H}) \cdot P(s_{I2}|k_i, \mathcal{H}) \cdot \dots \cdot P(s_{In}|k_i, \mathcal{H}) . \quad (6.16)$$

Die A-posteriori-Wahrscheinlichkeit für eine defekte Komponente k_i aus (6.13) kann so vereinfacht werden zu

$$P(k_i|S_I, \mathcal{H}) = \frac{\left[\prod_{s_j \in S_I} P(s_j|k_i, \mathcal{H}) \right] \cdot P(k_i|\mathcal{H})}{P(S_I|\mathcal{H})} . \quad (6.17)$$

Der Nenner, nach (6.12) auch Evidenz oder Normierungskonstante genannt, wurde bisher nicht betrachtet. Mit der Unabhängigkeitsannahme nach (6.14) wäre eine Umformung mit

$$P(S_I|\mathcal{H}) = P(s_{I1}|\mathcal{H}) \cdot P(s_{I2}|\mathcal{H}) \cdot \dots \cdot P(s_{In}|\mathcal{H}) = \prod_{s_j \in S_I} P(s_j|\mathcal{H}) \quad (6.18)$$

möglich und der Term somit einfach zu bestimmen. Falls jedoch für einige Symptome mit $S_X \subseteq S_I$ diese Annahme verletzt ist, kann es zur Verletzung des Normierungsaxioms mit $P(k_i|S_I, \mathcal{H}) > 1$ kommen. Die exemplarische Anwendung auf Tabelle 6.1 genügt als Widerspruchsbeweis zum Normierungsaxiom. Nun kann $P(S_I|\mathcal{H})$ als Randwahrscheinlichkeit auch mittels der Produktregel bestimmt werden mit

Normierungs-
verletzung

$$\sum_{k'_i \in K_\Sigma} P(S_I|k'_i, \mathcal{H}) P(k'_i|\mathcal{H}) \quad (6.19)$$

und unter Anwendung der Unabhängigkeitsannahme vereinfacht werden zu

$$\sum_{k'_i \in K_\Sigma} \left[\prod_{s_j \in S_I} P(s_j|k'_i, \mathcal{H}) \right] \cdot P(k'_i|\mathcal{H}) . \quad (6.20)$$

Wegen $k_i \in K_\Sigma$ für (6.17) und $\sum_{k'_i \in K_\Sigma} \dots$ in (6.20) ist $0 \leq P(k_i|S_I, \mathcal{H}) \leq 1$ und damit das Normierungsaxioms erfüllt.

Analog zum deskriptiven Diagnostik-Modell kann der Vertrauenswert v einer Diagnostik-Hypothese H_D mit der A-posteriori-Wahrscheinlichkeit bestimmt werden. Auch hier ist eine Normierung erwünscht, die jedoch schon implizit durch den Satz von Bayes durch die Normierungskonstante $P(S_I|\mathcal{H})$ mit

Vertrauenswert v

$$\sum_{k_j \in K_\Sigma} P(k_j|S_I, \mathcal{H}) \equiv \sum_{k_j \in K_\Sigma} v_{k_j, norm} \equiv 1 \quad (6.21)$$

gegeben ist und für die durchgeführten Vereinfachungen bewiesen wurde. Damit ist auch die in (2.7), Abschnitt 2.2.4 geforderte Bedingung $v \in [0; 1]$ erfüllt. Daraus folgt die Definition für ein inferentielles Diagnostik-Modell.

Definition 6.2 (Inferentielles Diagnostik-Modell) Sei S_Σ die Menge aller möglichen Symptome und K_Σ die Menge aller möglichen Komponenten, dann ist ein deskriptives Diagnostik-Modell Π_{inf} , das unter Verwendung deskriptiver statistischer Verfahren eine Menge von

Symptomen S_I auf eine Menge an Diagnostik-Hypothesen H_D abbildet, definiert mit

$$\begin{aligned} \Pi_{inf} : S_\Sigma &\rightarrow K_\Sigma, S_I \mapsto \Pi_{inf}(S_I) \\ &:= \bigcup_{k_i \in K_\Sigma} \left\{ \left(k_i, \frac{\left[\prod_{s_j \in S_I} P(s_j | k_i, \mathcal{H}) \right] \cdot P(k_i | \mathcal{H})}{\sum_{k'_i \in K_\Sigma} \left[\prod_{s_j \in S_I} P(s_j | k'_i, \mathcal{H}) \right] \cdot P(k'_i | \mathcal{H})} \right) \right\}. \end{aligned}$$

Aspekte der
Realisierbarkeit

Im Gegensatz zum deskriptiven Diagnostik-Modell sind die komplexesten Fallbasisoperationen zur Berechnung des Modells der Art $P(s_j | k_i, \mathcal{H})$. Diese lassen sich durch die Überführung der Fallbasis in einzelne Vierfeldermatrizen effizient berechnen. Weitere Vorteile sowie Nachteile sollen im Folgenden kurz aufgeführt werden:

Geringe Rauschsensitivität Durch die stark vereinfachten Terme $P(s_j | k_i, \mathcal{H})$ geht ein zufällig zusätzlich vorhandenes Symptom lediglich mit $P(s_j | k_i, \mathcal{H})$ in das Ergebnis mit ein, dessen quantitativer Wert gering zu erwarten ist. Das gilt gleichermaßen für die in der Fallbasis vorhandenen „verrauschten“ Fälle.

Multiple Fehler Das Modell kann mit multiplen Fehlern, zum einen in der Fallbasis und zum anderen in einer konkreten Diagnostik-Aufgabe, umgehen. Zwar werden solche an sich nicht erkannt, jedoch werden zwei Fehler auch zwei Diagnostik-Hypothesen mit hohen Vertrauenswerten erzeugen.

Geringere Modellgröße Das Modellwissen an sich ist durch Vierfeldertafeln in Matrixform repräsentiert und nicht durch eine große Fallbasis.

Aufwändige Berechnung Trotz der stark vereinfachten Terme der Komplexität $P(s_j | k_i, \mathcal{H})$ müssen bedingt durch den Nenner eine Vielzahl an Operationen durchgeführt werden. Dies wirkt sich auf die Anwendungsgeschwindigkeit aus.

Komplexitätsbeschränkung Ein Nachteil der Unabhängigkeitsannahme zwischen Symptomen ist die mögliche abbildbare Komplexität. Sobald zwischen zwei Symptomen entgegen den Annahmen dennoch eine Abhängigkeit besteht, wird diese nicht exakt abgebildet. Ungünstig wird dies bei XOR-ähnlichen Abhängigkeiten, wie das folgende Beispiel in Tabelle 6.1 veranschaulicht.

Das Beispiel in Tabelle 6.1 zeigt, dass die Unabhängigkeitsannahme für AND-(und OR-)ähnliche Daten keine Nachteile mit sich bringt. Die ausführlichen Berechnungstabellen hierzu im Anhang D.1 zeigen, dass die Normierung mit (6.21) und die für die Daten

⁴ Die unterschiedlichen Fallzahlen sind notwendig um realistische Verhältnisse zu vermitteln.

Anz. ⁴ Fälle	s_1	s_2	k_1	Vertrauenswert $v_{k_1,n}$	Anz. Fälle	s_1	s_2	k_1	Vertrauenswert $v_{k_1,n}$
10.000	0	0	0	0,05 %	10.000	0	0	0	0,03 %
10	1	0	1	33,4 %	10	1	0	1	100,0 %
10	0	1	1	33,4 %	10	0	1	1	100,0 %
10	1	1	0	99,8 %	10	1	1	1	100,0 %

Tabelle 6.1.: Vertrauenswerte des inferentiellen Diagnostik-Modells nach Definition 6.2 für Links: XOR-ähnliche Daten; Rechts: AND-ähnliche Daten als Vergleich.

typische große Anzahl an Fällen mit $a_1 = 0 \wedge a_2 = 0 \wedge k_1 = 0$ dafür verantwortlich sind.

Generalisierungsfähigkeit Mit der geringen Rauschsensitivität steigt reziprok auch die Generalisierungsfähigkeit. Kleine „Ausreißer“ wirken sich nur gering auf das Ergebnis aus.

Mit dem deskriptiven und inferentiellen Diagnostik-Modell wurde beginnend mit der Falldatenmodellierung eine durchgehend wahrheitstheoretische Sichtweise angewendet. Die Bewertung des deskriptiven Diagnostik-Modells zeigt, dass eine rein frequentistische Herangehensweise für die Charakteristik der Daten bzw. der Aufgabe ungeeignet ist. Mit der Einführung des Bayes'schen Wahrscheinlichkeitsbegriffs und des inferentiellen Diagnostik-Modells konnten zum einen notwendige Annahmen integriert und zum anderen die Komplexität stark reduziert werden. Das so entstandene sehr viel kompaktere und robustere Modell soll als Grundlage für weitere, im nächsten Kapitel folgende Betrachtungen dienen.

7 Neuronale Diagnostik-Modelle

Bisher wurde zur vereinfachten Betrachtung der Fahrzeugkontext C vernachlässigt. Die Komplexitätsbeschränkung des inferentiellen Diagnostik-Modells macht eine sinnvolle Integration des Fahrzeugkontexts nicht möglich, wie später noch gezeigt werden wird. Hinzu kommt die Vielzahl der Berechnungen in der Anwendung des Modells. So werden leistungsfähigere und dennoch beherrschbare Modelle mit größerer Komplexität notwendig. Dabei sind neuronale Netze u.a. für ihre vielfältige, je nach Bedarf modellierbare Komplexität sowie der Effizienz in der Anwendung bekannt.

In diesem Kapitel werden sukzessive drei aufeinander aufbauende neuronale Diagnostik-Modelle entwickelt. Nach einer kleinen Einführung in neuronale Netze, die die nötigen Grundlagen zum Verständnis der weiteren Abschnitte legt, folgt zunächst eine wahrscheinlichkeitstheoretische Interpretation einschichtiger neuronaler Netze. Anhand dieser wird nachgewiesen, dass das inferentielle Diagnostik-Modell in ein einschichtiges neuronales Diagnostik-Modell überführt werden kann. Hierdurch wird eine Interpretation des Modells möglich. Zur Berücksichtigung des Fahrzeugkontexts C wird daraufhin ein Multi-Layer-Perceptron als Diagnostik-Modell vorgestellt. Die Analyse zeigt jedoch eine zu große abbildbare Komplexität des Modells, so dass mit einer exakten Definition der abzubildenden Abhängigkeiten ein komplexitätserweitertes Diagnostik-Modell entwickelt wird. Zur Steigerung der Adaptivität folgt eine Erweiterung des Trainingsverfahrens zum Verlernen alter Reparaturfälle. Um das Training der Modelle mit den großen Datenmengen und hochdimensionalen Merkmalsräumen beherrschbar zu machen, sind Optimierungsverfahren hinsichtlich des Ressourcenbedarfs notwendig. Dabei wird ein Verfahren zur Merkmals-Reduktion als Preoptimierung und ein Verfahren zum Pruning als Postoptimierung vorgestellt.

7.1. Eine kleine Einführung in neuronale Netze

Wie so oft in der Wissenschaft wurde auch die Entwicklung neuronaler Netze durch ein natürliches Vorbild motiviert. Sie sind definiert durch hochparallele Netzwerke, bestehend aus einfachen (meist adaptiven) Elementen, angeordnet in einer hierarchischen Struktur und dem Ziel mit Objekten der Umwelt zu interagieren, wie es auch für biologische Nervensysteme gilt.

Die frühesten Arbeiten über neuronale Netze sind auf McCullochs und Pitts's M-P Modell eines Neurons zurückzuführen [MP43]. Aufbauend

darauf entstanden in den 50ern und 60ern beachtliche Arbeiten mit einschichtigen neuronalen Netzen [Ros62], [WH60]. Obwohl einschichtige neuronale Netze in vielen Klassifikationsaufgaben erfolgreich waren, besitzen sie eine Reihe von Einschränkungen, so dass bestimmte (auch einfache) Funktionen wie z.B. das XOR-Muster nicht gelernt werden konnten [MP69]. Diese Einschränkungen führten in den 70ern zu einer breiten Ablehnung der Forschung in diesem Gebiet. Erst in den frühen 80ern wurden neuronale Netze wieder aufgegriffen, wesentlich durch die Entwicklung neuer erfolgreicher Lernalgorithmen für mehrschichtige Feedforward-Netze bedingt. Mit diesen sehr viel leistungsfähigeren Netzen konnten dann die Einschränkungen einschichtiger Netze überwunden werden, was zur zweiten Blütezeit der neuronalen Netze führte.

Ein kurzer geschichtlicher Abriss

Aktuell existieren eine Vielzahl verschiedener Arten neuronaler Netze wie mehrschichtige Feedforward-Netze (engl. multilayer feed forward Networks), RBF-Netze (Radial Basis Function Networks), ART-Netze (Adaptive Resonance Theory Networks), SOFM-Netze (Self-Organizing Feature Mapping Networks), etc. Mit diesem Repertoire stehen mit neuronalen Netzen leistungsfähige theoretische und praktische Techniken für das Lernen aus Daten für viele Anwendungen und Domänen zur Verfügung.

In dieser Arbeit werden mehrschichtige Feedforward-Netze verwendet und erweitert, so dass die folgenden Betrachtungen sich auf diese Klasse beschränken sollen. Weitere über diese Einführung hinausgehende Literatur ist beispielsweise in [BR07], [Zel96], [Spe96], [Hof93] und [Kre01] zu finden.

Beschränkung auf mehrschichtige Feedforward-Netze

7.1.1. Architektur neuronaler Netze

Feedforward-Netze konstituieren sich aus Neuronen, angeordnet in Schichten (engl. layers), wobei die erste Schicht Eingaben verarbeitet und die letzte Schicht Ausgaben bereitstellt (vgl. Abbildung 7.1). Die mittleren Schichten haben keine Verbindung zur Außenwelt und werden daher auch *verdeckte Schichten* (*hidden layers*) genannt. Jedes Neuron in einer Schicht ist mit jedem Neuron der nächsten Schicht verknüpft, wobei keine Verknüpfungen innerhalb einer Schicht existieren. Diese Verknüpfungen werden in neuronalen Netzen auch *Gewichte* (*weights*) genannt. Somit werden Informationen konstant von einer Schicht zur nächsten weitergeleitet und verarbeitet (feed forward). Die Gewichte in Feedforward-Netzen werden durch die Minimierung einer Fehlerfunktion, angewendet auf Trainingsdaten, bestimmt. Typische Fehlerfunktionen definieren sich aus der quadrierten Abweichung zwischen dem Netzausgabewert und dem Sollwert für jedes Trainingstupel der Trainingsdaten. Der populärste Ansatz dazu ist der *Backpropagation Algorithmus* [RM86], welcher ein Gradientenabstiegsverfahren zur iterativen Bestimmung der Gewichte

verwendet, indem die Fehlerwerte der Ausgangsschicht sukzessive zu den verdeckten Schichten rückgeleitet werden.

Abbildung 7.1.: Architektur eines mehrschichtigen neuronalen Feedforward-Netzes

Formale Definition:
Neurales Netz

Mit den vorangehenden Ausführungen kann ein künstliches neuronales Netz allgemein definiert werden wie folgt:

Definition 7.1 (Neurales Netz) *Ein neuronales Netz besteht aus einem gerichteten und gewichteten Graph $G = (V, E, w)$ mit der Menge der Neuronen V , der Menge der Verbindungen $E = \{(u, v) : u, v \in V\}$ sowie der Funktion $w : (u, v) \rightarrow \mathbb{R}$ für $u, v \in V$, die das Gewicht einer Verbindung von u nach v definiert.*

Für ein Feedforward-Netz ist unbedingt Rückkopplungsfreiheit gefordert, sodass gelten muss $\nexists p \in E^+, p = (v_{11}, v_{2n}) : v_{11} = v_{2n}$, wobei E^+ den transitiven irreflexiven Abschluss¹ aller möglichen Pfade p in E definiert. Zur besseren Verständlichkeit werden Eingangsneuronen nachfolgend mit v_{ij} , verdeckte Neuronen mit v_{hj} und Ausgabeneuronen mit v_{oj} notiert.

Adjazenzmatrix

Die Gewichte eines Netzes, in Abbildung 7.1 dargestellt mit w_{ij} , stellen die eigentliche Information dar. Nach der Hinton-Darstellung können diese auch in Form einer quadratischen Adjazenzmatrix nach Gleichung (7.1) dargestellt werden. Bezüglich der Indizierung ist hier die Reihenfolge von Quellneuron i zu Zielneuron j zu beachten. Nicht existente Verbindungen werden dabei mit einem Verbindungsgewicht von 0 dargestellt. Gewichtungen mit negativem Vorzeichen haben inhibitorischen (hemmenden) und mit positivem Vorzeichen exzitatorischen (verstärkenden) Einfluss.

¹ Der transitive Abschluss ist äquivalent zur transitiven Hülle E^* ohne Nullelement (irreflexiv).

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \quad (7.1)$$

Mit der Adjazenzmatrix wird neben den Gewichten auch die Netzstruktur definiert. In Abbildung 7.2 sind dazu zwei Beispiele zu sehen: Auf der linken Seite ein Feedforward-Netz, auf der rechten Seite ebenfalls ein Feedforward-Netz, jedoch mit zusätzlichen *Shortcut-Connections* (Verbindungen, die eine Schicht überspringen). In der jeweils darunter dargestellten Adjazenzmatrix sind alle Elemente mit Werten $\neq 0$ grau eingefärbt. Für Feedforward-Netze müssen alle Elemente in und unterhalb der Diagonalen den Wert 0 besitzen. Typisch ist für vollvernetzte Schichten auch die Blockstruktur innerhalb der Matrix.

Beispiele für
Netzstrukturen

Abbildung 7.2.: Beispiele für Netzstrukturen und Adjazenzmatrizen: *Links:* Feedforward-Netz mit drei Schichten. Darunter die zugehörige Adjazenzmatrix, wobei die Einträge mit Werten $\neq 0$ grau eingefärbt sind. *Rechts:* Feedforward-Netz mit drei Schichten und *Shortcut-Connections* (Verbindungen, die eine Schicht überspringen) [Kri07]

7.1.2. Künstliche Neuronen

Ein künstliches Neuron setzt sich, angelehnt an das Vorbild eines biologischen Neurons, aus einer Propagierungsfunktion f_{prob} der Aktivierungs-

funktion² f_{act} sowie der Ausgabefunktion f_{out} zusammen (Abbildung 7.3).

Abbildung 7.3.: Aufbau eines künstlichen Neurons

Propagierungs-
funktion

Die Propagierungsfunktion f_{prob} bildet die Reizweiterleitung und Reizakkumulation der Vorgängerneuronen zu einem Reizwert r_j des Neurons v_j ab³. Für Feedforward-Netze wird hier vereinfacht die lineare Summation nach Gleichung 7.3 verwendet. Sie stellt die Multiplikation des Ausgabewerts y_j eines jeden Neurons v_j mit dem Verbindungsgewicht w_{ij} und die anschließende Aufsummierung der Ergebnisse dar.

Definition 7.2 (Propagierungsfunktion)

Es sei $V = \{v_1, v_2, \dots, v_n\}$ die Menge der Neuronen, für die gilt $\forall z \in \{1, \dots, n\} : \exists w_{v_z, j}$. Die Propagierungsfunktion eines Neurons j sei dann allgemein definiert als

$$f_{prop} : \{y_1, \dots, y_n\} \times \{w_{1j}, \dots, w_{nj}\} \rightarrow r_j, \quad (7.2)$$

wobei y_z die Ausgabe des Neurons v_z darstellt und die Gewichte w_{ij} gemäß Definition 7.1 bestimmt sind.

$$r_j = \sum_i (y_i \cdot w_{ij}) \quad (7.3)$$

Aktivierungsfunktion

Ausgehend vom Reizwert r_j wird analog nach dem Vorbild der Natur das Aktivitätspotenzial, im Folgenden Aktivierung³ a_j genannt, bestimmt. Diese wird allgemein als Funktion der Zeit t , des Reizwertes r_j und des sogenannten Schwellwertes Θ_j bestimmt.

Definition 7.3 (Aktivierungsfunktion) Für ein Neuron v_j sei die Aktivierungsfunktion allgemein definiert als

$$f_{act} : r_j(t) \times a_j(t-1) \times \Theta_j \rightarrow a_j(t) \quad (7.4)$$

² Die Aktivierungsfunktion wird in der Literatur häufig auch als Transferfunktion bezeichnet.

³ In der Literatur sind die Begrifflichkeiten nicht eindeutig definiert. In [Zel96] und [Kri07] wird beispielsweise von einer *Netzeingabe* net_j gesprochen. Diese Begrifflichkeiten werden jedoch aus der Sicht des Autors als widersprüchlich angesehen und die treffendere Bezeichnung *Reizwert* r_j vorgezogen. In den vorhergehend erwähnten Literaturangaben wird der Begriff *Reizwert* als Synonym für die *Aktivierung* verwendet. Auch diese Begrifflichkeit ist aus Sicht des Autors widersprüchlich gegenüber dem biologischen Vorbild.

Die Definition eines Schwellenwerts Θ für ein Neuron ist ebenfalls biologisch motiviert. Der Schwellenwert stellt eine Reizschwelle dar, ab der ein Neuron „feuert“. Für die meisten Aktivierungsfunktionen von künstlichen Neuronen entspricht das dem Bereich der größten Sensitivität. Mathematisch korreliert das mit dem Bereich der größten Steigung $\max(f'_{act})$ der Aktivierungsfunktion. Der Schwellenwert wird häufig, wie auch in dieser Arbeit, durch eine äquivalente Gewichtung zu einem sogenannten Bias-Neuron ersetzt, das sind Neuronen, die ständig das Aktivitätspotential 1 besitzen.

In den Anfängen neuronaler Netze wurden Schwellenwertfunktionen als Aktivierungsfunktion eingesetzt. Diese gehören jedoch zur Gruppe der unstetigen Funktionen und sind damit nicht in jedem Punkt differenzierbar. Viele Trainingsalgorithmen wie z.B. der Backpropagation fordern jedoch gerade eine stetige Differenzierbarkeit. Die sogenannten *Sigmafunktionen* gehören zu einer Funktionsklasse, die diese Anforderung an stetiger Differenzierbarkeit erfüllen. Weiterhin erfüllen diese Funktionen Eigenschaften wie einen begrenzten Wertebereich und Monotonie. Ein bekannter Vertreter dieser Klasse ist die *Fermifunktion*. Sie ist allgemein definiert durch

$$f(x) = \frac{1}{1 + e^{\frac{-x}{T}}} \quad (7.5)$$

wobei T ein sogenannter Temperaturparameter ist, der die Steigung im Ursprung bestimmt. In Abbildung 7.4 sind sehr anschaulich ein typischer Verlauf sowie die Auswirkung des Temperaturparameters zu sehen.

Abbildung 7.4.: Die sigmoide Funktion für verschiedene $T \in \{0.5; 1; 2; 10\}$

Die einfachste und sehr beliebte Form dieser Funktion mit $T = 1$ wird häufig auch als *logistische Funktion* bezeichnet. Sie besitzt einen begrenzten Wertebereich von $(0, 1)$, Monotonie sowie stetige Differenzierbarkeit und ist damit für viele Trainingsalgorithmen gut geeignet. Die logistische Funktion ist wegen ihrem Zusammenhang zur Statistik eine besonders häufig eingesetzte Aktivierungsfunktion und darum auch in dieser Arbeit gewählt.

Ein weiterer, ebenfalls häufig verwendeter Vertreter der Sigmafunktionen ist der *Tangens Hyperbolicus*. Gegenüber der logistischen Funktion beschränkt sich sein Wertebereich auf $(-1, 1)$. Die Erweiterung des

Sigmafunktionen

Logistische Funktion als spezielle Sigmafunktion

Logistische Funktion
vs. Tangens
Hyperbolicus

Wertebereichs auf -1 kann für spezielle Aufgabenstellungen Vorteile hinsichtlich Trainings- und Konvergenzgeschwindigkeit bringen (vgl. [Zel96], Kapitel 6.2.3). Es kann nachgewiesen werden, dass der Tangens Hyperbolicus nach Gleichung 7.6 mit der Beziehung $e^x = \frac{1}{e^{-x}}$ wieder in eine Fermifunktion nach Gleichung 7.7 überführt werden kann, so dass für ihn dieselben positiven Eigenschaften wie für Fermifunktionen gelten.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (7.6)$$

$$= 2 f(x) - 1 \quad (7.7)$$

Ausgabefunktion

Die Ausgabefunktion f_{out} ist zumeist definiert über die Identität, so dass

$$y_j = f_{out}(a_j) = a_j, \quad (7.8)$$

und sie damit nicht weiter betrachtet werden wird.

7.1.3. Berechnung neuronaler Netze

Matrixdarstellung

Für reine Feedforward-Netze mit einer verdeckten Schicht, wie sie nachfolgend verwendet werden, ist es für die Berechnung der Ausgabewerte hilfreich, zunächst die Gewichtsmatrix W in eine Blockmatrix zu überführen mit

$$W = \begin{bmatrix} 0 & W_I & 0 \\ 0 & 0 & W_O \\ 0 & 0 & 0 \end{bmatrix} \quad (7.9)$$

oder kürzer, mit der direkten Summe⁴ $W_I \oplus W_O$, mit

$$W = \begin{bmatrix} 0 & W_I \oplus W_O \\ 0 & 0 \end{bmatrix}. \quad (7.10)$$

Dabei ist 0 die Nullmatrix (die Dimension wurde der Übersicht wegen jeweils nicht angegeben und ergibt sich aus (7.1)), W_I die Gewichtsmatrix zwischen der Eingangsschicht und der verdeckten Schicht und W_O die Gewichtsmatrix zwischen der verdeckten Schicht und der Ausgangsschicht mit

⁴Die *direkte Summe* mit dem Symbol $\oplus$ darf nicht verwechselt werden mit der „normalen“ Summe $+$.

$$W_I = \begin{bmatrix} w_{i_1 h_1} & w_{i_1 h_2} & \cdots & w_{i_1 h_n} \\ w_{i_2 h_1} & w_{i_2 h_2} & \cdots & w_{i_2 h_n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{i_n h_1} & w_{i_n h_2} & \cdots & w_{i_n h_n} \end{bmatrix} \text{ und} \quad (7.11)$$

$$W_O = \begin{bmatrix} w_{h_1 o_1} & w_{h_1 o_2} & \cdots & w_{h_1 o_j} \\ w_{h_2 o_1} & w_{h_2 o_2} & \cdots & w_{h_2 o_j} \\ \vdots & \vdots & \ddots & \vdots \\ w_{h_n o_1} & w_{h_n o_2} & \cdots & w_{h_n o_j} \end{bmatrix}. \quad (7.12)$$

Mit der Zusammenfassung von Propagierungsfunktion f_{prop} , Aktivierungsfunktion f_{act} und Ausgabefunktion f_{out} zur Ausgabefunktion $y_k(\mathbf{x})$ ⁵ eines Neurons k folgt bei gegebenem Eingabevektor $\mathbf{x}$

Resultierende
Berechnung

$$y(\mathbf{x}) = f_{out}(f_{act}(f_{prop}(\mathbf{x}))) \quad (7.13)$$

und weiter

$$y_k(\mathbf{x}) = f_{act} \left(\sum_{a=1}^{n_H} w_{h_a o_k} f_{act} \left(\sum_{b=1}^{n_x} w_{i_b h_a} x_b + \Theta_{h_a} \right) + \Theta_{h_k} \right). \quad (7.14)$$

Die Betrachtung von Eigenschaften eines einzelnen Neurons, wie auch eine statistische Interpretation folgen im nächsten Abschnitt. Auf weitere, für diese Arbeit notwendigen Elemente, wird aus Gründen der Verständlichkeit jeweils später in den entsprechenden Abschnitten näher eingegangen.

7.1.4. Training neuronaler Netze

Analog zur Zahl der unterschiedlichen Topologien und Netzvarianten gibt es auch entsprechend viele Lernverfahren. Prinzipiell können diese nach [Zel96, S. 93] jedoch in drei verschiedene Kategorien eingeteilt werden:

Unüberwachtes Lernen (Unsupervised Learning): Lernverfahren dieser Kategorie besitzen starke Analogien zu Clusteralgorithmen. Sie zeichnen sich dadurch aus, dass zwar Trainingsdaten existieren, jedoch keine Bewertungen oder Klasseneinteilungen. Sie haben meist das Ziel, Gruppen (Cluster) oder Strukturen in den Daten zu finden. Verwendet werden diese z.B. zur Datenreduktion oder -kodierung. Der bekannteste Vertreter hierfür sind selbstorganisierende Karten (self organizing maps).

Bestärkendes Lernen (Reinforcement Learning); Auch in dieser Kategorie an Lernverfahren sind nur Trainingsdaten ohne Klasseneinteilung (labels) vorhanden. Jedoch existiert eine Bewertungsfunktion

⁵ Die fettgedruckte Schreibweise $\mathbf{x}$ oder $\mathbf{w}$ steht hier für Vektoren.

für die Netzausgabe, die das Gesamtergebnis aber nur als richtig oder falsch bewertet. Verfahren dieser Klasse kommen häufig dem Prinzip „*trial and error*“ nahe.

Überwachtes Lernen (Supervised Learning): Für Algorithmen des überwachten Lernens stehen Daten mit Klasseneinteilungen oder (zu lernenden) Zielwerten (labels) zur Verfügung. Für jedes Trainingsbeispiel (Pattern) einer Trainingsmenge stehen folglich ein vollständiger Eingabe- und Ausgabevektor zur Verfügung. Algorithmen dieser Klasse minimieren meist über Gradientenabstiegsverfahren eine Fehlerfunktion über alle Trainingsdatensätze.

Hebbsche Lernregel Im Rahmen dieser Arbeit ist die letzte Kategorie, das überwachte Lernen relevant. Ausgangspunkt vieler Verfahren des überwachten Lernens ist die Hebbsche Lernregel, die in ihrer ursprünglichen Form auf biologische Untersuchungen durch Donald O. Hebb [Heb49, S. 60-78] zurück geht. Die Veränderung eines Gewichts wird nach Hebb bestimmt durch das Produkt der Ausgabe y_i des vorausgehenden Neurons v_i mit der Aktivierung a_j des nachfolgenden Neurons v_j , gewichtet mit einer Lernrate η :

$$\Delta w_{ij} = \eta y_i a_j. \quad (7.15)$$

Delta-Regel Daraus leitet sich die später eingeführte Delta-Regel ab, die definiert ist mit

$$\begin{aligned} \Delta w_{ij} &= \eta y_i \delta_j \\ \delta_j &= \mathbf{t}_j - y_j. \end{aligned} \quad (7.16)$$

Der Fehlerterm δ_j bestimmt dabei die Abweichung zwischen der Ausgabe y_j und dem Sollwert $\mathbf{t}_j$. Zum Training eines gesamten Netzes G werden iterativ für jedes Trainingsbeispiel p einer Trainingsmenge P die Gewichte gemäß der Delta-Regel modifiziert. Eine Iteration durch alle Trainingsbeispiele einer Trainingsmenge wird dabei als eine *Periode* bezeichnet. Ein Trainingsbeispiel $p \in P$ besteht aus einem Tupel mit dem Eingabevektor $\mathbf{i}$ und dem Ausgabevektor $\mathbf{t}$ mit $p = (\mathbf{i}, \mathbf{t})$.

Training als Fehlerminimierung

Das Vorgehen zeigt, dass ein Training immer die (iterative) Minimierung einer Fehlerfunktion E , wie z. B. den Fehlerterm δ_j in der Delta-Regel, zum Ziel hat. Am häufigsten wird die quadratische Fehlerfunktion verwendet mit

$$E(G, P) = \sum_{p \in P} \sum_j \frac{1}{2} (\mathbf{t}_{j,p} - y_{j,p})^2, \quad \mathbf{t}_{j,p} \in p. \quad (7.17)$$

Der Term entspricht dabei dem Gesamtfehler E des Netzes G über alle Trainingsbeispiele $p \in P$ und aller Ausgabewerte y_j .

Mit der Definition von Training als Fehlerminimierung stellen sich Trainingsverfahren als Suchverfahren nach einem (globalen) Optimum in einem durch die Gewichtsmatrix W aufgespannten Suchraum dar. Dies veranschaulicht z. B. Abbildung 7.5 für ein Gradientenabstiegsverfahren. Allerdings sei angemerkt, dass ein solches globales (eindeutiges) Minimum nicht immer existieren muss. Häufig existieren mehrere lokale, aber nicht globale Minima.

Lernen als Suche
nach dem Optimum

Abbildung 7.5.: Veranschaulichung des Gradientenabstiegsverfahrens für einen zweidimensionalen Suchraum (nach [Dre01])

Ein solches Gradientenabstiegsverfahren ist auch das weit verbreitete Backpropagation-Verfahren, dass zum weiteren Verständnis der Arbeit hier kurz eingeführt werden soll. Das Backpropagation-Verfahren verwendet ebenfalls die quadratische Fehlerfunktion E , die in eine Fehlerfunktion E_p für ein konkretes Trainingsbeispiel p zerlegt werden kann:

Der Backpropagation-Algorithmus

$$E_p = \sum_j \frac{1}{2} (\mathbf{t}_{j,p} - y_{j,p})^2, \quad \mathbf{t}_{j,p} \in p. \quad (7.18)$$

Der für ein Gradientenabstiegsverfahren notwendige Gradient g_{ij} bzw. Partialgradient $g_{ij,p}$ für ein Trainingsbeispiel p wird mit E gebildet durch

$$g_{ij} = \frac{\partial E}{\partial w_{ij}} \quad (7.19)$$

$$g_{ij,p} = \frac{\partial E_p}{\partial w_{ij}} \quad (7.20)$$

$$g_{ij} = \sum_p g_{ij,p}. \quad (7.21)$$

Für Online-Backpropagation⁶ berechnet sich dann die Gewichtsänderung Δw_{ij} analog zur Delta-Regel mit

$$\Delta w_{ij} = \eta g_{ij,p} \quad (7.22)$$

mit

$$g_{ij,p} = y_{i,p} \delta_{j,p}. \quad (7.23)$$

Gradienten-
berechnung

Anschaulich stellt dies analog zur Abbildung 7.5 eine Gewichtsänderung von w_{ij} in Richtung des Gradienten dar. Umso stärker der Gradient (Änderung in der Fehlerfunktion E_p , desto stärker auch die Gewichtsänderung. Zur Berechnung von Δw_{ij} sind weitere Schritte notwendig. Unter Anwendung der Kettenregel kann (7.22) faktorisiert werden zu

$$g_{ij,p} = \frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial r_{j,p}} \cdot \frac{\partial r_{j,p}}{\partial w_{ij}} \quad (7.24)$$

mit $r_{j,p} = \sum_i y_i w_{ij}$ als gewichtete Eingabe aus (7.3) für das Trainingsbeispiel p . Diese nach w_{ij} differenziert ergibt entsprechend die Ausgabe des Vorgängerneurons v_i wegen

$$\frac{\partial r_{j,p}}{\partial w_{ij}} = \sum_a y_{a,p} w_{aj} \frac{\partial}{\partial w_{ij}} = y_{i,p}^7. \quad (7.25)$$

Fehlerterme

Der erste Term aus (7.24), der Ableitung der Fehlerfunktion nach der gewichteten Summe $r_{j,p}$, wird als Fehlerterm $\delta_{j,p}$ definiert und zur Berechnung nochmals faktorisiert zu

$$\delta_{j,p} = -\frac{\partial E_p}{\partial r_{j,p}} = -\frac{\partial E_p}{\partial y_{j,p}} \cdot \frac{\partial y_{j,p}}{\partial r_{j,p}}. \quad (7.26)$$

Der zweite Term, die Ableitung der Ausgabefunktion $y_{j,p}$ nach der gewichteten Summe $r_{j,p}$, wird abgekürzt mit

$$y'_{j,p} = \frac{\partial y_{j,p}}{\partial r_{j,p}}. \quad (7.27)$$

Zur letztlichen Bestimmung des Fehlerterms $\delta_{j,p}$ aus (7.26) müssen zwei Fälle unterschieden werden:

1. Es handelt sich um ein Ausgabeneuron v_{o_j}
2. Es handelt sich um ein verdecktes Neuron v_{h_j}

⁶ Es existieren zwei Varianten, Online- und Offline-Backpropagation. Verwendet wir hier Online-Backpropagation, das wegen der Verwendung des partiellen Gradienten $g_{ij,p}$ anstelle des absoluten Gradienten g_{ij} auch stochastisches Gradientenabstiegsverfahren genannt wird.

⁷ Die Ableitung $y_{a,p} w_{aj} \frac{\partial}{\partial w_{ij}}$ ist nur für den Fall $a = i$ ungleich null.

Im **ersten Fall** ergibt sich der Fehlerterm $\delta_{j,p}^o$ für das Ausgabeneuron v_{o_j} wegen

Fehlerterm für ein Ausgabeneuron

$$\frac{\partial E_p}{\partial y_{j,p}} = \sum_a \frac{1}{2} (\mathbf{t}_{a,p} - y_{a,p})^2 \frac{\partial}{\partial y_{j,p}} = \mathbf{t}_{j,p} - y_{j,p} \quad (7.28)$$

zu

$$\delta_{j,p}^o = -y_{j,p}^o ' \frac{\partial E_p}{\partial y_{j,p}} = -y_{j,p}^o ' (\mathbf{t}_{j,p} - y_{j,p}^o) . \quad (7.29)$$

Im weiteren Verlauf der Arbeit wird als Aktivierungsfunktion $y_{j,p}$ nur noch die logistische Funktion verwendet, so dass sich der Fehlerterm mit

$$y_{j,p}' = \frac{1}{1 + e^{-r_{j,p}}} = y_{j,p}(1 - y_{j,p}) \quad (7.30)$$

weiter konkretisieren lässt in seine endgültige Form zu

$$\delta_{j,p}^o = y_{j,p}^o (1 - y_{j,p}^o) (\mathbf{t}_{j,p} - y_{j,p}^o) . \quad (7.31)$$

Im **zweiten Fall** lässt sich der Fehler des verdeckten Neuron v_{h_j} nicht direkt aus dessen Ausgabe berechnen, da hier kein Sollwert $\mathbf{t}_j$ wie im Fall von Ausgabeneuronen existiert. Das Backpropagation-Verfahren behilft sich damit, den Fehler des verdeckten Neurons durch die Rückverfolgung und Aufsummierung der Fehler der direkt verbundenen Ausgabeneuronen zu berechnen (daher der Name Backpropagation). Für mehrschichtige Netze geschieht dies rekursiv. Die Fehlerterme $\delta_{j,p}^h$ berechnen sich folglich mit

Fehlerterm für ein verdecktes Neuron

$$\delta_{j,p}^h = -y_{j,p}^h ' \frac{\partial E_p}{\partial y_{j,p}} = y_{j,p}^h ' \sum_k \delta_{k,p}^o w_{jk} \quad (7.32)$$

und mit der Konkretisierung auf die sigmoide Aktivierungsfunktion mit

$$\delta_{j,p}^h = y_{j,p}^h (1 - y_{j,p}^h) \sum_k \delta_{k,p}^o w_{jk} . \quad (7.33)$$

Die Gewichtsänderung Δw_{ij} pro Trainingsbeispiel p berechnet sich für das Online-Backpropagationverfahren letztlich durch

Gewichtsänderung

$$\begin{aligned} \Delta w_{ij} &= \Delta w_{ij,p} = -\eta g_{ij,p} = -\eta y_{i,p} \delta_{j,p} \\ &= \begin{cases} -\eta y_{i,p} y_{j,p}^o (1 - y_{j,p}^o) (\mathbf{t}_{j,p} - y_{j,p}^o) & , \text{ falls } v_j \text{ Ausgabeneuron} \\ -\eta y_{i,p} y_{j,p}^h (1 - y_{j,p}^h) \sum_k \delta_{k,p}^o w_{jk} & , \text{ falls } v_j \text{ verdecktes Neuron} \end{cases} \end{aligned} \quad (7.34)$$

mit den je nach Neurontyp mit (7.31) und (7.33) bestimmten Fehlertermen $\delta_{j,p}$. Bemerkenswert hierbei ist, dass eine Gewichtsänderung nur dann stattfindet, wenn die Netzeingabe $y_{i,p} \neq 0$ ist. Dies ist vor allem für die Wahl der Eingangskodierung von Bedeutung. Für die Wahl der Ausgabekodierung ist von Bedeutung, dass die Gewichtsänderung wegen

$y_{j,p}(1-y_{j,p})$ für den Wert 0,5 maximal und an den Rändern minimal wird.

Zusammenfassung

Zusammenfassend ist der Backpropagation-Algorithmus ein Gradientenabstiegsverfahren, das für jedes Trainingsbeispiel p den Fehler E_p zwischen Netzausgabe und Sollwert berechnet, den Gradienten $g_{ij,p}$ bildet und anschließend die Verbindungsgewichte w_{ij} mit einer zuvor festgelegten Lernrate η sukzessive in Richtung des Gradienten modifiziert, bis ein Optimum gefunden wurde (vgl. dazu nochmals Abbildung 7.5).

Damit wurden die für das Verständnis der Arbeit notwendigen Grundlagen gelegt. Für weiterführende Studien zum Thema neuronale Netze wird der Leser auf die entsprechende Fachliteratur wie z.B. [BR07], [Zel96], [Spe96], [Hof93] oder [Kre01] verwiesen.

7.2. Einschichtige neuronale Diagnostik-Modelle

Im Nachfolgenden wird zunächst gezeigt, dass das in Definition 6.2 hergeleitete inferentielle Diagnostik-Modell unter Voraussetzung einiger Randbedingungen, wie die geeignete Wahl der Aktivierungsfunktion und des Wertebereichs der Eingangsneuronen, als einschichtiges neuronales Netz modelliert werden kann. Darauf aufbauend wird das einschichtige Diagnostik-Modell definiert und analysiert. Der Vorteil eines solchen Vorgehens ist die damit mögliche wahrscheinlichkeitstheoretische Betrachtungsweise neuronaler Netze, die der Veranschaulichung der Funktionsweise dient und damit der weit verbreiteten „Blackbox“-Sichtweise entgegenwirkt. Weiterhin werden einige Zuverlässigkeits- und Grenzwertabschätzungen möglich, wie später noch gezeigt werden wird. Nicht zuletzt erlaubt die Betrachtungsweise die Interpretation der gelernten Verbindungsgewichte des Netzes und macht sie so für weitere Auswertungen nutzbar.

7.2.1. Wahrscheinlichkeitstheoretische Interpretation neuronaler Diagnostik-Modelle

Beginnend mit einem einfachen Neuron, kann dessen Ausgabewert $y(\mathbf{x})$ ⁸ bei gegebenen Eingabevektor $\mathbf{x}$ berechnet werden mit

Ein sigmoides Neuron

$$y(\mathbf{x}) = f_{out}(f_{act}(f_{prop}(\mathbf{x}))) . \quad (7.35)$$

Als Ausgabefunktion f_{out} wird die Identität gewählt. Als Aktivierungsfunktion f_{act} wird die im vorhergehenden Abschnitt eingeführte Sigma- bzw. logistische Funktion gewählt. Der prinzipielle Verlauf der Sigmafunktion für verschiedene $\mathbf{w}$ ist in Abbildung 7.4 dargestellt. Der Ausgabewert $y(\mathbf{x})$ eines Neurons berechnet sich dann mit

$$y(\mathbf{x}) = \frac{1}{1 + e^{-\alpha(\mathbf{x})}} . \quad (7.36)$$

Die Exponentenfunktion $\alpha(\mathbf{x})$ entspricht dabei der Propagierungsfunktion f_{prop} , eine lineare Funktion des Eingabevektors $\mathbf{x}$ mit

$$\alpha(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \Theta . \quad (7.37)$$

Dabei ist $\mathbf{w}$ der (partielle) Gewichtsvektor, d.h. der Teil der Adjazenzmatrix W aus (7.1), der die Verbindungsgewichte zu diesem Neuron enthält, und Θ der Bias. Das Skalarprodukt $\mathbf{w}^T \mathbf{x}$ entspricht hier der in Gleichung (7.3) definierten gewichteten Summe. Wegen (7.37) wird diese Form in der Literatur auch als verallgemeinertes lineares Modell („generalized

⁸ Die fettgedruckte Schreibweise $\mathbf{x}$ oder $\mathbf{w}$ steht hier für Vektoren.

linear model“, vgl. [Bis06, S.192ff.]) bezeichnet, woraus einige Vorteile resultieren, wie die später folgenden Untersuchungen zeigen werden.

Optimaler Dekoder

Um die Äquivalenz der Modelle zu belegen, muss eine Verbindung zur A-posteriori-Wahrscheinlichkeit aus (6.13) gelingen. Ein Lösungsansatz bietet hier das aus der Informationstheorie bekannte Verfahren des *optimalen Dekoders* (vgl. dazu [Mac07, S.152], The optimal decoder). Angewandt auf (6.13) ergibt sich dieser durch

$$\hat{k}_{\text{optimal}} = \arg \max_{k'_i \in K_\Sigma} P(k'_i | S_I, \mathcal{H}) . \quad (7.38)$$

A-posteriori-Wahrscheinlichkeitsverhältnis

Für die Bestimmung von $\hat{k}_{\text{optimal}}$ genügt daher der Vergleich der A-posteriori-Wahrscheinlichkeitsverhältnisse

$$\frac{P(k = k_i | S_I, \mathcal{H})}{P(k \neq k_i | S_I, \mathcal{H})} = \frac{P(S_I | k = k_i, \mathcal{H}) P(k = k_i | \mathcal{H})}{P(S_I | k \neq k_i, \mathcal{H}) P(k \neq k_i | \mathcal{H})} , \quad (7.39)$$

so dass der Nenner von (6.13) irrelevant wird. Der Term $k \neq k_i$ ist dabei zu interpretieren mit $k = \{k_1, \dots, k_{i-1}, k_{i+1}, \dots, k_n\}$, also die Menge aller Komponenten außer k_i .

Ausgehend vom optimalen Dekoder wird zunächst das zur Basis e logarithmierte A-posteriori-Wahrscheinlichkeitsverhältnis nach (7.39) betrachtet und als Exponentenfunktion $\alpha(\mathbf{x})$ angenommen, so dass

$$\alpha(S_I) = \ln \left(\frac{P(k = k_i | S_I, \mathcal{H})}{P(k \neq k_i | S_I, \mathcal{H})} \right) \quad (7.40)$$

mit S_I als Eingabevektor. Mit dem Normierungsaxiom

$$\sum_{k_j \in K_\Sigma} P(k = k_j | S_I, \mathcal{H}) = 1 \quad (7.41)$$

kann der Numerus in ein *Odds*⁹ und $\alpha(S_I)$ in ein *Logit*¹⁰ überführt werden, so dass

$$\alpha(S_I) = \text{Logit}(k = k_i | S_I, \mathcal{H}) = \ln \left(\frac{P(k = k_i | S_I, \mathcal{H})}{1 - P(k = k_i | S_I, \mathcal{H})} \right) . \quad (7.42)$$

⁹ Die „*Odds*“, auch Chancenverhältnis genannt, beschreiben das Verhältnis der Eintrittswahrscheinlichkeit eines Ereignisses zu seiner Gegenwahrscheinlichkeit (vgl. [AHK97, S.142], [Men01, S.12 ff. - Kap.1.3]).

¹⁰ An diesem Punkt existieren Parallelen zur logistischen Regressionsanalyse, in der ebenfalls mit logarithmierten *Odds*, dann *Logit* genannt, gerechnet wird. Eine ausführliche Behandlung wird beispielsweise in [HL00] gegeben. $\text{Logit}(x_{1/0}) = \ln(\text{Odds}(x_{1/0})) = \ln \left(\frac{P(x=1)}{1-P(x=1)} \right)$.

Durch Exponentierung folgt

$$e^{\alpha(S_I)} = Odds(k = k_i | S_I, \mathcal{H}) = \frac{P(k = k_i | S_I, \mathcal{H})}{1 - P(k = k_i | S_I, \mathcal{H})} . \quad (7.43)$$

Umgestellt auf den Wahrscheinlichkeitsterm $P(k = k_i | S_I, \mathcal{H})$ ergibt sich

$$P(k = k_i | S_I, \mathcal{H}) = \frac{e^{\alpha(S_I)}}{e^{\alpha(S_I)} + 1} = \frac{1}{1 + e^{-\alpha(S_I)}} . \quad (7.44)$$

Damit wurde gezeigt, dass mit Hilfe des optimalen Dekoders und dem A-posteriori-Wahrscheinlichkeitsverhältnis die allgemeine Form eines Neurons mit sigmoider Aktivierungsfunktion hergeleitet werden kann. Dabei ist die Äquivalenz zur Form in (7.36) zwar offensichtlich, jedoch muss im Folgenden noch die Exponentenfunktion $\alpha(\mathbf{x})$ bestimmt werden, um eine vollständige Äquivalenz beweisen zu können.

Wird die Berechnung der Exponentenfunktion $\alpha(\mathbf{x})$ ausgeschrieben, ergibt sich mit dem Skalarprodukt

Die Exponentenfunktion $\alpha(\mathbf{x})$

$$\alpha(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \Theta = \sum_{j \in J} (\mathbf{w}_j \mathbf{x}_j) + \Theta = \sum_{j \in J} (\mathbf{w}_j \mathbf{x}_j + \Theta_j^{'}) + \Theta_0^{'}, \quad (7.45)$$

mit $\sum_{j \in J} \Theta_j^{' } + \Theta_0^{' } = \Theta$. Mit der Bedingung der Unabhängigkeit nach (6.14) folgt wie auch schon beim inferentiellen Diagnostik-Modell

$$P(S_I | k = k_i, \mathcal{H}) = P(s_{I1} | k_i, \mathcal{H}) \cdot P(s_{I2} | k_i, \mathcal{H}) \cdot \dots \cdot P(s_{In} | k_i, \mathcal{H}) . \quad (7.46)$$

Das A-posteriori-Wahrscheinlichkeitsverhältnis aus (7.39) kann so umgeschrieben werden zu

$$\frac{P(k = k_i | S_I, \mathcal{H})}{P(k \neq k_i | S_I, \mathcal{H})} = \frac{\left[\prod_{s_j \in S_I} P(s_j | k = k_i, \mathcal{H}) \right] \cdot P(k = k_i | \mathcal{H})}{\left[\prod_{s_j \in S_I} P(s_j | k \neq k_i, \mathcal{H}) \right] \cdot P(k \neq k_i | \mathcal{H})} . \quad (7.47)$$

Durch Logarithmierung des gesamten Terms zur Basis e wird vorerst eine Überführung der Multiplikationsketten in Additionsketten erreicht:

$$\ln \left(\frac{P(k = k_i | S_I, \mathcal{H})}{P(k \neq k_i | S_I, \mathcal{H})} \right) = \sum_{s_j \in S_I} \ln \left(\frac{P(s_j | k = k_i, \mathcal{H})}{P(s_j | k \neq k_i, \mathcal{H})} \right) + \ln \left(\frac{P(k = k_i | \mathcal{H})}{P(k \neq k_i | \mathcal{H})} \right) \quad (7.48)$$

Eine Betrachtung der Terme der rechten Seite zeigt, dass lediglich der Summenterm eine Abhängigkeit von S_I besitzt. Darum soll dieser zunächst in die allgemeine Form des Exponenten aus (7.45) überführt werden. Für ein einzelnes Symptom $s_j \in S_I$ bzw. für einen einzelnen Term in der Summe gilt dann

$$\ln \left(\frac{P(s_j | k = k_i, \mathcal{H})}{P(s_j | k \neq k_i, \mathcal{H})} \right) \stackrel{!}{=} \mathbf{w}_j s_j + \Theta_j^{' } . \quad (7.49)$$

Nach Definition 5.1 kann ein einzelnes Symptom die zwei Zustände 0 und 1, im Folgenden mit $s_j = 0$, $s_j = 1$ bzw. s_{j_0} , s_{j_1} bezeichnet, annehmen. Damit ergeben sich für die zwei Fälle folgende Ausdrücke

$$s_i = 0 : \quad \ln \left(\frac{P(s_{j_0}|k = k_i, \mathcal{H})}{P(s_{j_0}|k \neq k_i, \mathcal{H})} \right) = \mathbf{w}_j s_{j_0} + \Theta_j^{\epsilon} , \quad (7.50)$$

$$s_i = 1 : \quad \ln \left(\frac{P(s_{j_1}|k = k_i, \mathcal{H})}{P(s_{j_1}|k \neq k_i, \mathcal{H})} \right) = \mathbf{w}_j s_{j_1} + \Theta_j^{\epsilon} , \quad (7.51)$$

$$\Rightarrow \mathbf{w}_j = \frac{\ln \left(\frac{P(s_{j_1}|k=k_i, \mathcal{H})}{P(s_{j_1}|k \neq k_i, \mathcal{H})} \right) - \Theta_j^{\epsilon}}{s_{j_1}} . \quad (7.52)$$

Mit (7.52) in (7.50) ergibt sich für Θ_j^{ϵ}

$$\Theta_j^{\epsilon} = \frac{\ln \left(\frac{P(s_{j_0}|k=k_i, \mathcal{H})}{P(s_{j_0}|k \neq k_i, \mathcal{H})} \right) - \ln \left(\frac{P(s_{j_1}|k=k_i, \mathcal{H})}{P(s_{j_1}|k \neq k_i, \mathcal{H})} \right) \frac{s_{j_0}}{s_{j_1}}}{1 + \frac{s_{j_0}}{s_{j_1}}} \quad (7.53)$$

und mit (7.53) in (7.51) folgt für $\mathbf{w}_j$

$$\mathbf{w}_j = \frac{\ln \left(\frac{P(s_{j_1}|k=k_i, \mathcal{H})}{P(s_{j_1}|k \neq k_i, \mathcal{H})} \right)}{s_{j_1}} - \frac{\ln \left(\frac{P(s_{j_0}|k=k_i, \mathcal{H})}{P(s_{j_0}|k \neq k_i, \mathcal{H})} \right) - \ln \left(\frac{P(s_{j_1}|k=k_i, \mathcal{H})}{P(s_{j_1}|k \neq k_i, \mathcal{H})} \right) \frac{s_{j_0}}{s_{j_1}}}{s_{j_1} + s_{j_0}} . \quad (7.54)$$

Eingangskodierung
 s_{j_1}, s_{j_0}

Die Ausprägungen s_{j_1} und s_{j_0} entsprechen nach (7.37) der quantitativen Eingangskodierung eines Eingangsneurons. Werden diese geschickt gewählt mit $s_{j_1} = 1$ und $s_{j_0} = 0$, vereinfachen sich die Gleichungen (7.53) und (7.54) zu

$$\Theta_j^{\epsilon} = \ln \left(\frac{P(s_{j_0}|k = k_i, \mathcal{H})}{P(s_{j_0}|k \neq k_i, \mathcal{H})} \right) \quad (7.55)$$

und

$$\mathbf{w}_j = \ln \left(\frac{P(s_{j_1}|k = k_i, \mathcal{H})}{P(s_{j_1}|k \neq k_i, \mathcal{H})} \right) - \ln \left(\frac{P(s_{j_0}|k = k_i, \mathcal{H})}{P(s_{j_0}|k \neq k_i, \mathcal{H})} \right) . \quad (7.56)$$

Damit kann nun Θ nach (7.37) bestimmt werden mit

$$\Theta = \sum_{j \in J} \ln \left(\frac{P(s_{j_0}|k = k_i, \mathcal{H})}{P(s_{j_0}|k \neq k_i, \mathcal{H})} \right) + \ln \left(\frac{P(k = k_i, \mathcal{H})}{P(k \neq k_i, \mathcal{H})} \right) . \quad (7.57)$$

Der letzte Term entspricht dabei Θ_0^{ϵ} und ergibt sich aus dem rechten Symptom-abhängigen Term in (7.48). Die Gleichung (7.44) kann mit den Parametern (7.56) und (7.57) als einzelnes Neuron mit je einem Eingang pro Symptom s_j gesehen werden (vgl. Abbildung 7.6).

Abbildung 7.6.: Einzelnes Neuron als Klassifikator.

Die Abhängigkeit von k_i macht deutlich, dass für jede Komponente k_i ein eigenes Ausgangsneuron mit den Parametern $\mathbf{w}^{k_i} = W_{(k_i)}$ und Θ_{k_i} existieren muss¹¹. Daraus folgt die Definition eines ersten (einschichtigen) neuronalen Diagnostik-Modells, das in Abbildung 7.7 vereinfacht dargestellt ist.

Definition 7.4 (Einschichtiges neuronales Diagnostik-Modell)

Sei S_Σ die Menge aller möglichen Symptome und K_Σ die Menge aller möglichen Komponenten, dann ist ein einschichtiges neuronales Diagnostik-Modell Π_{neuro} , das eine Menge von Symptomen $S_I \subseteq S_\Sigma$ auf eine Menge an Diagnostik-Hypothesen $H_D \subseteq D_\Sigma$ abbildet, definiert mit

$$\Pi_{neuro} : \mathcal{P}(S_\Sigma) \rightarrow \mathcal{P}(D_\Sigma), S_I \mapsto \Pi_{neuro}(S_I) := \bigcup_{k_i \in K_\Sigma} \left\{ \left(k_i, \frac{1}{1 + e^{-\alpha_{k_i}(\mathbf{S}_I)}} \right) \right\}$$

wobei die Exponentenfunktion $\alpha_{k_i}(\mathbf{S}_I)$, eine lineare Funktion des Eingabevektors $\mathbf{S}_I$, mit

$$\alpha_{k_i}(\mathbf{S}_I) = \mathbf{w}^{k_i^T} \mathbf{S}_I + \Theta_{k_i}$$

definiert ist.

Mit der Bestimmung der Parameter $\mathbf{w}_j$ und Θ_j konnte die Äquivalenz mit dem inferentiellen Diagnostik-Modell nach Definition 7.4 gezeigt werden. Randbedingungen für die Äquivalenz ist ein einschichtiges feed-forward vernetztes neuronales Netz mit sigmoider Aktivierungsfunktion und einer Eingangskodierung der Symptome nach (5.1) mit $s_{j_1} = 1$ und $s_{j_0} = 0$.

Q.E.D.

¹¹ Die Notation $W_{(k_i)}$ steht für den zu Neuron v_{k_i} gehörenden Spaltenvektor der Gewichtsmatrix W .

Abbildung 7.7.: Einschichtiges neuronales Diagnostik-Modell nach Definition 6.2, bestehend aus einem einstufigen neuronalen Netz mit sigmoider Aktivierungsfunktion (die Bias Θ_{k_i} sind nicht dargestellt)

Adjazenzmatrix

Auch hier kann für das resultierende Netz eine Adjazenzmatrix analog zu (7.1) definiert werden. Diese setzt sich aus den einzelnen Gewichtsvektoren $\mathbf{w}^{k_i}$ zusammen, so dass

$$W = \begin{bmatrix} 0 & W_I \\ 0 & 0 \end{bmatrix} \text{ mit} \quad (7.58)$$

$$W_I = \begin{bmatrix} \mathbf{w}^{k_1} & \mathbf{w}^{k_2} & \dots & \mathbf{w}^{k_n} \end{bmatrix} \quad (7.59)$$

$$W_I = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nn} \end{bmatrix}.$$

Das resultierende Netz ist zusammenfassend in Abbildung 7.7 dargestellt. Hierbei gilt $w_{ij} = \mathbf{w}_i^{k_j} \in W$, d.h. w_{ij} stellt das Verbindungsgewicht von Symptom s_i zu Komponente k_j dar.

7.2.2. Interpretation der Verbindungsgewichte

Mit der Herleitung des neuronalen Diagnostik-Modells nach Definition 6.2 konnte nicht nur die Äquivalenz mit dem inferentiellen Diagnostik-Modell gezeigt werden, sondern vielmehr lassen die Gleichungen (7.56) und (7.57) eine Interpretation der Gewichte eines Netzes zu. Hierzu sind Vereinfachungen und Umformungen notwendig, die bisher aus Anschauungsgründen nicht durchgeführt wurden. Durch Umformen wird (7.56) zu

$$w_{ji} = \ln \left(\frac{P(s_{j_1}|k = k_i, \mathcal{H})}{P(s_{j_0}|k = k_i, \mathcal{H})} \cdot \frac{P(s_{j_0}|k \neq k_i, \mathcal{H})}{P(s_{j_1}|k \neq k_i, \mathcal{H})} \right). \quad (7.60)$$

Mit $P(s_{j_0}|k = k_i, \mathcal{H}) + P(s_{j_1}|k = k_i, \mathcal{H}) = 1$ kann der Logarithmand in ein Odds Ratio¹² überführt werden, so dass

$$w_{ji} = \ln \left(\frac{\text{Odds}(s_{j_1}|k = k_i, \mathcal{H})}{\text{Odds}(s_{j_1}|k \neq k_i, \mathcal{H})} \right) \quad (7.61)$$

und

$$\text{OR}(s_{j_1}|k_{i1/0}, \mathcal{H}) = \frac{\text{Odds}(s_{j_1}|k = k_i, \mathcal{H})}{\text{Odds}(s_{j_1}|k \neq k_i, \mathcal{H})} = e^{-w_{ji}}. \quad (7.62)$$

Veranschaulicht lässt sich somit aus einem Gewichtsvektor w_{ji} ein Chancenverhältnis (Odds Ratio $\text{OR}(s_{j_1}|k_{i1/0}, \mathcal{H})$) zwischen der Chance, dass das Symptom s_j ohne eine defekte Komponente k_i und der Chance, dass das Symptom s_{j_1} bei defekter Komponente k_i auftritt, ableiten.

Interpretation als Chancenverhältnis zwischen Symptomen

Eine weitere Interpretation wird möglich, wenn das gesamte Neuron aus Gleichung (7.43) betrachtet wird. Wird ebenfalls ein Odds Ratio gebildet, hier mit zwei verschiedenen Symptomatiken S_{I_0} und S_{I_1} , folgt

$$\text{OR}(k_i|S_{I_1/0}, \mathcal{H}) = \frac{\text{Odds}(k = k_i|S_{I_1}, \mathcal{H})}{\text{Odds}(k = k_i|S_{I_0}, \mathcal{H})} = \frac{e^{\sum(w_{ji}s_{j_1}) + \Theta}}{e^{\sum(w_{ji}s_{j_0}) + \Theta}} \quad (7.63)$$

$$= e^{\sum(w_{ji}(s_{j_1} - s_{j_0}))} \\ = \prod_{j \in J} e^{w_{ji}(s_{j_1} - s_{j_0})}. \quad (7.64)$$

Interpretierbar wird $\text{OR}(k_i|S_{I_1/0}, \mathcal{H})$, wenn sich in der Symptomatik $S_{I_1/0}$ nur ein Symptom s_j ändert. Wegen dem Term $(s_{j_1} - s_{j_0})$ folgt dann

$$\text{OR}(k_i|S_{j_1/0}, \mathcal{H}) = \frac{\text{Odds}(k = k_i|s_{j_1}, \mathcal{H})}{\text{Odds}(k = k_i|s_{j_0}, \mathcal{H})} = e^{w_{ji}(s_{j_1} - s_{j_0})} = e^{w_{ji}} \quad (7.65)$$

$$w_{ji} = \ln \left(\text{OR}(k_i|s_{j_1/0}, \mathcal{H}) \right). \quad (7.66)$$

So kann aus dem Verbindungsgewicht w_{ji} das Chancenverhältnis zwischen der Chance für den Defekt der Komponente k_i bei vorhandenem Symptom s_j zu der Chance für den Defekt der Komponente k_i bei nicht vorhandenem Symptom s_j bestimmt werden. Die symmetrische Interpretierbarkeit von (7.66) und (7.62) ist ausschließlich mit der Unabhängigkeitsannahme aus (6.14) möglich. Wäre diese verletzt, würde zumindest erstere Interpretation (7.62) mit zunehmenden Abhängigkeiten zunehmend ungenauer werden. Abbildung 7.8 zeigt beispielhaft den Zusammenhang zwischen dem Odds-Ratio (7.65) und dem Verbin-

Interpretation als Chancenverhältnis zwischen Komponenten

¹² Eine sehr anschauliche Einführung zu Odds Ratios wird beispielsweise in [Fro05, Kap.3 ff.] oder [KK02, S.11ff.] gegeben. Tiefergehende Betrachtungen finden sich u. a. in [Chr97, S.29ff.].

dungsgewicht w_{ji} . Für ein Gewicht von $w_{ji} = 0$ wird das Odds-Ratio wie zu erwarten 1 : 1, d.h. das Symptom s_j besitzt keinen Einfluss auf die Komponente k_i . Mit steigendem Gewicht nimmt auch der Einfluss exponentiell zu. Mit zunehmend negativ werdendem Gewicht nimmt der Einfluss negativ zu. Die Symmetrie wird durch das ebenfalls dargestellte inverse Odds-Ratio veranschaulicht.

Abbildung 7.8.: Odds-Ratio und inverses Odds-Ratio im Verhältnis zum Verbindungsgewicht w_{ji} .

Zusammenfassung

Die Interpretation der Verbindungsgewichte w_j lassen zusammenfassend die Stärke und die Richtung des Einflusses eines Symptoms s_j auf den Defekt einer Komponente k_i qualitativ abschätzen. Eine Ableitung quantitativer Wahrscheinlichkeiten ist schon wegen dem kumulativen Bias Θ und der damit verbundenen Modellkomprimierung nicht möglich. Weiterhin existiert aufgrund der nichtlinearen Aktivierungsfunktion keine lineare Abhängigkeit zwischen Symptom und Komponente. Vielmehr ist der Einfluss eines Symptoms, je nach Vorhandensein weiterer Symptome, unterschiedlich (vgl. Abbildung 7.4). Mit der relativ geringen Auftretswahrscheinlichkeit eines einzelnen Symptoms und der damit verbundenen niedrigen Anzahl Symptome pro Reparaturfall scheint eine Interpretation als Chancenverhältnis jedoch sinnvoll und nützlich.

7.3. Multi-Layer-Perceptrons als Diagnostik-Modelle

In den bisherigen Betrachtungen wurde der Fahrzeugkontext C vernachlässigt. Ein Grund hierfür war die Komplexitätsbeschränkung der Modelle sowie die Annahme, Symptome seien untereinander unabhängig. Die Annahme hätte mit den eingeführten Vereinfachungen so auch für den Fahrzeugkontext gelten müssen. Tatsächlich jedoch ist die Unabhängigkeitsannahme gerade hierfür nicht erfüllt, wie folgendes Beispiel zeigt.

Unabhängigkeitsbetrachtung $s_i \perp c_j$

7.3.1. Unabhängigkeitsbetrachtung des Fahrzeugkontexts

Es sei angenommen, ein Fahrzeugmodell wurde zuerst mit einem Motorsteuergerät (MSG) der Version $c_{Version} = \beta$ und später in einer ausgereiften Version γ ausgeliefert. Für die letztere Version existiert entsprechend auch ein optimierter Nockenwellensensor, der infolge dessen eine geringere Defektwahrscheinlichkeit als die erstere Version besitzt. Das Motorsteuergerät (MSG) überwacht den Sensor und erzeugt im Fehlerfall (meistens) den Ereignisspeichereintrag (DTC) s_1 . In der Fallbasis R_{FB} existieren dazu folgende in Tabelle 7.1 dargestellten Fälle, die kurz erläutert werden sollen.

Beispiel Nockenwellensensor

	Anz. Fälle	c_V^a	s_1^b	k_1	Kommentar
1	25.000	γ	0	0	Defekt anderer Komponenten, ähnliche Häufigkeit beider Versionen
2	25.000	β	0	0	
3	16	γ	1	0	Kabelsatz defekt, nicht k_1 , ähnliche Häufigkeit
4	17	β	1	0	
5	3	γ	0	1	k_1 defekt ohne Symptom, ähnliche Häufigkeit
6	4	β	0	1	
7	5	γ	1	1	k_1 für Version γ selten defekt
8	78	β	1	1	k_1 für Version β häufiger defekt

^a Hard- und Softwareversion $c_{Version}$ des Steuergerätes c_{ecu} .

^b Symptom (Ereignisspeichereintrag) im Steuergerät c_{ecu} .

Tabelle 7.1.: Vertrauenswerte des inferentiellen Diagnostik-Modells nach Definition 6.2 für kontextabhängige Daten.

An den ersten beiden Zeilen kann abgelesen werden, dass beide Versionen ungefähr gleich häufig vorkommen, Fahrzeuge der neueren Version haben folglich die der älteren noch nicht verdrängt. Die Zeilen 3 und 4 zeigen, dass mit s_1 vom MSG ein Fehler erkannt wurde, defekt war jedoch nicht der Sensor, sondern der Kabelsatz, d. h. $k_1 = 0$. Der Defekt

des Kabelsatzes ist für beide Varianten gleich wahrscheinlich. In den Zeilen 5 und 6 ist wiederum in seltenen Fällen die Komponente k_1 defekt, ohne dass der Fehler vom MSG mit einem Symptom s_1 dazu erkannt wurde. Besonders relevant sind nun die letzten beiden Zeilen. Demnach ist der Defekt des Nockenwellensensors der älteren Version β wie zu erwarten sehr viel häufiger als in Version γ . Wäre nun das Symptom s_1 statistisch unabhängig vom Fahrzeugkontext, hier also der Version $c_{Version} \in \{\beta, \gamma\}$, müsste allgemein gelten

$$P(s_i|c_{Version}) \stackrel{!}{=} P(s_i). \quad (7.67)$$

Bedingte
Unabhängigkeit

Damit wäre die Unabhängigkeitsbedingung $s_i \perp c_{Version}$. Die Zahlen des Beispiels zeigen jedoch, dass z. B. $P(s_1|c_\gamma) = \frac{21}{25.024} = 0,84 \cdot 10^{-3}$, dagegen aber $P(s_1) = \frac{116}{50.123} = 2,3 \cdot 10^{-3}$ und die Unabhängigkeitsbedingung damit nicht erfüllt ist. Wäre die Häufigkeit in Zeile 8 in etwa gleich der in Zeile 7, wäre die Bedingung erfüllt. Von Interesse wäre eine Prüfung, ob die Unabhängigkeit wenigstens bei gegebener Komponente k_1 gilt, also allgemein

$$P(s_i|c_{Version}, k_j) = P(s_i|k_j) \quad (7.68)$$

erfüllt ist. Das Zahlenbeispiel zeigt aber auch hier, dass $P(s_1|c_\gamma, k_1) = \frac{5}{8} = 62,5\%$, jedoch $P(s_1|k_1) = \frac{83}{90} = 92\%$. Somit ist auch die bedingte Unabhängigkeit nicht erfüllt.

Worst Case: Defekt-
wahrscheinlichkeiten
ohne Symptome

Würde der Fahrzeugkontext $c_x \in C$ ungeachtet dessen im einschichtigen neuronalen Modell wie die Symptome s_i behandelt werden, könnte in Diagnostiksessions, in denen kein Symptom vorhanden ist, die Defektwahrscheinlichkeit einer Komponente allein aufgrund einer bestimmten Version c_x Werte > 0 annehmen, d. h. $P(k_j|c_x, \mathcal{H}) > 0$. Würden im gegebenen Beispiel wesentlich mehr verschiedene Versionen verbaut werden, wie es in der Realität der Fall ist, so wäre für eine Häufigkeit in Zeile 2 von $\#c_\beta = 200$, $P(k_j|c_\beta, \mathcal{H}) = \frac{82}{299} = 27,4\%$. Dies konnte in Experimenten bestätigt werden.

7.3.2. Eine Modeldefinition

Das Beispiel zeigt, dass gerade für die Berücksichtigung des Fahrzeugkontext C die Unabhängigkeitsannahme, hier zwischen Kontext und Symptom mit $s_i \perp c_x$, nicht hält, sondern vielmehr zu falschen inakzeptablen Ergebnissen führt. Es gibt eine Reihe von Lösungsansätzen, die Komplexität des Modells zu erweitern und damit auch Abhängigkeiten abbilden zu können. Der am weitesten verbreitete Ansatz hierfür ist die Integration einer zusätzlichen (verdeckten) Schicht an Neuronen (engl. „*hidden layer*“), also die Erweiterung des Netzes zum *Multi-Layer-Perceptron* (MLP). Die erforderlichen Grundlagen dazu wurden bereits in Kapitel 7.1 eingeführt. Ein solches, an die Aufgabenstellung angepasstes Netz ist in Abbildung 7.9 dargestellt. Eingangsgrößen sind Symptome S_R und Kontextinformationen C_R . Die Ausgangsgrößen sind hier auf-

Komplexere Mo-
delle notwendig

Abbildung 7.9.: Multi-Layer-Perceptron als neuronales Diagnostik-Modell (die Bias Θ_{k_i} sind nicht dargestellt)

grund der mangelnden statistischen Beschreibbarkeit von MLPs nicht mehr bedingte Wahrscheinlichkeiten, wie sie z.B. in Kapitel 7.2 mit (7.44) beschrieben wurden, sondern allgemein Vertrauenswerte v_{k_j} von Diagnose-Hypothesen h_{k_j} mit $h_{k_j} = (k_j, v_{k_j})$ (vgl. dazu auch Gleichung (2.7)). Die Gewichtsmatrix W wird dann analog zu Gleichung (7.9) beschrieben mit

$$W = \begin{bmatrix} 0 & W_I & 0 \\ 0 & 0 & W_O \\ 0 & 0 & 0 \end{bmatrix}, \quad (7.69)$$

wobei W_I und W_O den Gleichungen (7.11) und (7.12) entsprechen. Die Berechnung des Vertrauenswertes für eine Komponente k_j erfolgt gemäß (7.14) mit

Berechnung der
Vertrauenswerte

$$v_{k_j}(\mathbf{s}, \mathbf{c}) = f_{act} \left(\sum_{a=1}^{n_H} w_{h_a o_{k_i}} f_{act} \left(\sum_{b=1}^{n_S} w_{i_{s_b} h_a} \mathbf{s}_b + \sum_{b=1}^{n_C} w_{i_{c_b} h_a} \mathbf{c}_b + \Theta_{h_a} \right) + \Theta_{o_{k_i}} \right) \quad (7.70)$$

mit f_{act} als sigmoide Aktivierungsfunktion nach (7.17), $w_{h_x o_{k_i}}$ als Verbindungsgewicht vom verdeckten Neuron h_x zum Ausgangsneuron o_{k_i} , Θ_{h_x} als Bias des verdeckten Neurons h_x und $\Theta_{o_{k_i}}$ als Bias des Ausgangsneurons o_{k_i} . Darauf basierend kann nun die Definition für ein Multi-Layer-Perceptron-Diagnostik-Modell formuliert werden:

Definition 7.5 (Multi-Layer-Perceptron-Diagnostik-Modell)

Sei S_Σ die Menge aller möglichen Symptome, C_Σ die Menge aller möglichen Kontextinformationen und K_Σ die Menge aller möglichen Komponenten, dann ist ein Multi-Layer-Perceptron-Diagnostik-Modell Π_{MLP} ,

das eine Menge von Symptomen $S_I \subseteq S_\Sigma$ bei gegebenen Kontext $C_I \subseteq S_\Sigma$ auf eine Menge an Diagnostik-Hypothesen $H_D \subseteq D_\Sigma$ abbildet, definiert mit

$$\Pi_{MLP} : \mathcal{P}(S_\Sigma) \times \mathcal{P}(C_\Sigma) \rightarrow \mathcal{P}(D_\Sigma), \quad S_I \times C_I \mapsto \Pi_{MLP}(S_I, C_I) := \bigcup_{k_i \in K_\Sigma} \left\{ \left(k_i, f_{act} \left(\sum_{a=1}^{n_H} w_{h_a o_{k_i}} f_{act} \left(W_{I[a]}^T \begin{bmatrix} \mathbf{C}_I \\ \mathbf{S}_I \end{bmatrix} + \Theta_{h_a} \right) + \Theta_{o_{k_i}} \right) \right) \right\}$$

wobei f_{act} als sigmoide Aktivierungsfunktion mit $f_{act}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$, Θ_v als Bias des Neurons v und W_I als Teilmatrix der Gewichtsmatrix mit $W_I = \{w_{i_x h_a}\} \in W$ definiert ist.

7.3.3. Eigenschaftsanalyse des MLP-Diagnostik-Modells

Darstellungsmächtigkeit mehrschichtiger neuronaler Netze

Die Darstellungsmächtigkeit ist die Antwort auf die Frage, ob MLPs in der Lage sind, beliebige bedingte Wahrscheinlichkeiten abzubilden. Nach Fouriers Theorem kann jede stetige Funktion $y(\mathbf{x})$ beliebig genau mit einer (möglicherweise unendlichen) Summe harmonischer Funktionen approximiert werden. Wird ein neuronales Netz mit harmonischen Aktivierungsfunktionen in den Neuronen der verdeckten Schicht angenommen, dann würde die Gewichtsmatrix W_O den Koeffizienten der Fourieranalyse entsprechen, wodurch ein solches Netz die gewünschte Funktion implementieren würde. Tatsächlich werden hierfür nicht einmal harmonische Funktionen benötigt, sondern es genügen bereits Sigmafunktionen, wozu auch die hier verwendete logistische Funktion gehört. Zusammengefasst sind MLPs, genügend Hidden-Neuronen vorausgesetzt, in der Lage, jede beliebige stetige Funktion und damit auch jede statistische Verteilung und jede bedingte Abhängigkeit darzustellen (vgl. auch [DHS00, S. 287] oder [HSW89, S. 359ff.]).

MLPs und die
Fourieranalyse

Interpretierbarkeit der Verbindungsgewichte

Das Problem von MLPs ist, dass trotz der Darstellungsmächtigkeit eine geeignete Gewichtsmatrix W im Lösungsraum $\mathbb{R}^{|W|}$ gefunden werden muss und analytische Lösungen hierfür nicht existieren. Darum kommen iterative Näherungsverfahren (Trainingsverfahren) wie der später eingeführte Backpropagation-Algorithmus zum Einsatz, die jedoch nicht garantieren können, dass die „beste“ Lösung (hinsichtlich des Lösungsraums) oder im Mindesten eine befriedigende Lösung gefunden wird. Entsprechend wäre eine Interpretation der Gewichtsmatrix W von Vorteil, die jedoch aus verschiedenen Gründen nicht möglich bzw. sinnvoll ist. Dies kann durch das einfache XOR-Muster veranschaulicht werden. Die zugehörige Wahrheitstabelle ist in Tabelle 7.2 dargestellt. Zur Lösung

Ein Beispiel zur
Interpretierbarkeit

	x_1	x_2	z
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

Tabelle 7.2.: Das XOR-Muster

dieser Aufgabe genügt ein MLP-Netz mit 2 Eingangs- und 2 Hiddenneuronen nach Abbildung 7.10. Das Netz kann mit der Gewichtsmatrix W

Abbildung 7.10.: Neuronales Netz für das XOR-Muster

und der Schwellenwertfunktion als Aktivierungsfunktion mit folgender Gleichung beschrieben werden:

$$z(x_1, x_2, W) = f_{act}(w_5 f_{act}(w_1 x_1 + w_3 x_2 + \Theta_1) + \quad (7.71)$$

$$w_6 f_{act}(w_2 x_1 + w_4 x_2 + \Theta_2) + \Theta_3) \quad (7.72)$$

Anschaulich kann ein einzelnes Neuron, wie z.B. v_1 mit zwei Eingängen und Schwellenwertfunktion, die vier in Tabelle 7.3 aufgezeigten logischen Verknüpfungsarten realisieren. Weitere boolsche Verknüpfungsarten sind

Logische Funktion	w_1	w_3	Θ_1
OR	>0	>0	$-\Theta < \min(w_1 + w_2)$
NOR	<0	<0	$\Theta < \min(w_1 + w_2)$
AND	>0	>0	$w_1 < -\Theta < w_1 + w_2$
NAND	<0	<0	$w_1 > -\Theta > w_1 + w_2$

Tabelle 7.3.: Realisierbare Verknüpfungsarten und die korrespondierenden Gewichtsregeln für ein Neuron mit 2 Eingängen

möglich. Daraus resultieren die vier in Tabelle 7.4 dargestellten Möglich-

Vier Varianten zur Realisierung der XOR-Funktion

keiten mit dem dargestellten Netz eine XOR-Funktion zu realisieren¹³. Weitere Varianten außerhalb der Standardoperatoren sind möglich. Die

	v_3	v_1	v_2
1	AND	OR	NAND
2	AND	NAND	OR
3	NOR	NOR	AND
4	NOR	AND	NOR

Tabelle 7.4.: Varianten zur Realisierung der XOR-Funktion

Aufstellung zeigt deutlich, dass für eine Aufgabenstellung bzw. einen Datensatz mehrere Lösungen der Gewichtsmatrix W existieren. Dies liegt nicht zuletzt an der Permutationsfähigkeit der Neuronen. Für n Hiddenneuronen existieren allein 2^n mögliche Permutationen bei identischer Netzfunktion. Es stellt sich nun die Frage, wie ein einzelnes Gewicht, z. B. w_5 , interpretiert werden kann. Das Beispiel zeigt, dass eine Interpretation der Verbindungsgewichte nicht mehr ohne weiteres möglich ist. Für rein boolsche Operationen und einer Schwellenwertfunktion als Aktivierungsfunktion wäre zwar noch eine Interpretation des gesamten Netzes mit einer Aneinanderreihung boolscher Operationen möglich, dies gilt aber nicht mehr für sigmoide Aktivierungsfunktionen und noch weniger für hochdimensionale Merkmalsräume. Eine detailliertere Diskussion zur statistischen Interpretation von neuronalen Netzen ist beispielsweise in [Rip93], [Sar94] und [Whi89] zu finden.

Bewertung von MLPs als Diagnostik-Modell

Vorteile des MLP

Die Popularität von MLP-Netzen begründet sich nicht zuletzt auf einer Reihe von Vorteilen. Ein Wesentlicher davon ist die beschriebene Darstellungsmächtigkeit solcher Netze. So kann ein Netz mit nur einer verdeckten Schicht („*hidden*“-Layer) jede beliebige stetige Funktion beliebig genau approximieren, genügend Neuronen in der verdeckten Schicht vorausgesetzt. Jede wie auch immer geartete Abhängigkeit zwischen Symptomen wie auch zwischen Symptomen und Kontext ist folglich abbildbar. In der Praxis setzt eine optimale Approximation jedoch eine optimale Bestimmung der Parameter wie Anzahl und Aktivierungsfunktion der Neuronen, verwendeter Lernalgorithmus, Abbruchbedingungen, u.s.w. voraus. Arbeiten wie [Pre95, S. 21ff.] zeigen, dass der Erfolg wesentlich von diesen Parametern abhängt. Die Bestimmung der „*richtigen*“ Parameter bzw. die Sensitivität hierzu steigt mit geringer Fallzahl und qualitativ schlechter werdenden Trainingsdaten, wie es für diese Arbeit gegeben ist. Dann wird zwar i. d. R. für das Trainingsset eine gute Performanz erreicht, gleichzeitig sinkt jedoch die Generalisierungsleistung,

Nachteile des MLP

¹³ Eine ausführliche Abhandlung zur Darstellung boolscher Funktionen mit neuronalen Netzen ist beispielsweise in [Koh07] zu finden. Eine Berechnung der möglichen Realisierungsvarianten für das XOR-Problem wird dort jedoch nicht gegeben.

d.h. die Leistungsfähigkeit bei Daten, die nicht Teil der Trainingsmenge sind. Angewendet auf ein Diagnostik-System würde das für Defekte mit zu wenigen Falldaten unter Umständen unsinnige und unverständliche Symptom-Hypothesen h_D bedeuten, die für den praktischen Einsatz nicht tolerierbar sind. Die Punkte konnten für ein prototypisch umgesetztes Modell im Rahmen der Evaluierung in Abschnitt 9.3.5 bestätigt werden. Die Evaluierungsquote q_n des MLP-Modells sank mit steigender Anzahl an verdeckten Neuronen auf 88,3 % (−4,8 %) für Variante 1 bzw. auf 85,7 % (−7,4 %) für Variante 2. Als Konsequenz sind Standard-MLP-Netze für diese Arbeit nicht geeignet.

Konsequenz —
MLP-Netze un-
geeignet

7.4. Komplexitätserweiterte neuronale Diagnostik-Modelle

Mit der geringen Eignung einer pauschalen Komplexitätserweiterung des Lösungsraums, wie das für MLP-Modelle gilt und gezeigt wurde, wird deutlich, dass nur eine gezielte Komplexitätserweiterung, angepasst an die Charakteristik der Falldaten, erfolgreich sein kann.

7.4.1. Lösungsansätze zur gezielten Komplexitätserweiterung

Geforderte Komplexität des Lösungsraums

Zunächst ist damit interessant, welche Komplexität gefordert ist. Das sind Abhängigkeiten zwischen Symptomen $s \in S_\Sigma$ und Komponenten $k \in K_\Sigma$ sowie zwischen Kontextinformationen $c \in C_\Sigma$, Symptomen s und Komponenten k . Nicht gefordert hingegen sind paarweise Abhängigkeiten zwischen Symptomen s_i, s_j , zwischen Komponenten k_i, k_j und zwischen Kontextinformationen c_i, c_j . Zusammengefasst ist somit gefordert:

<i>Forderung</i>			
1	Unabhängigkeit zwischen Symptomen	$s_i \perp s_j$	(7.73)
2	Unabhängigkeit zwischen Kontextinformationen	$c_i \perp c_j$	(7.74)
3	Unabhängigkeit zwischen Komponenten	$k_i \perp k_j$	(7.75)
4	Abhängigkeit zwischen Symptomen und Komponenten	$P(k s_i) \neq P(k)$	(7.76)
5	Abhängigkeit zwischen Symptomen und Kontextinformationen	$P(s_i c_i, k) \neq P(s_i k)$	(7.77)
mit		$s \subseteq S_\Sigma, k \in K_\Sigma, c \subseteq C_\Sigma$	

Tabelle 7.5.: (Un)Abhängigkeitsforderungen

Das mit Definition 7.4 eingeführte einschichtige neuronale Diagnostik-Modell Π_{neuro} erfüllt diese Anforderungen, mit Ausnahme der Modellierung der bedingten Abhängigkeiten zwischen Kontextinformationen c und Symptomen s (7.77). Gelingt es, das Modell gezielt um diese zu erweitern, wäre die Forderung nach minimaler Komplexität des Lösungsraums erfüllt.

Konstruktive Lernverfahren

Einen Ansatz hierfür bietet die Klasse der „konstruktiven Lernverfahren“. Es gibt eine beträchtliche Zahl publizierter unterschiedlicher Verfahren hierzu. Diese lassen sich im Wesentlichen in

- additive Verfahren,

- subtraktive Verfahren und
- additive-subtraktive Verfahren

einteilen (vgl. [Pre95, S. 39ff] für einen ausführlichen Vergleich).

Additive Verfahren beginnen mit einem kleinen Netz, von dem anzunehmen ist, dass es eine zu geringe Komplexität des Lösungsraums besitzt, um die Lernaufgabe befriedigend zu lösen. Je nach Verfahren werden in Abhängigkeit des Lernfortschritts oder anderer Messgrößen gezielt Neuronen und Verbindungen hinzugefügt. Das Ziel ist hier, eine möglichst kleine oder möglichst komplexitätsreduzierte Netzarchitektur zu finden, die die Lernaufgabe zufriedenstellend lösen kann.

Additive Verfahren

Bekannte Beispiele hierfür, die auch für diese Arbeit betrachtet wurden, sind Gallant [Gal86, S. 652 ff.], der *Tiling*-Algorithmus [MN89, S. 2191 ff.] und dessen Erweiterung durch Nadal [Nad89, S. 55 ff.], [BZ92, S. 392 ff.], der *Upstart*-Algorithmus von Freat [Fre90, S. 198 ff.], der *BINCOR*-Algorithmus von Simon [Sim93], [TBB⁺91], das Verfahren von Zollner et al. [ZSWK92, S. 771 ff.], das auf Sigmoide Aktivierungsfunktionen spezialisierte *DNC* (*dynamic node creation*) von Ash [Ash89, S. 365 ff.] (ebenfalls vorgeschlagen in [BF91, S. 130 ff.]), ein Verfahren mit zwei verdeckten Schichten von Wang, di Massimo, Tham und Morris [WDMTM94, S. 291 ff.], ein auf mehreren spezialisierten Teilnetzen aufbauendes Verfahren nach Waibel [Wai89, S. 39 ff.], *Meiosis* nach Hanson [Han89, S. 533 ff.], *node splitting* nach Wynne-Jones [WJ93, S. 17 ff.] und der weit verbreitete *Cascade Correlation* (*CasCor*) [FL89a, S. 524 ff.], [FL91], der mit den Erweiterungen durch 2CCA [Sjø91], [Sjø92, S. 59 ff.], [Yeu91, S. 228 ff.], *limited fan-in random-wired CasCor* [KS93, S. 79 ff.], *CasEr* und *CasQEF* [LR92, S. 398 ff.], zu den erfolgreichsten und am meisten eingesetzten Verfahren zählt.

Betrachtete additive Verfahren

Subtraktive Verfahren beginnen im Gegensatz zu additiven Verfahren mit einem Netz, das als „zu groß“ angenommen wird. Es wird implizit angenommen, dass ein „Teilnetz“ die Aufgabe hinreichend lösen kann. Das Ziel der Verfahren ist es, damit die Komplexität des Lösungsraums soweit abzusenken, dass sowohl eine Überanpassung vermieden als auch die Aufgabe noch gelöst werden kann. In dieser Hinsicht ähneln Beschneidungs- oder subtraktive Verfahren in ihrer Wirkung regularisierenden Trainingsverfahren [Ree93, S. 740 ff.]. Der Unterschied liegt darin, das letztere nur die *effektive* Zahl an Parametern, erstere die *tatsächliche* Zahl an Parametern reduzieren, wodurch keine Beschränkung in der Parameterwahl (wie z. B. max. Gewichtsstärke, etc.) stattfinden muss [FHZ93, S. 771 ff.].

Subtraktive Verfahren

Bekannte Beispiele hierfür, die auch für diese Arbeit betrachtet wurden, sind Sietsma und Dow [SD88, S. 325 ff.], [SD91, S. 67 ff.], *skeletonization* nach Mozer und Smolensk [MS89, S. 107 ff.] sowie das ähnliche *ALADIN* nach Karayiannis [Kar92, S. 601 ff.], die beide ein *Relevanzmaß* entwickeln, nach dem Knoten entfernt werden. Weitere Beispiele sind Ji,

Betrachtete subtraktive Verfahren

Snapp und Psaltis [JJ90, S. 312 ff.], Sperduti und Starita mit einer Erweiterung des Backpropagation-Verfahrens [SS93, S. 365 ff.], Omidvar und Wilson mit einer stochastischen Methode [OW92, S. 594 ff.], Karnin mit einem speziellen Sensitivitätsmaß [KKB93, S. 187 ff.], *Ace of Bayes* von Thodberg, der auf Bayes'schen A-priori-Annahmen aufbaut und mathematisch hergeleitet wird [KKP91, S. 342 ff.], *optimal brain damage (OBD)* von Le Cun, Denker und Solla, das zwar häufig zitiert, aber dennoch selten eingesetzt wird [LDS89, S. 598 ff.], *optimal brain surgeon (OBS)* von Hassibi und Stork als Weiterentwicklung von OBD, die jedoch sehr aufwändig ist [HS93, S. 164 ff.], ein Algorithmus von Levin, Leen und Moody, der einen Kompromiss zwischen OBS und OBD darstellt [LLM93, S. 35 ff.], *autoprune* von Finno, Hergert und Zimmermann, die spezielle Teststatistiken und mit ihren Verfahren zu den Leistungsfähigsten der Kategorie gehören [FHZ92, S. 980 ff.], [FHZ93, S. 771 ff.].

Additiv-subtraktive Verfahren

Additiv-subtraktive Verfahren verbinden die Ansätze von additiven und subtraktiven Verfahren. Jedoch ist die Leistungsfähigkeit dieser in der Literatur umstritten (vgl. z. B. [Pre95, S. 46]). Die Herausforderung gegenüber rein additiven oder rein subtraktiven Verfahren, die bisher keineswegs einen sehr ausgereiften Eindruck erwecken, liegt zusätzlich in der Balance zwischen den zwei Elementen. Zudem arbeiten die meisten Verfahren lediglich auf Knotenebene und nicht auf der viel feingranulareren Kantenebene.

Betrachtete additiv-subtraktive Verfahren

Bekannte Beispiele hierfür sind Hirose, Yamashita und Hijiya [HYH91, S. 61 ff.], Hagiwara, der einen *badness factor* verwendet [Hag90, S. 625 ff.], das *dynamic node architecture learning (DNAL)* von Bartlett [Bar94, S. 129 ff.], Moody mit einer Kombination von *sequential network construction (SNC)*, *sensitivity based pruning (SBP)* und *optimal brain damage (OBD)* [Moo94, S. 147 ff.], *parsimonious higher order networks (parsiHONs)* von Fahner und Eckmiller [FE94, S. 279 ff.], *divide and conquer* von Romaniuk und Hall [RH93, S. 1105 ff.] und *generate and test* von Nabhan und Zomaya [NZ94, S. 89 ff.].

Weitere Verfahren

Weitere Verfahren: Es gibt noch eine Reihe an weiteren konstruktiven Verfahren, die jedoch streng genommen keine neuronalen Verfahren darstellen, aber dennoch betrachtet wurden. Dazu zählen *Perceptron-Bäume* und ähnliche Verfahren, die Entscheidungsbäume mit spezialisierten neuronalen Netzen verbinden. In den Ansätzen wird mit jedem Blatt des Baumes ein auf die Aufgabe spezialisiertes Netz verbunden [SN90, S. 423 ff.], [Def90, S. 637 ff.], [Bre91, S. 346 ff.]. Weitere Verfahren sind beispielsweise [Fie93, S. 173 ff.], das mit der Betrachtung aller möglichen Architekturen zum Finden eines optimalen Netzes höherer Ordnung arbeitet, [FL89b, S. 57 ff.] mit einem *Simulated Annealing*, um Knoten mit zufälligen Verbindungen und unterschiedlichen Aktivierungsfunktionen zu erzeugen, sowie [BL90, S. 904 ff.], das ein sogenanntes *Orakel* anstelle einer beschränkten Trainingsmenge verwendet, um gezielt gute Trennflächen für Klassifikationsnetze bauen zu können.

Bewertung konstruktiver Lernverfahren

Mit den betrachteten Verfahren fällt auf, dass keines für jede Aufgabe geeignet ist. Jedes Verfahren besitzt bevorzugte und weniger geeignete Problemklassen. Bedauerlicherweise wurden viele Verfahren lediglich an *Spielzeugproblemen* evaluiert und nicht selten nur mit wenigen anderen Verfahren verglichen. So berücksichtigt keines der betrachteten Verfahren die für die Aufgabenstellung dieser Arbeit typische Dualität der Reparaturfalldaten. Auch wenn mit *CasCor* bzw. die jeweiligen erweiterten Verfahren oder auch mit *autoprune* vermutlich bessere Ergebnisse wie mit einem MLP nach Abschnitt 7.3.2 erreicht werden würden, wie eine Untersuchung in [Pre95, S. 116ff] mit ähnlichen Bedingungen nahelegt, kann ein Optimum erst mit der expliziten Berücksichtigung der mit (7.73) bis (7.77) definierten Anforderungen erreicht werden. Damit wird ein speziell auf die Aufgabenstellung zugeschnittenes additives Lernverfahren notwendig, das im Folgenden entwickelt und diskutiert werden soll.

Die Randbedingungen erfordern ein adaptiertes Verfahren

7.4.2. Verfahren zur gezielten Komplexitätserweiterung neuronaler Netze

Ausgangspunkt des Verfahrens ist das einschichtige Diagnostik-Modell Π_{neuro} nach Definition 7.4. Gemäß den Forderungen nach (7.73) bis (7.77) erfüllt dieses Modell lediglich die Forderung (7.77) zur Abhängigkeit zwischen Symptomen s_i und Kontextinformationen c_j nicht. Gemäß dem MLP-Modell wird zunächst eine neue, verdeckte Schicht notwendig. Die Anforderungen dabei sind jedoch, dass im Gegensatz zum MLP-Modell keine zusätzlichen Abhängigkeiten zwischen Symptomen abgebildet werden können. Zur Lösung wird zunächst jedes Eingangsneuron v_{s_i} für ein Symptom s_i in ein Hiddenneuron $v_{s_i}^*$ mit ebenfalls sigmoider Aktivierungsfunktion konvertiert. Die ursprüngliche Gewichtsmatrix W_I^* wird somit zur Ausgabegewichtsmatrix, d.h. $W_O \equiv W_I^*$. Gleichzeitig wird für jedes Hiddenneuron $v_{s_i}^*$ ein neues Eingangsneuron v_{s_i} eingefügt und mit diesem verbunden. Anschließend wird für jede Kontextinformation $c_i \in C_\Sigma$ ein neues Eingangsneuron v_{c_i} erzeugt und jeweils mit jedem verdeckten Neuron $v_{s_i}^*$ verbunden. Damit entsteht das in Abbildung 7.11 dargestellte Netz.

Die spezielle Konstruktionsmethode erfüllt durch

$$w_{s_i s_j^*} = 0 \quad \forall i, j \in \{1, \dots, \#S_\Sigma\} \wedge i \neq j$$

die Forderung (7.73) nach gegenseitig unabhängigen Symptomen $s_i \perp s_j$ und gleichzeitig nach gegenseitig unabhängigen Komponenten $k_i \perp k_j$. Die Forderung (7.74) nach untereinander unabhängigen Kontextinformationen $c_i \perp c_j$ wird ebenfalls erfüllt, da $\{v_{s_i^*}, v_{s_i}, v_{c_j}\} \quad \forall j \in \{1, \dots, \#C_\Sigma\}$ ein einschichtiges Subnetz darstellt und mit der wahrscheinlichkeitstheoretischen Betrachtung nach Abschnitt 7.2.1 die Unabhängigkeit gilt.

Nachweis der Erfüllung der Forderungen

Abbildung 7.11.: Erweitertes neuronales Diagnostik-Modell (die Bias Θ_{k_i} sind nicht dargestellt)

Spezielles Nach-
training zur Wahr-
nung der Interpretierbarkeit

Die meisten additiven Konstruktionsverfahren führen nach dem Schritt der Erweiterung ein Nachtraining des gesamten Netzes durch. Allerdings würde dadurch die Interpretierbarkeit der Gewichte verloren gehen. Es wird darum nicht das gesamte Netz, sondern lediglich die neu hinzugekommene Eingabeschicht W_I per Backpropagation-Algorithmus nachtrainiert. Diese Vorgehensweise macht dennoch eine Modifikation der Verbindungsgewichte W_O der Ausgabeschicht erforderlich. Bedingt durch die sigmoide Aktivierungsfunktion der verdeckten Neuronen $v_{s_i^*}$ kann der ursprüngliche Definitionsbereich von $s_i \in \{0, 1\}$ nur noch für Gewichte $w_{ij} \rightarrow \infty$ erreicht werden, was faktisch ebenfalls nur durch ∞ Trainingsperioden erreicht werden würde. Als Folge wären alle Defektwahrscheinlichkeiten $P(k_i | S, C, \mathcal{H})$ um einen nichtlinearen Faktor ϵ_i reduziert. Durch eine Modifikation der Gewichte und des Bias der Ausgabeschicht mit

$$w_{s_i^* k_j} = \lambda \cdot w_{s_i k_j}^* \quad \text{bzw.} \quad W_O = \lambda \cdot W_I^* \quad \text{und} \quad (7.78)$$

$$\Theta_{k_i} = \lambda \Theta_{k_i}^*,$$

wobei λ ein konstanter Faktor mit $\lambda > 1$, $w_{s_i k_j}^*$ das ursprüngliche Gewicht $w_{s_i k_j}$ und $\Theta_{k_i}^*$ den ursprünglichen Bias darstellt, wird dieser Nachteil kompensiert und die Interpretierbarkeit dennoch erhalten¹⁴.

¹⁴ Die Anwendung des Nachtrainings auf das gesamte Netz hätte ebenfalls zu einer Verstärkung der Gewichte $w_{s_i^* k_j}$ geführt, jedoch die Interpretierbarkeit verwirkt

Interpretation der neuen Eingabeschicht

Der Erhalt der alten Eingabeschicht ermöglicht nun eine Interpretation der neu hinzugekommenen Verbindungsgewichte und Neuronen. Für jedes ursprüngliche Eingangsneuron für ein Symptom s_i ist ein neues, als unabhängig betrachtbares Subnetz G_{s_i} , definiert durch $\{v_{s_i^*}, v_{s_i}, v_{c_j}\} \forall j \in \{1, \dots, \#C_\Sigma\}$, entstanden. Es stellen sich damit drei Fragen:

1. Wie kann das Hiddenneuron $v_{s_i^*}$ interpretiert werden,
2. welche Bedeutung haben die neuen Kontextneuronen v_{c_j} und deren Verbindungsgewichte $w_{c_j s_i^*}$ und
3. warum und wie werden die Abhängigkeiten $s_i \perp c_j$ abgebildet ?

Mit dem Erhalt der alten Verbindungsgewichte bestimmen diese auch mit $w_{s_i^* k_j}$ den Einfluss des Hiddenneurons auf die Defektwahrscheinlichkeit von k_i mit $P(k_i | S^*, \mathcal{H})$. Allerdings kann dieser durch die Gewichtsverstärkung in (7.78) stärker oder mit dem Einfluss des Kontext c_j schwächer ausfallen. Anschaulich ist damit s_j^* ein virtuelles kontextkorrigiertes Symptom. Der Sachverhalt wird deutlicher mit der Betrachtung der neuen Kontextneuronen v_{c_j} und deren Verbindungsgewichte $w_{c_j s_i^*}$.

Virtuelle kontextkorrigierte Symptome

Eine sinnvolle Interpretation wird mit der Berücksichtigung der eingangs erwähnten Dualität der Falldaten möglich. Kontextinformationen c_j sind zuverlässig für jedes Fahrzeug und in jedem Reparaturfall verfügbar. Mit der Wahl der Kodierung „1-aus-m“ (vgl. Abschnitt B.1) ist pro Steuergerät jeweils nur ein Eingangsneuron aktiv (jedes SG kann nur eine Version besitzen). Mit den geringen Auftrittswahrscheinlichkeiten eines Symptoms s_i und einer Komponente k_j existieren für ein Subnetz G_{s_i} sehr viele Fälle, für die gilt

$$s_i = 0 \wedge k_j = 0 \wedge (c_1 \vee c_2 \vee \dots \vee c_n) = 1. \quad (7.79)$$

Hierdurch bildet sich für ein Hiddenneuron $v_{s_i^*}$ zwangsläufig eine AND-Charakteristik aus mit

$$s_i^* \approx s_i \wedge (c_1 \vee c_2 \vee \dots \vee c_n). \quad (7.80)$$

Für den Bias $\Theta_{s_i^*}$ gilt damit $-\Theta_{s_i^*} > \max_{j \in \Lambda}(w_{c_j})$ und zur Erfüllung der Bedingung (7.80) zwangsläufig $w_{s_i s_i^*} > \max_{j \in \Lambda}(w_{c_j})$. Mit der Möglichkeit der isolierten Betrachtung von G_{s_i} als Subnetz gilt auch hier die Gewichtsinterpretation nach Abschnitt 7.2.2 analog zu (7.65) als Chancenverhältnis mit

$$OR(s_i^* | C_{j1/0}, \mathcal{H}) = \frac{Odds(s_i^* = 1 | c_{j1}, \mathcal{H})}{Odds(s_i^* = 1 | c_{j0}, \mathcal{H})} = e^{w_{ij}(s_{c1} - s_{c0})} = e^{w_{c_j s_i^*}}. \quad (7.81)$$

Die hierfür notwendige Annahme, dass sich für die Betrachtung jeweils nur ein Kontext c_j ändert, ist mit der „1-aus-m“-Kodierung inhärent erfüllt. Durch (7.80) gewinnt das Chancenverhältnis jedoch nur an Relevanz bzw. absoluten Einfluss auf s_i^* , wenn das Symptom s_i vorhanden

ist. Damit wird die Bezeichnung von s_i^* als virtuelles kontextkorrigiertes Symptom deutlich. Ist das Symptom s_i vorhanden, ist auch, und nur dann, das virtuelle Symptom s_i^* vorhanden. Dessen Stärke wird jedoch je nach vorhandenem Kontext geschwächt, falls $w_{c_j s_i^*} < 0$ oder gestärkt, wenn $w_{c_j s_i^*} > 0$. Die Fragen nach der Interpretation der Hiddenneuronen $v_{s_i^*}$ sowie der Kontextneuronen v_{c_j} und deren Verbindungsgewichte $w_{c_j s_i^*}$ sind damit beantwortet.

Abhängigkeit zwischen Symptomen und Kontextinformationen

Die letzte Frage nach der Abhängigkeit zwischen einem Symptom s_i und dem Kontext c_j wird implizit durch die Bedingung (7.80) beantwortet. Ohne diese wären Symptome und Kontextinformationen gemäß (7.47) unabhängig. Mit der Bedingung (7.80) kann jedoch $P(s_i|c_i, k) = P(s_i|k)$ aus (7.77) nicht mehr erfüllt sein, da dies so sonst nicht hätte formuliert werden können. Noch deutlicher wird das mit der Umformulierung von (7.80) zu $P(c_i|s_i, k) = P(c_i|k)$, was offensichtlich nicht erfüllt ist. Gleichzeitig jedoch gilt die gegenseitige Unabhängigkeit von Kontextinformationen $c_i \perp c_j$, denn hier lassen sich weder aus den Daten noch aus (7.79), (7.80) oder (7.81) Abhängigkeiten ableiten, so dass (7.47) weiterhin gilt.

Bestimmung der Parameter unter Sparse Data

Bisher wurde die Bestimmung des Parameters λ wie auch die Initialisierung der neuen Gewichte $w_{s_i s_i^*}$ und $w_{c_j s_i^*}$ nicht näher betrachtet. Besonders kritisch werden diese jedoch unter der Voraussetzung besonders geringer Datenmengen bzw. besonders wenigen Fällen pro Modellparameter. Denn dann ist die Bestimmung dieser Parameter nicht mehr eindeutig durch den Lernalgorithmus möglich, was zu unerwarteten Diagnostik-Hypothesen bzw. unerwartet modellierten Abhängigkeiten führen kann. Entgegengewirkt wird dem zum einen durch die Wiederverwendung des einschichtigen neuronalen Modells und zum anderen durch die Einführung zusätzlicher Annahmen $\mathcal{H}$.

Notwendige Annahmen $\mathcal{H}$

Die erste wichtige Annahme ist dabei, dass davon ausgegangen wird, dass das einschichtige Modell bereits sehr gut die Diagnostik-Aufgabe löst. Die zweite Annahme ist, dass kongruent dazu zuerst von einem geringen bis keinem Einfluss eines konkreten Kontextneurons c_j ausgegangen wird. Dieser muss sich erst anhand der Trainingsdaten zeigen. Als Konsequenz muss die ursprüngliche Netzfunktion nach der Erweiterung erhalten bleiben bzw. näherungsweise identisch sein. Damit sind die Randbedingungen zur Bestimmung des Parameters λ und der Initialisierung des Bias $\Theta_{s_i^*}$ und des Verbindungsgewichts $w_{s_i s_i^*}$ vorgegeben.

Der Parameter λ wird mit $\lambda = 2$ bestimmt. Damit erfüllt ein Ausgabe-
wert von $v_{s_i^*} = 0,5$ die ursprüngliche Netzfunktion für das Vorhandensein des Symptoms s_i . Gleichzeitig ist das der Punkt der größten Sensitivität des Neurons (vgl. Abbildung 7.4), wodurch ferner ein symmetrischer Einfluss des Kontexts erreicht wird. Mit der vereinfachten Annahme, dass nur

ein Symptom s_i vorhanden ist, kann die Berechnung des Netzausgangs (7.70) für ein Neuron v_{k_j} vereinfacht werden zu

$$v_{k_j} = \frac{1}{1 + e^{-\left[\lambda w_{s_i k_j} \frac{1}{1 + e^{-[s_i w_{s_i s_i^*} + \Theta_{s_i^*}]} + \lambda \Theta_{k_j} \right]}}. \quad (7.82)$$

Die Initialisierungsparameter müssen damit folgendes Gleichungssystem erfüllen:

$$s_i = 0 : v_{k_j} = \frac{1}{1 + e^{-\left[\lambda w_{s_i k_j} \frac{1}{1 + e^{-\Theta_{s_i^*}}} + \lambda \Theta_{k_j} \right]}} \stackrel{!}{=} \frac{1}{1 + e^{-\Theta_{k_j}}} \quad (7.83)$$

$$s_i = 1 : v_{k_j} = \frac{1}{1 + e^{-\left[\lambda w_{s_i k_j} \frac{1}{1 + e^{-[w_{s_i s_i^*} + \Theta_{s_i^*}]} + \lambda \Theta_{k_j} \right]}} \stackrel{!}{=} \quad (7.84)$$

$$\frac{1}{1 + e^{-[w_{s_i k_j} + \Theta_{k_j}]}} \quad (7.85)$$

Aus (7.83) folgt durch Umformung

$$\begin{aligned} e^{-\left[\lambda w_{s_i k_j} \frac{1}{1 + e^{-\Theta_{s_i^*}}} + \lambda \Theta_{k_j} \right]} &\stackrel{!}{=} e^{-\Theta_{k_j}} \\ \Theta_{s_i^*} &= -\ln \left(\frac{\lambda}{1 - \lambda} \frac{w_{s_i k_j}}{\Theta_{k_j}} - 1 \right) \stackrel{\lambda=2}{=} -\ln \left(-2 \frac{w_{s_i k_j}}{\Theta_{k_j}} - 1 \right) \end{aligned} \quad (7.86)$$

und durch Umformung von (7.84) und Einsetzen von (7.86) weiter

$$\begin{aligned} e^{-\left[\lambda w_{s_i k_j} \frac{1}{1 + e^{-w_{s_i s_i^*} \left(\frac{\lambda}{\lambda - 1} \frac{w_{s_i k_j}}{\Theta_{k_j}} - 1 \right)} + \lambda \Theta_{k_j} \right]} &\stackrel{!}{=} e^{-(w_{s_i k_j} + \Theta_{k_i})} \\ w_{s_i s_i^*} &= \ln \left(\frac{\frac{\lambda w_{s_i k_j}}{(1 - \lambda) \Theta_{k_j}} - 1}{\Theta_{s_i^*}} \right). \end{aligned} \quad (7.87)$$

Mit dieser Parameterbestimmung behält das erweiterte Netz die ursprüngliche Funktion bezüglich der Symptome s_i . Die Initialisierung der neuen Kontextgewichte $w_{c_j s_i^*}$ erfolgt unter der zweiten Annahme mit sehr geringen Zufallswerten mit $w_{c_j s_i^*} \in [-0,01, 0,01]$. Zufallswerte zur Initialisierung sind aufgrund der Konvergenzeigenschaften des Lernalgo-

Ursprüngliche Funktion bleibt erhalten

rithmus notwendig, gleichzeitig ist der Einfluss mit dem Wertebereich äußerst gering.

Nachtraining der neuen Eingabeschicht

Nach der Bestimmung der Parameter und der Initialisierung der neuen Verbindungsschicht realisiert das Netz nach wie vor die identische Funktion wie das einschichtige Modell, so dass Kontextinformationen entsprechend (noch) keinen Einfluss haben. Mit dem Ziel, dass nur solche Kontextinformationen c_j an Einfluss gewinnen, d. h. $w_{c_j s_i^*} > 0$, muss ein Nachtraining der neuen Schicht durchgeführt werden. Verwendet wird hierfür dieselbe Fallbasis R_{train} wie bereits für das einschichtige Modell. Als Trainingsalgorithmus kommt Backpropagation-Momentum zum Einsatz.

Nachfolgend wird aufgezeigt, wann und wie durch das Nachtraining eine Anpassung einzelner Kontextverbindungsgewichte $w_{c_j s_i^*}$ stattfindet. Hierfür wird nochmals die Regel zur Gewichtsmodifikation nach (7.34) dargestellt¹⁵:

$$\begin{aligned} \Delta w_{ij} &= \Delta w_{ij,p} = -\eta \cdot g_{ij,p} = -\eta \cdot y_{i,p} \delta_{j,p} \\ &= \begin{cases} -\eta y_{i,p} y_{j,p}^o (1 - y_{j,p}^o) (t_{j,p} - y_{j,p}^o) & , \text{ falls } v_j \text{ Ausgabeneuron} \\ -\eta y_{i,p} y_{j,p}^h (1 - y_{j,p}^h) \sum_k \delta_{k,p}^o w_{jk} & , \text{ falls } v_j \text{ verdecktes Neuron} \end{cases} \end{aligned} \quad (7.34)$$

Interpretation des
Nachtrainings

Für ein konkretes Trainingsbeispiel p der Trainingsmenge, hier äquivalent mit einem Reparaturfall $R \in R_{train}$, findet eine Gewichtsmodifikation von $w_{c_j s_i^*}$ gemäß (7.34) nur dann statt, wenn zum einen die entsprechende Kontextinformation c_j vorhanden ist, d. h. $y_{c_j,p} = 1$, und zum anderen ein Fehlerterm mit $\delta_{s_i^*,p} \neq 0$ existiert. Letzteres ist dann erfüllt, wenn gemäß der Backpropagation-Regel ein rückgeführter Fehlerterm der Ausgangsneuronen v_{k_i} existiert, d. h.

$$\delta_{s_i^*,p} = \sum_{k \in K_\Sigma} \delta_{k,p}^o w_{s_i^* k} \stackrel{!}{\neq} 0. \quad (7.88)$$

Das ist anschaulich formuliert immer dann der Fall, wenn das Modell eine für den Beispielfall p fehlerhafte Diagnose-Hypothese aufgestellt hat. Je nachdem, ob der Fehlerterm $\delta_{s_i^*,p}$ positiv ist, d. h. eine unge-rechtfertigte Diagnose-Hypothese aufgestellt wurde oder negativ ist, d. h. eine Diagnose-Hypothese hätte aufgestellt werden müssen, findet eine Adaption all derjenigen Verbindungsgewichte $w_{c_j s_i^*}$ statt, für die eine Kontextinformation c_j vorhanden ist. Je nach Vorzeichen des Fehlerterms $\delta_{s_i^*,p}$ ist auch die Adaption $\Delta w_{c_j s_i^*}$ positiv oder negativ.

Mit der detaillierten Erörterung wird deutlich, warum nur Kontextinformationen, die eine im Mittel optimierte Diagnostik ermöglichen,

¹⁵ Die Indizierung ist mit den eingeführten Indizes für Symptome und Komponenten teilweise verschieden.

an Relevanz gewinnen bzw. andere nicht. Sie bestätigt weiterhin die Interpretation der Kontext-Verbindungsgewichte als Odds-Ratio und die Interpretation der verdeckten Neuronen $v_{s_i^*}$ als virtuelle kontextkorrigierte Symptome.

Zusammenfassung der Komplexitätserweiterung

Mit der Feststellung der geringen Eignung von Feedforward-Netzen (MLP) wurden zunächst verschiedene existierende Verfahren zur Komplexitätserweiterung analysiert und bewertet. Dabei erwies sich zwar die Klasse der additiven Methoden als vielversprechend, jedoch war keines der Verfahren zur Verwendung der in diesem Projekt verfügbaren Daten mit den bereits beschriebenen speziellen Eigenschaften geeignet. Es wurde daher eine spezialisierte Methode zur gezielten Komplexitätserweiterung des einschichtigen Diagnostik-Modells entwickelt, welche nur dort zusätzliche Merkmals-Abhängigkeiten erlaubt, wo sie gefordert und erwünscht sind. Hierzu wurden die Eingabeschicht des vorhandenen, bereits trainierten Modells Π_{neuro} in eine verdeckte Schicht und die Symptomen s_i zu virtuellen kontextkorrigierten Symptomen s_i^* konvertiert. Gleichzeitig wurde eine neue Eingabeschicht mit denselben (ursprünglichen) Symptomen s_i , jedoch mit zusätzlichen Kontextinformationen c_j , eingefügt. Die Verbindungsgewichte wurden dabei in einer Weise neu berechnet, dass die ursprüngliche Funktion des Modells vollständig erhalten blieb. Durch ein Nachtraining der neuen Eingabeschicht wird sukzessive gelernt, welche Kontextinformationen zu einer Verbesserung der Gesamtleistungsfähigkeit führen.

Das Vorgehen ermöglicht damit eine Interpretierbarkeit des neuen Modells. Eine solche wurde speziell für die neuen Kontextverbindungen vorgestellt und diskutiert. Gleichzeitig ermöglicht die Vorgehensweise, bereits bei wenigen Falldaten und unter Sparse Data funktionsfähige kontextsensitive Modelle zu erzeugen. Die Evaluierung in Abschnitt 9.3.6 hat gezeigt, dass mit dem Verfahren nochmals eine Steigerung der Evaluierungsquote q_n um 3,3 % auf 96,4 % d. h. bis knapp an das Maximum der Evaluierungsquote q_n erreicht wurde. Die größte Steigerung konnte für die Maßnahmenklasse ET mit +15,7 % erreicht werden. Der Vorteil dürfte mit steigender Diversifizierung der in der Fallbasis enthaltenen Fahrzeugmodelle noch deutlicher ausfallen. Damit kann letztlich die Zahl der von einem Diagnostik-Modell abdeckbaren Fahrzeugvarianten deutlich gesteigert werden.

7.5. Adaptivität und Verlernen von altem Wissen

Veränderungen im
Produktlebenszyklus

Ein über Jahre eingesetztes lernendes System häuft große Mengen an Daten an. Um so „wissender“ und zuverlässiger (für bekannte Fälle) wird das System, aber auch umso unflexibler gegenüber Veränderungen. Solche treten im Laufe des Lebenszyklus einer Modellreihe häufiger auf. Dazu zählen im Anfangsstadium vor allem spezielle Fehlerabstellmaßnahmen wie Softwareupdates oder Hardwareänderungen zur Beseitigung von bekannt gewordenen Produktfehlern. Später kommen Maßnahmen zur Qualitätssteigerung, der Wechsel von Bauteillieferanten oder Veränderungen/Optimierungen der Produktion etc. hinzu. Später treten fahrzeugmodelltypische Alters- bzw. Verschleißerscheinungen auf, die sich mit zunehmenden Alter nochmals verändern. Ein leistungsfähiges Diagnostik-System ist in der Lage, auch die Diagnostik an solche Veränderungen anzupassen. Besonders relevant sind dabei solche Veränderungen, die bei gleichem oder ähnlichem Symptombild S zu veränderten Reparaturmaßnahmenwahrscheinlichkeiten führen, d. h.

$$P(k_i|S, t \leq t_v) \neq P(k_i|S, t > t_v), \quad (7.89)$$

Ohne Anpassung
längere Fehlersuch-
zeiten und höhere
Fehltauschkosten

mit t_v als Zeitpunkt einer Veränderung. Würde hier keine Anpassung erfolgen, würden als defekt vermutete Komponenten k_i in der falschen Reihenfolge und mit dem falschen Vertrauenswert v vorgeschlagen werden. Dies würde im Mittel zu längeren Fehlersuchzeiten und zu einer erhöhten Fehltauschrate führen, da häufiger zuerst die „falsche“ Komponente überprüft oder getauscht werden würde. Letztlich würden so höhere und unnötige Kosten entstehen.

Anpassung durch
Verlernen alter
Reparaturfälle

Eine Lösung hierfür ist das „Verlernen“ älterer bzw. eine höhere Priorisierung jüngerer Reparaturfälle. Hierfür sind grundsätzlich zwei Methoden möglich. Zum einen eine Modifikation des Lernalgorithmus selbst und zum anderen eine Modifikation der Vorgehensweise des Lernens.

Eine Modifikation des Lernalgorithmus würde beispielsweise darin bestehen, die Lernrate η für ältere Fälle herabzusetzen und für neuere Fälle zu erhöhen. Alternativ könnten auch neuere Fälle in der Trainingsmenge um so häufiger dupliziert werden, je jünger sie sind. Erstere Variante besitzt jedoch den Nachteil, dass eine Veränderung der Lernparameter das Konvergenzverhalten beeinflusst. Die Evaluierung der Auswirkung verschiedener Parametersets (vgl. Abbildung C.2) zeigt eine große Auswirkung in der Leistungsfähigkeit bereits bei kleinen Veränderungen. Letztere Variante könnte schnell zu Performanzproblemen und Übertrainingseffekten führen, da Reparaturfälle nicht „geteilt“ werden können und folglich neue Fälle entsprechend häufig (zu häufig) dupliziert werden müssten.

Im Rahmen dieser Arbeit wurde darum eine spezielle Vorgehensweise des Lernens entwickelt. Zunächst wird das Diagnostik-Modell nach den in den Abschnitten 7.2 bis 7.1.4 beschriebenen Verfahren auf Basis aller verfügbaren Trainingsfalldaten R_{train} erzeugt und trainiert. Das Training erfolgt jedoch mit verminderter Trainingsepochenzahl. Dadurch wird erreicht, dass zu Beginn alle verfügbaren und relevanten Symptome S wie auch Verbindungen w_{ij} im Modell vorhanden sind. Nicht vorhandene Symptome wie auch Verbindungen würden später nicht mehr hinzugefügt werden. Gleichzeitig findet durch das „Vortraining“ eine Initialisierung mit über alle Fälle gewichteten Werten statt. Anschließend wird eine disjunkte Partitionierung der Trainingsdaten R_{train} in nach Datum aufsteigend sortierten Blöcken $R_{t_1}, \dots, R_{t_n} \in R_{train}$ vorgenommen. Das Modell wird dann beginnend mit dem ältesten Block bis hin zum neuesten Block mit angepasster Trainingsepochenzahl trainiert. Nach dem Training mit dem letzten Block ist das Verfahren abgeschlossen. Die Vorgehensweise ist in Abbildung 7.12 skizziert. Für die Partitionierung der Trainingsdaten wie auch die Wahl der Trainingsepochen gilt es folgende Aspekte zu beachten:

Modifiziertes Lernen

Blockgröße Die Blockgröße sollte derart gewählt werden, dass sie zum einen nicht so klein ist, dass Symptom- und Maßnahmenhäufigkeiten $P(s_i)$ bzw. $P(k_i)$ nicht mehr hinreichend genau bestimmt werden können. Dies würde zu Fehlanpassungen der Verbindungsgewichte e_{ij} führen. Zum anderen sollte die Blockgröße nicht so groß gewählt werden, dass ein Verlerneffekt nicht mehr entsteht. Ist aufgrund zu weniger aber zeitlich gestreckter Falldaten keine disjunkte Partitionierung möglich, sollte besser eine überlappende als eine zu kleine Partitionierung vorgenommen werden.

Trainingsepochen Die Gesamtzahl der Trainingsschritte t_s sollte identisch mit dem aus der Evaluierung in Abschnitt 9.3.4 festgestellten optimalen Wert $t_{s_{opt}}$ sein. Die Trainingsschritte berechnen sich aus der Epochenzahl t_e , multipliziert mit der Trainingsfallzahl $t_s = t_e \cdot \#R_{train}$. Damit ist für die Gesamtschrittzahl gefordert

Parameterbestimmung

$$t_s = t_{vor} \cdot \#R_{train} + \sum_{i=1}^n t_i \cdot \#R_{t_i} \stackrel{!}{\approx} t_{s_{opt}}. \quad (7.90)$$

Gute Ergebnisse wurden dabei mit einer Aufteilung gemäß $t_i = t_j$ für $i, j \in \{1..n\}$ (d. h. gleiche Blockgröße und Epochenzahl) und $t_{vor} = 3$ Epochen erzielt.

Durch das sukzessive Training mit immer jüngeren Blöcken werden die einzelnen Verbindungen an die immer neueren Falldaten adaptiert, das Netz verlernt also schrittweise die alten Werte. Diese Adaption findet jedoch nicht sprunghaft, sondern langsam und iterativ statt. Das heißt, das Wissen bzw. die Gewichtswerte aus älteren Falldaten (Blöcken) gehen nicht vollständig verloren. Weiterhin findet eine Anpassung nur dann

Anpassungsverhalten

Abbildung 7.12.: Schematische Darstellung der Vorgehensweise zum Verlernen alter Reparaturfälle

statt, wenn eine solche notwendig ist, also das jeweilige Ausgangsneuron k_j einen Fehlerterm $\delta_j^o \neq 0$ erzeugt und nur dann eine Gewichtsänderung zum verursachenden Symptom s_i mit $\Delta w_{ij} = -\eta \cdot y_{s_i} \delta_j^o$ hervorruft (vgl. Abschnitt 7.1.4).

Das Adaptionsverhalten soll anhand eines Beispiels demonstriert werden. Hierzu wird angenommen, dass zu einem fiktiven Zeitpunkt $t_v = 30.04.2006$ eine Fehlerabstellmaßnahme bezüglich des auffällig häufig defekten Luftmassenmessers durchgeführt wurde, die entsprechend eine Veränderung der Reparaturmaßnahmenwahrscheinlichkeiten nach (7.89) hervorruft. Als Syndrom S wird das Symptom *DTC16485 - Luftmassenmesser-G70 unplausibles Signal* im Motorsteuergerät, als Maßnahme k_1 der Tausch des Luftmassenmessers, als Maßnahme k_2 der Tausch des Druckrohrs und mit k_x weitere verschiedene Maßnahmen angenommen. Die gewählten Parameter sind in Tabelle 7.6 dargestellt. Ausgehend von einer Falldatengröße von 56.000 Fällen sind die bedingten

Beispiel
Luftmassenmesser

Anz. Fälle mit S	Datumsbereich	$P(k_1 S,t)$ ($\#k_1$)	$P(k_2 S,t)$ ($\#k_2$)	$P(k_x S,t)$ ($\#k_x$)
40	01.01.2006 - 30.04.2006	80% (32)	13% (5)	8% (3)
32	01.05.2006 - 31.08.2006	75% (24)	16% (5)	9% (3)
24	01.09.2006 - 31.12.2006	67% (16)	21% (5)	13% (3)
16	01.01.2007 - 30.04.2007	50% (8)	31% (5)	19% (3)
12	01.05.2007 - 31.08.2007	33% (4)	42% (5)	25% (3)
11	01.09.2007 - 31.12.2007	27% (3)	45% (5)	27% (3)

Tabelle 7.6.: Beispieldaten zur Untersuchung des Adaptionsverhaltens

Wahrscheinlichkeiten für die Maßnahme k_1 zunächst hoch, bis ab k_v Fahrzeuge mit einer älteren Version des Luftmassenmessers seltener werden und $P(k_1|S)$ wie auch die absolute Häufigkeit $\#k_1$ entsprechend sinkt. Die Defektwahrscheinlichkeit des Druckrohrs $P(k_2|S)$ bleibt dagegen unverändert gleich.

In Abbildung 7.13 sind die Ergebnisse jeweils mit und ohne Verlernen dargestellt. Die Blockgröße war dabei immer 4 Monate und die Trainingsepochenanzahl nach (7.90) bestimmt. Es ist deutlich zu sehen, dass zwar auch ohne Verlernen eine langsame Anpassung an die neuen Wahrscheinlichkeiten $P(k_i|S,t)$ stattfindet, diese aber einen deutlich längeren Zeitraum benötigt. Die Bedeutung der Anpassung auf die Reparaturkosten kann aus der Reihenfolge der Vertrauenswerte abgeleitet werden. Für das Modell ohne Verlernen würde sehr lange der Luftmassenmesser als defekte Komponente vorgeschlagen werden, was zum einen zu einer zeit-aufwändigen Überprüfung dessen führt, wie auch die Wahrscheinlichkeit für einen Fehltausch erhöht.

Das Beispiel zeigt damit die Notwendigkeit wie auch die Wirksamkeit des Verfahrens. Die Relevanz für reale Reparaturfalldaten wurde durch die Evaluierung in Abschnitt 9.3.7 gezeigt. Selbst in der relativ kurzen

Ergebnisse und
Diskussion

Abbildung 7.13.: Beispiel des Adaptionungsverhaltens nach einer Fehlerabstellmaßnahme, mit und ohne Verlernen

Zeitspanne dieser Daten wurde durch Verfahren eine Leistungssteigerung um 1,4 % - 3,4 % erreicht.

7.6. Beschleunigung durch Merkmals-Reduktion

Mit zunehmender Anzahl von Falldatensätzen werden Lernverfahren mit mehreren Trainingsepochen ressourcenaufwändig hinsichtlich Rechen- und Speicherbedarf. Die hier beschriebenen Diagnostik-Modelle konnten zwar im Rahmen der Evaluierung noch mit handelsüblichen Rechnern trainiert werden, für den Praxiseinsatz und die zu erwartenden größeren Fallzahlen würden so jedoch Rechencluster notwendig werden. Eine Lösung hierfür ist die vorhergehende Reduktion von irrelevanten Merkmalen. Wie die Evaluation der Pruning-Verfahren zeigt, existieren pro Komponente k_i weit weniger als 100 relevante Merkmale ($\#S_i < 100$) von gesamt ca. 7.100 verfügbaren Merkmalen. Eine Merkmals-Reduktion an sich verringert dabei linear die Anzahl an Verbindungsgewichten der neuronalen Netze und damit des gesamten Ressourcenbedarfs. Im Rahmen der Dissertation wurde hierfür ein dreistufiges, auf die Charakteristik der Daten angepasstes Merkmal-Reduktions-Verfahren entwickelt, das im Folgenden vorgestellt wird.

Wenige relevante Merkmale pro Komponente

Um die positiven Eigenschaften der neuronalen Netze wie Mustererkennung und Generalisierung zu erhalten, sollten in einem Netz möglichst diejenigen Symptome und Komponenten zusammengefasst werden, zwischen denen Abhängigkeiten existiert. Hierfür bietet die Statistik eine Reihe von Abhängigkeitstests. Für die hier vorhandenen nominal dichotomen Merkmale S , K sind Signifikanztests geeignet. Mögliche Methoden hierfür, die auch in dieser Arbeit untersucht wurden, sind der χ^2 -Test (Chi-Quadrat-Test) mit

Signifikanztests zur Abhängigkeitsbestimmung

$$\chi^2 = \sum_{j=1}^u \sum_{k=1}^r \frac{(n_{jk} - n_{jk}^*)^2}{n_{jk}^*} \quad (7.91)$$

χ^2 = Prüfgröße für den χ^2 -Test

n_{jk} = Absolute Randhäufigkeit

n_{jk}^* = Erwartete Randhäufigkeit bei Unabhängigkeit (H_0)

u, r = Merkmaldimension (hier 2)

und der exakte Test nach R. A. Fisher [Sac02, S. 479] (Der exakte Test nach R. A. Fisher liefert dabei für extrem einseitige Verteilungen mit geringen Kategorie-Häufigkeiten¹⁶ bessere Ergebnisse¹⁷). In Signifikanztests wird zunächst angenommen, zwei Merkmale seien unabhängig (genannt Nullhypothese H_0). Mit dieser Annahme wird dann eine Prüfgröße χ^2 berechnet, die ein Maß für die Abweichung der realen, von der für die Nullhypothese H_0 erwarteten Verteilung darstellt. In (7.91) wird diese durch die normierte quadratische Abweichung aller absoluten Randhäufigkeiten n_{jk} zu den erwarteten Randhäufigkeiten n_{jk}^* berechnet. Die resultierende Prüfgröße χ^2 kann dann in eine Irrtumswahrscheinlichkeit p (auch p-Wert genannt) umgerechnet werden, die angibt, wie wahrscheinlich ein statistischer Zusammenhang zwischen den Merkmalen, hier einem Symptom s_i und einer Komponente k_i , ist. Ein p-Wert von 1 bedeutet dabei eine 100%-ige Irrtumswahrscheinlichkeit für den Zusammenhang (H_0 gilt als erfüllt; es gibt also keinen Zusammenhang) und ein p-Wert von 0 bedeutet eine 0%-ige Irrtumswahrscheinlichkeit (H_0 wird verworfen; es gibt einen eindeutigen Zusammenhang).

Im ersten Schritt des Merkmals-Reduktions-Verfahrens wird zunächst sukzessiv für alle Symptome $s_i \in S_\Sigma$ und Komponenten $k_j \in K_\Sigma$ die

¹⁶ Unter einer Kategorie wird in der Nomenklatur der Unabhängigkeitstests eine Merkmalskombination verstanden. Im konkreten Fall wäre eine Kategorie z. B. die Kombination von Symptom s_i = vorhanden und Komponente k_j = defekt.

¹⁷ Der χ^2 -Test wird für zu geringe Häufigkeiten einer Kategorie ungenau. Hier hat der exakte Test nach Fisher Vorteile, ist allerdings aufwändiger in der Berechnung.

Schritt 1: Signifikanz (repräsentiert durch den p-Wert) berechnet, wodurch sich die Signifikanzmatrix **S** ergibt mit

$$\mathbf{S} = \begin{bmatrix} p_{k_1 s_1} & \cdots & p_{k_1 s_i} \\ \vdots & \ddots & \vdots \\ p_{k_j s_1} & \cdots & p_{k_j s_i} \end{bmatrix}, \quad (7.92)$$

wobei die Elemente $p_{k_1 s_1}$ jeweils den p-Wert für die Signifikanz zwischen der Komponente k_j und dem Symptom s_i darstellen. Der p-Wert selbst wird dabei mit der Näherungsformel

$$p = \frac{1}{2} \cdot 10^{-\frac{\chi^2}{3.84}} \quad (7.93)$$

aus der mit (7.91) berechneten Prüfgröße χ^2 bestimmt.

Schritt 2: In Schritt 2 wird die Signifikanzmatrix **S** in eine Distanzmatrix **D** mit Distanzmatrix

$$D = \begin{bmatrix} d_{k_1 s_1} & \cdots & d_{k_1 s_i} \\ \vdots & \ddots & \vdots \\ d_{k_j s_1} & \cdots & d_{k_j s_i} \end{bmatrix} \quad (7.94)$$

überführt. Es wird dabei für jede Komponente und jedes Symptom berechnet, welche Distanz $d_{k_j s_i}$ ein Symptom s_i zu einer Komponente k_i besitzt. Hierfür werden zunächst die einzelnen p-Werte $p_{k_1 s_1} \in \mathbf{S}$ zwischen Komponenten und Symptomen als bipartiter, ungerichteter, gewichteter Graph G mit

$$G = (V, E) \quad (7.95)$$

dargestellt, wobei sich für die Knotenmenge $V = V_S \cup V_K$ eine disjunkte/echte Partitionierung in Komponenten-knoten V_K und Symptom-knoten V_S ergibt. Die Kanten E werden mit

$$E \subseteq V_K \times V_S \quad (7.96)$$

jeweils mit den Kantengewichten $p_{s,k}$ aus der Signifikanzmatrix **S** für Kanten $e_{s,k} \in E$ definiert. Der kürzeste Weg zwischen der Komponente k_i und dem Symptom s_j ist nun gegeben als Kantenmenge

$$E_{K_i, S_j} = \bigcup_{x=1..n} \left\{ e_x \in E \mid \begin{array}{l} e_1 = (k_i, s_1), e_2, \dots, e_{n-1}, e_n \\ = (k_n, s_j) \end{array} \right\}. \quad (7.97)$$

Damit ergibt sich die Summe der Kantengewichte für den kürzesten Weg zwischen k_i und s_j mit

$$d_{k_i s_j} = \sum_{e_{k,s} \in E_{K_i, S_j}} p_{k,s}. \quad (7.98)$$

In Abbildung 7.14 ist dies beispielhaft für den Pfad zwischen k_1 und s_8 dargestellt. Die zugehörige Distanz berechnet sich mit $d_{k_1 s_8} = p_{k_1 s_3} + p_{s_3 k_2} + p_{k_2 s_7} + p_{s_7 k_3} + p_{k_3 s_8}$.

Abbildung 7.14.: Beispielhaft dargestellter Distanzgraph G und markierter Pfad zwischen s_8 und k_1

Die Notwendigkeit, Wege über andere Komponentenknoten hinweg zu wählen, ergibt sich aus der Verteilung der Falldaten. Typische Auftretenswahrscheinlichkeiten für ein beliebiges Symptom s_i unter der Annahme, dass ein Fehlerfall F vorliegt, liegen im Bereich von $0,02\% \leq P(s_x|F) \leq 0,2\%$. Dadurch können sehr gut positive Korrelationen, jedoch keine negativen Korrelationen festgestellt werden. Um dies zu zeigen, wird zunächst die Definition des χ^2 -Test nach (7.91) für den hier konkret vorliegenden Fall mit zwei nominal dichotomen Merkmalen (Symptom vorhanden (h_{x1}) / nicht vorhanden (h_{x2}); Komponente defekt (h_{1x}) / nicht defekt (h_{2x})) in

Herausforderung der
Negativ-Korrelationen

$$\chi^2 = \frac{n \cdot (h_{11}h_{22} - h_{12}h_{21})^2}{(h_{11} + h_{12}) \cdot (h_{21} + h_{22}) \cdot (h_{11} + h_{21}) \cdot (h_{12} + h_{22})} \quad (7.99)$$

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \text{Häufigkeitstabelle} \quad (7.100)$$

überführt. Dieser Spezialfall wird auch χ^2 -Vierfeldertest genannt. Die χ^2 -Prüfgröße ist jedoch für einen anschaulichen Vergleich ungeeignet und wird darum mit der Beziehung

$$\chi^2 = n \cdot \Phi^2 \quad (7.101)$$

in den Phi-Koeffizient

Phi-Koeffizient

$$\Phi = \frac{h_{11}h_{22} - h_{12}h_{21}}{\sqrt{(h_{11} + h_{12}) \cdot (h_{21} + h_{22}) \cdot (h_{11} + h_{21}) \cdot (h_{12} + h_{22})}} \quad (7.102)$$

Minimum der
 Φ -Korrelation

überführt. Der Phi-Koeffizient stellt im Gegensatz zur χ^2 -Prüfgröße ein normiertes Zusammenhangsmaß (auch Korrelation genannt) mit dem Wertebereich von $-1 \leq \Phi \leq 1$ dar. Ein $\Phi > 0$ bedeutet dabei eine positive Korrelation und ein $\Phi < 0$ eine negative Korrelation. Jetzt wird ersichtlich, dass für ein Maximum bzw. ein Minimum der Korrelation die Felder der Hauptdiagonalen oder Nebendiagonalen von $\mathbf{H}$ gleich null sein müssen. Für einen positiven Zusammenhang kann dies leicht erfüllt werden, da hier die Felder h_{12} und h_{21} null sein müssen. Darüber hinaus bewirkt der durch die Falldaten bedingt große Wert von h_{22} , dass diese beiden Felder weniger ins Gewicht fallen. Für eine negative Korrelation müssen jedoch die Felder h_{11} und h_{22} null oder nahe null sein. Für h_{22} ist dies nicht möglich. Eine Extremwertuntersuchung von (7.102) ergibt, dass die maximal negative Korrelation für die Bedingung $h_{12} = h_{21}$ erreicht wird. Damit kann das Minimum für Φ mit

$$\Phi_{\min} = -\frac{h_{12}}{h_{12} + h_{22}} \quad \text{für } h_{12} = h_{21} \text{ und } h_{11} = 0 \quad (7.103)$$

bestimmt werden. Für die vorhergehend bereits erwähnten typischen Auftretswahrscheinlichkeiten $P(s_x|F)$ kann der Wertebereich von $\Phi_{\min}$ auf $10^{-4} \leq \Phi_{\min} \leq 10^{-2}$ bestimmt werden. Damit wird deutlich, warum negative Korrelationen nur schwer bzw. nicht erkannt werden können. Das zeigt auch eine beispielhafte Untersuchung der Korrelationswerte für Ersatzteilrumpfnummern (ETRN) in Abbildung 7.15. Negative Korrelationen sind nicht nur sehr selten, sondern auch nur bis zu einer

Abbildung 7.15.: Verteilung (linke Achse, blau) und akkumulierte Häufigkeit (rechte Achse, rot) der Phi-Korrelationswerte für Ersatzteilrumpfnummern (ETRN) mit einer Klassenbreite von 0,01. Beide Achsen beziehen sich auf die 18.211.500 möglichen Symptom-Komponenten-Kombinationen $\Phi_{s_i k_j}$ (für ETRN)

Korrelation von $|\Phi| < 0,01$ vorhanden. Die wenigen größeren negativen Korrelationen sind auf Fehlerabstellmaßnahmen wie z. B. das Anbringen eines Hinweisschildes zurückzuführen. Unter der Annahme, dass ein solcher Negativzusammenhang für eine andere Komponente einen Positivzusammenhang darstellt, ist die Wegbildung über andere Komponenten hinweg eine Lösung.

Die erzeugte Distanzmatrix $\mathbf{D}$, definiert in (7.94), dient nun als Grundlage für die Merkmals-Auswahl in Schritt 3. Alle für eine Komponente k_j vorhandenen Symptom-Distanzen $d_{s_i k_j}$ sind im j -ten Zeilenvektor der Distanzmatrix $\mathbf{D}$ definiert. Würden alle Symptome ausgewählt, würde keine Reduktion stattfinden. Daher wird eine Maximaldistanz d_{th} definiert. Damit kann die für eine Komponente k_j zu betrachtende Symptommenge S_{k_j} bestimmt werden mit

$$S_{k_j} = \bigcup_{s \in S_\Sigma} \{d_{k_j s} \leq d_{th}, d_{k_j s} \in \mathbf{D}\}. \quad (7.104)$$

Es werden folglich all diejenigen Symptome $s_i \in S_\Sigma$ der Symptommenge S_{k_j} hinzugefügt, deren Distanz unterhalb der Maximaldistanz d_{th} liegt. Die Maximaldistanz d_{th} als alleiniges Auswahlkriterium führt für Datenbasen mit vielen niedrigen Signifikanzwerten $p_{k_j s_i}$ jedoch dazu, dass für eine Maßnahme k_j viele Symptome s_i über viele Komponenten- und Symptomknoten $V_K, V_S \in G$ hinweg ebenfalls sehr niedrige Distanzwerte $d_{k_j s_i}$ erreichen würden. Das würde damit zur Auswahl zu vieler Symptome führen. Daher wird als zusätzliches Auswahlkriterium eine maximale Tiefe δ_{th} eingeführt, die angibt, über wie viele Komponentenknoten V_K hinweg Wege gebildet werden können, d. h.

$$\#E_{K_i, S_j} \leq 2(\delta_{th} + 1). \quad (7.105)$$

Eine Tiefe δ von null bedeutet damit, dass nur direkte Nachbar-Symptomknoten V_S des entsprechenden Komponentenknoten V_K betrachtet werden, also $\#E_{K_i, S_j} = 2$, eine Tiefe δ von eins und somit auch Wege über max. einen Komponentenknoten V_K hinweg möglich sind, also $\#E_{K_i, S_j} \leq 4$. Zur Berücksichtigung des Kriteriums Tiefe muss die Distanzfunktion (7.98) modifiziert werden zu

$$d_{k_i s_j} = \sum_{e_{k,s} \in E_{K_i, S_j}^*} p_{k,s} \quad (7.106)$$

mit $E_{K_i, S_j}^* = \begin{cases} E_{K_i, S_j} & \text{für } \#E_{K_i, S_j} \leq 2(\delta_{th} + 1) \\ \{\} & \text{sonst} \end{cases}.$

In Abbildung 7.16 wird die Merkmalsauswahl anhand der Distanzgrenze d_{th} anschaulich dargestellt. Symptome mit einer zu großen Tiefe δ sind mit (7.106) nicht in der Distanzmatrix D enthalten und können daher auch nicht ausgewählt werden.

Schritt 3:
Merkmals-Auswahl

Auswahlkriterium
Tiefe

Abbildung 7.16.: Anschauliche Darstellung der Merkmals-Auswahl anhand der Distanzmaße $d_{s_i k_j}$ und dem Parameter d_{th}

Verschiedene
Distanzgren-
zen pro Tiefe

Es scheint evident, dass direkte Signifikanzen $p_{k_j s_i}$ zwischen einer Komponente und einem Symptom für ein Diagnostik-Modell bedeutsamer sind als akkumulierte Signifikanzen. Darüberhinaus zeigen die Analysen von Distanzwerten der Tiefe $\delta = 0$ (= Signifikanzen) und Distanzwerten der Tiefe $\delta = 1$ (= akkumulierte Signifikanzen) äußerst unterschiedliche Häufigkeitsverteilungen (vgl. Abbildung 9.10 und 9.11). Es ist daher naheliegend, mit der Definition von Tiefen δ auch verschiedene Distanzgrenzwerte d_{th} pro Tiefe δ zu definieren. Die maximale Distanz für die Tiefe $\delta = 0$ wird dann durch d_{th_0} , für die Tiefe $\delta = 1$ durch d_{th_1} usw. definiert. Hierfür wird der Weg E_{K_i, S_j}^* der Distanzberechnung in (7.106) modifiziert mit

$$E_{K_i, S_j}^* = \begin{cases} E_{K_i, S_j} & \text{für } \alpha \leq 2(\delta_{th_\alpha} + 1), \quad \alpha = \#E_{K_i, S_j} \\ \{\} & \text{sonst} \end{cases}, \quad (7.107)$$

unter der (sinnvollen) Voraussetzung, dass die Distanzgrenzen d_{th_i} für steigende Tiefen δ geringer werden, d. h. $d_{th_0} \geq d_{th_1} \geq \dots \geq d_{th_n}$.

Ergebnis – Reduk-
tion auf 0,64 ‰

Wie die Evaluierung verschiedener Varianten der für Tiefe und Distanzgrenzen d_{th_i} zeigt (vgl. Abschnitt 9.3.8), konnte mit dem hier beschriebenen Verfahren zur Merkmals-Reduktion die Anzahl der Verbindungen auf 0,64 ‰ gesenkt und damit ein enormer Performanzgewinn in der Trainingsgeschwindigkeit und dem Speicherbedarf erreicht werden. Hinweise zur praktischen Bestimmung der Parameter werden im Evaluierungskapitel gegeben.

7.7. Pruning von Diagnostik-Modellen

Mit der Merkmals-Reduktion wurde bereits ein Verfahren vorgestellt, das schon vor der Netzerstellung und dem Netztraining die Anzahl der Neuronen und Verbindungen bei nahezu gleich bleibender Klassifizierungsleistung reduziert. Solche Optimierungen der Netzstruktur sind aufgrund der Komplexitätserweiterung (vgl. Kapitel 7.4) unbedingt notwendig, da mit jedem zusätzlichem Neuron der Trainingsaufwand unter Verwendung des Backpropagation-Algorithmus überproportional steigt (die Zahl der Verbindungen steigt linear).

Mit der Merkmals-Reduktion konnte die Zahl der Neuronen und Verbindungen zwar drastisch gesenkt werden, es bestehen aber immer noch viele Verbindungen, die für die Leistungsfähigkeit des Modells wenig Relevanz besitzen. Der Grund hierfür ist, dass erst nach dem Training feststellbar ist, welche Neuronen und Verbindungen tatsächlich benötigt werden. Mit abgeschlossenem Training können dann sogenannte Pruning-Verfahren zur weiteren Optimierung der Netzstruktur eingesetzt werden. Hierfür wurde zuerst ein Pruning des einschichtigen neuronalen Diagnostik-Modells durchgeführt. Nach dessen Erweiterung auf die Berücksichtigung des Fahrzeugkontexts wurde nochmals ein Pruning der neu hinzu gekommenen Verbindungen durchgeführt.

Zu den bekanntesten Pruningverfahren zählen *Weight Decay* (WD) [Wer88], *Magnitude Based Pruning* (MBP) [Sch94, Kap. 4], *Optimal Brain Damage* (OBD) [LDS90], *Optimal Brain Surgeon* (OBS) [HS93], *Optimal Skeletonization* (OS) [MS89] und evolutionäre Netzwerkoptimierung (ENZO) [Bra97]. Prinzipiell können diese Verfahren und weitere in zwei Arten des Prunings eingeteilt werden:

Pruning-Verfahren

Weight Pruning, d.h. das Entfernen von Verbindungsgewichten (dazu zählen WD, MBP und OBD).

Unit Pruning, das Entfernen von Neuronen und allen damit verbundenen Verbindungen (dazu zählen OS und ENZO; vgl. [Zel96, Kap. 25] oder [Zim94, S. 66ff.] für eine detailliertere Übersicht zu Pruning-Verfahren).

Beide Varianten haben das Ziel, möglichst diejenigen Elemente zu entfernen, die die geringste Relevanz (*salience*) besitzen und damit deren Entfernen auch den geringsten Netzwerkfehler verursacht. Im mit Definition 7.4 vorgestellten einschichtigen Diagnostik-Modell kann der durch das Entfernen einer Verbindung $w_{ij} \in \mathbf{W}$ entstehende Netzwerkfehler analog zu (7.65) mit dem Odds Ratio abgeschätzt werden durch

Relevanz von
Verbindungen

$$OR(k_i | S_{j1/0}, \mathcal{H}) = \frac{Odds(k = k_i | s_{j1}, \mathcal{H})}{Odds(k = k_i | s_{j0}, \mathcal{H})} = e^{w_{ij}}. \quad (7.108)$$

Das Odds Ratio stellt dabei den relativen Fehler des Vertrauenswertes v der Diagnose-Hypothese $h_D = (k_i, v)$ dar. Der absolute Fehler hängt

von der jeweiligen Symptommenge S_R des Reparaturfalls ab und kann daher nicht allgemein bestimmt werden. Mit dem Odds Ratio ist jedoch ersichtlich, dass durch den exponentiellen Einfluss das Entfernen von Verbindungen w_{ij} mit geringen Gewichtswerten auch nur geringe Fehler zur Folge hat (vgl. auch Abbildung 9.14), dass heißt die *saliency* eines Gewichts ist proportional zur Gewichtsstärke $|w_{ij}|$. Mit diesen Eigenschaften eignen sich besonders *Magnitude-Based-Pruning*-Verfahren (MBP) für die Netzoptimierung.

Der ursprüngliche Algorithmus, beschrieben z.B. in [Sch94, Kap. 4], sieht vor, dass iterativ zuerst das Netz trainiert, dann genau ein Gewicht entfernt und anschließend wieder nachtrainiert usw. wird. Dies wird solange fortgeführt, bis der Gesamtfehler beginnt anzusteigen. Ein solches Vorgehen würde im Kontext dieser Arbeit zum einen erhebliche Ressourcenanforderungen mit sich bringen, bedingt durch das Nachtraining, und zum anderen ist eine iterative Entfernung einzelner Gewichte für die hier verwendeten einschichtigen Netze, bedingt durch die begrenzt abbildbare Komplexität, nicht sinnvoll.

Im hier leicht modifizierten Verfahren werden im ersten Schritt, ausgehend von einem trainierten Netz mit der Gewichtsmatrix $\mathbf{W}$, alle Verbindungen mit einem Gewicht unter einer zuvor festgelegten Pruning-Grenze Pr_{th} entfernt, d.h. die neue geprunte Gewichtsmatrix $\mathbf{W}^{Pr}$ wird mit

$$W^{Pr} = \begin{bmatrix} pr(w_{11}) & pr(w_{12}) & \cdots & pr(w_{1n}) \\ pr(w_{21}) & pr(w_{22}) & \cdots & pr(w_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ pr(w_{n1}) & pr(w_{n2}) & \cdots & pr(w_{nn}) \end{bmatrix} \quad (7.109)$$

und der Pruning-Funktion

$$pr : \mathbb{R} \rightarrow \mathbb{R}, pr(w) \mapsto \begin{cases} w & \text{für } |w| \geq Pr_{th} \\ 0 & \text{für } |w| < Pr_{th} \end{cases} \quad (7.110)$$

gebildet.

Durch das Pruning können Eingangsneuronen ohne weitere Verbindungen entstehen, die im zweiten Schritt entfernt werden müssen. Hierfür werden sukzessive alle Zeilen $\mathbf{W}_{[I_i]}$ der Teilgewichtsmatrix $\mathbf{W}_I$ nach (7.59) durchlaufen und diejenigen Eingangsneuronen entfernt, für die $\mathbf{W}_{[I_i]} = \vec{0}$ gilt.

Der dritte Schritt besteht analog zum MBP in einem aus wenigen Epochen bestehenden Nachtraining. Durch das Pruning entfallen zwar nur sehr kleine Gewichte, bedingt durch die Menge ist jedoch ein Einfluss auf die Leistungsfähigkeit feststellbar, so dass eine Optimierung der verbleibenden Gewichte auf die neue Struktur notwendig ist. Keines der verbleibenden Gewichte wird durch das Nachtraining jedoch stark unterhalb der Pruninggrenze fallen, so dass ein weiterer Pruning-Schritt, in Kontrast zum MBT, nicht notwendig bzw. sinnvoll ist. Die Auswirkung

Abbildung 7.17.: Veränderung der Stärke von Verbindungsgewichten durch Nachtraining am Beispiel der Komponente „Schalter für GRA“

des Nachtrainings auf die Stärke der Verbindungsgewichte ist beispielhaft für das mit dem Ersatzteil „Schalter für GRA“¹⁸ verbundene Neuron in Abbildung 7.17 dargestellt. Das Neuron besitzt nach dem Pruning noch 97 Verbindungen (horizontale Achse), deren Wert ohne Nachtraining jeweils mit einem roten Punkt dargestellt ist. In grün folgen direkt danach, mit einer grauen Linie verbunden, die Werte nach 3 Trainingsepochen und in blau, die Werte nach 10 Trainingsepochen. Es ist ersichtlich, dass die Gewichtsänderungen begrenzt ausfallen und nur wenige Verbindungen wieder in die Pruninggrenze Pr_{th} , dargestellt durch die gestrichelten Linien, zurückfallen.

Ergebnis des
Prunings

Der Erfolg in der Reduktion der Verbindungsgewichte ist beispielhaft für die Komponente Ölstandsensor (ETRN) in Abbildung 7.18 dargestellt. Auf der linken Seite ist das Teilnetz für alle mit der Komponente verbundenen Symptome und auf der rechten Seite dasselbe Netz nach dem Pruning dargestellt. Die Verbindungen sind entsprechend ihrer Gewichtsstärke eingefärbt. Die Grafik zeigt, dass im nicht geprunten Netz noch viele grau eingefärbte, d. h. nicht relevante Verbindungen existieren. Auf der rechten Seite ist die Anzahl dieser stark reduziert. Insgesamt wurde für dieses Teilnetz die Anzahl der Verbindungen von 500 auf 15 reduziert.

Die Evaluierung in Abschnitt 9.3.9 hat ergeben, dass die Zahl der Verbindungen im Durchschnitt auf 0,05 % der möglichen Verbindungen und auf 3 % der Verbindungs-Anzahl nach der Merkmals-Reduktion verringert werden konnte. Die Verteilung zeigt, dass die meisten Netze nach dem Pruning 20 und weniger Verbindungen und damit relevante Symptome¹⁹ aufweisen. Die Leistungsfähigkeit blieb dabei gleich, sofern die Pruninggrenze Pr_{th} nicht zu hoch gewählt wurde.

Ölstandsensor 1J0907660C

¹⁸ GRA steht für Geschwindigkeitsregelanlage

¹⁹ Für das einschichtige neuronale Diagnostik-Modell ist jedes Symptom mit genau einer korrespondierenden Verbindung verknüpft.

Abbildung 7.18.: Beispiel für Pruning: Links: Teilnetz für die Komponente Ölstandsensoren mit 500 Verbindungen. Rechts: Dasselbe Teilnetz nach dem Pruning. Grün eingefärbte Verbindungen: stark positives Gewicht, rot: stark negativ, grau: nahe null

TEIL II: EVALUIERUNG

Zur Evaluierung der vorgestellten Ideen und Algorithmen wurde ein Prototyp entwickelt. Darauf aufbauend wird eine umfassende dreistufige Evaluierung, bestehend aus einer Konzeptevaluierung, einer Evaluierung mit realen Falldaten und einer Praxiserprobung, vorgestellt, die die Leistungsfähigkeit, auch im Vergleich zu anderen Verfahren wie die GFS, aufzeigt.

8 Prototypische Umsetzung

Die Evaluierung der vorgestellten Verfahren machte zunächst deren Implementierung in Prototypen notwendig. Dabei entstanden zwei getrennte, eigenständig nutzbare Anwendungen. Zum einen *Neucus*, eine hochflexible Trainings- und Validierungsumgebung für Diagnostik-Modelle und zum anderen *Neucus-Desktop*, eine Demonstrations- und Anwendungssoftware für in Neucus erstellte Diagnostik-Modelle, die unter anderem eine Fahrzeuganbindung besitzt. Beide werden nachfolgend kurz vorgestellt.

8.1. Die Trainings- und Validierungsumgebung Neucus

Anforderungen
an die Trainings-
und Validie-
rungsumgebung

Die Implementierung der Trainings- und Validierungsumgebung *Neucus* wurde bereits sehr früh in der Konzeptphase begonnen und diente der entwicklungsbegleitenden Validierung verschiedener Konzepte. Entsprechend waren die Anforderungen wie Modularität, Flexibilität, leistungsstarke Analysemöglichkeiten etc. darauf zugeschnitten. Dabei sollten Diagnostik-Modelle jedoch nicht nur trainiert, sondern auch anhand von Falldaten automatisiert validiert werden können. Viele Parameter der vorgestellten Diagnostik-Modelle können nicht analytisch bestimmt werden und müssen daher empirisch ermittelt werden, so dass auch automatisierte Suche im Parameterraum sowie verteiltes Rechnen zur Trainingszeitverkürzung zu den Anforderungen gehören. Die einzelnen Anforderungen (funktional wie nicht-funktional) lassen sich wie folgt zusammenfassen, wobei die Realisierung der jeweiligen Anforderungen mit beschrieben wird:

Realisierung

Eine Realisierung, die die genannten Anforderungen erfüllt, wird durch die Verwendung von RapidMiner [Rap10] in Verbindung mit weiteren spezifischen *Operatoren* erreicht. Die Anforderungen sowie deren Umsetzung werden nachfolgend kurz skizziert.

Wiederverwendung existierender, frei verfügbarer Software

Um den Aufwand für die Entwicklung und Implementierung des Prototypen bei gleichzeitig hoher Qualität gering zu halten, soll, soweit möglich, existierende frei verfügbare Software verwendet werden.

Um diese Anforderung zu erfüllen, wurde zunächst eine Recherche von frei verfügbarer Data-Mining-Software durchgeführt (vgl. z.B. [Duc] für eine umfangreiche und ständig erweiterte Aufstellung).

Überzeugen konnte letztlich das Data-Mining-System *RapidMiner*, das mittlerweile zu den am weitesten verbreiteten Open-Source-Systemen für Data Mining gehört [Rap10]. RapidMiner wurde 2001 unter dem Namen *Yale* vom Lehrstuhl für künstliche Intelligenz der Technischen Universität Dortmund entwickelt und ist seit 2004 bei SourceForge 4 frei verfügbar. Im Jahr 2007 erfolgte die Umbenennung in RapidMiner. Die Software bietet die Möglichkeit, sogenannte *Experimente* aus nahezu beliebig verschachtelten und in Pipes angeordneten *Operatoren* aufzubauen. Ein Operator ist dabei ein funktionaler Baustein, der den Inhalt der Pipeline als Input verwendet und die Ausgabe wieder in die Pipeline zurückschreibt, um sie an nachfolgende Operatoren weiterzureichen. Die Experimentgestaltung erfolgt dabei grafisch durch Drag&Drop oder über Menüs sehr intuitiv. Auch die Experimentdurchführung wird durch verschiedene Funktionen wie Speichermonitoring, Debug-Punkte und eine Ergebniszwischenansicht unterstützt. Ein Experiment kann als XML-Datei gespeichert und ausgetauscht werden. RapidMiner bietet neben über 100 eigenen Operatoren eine Schnittstelle zur Implementierung eigener Operatoren, die bei der Implementierung von Neucus verwendet wurde.

RapidMiner

Modular und flexibel Häufig steht zu Beginn eines Forschungsprojektes die letzte Realisierung bestenfalls skizzenhaft fest. Modularität der Funktionalität, einfache Austauschbarkeit von Varianten der Datenvorverarbeitung oder des Lernverfahrens etc. sind somit unbedingt notwendig.

Die Modularität und Flexibilität wird im RapidMiner-System durch Operatoren mit festgelegten (einheitlichen) Schnittstellen gewährleistet, die, sofern die Schnittstellen kompatibel sind, beliebig kombiniert werden können. Leider konnten zur Umsetzung der Algorithmen nur wenige der bereits vorhandenen Operatoren verwendet werden. Die Gründe hierfür waren vor allem in der besonderen Datenstruktur der Reparaturfalldaten zu finden. So wurden letztlich über 48 weitere Operatoren implementiert.

Eigene Operatoren
notwendig

Robust Robustheit ist in mehrerlei Hinsicht gefordert. Data-Mining-Verfahren besitzen die Eigenschaft, dass oftmals trotz fehlerhafter Implementierung (ausgenommen triviale Syntaxfehler) Ergebnisse entstehen, denen Implementierungsfehler nicht offensichtlich anzusehen sind. Robustheit gegenüber Inkonsistenzen ist daher sinnvoll. Bedingt durch die Rahmenbedingungen ist zu erwarten, dass viele Implementierungen durch Studenten vorgenommen werden, so dass eher selten von ausreichend Praxiserfahrung ausgegangen werden kann. Darum wurde als Programmiersprache Java gefordert. RapidMiner ist ebenfalls in Java implementiert.

Drag&Drop

Intuitive, grafische Bedienung Eine intuitive Bedienung und Verständlichkeit der durchgeführten Versuche reduzierten die Fehlerzahl und erhöhten die Effizienz.

Eben diese Anforderungen stehen auch im Anwendungskonzept von RapidMiner im Vordergrund. Ein Screenshot der Trainings- und Validierungsumgebung Neucus ist in Abbildung 8.1 zu sehen, hier mit den Eigenschaften des Operators *GetNeucusFalldaten*. Links in Abbildung 8.1 ist die Struktur des Experiments, bestehend aus verschachtelten Operatoren zu sehen. Die Operatoren können per Drag&Drop aneinandergereiht oder verschoben werden. Auch individuelle Benennungen zur besseren Verständlichkeit sind möglich. Die rechte Seite der Abbildung zeigt die Eigenschaften des markierten Operators.

Komfortable Analysemöglichkeiten Um die Leistungsfähigkeit der einzelnen Module wie Datenvorverarbeitung, Lernverfahren usw. beurteilen zu können, sind effiziente Analysemöglichkeiten wie Scatterplots, Histogramme, etc. gefordert.

Abbildung 8.2 zeigt ein Beispiel für integrierte Analysemöglichkeiten, hier ein Histogramm der durch den Operator *Signifikanzmatrix* erzeugten Daten. Im Rahmen der Neucus-Operatoren wurden weitere, spezialisierte Analyse-Operatoren, wie die automatisierte Validierung durch Falldaten, implementiert (diese wird an anderer Stelle näher vorgestellt).

Standardisierte Versuchsbeschreibung Um eine Auswertung der Ergebnisse vieler Versuche über einen längeren Zeitraum und mehrere Versionen hinweg zu gewährleisten, ist eine standardisierte Versuchsbeschreibung sehr hilfreich.

Versuche, in RapidMiner *Experimente* genannt, werden im XML-Format gespeichert und ausgetauscht. Dabei werden Operatorname, Operatorbeschreibungen und Operatorparameter in einem festgelegten Format definiert.

Abbildung 8.1.: Screenshot der Trainings- und Validierungsumgebung Neucus, hier mit den Eigenschaften des Operators *GetNeucusFalldaten*

Abbildung 8.2.: Screenshot der Analyse-Möglichkeiten, hier ein Histogramm der durch den Operator *Signifikanzmatrix* erzeugten Daten.

Einfache Automatisierbarkeit Lange Versuchsreihen mit verschiedenen Parametern müssen automatisiert durchführbar sein.

Hinsichtlich Automatisierbarkeit bietet RapidMiner spezielle Operatoren an, in denen einzelne Versuche automatisiert geladen und durchgeführt werden können und Parameteriterationen/-optimierungen möglich sind. Eine weitere Automatisierbarkeit ist durch die einfache Integration in andere Anwendungen (wenige Zeilen Code) ohne GUI oder, alternativ, über eine Konsolenschnittstelle zum Scripting möglich.

Parallelisierbarkeit, verteiltes Rechnen Einzelne Versuchsreihen, speziell Parameteriterationen, können einen großen Rechenaufwand bedeuten. Um die Versuchszeit zu verkürzen, soll eine Parallelisierung auf mehrere Rechner möglich sein.

Parallelverarbeitung
durch JPPF

Gerade Experimente mit Suchvorgängen im Parameterraum, die zum Ziel haben, ein möglichst optimales Parameterset z. B. für das Training von neuronalen Netzen zu finden, sind sehr rechenaufwändig. Für jedes Parameterset muss ein eigenes Diagnostik-Modell erstellt, trainiert und validiert werden. Je nach Modellart und Rechnerleistung kann ein solcher Vorgang bis zu 26 Stunden dauern¹. Zur Parallelisierung der Aufgabe wurde das *Java Parallel Processing Framework* (JPPF) [JPP10] verwendet und als eigenes Operatorset in RapidMiner integriert. Das Konzept wird in Abbildung 8.3 verdeutlicht. Ein oder mehrere *Clients* vergeben Sets von

Abbildung 8.3.: Das Konzept von JPPF [JPP09]

¹ gilt für einen Intel Core Duo P8600 @ 2,4GHz und 2GB RAM

Jobs an einen *JPPF-Server*. Dieser verwaltet eine Reihe an *Nodes*, die *Jobs* annehmen und verarbeiten können. Je nach Auslastung und Konfiguration der *Nodes* werden die *Jobs* durch den Server dynamisch verteilt und die Ergebnisse wieder an den entsprechenden *Client* zurückgeschickt. Für Neucus wurde JPPF als *Parallel Processing Operator* in RapidMiner implementiert und eingebunden. Der jeweilige Versuch in RapidMiner war dabei der Client. Als Nodes wurden die im Institut verfügbaren und gerade nicht verwendeten Rechner verwendet. Die *Nodes* wurden dabei als Bildschirmschoner realisiert, d. h. sobald der JPPS-Bildschirmschoner aktiviert wurde begann dieser als *Node Jobs* anzunehmen und zu verarbeiten. Im JPPF-Framework integriert ist die automatisierte Propagation des für die Jobs notwendigen Programm-Codes. Somit müssen trotz Softwareänderungen die JPPF-Bildschirmschoner nur einmal installiert werden. Auf diese Weise wurden die umfangreichen Untersuchungen unter den zeitlichen Rahmenbedingungen erst so möglich.

Bildschirmschoner
als JPPF-Node

Trotz der Verwendung von RapidMiner als leistungsstarke Plattform sind mit der Summe der Operatoren und Analyse-Möglichkeiten mehr als 29.000 Lines of Code entstanden. Die Code-Qualität wurde durch Code-Styles, automatisierte Code-Metriken, gegenseitige Code-Reviews und Unit-Tests sichergestellt [Jon08].

8.2. Die Anwendungssoftware Neucus-Desktop

Für die praktische Erprobung der Verfahren war eine reine Trainings- und Validierungsumgebung nicht ausreichend. Die Anforderungen hierfür waren die direkte Demonstration der Lösungsvorschläge (Komponenten K) für verschiedene vorhandene Fehlersymptome S , inkl. Fahrzeugkontext C , eine direkte Anbindung an ein Diagnoseinterface zur schnelleren Erprobung direkt am Fahrzeug, die Evaluation verschiedener Diagnostik-Modelle, usw. Zusammengefasst mussten folgende Anforderungen erfüllt werden:

Eigenständige, plattformunabhängige Anwendung Das primäre Ziel der Demonstration der Funktionsfähigkeit der Verfahren erfordert eine eigenständige, möglichst plattformunabhängige Anwendung.

Dies wurde durch eine mit AWT² implementierte Java-Anwendung realisiert. Dabei konnte zwar auf viele Standardbibliotheken zurückgegriffen werden, Softwareanwendungen, die direkt oder mit

²Das Abstract Window Toolkit [AWT] ist eine API zum Entwickeln quasi-plattformunabhängiger grafischer Oberflächen mit Java.

leichten Modifikationen verwendet werden hätten können, existierten jedoch nicht.

Ansprechende Benutzeroberfläche Eine Anforderung, die sich aus der Aufgabenstellung ergibt, ist ein ansprechendes Design der Benutzeroberfläche.

In der Praxis muss hier immer auch ein Kompromiss zwischen Design, Aufwand und „*Entwickleroptionen*“ gefunden werden.

Direkte Fahrzeuganbindung Die direkte Erprobung von Fehlerfällen am Fahrzeug, wie sie später vorgestellt wird, erfordert eine Fahrzeuganbindung.

Diese wurde mittels des Diagnose-Interfaces VAS 5053, eine spezielle PCMCIA-Einsteckkarte, realisiert. Der Auslesevorgang wird per Mausklick gestartet und die Ergebnisse sowie die Lösungsvorschläge wurden automatisiert berechnet und dargestellt.

Verwaltung von Fehlerbildern Die ausgelesenen Fehlerbilder sollen im XML-Format gespeichert werden können. Weiterhin sollen aber auch Tester-Protokolle des VAS 5051 geöffnet werden können.

Verwaltung von Diagnostik-Modellen Ein erklärtes Ziel ist der experimentelle Vergleich verschiedener Diagnostik-Modelle. Eine Verwaltung zur Auswahl verschiedener Modelle ist dabei obligatorisch.

Um die Kompatibilität der Modelle zu gewährleisten, wurde das Interface *PredictionModell* entwickelt, das die Funktionalität eines Diagnostik-Modells entsprechend kapselt. So können Bayes'sche Modelle ebenso wie statistische und neuronale Modelle verwendet werden.

Filtermöglichkeiten der Ergebnisdarstellung Im Rahmen der Falldatenanalyse (Kapitel 5) werden verschiedene Lösungselemente wie Ersatzteilnummern, Arbeitspositionen, Kundendienstnummern, etc. vorgestellt. Eine entsprechende Filterung nach diesen Kategorien wird gefordert.

Annotation der Ergebnisse mit statistischen Zusatzinformationen

Um die Plausibilität eines Lösungsvorschlags überprüfen zu können und ggf. eine Fehlersuche zu erleichtern, sollen zusätzlich statistische Informationen wie die Anzahl der Referenzfälle, die Werte der bedingten Wahrscheinlichkeiten, etc. dargestellt werden.

Realisiert wurde das durch optionale Annotationen im Diagnostik-Modell. Solche (statistischen) Informationen sind naturgemäß nicht modellinhärent und müssen daher beim Training separat berechnet

werden. Die Darstellung erfolgt über ein Infosymbol mit Hover-Funktion³.

Ein Screenshot der mit diesen Anforderungen entstandenen Anwendung ist in Abbildung 8.4 zu sehen. Darin sind die Lösungsvorschläge für den Ereignisspeichereintrag 895 (Bedienteil für Geschwindigkeitsregelanlage (GRA)) im Steuergerät „Lenkradelektronik“ dargestellt. Auf der rechten Seite sind die Schaltflächen für das Speichern und Öffnen von Fehlerbildern, dem Einlesen von Symptomen über das Diagnose-Interface usw. dargestellt. Die Filtermöglichkeiten sind über die Leiste am unteren Rand auswählbar. Für eine ausführliche Dokumentation der Trainings- und Validierungsumgebung sowie der Demonstrations- und Anwendungssoftware wird auf [Wit08], [Met09] und [Mül08] verwiesen.

³ Auch Mouse-over-Effekt genannt. Beim Überfahren des Infosymbols mit der Maus wird temporär ein Informationsfenster angezeigt.

Abbildung 8.4.: Screenshot der Anwendungsumgebung Neucus-Desktop mit den Lösungsvorschlägen für den Ereignisspeichereintrag 895 (Bedienteil für Geschwindigkeitsregelanlage (GRA))

9 Evaluierung von Diagnostikverfahren

Die Evaluation hatte zum Ziel, nicht nur einen Nachweis für die praktische Anwendbarkeit der Ideen und Konzepte zu erbringen, sondern darüber hinaus eine vergleichende Bewertung zwischen den entwickelten Diagnostik-Modellen selbst und etablierten Verfahren wie der Geführten Fehlersuche zu geben. Neben dem Vergleich wird die für einen Praxiseinsatz tatsächlich erwartbare Leistungsfähigkeit abgeschätzt und Randbedingungen wie die notwendige Mindestfallzahl und der Einfluss verschiedener Modell-Parameter etc. untersucht.

Die Diversität der zu vergleichenden Verfahren erfordern ein mehrstufiges Vorgehen auf verschiedenen Ebenen. Hierzu wurde ein mehrstufiges entwicklungsbegleitendes Evaluationsverfahren entwickelt, das zunächst allgemein eingeführt wird. Anschließend wird es auf die in dieser Arbeit entwickelten Verfahren angewendet und die Ergebnisse der einzelnen Stufen präsentiert und diskutiert. Die erste Stufe umfasst eine Konzeptevaluierung, die die Leistungsfähigkeit des lernenden Verfahrens mit der der Geführten Fehlersuche (GFS) vergleicht. In der zweiten Stufe, der Evaluierung mit Felddaten, wird insbesondere auf die Leistungsfähigkeit der einzelnen Diagnostik-Modelle sowie Aspekte der Bestimmung der Modellparameter eingegangen. Abgeschlossen wird die Evaluierung durch eine Praxiserprobung, die eine manuelle Untersuchung mit einer begrenzten Anzahl von Fällen, eine Untersuchung weniger exemplarischer Fälle in Zusammenarbeit mit der Qualitätssicherung der Volkswagen AG und einer Praxiserprobung am Fahrzeug umfasst. Hier soll vor allem die praktische Anwendbarkeit der Modelle gezeigt und ein direkter Vergleich mit der GFS durchgeführt und bewertet werden.

Evaluierung kann allgemein als Prozess der Untersuchung und der Bewertung verstanden werden [Bal98]. Sie grenzt sich von der Validierung insofern ab, wie sie nicht funktionale, sondern Qualitätskriterien betrachtet. Die reine Funktion des zu evaluierenden Systems gemäß den Anforderungen muss als gegeben vorausgesetzt werden und ist daher nicht Bestandteil dieses Kapitels. Das Ergebnis einer Evaluation ist eine Aussage anhand eines Qualitätsmaßes $q \in [0\%..100\%]$. Von einem *Evaluierungsverfahren* werden als Konsequenz Eigenschaften wie Vergleichbarkeit mit anderen Verfahren und eine ganzheitliche Betrachtungsweise gefordert, die Vorteile und Nachteile in unterschiedlichen Bereichen mit einbezieht und bewertet. Eine besondere Herausforderung stellt die Heterogenität der Verfahren dar. Ein Vergleich von z.B. regelbasierten Verfahren wie die Geführte Fehlersuche mit einem lernenden Verfahren, wie das hier zu evaluierende, bedarf neuer Evaluierungsverfahren.

Anforderungen
an ein Evalua-
tionsverfahren

9.1. Entwicklungsbegleitende dreistufige Evaluierung

Die Evaluation von Offboard-Diagnostikverfahren unterliegt einigen Besonderheiten, so dass hier eine mehrstufige entwicklungsbegleitende Strategie aus drei Gründen für sinnvoll erachtet wurde:

Besonderheiten der
Evaluation von
Diagnostikverfahren

1. Diagnostiksysteme werden speziell für einen bestimmten Anwendungsfall entwickelt,
2. der Vergleich verschiedenartiger Diagnostiksysteme ist nicht trivial und
3. die Erprobung von Diagnostiksystemen ist mit hohem Aufwand verbunden.

Der klassische Evaluierungsansatz (vgl. [Bal98]) ist für den hier vorhandenen Anwendungsfall folglich ungeeignet und muss entsprechend adaptiert werden. Hier hat sich eine dreistufige entwicklungsbegleitende Evaluierung als geeignet herausgestellt, die neben einer reinen Bewertung auch das Erreichen des übergeordneten Ziels, eine optimierte Diagnostik, mit unterschiedlichsten Anforderungen und Stakeholdern systematisch sicherstellt.

Phase 1 – Konzeptbewertung In der erste Phase, der Konzeptbewertung, werden die Diagnostikkonzepte, die eingesetzten Verfahren und das mögliche Vorgehen bei der Einführung des Systems bewertet. Während dieser Phase muss noch kein funktionierendes System vorliegen. Im ersten Schritt wird überprüft, ob die gewählten Diagnostikverfahren für den geplanten Einsatzbereich, wie Werkstätten oder Produktionsanlagen, anwendbar sind.

Der zweite Schritt ist die Überprüfung der technischen Machbarkeit. Die Konzeptbewertung wird im nächsten Kapitel ausführlich beschrieben. Ergebnis dieser Phase ist eine Aussage darüber, ob ein bestimmtes Konzept weiter verfolgt oder verworfen wird. In dieser Phase werden auch zusätzliche Anforderungen an das fertige Produkt und den Prototypen, der in Phase 2 eingesetzt werden soll, definiert.

Phase 2 – Evaluierung mit Falldaten In der Phase 2 werden prototypisch umgesetzte Diagnostiksysteme untersucht, deren zugrundeliegendes Konzept bereits in der ersten Phase als erfolgversprechend erachtet wurde. Dies erfolgt sowohl als reine Laboruntersuchung mit künstlich erzeugten Daten, als auch mit realen Falldaten. Die Prototypen müssen noch nicht den vollen Funktionsumfang des Diagnostiksystems enthalten. Es sollten jedoch das grundlegende Konzept sowie Funktionalitäten, die in der ersten Phase als kritisch eingestuft wurden, umgesetzt sein. Mit Hilfe der Prototypen muss

es möglich sein, die zu erwartende Leistungsfähigkeit der Systeme zu ermitteln. Im Abschnitt 9.4 wird eine Technik dafür vorgestellt, die die Möglichkeit bietet, auch sehr unterschiedliche Verfahren miteinander vergleichen zu können.

Phase 3 – Praxiserprobung Für die Phase 3 der entwicklungsbegleitenden Evaluierung muss das Diagnostiksystem in einer sehr ausgereiften Form vorliegen. Das Ziel dieser Phase ist es, sicherzustellen, dass das Diagnostiksystem auch für den praktischen Einsatz am Fahrzeug geeignet ist. Hierzu sollte das System an möglichst vielen Fallbeispielen aus der Praxis erprobt werden, um Stück für Stück Softwarefehler wie auch Verfahrensfehler zu reduzieren. Zu den kritischen Funktionalitäten, die mit Hilfe der Prototypen geprüft werden, gehören in der Regel auch Schnittstellen zu anderen Systemen sowie der Import und Export von Daten. Dies ist wichtig, um den entstehenden Aufwand beim Rollout der Applikation abschätzen zu können. Auch eine Bewertung der Benutzerschnittstelle ist in dieser Phase sinnvoll, um eine entsprechende Benutzerakzeptanz zu erreichen. Dabei ist es nicht nur wichtig, wie der KD-Mitarbeiter in der Werkstatt mit dem System arbeiten kann, sondern es muss auch untersucht werden, ob die Eingabe und Pflege des Diagnostikwissens mit dem zu erwartenden Aufwand durchführbar ist.

9.2. Konzeptbewertung

Umfassendes
Kriterienset

Als Vergleichsverfahren in der ersten Phase der Evaluation wird die etablierte Geführte Fehlersuche herangezogen. In [MFK08] wurde dazu gemeinsam ein umfassendes Kriterienset entwickelt und vorgestellt, das im Folgenden verwendet werden soll (vgl. Abbildung 9.1). Es wurden dabei 5 Hauptkriterien mit insgesamt 25 Unterkriterien identifiziert.

Qualitative Be-
wertung gefordert

Evaluationsverfahren erfordern normalerweise eine Gewichtung und Integration dieser Kriterien. Wie in [MFK08] angemerkt, hängt die Bewertung der einzelnen Kriterien jedoch auch von Unternehmenszielen, d.h. strategische und politische Überlegungen, subjektiven Wahrnehmungen, Überzeugungen und Bedenken ab, so dass ein quantifiziertes Gesamtevaluationsergebnis in dieser Phase zunächst keinen Sinn macht. Vielmehr ist eine qualitative Bewertung und vergleichende Gegenüberstellung der Kriterien gefragt. Hierzu wurden übereinstimmend mit [Kriar] die in Tabelle 9.2 dargestellten fünf Bewertungsstufen entwickelt.

Ausführliche Be-
gründung im
Anhang C.1

Die einzelnen Kriterien aus Abbildung 9.1 werden in Tabelle 9.1 bewertet und mit kurzen Stichpunkten begründet. Eine ausführliche Diskussion der jeweiligen Bewertung, die an dieser Stelle zu weit gehen würde, ist im Anhang C.1 zu finden.

Abbildung 9.1.: Ergebnis der Konzeptevaluierung

Zusammenfassung der Konzeptbewertung

Die Diskussion der einzelnen Kriterien konnte Vor- wie auch Nachteile von Neucus gegenüber der GFS aufzeigen. Die Ergebnisse können anhand der Abbildung 9.1 veranschaulicht werden. Hierbei wurden die einzelnen Kriterien mit den passenden Ampelfarben gemäß der Bewertungen aus Tabelle 9.1 eingefärbt. Es wurde dabei auch auf eine Vergleichbarkeit mit [Kriar] geachtet.

Die Abbildung verdeutlicht, dass Neucus der bisherigen GFS in vielen Punkten überlegen ist, jedoch inhärent in der Differenzierbarkeit Schwächen besitzt. Neucus identifiziert defekte Komponenten, kann aber schwer feingranularere Ursachen angeben. Die GFS hat im Gegensatz dazu die Möglichkeit, detaillierte Komponentenprüfungen zu realisieren, die zudem auch hinsichtlich der Gesamtkomplexität des Fahrzeugs beherrschbar sind. Mit der Konzeptevaluierung bestätigen sich die Leistungsfähigkeit von Neucus und die Maximierung des Nutzens in der Kombination mit der GFS.

Stärke: Trefferquote;
Schwäche: Differenzierbarkeit

<i>Kriterium</i>	<i>Begründung</i>	<i>Bewertung</i>
Anwendungsfälle und Einsatzbereich		
elektrische oder mechanische Fehler	Es lassen sich elektrische, mechatronische und mechanische Fehler diagnostizieren	⊙
Kundendienstwerkstatt	Stand-Alone möglich, deutliche Vorteile in der Kombination mit der GFS durch kollektive Wissensakquisition usw.	⊕⊕
Fahrzeugproduktion	Einsatz zur Optimierung der Fehlersuche am Ende der Produktionslinie	⊕
Entwicklung, Forschung	Scheidet als Einsatzbereich aufgrund mangelnder Lerndaten prinzipbedingt aus	⊖⊖
Technische Machbarkeit		
Zusammenspiel mit IT-Systemen	Infrastruktur zur Datenerfassung, Verbreitung der Modelle und zur Datenanalyse vollständig vorhanden.	⊕⊕
Zusammenspiel mit Fahrzeugen	Für alle Fahrzeuge geeignet. Anpassungsfähig gegenüber zukünftigen Fahrzeugarchitekturen.	⊕
Kosten		
Investitionskosten	Keine signifikanten Investitionen notwendig	⊙
Laufende Kosten	Keine manuelle Pflege der Wissensbasis	⊕⊕
Kosten für neue Fahrzeugprojekte	Kein großer manueller Aufwand notwendig	⊕⊕
Leistungsfähigkeit		
Trefferquote	Beide Systeme können hohe Trefferquoten erzielen, für die GFS jedoch mit hohem Aufwand	⊕
Differenzierbarkeit	Beide Systeme können theoretisch eine hohe Differenzierbarkeit erreichen, in der Praxis ist diese bei beiden geringer	⊖
Fehlersuchzeit	Durch die Minimierung des mittleren quadratischen Fehlers hat Neucus Vorteile	⊕
Erweiterbarkeit, Skalierbarkeit	Neucus ist plattformunabhängig und baut auf Open-Source-Software auf. Erweiterungen der GFS sehr aufwändig.	⊕⊕
Variantenvielfalt beherrschbar	Für die GFS sehr aufwändig. Neucus lernt ohne Mehraufwand Varianten zu unterscheiden.	⊕⊕
Anwendbarkeit und Nutzerakzeptanz		
Internationalisierung	Übersetzungen für Reparatur-Lösungen und Symptomkataloge vorhanden. GFS-Programme müssen übersetzt werden.	⊕
Flexibilität der Benutzerverführung	Maximale Flexibilität durch gewichtete Liste an möglich Ursachen. GFS sehr starr.	⊕⊕
Anzeige von Zusatzinformationen	Beide Systeme können vielfältige Informationen integrieren/verknüpfen.	⊙
Nachvollziehbarkeit	Nachvollziehbarkeit der GFS theoretisch gegeben, praktisch aber oft vernachlässigt. In Neucus möglich durch zus. stat. Informationen.	⊙

Tabelle 9.1.: Übersicht der Einzelbewertungen der Konzeptevaluierung

Symbol	Bedeutung
$\oplus\oplus$	Neucus ist bei diesem Kriterium wesentlich besser als die GFS
$\oplus$	Neucus hat bezüglich dieses Kriteriums Vorteile gegenüber der GFS
$\odot$	Die Stärken und Schwächen beider Systeme sind vergleichbar
$\ominus$	Neucus hat bezüglich dieses Kriteriums Schwächen gegenüber der GFS
$\ominus\ominus$	Neucus stellt bezüglich dieses Kriteriums keine Alternative zur GFS dar (KO-Kriterium)

Tabelle 9.2.: Bewertungssymbole und deren Bedeutung

9.3. Evaluierung mit Reparaturfalldaten

Die zweite Phase der Evaluierung hat das Ziel, die in der Konzeptevaluierung noch grob abgeschätzte Leistungsfähigkeit konkret zu quantifizieren. Dabei wird nicht nur die Leistungsfähigkeit der einzelnen, in Kapitel 7 entwickelten Verfahren ermittelt, sondern auch die Bestimmung und Auswirkung der verschiedenen Parameter untersucht und diskutiert. Die Voraussetzung für diese Evaluierung ist zum einen ein prototypisch umgesetztes Diagnostiksystem, wie es mit dem Prototyp Neucus in Kapitel 8 vorgestellt wurde und zum anderen aussagekräftige Referenzfälle. Als Referenzfälle soll die in Kapitel 5 vorgestellte Datenbasis mit realen Reparaturfällen dienen. Dargestellt werden hier nur die aus Sicht des Autors relevantesten Ergebnisse. Weitere Untersuchungen sind im Anhang C.2 zu finden.

9.3.1. Kriterien und praktische Umsetzung

Die Evaluierung mit Reparaturfalldaten kann nur hinsichtlich des Kriteriums *Trefferquote* erfolgen. Das Unterkriterium Differenzierbarkeit lässt sich nicht automatisiert anhand der zur Verfügung stehenden Daten ermitteln. Wiederum indirekt berücksichtigt werden kann das Kriterium *Fehlersuchzeit*. Dieses hängt, wie auch die Fehltauschrate¹, unmittelbar von der Trefferquote ab. Der Zusammenhang wird deutlich, wenn eine beispielhafte Ausgabe des Systems in Tabelle 9.3 betrachtet wird (vgl. auch Screenshots von Neucus in Kapitel 8.2).

Wird die tatsächlich defekte Komponente K_i auch vom System an erster Stelle mit hohem Vertrauenswert v vorgeschlagen, wird sich die Fehlersuchzeit auf die Überprüfung des tatsächlichen Defekts der Komponente beschränken. Ein Fehltausch wird ebenfalls unwahrscheinlich.

¹ Die Fehltauschrate ist ebenfalls ein Unterkriterium der Fehltauschkosten, wurde aber in der Konzeptevaluierung aufgrund des skizzierten Zusammenhangs zur Trefferquote nicht gesondert aufgeführt.

Zusammenhang
von Trefferquote,
Fehlersuchzeit und
Fehltauschrate

Anders verhält es sich, wenn die defekte Komponente vom System erst in den unteren Rängen mit niedrigem Vertrauenswert oder gar nicht vorgeschlagen wird. Lange Fehlersuchen wie auch auf Verdacht getauschte Komponenten werden sehr viel wahrscheinlicher. Nicht berücksichtigt werden kann dabei der Einfluss der Erfahrung des Mechanikers. Durchaus kann dieser aufgrund seiner Erfahrung zuerst mit der Überprüfung einer anderen als der zuerst vorgeschlagenen Komponente beginnen usw.

Pos.	Diagnose-Hypothese	Vertrauenswert
1	Komponente xyz	77 %
2	Komponente xyz	56 %
3	Komponente xyz	24 %
4	Komponente xyz	18 %
5	Komponente xyz	17 %

Tabelle 9.3.: Ausgabe des Diagnostik-Systems

Evaluierung
mit der GFS

Weiterhin interessant wäre ein direkter Vergleich mit der GFS. Dieser wäre insofern möglich, da die einzelnen Fehlersuchprogramme Komponenten verdächtigen und prüfen (lassen). Allerdings würde das eine maschinelle Auswertbarkeit der Fehlersuchprogramme voraussetzen, die zum Zeitpunkt des Projekts nicht gegeben war. Inwieweit das mit dem Trend zur *modularisierten GFS* zukünftig möglich sein wird, bleibt offen. Die nachfolgende Evaluierung kann sich daher nicht direkt auf die GFS beziehen. Evaluiert werden jedoch verschiedene Varianten von Modellen oder Modellparametern, so dass die Nomenklatur *Evaluierung* nach wie vor gerechtfertigt bleibt.

Realisierung als Plu-
gin in RapidMiner

Für die Evaluierung des vorgestellten Verfahrens wurde ein eigenes Plugin in RapidMiner (siehe Kapitel 8.1) entwickelt. Dieses Plugin ermöglicht die vollautomatisierte Evaluation mit Hilfe einer Menge von Reparaturfalldatensätzen.

9.3.2. Definition und Berechnung der Evaluierungsquote

Automatisierte
Berechnung der
realen Trefferquo-
te nicht möglich

Mit der Analyse der Datenbasis wurde deutlich, dass eine automatisierte Berechnung der realen Trefferquote nicht möglich ist. Das hat mehrere Gründe: Die als inkongruent (vgl. Abschnitt 5.1) bezeichneten Fälle besitzen die Eigenschaft, dass sie per Definition keinen plausiblen Zusammenhang zwischen Symptomen S_R und Reparaturmaßnahmen K_R zulassen. Der Beispielfall in Tabelle B.6 veranschaulicht das. Keine der dort vorhandenen Ereignisspeichereinträge zu Abblendlicht, Dimmung und Fensterheber lassen auf die Reparaturmaßnahme zum Magnetventil schließen. Es fehlt hierzu die passend kodierte Kundenwahrnehmung,

die vermutlich ein geeignetes Symptom dargestellt hätte. Eine Evaluierung der Trefferquote wäre mit diesem Fall nicht sinnvoll, denn ein gutes Diagnostik-Modell sollte basierend auf den vorhandenen Symptomen nicht auf die Reparaturmaßnahme Magnetventil schließen. Ein weiterer Grund ist die Diversität in der Darstellung der Reparaturmaßnahme. In einigen Fällen liefert beispielsweise nur das Ersatzteil (ET) den passenden Lösungshinweis und eine Werkstattwahrnehmung (M-DISS) ist nicht vorhanden (der Sachverhalt wurde bereits in Abschnitt 5.2, speziell Abbildung 5.2 erörtert). Würde das Diagnostik-Modell jetzt jedoch die Werkstattwahrnehmung vorschlagen, könnte die automatisierte Evaluierung keinen Treffer feststellen, denn eine Verknüpfung zwischen Ersatzteilen und Werkstattwahrnehmungen hinsichtlich der betroffenen Komponente existiert nicht. Konsequenterweise würde mit der automatisierten Evaluierung selbst ein *perfektes* Diagnostik-Modell keine 100 %-Trefferquote erreichen können. Das ist nur mit der später vorgestellten manuellen Evaluierung möglich, in der eine solche Zuordnung manuell berücksichtigt wird. Daher wird im Rahmen der automatisierten Evaluierung nicht von einer Trefferquote, sondern von einer *Evaluierungsquote* gesprochen.

Verfahren zur automatisierten Evaluierung

Das Verfahren zur Berechnung der Evaluierungsquote wird vereinfacht in Abbildung 9.2 dargestellt. Ausgehend von der zur Verfügung stehenden Fallbasis R_{FB} muss zunächst eine Evaluierungsmenge R_{Eval} ausgewählt werden (oben links skizziert). Die Evaluierungsmenge darf dabei nicht Teil der Trainingsmenge des zu evaluierenden Modells Π sein. Es wird sukzessive jeder Reparaturfall $R_i \in R_{Eval}$ betrachtet. Das zu evaluierende Diagnostik-Modell Π berechnet dann anhand der vorhandenen Symptome $S_R \in R_i$ eine Menge an nach Vertrauenswert $v_i \in h_D$ absteigend sortierten Diagnostik-Hypothesen H_D , die mit den tatsächlich im Reparaturfall R_i vorhandenen Komponenten $K_R \in R_i$ verglichen werden. Dazu wird zunächst jede Diagnose-Hypothese h_D aus der absteigend sortierten Menge H_D erweitert um einen Index $i \in \mathbb{N}$ mit $h_{Di} = (k, v, i)$, $h_{Di} \in H_{Di}$ und $v_1 \geq v_2 \geq \dots \geq v_n$ für aufsteigende i . Der Trefferindex $t_{i,R}$ des Reparaturfalls R_i berechnet sich dann aus der Treffermenge T_R mit

$$t_{i,R} = \min(I), I = \bigcup_{t \in T_R} (i | i \in t, i \leq t_{max}), \quad (9.1)$$

wobei die Treffermenge T_R definiert ist mit

$$T_R = \bigcup_{h_{Di} \in H_{Di}} \{(k, v, i) | k \in K_R \wedge k \in h_{Di} \wedge v \in h_{Di} \wedge i \in h_{Di}\}. \quad (9.2)$$

Betrachtet werden, wie Gleichung (9.1) zeigt, nur die ersten t_{max} Diagnostik-Hypothesen, wobei t_{max} hier auf 5 festgelegt wurde. Die indizierten

Evaluierungs- vs.
Trainingsmenge

Trefferindex

defekte Komponente verweisen, aber in Fällen, in denen der Verweis z. B. in der Werkstattwahrnehmung (M-DISS) zu finden ist, würde die Evaluierung ungerechtfertigt keinen Treffer feststellen. Das Maximum der 100 %-Grenze kann jedoch sicher mit dem Anteil an kongruenten Fällen definiert werden. Dieser Anteil wurde bereits im Rahmen der manuellen Falldatenuntersuchung (Tabelle 5.2) mit 77 % bestimmt. Mit einer derart definierten Grenze würde nur ein solches „ideales“ Diagnostik-Modell diese Grenze erreichen, welches jeweils alle vier verfügbaren Lösungshinweise auf eine defekte Komponente gleichzeitig erzeugt. Dennoch scheint das ein guter Kompromiss zur Normalisierung zu sein. Die normalisierte Evaluierungsquote berechnet sich dann zu

$$q_{R_{Eval, Norm}} = \frac{q_{R_{Eval}}}{q_{R_{Eval, Max}}} \text{ mit } q_{R_{Eval, Max}} = 77 \%. \quad (9.5)$$

Der Einfachheit halber wird im Folgenden nur noch die Bezeichnung q_n anstelle von $q_{R_{Eval, Norm}}$ verwendet.

9.3.3. Partitionierung der Evaluierungs- und Trainingsdaten

Üblicherweise wird im Data-Mining zur Bestimmung von Modell-Güten eine *Kreuzvalidierung* durchgeführt. Aufgrund der chronologischen Veränderungen der Falldaten ist eine Kreuzvalidierung hier jedoch nicht sinnvoll. Insbesondere die Eigenschaft, dass bestimmte Fehler erst ab einem bestimmten (Produktions-)Datum im Feld auftauchen oder auch wieder verschwinden (Fehlerabstellmaßnahmen, Rückrufaktionen, etc.) machen es notwendig, dass die Evaluierungsmenge hinsichtlich des Reparaturdatums immer nach der Trainingsmenge liegt, was für eine Kreuzvalidierung nicht gegeben wäre. Für die nachfolgenden Evaluierungen wurden daher, sofern nicht gesondert erwähnt, Reparaturfälle im Zeitraum von Januar 2006 bis August 2007 (die ersten 20 Monate, ca. 50.000 Fälle) verwendet. Für die Evaluierung wurden die Fälle der nachfolgenden vier Monate vor September 2007 bis Dezember 2007 herangezogen² (ca. 6.000 Fälle).

Kreuzvalidierung
nicht sinnvoll

Fallzahl und
Partitionierung

9.3.4. Evaluierung der neuronalen Diagnostik-Modelle

Für die Evaluierung der Diagnostik-Modelle ist zuerst das einschichtige neuronale Diagnostik-Modell nach Definition 7.4 von Bedeutung. Es stellt den zentralen Kern der neuronalen Modelle dar, der auch nach der späteren Erweiterung unverändert bleibt (trainiert werden die zusätzlichen Verbindungen).

² Inwiefern sich das Verschieben der Grenze auf die Evaluierungsquote q_n auswirkt, wird u. a. im folgenden Kapitel (Abbildung 9.5) untersucht.

Evaluierungsquote für verschiedene Maßnahmenklassen

Ergebnis: Alle
Maßnahmenklas-
sen sind relevant

Von besonderem Interesse ist die Bestätigung der Bedeutung der verschiedenen Maßnahmenklassen für die Evaluierungsquote. Mit der manuellen Untersuchung (vgl. Tabelle 5.2) wurde festgestellt, dass mit keiner Maßnahmenklasse für sich ein Optimum erreicht werden kann. Die in Abbildung 9.4 dargestellten Ergebnisse einer Evaluierung der vier Maßnahmenklassen AP, M-DISS, ET und ETRN bestätigen das³. Zusätzlich wurde die Evaluierungsquote für den Zusammenschluss aller Einzelmodelle (gekennzeichnet mit „*ALLE*“) ermittelt. Wie zu erwarten, liegt diese weit über den Ergebnissen der Einzelmodelle und mit 93,1 % nah an der später in der manuellen Evaluierung ermittelten Trefferquote von 94 %. Weiterhin sichtbar ist eine gewisse Varianz der Ergebnisse, die jedoch sehr gering bleibt (vgl. Anhang C.2.2 für eine detaillierte Untersuchung zur Varianz).

Abhängigkeit der Evaluierungsquote von der Fallzahl

Lernende Systeme setzen die Existenz von Daten voraus. Wie viele Datensätze jedoch notwendig sind und welche Leistungsfähigkeit mit welchen Datenmengen erwartet werden kann, ist analytisch nur schwer bestimmbar und muss daher empirisch bestimmt werden⁴. Abbildung 9.5 zeigt die Evaluierungsquote q_n in Abhängigkeit von der Trainingsfallzahl. Für die Evaluierung wurden wie in Abbildung 9.3 skizziert zum einen die direkt auf die Trainingsmenge folgenden 5500 Reparaturfälle verwendet (grüner Graph) und zum anderen die letzten 5500 Fälle der Fallbasis (blauer Graph). Die Ergebnisse verdeutlichen den Anpassungseffekt der

Abbildung 9.3.: Partitionierung der Falldaten

Anpassungsfähigkeit

Diagnostik-Modelle. Während für die Evaluierung jeweils mit den direkt nachfolgenden Fällen bereits nach 10.000 Fällen eine Konvergenz nahe dem Optimum erreicht wird, gilt dies für die Evaluierung mit den letzten 5.500 Falldaten erst mit einem Trainingsmengenende kurz vor Beginn der Evaluierungsmenge. Ein weiterer Beleg für die Anpassungsfähigkeit der Modelle ist der Knick bei ca. 40.000 – 50.000 Fällen. Dieser ist auf die Einführung des Golfs, Modelljahr 2007, zurückzuführen.

³ Eine detaillierte Beschreibung der Maßnahmenklassen AP, M-DISS, ET und ETRN ist im Anhang B.2 zu finden.

⁴ Für die maximale zu erwartende Symptom- und Maßnahmenmenge ist jedoch eine analytische Bestimmung möglich (vgl. Anhang C.2.3).

Abbildung 9.4.: Evaluierungsquote q_n für verschiedene Maßnahmenklassen und Trainingsepochen

Abbildung 9.5.: Abhängigkeit der Evaluierungsquote q_n von der steigenden Fallzahl $\#R_{FB}$ (alle Maßnahmenklassen)

Wie auch Abbildung C.4 zeigt, kamen mit den neuen Fahrzeugvarianten auch neue Reparaturmaßnahmen hinzu (stärkerer Anstieg bei ca. 50.000), die noch nicht im Modell abgebildet waren. Bereits nach 10.000 weiteren Fällen sind diese wieder adaptiert. Für die stark reduzierten Falldaten, die für das Projekt zur Verfügung standen, entsprechen 10.000 Fälle ca. 5 Monaten. Die im Data Warehouse vorhandene Fallzahl für diesen Zeitraum und des Typs Golf lag ca. um den Faktor 7 höher, so dass bereits drei bis vier Wochen für gute Ergebnisse ausreichen könnten.

9.3.5. Evaluierung des Multi-Layer-Perceptron als Diagnostik-Modell

Das bisher evaluierte einschichtige Diagnostik-Modell Π_{neuro} kann den Fahrzeugkontext aufgrund der geringen modellierbaren Komplexität nicht mit einbeziehen. Eine höhere Komplexität entsteht durch die statistische Abhängigkeit zwischen Symptomen s_i und Kontextinformationen c_j mit $s_i \perp c_j$. Ein Lösungsansatz, der in Abschnitt 7.3 vorgestellt wurde, ist die Verwendung eines Multi-Layer-Perceptrons (MLPs) als Diagnostik-Modell. Es wurde nachgewiesen, dass ein MLP, genügend Neuronen in der verdeckten Schicht vorausgesetzt, jede beliebige Funktion und damit auch jede beliebige Komplexität abbilden kann. Damit wird die Berücksichtigung des Kontexts möglich. Ein solches MLP-Diagnostik-Modell Π_{MLP} wurde implementiert und mit verschiedenen Parametern hinsichtlich der Anzahl an verdeckten Neuronen n_H pro Ausgangsneuron v_{k_i} und der Anzahl an Trainingsperioden evaluiert. Abbildung 9.6 zeigt dabei das Optimum hinsichtlich der Trainingsperiodenzahl jeweils für $n_{H2} = \sqrt{\#S_{k_i} + \#C_{k_i}}$ und $n_{H4} = \sqrt[4]{\#S_{k_i} + \#C_{k_i}}$ im Vergleich zum Referenzwert. Dabei ist $\#S_{k_i}$ die Anzahl an Symptom-Eingangsneuronen und $\#C_{k_i}$ die Anzahl an Kontext-Eingangsneuronen für das Neuron v_{k_i} .

Die Ergebnisse zeigen, dass die Evaluierungsquote q_n mit dem Hinzufügen von verdeckten Neuronen in Feedforward-Vernetzung mit steigender Anzahl derer ($n_{H4} < n_{H2}$) sinkt. Für Variante 1 mit n_{H4} verdeckten Neuronen wurde das Optimum bei 10 Trainingsperioden mit $q_n = 88,3\%$ ($-4,8\%$) und für Variante 2 (n_{H2} verdeckten Neuronen) bei 25 Trainingsperioden und $q_n = 85,7\%$ ($-7,4\%$) festgestellt. Der Grund hierfür liegt in der stark steigenden darstellbaren Komplexität, die für alle Eingangsneuronen (Symptom- wie Kontextneuronen) gilt. Es werden damit auch Abhängigkeiten zwischen Symptomen wie auch Abhängigkeiten zwischen Kontextinformationen abgebildet. In Kombination mit einem hochdimensionalen Merkmalsraum bei gleichzeitig geringer Fallzahl pro Merkmal führt dies zwangsläufig zu einer starken Überanpassung und damit geringer Generalisierung. Ein treffendes Zitat aus der Anwendungsdomäne des Text Mining, für die ebenfalls die Charakteristik der hohen

Erhöhte Komplexität durch Kontextinformationen

Pauschale Komplexitätserhöhung nicht zielführend

Merkmalsräume und geringer Fallzahl pro Merkmal gilt, lautet:

The simplest kind of a neural network is a perceptron. It has only two layers - the input and the output nodes such network is equivalent to a linear classifier. More complex networks contain one or more hidden layers between the input and output layers. However, the experiments have shown very small - or no - improvement of nonlinear networks over their linear counterparts in the text categorization task.

Schultze, Hull and Pederson 1995; Wiener 1995 [FS06, S.75]

Das Zitat kann vor dem Hintergrund der Domäne des Text Mining so interpretiert werden, dass das „ziellose“ Hinzufügen einer verdeckten Schicht keinen Erfolg verspricht.

9.3.6. Evaluierung des erweiterten neuronalen Diagnostik-Modells

Aufgrund der schlechten Ergebnisse des MLP-Diagnostik-Modells Π_{MLP} , wurde mit dem Komplexitätserweiterten Diagnostik-Modell eine gezielte Erweiterung mit verdeckten Neuronen vorgenommen, die die speziellen Eigenschaften, insbesondere Dualität und Sparse Data, der zugrunde liegenden Daten berücksichtigt. Ausgehend vom trainierten einschichtigen Diagnostik-Modell Π_{neuro} wurde jedes Symptom-Neuron v_{s_i} in ein verdecktes Neuron konvertiert. Für jedes konvertierte Neuron wurde anschließend eine neue Eingabeschicht, bestehend aus dem Symptom selbst und dem zugehörigen Kontext, hinzugefügt. Lediglich die neue Eingabeschicht wird anschließend nachtrainiert. Durch die besondere Konstruktionsart werden zum einen nur Abhängigkeiten zwischen Symptomen s_i und Kontextinformationen c_j abgebildet und gleichzeitig die Anforderungen aus den Eigenschaften Sparse Data und hochdimensionaler Merkmalsraum erfüllt. Die Ergebnisse sind in Abbildung 9.7 dargestellt. Verglichen wird dabei die Evaluierungsquote q_n des Komplexitätserweiterten Diagnostik-Modells Π_{erw} mit dem einschichtigen Modell Π_{neuro} , für das die Kontextinformationen wie Symptome in die Eingabeschicht hinzugefügt wurden. Als Referenz ist nochmals der Median aus der Streuungsanalyse dargestellt.

Gezielte Komplexitätserweiterung

Die Ergebnisse zeigen, dass durch die spezielle Konstruktionsmethode für alle Maßnahmenklassen, mit Ausnahme von DISS-Maßnahmen, eine zum Teil deutliche Steigerung erreicht werden konnte. Die größte Steigerung wurde dabei für Ersatzteile erreicht, was nicht weiter verwundert, da gerade diese von Hardwareversionen abhängen. Die Gesamtevaluierungsquote konnte nochmals um 3,3 % auf 96,4 % gesteigert werden. Im Gegensatz dazu sinkt die Evaluierungsquote des einschichtigen Modells, sobald dort Kontextinformationen hinzugefügt werden. Der Grund hierfür wurde bereits in Abschnitt 7.3.1 erörtert: Durch die mangelnde

Ergebnis: Deutliche Steigerung der Evaluierungsquote

Abbildung 9.6.: Evaluierungsquote q_n der MLP-Diagnostik-Modelle: **Ref.:** Referenzwert aus Streuungsanalyse; **4√MLP:** Evaluierung mit $n_{H4} = \sqrt[4]{\#S_{k_i} + \#C_{k_i}}$ verdeckter Neuronen; **2√MLP:** Evaluierung mit $n_{H2} = \sqrt{\#S_{k_i} + \#C_{k_i}}$ verdeckter Neuronen

Abbildung 9.7.: Evaluierungsquote q_n der komplexitätserweiterten Diagnostik-Modelle: **Ref.:** Referenzwert aus Streuungsanalyse (ohne Kontext); **$\Pi_{erw.}$:** Komplexitätserweitertes Modell; **$\Pi_{neuro.}$:** Einsichtiges Modell inkl. Kontextinformationen

Abbildbarkeit der Abhängigkeit des Kontexts c_j vom Symptom s_i , die je nach Einflussart vereinfacht die Formen $k_i = s_i \wedge c_j$, $k_i = s_i$, jedoch auch $k_i = c_j$ annehmen kann, verursacht gerade letztere, dass unter Umständen Diagnose-Hypothesen allein aufgrund des Vorhandenseins einer bestimmten Steuergeräteversion erzeugt werden, ohne dass ein Symptom hierfür vorhanden ist.

Ein noch größerer Gewinn des erweiterten Modells wäre mit der Erhöhung der Modellvarianten in der Fallbasis zu erwarten, da dann eine Diversifizierung der jeweiligen Software- und Hardwareversion zunehmend von Bedeutung ist.

9.3.7. Verlernen alter Reparaturfälle

Jüngere Reparaturfälle sind relevanter

Die Analyse des Einflusses der Trainingsmenge auf die Evaluierungsquote hat gezeigt, dass die Evaluierungsquote mit Falldaten, die zeitlich direkt nach den Trainingsdaten liegen, höher ist, als die mit Falldaten späteren Datums. Im Umkehrschluss sind Reparaturfälle aus jüngerer Vergangenheit wichtiger als Reparaturfälle, die weiter zurück liegen. Die Gründe hierfür liegen z.B. in durch Verbesserungs- oder Fehlerabstellmaßnahmen veränderte Reparaturmaßnahmenwahrscheinlichkeiten, auf die eine Anpassung erfolgen muss.

Abbildung 9.8.: Modifiziertes Lernvorgehen mit Verlernen alter Reparaturfälle; links mit disjunkter, rechts mit überlappender Partitionierung der Falldaten

Ergebnis: Verlernen alter Fälle sinnvoll

Mit dem in Kapitel 7.5 vorgestellten Verfahren wird das Diagnostik-Modell zunächst basierend auf allen Falldaten erstellt und vortrainiert und anschließend mit in nach Datum aufsteigend sortierten Blöcken partitionierter Falldaten nachtrainiert. Wie in Abbildung 9.8 dargestellt wurde im Rahmen der Evaluierung zum einen die Partitionierung in disjunkte und zum anderen in überlappende Blöcke untersucht. Abbildung

Abbildung 9.9.: Einfluss des Verlernens alter Reparaturfälle auf die Evaluierungsquote q_n in Abhängigkeit der Maßnahmenklasse und disjunkter Partitionierung (Ref. = Referenzwert aus Streuungsanalyse)

9.9 zeigt den damit erreichten Leistungsgewinn pro Maßnahmenklasse und Partitionierungsvariante. Mit Ausnahme von Werkstattwahrnehmungen (DISS) konnte für alle Maßnahmenklassen eine Verbesserung von 1,4 % – 3,4 % erreicht werden. Die Gesamtverbesserung beträgt 2,0 %.

9.3.8. Evaluierung der Merkmals-Reduktion

Mit wachsender Anzahl von Falldatensätzen werden Lernverfahren mit mehreren Trainingsepochen zunehmend ressourcenaufwändig hinsichtlich Rechen- und Speicherbedarf. Eine vorhergehende Merkmals-Reduktion reduziert dabei linear die Anzahl an Verbindungsgewichten eines neuronalen Netzes und damit des gesamten Ressourcenbedarfs. Das in Kapitel 7.6 vorgestellte Verfahren verwendet hierfür die statistische Signifikanz zwischen einem Symptom s_i und einer Reparaturmaßnahme k_i als Indikator für die Relevanz des Symptoms s_i . Abbildung 9.10 zeigt die Verteilung der Signifikanzwerte in einem Histogramm.

Das Histogramm zeigt, dass zwar im Vergleich zu möglichen Symptom-Maßnahmenkombinationen (ca. $1 \cdot 10^{-7}$) nur wenige Signifikanzen existieren, diese dann allerdings sehr deutlich ausfallen (d. h. $p_{k_j s_i}$ klein). Über 22 % aller p-Werte sind kleiner als 0,005 und über 49 % kleiner als 0,05⁵. Mit dem daraus resultierenden Graph G können damit lange Wege

Verteilung der Signifikanzwerte

⁵ Ein p-Wert $< 0,05$ wird allgemein als hochsignifikant angesehen.

Verteilung der Distanzwerte	<p>über viele Knoten hinweg entstehen, bis die Distanzgrenze d_{th} erreicht wird. Aus diesem Grund muss die zulässige Tiefe δ für die Berechnung der Distanzen beschränkt werden, damit später nicht zu viele Symptome pro Maßnahme ausgewählt werden. In Abbildung 9.11 ist die Häufigkeitsverteilung der Distanzwerte für verschiedene Tiefen als Histogramm dargestellt. Die Häufigkeitsachse musste dabei logarithmisch skaliert werden, um einen direkten Vergleich mit den Signifikanzhäufigkeiten zu ermöglichen. Die Abbildung erklärt, warum gerade für die Distanzwerte d_{th_0} der Tiefe $\delta = 0$ andere Werte als für die Distanzwerte d_{th_1} der Tiefe $\delta = 1$ gewählt werden sollten. Während für Signifikanzwerte (= Distanzwerte der Tiefe 0) 22 % der Werte kleiner als 0,005 sind und damit 14.471 Verbindungen w_{ij} erfasst werden, sind es für Distanzwerte der Tiefe 1 bereits 47 % und 4.936.923 Verbindungen.</p>
Parameterbestimmung	<p>Die Zahlen verdeutlichen, dass die Festlegung geeigneter Distanzgrenzen d_{th_i} sowie der notwendigen Tiefe δ sorgfältig erfolgen muss. Abbildung 9.12 zeigt die Auswirkung auf die Evaluierungsquote q_n und die Anzahl der Verbindungen $\#w_{ij} > 0$ für verschiedene Distanzgrenzen d_{th_i} und Tiefen δ am Beispiel von Ersatzteilrumpfnummern (ETRN). Dabei werden drei Varianten dargestellt. In Variante eins werden lediglich Signifikanzwerte (Tiefe $\delta = 0$) zur Merkmals-Reduktion verwendet. Entsprechend gering ist die Zahl der Verbindungsgewichte. Aber auch die Leistungsfähigkeit ist deutlich geringer, steigt aber kontinuierlich mit zunehmender Verbindungszahl. In Variante zwei wird an genau dem letzten Punkt der Variante eins mit $d_{th_0} = 1,0$ fortgesetzt und die Tiefe auf $\delta = 1$ erhöht. Mit zunehmender Distanzgrenze d_{th_1} steigt die Verbindungszahl weiter. Auch die Evaluierungsquote q_n steigt zunächst, bis ein Optimum bei ca. $d_{th_1} \approx 1 \cdot 10^{-10}$ erreicht wird. Die einzelnen blauen Punkte, die die einzelnen ungemittelten Evaluierungsergebnisse der Reihe wiedergeben, zeigen hier aber auch, dass von Trainingsvorgang zu Trainingsvorgang eine deutliche Streuung von bis zu $\pm 1\%$ zu beobachten ist. In Variante drei wurden letztlich beide Distanzgrenzen d_{th_0} und d_{th_1} gleichgesetzt. Wie die Histogramme in den Abbildungen 9.11 und 9.10 zeigen, führt das nach der Merkmals-Reduktion zu einer geringen Anzahl direkt signifikanter Symptome, dafür jedoch einer höheren Anzahl indirekt hochsignifikanter Symptome. Die Evaluierungsquoten sind insbesondere im Verhältnis zur Verbindungszahl schlechter.</p>
Ergebnis: Deutliche Reduktion der Merkmale	<p>Gute Ergebnisse haben sich somit für eine Tiefe von $\delta = 1$ bei unterschiedlichen Distanzgrenzen ergeben. Die Distanzgrenze d_{th_0} sollte mit ≈ 1 gewählt werden, das heißt, es werden alle direkt signifikanten Symptome berücksichtigt. Die Distanzgrenze d_{th_1} sollte mit $\approx 1 \cdot 10^{-10}$ gewählt werden, so dass fast nochmals so viele indirekt hochsignifikante Symptome hinzu kommen. Mit diesem Parameterset wird eine stabile Merkmals-Reduktion erreicht, deren Ergebnisse auch durch steigende Verbindungszahlen nicht weiter verbessert werden können. Insgesamt konnte die Anzahl der Verbindungen für alle Maßnahmenklassen zusammen von ursprünglich $\approx 1 \cdot 10^8$ auf $\approx 1,7 \cdot 10^6$ (-98,3 %) reduziert werden. Die</p>

Abbildung 9.10.: Verteilung und akkumulierte Häufigkeit der Signifikanzwerte für Ersatzteilrumpfnrnummern (ETRN) mit einer Klassenbreite von $\Delta d = 0,005$. Über 22 % aller vorhandenen Signifikanzen (64.800 von 100.820.000 möglichen) haben einen p-Wert $< 0,005$

Abbildung 9.11.: Verteilung und akkumulierte Häufigkeit der Distanzwerte für Ersatzteilrumpfnrnummern (ETRN) mit einer Klassenbreite von $\Delta d = 0,005$ und verschiedenen Tiefen δ_{th} . Über 47 % aller vorhandenen Distanzen (4.936.923 von 100.820.000 möglichen) haben einen Wert $d_{k_j s_i} < 0,005$

Abbildung 9.12.: Zusammenhang zwischen Evaluierungsquote q_n und Anzahl der Verbindungen $\#w_{ij} > 0$ für verschiedene Distanzgrenzen d_{thi} (Wert an den Markierungen) und Tiefen δ der Merkmals-Reduktion am Beispiel von Ersatzteilrumpfnummern (ETRN); **blau:** Tiefe $\delta_{th} = 0$, $d_{thi} = d_{th0}$; **grün:** Tiefe $\delta_{th} = 1$, $d_{th0} = 0,99$, $d_{thi} = d_{th1}$; **rot:** Tiefe $\delta_{th} = 2$, $d_{thi} = d_{th0} = d_{th1}$; **Punkte:** Einzelne Evaluierungsquoten q_n ; **durchgezogene Graphen:** gleitender Durchschnitt

Merkmals-Reduktion hat die Trainingszeiten um ein Vielfaches verkürzt und damit aufwändige Untersuchungen wie die Lernparameter-Iteration (Abbildung C.2) erst ermöglicht.

9.3.9. Effektivität von Pruning

Die Struktur des neuronalen Diagnostik-Modells ist hinsichtlich der Kompaktheit, d.h. der Anzahl der Verbindungsgewichte und Neuronen nicht optimal. Zwar kann die Kompaktheit bedingt durch eine statistische Merkmals-Reduktion erhöht werden. Welche Verbindungsgewichte letztlich jedoch notwendig sind, ist erst nach dem Training bekannt. Mit der späteren Erweiterung des Modells ist Kompaktheit jedoch unbedingt gefordert.

Reduktion der
Verbindungszahl

Hierfür wurde ein spezialisiertes Magnitude-Based-Pruning-Verfahren (MBP) entwickelt. Der Erfolg in der Reduktion der Verbindungsgewichte ist beispielhaft für den Maßnahmentyp AP (Arbeitspositionen) in

Abbildung 9.13.: Reduktion der Anzahl der Verbindungen durch Pruning für Arbeitspositionen (AP) und $Pr_{th} = 0,2$: Die Balken zeigen die Anzahl von Netzen pro Verbindungsanzahl-Klasse, jeweils mit und ohne Pruning, die Graphen die dazu gehörende akkumulierte Gesamtanzahl der Verbindungen

Abbildung 9.13 dargestellt. Die roten Balken stellen die Häufigkeit der Verbindungsgewichte pro Maßnahmenneuron und die rote Kurve deren Akkumulation zur Gesamtanzahl dar. Die blauen Balken zeigen die Häufigkeitsverteilung der Verbindungsgewichte nach dem Pruning der Netze. Während durch die Merkmals-Reduktion die Zahl der Verbindungen für Arbeitspositionen (AP) bereits von ursprünglich $46,1 \cdot 10^6$ auf $0,5 \cdot 10^6$ deutlich reduziert werden konnte, wird die Zahl der Verbindungen nach dem Pruning nochmals auf 21.849 ($-95,6\%$) reduziert. Die Verteilung zeigt, dass die meisten geprunten Netze 20 und weniger Verbindungen und damit relevante Symptome⁶ aufweisen.

Wie sich Pruning letztlich auf die Evaluierungsquote q_n auswirkt, ist in Abbildung 9.14 dargestellt. Gegenübergestellt sind hier die Ergebnisse für die Maßnahmenklassen ET, ETRN, AP und DISS sowie dem Gesamtmodell (ALL) in Abhängigkeit von der Pruning-Grenze Pr_{th} und der Anzahl der Verbindungen. Der Punkt *Ref.* markiert den Referenzwert aus dem Mittel der Streuungsanalyse. Interessant ist hierbei, dass sich die

Einfluss auf die
Evaluierungsquote

⁶ Für das einschichtige neuronale Diagnostik-Modell ist jedes Symptom mit genau einer korrespondierenden Verbindung verknüpft.

Abbildung 9.14.: Einfluss der Pruning-Grenze Pr_{th} (Wert über Markierungen) auf die Evaluierungsquote q_n und die Anzahl der Verbindungen (**Ref.:** Referenzwert aus Streuungsanalyse)

Evaluierungsquote bis zu einer bestimmten Pruning-Grenze nur marginal verändert, bevor er dann merklich sinkt. Der Schwellenwert liegt bei ca. $Pr_{th} = 0,2$. Für diesen Wert wird die Anzahl der Verbindungen bereits auf 1,9 % gegenüber dem ungeprunten Netz reduziert. Das Ergebnis sind schlanke, leistungsfähige Modelle.

9.3.10. Zusammenfassung der Evaluierung mit Falldaten

Die Evaluierung mit Falldaten hatte zum Ziel, die in der Konzeptevaluierung noch grob abgeschätzte Leistungsfähigkeit derart zu quantifizieren, dass daraus die für einen Praxiseinsatz tatsächliche erwartbare Leistungsfähigkeit abgeschätzt werden kann. Hierzu wurde zunächst ein normiertes Qualitätsmaß, die Evaluierungsquote q_n , ein Verfahren zur automatisierten Berechnung dessen sowie Partitionierungsmöglichkeiten der zur Verfügung stehenden Daten entwickelt.

Das Evaluierungsverfahren wie auch die Verwendung von realen Reparaturfalldaten ließen eine sehr detaillierte Abschätzung der in der Praxis erwartbaren Leistungsfähigkeit zu. Bereits das einschichtige Diagnostik-Modell erreicht eine Evaluierungsquote von 93,1 %, die durch verschiedene Optimierungen, speziell das Verlernen von alten Reparaturfällen, noch

auf 95,1 % gesteigert werden konnte. Das komplexitätserweiterte Diagnostik-Modell ermöglichte die Berücksichtigung des Fahrzeugkontexts und erreichte eine Steigerung der Evaluierungsquote auf 96,4 %, trotz der relativ kurzen Zeitspanne und der geringen Fahrzeugmodellvarianz der zur Verfügung stehenden Falldaten und den damit nur gering auftretenden Kontextänderungen. Bestätigt hat sich auch die als geringer zu erwartende Leistungsfähigkeit des MLP-Modells mit $-4,8\%$ auf 88,3 %, die damit die spezielle Konstruktionsmethode des komplexitätserweiterten Modells noch bestärkt.

<i>Evaluation</i>	<i>Ergebnis</i>
Trainingsparameter ^a	Optimum bei $n10 - d0 - \eta0,5 - I0,01$: n = Trainingsperioden, $d = d_{max}$ (max. tol. Fehler), η = Lernrate, I = Initialisierungsintervall
Evaluierungsquote für verschiedene Maßnahmenklassen	Maximum = 93,1 %, für 5 Trainingsperioden und alle Maßnahmenklassen AP, DISS, ET und ETRN
Varianz der Ergebnisse ^b	+1,7 % bis $-1,3\%$
Abhängigkeit von der Fallzahl	Ab 10.000 Fälle wird $q_n > 88\%$ erreicht. Symptom- und Maßnahmenzahl noch bei 70.000 Fällen steigend
MLP-Diagnostik-Modell	$-4,8\%$ auf 88,3 %; für alle Maßnahmenklassen eine Verschlechterung
Erweitertes neuronales Diagnostik-Modell	+3,3 % auf 96,4 %; bis auf M-DISS eine deutliche Verbesserung für alle Maßnahmenklassen, z. B. für ET: +15,7 %
Optimierungen	
Verlernen alter Reparaturfälle	+2 % auf 95,1 %
Merkmals-Reduktion	Reduktion von $\approx 1 \cdot 10^8$ auf $\approx 1,7 \cdot 10^6$ ($-98,3\%$) Verbindungen
Pruning	Reduktion von $\approx 1,7 \cdot 10^6$ auf ≈ 32.004 ($-98,1\%$) Verbindungen bei optimaler Pruninggrenze $Pr_{th} = 0,2$

^a Im Anhang C.2.1 gelistet.

^b Im Anhang C.2.2 gelistet.

Tabelle 9.4.: Zusammenfassung der Ergebnisse der Evaluierung mit Falldaten

Durch die entwickelten Verfahren zur Merkmals-Reduktion und Pruning wurden zum einen die aufwändigen Untersuchungen erst in akzeptabler Rechenzeit ermöglicht und zum anderen bleiben die letztlich erzeugten Diagnostik-Modelle damit kompakt und effizient. So konnte die Merkmals-Reduktion die Anzahl der Verbindungen von ursprünglich $1 \cdot 10^8$ auf $1,7 \cdot 10^6$ ($-98,3\%$) senken. Das Pruning nach dem Training

der Modelle erreichte nochmals eine Reduktion auf ca. 32.004 (−98,1 %) Verbindungen. Diese wurden zwar durch die Komplexitätserweiterung wieder erhöht, jedoch sind die resultierenden Modelle mit ca. 125.814 (−98,1 %) Verbindungen immer noch sehr kompakt. Die einzelnen Evaluierungsschritte und deren Ergebnisse sind zusammengefasst in Tabelle 9.4 dargestellt.

Bestätigung der aufgestellten Thesen

Neben der Evaluierung der Leistungsfähigkeit konnten die im Theorierteil aufgestellten Thesen durch die Untersuchungen bestätigt werden. Dazu zählt insbesondere die Auswahl der notwendigen Maßnahmenklassen, angenommene statistische Verteilungen und deren Auswirkungen, Vor- und Nachteile verschiedener neuronaler Modelle, insbesondere die schlechte Eignung von Standard-MLP-Modellen und die Problematik der vorhergehenden Bestimmung unrelevanter Verbindungen.

Bedeutung der automatisierten Evaluation für die Entwicklung

Die Möglichkeit der automatisierten Evaluation hat sich in der Praxis als sehr hilfreich herausgestellt. Insbesondere bei Verfahren mit neuronalen Netzen gibt es, wie bereits beschrieben, eine große Anzahl an möglichen Architekturen sowie an verschiedenen Parametern. Erst durch den Automatismus konnte die Leistungsfähigkeit verschiedener Kombinationen von Architekturen und Parametern in großem Umfang untersucht werden. Ohne eine automatisierte Evaluation wäre dies nur mit enormem Aufwand möglich gewesen. Das Vorgehen mit einer dreistufigen entwicklungsbegleitenden Evaluierung war im Projekt von entscheidender Bedeutung und hat die Qualität wie auch die Entwicklungszeit stark optimiert.

9.4. Praxiserprobung

Die automatisierte Evaluierung ermöglichte eine effiziente und gute Abschätzung der Gesamtleistungsfähigkeit, hat jedoch auch Nachteile. Dazu zählt, dass die Leistungsfähigkeit (repräsentiert durch die Evaluierungsquote q_n) verfahrensbedingt über dem Durchschnitt aller Fälle der Evaluierungsfallbasis R_{eval} berechnet wird. Analog dazu optimieren die verwendeten Lernverfahren, Varianten des Backpropagation-Algorithmus, die Stärke der Verbindungen ebenfalls mittels der Minimierung einer Fehlerfunktion $E(G, R_{train})$ über alle Fälle der Trainingsmenge R_{train} (vgl. (7.17)). Als Folge werden einzelne ungünstige oder fehlerhafte Diagnose-Hypothesen h_D mit geringer Häufigkeit $P(h_D)$ wenig berücksichtigt. Ein weiterer Nachteil der automatisierten Evaluierung ist die Überzeugung von Projektträgern von der Leistungsfähigkeit des Systems. Gemeinhin ist dort eine Skepsis gegenüber statistischen Methoden, Auswertungen und Ergebnissen zu beobachten.

Um die Leistungsfähigkeit der Diagnostik-Modelle auch für Fehlerdiagnosen geringer Häufigkeit hinreichend detailliert analysieren zu können,

wurden im Rahmen der Praxiserprobung drei weitere Evaluierungsschritte durchgeführt:

1. Manuelle Evaluierung der Leistungsfähigkeit mit 100 Falldaten
2. Untersuchung exemplarischer, relevanter Fälle in Zusammenarbeit mit der Qualitätssicherung der Volkswagen AG
3. Praxiserprobung am Fahrzeug

Die Ergebnisse der einzelnen Schritte werden nachfolgend vorgestellt und diskutiert.

9.4.1. Manuelle Evaluierung der Leistungsfähigkeit

Die Skepsis gegenüber einer automatisierten Evaluierung ist neben dem geringen Einfluss einzelner Fehldiagnosen h_D auf das Gesamtergebnis q_n insofern gerechtfertigt, da für deren Realisierung Annahmen getroffen und Kompromisse eingegangen werden mussten. Dazu zählte z.B. die Definition des Maximums der Evaluierungsquote q , um eine Normierung q_n zu ermöglichen, oder der Kompromiss zur Qualitätsrate anstatt einer absoluten Trefferquote aufgrund fehlender Komponentenzuordnungen für die jeweiligen Maßnahmenklassen AP, ET, ETRN und DISS. Damit hat die manuelle Evaluierung neben der Analyse von Einzelfällen auch das Ziel, eine absolute Trefferquote zu ermitteln. Evaluiert wurde das in Abschnitt 9.3.4, Abbildung 9.4, Parameterset „*ALLE-5*“ vorgestellte Modell, das eine Evaluationsquote q_n von 93,1 % erreichte. Als Evaluierungsdatensatz dienten 100 zufällig ausgewählte Fälle der Evaluierungsmenge R_{Eval} , die auch zur Charakterisierung der Falldaten in Kapitel 5.1, Tabelle 5.2 dienten. Wie dort bereits beschrieben, konnten nicht alle Datensätze ausgewertet werden. Der häufigste Grund hierfür waren eine geringe Nachvollziehbarkeit der Reparaturaufgabe und der durchgeführten Reparaturmaßnahmen sowie deren Erfolg (vgl. Abbildung 5.1).

Zur Ermittlung der Trefferquote wurden für jeden einzelnen auswertbaren Datensatz R_i gemäß Abbildung 9.2 zuerst die Diagnose-Hypothesen $\{h_D\}$ mittels des gewählten Diagnostik-Modells auf Basis der Symptome $S_R \in R_i$ berechnet. Anschließend wurde jede Diagnose-Hypothese h_D mit den tatsächlich im Reparaturfall R_i durchgeführten Maßnahmen verglichen und ausgewertet. Abbildung 9.15 zeigt einen Screenshot des MS Access-Formulars, das zur manuellen Auswertung verwendet wurde. Im obersten Abschnitt sind allgemeine Metadaten des Reparaturfalls, darunter eine zum Teil vorhandene textuelle Beanstandung, gefolgt von der Menge der Ereignisspeichereinträge (DTC) und den Kundenbeanstandungen (S-DISS), dargestellt. Unter Maßnahmen sind die abgerechneten Ersatzteile (ET) und Arbeitspositionen (AP) sowie vorhandene Werkstattwahrnehmungen (M-DISS) aufgelistet. Im untersten Abschnitt sind die Diagnose-Hypothesen des zu evaluierenden Diagnostik-Modells inklusive der Vertrauenswerte $v \in h_D$ und einer Kurzbeschreibung zu sehen.

Ziel: Absolute Trefferquote und Analyse von Einzelfällen

Durchführung

Ein Beispiel

Temperatursensor
VAG 06A919501

Reparaturfall_ 52 REP_ANNAH 23.06.2006 MDJ 2005 VMODEL 1TK10A1 GFS_ZIELFU 0 GFS_ZIELFU 0 ANTRAG_ID DEU14327324108552006063000
PROT_PROT 1929903 MKB BCA MOD_JAHR 2005 VERKAUFS_1 1TK10A1 REP_ERFOL 0 REP_ERFOL 0 Fahrgestellnr WVVZZZ1K000081
PROT_ANZA 1

Beanstandungstext
BEAN_KUNDENTEXT BEAN_WERKSTATTEXT BEAN_TPL_TITEL

Datensatz: 14 1 von 1 Kein Filter Suchen

DTC-Symptome
ID Re SYMPTOM SY STG_Bezeichnung Fehlerbeschreibung Faultype1_Sum Bezeichn Faultype2b
42153 52 1_0116502_23 DTC Motorelektronik Geber für Kühlmitteltemperatur-G62 Signal zu groß unterer Grenzwert unterschr statisch

Datensatz: 14 1 von 1 Kein Filter Suchen

DISS-Symptome
ID R SYMPTOM SY BEZEICHNUNG

Datensatz: 14 1 von 1 Kein Filter Suchen

Massnahmen
ID R MASSNAHME Massn Comment KOSTEN
165 52 1_01500000 AP GFS/Geführte Funktion 47,57 €
167 52 1_19010100 AP Kühlsystem geprüft 14,20 €
168 52 1_19111900 AP Geber f Kühlmittelstandanzeige a++ 28,40 €
52 52 3_1911 KDNR 1911 0,00 €
166 52 4_06A919501A ET TEMP.GEBER 14,47 € V

Datensatz: 14 1 von 5 Kein Filter Suchen

Diagnose-Hypothesen
Cr MassnahmenId Vertrauenswert Erfolge Massn Kurzbezeichnung Kosten
1 M4N_90316802 0,857086593520965 ET DICHTRING 0,99 €
2 M2_0114131098 0,750533561911311 DISS Motor - Bereich Kühlkreislauf für Wasserkühlung - Temperaturgeber - erhö.Verschleiss / erhö.Verbrauch 0,00 €
3 M1_19781900 0,674866666638929 AP Temperaturgeber aus- u.eingebaut 40,77 €
4 M4_06A919501A 0,584817118766642 X ET TEMP.GEBER 20,26 €
5 M4_06A919501 0,414794495639074 ET TEMP.GEBER 14,72 €
6 M4G_012A8FS0 0,346203369904591 ET KUEHLZUSAT 22,68 €

Abbildung 9.15.: Screenshot des MS Access-Formulars zur manuellen Evaluierung

Das Feld „Erfolgreich“ ist mit einem „X“ markiert, sofern die automatisierte Evaluierung die Diagnose-Hypothese als zutreffend eingestuft hat. Der dort dargestellte Fall zeigt anschaulich ein Beispiel für eine Fehlbewertung durch die automatisierte Evaluierung. Der Defekt war in diesem Fall offensichtlich der Temperaturegeber. Darauf weist bereits die zweite Diagnose-Hypothese mit einer M-DISS zutreffend hin. Jedoch existiert diese M-DISS nicht auch im Beispielfall, so dass sie automatisiert nicht als zutreffend eingestuft werden konnte. In der manuellen Evaluierung wäre dies dagegen der Fall. Analog gilt das für die dritte Diagnose-Hypothese, die ebenfalls auf den Temperaturegeber hinweist. Auch die erste Diagnose-Hypothese weist mit dem für den Wechsel des Gebers notwendigen Dichtring hin. Allerdings handelt es sich hier um ein Normbauteil, das auch an anderen (wenigen) Stellen verbaut wird und darum nicht als Lösungshinweis gewertet wird.

Ergebnis

Das Ergebnis der Bewertung ist in Tabelle 9.5 dargestellt. Das evaluierte Diagnostik-Modell erzeugte dabei in 78 % aller Reparaturfälle eine korrekte Diagnose-Hypothese an erster Stelle. Insgesamt war in 94 % aller Fälle die korrekte Diagnose-Hypothese in der Menge der Vorschläge. Damit übertrifft das Ergebnis sogar die durch die automatisierte Evaluierung ermittelte Evaluierungsquote q_n von 93,1 %.

Position	Trefferquote
Treffer an 1. Stelle	78 %
Treffer an 2. Stelle	6 %
Treffer an 3. Stelle	5 %
Treffer an 4. Stelle oder darüber	5 %
Gesamt	94 %

Tabelle 9.5.: Ergebnis der manuellen Evaluierung

Neben der absoluten Trefferquote wurde erst durch die manuelle Evaluierung ein Verständnis der Daten wie auch eine präzise Falldatencharakterisierung möglich. Die dort auffälligen Einzelfälle, wie z. B. jener aus Tabelle B.4, B.5 und B.6, ermöglichten individuelle Anpassungen solcher Modell-Parameter, die sich zwar wenig auf die Gesamtevaluierungsquote auswirken, dafür jedoch in Einzelfällen starke Verbesserungen bringen. Die manuelle Evaluierung umgeht damit die Schwächen der automatisierten Evaluierung und liefert einen belastbaren Wert für die im Praxiseinsatz zu erwartende Leistungsfähigkeit. Gleichzeitig stellt die manuelle Evaluierung aber auch fest, dass die Abweichungen der Ergebnisse der manuellen zur automatisierten Evaluierung überraschend gering sind und sich letztere nachwievor für weitere Evaluierungen eignet.

9.4.2. Untersuchung exemplarischer, relevanter Fälle

Ein großer Nutzer der aufgezeichneten Reparaturfalldaten ist die Qualitätssicherung. Die Daten stellen durch verschiedenste statistische Auswertungen das „Fenster zur Welt“ dar. Um auch hier die Verantwortlichen von der Leistungsfähigkeit zu überzeugen, wurde ein besonderes Vorgehen vereinbart: Die Qualitätssicherung stellte die Symptome (DTCs) vier besonders relevanter Fälle zur Verfügung, die ihnen aufgrund ihrer Funktion bekannt sind. Die Ergebnisse des Diagnostik-Modells wurden dann gemeinsam mit allen Beteiligten „live“ evaluiert, ohne dass eine wie auch immer geartete „Präparierung“ hätte stattfinden können. Die vier Fälle und das Ergebnis der Evaluierung sind in Tabelle 9.6 dargestellt, wovon ein Fall hier exemplarisch erörtert werden soll (die Detailbeschreibungen der anderen Fälle sind im Anhang C.3 zu finden).

Der im ersten Fall relevante Ereignisspeichereintrag war DTC 895 im Steuergerät Lenkradelektronik mit der Bezeichnung „*GRA/ADR Bedienteil-E45*“. Auffälligkeiten gab es für das Modell Golf. In der Datenbasis existieren mit diesem DTC 1794 Fälle, dabei sind 88 % als sporadisch und 12 % als statisch gekennzeichnet. Als Modell wurde das bereits in der manuellen Evaluierung untersuchte Modell aus Abbildung 9.4, Parameterset „*ALLE-5*“ verwendet. Die Diagnose-Hypothesen des Modells sind in Abbildung 9.16 dargestellt. Eine genauere Betrachtung der

Fall 1: Lenkstockschalter Tempomat

Interpretation der Häufigkeit vs. dem Vertrauenswert

einzelnen Hypothesen zeigt, dass mit Ausnahme von Hypothese vier und fünf, die notwendigen Arbeitspositionen für die Reparatur sind, alle auf dieselbe Komponente, den Lenkstockschalter des Tempomats, verweisen. Dabei soll erwähnt werden, dass die Hypothesen 6 und 7 auf dieselbe Komponente verweisen, diese jedoch durch die KD-Mitarbeiter unterschiedlich kodiert wurden, so dass auch zwei Arbeitspositionen hierfür existieren. Der exemplarische Fall wurde daher einstimmig als Erfolg des Diagnostik-Systems gewertet. Überzeugend war für diesen Fall auch ein Vergleich der Vertrauenswerte v , der Verbindungsgewichte w_{ij} mit den Werten der statistischen Häufigkeit $P(k_i)$. In Abbildung 9.17 sind die Werte absteigend nach statistischer Häufigkeit sortiert, dargestellt. Die Abbildung zeigt zum einen, dass die Vertrauenswerte nicht direkt mit den Häufigkeitswerten korrelieren. Der Grund hierfür ist, dass die Häufigkeit $P(k_j|s_i)$ lediglich ein Symptom s_i betrachtet, das Diagnostik-Modell jedoch alle Symptome S , d. h. das Vorhandensein, aber auch das Nichtvorhandensein jedes einzelnen Symptoms $s \in S$. Es gilt folglich $P(k_j|s_i) \neq P(k_j|S)$. Anschaulich ist so z. B. der Vertrauenswert für die Arbeitsposition „Lenkrad aus- & eingebaut“ deutlich geringer als die des Ersatzteils „Schalter GRA“, denn ersterer wird auch noch häufiger in

Abbildung 9.16.: Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 16- 895

Abbildung 9.17.: Vergleich der Vertrauenswerte v , der Verbindungsgewichte w_{ij} mit den Werten der statistischen Häufigkeit $P(k_i)$ für den Beispielfall Golf, DTC 16-895; a.& e. = aus- und eingebaut

Fällen mit anderen Symptomen $s \neq s_i$ ausgeführt, letzterer weniger. Ein weiteres Beispiel ist das vorletzte Ersatzteil „Schalter GRA“, das zum einen in der Häufigkeit selten ist und zum anderen einen sehr niedrigen Vertrauenswert besitzt. Eine Analyse der Falldaten zeigt, dass für dieses spezielle Ersatzteil in $P(\bar{s}_i|k_j) = 71\%$ der Fälle ohne das Symptom s_i getauscht wurde.

Dieselben Analysen wurden für alle Fälle durchgeführt, wobei hier nicht alle Fälle weiter im Detail erörtert werden sollen (weitere Details sind im Anhang C.3 zu finden). Die Ergebnisse sind in Tabelle 9.6 zusammengefasst. Für alle untersuchten Fälle wurden ähnlich gute Ergebnisse erzielt wie im vorgestellten Fall. Die „Live-Erprobung“ überzeugte nicht nur alle Beteiligten, sondern konnte zumindest exemplarisch zeigen, dass das Diagnostik-System auch für reale relevante Fälle gute Ergebnisse liefert.

Zusammenfassung

9.4.3. Praxiserprobung am Fahrzeug

Als letzter Schritt des dreistufigen Evaluierungskonzepts erfolgt eine Praxiserprobung am Fahrzeug. Sie hat zum einen zum Ziel, die Praxistauglichkeit von Neucus weiter zu belegen und ferner die Leistungsfähigkeit der GFS mit der von Neucus praktisch zu vergleichen. Dies war mit

Ziel: Vergleich zwischen GFS und Neucus, Praxistauglichkeit

<i>Ereignisspeichereintrag</i>	<i>Ergebnis</i>
Golf – 16 – 895 – Lenkradelektronik: GRA/ADR Bedienteil-E45	✓
Tiguan – 44 – 446 – Lenkhilfe: Funktionseinschränkung durch Unterspannung	✓
Tiguan – 09 – 2403 – Elektronische Zentralelektrik: Kl15 Innenraum	✓
Golf – 01 – 16683 – Motorsteuergerät: Ladedruckregelung, Regelgrenze unterschritten	✓

Tabelle 9.6.: Ergebnis der Untersuchung exemplarischer Fälle; **Format:** [Fahrzeugmodell]-[SG-Adresse]-[DTC]-[SG-Bezeichnung]:[DTC-Bezeichnung]; Zeichen ✓: Erfolgreich

den zur Verfügung stehenden Reparaturfalldaten nicht möglich, da hier entscheidende SG-Informationen wie Messwertblöcke etc., die von einem GFS-Programm benötigt werden, fehlen. Die Praxiserprobung wurde an einem VW Golf Modelljahr 2006 in einer Werkstatt der Volkswagen AG durchgeführt. Verwendet wurde hierfür ein Kundendiensttester VAS5051/B mit GFS und das Diagnostik-System Neucus auf einem Laptop mit der Diagnostik-Schnittstelle VAS5053. Als Diagnostik-Modell wurde auch hier das aus Abbildung 9.4, Parameterset „*ALLE-5*“ verwendet.

Ein direkter Vergleich zwischen den Diagnostik-Hypothesen von Neucus und der GFS ist nur über den Umweg von Komponentenzuordnungen für Prüfanweisungen möglich (vgl. Abschnitt 9.3.1). Es werden folglich alle nachfolgenden Prüfanweisungen mit einer entsprechenden Komponentenzuordnung dargestellt.

Evaluiert wurden sieben typische, in Tabelle 9.21 aufgelistete Fehler. Die Fälle wurden so ausgewählt, dass sie ein gutes Spektrum der Vor- und Nachteile abdecken und von den Verantwortlichen durchführbar waren. Dazu zählten vor allem Unterbrechungen einzelner Leitungen, Kurzschlüsse nach Masse oder Defekte in den Sensoren bzw. Aktuatoren.

Heißfilm-
Luftmassenmesser

Fehlerfall 1: (Luftmassenmesser)

Der Heißfilm-Luftmassenmesser mit Sitz zwischen Turbolader und Luftfilter wird vom Motorsteuergerät benötigt, um die von der Verbrennung angesaugte Luftmenge zu bestimmen. Diese wird zur Berechnung verschiedener Parameter, wie Einspritzmenge oder Zündzeitpunkt, benötigt. Simuliert wurde ein defekter Sensor, indem dessen Stecker abgezogen wurde. Nach dem Motorstart erzeugt das Motorsteuergerät den DTC 16486 („Luftmassenmesser- G70 Signal zu klein“).

Nach dem Auslesen der Ereignisspeichereinträge berechnet Neucus folgende Diagnostik-Hypothesen:

<i>Diagnose-Hypothese</i>		<i>Vertrauenswert</i>
1	Luftmassenmesser (ETRN 906 461)	77 %
2	Zentral-Leitungssatz instandsetzen (AP)	42 %
3	Luftmassenmesser aus- u. eingebaut (AP)	35 %
4	Turbolader (ETRN 253 014)	34 %
5	Leitungssatz im Motorraum instandsetzen (AP)	32 %
6	Turbolader aus- u. eingebaut (AP)	23 %

Tabelle 9.7.: Fehlerfall 1 – Diagnostik-Hypothesen von Neucus

Der erste Punkt in der Vorschlagsliste trifft den Fehler. An zweiter Stelle wird ein Fehler im Leitungssatz vermutet. Der dritte Punkt betrifft eine Arbeitsposition, die ebenfalls den Tausch des Luftmassenmessers meint und mit einer verfügbaren Komponentenzuordnungsliste hier nicht erscheinen würde. An vierter Stelle wird ein defekter Turbolader vermutet. Der Defekt erzeugt einen zu niedrigen Druck und damit eine zu geringe Luftmasse. Die interne Plausibilisierung der Messwerte würde damit ebenfalls ein zu kleines Signal feststellen. Hier zeigt sich, dass Neucus nicht nur elektrische, sondern auch mechanische Defekte mit komplexen mechatronischen Wirkketten handhaben kann, sofern diese letztlich einen DTC oder eine vom Kunden wahrnehmbare Beanstandung (S-DISS) hervorrufen.

Im Anschluss daran wurde die GFS durchgeführt. Die einzelnen Prüfschritte und deren Komponentenzuordnung sind in Tabelle 9.8 dargestellt. Der direkte Vergleich zeigt, dass Neucus bereits an erster Stelle die Komponente Luftmassenmesser vorschlägt und erst an zweiter Stelle den Leitungssatz. Die GFS kommt erst mit dem dritten Prüfschritt zum eigentlichen Defekt. Hinsichtlich des Prüfaufwands soll angemerkt werden, dass der erste Prüfschritt, der Luftfilter, zwar noch schnell überprüft wer-

Ergebnis:
Neucus – 1. Stelle
GFS – 3. Stelle

<i>Prüfanweisung</i>	<i>Komponente</i>
1 Überprüfung des Luftfiltereinsatzes auf Durchnässung, Beschädigung und korrekten Sitz	Luftfilter
2 Überprüfung aller Massepunkte im Motorraum gemäß Stromlaufplan	Kabelstrang allgemein
3 Stommessung am Pin 2 des HFL (Soll: 28 mA)	Luftmassenmesser
4 Spannungsmessung an einer Signalleitung (Soll: 1,14 V bis 1,185 V)	Kabelstrang Luftmassenmesser
5 manuelle Plausibilisierung mehrerer Messwertblöcke aus dem MSG	Kabelstrang & Luftmassenmesser

Tabelle 9.8.: Fehlerfall 1 – Prüfanweisungen der GFS und deren Komponentenzuordnungen

den kann, jedoch der zweite Prüfschritt (Überprüfung aller Massepunkte im Motorraum) einen erheblichen Aufwand bedeutet.

Kraftstofftemperaturgeber

Fehlerfall 2: (Kraftstofftemperaturgeber)

Die Viskosität des Kraftstoffs ist temperaturabhängig. Zur Berechnung der notwendigen Einspritzdauer existiert ein Kraftstofftemperaturgeber, der mit dem Motorsteuergerät verbunden ist. Als Fehlerfall wurde ein defekter Sensor simuliert, indem auch hier der Stecker abgezogen wurde. Das Motorsteuergerät erkennt eine Abweichung vom zulässigen Sensorbereich und legt den DTC 16567 („Geber für Kraftstofftemp.-G81 Unterbrechung/Kurzschluss nach Plus“) im Speicher ab. Neucus erzeugt hierzu folgende Diagnostik-Hypothesen:

Diagnose-Hypothese			Vertrauenswert
1	Kraftstofftemperaturgeber (ETRN 906 081)		95 %
2	Temperaturgeber aus- u. eingebaut (AP)		24 %
3	Zentral-Leitungssatz instandsetzen (AP)		17 %

Tabelle 9.9.: Fehlerfall 2 – Diagnostik-Hypothesen von Neucus

Ergebnis:
Neucus – 1. Stelle,
GFS nur be-
dingt hilfreich

Auch hier bestätigt sich die erste Diagnose. Die zweite Position ist wieder identisch mit der ersten und erst mit 17 % wird ein Fehler im Leitungssatz vermutet. Die GFS ergab für diesen Fehler die in Tabelle 9.10 dargestellten Prüfanweisungen. Zwar verweist die Ausgabe des

Prüfanweisung		Komponente
1	Ausgabe des Messwerts Kraftstofftemperatur aus dem Motorsteuergerät (255 °C)	Kraftstofftemperaturgeber & Kabelstrang
2	Manuelle Überprüfung nach Stromlaufplan	Kabelstrang allgemein

Tabelle 9.10.: Fehlerfall 2 – Prüfanweisungen der GFS und deren Komponentenzuordnungen

gemessenen Temperaturwerts auf einen Fehler in der Temperaturmessung, konkretisiert aber nicht zwischen dem Sensor selbst und dem Leitungssatz. Der Punkt wird dennoch als Treffer angesehen, jedoch mit Vorbehalt wegen mangelnder Granularität. Spätestens mit der Prüfanweisung zur manuellen Überprüfung nach Stromlaufplan würde ein KD-Mitarbeiter den Fehler finden.

Im Vergleich haben zwar beide Verfahren einen Treffer mit der ersten verdächtigten Komponente, jedoch ist Neucus hier feingranularer und verdächtigt an erster Stelle den Sensor selbst.

Fehlerfall 3: (Nockenwellensensor)

Als dritter Fehlerfall wurde ein defekter Nockenwellensensor simuliert, indem auch hier der entsprechende Stecker abgezogen wurde. Vom Motorsteuergerät wurde dazu der DTC 19463 („Nockenwellenpositionssensor - G40 kein Signal“) erzeugt. Die Diagnostik-Hypothesen von Neucus sind hierzu:

Nockenwellensensor

Diagnose-Hypothese		Vertrauenswert
1	Nockenwellensensor (ETRN 957 147)	64 %
2	Nockenwellensensor aus- u. eingebaut (AP)	37 %
3	Kabelreparatursatz (ETRN 979 133)	10
4	Quetsch-Föhn-Verbinder (ETRN 971 940)	9
5	Motorsteuergerät aus- u. eingebaut (AP)	8

Tabelle 9.11.: Fehlerfall 3 – Diagnostik-Hypothesen von Neucus

Auch hier bestätigt sich die erste Diagnose. Die dritte wie auch vierte Position (die eigentlich die Zweite ist), vermutet einen Defekt im Kabelsatz. Nicht auszuschließen, aber wenig wahrscheinlich ist mit der 5. Position ein Defekt des Motorsteuergeräts.

Die GFS ergab für diesen Fehler die in Tabelle 9.12 dargestellten Prüfanweisungen. Die erste Prüfanweisung stellt hier eine Art „Rund-

Ergebnis:
Beide erfolgreich,
GFS mit umfang-
reichem Prüfpro-
gramm

Prüfanweisung		Komponente
1	Signalmessung mit digitalem Speicheroszilloskop an Pin 2	Nockenwellensensor inkl. Kabelstrang & Stecker
2	Überprüfung der Versorgungsspannung des Sensors	Kabelstrang Nockenwellensensor
3	Durchgangsprüfungen aller Leitungen zw. Sensor und MSG	Kabelstrang Nockenwellensensor

Tabelle 9.12.: Fehlerfall 3 – Prüfanweisungen der GFS und deren Komponentenzuordnungen

umschlag“ zur gleichzeitigen Prüfung des Sensors, des Kabelstrangs und des Steckers dar. Hierfür existiert im Rahmen der Prüfanweisung eine detaillierte Anleitung zum Einrichten der Messung wie auch Beispielsignalverläufe zur Validierung der Messungen. Je nach Signalverlauf kann der Fehler identifiziert oder zumindest stark eingegrenzt werden. Die Prüfanweisung ist ein gutes Beispiel dafür, wie eine Diagnose-Hypothese mit einer Komponentenprüfung verknüpft werden könnte.

Im Vergleich haben beide Verfahren einen Treffer an erster Stelle. Dabei ist die Prüfanweisung der GFS aber wesentlich feingranularer und unterstützt stark die letztliche Fehlerfeststellung.

Spannungs-
versorgungsrelais
Motorsteuergerät

Fehlerfall 4: (Spannungsversorgungsrelais Motorsteuergerät)

Ein sehr anspruchsvoller Fehler wurde mit dem Defekt des Spannungsversorgungsrelais für das Motorsteuergerät simuliert. Hierzu wurde das Relais abgezogen. Als Folge entstanden 8 DTCs in verschiedenen Steuergeräten:

	<i>Steuergerät (Adr.)</i>	<i>DTC – Fehlertext – Fehlerart</i>
1	Getriebeelektronik (Adr. 2)	18258 – Datenbus Antrieb fehlende Botschaft vom Motorsteuergerät
2	Bremselektronik (Adr. 3)	1314 – Motorsteuergerät – kein Signal/Kommunikation
3	Klima-/Heizungselektronik (Adr. 8)	1314 – Motorsteuergerät – kein Signal/Kommunikation
4	Elektronische Zentralelektronik (Adr. 9)	1314 – Motorsteuergerät – kein Signal/Kommunikation
5	Schaltafeneinsatz (Adr. 17)	1314 – Motorsteuergerät – kein Signal/Kommunikation
6	Diagnoseinterface für Datenbus (Adr. 19)	1314 – Motorsteuergerät – kein Signal/Kommunikation
7	Lenkhilfe (Adr. 44)	1314 – Motorsteuergerät – kein Signal/Kommunikation
8	Leuchtweitenregulierung (Adr. 55)	1314 – Motorsteuergerät – kein Signal/Kommunikation

Neucus erzeugt hierzu folgende Diagnostik-Hypothesen:

	<i>Diagnose-Hypothese</i>	<i>Vertrauenswert</i>
1	Motorsteuergerät programmieren (AP)	97 %
2	Spannungsversorgungsrelais Motorsteuergerät aus- u. eingebaut (AP)	95 %
2	Motorsteuergerät defekt (DISS)	81 %
	[...]	[...]
6	Zentralelektrik instandsetzen (AP)	72 %

Tabelle 9.13.: Fehlerfall 4 – Diagnostik-Hypothesen von Neucus für ein defektes Spannungsversorgungsrelais des Motorsteuergeräts

Die GFS ergab für diesen Fehler die in Tabelle 9.14 dargestellten Prüfanweisungen. Bemerkenswert ist für diesen Fehlerfall, dass obwohl alle Ereignisspeichereinträge auf ein nicht ansprechbares Motorsteuergerät hinweisen, die GFS mit der Prüfanweisung „*Suche nach Stromlaufplan*“ keine Unterstützung für den KD-Mitarbeiter ist, denn die Ursachen hierfür können vielfältig sein. Neucus schlägt hierfür bereits an zweiter Stelle das Spannungsversorgungsrelais mit einem hohen Vertrauenswert vor.

Ergebnis:
Neucus – 2. Stelle
GFS nicht hilfreich

Prüfanweisung	Komponente
1 Suche nach Stromlaufplan	---

Tabelle 9.14.: Fehlerfall 4 – Prüfanweisungen der GFS für ein defektes Spannungsversorgungsrelais des Motorsteuergeräts

Eine Recherche in den Datensätzen zeigt, dass ein solcher Fehler auch in der Praxis, mit diesem oder einem ähnlichen Syndrom, vorkommt. Wäre dies nicht der Fall, würde Neucus hierfür keine Diagnostik-Hypothesen mit hohem Vertrauenswert erzeugen. Der Fall zeigt anschaulich, dass Neucus gerade für komplexe Syndrome, bestehend aus mehreren DTCs, seine Stärken besitzt.

Fehlerfall 5: (Kraftstoffpumpe/Vorratsanzeige)

Für den Kraftstoffgeber existieren zwei Varianten. In der ersten ist der Sensor gemeinsam mit der Kraftstoffpumpe und in der zweiten als getrenntes Bauteil verbaut. Als Fehlerfall wurde ein defekter Kraftstoffgeber simuliert, indem der Stecker am Sensor abgezogen wurde. Das Kombiinstrument erkennt darauf eine Abweichung vom zulässigen Sensorbereich und legt den DTC 0771 („Geber für Kraftstoffvorratsanzeige kein Signal“) im Speicher ab. Neucus erzeugt hierzu nachfolgende Diagnostik-Hypothesen:

Diagnose-Hypothese	Vertrauenswert
1 Kraftstoffgeber aus- u. eingebaut (AP)	91 %
2 Kraftstoffgeber (ET 919 673)	67 %
3 Dichtring für Kraftstoffgeber (ET 919 133)	61 %
4 Kraftstoffpumpe (ET 919 050)	37 %
5 Kombiinstrument aus- u. eingebaut (AP)	37 %
6 Zentral-Leitungssatz instandsetzen (AP)	34 %
7 Leitungssatz für Schalttafel prüfen (AP)	33 %

Tabelle 9.15.: Fehlerfall 5 – Diagnostik-Hypothesen von Neucus

Neucus gibt auch hier mit der ersten Stelle den Hinweis auf die defekte Komponente. Die zweite Position ist dazuzurechnen. Die dritte Position ist leider weniger eindeutig. Allerdings wird dieser Dichtring für die Instantsetzung der Pumpe oder des Gebers benötigt. Erst die nachfolgenden Hypothesen verdächtigen das Kombiinstrument und die entsprechenden Leitungssätze. Die Prüfanweisungen der GFS, dargestellt in Tabelle 9.16, sind für diesen Fehler wenig hilfreich. Zwar verweist die erste Prüfanweisung auf einen Fehler im System Kraftstoffpumpe/-Geber, ist dabei aber zu ungenau. Dasselbe gilt für die zweite Prüfanweisung der GFS.

Ergebnis:
Neucus – 1. Stelle,
GFS wenig hilfreich

Prüfanweisung	Komponente
1 Ausgabe relevanter Messwerte aus dem Kombiinstrument	Kabelstrang Kraft-stoff-pumpe/-Geber
2 Suche nach Stromlaufplan	Kabelstrang Kraft-stoff-pumpe/-Geber

Tabelle 9.16.: Fehlerfall 5 – Prüfanweisungen der GFS und deren Komponentenzuordnungen

Lichtdrehschalter

Fehlerfall 6: (Lichtdrehschalter)

Ein vergleichsweise einfacher Fehlerfall wurde mit dem Defekt des Hauptlichtschalters simuliert. Auch hier kann dazu ohne großen Aufwand der Stecker abgezogen werden. Das Bordnetzsteuergerät (BSG) erkennt daraufhin unplausible bzw. fehlende Signale und erzeugt den DTC 1800 („Lichtschalter-E1, kein Signal/Kommunikation“). Neucus erzeugt hierzu folgende Diagnostik-Hypothesen:

Diagnose-Hypothese	Vertrauenswert
1 Lichtdrehschalter (ETRN 941 431)	26 %
2 Hauptlichtschalter aus- u. eingebaut (AP)	4

Tabelle 9.17.: Fehlerfall 6 – Diagnostik-Hypothesen von Neucus

Die erste Diagnose-Hypothese trifft den simulierten Fehler. Auch die zweite Position passt dazu. Die GFS gibt hierzu die in Tabelle 9.18 dargestellten Prüfanweisungen aus. Durch die Anzeige des Messwerteblocks

Prüfanweisung	Komponente
1 Auslesen und Anzeigen des Messwerteblocks „Schalterstellung Lichtdrehschalter“	Lichtdrehschalter, Kabelstrang (allg.)
2 Überprüfung der Sicherung SC13	— — —
3 Leitungsprüfung nach Stromlaufplan	Kabelstrang Lichtdrehschalter

Tabelle 9.18.: Fehlerfall 6 – Prüfanweisungen der GFS und deren Komponentenzuordnungen

Ergebnis:
Neucus – 1. Stelle
GFS wenig hilfreich

verweist die GFS zwar auf einen Fehler im Bereich des Lichtdrehschalters, konkretisiert aber nicht zwischen dem Schalter oder dem Kabelsatz. Der Punkt wird dennoch als Treffer angesehen, jedoch mit Vorbehalt, wegen mangelnder Granularität. Mit der Prüfanweisung zur manuellen Überprüfung nach Stromlaufplan würde ein KD-Mitarbeiter den Fehler spätestens finden.

Fehlerfall 7: (Rücklicht)

Ein weiterer, einfacher und zugleich häufiger Fehlerfall ist ein defektes Rücklicht. Zur Simulation des Fehlers genügt das Abziehen des Steckers. Der Fehler wird durch das Bordnetzsteuergerät (BSG) erst bei aktiver Beleuchtung festgestellt, denn nur dann wird der Stromverbrauch gemessen und plausibilisiert. Das BSG erzeugt im Fehlerfall den DTC 00984 („Lampe für Schlusslicht links-M4 elektrischer Fehler im Stromkreis“). Gleichzeitig erscheint ein Hinweis im Kombiinstrument. Die Diagnose-Hypothesen von Neucus dazu sind:

Glühbirne – Rücklicht

Diagnose-Hypothese		Vertrauenswert
1	Schlussleuchte defekt (M-DISS)	45 %
2	Kombiinstrument aus- u. eingebaut (AP)	37 %
3	Schlussleuchte aus- u. eingebaut (AP)	27 %
4	Zentral-Leitungssatz instandsetzen (AP)	27 %

Tabelle 9.19.: Fehlerfall 7 – Diagnostik-Hypothesen von Neucus

Die erste Diagnose trifft den simulierten Fehler. Zunächst erstaunlich ist aber dennoch, dass die zweite Position auf das Kombiinstrument abzielt. Die Problematik hier ist, dass das Symptom *Rücklicht defekt* nebenbei in vielen Reparaturfällen vorhanden ist, aber kaum Beachtung findet und nur selten in den Abrechnungsdaten auftaucht. Entsprechend gering ist der Sicherheitswert der ersten Position und umso mehr werden andere Hypothesen wie ein Fehler im Leitungssatz wahrscheinlich (Position 4).

Die GFS ergab für diesen Fehler ebenfalls mit der ersten Prüfanweisung in Tabelle 9.20 einen Treffer. Die weiteren Prüfanweisungen sind

Ergebnis:
Beide Verfahren
erfolgreich, GFS
detaillierter

Prüfanweisung		Komponente
1	Sichtprüfung Lampe Rücklicht links	Rücklicht
2	Sichtprüfung Stecker Rücklicht links und Stecker am BSG	Kabelsatz Rücklicht
3	Durchgangsprüfung mit Angabe der Pins zw. Stecker BSG und Stecker Rücklicht	Kabelsatz Rücklicht
4	Prüfung der Leitung zw. BSG und Rücklicht auf Kurzschluss nach Masse	Kabelsatz Rücklicht
5	Sichtprüfung des Massepunktes im Kofferraum	Kabelsatz Rücklicht

Tabelle 9.20.: Fehlerfall 7 – Prüfanweisungen der GFS und deren Komponentenzuordnungen

ebenfalls sinnvoll. Es ist zu erkennen, dass es sich hier um ein häufig genutztes und entsprechend optimiertes GFS-Programm handelt. Auch hier zeigt sich, dass eine Kombination von Neucus und GFS-Program-

men, besonders für optimierte GFS-Programme, sinnvoll ist.

Zusammenfassung
Praxiserprobung

Es wurde dokumentiert, wie 7 ausgewählte Fehlerfälle im Fahrzeug ausgelöst und die jeweiligen Diagnosen von Neucus und der GFS ausgewertet wurden. Die Ergebnisse sind in Tabelle 9.21 zusammengefasst. Sie zeigen,

Fehler	Neucus	GFS
Luftmassenmesser defekt	1.	3.
Geber für Kraftstofftemp. defekt	1.	MB ^a
Nockenwellensensor defekt	1.	1.
Spannungsversorgungsrelais MSG defekt	2.	SnS ^b
Kraftstoffgeber defekt	1.	MB, SnS
Lichtdrehschalter defekt	1.	MB, SnS
Rücklicht defekt	1.	1.

^a MB: Es wurde als Lösung lediglich der Messwerteblock angezeigt.

^b SnS: Als Lösung wurde „Suche nach Stromlaufplan“ ausgegeben

Tabelle 9.21.: Ergebnis der Praxiserprobung am Fahrzeug hinsichtlich Trefferquote und Granularität

dass Neucus bereits mit den zur Projektzeit zur Verfügung stehenden Daten gute und praxistaugliche Ergebnisse liefert. Sie zeigen aber auch, dass für einen breiten Praxiseinsatz noch Verbesserungen hinsichtlich verschiedener Datenfilter aber auch Komponentenzuordnungen durchgeführt werden müssen, so dass beispielsweise keine Mehrfacheinträge entstehen, die dieselbe Komponente verdächtigen oder wenig aussagekräftige Kleinteile wie Dichtringe vorgeschlagen werden. Die Praxiserprobung zeigt auch, dass Neucus Komponenten wie Bauteile oder Kabelsätze als Diagnose-Hypothesen erzeugt. Doch gerade bei Kabelsätzen wären feingranularere Diagnosen hilfreich. Hier empfiehlt sich die Kopplung mit auf elektrische Fehler spezialisierte Verfahren, wie sie z. B. in [Kriar] vorgestellt werden. Zur Überprüfung, ob eine als defekt vermutete Komponente tatsächlich defekt ist und ausgetauscht werden sollte, empfiehlt sich die Kopplung mit spezialisierten und gut ausgearbeiteten Prüfanweisungen, wie sie die GFS (teilweise) zur Verfügung stellt.

Neucus auf
Komponenten-
Ebene überlegen

Insgesamt konnte die Praxiserprobung zeigen, dass Neucus hinsichtlich der Diagnose-Hypothesen-Generierung auf Komponentenebene der GFS überlegen ist. Bis auf Fehlerfall 4 wurde die tatsächlich defekte Komponente immer an erster Stelle angezeigt. Die GFS stellte in einigen Fällen nur wenig Unterstützung für den KD-Mitarbeiter bereit (vgl. Anzeige von Messwerteblocken (MB) oder „Suche nach Stromlaufplan“ (SnS)). Besonders erwähnenswert ist hierbei der Fehlerfall 4 (Spannungsversorgungsrelais MSG). Für die erzeugte Symptomkombination, bestehend aus 8 Ereignisspeichereinträgen, konnte Neucus zeigen, dass es insbeson-

dere auch für komplexere Fälle geeignet ist. Die GFS konnte hier keine Unterstützung bieten. Mit dem Fehlerfall 1 (Luftmassenmesser) konnte Neucus zeigen, dass es nicht nur elektrische Fehler, sondern gerade auch Kombinationen mit mechanischen Fehlern diagnostizieren kann.

TEIL III: ABSCHLUSS

Im abschließenden Kapitel werden die Ergebnisse der Arbeit zusammengefasst und bewertet. Es folgt ein Ausblick auf weitere Optimierungsmöglichkeiten, alternative Herangehensweisen und interessante wissenschaftliche Fragestellungen. Abschließend wird ein Migrationskonzept zwischen neuronalen Diagnostik-Modellen und der GFS skizziert.

10 Bewertung und Diskussion

Die beschriebene Evaluierung bildet eine umfangreiche und fundierte Grundlage für eine abschließende Bewertung und Diskussion. Die Ergebnisse werden dazu zunächst zusammengefasst und anschließend bewertet und diskutiert. Teil der Diskussion ist vor allem auch ein Ausblick auf weitere Optimierungsmöglichkeiten, alternative Herangehensweisen und interessante wissenschaftliche Fragestellungen. Es werden aber auch notwendige Schritte für einen Praxiseinsatz aufgezeigt. Zwangsläufig gehört dazu auch die Diskussion möglicher Integrationsszenarien, wovon eine Variante konkreter vorgestellt wird.

In dieser Arbeit wurde großen Wert auf eine umfangreiche und fundierte Evaluierung gelegt, die über den rein wissenschaftlichen Funktionsnachweis hinausgeht. Sie stellt damit eine Grundlage für notwendige Diskussionen zu Anwendungs- und Integrationsszenarien dar. Hierzu wurde ein dreistufiges, entwicklungsbegleitendes Evaluationsverfahren, speziell für Offboard-Diagnostikverfahren entwickelt, bestehend aus folgenden Schritten:

1. Einer Bewertung des vorgestellten Konzepts im Vergleich zur GFS
2. Einer Evaluierung anhand 56.000 realer Reparaturfalldaten
3. Einer Praxiserprobung, bestehend aus
 - a) einer manuellen Untersuchung mit ca. 100 Falldaten,
 - b) einer Untersuchung exemplarischer Fälle in Zusammenarbeit mit der Qualitätssicherung der Volkswagen AG und
 - c) einer Praxiserprobung am Fahrzeug.

Konzeptbewertung

Die Konzeptbewertung umfasste eine Evaluierung der Verfahren Neucus und GFS auf Anforderungsebene, wofür ein umfangreiches hierarchisches Kriterienset entwickelt und angewendet wurde. Hier war Neucus der GFS in vielen, aber nicht allen Punkten überlegen. Die Ergebnisse sind knapp in Tabelle 10.1 zusammengefasst.

Der *Einsatzbereich* wird maßgeblich von den zur Verfügung stehenden Daten bestimmt und wurde für Neucus mit dem Kundendienst identifiziert. Ein Einsatz in der Produktion wäre nur bedingt möglich. Für den Einsatz in der Forschung und Entwicklung ist das Verfahren aufgrund der Absenz von Reparaturfalldaten nicht geeignet. Ein Vorteil von Neucus ist die *technische Machbarkeit*. Neucus ist einfach in die bestehende Infrastruktur integrierbar. Server- und Datenstrukturen, Onlineanbindung der Tester, Datenerfassung und Updateverfahren sind bereits vorhanden und müssen nur angepasst werden. Weder bei aktuellen noch bei neuen Fahrzeugen müssten für die Verwendung von Neucus Anpassungen

durchgeführt werden. Die zu erwartenden laufenden *Kosten* für den Einsatz in aktuellen Fahrzeugprojekten sind für Neucus aufgrund des hohen Automatisierungsgrades verhältnismäßig gering. Gerade bezüglich neuer Fahrzeugprojekte besitzt Neucus Vorteile, da hier im Gegensatz zur GFS nicht erst aufwändig Prüfprogramme neu geschrieben oder angepasst werden müssen (als erste Lernbasis können Reparaturfalldaten ähnlicher Fahrzeugprojekte verwendet werden, so dass sich die neue Datenbasis dann mit jedem neuen Reparaturfall aktualisiert). Mit den komplexitätserweiterten Modellen kann Neucus gut mit einer hohen *Variantenvielfalt* umgehen. Variantenspezifische Besonderheiten werden gelernt. Erweiter- und Skalierbarkeit ist durch Standard-Software für Mehrserversysteme kein Problem. Bezüglich der *Anwendbarkeit* ist Neucus in der möglichen Benutzerführung aufgrund gewichteter Diagnose-Hypothesen sehr flexibel, die Internationalisierung ist aufgrund bereits bestehender Bauteilübersetzungen gut lösbar. Die *Nutzerakzeptanz* wird jedoch stark vom Anwendungskonzept abhängen. Neuronale Netze haben per se einen *Black-Box-Charakter*, dem durch geeignete Analogien wie „Andere Nutzer mit demselben Symptombild haben [...] durchgeführt.“ entgegengewirkt werden muss. Die GFS ist im Punkt *Nachvollziehbarkeit* jedoch nicht besser zu bewerten. Im Punkt *Leistungsfähigkeit* wurde für Neucus aufgrund der kontinuierlichen Optimierung durch immer neue, aktuelle Falldaten eine hohe Trefferquote und damit eine geringere Fehlersuchzeit und geringere Fehltauschkosten identifiziert. Anpassungen auf veränderte Defektwahrscheinlichkeiten oder Fehlerabstellmaßnahmen werden automatisiert und schnell durchgeführt. Gerade in diesem Punkt ist Neucus der GFS deutlich überlegen. Letztere besitzt dagegen Vorteile

<i>Kriterium</i>	<i>Bewertung</i>
Einsatzbereich	Kundendienst, bedingt auch Produktion
Technische Machbarkeit	Infrastruktur vorhanden, einfach integrierbar, existierende Daten nutzbar, für bisherige Fahrzeuge einsetzbar
Kosten	Investitionskosten gering, laufende Kosten gering, Kosten für neue Fahrzeugprojekte gering
Leistungsfähigkeit	Hohe Trefferquote, geringe Fehlersuchzeit, Variantenvielfalt handelbar, erweiter- & skalierbar, Granularität datenabhängig
Anwendbarkeit und Nutzerakzeptanz	Abhängig vom Anwendungskonzept, flexibel in der Benutzerführung, einfache Internationalisierung

Tabelle 10.1.: Zusammenfassung der Konzeptbewertung

in der Differenzierbarkeit, da Neucus Diagnose-Hypothesen auf Komponentenebene erzeugt und keine Komponentenprüfprogramme erstellt. Schon hier wurde eine Kombination der Stärken beider Verfahren als möglich und sinnvoll angesehen, um den maximalen Nutzen zu erzielen.

Evaluierung mit realen Reparaturfalldaten

In der zweiten Stufe, der Evaluierung mit realen Reparaturfalldaten, wurden vor allem die Kriterien Leistungsfähigkeit konkret quantifiziert. Hierzu wurde ein Verfahren zur automatisierten Evaluation mit einem speziellen Qualitätsmaß, der normierten Evaluierungsquote q_n entwickelt. Dies war aufgrund der hohen Zahl inkongruenter Fälle notwendig. Die Normierung wurde mit der manuellen Untersuchung der Fallbasis ermittelt. Ein automatisierter Vergleich mit der GFS war aufgrund mangelnder Automatisierbarkeit nicht möglich, jedoch auch mit der angestrebten Kombination der Verfahren auch nicht zwingend notwendig.

Zur Verfügung standen ca. 56.000 Fälle, wovon die ersten 50.000 zum Training und die weiteren 6.000 zur Evaluierung verwendet wurden (disjunkte Partitionierung). Eine Kreuzvalidierung ist aufgrund der kontinuierlichen Veränderung von Defektwahrscheinlichkeiten in den Falldaten nicht sinnvoll. Evaluert wurde zunächst das einschichtige Diagnostik-Modell. Es wurden dabei verschiedene Trainingsparameter untersucht, wie auch der Einfluss verschiedener Maßnahmenklassen und Trainingsperioden. Das Optimum wurde mit der Verwendung der Maßnahmenklassen AP, DISS, ET und ETRN bei einer Trainingsperiodenzahl von fünf erreicht mit einer Evaluierungsquote q_n von 93,1 %. Untersucht wurde weiterhin, ab welcher Fallzahl das Verfahren welche Leistungsfähigkeit erbringt. Für eine Evaluierungsquote von 88 %, knapp unterhalb des Optimums, sind 10.000 Falldaten notwendig. Bereits ab 500 Fällen werden ca. 58 % erreicht. Darauf aufbauend wurde das komplexitätserweiterte Modell untersucht, das den Fahrzeugkontext mit einbezieht. Es hat sich gezeigt, dass dies auch für den zeitlich eng begrenzten Datensatz, der zur Verfügung stand, nochmals eine Verbesserung um +3,3 % auf 96,4 % bewirkt. Abschließend wurden die verschiedenen Optimierungen evaluiert. Das Verlernen alter Reparaturfälle erzielte nochmals +2 %. Die Merkmals-Reduktion erreichte eine Reduktion der Anzahl an Verbindungen von ursprünglich $1 \cdot 10^8$ auf $1,7 \cdot 10^6$ (−98,3 %) und das Pruning trainierter Netze reduzierte nochmals auf 32.004 (−98,1 %) Verbindungen, so dass als Ergebnis hochkompakte effiziente Modelle entstanden. Die durchschnittliche Verbindungszahl pro Ausgangsneuron bzw. pro Komponente beträgt dann ca. 20. Die Merkmals-Reduktion wie auch das Pruning waren zwingend notwendig, damit der Rechenaufwand von Parameteriterationen, bestehend aus vielen hundert verschiedenen Einzelmodellen, mit den zur Verfügung stehenden Ressourcen in einem akzeptablen Zeitrahmen bewältigt werden konnte. Die Evaluierung konnte neben der konkreten Quantifizierung der Leistungsfähigkeit auch Thesen wie die zur Maßnahmenklassenauswahl oder die geringe Eignung von MLPs bestätigen. Sie hat ferner, durch die automatisierte einfache Anwendbarkeit, den Entwicklungsprozess stark beschleunigt.

Praxiserprobung

Die dritte Stufe der Evaluierung umfasste eine Praxiserprobung, bestehend aus einer manuellen Untersuchung mit einer begrenzten Anzahl von Fällen, einer Untersuchung weniger exemplarischer Fälle und einer Praxiserprobung am Fahrzeug. In der manuellen Untersuchung von Fällen

wurde die Evaluierung der Trefferquote eines ausgewählten Diagnostik-Modells händisch durchgeführt und damit Falldaten-spezifische Unzulänglichkeiten wie inkongruente Fälle umgangen. Die jeweiligen Komponentenzuordnungen der vorgeschlagenen Maßnahmen wurde ebenfalls händisch vorgenommen. Es wurde damit eine reale Trefferquote von 94 % erreicht, die sich mit der automatisiert ermittelten Evaluierungsquote q_n von 93,1 % des verwendeten Modells deckt. Damit wurde nicht nur die tatsächliche Leistungsfähigkeit belegt, sondern auch die automatisierte Evaluierung als sehr geeignet und hinreichend exakt festgestellt.

In jeder Evaluierung mit vielen Falldaten können Einzelfälle nur gering beachtet werden. Darum wurde in Zusammenarbeit mit der Qualitätssicherung der Volkswagen AG eine spezielle Vorgehensweise zur Evaluierung realisiert. Dabei wurde Neucus mit vier besonders relevanten Fällen live zusammen mit allen Beteiligten erprobt. Die Ergebnisse konnten alle Beteiligten überzeugen.

Zur Evaluierung der praktischen Anwendbarkeit von Neucus wie auch für einen direkten Vergleich mit der GFS wurde eine Praxiserprobung am Fahrzeug durchgeführt. Hierzu wurden sieben ausgesuchte Fehlerfälle im Fahrzeug ausgelöst und anschließend beide Systeme am Fahrzeug erprobt. Für alle Fälle erreichte Neucus gute bis sehr gute Treffer, meist an 1. Stelle. Sie zeigte außerdem, dass Neucus bereits mit den zur Projektlaufzeit zur Verfügung stehenden Daten gute Ergebnisse erzielt. Besondere Stärken bewies Neucus für komplexere Fehlerbilder, bestehend aus mehreren Symptomen, und für Fehler mit mechatronischen Wirkketten. Verschiedene Optimierungen wären dennoch wünschenswert. Im Vergleich zur GFS war Neucus hinsichtlich Trefferquote auf Komponentenebene überlegen. Die GFS konnte dafür jedoch mit teilweise aufwändigen und detaillierten Komponentenprüfprogrammen überzeugen. Teilweise lieferte die GFS jedoch auch keine oder wenig Unterstützung bei der Fehlersuche.

Mit der Evaluierung konnte ein Nachweis für die praktische Anwendbarkeit und die Leistungsfähigkeit der Ideen und Konzepte erbracht werden. Gleichzeitig wurde in einer vergleichenden Bewertung mit der Geführten Fehlersuche gezeigt, dass die Verfahren auf Komponentenebene der GFS überlegen sind. Letztere besitzt dafür den Vorteil ausführlicher, anleitender Komponentenprüfprogramme. Es wurden Randbedingungen wie die notwendige Mindestfallzahl und der Einfluss verschiedener Modell-Parameter etc. untersucht. Die jeweiligen Vor- und Nachteile wurden dabei ausführlich diskutiert. Im Rahmen der Evaluierung wurde damit eine umfassende Untersuchung auf verschiedenen Ebenen durchgeführt, die den neuronalen Diagnostik-Modellen eine auf Komponentenebene hohe Leistungsfähigkeit bescheinigt, damit eine Abschätzung der für einen Praxiseinsatz tatsächlich erwartbaren Leistungsfähigkeit zulässt und eine Kombination beider Verfahren empfiehlt.

Abschließende
Bewertung

10.1. Ausblick

Ausgehend von den Erfahrungen aus der wissenschaftlichen Arbeit und den praktischen Erfahrungen aus der Umsetzung werden im folgenden eine Auswahl an aus Sicht des Autors als sinnvoll erachteten Optimierungsmöglichkeiten der Diagnostik-Modelle vorgestellt. Erst anschließend werden zwei weitere wissenschaftlich interessante Fragestellungen adressiert, bevor dann auf verschiedene Integrationsmöglichkeiten eingegangen wird.

10.1.1. Optimierung der Diagnostik-Modelle

Im Folgenden wird eine Auswahl an Optimierungsmöglichkeiten der Diagnostik-Modelle vorgestellt:

Interaktive Symptomabfrage: Bisher werden Kundenwahrnehmung (SDISS-Symptome), deren Status (vorhanden/nicht vorhanden) nicht bekannt ist, als nicht vorhanden angenommen. Ein Status *unbekannt* kann für neuronale Modelle nicht implementiert werden. Die Annahme als *nicht vorhanden* ist mit einer A-priori-Auftrittswahrscheinlichkeit von $\tilde{P}(s_i) \approx 0,14\%$ jedoch hinreichend genau. Mit dem Vorhandensein von DTCs ändern sich jedoch auch die Defektwahrscheinlichkeiten von Komponenten $P(k_i|S_R)$, und damit könnte das Vorhandensein einzelner Kundenwahrnehmungen mit $P(s_i|H_D)$ sehr viel wahrscheinlicher werden. Bayes'sche Netze wären zwar in der Lage mit einer Rückpropagation von Evidenz diese Wahrscheinlichkeiten zu berechnen, haben sich bisher jedoch aus Performanzgründen als ungeeignet erwiesen. Könnte $P(s_i|H_D)$ mittels der neuronalen Modelle abgeschätzt werden, wäre ein interaktives Nachfragen von Kundenwahrnehmungen möglich, welches dadurch die Diagnosen präzisieren und damit Zeit und Aufwand für Messungen und Komponentenprüfungen reduzieren könnte. Anstrengungen zur vermehrten Kundenbefragung zur Reduktion des Fehlersuchaufwands des KD-Mitarbeiters gibt es bereits unter dem Begriff „*Symptom-basierte Diagnose*“. Dies wird bisher jedoch aufwändig mit durch Experten erstellten Musterfällen realisiert.

Beschleunigung des Lernens: Mit der im Projekt verwendeten Trainings- und Evaluierungsumgebung Neucus wird für das Training eines Diagnostik-Modells je nach Parameterset zwischen 10 und 30 Stunden benötigt¹. Zwar kann die Zeit durch das vorgestellte Parallelprocessingframework JPPF stark reduziert werden, dennoch würde die notwendige Rechenleistung mit zunehmender Fallzahl, wie sie im Praxiseinsatz zu erwarten ist, linear steigen. Hierfür gibt es mehrere Optimierungsmöglichkeiten:

¹ Die Angaben gelten für einen Intel Core Duo P8600@2,4GHz und 2GB RAM

1. Optimierung der Softwarestrukturen
2. Optimierung der Merkmals-Reduktion
3. Verwendung effizienterer Verfahren zur Bestimmung der Verbindungsgewichte

Der erste Punkt ist evident und muss nicht weiter erörtert werden. Bezüglich des zweiten Punkts, der Merkmals-Reduktion, zeigt sich, dass mit dem jetzigen Verfahren zwar die Verbindungszahl um 98,5 % auf ca. $1,5 \cdot 10^6$ gesenkt wurde, das Pruning zeigt jedoch, dass gerade einmal 32.000 Verbindungen (0,032 %) der ursprünglichen Verbindungen notwendig sind. Mit bivariaten statistischen Tests, wie der Signifikanztest oder der Korrelationskoeffizient, wird eine weitere Optimierung der Merkmals-Reduktion vermutlich nicht möglich sein, wie die Evaluierung hinsichtlich der Distanzgrenze d_{th} zeigt. Hier könnten multivariate Tests noch Vorteile bringen.

Beschleunigung durch Regressionsverfahren: Weit größeres Potential zur Beschleunigung des Lernens ist jedoch bei der Wahl des Lernverfahrens selbst zu erwarten. Bereits in der Fokussierung auf der Forschungsaufgabe in Kapitel 3.3 wurde der Zusammenhang von einschichtigen neuronalen Netzen zur logistischen Regression aufgezeigt (vgl. dazu auch [KK02, S. 43ff.]). Die Regressionsparameter $\beta = \{\beta_0, \dots, \beta_n\}$ einer Variablen k_i entsprechen dabei den Verbindungsgewichten $W_{(i)} \in W$ zum Ausgangsneuron der Komponente k_i . In der Bestimmung der Parameter gibt es jedoch Unterschiede, da je nach Verfahren andere Fehlerfunktionen verwendet werden als in neuronalen Netzen. Dennoch könnten die so entstehenden Unterschiede durch ein Nachtraining beseitigt werden. In [KH05] werden nun drei schnelle auf sparse multinominale Daten optimierte Verfahren zur Bestimmung der Regressionsparameter β vorgestellt. Die Performanz hinsichtlich Rechenzeit ist dabei bis zu doppelt so schnell wie SVM-Light, eine SVM-Implementierung bestehend aus hochoptimiertem C-Code. In [RK06] wird ein Verfahren, ebenfalls zur Bestimmung der Regressionsparameter für sparse Daten mit bis zu $1,7 \cdot 10^6$ Datensätzen und 46.000 Merkmalen vorgestellt. Das Verfahren ist nach Angaben der Autoren bis zu 353-mal schneller als etablierte Standardalgorithmen. Der Einsatz eines dieser Verfahren zur schnellen approximativen Bestimmung der Verbindungsgewichte des einschichtigen Diagnostik-Modells würde auch hier einen enormen Gewinn bringen. Anhand der durch die Regressionsparameter bestimmten Verbindungsgewichte wäre dann ein Pruning wie in Kapitel 7.7 vorgestellt möglich. Die so entstehenden kompakten Netze könnten dann nach den bereits beschriebenen Verfahren zur Komplexitätserweiterung weiter verwendet werden. Ein Unterschied in der Leistungsfähigkeit ist aufgrund des Nachtrainings und der damit verbundenen Modifikation der Verbindungsstärken auf die durch

den hier verwendeten Lernalgorithmus definierte Fehlerfunktion, nicht zu erwarten.

Neben den vorgestellten Optimierungsmöglichkeiten für die hier entwickelten Verfahren sollen im Folgenden noch zwei weiterführende, wissenschaftlich interessante Arbeitsfelder aufgezeigt werden.

10.1.2. Integration von A-priori-Wissen in neuronale Modelle

Die große Anzahl an Fehlersuchprogrammen der GFS stellt ein beachtliches Wissen dar. Mit einer automatisierten Auswertung dieses Wissens wäre es sinnvoll, dieses a priori in die Diagnostik-Modelle zu integrieren. Damit wären bereits Diagnosen ohne Falldaten möglich. Mit den folgenden Fallzahlen würden diese dann jedoch stetig genauer bzw. adaptieren sich an sich verändernde Bedingungen. Ein solches Vorgehen wäre vor allem für neue Fahrzeugprojekte interessant. Hier wäre es auch interessant, Wissen aus FMEA-Analysen semi-automatisiert zu verwenden (vgl. dazu auch [FJ05]). Bereits sehr früh wird in [Gal93, S. 255ff.] eines der ersten hybriden Neuro-Expertensysteme, genannt *MACIE*, beschrieben. In diesem für die medizinische Domäne zugeschnittenen System, werden von Experten (Ärzten) aufgestellte Expertenregeln der Form IF-THEN in ein zunächst sehr einfaches neuronales Netz überführt, das dann durch zunehmende Ergebnisse optimiert wird. Viele ähnliche Projekte wie z. B. [TS94] und [Gha98] folgten. In [CZ00, S. 37ff] werden allgemein verschiedene Techniken zur Integration verschiedenen Wissens in neuronale Netze und deren Anwendung aufgezeigt. [BC04] beschreibt in seiner Dissertation, wie aus allgemeinen MLP-Netzen Regeln nicht integriert, sondern in der Form IF-THEN extrahiert werden können. Teilweise wird dann auch von *Knowledge-based neural networks* (KB-NNs) gesprochen [Gha98]. McGarry geht in [MM99] noch einen Schritt weiter und beschreibt einen iterativen Prozess, in dem zuerst aus Regeln neuronale Netze erzeugt werden. Anschließend werden die Netze durch reale Daten weiter trainiert bzw. optimiert. Diese Netze werden dann wieder zurück in Regeln konvertiert, die dann wiederum selbst modifiziert werden können. Abbildung 10.1 veranschaulicht das Prinzip. Solche Verfahren benötigen damit weit weniger Falldaten für die gleiche oder eine bessere Diagnostik-Leistung (bewiesen z. B. in [Fu93, S. 173]). Sie sind in ihrer Funktionsweise transparenter aufgrund der Fähigkeit zur Regelextraktion und bieten darüber hinaus wertvolles Wissen in Form von Zusammenhängen für die Qualitätssicherung. Wissenschaftlich interessant ist zum einen, ob diese Verfahren auch für Sparse Data und hochdimensionalen Merkmalsräumen anwendbar sind, und zum anderen, solche Verfahren zu entwickeln.

Abbildung 10.1.: Funktionsprinzip nach [MM99]: Zyklische Integration, Verfeinerung und Extraktion von Regeln

10.1.3. Interaktive Diagnostik mittels Bayes'scher Netze

In dieser Arbeit fiel die Entscheidung (aus gutem Grund) auf neuronale Netze. Dennoch bieten Bayes'sche Netze mit der Rückwärtspropagation von Evidenz eine interessante Eigenschaft. So wäre ein Bayes'sches Netz in der Lage, die Defektwahrscheinlichkeiten von Komponenten neu zu berechnen, wenn eine als defekt vermutete Komponente als gut geprüft wurde (diese Zusammenhänge sind nicht immer linear, so dass eine Neuordnung in der Reihenfolge stattfinden kann). Weiterhin können Bayes'sche Netze mit unbekannten Symptomen umgehen². Dies ist gerade für Kundenwahrnehmungen (S-DISS) hilfreich, deren Vorhanden- oder Nichtvorhandensein bis auf die Beanstandung selbst normalerweise nicht bekannt ist. Mit der Rückwärtspropagation kann auch $P(s_i|K_R)$ berechnet werden, also die Wahrscheinlichkeit für das Vorhandensein des Symptoms s_i bei gegebener Evidenz für den Defekt einer Komponente(nmenge) K . So könnten dann Symptome gezielt nachgefragt und die Diagnosen weiter präzisiert werden. Im Laufe des Projekts wurde bereits mit Bayes'schen Netzen experimentiert, sie erwiesen sich jedoch aus zwei Gründen für ungeeignet: 1. Aufgrund der geringen Fallzahl pro Merkmal ist Strukturlernen nicht möglich. 2. Wird die Struktur mithilfe der Signifikanzmatrix vorgegeben, ist ein solches Netz aufgrund der hohen Verbindungszahl nicht mehr berechenbar. Auch nach Optimierungsverfahren wie *divorcing* [Jen02, S. 78ff.] waren die Netze noch zu groß. In [NZ06, S. 267] ist dazu folgendes Zitat zu finden, das auf die beschriebene Schwachstelle hinweist: „*For high-dimensional classification problems, however, a purely Bayesian approach may sometimes be impractical [...]*“. Ein Lösungsansatz wäre es, Mengen von Bayes'schen

² Als unbekannt wird hier ein unbekannter Status (vorhanden/nicht vorhanden) des Symptoms verstanden.

Netzen zu erzeugen, jedes Netz spezialisiert auf ein Cluster von stark zusammenhängenden Komponenten und Symptomen. Weitere Lösungsansätze sind beispielsweise auch objektorientierte Bayes'sche Netze. In [HAJ⁺06] wird ein solches Netz anhand eines kleinen Subsystems aus dem Kraftfahrzeugbereich vorgestellt. Inwieweit das Verfahren für größere Netze anwendbar ist, bleibt unbekannt. In [NDS07] wird ein Verfahren zum iterativen Strukturlernen Bayes'scher Netze für Datensätze mit hoher Merkmalszahl vorgestellt. Dabei wird die Lernaufgabe in kleine Subbereiche aufgeteilt und diese iterativ erweitert. Das Verfahren wurde dort jedoch auch nur für 1110 Knoten und 5000 Falldaten validiert. Die wissenschaftliche Aufgabe besteht darin, ein Verfahren zu entwickeln, welches mit begrenzten Ressourcen die enorme Merkmalszahl und Datenmenge beherrschen kann.

Würde die Herausforderung der Komplexität gelöst, könnten damit noch präzisere Diagnosen und vor allem eine iterative interaktive Diagnostik möglich sein. Dies könnte die Fehlersuchzeit insofern weiter reduzieren, wie die Überprüfung von Kundenwahrnehmungen (S-DISS) i. d. R. mit weniger Aufwand als mit Messungen und Komponentenprüfungen zu erreichen ist.

10.2. Integrationsmöglichkeiten

10.2.1. Kombination verschiedener Diagnoseverfahren

Die Beschreibung von Verfahren zum Stand der Technik in Kapitel 3.2 macht deutlich, dass viele verschiedene Verfahren mit individuellen Stärken und Schwächen, unterschiedlichen Granularitäten in den Diagnosen und unterschiedlichen Arten der Wissensakquisition (Lernen aus Falldaten, Bordnetzbeschreibungen oder Expertenregeln, etc.) existieren. Bedingt durch die teilweise sehr konträren Technologien sind solche Verfahren bisher jedoch nicht in einem System kombinierbar. Dies wäre notwendig, um von den Stärken aller Verfahren in einem System profitieren zu können. Zu Publikationen, die eine Kombination mehrere Verfahren beschreiben gehört z. B. das SMMART-Projekt [SMM09]. Hier identifiziert ein CBR-System verdächtige *Highlevelkomponenten*. Die Komponentenüberprüfung geschieht dann entweder über Fehlerbäume (FT) oder einem modellbasiertem Verfahren (MBR). Eine echte Integration von FT oder MBR geschieht jedoch nicht. Es darf sich hier der KD-Mitarbeiter entscheiden, welches Verfahren er nutzen will. Für andere Projekte mit dem Ziel, mehrere Verfahren zu kombinieren, verhält es sich ähnlich. Ein allgemeingültiger Ansatz ist bisher nicht bekannt.

Es wurde daher begonnen, einen solchen weiter gefassten Ansatz zur Kombination von Verfahren mit verschiedenen Technologien und Diagnose-Granularitäten zu entwickeln. Das primäre Ziel dabei ist immer eine möglichst kostengünstige Reparatur. Mit der Konzeptanalyse in Kapitel

Kein allgemeingültiger Ansatz bekannt

Entwicklung eines neuen Ansatzes

9.2 lassen sich hierfür die beiden Hauptkriterien Leistungsfähigkeit und Anwendbarkeit/Nutzerakzeptanz identifizieren. Optimiert werden müssen folglich die Unterkriterien Fehlersuchzeit und Fehltauskosten unter den Randbedingungen der Nachvollziehbarkeit, Flexibilität und Akzeptanz. Für ein Optimum müssen zwangsläufig auch die Fähigkeiten und das Wissen des KD-Mitarbeiters mit einbezogen werden. Zur Minimierung der Reparaturkosten muss der KD-Mitarbeiter maximal z. B. durch kontextsensitive Informationsbereitstellung von Explosionszeichnungen, Reparaturleitfäden, etc. unterstützt werden.

Bezüglich der Optimierung der Fehlersuchkosten muss letztlich der Aufwand zur sicheren Identifikation des defekten Bauteils oder der zielführenden Reparatur minimiert werden. Dieser setzt sich aus der Summe der einzelnen Prüfanweisungen zusammen. Entscheidend ist damit, welche und in welcher Reihenfolge Prüfanweisungen durchgeführt werden. Dafür muss für jede bekannt sein, welche Komponenten oder Teilkomponenten (dazu zählen auch Leitungselemente) sie abprüft, welcher Aufwand hierfür notwendig ist, und welche Zustandsänderungen am Fahrzeug infolge von z. B. freigelegten Steckern, Messpunkten, etc. erfolgt sind. Letzterer Punkt muss zwangsläufig Teil der Optimierungsstrategie sein, da die eigentliche Messung meist den geringsten Zeitaufwand bedeutet, d. h. die Vor- und Nachbereitung das Gros ausmacht. Beispielsweise sind Messungen an einem bereits freigelegten Stecker sehr viel günstiger. In [Kriar] werden mit der automatisierten Prüfanweisungsgenerierung viele dieser Punkte adressiert, jedoch ist das Verfahren nur für elektrische Fehler anwendbar. Für eine Gesamtfahrzeugdiagnose ist eine Kombination mit weiteren Verfahren folglich zwingend notwendig.

Anforderungen

Der Prozess der Diagnostik als iterativer Prozess, wurde bereits ausführlich in Kapitel 2 beschrieben und durch Abbildung 2.4 illustriert. Nach der Ermittlung der ersten Symptome wird die Generierung von Diagnose-Hypothesen notwendig. Auch hinsichtlich einer Optimierung macht die Einführung von Hypothesen mit der Ungewissheit über den wahren Zustand einer Komponente und der dadurch notwendigen Quantifizierung mittels Defektwahrscheinlichkeiten Sinn. Zur Hypothesen-Generierung existieren ebenfalls verschiedene Verfahren, deren Ergebnisse zu einer gemeinsamen Diagnose-Hypothesenmenge fusioniert werden müssen. Allerdings ist mit geringen Falldaten auch die Defektwahrscheinlichkeit selbst mit einer mehr oder weniger großen Unsicherheit belegt. Ein Maß hierfür ist z. B. das sogenannte Vertrauensintervall. Die Fusion verschiedener Diagnose-Hypothesen muss diese Unsicherheiten berücksichtigen. Ansätze hierfür liefert z. B. die Dempster-Shafer-Evidenztheorie [Sha76] zur Kombination verschiedener Hypothesen, die klassischen Ansätzen wie z. B. der Maximum-Likelihood-Schätzungen vorgezogen werden sollte.

Ein Lösungsansatz

Mit einer fusionierten Diagnose-Hypothesen-Menge, folgt der Schritt der Aktionsplanung, d. h. es müssen Aktionen zur Überprüfung der verdächtigten Komponenten erfolgen. Wenn die Prüfanweisungseigenschaften, wie abgeprüfte (Teil-)Komponenten, Zustandsvorbedingungen, Zu-

- Hypothesen-Generierung
- Fusion mittels Vertrauensintervallen
- Aktionsplanug & Prüfplangenerierung

Herausforderung
– einheitliche Schnittstellen

standsänderung am Fahrzeug und Aufwand, die gleichzeitig Schnittstelleneigenschaften darstellen, bekannt sind, können Optimierungsverfahren wie z. B. Wahrscheinlichkeitsbäume o.ä. eingesetzt werden. Optimierungsverfahren können dann den günstigsten Pfad mit dem wahrscheinlich geringsten Gesamtaufwand ermitteln, wodurch die jeweils optimale nächste Prüfanweisung bestimmt ist. Mit der Schnittstellenbeschreibung ist es dabei unabhängig, von welchem Verfahren die Prüfanweisungen generiert werden. Gemäß dem iterativen Diagnostik-Prozess folgt nach ausgeführter Prüfanweisung eine Neuberechnung der Diagnose-Hypothesen, da mit jedem Prüfschritt Evidenz zum Zustand einer Komponente erlangt wird, die die optimale Prüfreihefolge ändern kann. Sind die Defektwahrscheinlichkeiten der Komponenten näherungsweise bestimmt, wird so gemittelt über alle Reparaturfälle ein Optimum hinsichtlich des Fehlersuchaufwands erreicht. Da nicht nur Diagnose-Hypothesen, sondern auch Erfolgswahrscheinlichkeiten von Prüfanweisungen und vage abschätzbare Prüfaufwände einer Unsicherheit unterliegen, sollte ein Optimierungsverfahren diese Ungewissheiten ebenfalls berücksichtigen. Auch hier könnte eine Modellierung durch Dempster-Shafer eingesetzt werden.

Mit dem grob skizzierten Verfahren zur umfassenden Kombination von Diagnostik-Algorithmen wäre eine optimierte Diagnostik möglich. Allerdings erfüllen vor allem etablierte Verfahren nicht die dafür geforderten Schnittstellen. Beispielsweise sind für die GFS weder Prüfaufwände, noch abzuprüfende Komponenten, noch Vor- und Nachbedingungen oder Zustandsänderungen etc. automatisiert auswertbar verfügbar. Mit dem Trend zur „*modularisierten GFS*“ kommen zwar Schnittstellenbeschreibungen hinzu, werden aber letztlich nicht vollständig erfüllt. Wann etablierte Verfahren solche Schnittstellen erfüllen, bleibt allerdings ungewiss.

10.2.2. Kombination von Diagnostik-Modellen mit der GFS

Mit der Annahme, dass eine umfassende Kombination verschiedener Verfahren nicht in naher Zukunft realisiert wird, wird nachfolgend ein mehr pragmatischer Ansatz zur Migration von den in dieser Arbeit beschriebenen lernenden Verfahren mit Prüfprogrammen der GFS entwickelt.

Ausgangssituation

Die Skizze oben links in Abbildung 10.2 stellt vereinfacht die Ausgangssituation der GFS dar. Für jedes DTC-Symptom s_i existiert genau ein Fehlersuchprogramm GFS_i der GFS. Theoretisch können auch für S-DISS-Symptome Fehlersuchprogramme existieren (gestrichelter Pfeil). Ein Fehlersuchprogramm GFS_i überprüft wiederum direkt oder indirekt eine oder mehrere Komponenten K . Dabei können auch verschiedene Fehlersuchprogramme dieselbe Komponente prüfen. Dies ist notwendig, wenn z. B. der Defekt einer Komponente verschiedene DTCs verursachen kann (vgl. z. B. Fehlerfall 7 – Spannungsversorgungsrelais MSG mit 8

Abbildung 10.2.: Pragmatisches Migrationskonzept von Diagnostik-Modellen und Prüfprogrammen der GFS

DTCs). Ein erster Migrationsschritt, dargestellt in Abbildung 10.2 oben rechts, wäre es dann, anstatt Diagnostik-Modelle auf Diagnose-Hypothesen $h_D \in H_D$ für Komponenten K die Modelle auf Diagnose-Hypothesen $h_{GFS} \in H_{GFS}$ für hilfreiche GFS-Programme GFS_i zu trainieren. Ein solcher Schritt ist möglich, da in den Reparaturfalldaten die ausgeführten Fehlersuchprogramme und deren Ergebnisse ebenfalls enthalten sind. Dadurch würde erreicht werden, dass 1.) im Fall von mehreren Symptomen (DTCs) nicht eine unsortierte Liste an GFS-Programmen ausgeführt, sondern diese Liste nach einem Vertrauenswert v sortiert wird, mit allen daraus resultierenden bereits diskutierten Vorteilen; 2.) würden automatisch auch DISS-Symptome mit GFS-Programmen verknüpft, sofern aus den Falldaten eine solche Verbindung hervorgeht; 3.) wäre auch die Erweiterung der Modelle auf den Fahrzeugkontext in diesem Schritt möglich und sinnvoll.

Im Zuge der Modularisierung der GFS-Programme, d. h. jedes GFS-Programm GFS_i überprüft dann genau eine Komponente k_i , werden im zweiten Schritt für diese Programme wieder Diagnose-Hypothesen für Komponenten gelernt (Abbildung 10.2 unten links). Eine solche heterogene Diagnose-Hypothesen-Formulierung ist mit den hier vorgestellten Verfahren möglich. Die Diagnose-Hypothesen werden damit zunehmend feingranularer, da nicht mehr nur GFS-Programme, die u.U. mehrere Komponenten überprüfen, verwendet werden.

Im dritten und letzten Schritt ist die Modularisierung abgeschlossen und die Diagnostik-Modelle lernen, wie ursprünglich vorgesehen, Diagnose-Hypothesen für Komponenten. Durch das schrittweise Migrationskonzept fällt die Entscheidung zum Einsatz eines lernenden

Schritt 1: Lernen von Diagnose-Hypothesen zu GFS-Programmen

Schritt 2: Modularisierung der GFS

Schritt 3: Modularisierung abgeschlossen

Verfahrens leichter, wie auch die notwendigen Schritte dazu geringer sind. Weiterhin wird das Risiko reduziert, da gerade im 1. Schritt der mögliche entstehende Schaden in einer ungünstigen Ordnung der GFS-Programme besteht. Mit zunehmendem Vertrauen in das System wie auch mit zunehmender Reife der Software und der Prozesse sinkt das Risiko für die weiteren Schritte.

Das vorgestellte Migrationskonzept ist jedoch nur eine denkbare Variante, die hier vielmehr als Denkanstoß für weitere Varianten verstanden werden will. Nicht zuletzt wird das Erfolgspotential solcher Systeme nicht nur rein technisch, sondern wesentlich auch von den Fähigkeiten und dem Willen der verantwortlichen Menschen bestimmt.

11 Zusammenfassung

Komplexer werdende Fahrzeuge mit wachsender Variantenvielfalt und immer kürzer werdenden Entwicklungszyklen machen neue Diagnostikverfahren für die Servicewerkstätten notwendig. Etablierte Expertensysteme, wie z. B. die für Volkswagen eingesetzte Geführte Fehlersuche (GFS), haben unter anderem den Nachteil, dass nicht alle Fehler und Fehlerkombinationen im Voraus bekannt sind, speziell im Fall von komplexen Fehlern, Fehlerauswirkungen unter Umständen nicht vollständig und nicht korrekt modelliert werden sowie Erstellung und Korrekturen der Wissensbasis langsam und aufwändig sind. Besonders relevant dürfte das für neue Fahrzeugkonzepte wie z. B. Hybrid- oder Elektrofahrzeuge werden, für die noch relativ wenig Expertenerfahrung vorhanden ist. Eine Lösung hierfür wurde mit dem Konzept des Lernens aus Reparaturfällen vorgestellt. Dabei werden mittels der in den Servicewerkstätten aufgezeichneten Reparaturfalldaten Diagnostik-Modelle trainiert, anschließend an die Werkstätten verteilt und dort zur Fehlerdiagnostik eingesetzt. Mit dem dadurch entstehenden iterativen Feedbackprozess entsteht ein sich kontinuierlich selbst optimierendes, adaptives Diagnostik-System. Neue Lösungen werden schnell und effizient gelernt.

Bereits heute werden umfangreich Falldaten aus Servicewerkstätten aufgezeichnet, primär jedoch für statistische Auswertungen. Die Falldaten besitzen somit nicht die für ein Lernsystem wünschenswerte Qualität. Weiterhin herausfordernd für ein Lernsystem ist der Kontrast (Dualität) zwischen Symptom- und Lösungsinformationen und Kontextinformationen. Während sich erstere durch eine nominal dichotome Kodierung, Unschärfe und Rauschen, eine hohe Merkmalsdimension und Sparsity (geringe Fallzahl pro Merkmal) sowie Mehrfachlösungen auszeichnen, sind Kontextinformationen multinominal, eindeutig und dense (hohe Fallzahl pro Merkmal). Hinzu kommen bedingte Abhängigkeiten zwischen Symptominformationen und Kontextinformationen.

Als Diagnostik-Modelle wurden verschiedene statistische und neuronale Modelle entwickelt und untersucht. Ausgehend von einer wahr-scheinlichkeitstheoretischen Betrachtung der Daten wurde zunächst ein deskriptives Modell abgeleitet. Dieses stellte sich jedoch als nicht generalisierungsfähig und rechenaufwändig heraus, so dass mit dem Satz von Bayes ein inferenzielles statistisches Modell entwickelt wurde. Dieses Modell verfügt bereits über eine begrenzte Generalisierungsfähigkeit, kann jedoch aufgrund der geringen abbildbaren Komplexität des Modells selbst keine bedingten Abhängigkeiten zwischen Symptominformationen und Kontextinformationen abbilden. Damit können fahrzeugindividuelle

Falldaten

Statistische
Diagnostik-Modelle

Neuronale Diagnostik-Modelle

Unterschiede nicht berücksichtigt werden, was zwangsläufig zu einer geringeren Diagnostikleistung führt.

Aus diesem Grund wurde im nächsten Schritt die Überführung des inferenziellen Modells in ein einschichtiges neuronales Netz mit sigmoider Aktivierungsfunktion gezeigt. Hierdurch wird es erstmals möglich, die Verbindungsgewichte des neuronalen Netzes zu interpretieren und damit dem „*Black-Box*“-Charakter der Netze entgegenzuwirken. Dennoch bleiben auch mit dem neuronalen Modell die Nachteile des inferenziellen Modells weitgehend bestehen.

Als Lösungsansatz hierfür wurden zunächst Multi-Layer-Perceptron-Netze (MLP) untersucht, die, geeignet modelliert, jede beliebige Abhängigkeit in den Daten abbilden können. Bedingt durch die hohe Merkmalszahl, der Sparsity und geringen Qualität der Daten haben sich diese MLPs jedoch wiederum als ungeeignet herausgestellt. Der Grund hierfür liegt vor allem in der geringen Fallzahl pro Merkmal, die zu einer Unterbestimmtheit (zu wenig Trainingsfälle pro Netzparameter) und damit zu einer geringen Generalisierungsleistung des Netzes führt. Weiterhin entstehen zusammen mit Kontextinformationen unter Umständen nicht nachvollziehbare Diagnose-Hypothesen. Ein weiterer Nachteil ist die mangelnde Interpretierbarkeit der Netzverbindungen. Damit wurde deutlich, dass eine auf die Anforderungen spezialisierte Konstruktions- und Trainingsmethode notwendig ist, die die Dualität der Falldaten explizit berücksichtigt. Eine Recherche bekannter konstruktiver Lernverfahren brachte jedoch kein Verfahren hervor, das die Anforderungen erfüllen konnte.

Spezialisierte Konstruktions- und Trainingsmethode

Das in dieser Arbeit entwickelte konstruktive Verfahren verwendet als Basis das trainierte einschichtige neuronale Modell, deren abbildbare Komplexität gezielt um die Anforderungen der Dualität und der bedingten Abhängigkeiten zwischen Symptominformationen und Kontextinformationen erweitert wurde. Hierzu wurde eine zusätzliche verdeckte Schicht integriert, dessen Neuronen als virtuelle kontextkorrigierte Symptom-Neuronen interpretiert werden können. Diese Neuronen sind nicht wie für ein MLP feed-forward vollvernetzt, sondern nur mit dem jeweiligen Symptom-Neuron, dafür jedoch mit allen Kontextneuronen verbunden. Durch ein gezieltes Training der neuen Verbindungen werden die Einflüsse der Kontextinformationen auf das jeweilige kontextkorrigierte Symptomneuron sichtbar. Ein abschließendes Pruning und Nachtraining erzeugte das finale kompakte Netz. Dieses wurde daher auch als komplexitätserweitertes neuronales Modell bezeichnet. Die Interpretierbarkeit der Verbindungsgewichte bleibt dabei durch das spezielle Training erhalten.

Optimierung der Adaptivität

Eine Optimierung, die speziell für einen nachhaltigen Einsatz im Kundendienst sinnvoll ist, ist das Verlernen von altem Wissen. Dies wird durch eine Partitionierung der Falldaten und einem seriellen Training mit jeder Partition erreicht. Dadurch werden Verbindungsgewichte immer dann modifiziert, wenn neuere Falldaten dies hinsichtlich der Minimierung des Netzfehlers erforderlich machen. So kann das Netz adaptiv

auf Veränderungen wie Fehlerabstellmaßnahmen, Änderungen in der Produktionsqualität oder Alterungsprozesse einer Flotte reagieren.

Für die prototypische Umsetzung der Verfahren waren weitere Optimierungen, insbesondere zur Beschleunigung des Trainings notwendig. Eine Optimierung zur Reduktion der A-priori-Verbindungszahl wurde durch eine Merkmalsreduktion erreicht. Hierzu wurde ein Verfahren entwickelt, das zunächst mittels Signifikanztests relevante Merkmale für jedes Ausgangsneuron schätzt. Aufgrund der Asymmetrie der Daten (Sparsity) werden dadurch jedoch relevante, negativ korrelierte Merkmale nicht erkannt. Um diese dennoch zu berücksichtigen, wurden die Signifikanzen in einen Graphen überführt, aus dem ein über mehrere Knoten hinweg definiertes Distanzmaß abgeleitet wurde. Damit werden implizit negativ korrelierte Merkmale wieder berücksichtigt. Das Verfahren konnte die A-priori-Verbindungszahl auf bis zu 0,64 ‰ reduzieren und dadurch das Training der Netze signifikant beschleunigen.

Optimierung der
Performance

Das entwickelte Konstruktionsverfahren setzt voraus, dass im einschichtigen neuronalen Netz, das als Basis verwendet wird, nur relevante Verbindungen vorhanden sind. Hierzu wurde ein Magnitude-Based-Pruningverfahren entwickelt, dass in einem Schritt alle Verbindungen mit geringem oder keinem Einfluss entfernt. Dies führt gleichzeitig zu einer erneuten Beschleunigung des Trainings.

Die Verfahren wurden umfangreich und fundiert evaluiert. Die Evaluation umfasste die Phasen Konzeptbewertung, Evaluierung mit Falldaten und Praxiserprobung. In der Konzeptbewertung wurde das Konzept des Lernens aus Reparaturfällen mit der etablierten GFS hinsichtlich den Kriterien Einsatzbereich, technische Machbarkeit, Kosten, Leistungsfähigkeit und Nutzerakzeptanz verglichen. Beide Verfahren haben ihre Stärken und Schwächen, die jedoch größtenteils disjunkt sind. Es wurde daher eine Kombination beider Verfahren als sinnvoll identifiziert. Insbesondere die Adaptivität und die Fähigkeit, komplexe Fehlerbilder variantenindividuell lernen zu können, wurde als äußerst positiv bewertet.

Dreistufige
Evaluierung

Die GFS bietet dagegen den Vorteil individuell geführter Komponentenprüfungen. In der zweiten Phase der Evaluierung wurden die Verfahren zusammen mit einer Anwendungs- und Trainingssoftware prototypisch implementiert. Mittels eines automatisierten Evaluationsverfahrens und realer Falldaten wurden die einzelnen Verfahren bewertet und gegenübergestellt. Aussagen und Abschätzungen aus der Theorie konnten hier bestätigt werden. Zur Absicherung der Ergebnisse wurden abschließend eine manuelle Evaluierung mit 100 zufällig ausgewählten Falldaten sowie eine Praxiserprobung am Fahrzeug durchgeführt. Insgesamt wurde das komplexitätserweiterte neuronale Modell mit einer Trefferquote von 94 % allen anderen Modellen als überlegen befunden.

Im Abschluss der Arbeit wurden sinnvolle Erweiterungen und Modifikationen der Verfahren kurz diskutiert, aber auch ein konkretes Konzept zur schrittweisen Migration der heute eingesetzten GFS zu einem fusio-

nierten System aus beiden Verfahren vorgestellt. Letztlich ermöglicht der Einsatz lernender Verfahren mit der in dieser Arbeit vorgestellten Ausprägung, dass auch in Zukunft immer komplexer werdende Fahrzeuge mit zahlreicher werdenden Varianten und verkürzten Entwicklungszyklen effizient diagnostiziert und instand gesetzt werden können.

ANHANG

A Anforderungen und Analysen

A.1. Allgemeine Anforderungen und Nutzer der Fahrzeugdiagnostik

Zusammengefasst können die Nutzer und deren Anforderungen, wie in Abbildung A.1 dargestellt, konkretisiert werden. Nachfolgend werden die einzelnen Punkte kurz erörtert.

Abbildung A.1.: Anforderungen und Nutzer der Fahrzeugdiagnostik [Sup05]

Ein bedeutender Nutzer der Fahrzeugdiagnostik ist bereits die *Entwicklung*. Bevor neue Steuergeräte im Verbund bzw. in ersten Prototypenfahrzeugen getestet werden, werden sie zuvor auf sogenannten Breadboards¹ meist in Kombination mit HIL-Systemen² geprüft. Für die *Funktionsabsicherung*, so wird die Prüfung einer Funktion auf Korrektheit bezeichnet, wird hier u. a. die Eigendiagnostikfähigkeit verwendet, die somit Aufschluss über noch fehlerhafte oder nicht implementierte Funktionen gibt. Weiterhin helfen Diagnostikfunktionen wie das Lesen von Messwerteblocks, Einblick in innere Betriebszustände und Messwerte zu bekommen, wodurch die Fehlersuchzeit für potentielle Fehler enorm erleichtert und verkürzt wird (vgl. Tabelle D.4). Ebenso können spezielle Entwicklungsfunktionen über die Diagnostikchnittstelle aktiviert werden. Hier ist speziell die Möglichkeit des Softwareupdates über die Diagnostikchnittstelle eine wichtige Anforderung, insbesondere

¹ Unter Breadboardaufbauten werden erste provisorische Inbetriebnahmen auf Steckbrettern bezeichnet.

² HIL steht für Hardware in the Loop. Für weiterführende Literatur wird auf [Gmb07] verwiesen.

für den Test neuer korrigierter Softwareversionen. Die Verbindung dieser Funktion mit der Fahrzeugdiagnostik verwundert zunächst, lassen sich doch Softwareupdates nur schwer mit dem in Abschnitt 2.1 eingeführten Diagnostikbegriff vereinbaren. Die wahrscheinlichste Begründung ist vermutlich Pragmatismus, insofern wie das Diagnostikinterface zusammen mit den Diagnostikservicefunktionen der Steuergeräte der einzige etablierte Weg ist, von einer externen Stelle alle Steuergeräte zu erreichen. Auch die Eingliederung mancher Funktionen (wie z. B. die Steuergerätekodierung) in die Diagnostikservicefunktionen lässt sich nur so befriedigend erklären.

In den späteren Phasen der Entwicklung, wie erste Integrationstests und Prototypenfahrzeuge, bleibt die Diagnostikfunktionalität ein wichtiger Bestandteil, insbesondere für die Funktionsabsicherung oder aber der Fehlersuche. Ein weiterer Punkt ist die Erfüllung *gesetzlicher Anforderungen* wie z. B. OBD³. Für OBD müssen bestimmte abgasrelevante Fehlerzustände wie eine defekte Lambdasonde oder ein undichtes Tanksystem erkannt werden. Die Überprüfung dieser Fehlererkennungsfunktionen muss ebenfalls in der Entwicklung durchgeführt und abgeschlossen werden.

Gehen die Fahrzeuge in die *Produktion*, ist auch hier die Fahrzeugdiagnostik ein elementarer Bestandteil, insbesondere für die *Inbetriebnahme*, zu der die Überprüfung der Fehlerfreiheit sowie das Flashen der passenden Softwareversionen gehören. Aber auch für den Fall, dass Produktionsfehler wie ein nicht verbundener Stecker auftreten, soll die Fahrzeugdiagnostik eine schnelle und effiziente Fehlersuche ermöglichen. Ebenfalls am Ende der Produktionslinie werden die Steuergeräte, je nach *Ausstattungsvariante*, also verbauten Komponenten wie Automatik- oder Schaltgetriebe, Klimaanlage, HMI-Variante, usw. passend *kodiert*, das heißt für den umgebenden Kontext konfiguriert. Vor der Auslieferung dienen ebenfalls wieder die Diagnostikservicefunktionen zur sogenannten *Bauzustandsdokumentation*, in der die verbauten Komponenten sowie deren Hardware- und Softwareversionsnummern inkl. eines Ereignisspeicherauszugs protokolliert werden. Diese Funktion wird im Übrigen auch in der Entwicklung zur Dokumentation von Testfahrten von Prototypenfahrzeugen konsequent verwendet. Eine Besonderheit kommt in der Produktion noch hinzu - die Zeit. Nach einigen Fertigungsschritten ist eine Diagnostik für die Absicherung der Korrektheit notwendig. Das bedeutet jedoch, dass die Zeit für die Diagnostik sehr direkt in die Fertigungszeit (F-Zeit) eingeht und damit kostenrelevant ist. Sicherlich ist auch in den Werkstätten die Diagnostikzeit von Interesse, in der Produktion geht es jedoch um Sekunden.

Nach der Auslieferung der Fahrzeuge dient die Fahrzeugdiagnostik nun als Informationslieferant der *Qualitätssicherung*. Zumindest für Gewährleistungsfälle, wird immer ein Diagnostikprotokoll, also ein vollständiger Ereignisspeicherauszug, zum OEM geschickt. Aus der Summe der Diagnostik- und Gewährleistungsprotokolle können damit im Rahmen der Qualitätssicherung verschiedene *Analysen* zu Fehlerhäufigkeiten von Komponenten usw. durchgeführt werden. Gegebenenfalls werden dann sogenannte Fehlerabstellmaßnahmen und Verbesserung eingeleitet. Die Auf-

³ OBD = Onboard-Diagnostik, ist eine seit den 1980er Jahren vorgeschriebene und genormte integrierte Diagnostik für abgasrelevante Systeme. Weiterführende Literatur hierzu ist in [WR06, S. 636] zu finden.

zeichnungen geben letztlich auch eine *Reifegradspiegelung* bzw. *Zustandserfassung* einer Fahrzeugflotte wieder.

Sollten letztlich doch Defekte an einem Fahrzeug auftreten, ermöglicht die Fahrzeugdiagnostik im *Service* eine effizientere *Fehlersuche* und *Ursachenermittlung*, indem Mess- und *Prüffunktionen* zur Verfügung gestellt werden. Auch für die *Fehlerbeseitigung* werden Diagnostikservicefunktionen benötigt, wenn beispielsweise ein Kombiinstrument getauscht werden soll und hierfür sämtliche Einstellungen wie Kilometerstand, Wegfahrsperrung usw. zuerst ausgelesen und anschließend in die neue Komponente wieder eingeschrieben werden müssen.

A.2. Anforderungen an System und Konzept

Nachfolgend findet sich eine zu Anforderungsgruppen komprimierte Darstellung der für diese Arbeit relevanten Anforderungen. Die Äquivalenz der nachfolgend angeführten Punkte mit den Anforderungen in [Kriar] ist in Klammern beginnend mit dem Literal *K* und die Äquivalenz mit den Anforderungen in [Mül08] beginnend mit *L* gekennzeichnet.

A_{Gr1}–Symptomvarianten Als Symptome soll das System Ereignisspeichereinträge (DTCs) und Kundenwahrnehmungen (DISS-kodiert⁴) berücksichtigen (K2, K3, K4, L30).

A_{Gr2}–Fahrzeugvarianten Das System soll verschiedene Fahrzeugvarianten berücksichtigen. Hierfür sollen Kontextinformationen wie Verbauliste sowie Soft- und Hardwareversionen der Steuergeräte berücksichtigt werden (K6 u. K7 in Teilen).

A_{Gr3}–Geringer Erstellaufwand Die Konstruktion und das Training des Lernsystems soll vollautomatisch erfolgen (K9, L33). Weiterhin soll der hierfür nötige Vorbereitungsaufwand, wie beispielsweise manuelle Datenverknüpfungen, möglichst gering sein.

A_{Gr4}–Geringer Wartungsaufwand Der Wartungsaufwand der Wissensbasis soll möglichst gering sein. Das bedingt implizit eine automatisierte Optimierung der Wissensbasis wie die automatisierte Verarbeitung neuer Reparaturfälle (L36).

A_{Gr5}–Feedbackgeschwindigkeit Die Zeit vom Erhalten eines neuen Reparaturfalls bis zum Propagieren der daraus optimierten Wissensbasis in die Servicewerkstätten soll weniger als zwei Wochen betragen.

A_{Gr6}–Gewichtete Liste mit Diagnose-Hypothesen Das System soll aufgrund der vorhandenen Fehlersymptomen sowie der gelernten Wissensbasis eine mit Vertrauenswerten gewichtete Liste mit Fehlerhypothesen (als defekt vermutete Komponenten) generieren (K1, K10 implizit, K11 implizit, K12 implizit, L31, L32).

⁴ Unter DISS-kodiert wird die Kodierung einer Kundenwahrnehmung/Werkstattwahrnehmung nach einem hierarchisch organisierten Symptom-/Ursachen-Katalog verstanden.

A_{Gr}7–Komplexe Symptomatiken Das System soll mit komplexen Symptomatiken umgehen können (K10). Dies korreliert mit dem aus der Bewertung verwendeten Terminus „*Fehlerdiagnostik komplexer Fehler*“.

A_{Gr}8–Multiple Fehler Das System soll mit multiplen Fehlern umgehen können (K14). Diese gehören ebenfalls in die Kategorie *komplexe Fehler*.

A_{Gr}9–Schnelle Anwendbarkeit Die Anwendung des Systems für eine Symptomatik soll weniger als 10 sek. benötigen (K19, K20).

A_{Gr}10–Manuelle Modifizierbarkeit Die Diagnostikwissensbasis soll manuell erweitert oder modifiziert werden können, um z.B. Änderungen in der Produktion zu berücksichtigen zu können (L34).

A_{Gr}11–Robustheit Fehlerhaft oder unzureichend erfasste Reparaturfalldaten dürfen keine dauerhaft „*fehlerhafte*“ Wissensbasis zur Folge haben (L35).

Nicht berücksichtigt werden die Anforderungen K5 (Verwendung eines Bayes'schen Netzes), K8 (Plausibilisierung der Fehlerhypothesen mit der Fahrzeugvariante), K13 (Funktion mit fehlende Ausfallwahrscheinlichkeiten) und K15; K16; K17; K18 (iteratives Vorgehen und Berücksichtigung von Strukturinformationen).

A.3. Zusätzliche Anforderungen an die Datenbasis

Aus den Betrachtungen der Datenbasis, insbesondere der manuellen Untersuchung in Kapitel 5, lassen sich einige zusätzliche Anforderungen an die Datenerfassung sowie an die Datenqualität ableiten. Einige Punkte daraus werden im Folgenden kurz diskutiert.

Verknüpfung der Daten Zwischen den erfassten Datensätzen muss eine eindeutige Verknüpfung bestehen, unabhängig davon, ob die Erfassung durch unabhängige Systeme geschieht.

Tracking des Reparaturers In einigen Fällen ist der Erfolg nicht unmittelbar nach der Reparatur bekannt. Das gilt insbesondere für sporadisch auftretende Fehler, die durch den KD-Mitarbeiter aus Zeitgründen nur schwer reproduziert werden können. Hierfür ist eine Art Tracking des Reparaturers notwendig. In Rahmen dieser Arbeit wurde die Annahme getroffen, dass der Kunde bei einer erfolglosen Reparatur binnen 2 Wochen wieder in die Werkstatt kommen würde. Während das für Gewährleistungsfälle noch hinreichend erfüllt sein mag, kann das für nicht-Gewährleistungsfälle nicht angenommen werden.

Qualität der DISS-Kodierung Die Qualität, Eindeutigkeit, Detailtiefe und Zuverlässigkeit der DISS-Kodierung für die Kundenwahrnehmung sowie für die Werkstattwahrnehmung muss stark verbessert werden. Bisher werden DISS-Kodierungen häufig in zu geringer Detailtiefe durchgeführt. Weiterhin wird ein und dieselbe Kundenwahrnehmung durch verschiedene Kodierungen protokolliert. Dies kann im Einzelfall daran liegen, dass eine passende Kodierung nicht vorhanden ist und daher die „*nächst Beste*“ verwendet wird. Eine größere Detailtiefe des Kataloges ist folglich gefordert. Dies sollte unbedingt mit einer

besseren Bedienbarkeit gepaart sein. Weiterhin wird des Öfteren die falsche Kodierung gewählt.

Qualität der Ursachenkennzeichnung Auch die Qualität der Ursachenkennzeichnung muss verbessert werden. Trifft eine „*Reparaturlösung*“ nicht auf eine Arbeitsposition oder ein Ersatzteil zu, wird häufig die Geführte Fehlersuche als Ursache gekennzeichnet. Beispiel für solche Fälle sind das Ändern einer Steuergerätekodierung oder eine Justierung mit Hilfe des Testers.

Abbildbarkeit der Lösung Generell muss aus jedem Reparaturfall hervorgehen, welche Reparaturmaßnahme zur Lösung geführt hat. Im Falle eines Komponenten- bzw. Teiletauschs ist dies mit Hilfe der Ursachenkennzeichnung ausreichend gegeben. Eine Herausforderung sind Reparaturmaßnahmen wie das Instandsetzen von Stecker, Leitungsabschnitten, Kodierung von Steuergeräten und ähnlichem. Dies kann im Moment nicht in genügender Detailtiefe, weder durch die Arbeitspositionen noch durch die DISS-Kodierungen geleistet werden. Hier muss nachgebessert werden. Die beiden vorhergehend aufgeführten Punkte können hier abgeleitete Anforderungen und damit mögliche Lösungsansätze darstellen.

Problemverständnis im Kundendienst Die vorhergehend geforderte Abbildbarkeit der Lösung bedingt zwangsläufig ein Problemverständnis des bedienenden Personals. Die Protokollierung sollte zwar so automatisiert und einfach wie möglich sein, dennoch ist eine gewisse Sorgfalt des Anwenders unabdingbar. Diese kann nur dann erreicht werden, wenn ein Bewusstsein für die Notwendigkeit dieser Sorgfalt beim Anwender vorhanden ist.

Eindeutigkeit der DTCs Bedingt durch den Reparaturverlauf werden während einer Reparatur häufig mehrere Diagnostikprotokolle mit DTCs erzeugt. Nach einem Steuergerätetausch müssen beispielsweise mögliche Ereignisspeichereinträge gelöscht werden und Anpassungen durchgeführt werden. Solche Protokolle sind bisher ebenfalls in den Daten vorhanden und können nur teilweise nicht vollständig von einander getrennt werden. Für ein Lernsystem ist vor allem das Symptombild interessant, das vor der Reparatur anzutreffen ist. Dies muss folglich eindeutig aus den Datensätzen abgeleitet werden können. Dies trifft auch auf die Verbauliste zu. Bei einem Steuergerätetausch sind hier häufig beide Steuergeräte und deren Softwareversionsnummern zu finden. Es kann hier nicht mehr getrennt werden, welches Steuergerät nun ersetzt wurde.

Die hier aufgeführten Punkte haben keinen Anspruch auf Vollständigkeit und können je nach Auslegung und Lösungsansatz der genannten Anforderungen noch variieren. Mögliche Lösungsansätze sind stark unternehmenspolitisch bedingt und können daher nicht an dieser Stelle diskutiert werden. Für die Betrachtungen in dieser Arbeit wie auch den Prototyp wurden die vorhandenen Datensätze mit den eben beschriebenen Mängeln verwendet.

A.4. Analyse diverser datenerzeugender Prozesse

Die Reparaturfall-Definition 4.1 berücksichtigt bisher nicht die Tatsache, dass Fahrzeugsysteme je nach Modell und Variante sehr verschieden sein können. Mit der Annahme, dass dies ebenfalls einen Einfluss auf das resultierende Diagnostik-Modell hat, wird eine Analyse diverser Systeme und den daraus entstehenden Daten notwendig.

Nicht nur in Definition 4.1, sondern in vielen weiteren Arbeiten über Methoden und Algorithmen werden die notwendigen Daten als gegeben vorausgesetzt, d.h. es ist dabei nicht von Interesse, wie diese Daten entstehen. Während dies für die Grundlagenforschung akzeptabel sein mag, hat die industrielle Forschung in vielen Fällen den Anspruch, die entwickelten Ideen, Methoden und Verfahren für bislang unerschlossene Anwendungen nutzbar zu machen. Dies setzt jedoch die Kenntnis der datenerzeugenden Prozesse voraus, da hier Unterschiede existieren, die für die Weiterverwendung von entscheidender Bedeutung sind, wie die folgende Analyse aufzeigt. Daten werden durch Prozesse erzeugt, die wiederum an ein oder mehrere Systeme Σ gebunden sind. Ein System Σ ist nach Definition 2.2 (vgl. Kapitel 2.1) die Gesamtheit aus einer Reihe von Einzelteilen bestehender funktionaler Einheit, die zur Ausführung einer Reihe von Aufgaben dient. Ein System ist gegen seine Umwelt abgegrenzt und in Komponenten K_Σ zerlegbar, zwischen denen Wechselwirkungen (Beziehungen $\Xi \in \Xi_\Sigma$) bestehen. Ein System ist gegen seine Umwelt offen, wenn es auf sein Umfeld einwirkt und dieses Umfeld auf das System einwirkt.

Ein Prozess $\mathfrak{P}$ ist deterministisch, wenn jeder Zustand $Z_{i,t}$ von $\mathfrak{P}$ kausal vom vorhergehenden Zustand $Z_{i,t-1}$ bestimmt wird. Für einen Prozess bedeutet das, dass $\mathfrak{P}$ einen funktionalen Zusammenhang zwischen den Eingangsgrößen X und den Ausgangsgrößen Y des Systems Σ beschreibt, indem jede Eingangs-Variablen-Konfiguration genau einer Ausgangs-Variablen-Konfiguration Z zugeordnet ist. Die erzeugten Daten Z folgen dabei einer bestimmten Verteilung P^5 .

Definition A.1 (Deterministischer Prozess) *Ein Prozess $\mathfrak{P}$ ist deterministisch, wenn $\mathfrak{P}$ einen funktionalen Zusammenhang zwischen Eingängen und Ausgängen des Systems beschreibt, also jeder Eingangs-Variablen-Konfiguration genau eine Ausgangs-Variablen-Konfiguration zugeordnet ist. Sonst ist der Prozess nicht-deterministisch.*

In der Realität muss hierfür das zugrunde liegende System in einer sehr hohen Detailtiefe bekannt sein, was in der Praxis jedoch nur selten der Fall bzw. oftmals unmöglich ist. In einer statistischen Betrachtungsweise, die für das hier gegebene Szenario sinnvoll erscheint, werden die Daten durch einem einzelnen Prozess $\mathfrak{P}$ in einem einzelnen System Σ erzeugt. In Abbildung A.2, die der Darstellung nach [CM98] folgt, bezeichnet X einen Vektor beobachtbarer Eingangs-Variablen, Y einen Vektor nicht beobachtbarer Eingangs-Variablen und Z den Vektor der Ausgangs-Variablen. Die durch das System Σ erzeugten Daten Z haben dann die Verteilung $P(Z|X,Y,\Sigma)$. Da jedoch Y nicht beobachtbar ist, und das System Σ als konstant

⁵ Aus der Systembetrachtung folgt die Existenz einer solchen Verteilung, die auch in der Praxis gegeben ist (vgl. [Bor04])

Abbildung A.2.: Datenerzeugung durch ein einzelnes System

angenommen wird, zielt die resultierende Lernaufgabe auf eine Modellbildung bzgl. der Verteilung $P(Z|X)$ ab.

In den vorhergehend angestregten Betrachtungen wird jede Reparatur r als ein abgeschlossener Prozess betrachtet. Das dem Prozess zugrunde liegende System Σ ist hierbei das jeweilige Fahrzeug an sich. Nun werden keinesfalls immer baugleiche Fahrzeuge repariert, vielmehr sind alle Fahrzeuge, selbst wenn sie von derselben Klasse sind, aufgrund der möglichen Varianten unterschiedlich. Dies wird angelehnt an [Bor04] durch Abbildung A.3 veranschaulicht.

Abbildung A.3.: Verschiedene Systemvarianten

Betrachtet werden müssen folglich n Daten generierende Systeme $\Sigma_1, \dots, \Sigma_n$ mit $\Sigma_i \neq \Sigma_j$, $i, j \in [1, n]$. Die Varianz der dargestellten Systeme kann dabei auch als fahrzeugspezifische Eigenschaften betrachtet werden. Sie sollen im Folgenden unter dem Systemkontext $C \subseteq C_\Sigma$ zusammengefasst werden, wobei C_Σ die Menge aller Systemkontexte darstellt. Werden nun alle beobachtbaren Größen $C_{b,i}$ eines System-Kontexts C_i als Bestandteil von X und alle nicht beobachtbaren Größen $C_{nb,i}$ als Bestandteil von Y betrachtet, so können die Systeme Σ_i , $1 \leq i \leq n$ in ein resultierendes System Σ^* mit $X^* = X \cup C_{b,i}$, $1 \leq i \leq n$ zusammengefasst werden. Daraus folgt, dass auch die Variantenvielfalt als Bestandteil der Lernaufgabe betrachtet werden kann bzw. muss.

Im Kontext dieser Arbeit sind mit Systemen ausnahmslos Fahrzeugsysteme gemeint, so dass anstelle vom System-Kontext im Folgenden vom Fahrzeugkontext gesprochen wird. Für einen konkreten Reparaturfall r sind dabei die beobachtbaren Größen $C_{b,i}$ relevant, die passender mit C_R bezeichnet werden.

B Zusatzinformationen zur Akquise und Aufbereitung der Falldaten

B.1. Akquise realer Falldaten

Wie zu erwarten, war zu Beginn der Datenakquisition keine umfassende und integrierte Erfassung von Reparaturfällen vorhanden. Eine solche war ursprünglich nicht notwendig, wie auch eine Erweiterung der Offboard-Diagnostik auf lernende Verfahren bis dahin nicht Teil der Konzernstrategie war. Dennoch wurden sie für diverse statistische Auswertungen aufgezeichnet. Zu den verfügbaren Daten gehörten unter anderem

- Gewährleistungsanträge,
- Prüfprotokolle der Diagnostik-Tester,
- Kundenwahrnehmung und
- Werkstattwahrnehmung.

Dennoch gestaltete sich die Akquise dieser Daten als unerwartet aufwändig. Die Ursachen hierfür waren unter anderem die Vielzahl der Systeme, an die die Daten übermittelt wurden und die Datensätze, die zwischen den Systemen nicht verknüpft waren. So wurden zum Beispiel die Prüfprotokolle¹ der Tester an das System EHDWTE² übermittelt, die Abrechnungsdaten für die Gewährleistung an das System SAGA² und die Kundenbeanstandungen an das System DISS (vergleiche [Hüb02]). Zwischen den Datensätzen³ bestanden, wie bereits erwähnt, keine Verbindungen. Dies war unter anderem auch datenschutzrechtlich gefordert, um eine zu weitreichende Überwachung der Werkstätten zu verhindern. Eine Verknüpfung der Daten konnte im Rahmen des Projekts lediglich über komplexe SQL-Skripte erzeugt werden, in denen, abhängig von der Fahrgestellnummer, Händlernummer und Datum, die Datensätze gegenseitig plausibilisiert wurden. Auch dies wurde dadurch erschwert, dass zwischen Kundenbeanstandung, Tester-Prüfprotokollen und Gewährleistungsantrag einige Tage, teilweise auch Wochen liegen konnten. Möglich wurde die Beschaffung einer hinreichenden Menge von Datensätzen erst durch die Unterstützung der Abteilung KGQS/5, der an dieser Stelle ausdrücklich gedankt werden soll.

Neben der Verknüpfung der Datensätze waren auch einige Filteroperationen notwendig. Um beispielsweise eine eindeutige Zuordnung der Symptome zu den

¹ Unter Prüfprotokollen werden die Aufzeichnungen einer Diagnostiksession (z. B. der Geführten Fehlersuche (GFS)) durch den Diagnostik-Tester bezeichnet. In diesen Protokollen sind die Ereignisspeichereinträge (DTCs) sowie die von der GFS ausgeführten Programme/Messwertaugnisse enthalten.

² Woher die Benennung der Systeme stammt, konnte nicht abschließend geklärt werden.

³ Ein Datensatz ist eine zusammengefasste Einheit von Datenfeldern (vgl. [Sch07, S. 35]).

durchgeführten Reparaturmaßnahmen zu erreichen, wurden Reparaturfälle mit mehr als einer Kundenbeanstandung ignoriert. Andernfalls konnte nicht zugeordnet werden, welche Symptome und welche Reparaturmaßnahmen zu welcher Beanstandung und damit zu welchem Fall gehörten. Weiterhin war es wünschenswert, nur Fälle zu betrachten, in denen der Reparatur Erfolg bekannt ist. Hierfür wurden alle Fälle herausgefiltert, in denen der Kunde binnen zwei Wochen erneut eine Werkstatt aufsuchte. Es wurde dabei angenommen, dass im Gewährleistungsfall der Kunde bei einer nicht erfolgreichen Reparatur in den meisten Fällen wiederkommen würde, zumal für ihn die Reparatur kostenfrei war. Weiterhin ist es in der Praxis üblich, dass ein Diagnostik-Tester im Laufe der Reparatur mehrmals mit dem Fahrzeug verbunden wird, um die verbleibenden Ereignisspeichereinträge auszulesen bzw. zu löschen. Diese Protokolle sind ebenfalls mit in den einzelnen Datenbanken. Daher wird hier nur der erste Datensatz einer Reparatur verwendet. Es kann allerdings nicht immer davon ausgegangen werden, dass vor dem ersten Auslesevorgang nicht bereits Reparaturmaßnahmen durchgeführt wurden, die weitere Ereignisspeichereinträge zur Folge hatten. Weitere Filterkriterien sind nachfolgend unter D.3 zu finden.

B.2. Attributbeschreibungen

In der akquirierten Datenbasis stehen nachfolgend aufgeführte und kurz erklärte Attribute für Symptome, Reparaturmaßnahmen, Fahrzeugkontext und Meta-Informationen zur Verfügung:

Symptom-Attribute

Ereignisspeichereinträge (DTCs) Die meisten Steuergeräte besitzen eine On-board-Diagnostik. Diese legt, sofern ein Defekt oder unplausibler Zustand erkannt wurde, einen Ereignisspeichereintrag im Speicher des Steuergeräts ab, der später über das Diagnostikinterface ausgelesen werden kann. Die Bezeichnung „*Ereignis*“ wurde bewusst gewählt, da ein solcher Eintrag nicht zwangsläufig ein Fehler bedeutet. *DTC* ist dabei die gebräuchliche (englische) Abkürzung.

Kundenwahrnehmungen (S-DISS) Der übliche Reparaturannahmeprozess in einer Servicewerkstatt sieht vor, dass der Kunde zunächst den von ihm wahrgenommenen Defekt des Fahrzeugs beschreibt. Diese sogenannte *Kundenwahrnehmung* wird anschließend mittels eines *DISS* genannten Katalogs mit über 18.000 möglichen Wahrnehmungen kodiert. Von den möglichen Kodierungen wird in der Praxis jedoch nur ein Bruchteil verwendet (vgl. Tabelle 5.1).

Reparaturmaßnahmen-Attribute

Ersatzteilnummern (ET) Aus den Abrechnungsdaten geht hervor, welche Komponenten getauscht wurden. Für diese existiert eine entsprechende Ersatzteilnummer.

Ersatzteilrumpfnummern (ETRN) Während sich ein Ersatzteil mit dessen Ersatzteilnummer auf ein spezielles Fahrzeugmodell, deren Variante und Farbe bezieht, bezeichnet die Ersatzteilrumpfnummer das übergeordnete Bauteil ohne modell- und variantenspezifische Ausprägung (z. B. Lenkrad (Ersatzteilrumpf); Varianten: Golf MDJ 2005, Leder schwarz, grau, etc.).

Arbeitspositionen (AP) Um einer willkürlichen Abrechnung von Fehlersuch- und Reparaturzeiten zu begegnen, wurden Kataloge mit festgelegten Arbeitsschritten inkl. dem möglichen Zeitrahmen eingeführt. In den Abrechnungsdaten sind entsprechend die durchgeführten Arbeitspositionen aufgelistet.

Werkstattwahrnehmung (M-DISS) Analog zur Kundenwahrnehmung, ist es dem KD-Mitarbeiter möglich, anhand des bereits erwähnten DISS-Katalogs, den Defekt zu codieren. Hierfür stehen über 64.000 mögliche Defekte und deren Kodierung zur Verfügung, von denen jedoch in der Praxis auch hier nur ein Bruchteil verwendet wird (vgl. Tabelle 5.1).

Kundendienstnummer (KDNR) KDNR steht für Kundendienstnummer. Es handelt sich hier um eine Katalogisierung nach Schadensgruppen wie z. B. Fensterheber hinten, Dreipunktgurt oder Zweimassenschwungrad. Diese sind speziell für die unternehmensinterne Organisation sinnvoll (Verantwortliche, etc.). Es stehen 2.420 verschiedene Schadensgruppen zur Verfügung.

Fahrzeugkontext-Attribute

Steuergerät-Hardwareteilenummer Diese bezieht sich auf die reine Hardware des Steuergeräts. Die Software wird hier nicht betrachtet.

Steuergerät-Softwareteilenummer Prinzipiell ist jedem Steuergerät eine darauf abgestimmte Software mit einer entsprechenden Teilenummer zugeordnet, wobei diese Zuordnung nicht zwangsläufig eine 1:1-Beziehung sein muss. Warum hier von einer Teilenummer und nicht von einer Version gesprochen wird, konnte im Rahmen der Arbeit nicht abschließend geklärt werden.

Steuergerät-Softwareversionsnummer Für jede Softwareteilenummer können mehrere Versionen existieren, wobei diese speziell für Weiterentwicklungen im Sinne von Fehlerbeseitigungen verwendet wird.

Aggregiertes Attribut Die drei Attribute zum Fahrzeugkontext können zu einem Attribut *Fahrzeugkontext* aggregiert werden. Dadurch wird für das spätere Diagnostik-Modell eine Merkmals-Reduktion erreicht.

Meta-Informationen-Attribute

Reparaturdatum Als Meta-Informationen standen u. a. das Reparaturdatum, die Werkstattpartnernummer, Beanstandungsnummer, Reparaturkosten zur Verfügung. Mit Ausnahme des Datums spielen diese Attribute hier jedoch eine untergeordnete Rolle.

B.3. Kodierung von Merkmalen

In Definition 5.1 wurden Symptome $s \in S_\Sigma$ wie auch Komponenten $k \in K_\Sigma$ als vorhanden/nicht vorhanden bzw. defekt/nicht defekt betrachtet. Eine solche dichotom nominale Betrachtungsweise von Merkmalen ist für die beschriebenen statistischen und neuronalen Modelle auch zwingend notwendig. In der Praxis sind jedoch nicht alle Merkmale dichotom, wie dies speziell für Symptomzusatzinformationen wie z.B. die *Fehlerart*⁴, aber auch für Kontextinformationen $C \in C_\Sigma$ gilt. Andere Merkmale sind zwar dichotom nominal, dafür jedoch hierarchisch partitioniert. Darunter fallen z.B. die sogenannten DISS-Kodierungen⁵ für Kunden- und Werkstattwahrnehmungen.

B.3.1. Nicht-dichotome nominale Merkmale

Unter diese Kategorie fallen insbesondere Symptomzusatzinformationen, wie sie für Ereignisspeichereinträge (DTCs) typisch sind. Ein Beispiel der zum eigentlichen Fehlercode weiter verfügbaren Informationen zeigt Tabelle B.1 (für eine Erklärung der einzelnen Elemente wird auf den Anhang verwiesen). Es kommt hinzu, dass nicht alle Details für alle Steuergeräteversionen und -arten verfügbar sind. Trotzdem

<i>Information</i>	<i>Bemerkung</i>
Adresse des Steuergerätes	HEX-Kodierte (Diagnose-)Adresse
Fehlercode	Der P-Code sowie weitere herstellerspezifische Codes
Fehlerart	Unterscheidung der Fehlerkategorie, z. B. „ <i>Kurzschluss nach Masse</i> “
Fehlerstatus	Weitere Unterscheidung in „ <i>statisch</i> “ (immer vorhanden) und „ <i>sporadisch</i> “ (nicht immer vorhanden)
Auftrittszeitpunkt	Erster Auftrittszeitpunkt nach Borduhrzeit
Auftrittskilometerstand	Erster Auftrittszeitpunkt nach Bordkilometerstand
Verlernzähler	Anzahl Motorstarts seit dem letzten Auftreten des Fehlers

Tabelle B.1.: Verfügbare Informationen zu einem Ereignisspeichereintrag

erfordert die maschinelle Auswertung eine Überführung in die später mit Definition 5.1 eingeführte dichotom nominale Form. Hierfür existieren prinzipiell zwei Möglichkeiten:

1. Definition je eines Einzelmerkmals (Symptom s_x bzw. Komponente k_x) für jede Kombinationsmöglichkeit der Detailinformationen
2. Jede Detailinformation wird als getrenntes Merkmal betrachtet

⁴ Erklärung siehe Tabelle B.1.

⁵ Unter DISS wird die Kodierung einer Kundenwahrnehmung/Werkstattwahrnehmung nach einem hierarchisch organisierten Symptom-/Ursachen-Katalog verstanden.

Letztere Möglichkeit reduziert zwar die Kardinalität der verfügbaren Merkmalsmenge S_Σ/K_Σ , ist jedoch bereits mit dem Auftreten von zwei oder mehr Ereignisspeichereinträgen gleichzeitig, die nicht dieselben Detailinformationen besitzen, ungeeignet. Weiterhin würden damit explizite Abhängigkeiten der Merkmale untereinander geschaffen, die weitere Nachteile mit sich bringen (der Sachverhalt, insbesondere von Symptomabhängigkeiten wird später in Abschnitt 6.3 und für neuronale Netze in Abschnitt 7.3 diskutiert). Es ist folglich notwendig, für jede Zusatzinformationskombination ein eigenes Merkmal zu erzeugen. Ein Beispiel anhand der in dieser Arbeit gewählten Kodierung zeigt Tabelle B.2.

Symptom: <i>S1_1_0116684__13</i>	
S1	Symptomtyp: Ereignisspeichereintrag
01	SG-Adresse: 01 = Motorsteuergerät
16684	Herstellerspezifischer Code: 16684 = Verbrennungsaussetzer erkannt
1	Fehlerkategorie: 1 = Oberer Grenzwert überschritten
3	Fehlerstatus: 3 = statisch

Tabelle B.2.: Beispiel zur Symptom-Kodierung (DTC)

Ebenfalls in die Klasse der nicht-dichotomen nominalen Merkmale fallen die Kontextinformationen $c \in C_\Sigma$. Auch hier ist eine Kodierung durch dichotome Merkmale gefordert. Jedoch existiert hier, im Gegensatz zu Zusatzinformationen, kein dichotomes „Trägermerkmal“ wie ein Symptom S oder eine Komponente K . Es wird hier daher die im Data Mining für nominale Variablen übliche „1-aus-n“-Kodierung verwendet, welche im Wesentlichen der zweiten anfangs vorgestellten Kodierungsvariante entspricht. Das bedeutet, es wird aus einer Kontextvariablen $c \in C_\Sigma$ jeweils ein Merkmal c_i für jeden Wert $\{(c_{ecu}, c_{version})\}$, den c annehmen kann, erzeugt. Ein Merkmal nimmt genau dann den Wert 1 an, wenn gilt $c = c_i$, ansonsten den Wert 0.

B.3.2. Hierarchische Merkmale

Während einige Zusatzinformationen in Ereignisspeichereinträgen, wie z.B. die Fehlerart, sehr zuverlässig verfügbar sind, trifft dies insbesondere für manuell eingegebene Symptome nicht zu. Typisch für manuell erfasst Symptome, das sind zumeist Kundenwahrnehmungen des Fehlers, ist eine schrittweise Verfeinerung der Wahrnehmung. Die Grundlage dafür sind häufig große, ebenfalls hierarchische Symptomkataloge (selbiges gilt für Werkstattwahrnehmungen). Ein Beispiel für ein solches Symptom ist in B.3 abgebildet. So wird mit der Hauptgruppe *Antriebsaggregat* begonnen. Anschließend wird die Wahrnehmung auf *Funktion* eingeschränkt und mit *Fahrverhalten (Motor)* weiter spezifiziert. Nachfolgend muss die Wahrnehmung gemäß dem vorgegebenen Katalog weiter verfeinert werden, was hier mit der Angabe *Aussetzer/Stottern* erfüllt ist. Eine weitere Verfeinerung geschieht jedoch nicht. Dies kann in der Praxis viele Gründe haben. Zu den häufigsten zählen mangelndes

Interesse, Zeitdruck oder eine fehlende Kodierungsmöglichkeit. Im Beispiel wäre

Symptom: S2 01 01 04 02 98	
S2	Symptomtyp: Kundenwahrnehmung
└01	Hauptgruppe: Antriebsaggregat
└01	Schadensklasse: Funktion
└04	Grobcode: Fahrverhalten (Motor)
└02	Mittelcode: Aussetzer/Stottern
└98	Feincode: Nicht angegeben

Tabelle B.3.: Beispiel zur Symptom-Kodierung (DISS)

z. B. die Feinkodierung mit „*bei kaltem Motor*“ o. ä. möglich gewesen.

Bedingt durch die unterschiedliche Detaillierung wird hier eine spezielle Verfahrensweise in der Merkmalskodierung notwendig. Die Aufteilung in Einzelmerkmale, je nach Kategorie (Hauptgruppe, Schadensklasse, Grobcode, etc.) ist aus denselben Gründen wie in der Ereignisspeichereintrags-Kodierung nicht sinnvoll. Ein Merkmal pro Kodierungsmöglichkeit ist folglich auch hier notwendig, wobei die Kategorie „*Nicht angegeben*“ explizit mit eingeschlossen wird. Ungünstig ist hierbei, dass für ein sehr detailliert angegebenes Merkmal s_x/k_x , im Beispiel also inklusive Feinkodierung „*bei kaltem Motor*“, implizit alle in der Hierarchie darüber stehenden Merkmale ebenfalls vorhanden sein müssten, z. B. das in Tabelle B.3 dargestellte Symptom. Diese müssen folglich ebenfalls als gegeben angenommen werden und entsprechend in die Menge der vorhandenen Merkmale S_I bzw. S_R oder K_R mit aufgenommen werden. Im Beispiel aus Tabelle B.3 würden neben dem Symptom $S2\ 01\ 01\ 04\ 02\ 98$ weiterhin die Symptome $S2\ 01\ 01\ 04\ 98$, $S2\ 01\ 01\ 98$ und $S2\ 01\ 98$ als vorhanden markiert werden. Eine ausführliche Kodierungstabelle, die auch für die Implementierung des Prototyps verwendet wurde, ist nachfolgend in D.4 abgedruckt.

B.4. Analyse von drei repräsentativen Fallbeispielen

Zum besseren Verständnis der Daten, der Beschaffenheit und Zuverlässigkeit werden hier drei konkrete Fallbeispiele von Reparaturfällen vorgestellt. Am Beispiel werden dabei auch einige für die Aufgabe relevanten Aspekte detaillierter erörtert.

In Tabelle B.4 ist der Datensatz eines realen Reparaturfalls vom 22.01.2007 dargestellt. Im ersten Abschnitt befinden sich die Informationen über den Reparaturfall selbst, wie Datensatz-ID, Datum, Informationen über das Fahrzeug (Modelljahr, Motor, Fahrgestellnummer (anonymisiert)) sowie der Kunden- und Werkstattfreitext⁶. Anschließend folgt eine Reihe von Symptomen und Reparaturmaßnahmen, die diesem Fall zugeordnet werden.

⁶ Mit dem Kundenfreitext wird den Werkstätten die Möglichkeit gegeben, auch außerhalb der fest vorgegebenen DISS-Kodierungen eine Kundenwahrnehmung als freien Text zu protokollieren. Der Werkstatttext verhält sich analog, hier werden jedoch häufig die Werkstattwahrnehmung des Defektes und/oder die Lösung des Falls protokolliert.

Im vorliegenden Reparaturfall kam der Kunde laut Kundenfreitext aufgrund einer Warnsignalfehlfunktion in den Kundendienst. Hierzu passt prinzipiell auch das verzeichnete DISS-Symptom. Bei genauerem Hinsehen entsteht jedoch der Verdacht, dass die DISS-Kodierung nicht passend gewählt wurde. Ein Blick in den DISS-Katalog zeigt, dass die Fehlfunktion aller Warnleuchten nicht kodiert werden kann, sondern anstatt dessen jede Warnfarbe einzeln kodiert werden müsste. Der Verdacht liegt also nahe, dass eine ähnliche DISS-Kodierung verwendet wurde, die den Fall jedoch nicht vollständig bzw. verfälscht beschreibt. Solche „*Fehl-Kodierungen*“ können in den Datensätzen öfters beobachtet werden (vgl. Tabelle 5.2 - S-DISS zu ungenau, S-DISS falsch kodiert). Ein Blick in die Reparaturmaßnahmen lässt vergebens nach der Lösung des Falls suchen. Als Ursache wurde die „*Geführte Fehlersuche*“ gekennzeichnet. Dies ist wenig aussagekräftig. Aufschluss gibt der Werkstattfreitext, der ist jedoch nicht automatisiert auswertbar. Weiterhin existiert eine DISS-Kodierung für die Werkstattwahrnehmung (Maßnahme 3). Aber auch diese ist falsch kodiert und passt nicht zum Werkstattfreitext. Das kann in den Datensätzen des Öfteren beobachtet werden (vgl. Tabelle 5.2 - M-DISS zu ungenau, M-DISS falsch kodiert).

Aus den vorhergehend genannten Gründen muss dieser Fall als „*inkongruent*“ eingestuft werden. Inkongruente Fälle sind für ein Lernsystem insofern eine Herausforderung, weil aus ihnen keine sinnvollen Zusammenhänge gelernt werden können. Weiterhin verfälschen sie die in Kapitel 9.3 vorgestellte automatisierte Evaluierung, da eine vom Diagnostik-System vorgeschlagene (richtige) Lösung zwangsläufig als falsch eingestuft wird. Dennoch lassen sich solche Fälle weder vermeiden noch herausfiltern, so dass ein Lernsystem hiermit umgehen können muss. Diese notwendige Eigenschaft wird als *Robustheit* bezeichnet

Eine weitere nennenswerte Beobachtung stellt das vorhandene DTC-Symptom (Symptom 1) dar. Für diesen DTC gibt es keinen erklärbaren Zusammenhang zur Kundenbeanstandung. Vielmehr ist er Ursache eines Entwicklungsfehlers, der durch eine Feldaktion mittels Softwareupdate Stück für Stück korrigiert wird. Das wirft eine weitere Herausforderung auf. Hierdurch existieren gleich zwei zu lösende Reparaturaufgaben in einem Reparaturfall. Eine eindeutige Zuordnung, welches Symptom zu welcher Reparaturaufgabe gehört, sowie welche Reparaturmaßnahme zu welcher Reparaturaufgabe gehört ist somit, zumindest in einer isolierten Betrachtung, nicht möglich. Solche Fälle sind jedoch mit 4 % vergleichsweise selten (vgl. Tabelle 5.2 - 2 od. mehr verschied. Fälle gleichzeitig).

Der in Tabelle B.5 gezeigte Fall kann durchaus als Musterfall bezeichnet werden. Das DISS-Symptom (Symptom 3) ist hinreichend genau kodiert und passt mit dem Kundenfreitext sowie den beiden DTCs zu einem eindeutigen Symptombild zusammen. Auch die protokollierten Maßnahmen stellen, auch jede für sich, einen Hinweis zur Lösung dar. Ebenfalls die DISS-Maßnahme (Maßnahme 3) ist detailliert genug kodiert, um als Lösungshinweis zu fungieren. Es gibt weiterhin keine Widersprüche im Datensatz. Dieser Fall ist *kongruent* und damit sehr gut lernbar. Fälle mit diesem vollständigen Symptombild, bestehend aus DTC und DISS zusammen mit einem vollständigen Maßnahmenbild, bestehend aus Arbeitsposition, Ersatzteil und DISS-Kodierung, sind mit 6 % (ermittelt über die manuelle Auswertung) aller-

Reparaturfalldaten	
Reparaturfall-ID	21
Reparaturannahmedatum	39104
Modelljahr (MDJ)	2005
Motorkennbuchstabe (MKB)	BKC
Verkaufsmodellcode	1K1131
Fahrgestellnummer (anonymisiert)	WVWZZZ1KXXXX21
Beanstandung, Kundenfreitext	Warnsignale ohne Funktion
Werkstattfreitext	Softwarestand f. Kombi geprüft u. Softwareanpassung durchgeführt.
Symptom 1	
Kodiert	1_0118104__01
Symptomtyp	DTC
Steuergeräteadr. und -bezeichnung	01 - Motorelektronik
Fehlercode und -beschreibung	118104 - Datenbus Antrieb unplausible Botschaft von Lenksäulenelektronik
Fehlertypcode 1 und -bezeichnung	0 - Keine Anzeige
Fehlertypbezeichnung 3	sporadisch
Symptom 2	
Kodiert	20601080602__
Symptomtyp	S-DISS
Bezeichnung	Fahrzeugelektrik - Funktion - Bereich Warnungen akustisch - Warnsummer Kombiinstrument - warnt nicht
Maßnahme 1 (als Ursache gekennzeichnet)	
Kodiert	1__01500000
Maßnahmentyp	AP
Bezeichnung	GFS/Geführte Funktion
Maßnahme 2	
Kodiert	1__45604199
Maßnahmentyp	AP
Bezeichnung	LENKWINKELSENSOR NACH UPDATE
Maßnahme 3	
Kodiert	2_1015251098
Maßnahmentyp	M-DISS
Bezeichnung	10: Elektrische Anlage 15: Bereich Kombiinstrument - Anzeigen 25: Warnsummer 10: erhö.Verschleiß /erhö.Verbrauch /Beschädigung 98: In DISS nicht codierbar
Maßnahme 4	
Kodiert	3____9025
Maßnahmentyp	KDNR
Bezeichnung	Kombiinstrument

Tabelle B.4.: Beispiel eines inkongruenten Datensatzes eines realen Reparaturfalls vom 22.01.2007

dings vergleichsweise selten. Hier wurden bereits die Fälle mit einbezogen, in denen zusätzlich nicht relevante Symptome vorhanden waren.

Abschließend wird mit dem in Tabelle B.6 dokumentierten Fall wieder ein unbrauchbarer bzw. inkongruenter Reparaturfall erörtert. Für diesen Fall existieren zunächst kein Kundenfreitext sowie Werkstattfreitext. Mit der Maßnahmenmenge

ist jedoch eindeutig, welche Reparatur durchgeführt wurde. Allerdings besitzt keines der drei Symptome eine erkennbare Relation zu den durchgeführten Maßnahmen. Dieser Fall ist daher *inkongruent* und für ein Lernsystem unbrauchbar. Derartige Fälle machten 32 % der nicht lernbaren Fälle aus (vgl. 5.2 - Sympt. - Maßn. widersprüchlich). Erschwerend kommt hinzu, dass, wenn der Fall für sich betrachtet wird, falsche Relationen gelernt werden könnten. Das Lernsystem muss hier in der Lage sein, in Zusammenhang mit weiteren ähnlichen Reparaturfällen zu erkennen, dass die hier vorhandenen Symptome keine Relation mit den Maßnahmen besitzen. Dies ist beispielsweise ein Fall, der deutlich macht, warum rein statistische Verfahren schlechtere Ergebnisse als neuronale Netze erzielen, was jedoch in Kapitel 6 ausführlicher diskutiert wird. Auch wenn der vorgestellte Fall ein Extrembeispiel darstellt, treten auch in vielen anderen Fällen als „*Begleiterscheinung*“ nicht relevante Symptome auf. Aus der manuellen Untersuchung lässt sich ableiten, dass dies vor allem für DTCs auf ca. 62 % der Fälle zutrifft. Eine so große Zahl an nicht relevanten Symptomen stellt eine große Herausforderung für ein Lernsystem dar.

Aus den Ergebnissen der manuellen Untersuchung lassen sich noch einige weitere Aspekte ableiten, die in den drei Beispielfällen nicht enthalten waren. Hier soll die sogenannte „*Ursachenkennzeichnung*“ betrachtet werden. Der Kundendienst ist verpflichtet, im Gewährleistungsfall das schadensverursachende Teil bzw. falls nicht vorhanden, die erfolgreiche Arbeitsposition zu kennzeichnen. Würde dies konsequent so praktiziert, könnte die Performanz eines Lernsystems stark verbessert werden. Allerdings stellte sich heraus, dass diese in 8 % der Fälle entweder falsch, oder eine nicht informative Maßnahme, wie z.B. die Geführte Fehlersuche, gekennzeichnet wurde. In diesen Fällen muss zwangsläufig eine nicht als Ursache gekennzeichnete Maßnahme betrachtet werden, um den Fall als *kongruent* und brauchbar einstufen zu können. Die Ursachenkennzeichnung wird in dieser Arbeit daher nicht berücksichtigt.

Reparaturfalldaten	
Reparaturfall-ID	100
Reparaturannahmedatum	39168
Modelljahr (MDJ)	2005
Motorkennbuchstabe (MKB)	BKG
Verkaufsmodellcode	1K10C1
Fahrgestellnummer (anonymisiert)	WVWZZZ1KXXXX154
Beanstandung, Kundenfreitext	Fz hat Aussetzer, Checklampe leuchtet
Werkstattfreitext	- - -
Symptom 1	
Kodiert	1_0116684__13
Symptomtyp	DTC
Steuergeräteadr. und -bezeichnung	01 - Motorelektronik
Fehlercode und -beschreibung	16684 - Verbrennungsaussetzer erkannt
Fehlertypcode 1 und -bezeichnung	1 - Oberer Grenzwert überschritten
Fehlertypbezeichnung 3	statisch
Symptom 2	
Kodiert	1_0116686__13
Symptomtyp	DTC
Steuergeräteadr. und -bezeichnung	01 - Motorelektronik
Fehlercode und -beschreibung	16686 - Zylinder 2 Verbrennungsaussetzer erkannt
Fehlertypcode 1 und -bezeichnung	1 - Oberer Grenzwert überschritten
Fehlertypbezeichnung 3	statisch
Symptom 3	
Kodiert	20101040298__
Symptomtyp	S-DISS
Bezeichnung	Antriebsaggregat - Funktion - Fahrverhalten (Motor) - Aussetzer/Stottern - In DISS nicht codierbar
Maßnahme 1	
Kodiert	1__01500000
Maßnahmentyp	AP (Arbeitsposition)
Bezeichnung	GFS/Geführte Funktion
Maßnahme 2	
Kodiert	1__28201914
Maßnahmentyp	AP (Arbeitsposition)
Bezeichnung	Zündspule aus- u. eingebaut
Maßnahme 3	
Kodiert	2_1008021011
Maßnahmentyp	M-DISS
Bezeichnung	Elektrische Anlage - Bereich Zündanlage und Ent-störung - Kerzenschachtzündspule - erhö.Verschleiß /erhö.Verbrauch /Beschädigung - Fehlfunktion mit Ereignisspeichereintrag
Maßnahme 4	
Kodiert	3____2820
Maßnahmentyp	KDNR
Bezeichnung	Zündspule
Maßnahme 5 (als Ursache gekennzeichnet)	
Kodiert	4_036905715E
Maßnahmentyp	ET (Ersatzteil)
Bezeichnung	Zündspule

Tabelle B.5.: Beispiel eines kongruenten Datensatzes eines realen Reparaturfalls vom 27.03.2007

Reparaturfalldaten	
Reparaturfall-ID	84
Reparaturannahmedatum	38761
Modelljahr (MDJ)	2005
Motorkennbuchstabe (MKB)	BKC
Verkaufsmodellcode	1K1031
Fahrgestellnummer (anonymisiert)	WVWZZZ1KXXXX126
Beanstandung, Kundenfreitext	- - -
Werkstattfreitext	- - -
Symptom 1	
Kodiert	1_09__979_101
Symptomtyp	DTC
Steuergeräteadr. und -bezeichnung	09 - Elektronische Zentralelektrik
Fehlercode und -beschreibung	979 - Lampe für Abblendlichtscheinwerfer rechts-M31
Fehlertypcode 1 und -bezeichnung	10 - Unterbrechung/Kurzschluss nach Masse
Fehlertypbezeichnung 3	sporadisch
Symptom 2	
Kodiert	1_09_1064_121
Symptomtyp	DTC
Steuergeräteadr. und -bezeichnung	09 - Elektronische Zentralelektrik
Fehlercode und -beschreibung	1064 - Klemme 58d - Dimmung Innenlicht
Fehlertypcode 1 und -bezeichnung	12 - Elektrischer Fehler im Stromkreis
Fehlertypbezeichnung 3	sporadisch
Symptom 3	
Kodiert	1_52_1035__01
Symptomtyp	DTC
Steuergeräteadr. und -bezeichnung	09 - Elektronische Zentralelektrik
Fehlercode und -beschreibung	1035 - Fensterheber Thermoschutz aktiv, Beifahrer
Fehlertypcode 1 und -bezeichnung	0 - keine Anzeige
Fehlertypbezeichnung 3	sporadisch
Maßnahme 1	
Kodiert	1__01500000
Maßnahmentyp	AP (Arbeitsposition)
Bezeichnung	GFS/Geführte Funktion
Maßnahme 2	
Kodiert	1__26450199
Maßnahmentyp	AP (Arbeitsposition)
Bezeichnung	Magnetventil geprüft
Maßnahme 3	
Kodiert	1__26451936
Maßnahmentyp	AP (Arbeitsposition)
Bezeichnung	Magnetventil f Abgasrückführung a+e
Maßnahme 4	
Kodiert	3____2645
Maßnahmentyp	KDNR
Bezeichnung	Magnetventil f Abgasrückführung
Maßnahme 5 (als Ursache gekennzeichnet)	
Kodiert	4__6Q0906625
Maßnahmentyp	ET (Ersatzteil)
Bezeichnung	VENT BLOCK

Tabelle B.6.: Beispiel eines inkongruenten Datensatzes eines realen Reparaturfalls vom 13.03.2006

C Zusatzinformationen zur Evaluierung

C.1. Diskussion und Begründung der Konzeptbewertung

Die Bewertungen der einzelnen Kriterien aus Abbildung C.1 werden in Kapitel 9.2 nur stichpunktartig begründet. Im Folgenden wird eine ausführliche Diskussion und Begründung der jeweiligen Bewertungen gegeben.

C.1.1. Anwendungsfälle und Einsatzbereich

Elektrische oder mechanische Fehler Ein lernendes System kann prinzipiell diejenigen Lösungsvorschläge erlernen, die hinsichtlich der Lerndaten verfügbar sind. Mit den im Rahmen dieser Arbeit verfügbaren Daten sind elektrische wie auch mechanische Elemente vorhanden, genauso wie Kundenwahrnehmungen als

Abbildung C.1.: Kriterien der Konzeptbewertung nach [MFK08]

Symptome (DISS-kodierte Symptome) verfügbar sind, die auch die Diagnose rein mechanischer Fehler möglich machen. Der Einsatzbereich von Neucus erstreckt sich damit wie auch der der GFS von rein elektrischen, über mechatronische, bis hin zu rein mechanischen Fehlern.

Kundendienstwerkstatt Das System wurde primär für den Einsatz im Kundendienst entwickelt. Mit den Ergebnissen der Analyse in Abschnitt 3.2.1 stellt Neucus hier einen großen Nutzen dar, insbesondere dann, wenn Neucus mit den Fehlersuchprogrammen der GFS kombiniert wird.

Fahrzeugproduktion In der Produktion ist der Einsatz des Systems insbesondere in der nachgelagerten Fehlersuche von Produktionsfehlern sinnvoll. Die zu erwartende hohe Qualität der entstehenden Falldaten wie auch die Fähigkeit, schnell neue Falllösungen adaptieren zu können, machen den Einsatz von Neucus, auch hier in Kombination mit der GFS, profitabel.

Forschung und Entwicklung In der Forschung und Entwicklung, die üblicherweise mit Prototypen arbeiten, sind inhärent keine Falldaten vorhanden. Folglich kann ein lernendes System hier nicht eingesetzt werden.

Die Ergebnisse zur Bewertung der Anwendungsfälle und des Einsatzbereiches sind in Tabelle C.1 zusammengefasst.

<i>Kriterium</i>	<i>Bemerkung</i>	<i>Bewertung</i>
elektrische oder mechanische Fehler	Es lassen sich elektrische, mechatronische und mechanische Fehler diagnostizieren	⊙
Kundendienstwerkstatt	Deutliche Vorteile in der Kombination mit der GFS durch kollektive Wissensakquisition usw.	⊕⊕
Fahrzeugproduktion	Zur Optimierung der Fehlersuche am Ende der Produktionslinie	⊕
Entwicklung, Forschung	Scheidet als Einsatzbereich aufgrund mangelnder Lerndaten prinzipbedingt aus	⊖⊖

Tabelle C.1.: Bewertung der Anwendungsfälle und Einsatzbereiche von Neucus gegenüber der GFS

C.1.2. Technische Machbarkeit

Zusammenspiel mit IT-Systemen Für Neucus sind die Infrastruktur wie auch die IT-Systeme vollständig vorhanden. Die Diagnostik-Modelle können im Rahmen der regelmäßigen Updates auf die Tester verbreitet werden. Infrastruktur, Datenerfassung und Datenauswertung (für statistische Zwecke) sind schon lange vorhanden. Einzig ein Serversystem zum Erzeugen und Training der Diagnostik-Modelle muss aufgebaut werden.

Zusammenspiel mit Fahrzeugen Neucus wurde auf die Struktur heutiger Fahrzeuge ausgerichtet und setzt auf die erzeugten Ereignisspeichereinträge der Onboard-Diagnostik auf. In Zusammenhang mit Kundenwahrnehmungen sind auch ältere Fahrzeuge diagnostizierbar. Hinsichtlich zukünftiger Fahrzeugarchitekturen ist das Verfahren anpassungsfähig, sofern nach wie vor eine Onboard-Diagnostik existiert, wovon ausgegangen werden kann. Hier besitzt das Verfahren Vorteile gegenüber der GFS.

<i>Kriterium</i>	<i>Bemerkung</i>	<i>Bewertung</i>
Zusammenspiel mit IT-Systemen	Infrastruktur zur Datenerfassung, Verbreitung der Modelle und zur Datenanalyse vollständig vorhanden.	⊕⊕
Zusammenspiel mit Fahrzeugen	Für alle Fahrzeuge geeignet. Anpassungsfähig gegenüber zukünftigen Fahrzeugarchitekturen.	⊕

Tabelle C.2.: Bewertung der technischen Machbarkeit von Neucus gegenüber der GFS

C.1.3. Kosten

Investitionskosten Die Investitionskosten können aufgeteilt werden in die Investitionskosten für Werkstattssysteme und die für den OEM. Der heute eingesetzte Diagnostiktester VAS50XX kann direkt für den Einsatz von Neucus verwendet werden. Die notwendige Infrastruktur ist vollständig vorhanden. Für den OEM entstehen zunächst Entwicklungskosten durch die Überführung des Prototyps in ein Produktivsystem und durch die Integration in die vorhandene Diagnostiktester-Software. Weiterhin entstehen (geringe) Kosten für die Erweiterung der Serversysteme. Die Kosten für den Aufbau der Wissensbasis lassen sich ebenfalls als gering einschätzen: Die notwendigen Falldaten werden bereits heute aufgezeichnet werden bzw. sind vorhanden. Insgesamt fallen die Investitionskosten daher verhältnismäßig gering aus, so dass dieser Punkt neutral bewertet wird.

Laufende Kosten Die laufenden Kosten ergeben sich im Wesentlichen aus dem Erhalt und der Pflege der Wissensbasis. Auch hier besitzt ein lernendes System wie Neucus große Vorteile gegenüber der GFS, die für die Pflege der Fehlersuchprogramme derzeit ein 15-köpfiges Team von Autoren(Fachexperten) benötigt.

Kosten für neue Fahrzeugprojekte Auch für neue Fahrzeugprojekte fallen die Kosten gegenüber der GFS äußerst gering aus. Gerade hier muss für neue Fehlersuchprogramme der GFS viel investiert werden. Für Neucus können hier teilweise Fallbasen ähnlicher Fahrzeugprojekte wiederverwendet werden, indem Falldaten sukzessive durch neue „ersetzt“ werden.

Kriterium	Bemerkung	Bewertung
Investitionskosten	Keine signifikanten Investitionen notwendig	⊙
Laufende Kosten	Keine manuelle Pflege der Wissensbasis	⊕⊕
Kosten für neue Fahrzeugprojekte	Kein großer manueller Aufwand notwendig	⊕⊕

Tabelle C.3.: Bewertung der Kosten von Neucus gegenüber der GFS

C.1.4. Leistungsfähigkeit

Eines der wichtigsten Kriterien der Konzeptbewertung ist die qualitative Abschätzung der Leistungsfähigkeit des Diagnostiksystems. Dies umfasst vor allem die zu erwartenden Fehltauschkosten, die sich aus der Trefferquote und der Differenzierbarkeit möglicher Ursachen ergeben.

Trefferquote Neucus Die Trefferquote gibt an, in wievielen Fällen eine „richtige Lösung“ vom Diagnostiksystem vorgeschlagen wurde. Sie ist damit ein zentrales Kriterium, das direkte Auswirkungen auf die Kosten (Fehlersuchzeit, Fehltauschkosten), die Kundenzufriedenheit und die Akzeptanz hat. Die theoretische Leistungsfähigkeit eines lernenden Systems ist in diesem Punkt sehr hoch. Prinzipiell kann jede Lösung erlernt werden, die in den Daten explizit oder implizit vorhanden ist. In der Praxis spielt hier jedoch noch die Größe und Qualität der Fallbasis eine entscheidende Rolle. Bedingt durch die enorme Anzahl an Service-Werkstätten und der damit hohen Fallzahlen und bedingt dadurch, dass ein lernendes System den mittleren quadratischen Fehler der Lösungsvorschläge minimiert, kann eine durchschnittlich hohe Trefferquote erwartet werden, wie auch durch die Untersuchungen mit realen Falldaten gezeigt werden konnte.

Trefferquote GFS Auch für die Fehlersuchprogramme der GFS ist theoretisch jeder Fehler diagnostizier- und lösbar. Das muss jedoch insofern relativiert werden, wie hierfür ein theoretisch unendlicher Aufwand notwendig wäre. Es können nur Fehler diagnostiziert werden, die zum einen von einem Autor erkannt und zum anderen (korrekt) in einem Fehlersuchprogramm integriert wurden. In der Praxis müssen hier Kompromisse bezüglich des zu investierenden Aufwands eingegangen werden. Weiterhin ungünstig ist das mangelnde Feedback über die tatsächliche Leistungsfähigkeit der Programme. Diese ist nur für einen Bruchteil der Programme bekannt (eine ausführliche Analyse wurde bereits in Abschnitt 3.2.1 gegeben).

Differenzierbarkeit Die Differenzierbarkeit ist ein Maß dafür, wie Feingranular die Diagnose gestellt wird. So macht es einen Unterschied, ob die Diagnose lautet „Fehler im Bereich Nockenwellensensor“ oder „Leitung 1 Nockenwellensensor $\Rightarrow$ Motorsteuergerät“. Die Granularität bzw. Differenzierbarkeit von Neucus hängt prinzipiell von der Granularität der Lösungen in der Fallbasis ab. Wären diese sehr feingranular, wären auch die Lösungsvorschläge von Neucus sehr feingranular. Für

die Fehlersuchprogramme der GFS gilt analog zur Trefferquote, dass auch hier die Differenzierbarkeit theoretisch sehr hoch sein kann. In der Praxis erscheint jedoch gerade für elektrische Fehler häufig ein „*Suche nach Stromlaufplan*“. In diesem Punkt sind die Ergebnisse aus Neucus feingranularer, für andere Fehler ist die GFS differenzierter. Insgesamt hat die GFS jedoch prinzipbedingte Vorteile. Auch hier kann eine Kombination aus beiden Verfahren den maximalen Vorteil bringen.

Fehlersuchzeit Die Fehlersuchzeit setzt sich zusammen aus der Anwendungsgeschwindigkeit, der Zeit die für das Stellen einer Diagnose benötigt wird und der Zeit die für die Durchführung der Fehlersuchaktionen benötigt wird. Die Anwendungsgeschwindigkeit von Neucus ist sehr hoch. Das liegt vor allem in der hohen Anwendungsgeschwindigkeit von neuronalen Netzen. Jedoch sind auch regelbasierte Systeme wie die GFS sehr schnell in der Anwendung. Hinsichtlich der Zeit für die Durchführung der Fehlersuchaktionen hat Neucus Vorteile. Dies liegt vor allem an der Funktionsweise des Lernalgorithmus, da hier der mittlere quadratische Fehler über alle Falldaten minimiert wird, was zwangsläufig zu einer Minimierung der falsch verdächtigten Komponenten und damit zu einer Minimierung der durchzuführenden Fehlersuchaktionen führt.

Erweiterbarkeit, Skalierbarkeit Neucus baut auf der plattformunabhängigen Sprache Java sowie Open-Source-Software auf. Damit sind Erweiterungen hinsichtlich des Funktionsumfangs wie auch der Lernaufgabe sehr einfach möglich. Bezüglich der GFS stellen sich Erweiterungen wesentlich aufwändiger dar. Hier sind nicht nur die Frage der Software-Rechte zu klären, sondern auch die Einschränkungen durch die existierende manuelle Fallbasis zu beachten. Die bisher existierenden Fehlersuchprogramme besitzen keine einheitlichen Schnittstellen und sind nur sehr eingeschränkt maschinell auswertbar. Hier besitzt Neucus große Vorteile.

Variantenvielfalt Eine bereits ausführlich diskutierte Schwachstelle der GFS ist die Beherrschung der Variantenvielfalt (vgl. Abschnitt 3.2.1). Der Erstellungsaufwand für neue Fahrzeuge und Varianten steigt linear bis exponentiell an. Die hier vorgestellten Verfahren gehen dynamisch mit verschiedenen Varianten um. Sofern (hinsichtlich der Falldaten) eine Unterscheidung zwischen Varianten notwendig ist, wird diese in der Struktur der Netze manifestiert. Für neue Fahrzeugprojekte bedeutet das, dass zunächst keine Varianten, jedoch mit zunehmend differenzierenden Fällen immer mehr Varianten betrachtet werden.

C.1.5. Anwendbarkeit und Nutzerakzeptanz

Internationalisierung Die Internationalisierung von Software kann unter Umständen mit großem Aufwand verbunden sein, insbesondere dann, wenn eine große Wissensbasis (wie im Fall der GFS) in viele Sprachen übersetzt werden muss. Neucus erlernt als Lösungselemente Ersatzteilnummern, Arbeitspositionen und DISS-kodierte Werkstattwahrnehmungen. Als Symptome werden Ereignisspeichereinträge und DISS-kodierte Kundenwahrnehmungen betrachtet. Für jedes dieser Elemente

Kriterium	Bemerkung	Bewertung
Trefferquote	Beide Systeme können hohe Trefferquoten erzielen, für die GFS jedoch mit hohem Aufwand	⊕
Differenzierbarkeit	Beide Systeme können theoretisch eine hohe Differenzierbarkeit erreichen, in der Praxis ist diese bei beiden geringer	⊖
Fehlersuchzeit	Durch die Minimierung des mittleren quadratischen Fehlers hat Neucus Vorteile	⊕
Erweiterbarkeit, Skalierbarkeit	Neucus ist plattformunabhängig und baut auf Open-Source-Software auf. Erweiterungen der GFS sehr aufwändig.	⊕⊕
Variantenvielfalt beherrschbar	Für die GFS sehr aufwändig. Neucus lernt ohne Mehraufwand Varianten zu unterscheiden.	⊕⊕

Tabelle C.4.: Bewertung der Leistungsfähigkeit gegenüber der GFS

sind, bedingt durch die GFS, bereits Übersetzungen vorhanden. Hier besteht ein Vorteil gegenüber der GFS insofern, wie Anweisungen in neuen Fehlersuchprogrammen individuell übersetzt werden müssen.

Flexibilität der Benutzerführung Das Ergebnis von Neucus ist eine gewichtete Liste von möglichen Ursachen. Es bleibt folglich dem Mechaniker überlassen, welcher Ursache er als erstes nachgeht. Eine Quantifizierung der Ursache durch einen Vertrauenswert unterstützt ihn dabei im Entscheidungsprozess. Damit wird dem KD-Mitarbeiter die Möglichkeit gegeben, die eigene Erfahrung oder aber Rücksprachen mit anderen Werkstätten bzw. dem Technical Servicecenter von VW mit in den Entscheidungsprozess einzubringen. Somit wird eine hohe Flexibilität erreicht. Die GFS bietet dagegen lediglich eine ungewichtete Liste an Fehlersuchprogrammen an, die an sich sehr starr sind. Auch in einer Kombination von Neucus mit der GFS würde ein Gewinn in der Flexibilität und Benutzerführung erreicht werden.

Anzeige von Zusatzinformationen Hinsichtlich der Integration von Zusatzinformationen stellt sich Neucus sehr flexibel dar. Informationen zur Entscheidungsfindung können z. B. die Anzahl an Referenzfällen (Häufigkeit), Informationen zu den einzelnen Symptomen, ähnliche Fälle usw. sein. Informationen zu den Ursachen können alle mit der verdächtigten Komponente zusammenhängenden Dokumente wie Stromlaufpläne, Aufrisszeichnungen, Arbeitswerte, etc. sein. In der GFS sind diese Informationen in den jeweiligen Fehlersuchprogrammen integriert, so dass dieser Punkt als neutral bewertet wird.

Nachvollziehbarkeit Die Nachvollziehbarkeit der Lösung ist für lernende Systeme prinzipbedingt schwer darzustellen. Während strukturbasierte Verfahren anhand

logischer Schlussfolgerungen über die Struktur die Lösung erklären könnten¹, bauen lernende Verfahren ihre Lösungen auf der Minimierung einer Fehlerfunktion über die Menge der Falldaten auf. Trotzdem kann eine Nachvollziehbarkeit intuitiv über Informationen zur Fallhäufigkeit und Fehlerwahrscheinlichkeit gelingen. Die Argumentation könnte beispielsweise die Form „*Mitarbeiter, die ein ähnliches Problem hatten, konnten folgende Ursachen feststellen...*“ annehmen. Hinsichtlich der GFS hat diese Argumentation insofern Vorteile, dass in Fehlersuchprogrammen zwar theoretisch eine sehr detaillierte Begründung möglich ist, diese aber aufgrund des damit verbundenen manuellen Aufwands oft vernachlässigt wird.

<i>Kriterium</i>	<i>Bemerkung</i>	<i>Bewertung</i>
Internationalisierung	Übersetzungen für Rep.-Lösungen und Symptomkataloge vorhanden. GFS-Programme müssen übersetzt werden.	⊕
Flexibilität der Benutzerverführung	Maximale Flexibilität durch gewichtete Liste an mögl. Ursachen. GFS sehr starr.	⊕⊕
Anzeige von Zusatzinformationen	Beide Systeme können vielfältige Informationen integrieren/verknüpfen.	⊙
Nachvollziehbarkeit	Nachvollziehbarkeit der GFS theoretisch gegeben, praktisch aber oft vernachlässigt. In Neucus mögl. durch zus. stat. Informationen.	⊙

Tabelle C.5.: Bewertung der Anwendbarkeit und Nutzerakzeptanz gegenüber der GFS

¹ Wie Schlussfolgerungen in hochkomplexen Systemen für Benutzer einfach und intuitiv dargestellt werden können, bleibt zwar fraglich, ist jedoch nicht ausgeschlossen.

C.2. Weitere Evaluierungsergebnisse der neuronalen Diagnostik-Modelle

C.2.1. Bestimmung der Trainingsparameter

Mit der Vorstellung des Lernalgorithmus (Abschnitt 7.1.4) haben die nachfolgenden vier Trainingsparameter den größten Einfluss:

Anzahl der Trainingsepochen n Die Anzahl n der Trainingsepochen gibt an, wie oft die Trainingsmenge durchlaufen wird. Die notwendige Anzahl ist stark abhängig von der Fallzahl und der Lernrate η und darf weder zu niedrig noch zu hoch sein (Generalisierungsverlust).

Initialwerte der Gewichte Die Initialwerte der Gewichte dürfen keine Symmetrien aufweisen, so dass eine Initialisierung mit Zufallswerten in einem festgelegten Intervall erfolgt. Die Intervallgröße I beeinflusst die Konvergenzgeschwindigkeit.

Lernrate η Die Lernrate bestimmt die Größe der Gewichtsveränderung. Sinnvolle Werte liegen für die hier vorgestellten Modelle zwischen 0,2 und 0,5.

Maximaler tolerierter Fehler d_{max} Für Fehler unterhalb von d_{max} werden die Gewichte nicht mehr verändert. Dadurch kann z. B. ein *Flat-Spot* aufgrund von zu geringen Gewichtswerten verhindert werden. Geeignete Werte liegen im Intervall $[0 \dots 0,1]$.

Es ist kein Verfahren zur analytischen Bestimmung der Parameter bekannt, so dass eine empirische Ermittlung notwendig ist. Hierzu wurde eine Parameteriteration mit insgesamt 96 Parametersets und einer aus Performanzgründen auf ca. 2.000 Fälle reduzierten Trainingsmenge durchgeführt. Mögliche Werte für die einzelnen Parameter wurden in vorhergehenden Untersuchungen ermittelt. In Abbildung C.2 ist das Ergebnis für die Maßnahmenklasse ET dargestellt. Die Abbildung zeigt, wie sehr die Evaluierungsquote von den Parametern abhängt, selbst wenn diese in sinnvollen Bereichen liegen. Das Optimum konnte hier für das Parameterset mit 10 Trainingsepochen, einer Lernrate von $\eta = 0,2$, einem Initialisierungsintervall von $[-0,01 \dots 0,01]$ und einem maximal tolerierten Fehlen von $d_{max} = 0$ erreicht werden. Auffällig ist, dass mit einer höheren Lernrate zwar schneller ein (lokales) Optimum erreicht werden kann, das jedoch unterhalb des globalen Optimums liegt (vgl. P.-Set $n3 - d0 - \eta 0,5 - I 0,01$). Der Grund hierfür liegt darin, dass das globale Optimum, dass sich in einer bestimmten Konfiguration der Gewichtsmatrix W manifestiert, übersprungen wird und die folgende Trainingsepoche bereits ein Overfitting erzeugt. Der Effekt von Overfitting ist für alle Parametersets deutlich zu beobachten. Zu viele Trainingsepochen können einen Verlust von bis zu 7% bedeuten. Bezüglich des Initialisierungsintervalls I und des maximal tolerierten Fehlers d_{max} zeigt sich, dass ein schnelles Erreichen von niedrigen Gewichtswerten von Vorteil ist. Mit einem Wert von $d_{max} > 0$ wird gleichzeitig auch der minimal mögliche Wert eines Gewichts $w_{ij} \in W$ begrenzt, und mit einem niedrigen Initialisierungsintervall werden schneller niedrige Gewichtswerte erreicht. Der Vorteil niedriger Gewichtswerte liegt vor allem darin begründet, dass für eine konkrete Reparaturmaßnahme K_i nur

Abbildung C.2.: Abhängigkeit der Evaluierungsquote q_n von den Trainingsparametern Trainingsepochen, maximal tolerierter Fehler $d_{max}(d)$, Lernrate η und Initialisierungsintervall (I)

wenige Symptome S_R relevant sind, also ein Verbindungsgewicht mit $w_{ij} > 0$ besitzen sollten.

C.2.2. Varianz der Ergebnisse

Die Varianz folgt zum einen aus der zufälligen Initialisierung der Verbindungsgewichte und zum anderen aus der zufallsbasierten Trainingsdatensatzauswahl (*Shuffle*-Funktion). Zur Bestimmung der Grenzen wurde für jede Maßnahmenklasse je ein Trainingsvorgang 10 mal wiederholt. Die Auswertung der Ergebnisse in Abbildung C.3 zeigt, dass die Varianz jedoch allgemein niedrig im Bereich von +1,7% bis -1,3% ausfällt. Die in Abbildung 9.4 feststellbaren Varianzen bestätigen das.

C.2.3. Abhängigkeit der Symptom- und Maßnahmenmenge von der Fallzahl

Hierbei ist zuerst von Interesse, wie sich die Größe der Symptommenge $\#S_R$ und der Maßnahmenmenge $\#K_R$ für steigende Fallzahlen $\#R_{FB}$ entwickelt. Die Ergebnisse der in Abbildung C.4 dargestellten Untersuchung zeigen, dass die Anzahl der in der Datenbasis enthaltenen Symptome und Maßnahmen mit zunehmend betrachteter Fallzahl zunächst stark ansteigt, dann jedoch zunehmend abflacht. Allerdings ist selbst mit 72.000 Falldaten (gefiltert 56.000) eine Konvergenz noch nicht erreicht. Das kann mit der stark einseitigen Häufigkeitsverteilung (vgl. Abbildung 5.3 und 5.3) erklärt werden.

Abbildung C.3.: Streuung der Evaluierungsquote q_n für verschiedene Maßnahmenklassen

Für eine grobe Abschätzung des Konvergenzverhaltens wird die Ankunft eines Fahrzeugs mit einem bestimmten Symptom s_i oder einer bestimmten Maßnahme k_i

Abbildung C.4.: Abhängigkeit der Symptommenge $\#S_R$ und der Maßnahmenmenge $\#K_R$ für steigende Fallzahlen $\#R_{FB}$ (alle Maßnahmenklassen). **Filter:** Symptom mind. 3 mal, Maßnahme mind. 5 mal in der betrachteten Menge vorhanden

als ein Bernoulli-Prozess² betrachtet. Die Wahrscheinlichkeitsverteilung von $P(s_i)$ kann mit den großen Fallzahlen $\#R_{FB}$ und der geringen Auftrittswahrscheinlichkeit von s_i dann mit der Poisson-Verteilung³ angenähert werden:

$$P(\#s_i = k|\lambda) = \frac{\lambda^k}{k!} e^{-\lambda} \quad (\text{C.1})$$

Sie bestimmt, mit welcher Wahrscheinlichkeit das Symptom s_i in einem festgelegten Zeitintervall genau k -mal auftritt. Der Parameter λ ist dabei Erwartungswert μ und Varianz δ^2 zugleich und repräsentiert die mittlere Auftrittshäufigkeit pro betrachtetes Intervall. Wird als Intervall die bekannte Falldatenmenge $\#R_{FB}$ angenommen, kann der Parameter λ für ein nach Abbildung 5.3 seltenes Symptom mit $\lambda = \frac{1 \cdot \text{Auftreten}}{\text{Intervall}} = 1$ bestimmt werden. Die Häufigkeitsverteilung ergibt sich damit nach Abbildung C.5 (linke Seite). Daraus geht auch hervor, dass 37 % der Symptome eine solche geringe Auftrittswahrscheinlichkeit besitzen und nicht in der Fallbasis vorhanden sind. Wie viele Falldaten notwendig sind, damit ein solches Symptom zumindest zu beispielsweise 90 % vorkommt, kann bestimmt werden, indem die Wahrscheinlichkeit, dass das Symptom s_i nicht in den Falldaten vorkommt, also $P(\#s_i = 0|\lambda)$, invertiert wird und die Abhängigkeit vom Parameter λ betrachtet wird:

$$P(\#s_i > 0|\lambda) = 1 - e^{-\lambda} \quad (\text{C.2})$$

Die Abbildung C.5 (rechte Seite) zeigt die Wahrscheinlichkeit, dass ein Symptom mindestens einmal in der Fallbasis vorkommt, d. h. $P(\#s_i > 0|\lambda)$, in Abhängigkeit des Parameters λ bzw. der Fallzahl. Werden beispielsweise nur die 41 % der Symptome mit $\tilde{P}(s_i) = 0,018\%$ betrachtet, ist noch mit einer Steigerung der Symptommenge von $7100 \cdot 41\% \cdot (100\% - 63,2\%) = 1061$ Symptomen zu rechnen. Wird dieses Vorgehen konsequent auf die Verteilung nach Abbildung 5.3 angewendet, folgt daraus eine zu erwartende Symptommenge von 8.301 Symptomen und eine Maßnahmenmenge von 18.106 Maßnahmen.

² Ein Bernoulli-Prozess ist ein zeitlich diskreter stochastischer Prozess, der aus einer endlichen oder abzählbar-unendlichen Folge von unabhängigen Ereignissen besteht (gedächtnisloser Prozess).

³ Die Poisson-Verteilung ist eine Näherung an die Bernoulli-Verteilung für $n \rightarrow \infty, p_n \rightarrow 0$ und $\lim_{n \rightarrow \infty} np_n = \lambda > 0$. Sie wird umgangssprachlich auch als die Verteilung der seltenen Ereignisse bezeichnet.

Abbildung C.5.: Poisson-Verteilungen für ein Symptom s_i mit $\tilde{P}(s_i) = 0,018\%$.

Oben: Häufigkeitsverteilung für $\lambda = 1$. **Unten:** Wahrscheinlichkeit von $P(\#s_i > 0 | \lambda)$ in Abhängigkeit von λ bzw. der Fallzahl $\#R_{FB}$

C.3. Evaluierungsergebnisse exemplarischer Fälle

C.3.1. Fall 1: Tiguan – GRA (Geschwindigkeitsregelanlage)

<i>Basisdaten Fall Tiguan 44–446</i>
Stg.–Adr. 16 = Lenkradelektronik
Fehlercode 895 = GRA/ADR Bedienteil-E45
1794 Fälle in der Datenbasis vorhanden
Fehlerarten:
12 % statisch
88 % sporadisch

Tabelle C.6.: Basisdaten des exemplarischen Falls Tiguan 16-895

Abbildung C.6.: Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 16-895

C.3.2. Fall 2: Tiguan – Funktionseinschränkung durch Unterspannung

<i>Basisdaten Fall Tiguan 44–446</i>
Stg.–Adr. 44 = Lenkhilfe (EPS_ZFLS Kl.2)
Fehlercode 446 = Funktionseinschränkung durch Unterspannung
226 Fälle in der Datenbasis vorhanden
Ausschließlich Fehlerart „unterer Grenzwert unterschritten“, sporadisch

Tabelle C.7.: Basisdaten des exemplarischen Falls Tiguan 44-446

Abbildung C.7.: Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 44-446

C.3.3. Fall 3: Tiguan – KI15 Innenraum

<i>Basisdaten Fall Tiguan 09-2403</i>
Stg.–Adr. 09 = Elektronische Zentralelektrik
Fehlercode 2403 = KI.15 Innenraum
85 Fälle in der Datenbasis vorhanden
Fehlerarten:
91 % Adaptionsgrenze (mul.) überschritten, (79 % sporadisch)
9 % Signal zu groß, (12,5 % sporadisch)

Tabelle C.8.: Basisdaten des exemplarischen Falls Tiguan 09-2403

Abbildung C.8.: Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 09-2403

C.3.4. Fall 4: Golf – Ladedruckregelung, Regelgrenze unterschritten

<i>Basisdaten Fall Golf 01–16683</i>
Stg.–Adr. 01 = Motorsteuergerät
Fehlercode 16683 = Ladedruckregelung, Regelgrenze unterschritten
395 Fälle in der Datenbasis vorhanden
Alle Fehlerart „keine Anzeige“:
12 % Keine Anzeige
5 % Sporadisch
38 % Pending
45 % Statisch

Tabelle C.9.: Basisdaten des exemplarischen Falls Golf 01–16683

Abbildung C.9.: Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 01–16683

D Tabellen

D.1. Beispielhafte Vertrauenswerte für XOR-, AND- und OR-ähnliche Daten

Anz. Fälle ^a	s_1	s_2	k_1	Zähler	Nenner	Vertrauenswert $v_{k_1, norm}$
10.000	0	0	0	0,0499%	99,6013%	0,05 %
10	1	0	1	0,0499%	0,0996%	33,4 %
10	0	1	1	0,0499%	0,0996%	33,4 %
10	1	1	0	0,0499%	0,0001%	99,8 %

^a Die unterschiedlichen Fallzahlen sind notwendig, um realistische Verhältnisse zu vermitteln.

Tabelle D.1.: Vertrauenswerte des inferentiellen Diagnose-Modells nach Definition 6.1 für XOR-ähnliche Daten

Anz. Fälle ^a	s_1	s_2	k_1	Zähler	Nenner	Vertrauenswert $v_{k_1, norm}$
10.000	0	0	0	0,0332%	99,734%	0,03 %
10	1	0	1	0,0665%	0,066%	100,0 %
10	0	1	1	0,0665%	0,066%	100,0 %
10	1	1	1	0,1329%	0,133%	100,0 %

^a Die unterschiedlichen Fallzahlen sind notwendig, um realistische Verhältnisse zu vermitteln.

Tabelle D.2.: Vertrauenswerte des inferentiellen Diagnostik-Modells nach Definition 6.1 für AND-ähnliche Daten

Anz. Fälle ^a	s_1	s_2	k_1	Zähler	Nenner	Vertrauenswert $v_{k_1, norm}$
10.000	0	0	0	0,000%	99,7%	0,0 %
10	1	0	0	0,000%	0,1%	0,0 %
10	0	1	0	0,000%	0,1%	0,0 %
10	1	1	0	0,099%	0,1%	99,9 %

^a Die unterschiedlichen Fallzahlen sind notwendig, um realistische Verhältnisse zu vermitteln.

Tabelle D.3.: Vertrauenswerte des inferentiellen Diagnostik-Modells nach Definition 6.1 für OR-ähnliche Daten

D.2. Diagnostikservicefunktionen

<i>Diagnostikservicefunktion</i>
Steuergeräteversion abfragen
Ereignisspeicher abfragen
Stellglieddiagnostik
Grundeinstellungen abfragen und einstellen
Ereignisspeicher löschen
Ausgabe beenden
Steuergerät codieren
Messwerteblock lesen
Anpassung z.B. (Service-Intervall Rückstellung)
Login-Prozedur
Automatischer Prüfablauf

Tabelle D.4.: Verschiedene Diagnostikservicefunktionen der Steuergeräte

D.3. Filterkriterien

Auswahlkriterien

- Nur Fälle im Jahr 2007
- Nur Fahrzeuge mit der Fahrgestellnummer 'WWW____1K
- Nur Reparaturfälle aus Deutschland
- Nur Reparaturfälle mit von Hand eingegebenen DISS-Freitext

Filterkriterien

- Es werden nur Reparaturfälle mit genau EINEM Schadensantrag betrachtet
Es werden nur Reparaturfälle mit einem Schadensantrag im Zeitraum von ± 10 Tagen betrachtet
- Es werden nur die Ereignisspeichereinträge des 1. übermittelten Protokolls betrachtet

Zusammenführung

- Gleiche Fahrgestellnummer in den Datensätzen
- Gleiche Partnernummer
in den Datensätzen
- Protokoll im Zeitraum von ± 10 Tagen des Reparaturannahmedatums des Schadensantrags

D.4. Kodierungstabellen für Symptome, Reparaturmaßnahmen und Fahrzeugkontext

D.4.1. Symptome und Fahrzeugkontext

Kodierung: Sxxxxxxxxxxxx

S - Konstanter Prefix (JavaNNS erlaubt keine Ziffer am Anfang eines Namens)

a [i] - Art des Sympt. (1-DTC, 2-DISS)

x - Kodierung abhängig von a, Anzahl Stellen variabel

Wenn a=1 (DTC)

Kodierung: Sa_snnnnnnffm

s [i] - Addr des Steuergerätes (HEX), 2 Stellen

n [i] - DTC-Nummer (Dezimal), 5 Stellen, vorangehende 0en mit '_' aufgefüllt

f [i] - Fehlerart1 (999 nicht vorhanden), 3 Stellen

m [i] - Fehlerart2 (9 nicht vorhanden), 1 Stelle

Wenn a=2 (DISS)

Kodierung: Sddddddddd

d - linksbündige Zahl (Diss-Kodierung 10 Stellen)

ddddddddd => 1122334455

11 - Struktur-Gruppe 1

22 - Struktur-Gruppe 2

33 - Struktur-Gruppe 3

44 - Fehlerbeschreibung

55 - Fehlerbeschreibung

44&55 sind wie Fehlerart 1 und Fehlerart 2 bei DTC zu sehen, wenn es um die visuelle Darstellung in Neucus geht

Wenn a=3 (Kontext)

Kodierung Sa_STEUERGERAET_SWTEIL_SWVERSION

STEUERGERAET - [0..9a..zA..Z]* (Steuergeraeteadresse)

SWTEIL - [0..9a..zA..Z]* (Softwareteilenr.)

SWVERSION - [0..9a..zA..Z]* (Softwareversionsnr.)

Wenn a=9 (Ungültiges Symptom)

Kodierung: Sa

D.4.2. Reparaturmaßnahmen

Kodierung: Mxxxxxxxxxx

M - Konstanter Prefix (JavaNNS erlaubt keine Ziffer am Anfang eines Namens)

a - Art der Maßn.

x - Kodierung abhängig von a, 11 Stellen.

Wenn a=1 (Arbeitsposition-Nummer)

Kodierung: M1_____d

d - Arbeitsposition-Nummer (8 Stellen)

Wenn a=2 (DISS)

Kodierung: M2_ddd d d d d d d d d d d

d - rechtsbündige Zahl (Diss-Kodierung 10 Stellen)

Wenn a=3 (Kundendienstnummer)

Kodierung: M3_____ d d d d

d - rechtsbündige Zahl (KdNr-Kodierung 4 Stellen)

Wenn a=4 (Ersatzteilnummer)

Aufbau der Kodierung siehe Dokument Teilenummernsystem

Kodierung: M4 e e e e e e e e e e e e e e e e

- Ersatzteilnummer rechtsbündig, max. 11 Stellen

- Leerzeichen werden mit ' _ ' ersetzt.

Wenn a=5 (Ersatzteil-Rumpfnummer)

(entspr. Feld 'SCHAD__RUMPF__NR' in EHDWANF__ANTRAEGE
__SCHAEDEN)

Kodierung: M5_____ d d d d d d

d - Ersatzteil-Rumpfnummer (6 Stellen)

Wenn a=6 (Ersatzteil-FL-Nummer)

Kodierung: M6_____ d d d d d d

d - FL-Nummer (max. 8 Stellen)

Wenn a=7 (Ersatzteil-FM-Nummer)

Kodierung: M7 e e e e e e e e e e e e [ee] (evtl. Überlänge))

e - FM-Mummer (max. 14 Stellen)

Wenn a=9 (Ungültige Massnahme)

Kodierung: Ma_____

E Patente

Im Rahmen der Kooperation mit Volkswagen sind folgende Patente entstanden:

Titel	K.-Nr.	Aktenzeichen
Diagnosesystem und Diagnoseverfahren für ein elektrische Komponenten umfassendes elektrisches System, insbesondere ein Kraftfahrzeugsystem	K 14756 DE	10 2007 006 715.3
Verfahren und Vorrichtung zur Unterstützung einer Diagnose eines elektrischen Systems mittels wahrscheinlichkeitsbasierter Fehlerkandidatenermittlung	K 14841 DE	10 2007 010 978.6
Diagnosesystem und Verfahren zum Erstellen eines Prüfablaufs für eine Diagnose eines mechatronischen Gesamtsystems	K 14856 DE	10 2007 018 732.9
Verfahren zur Herstellung eines Diagnosesystems, insbesondere für ein Kraftfahrzeug	K 14999 DE	10 2007 045 255.3

F Ausgewählte Publikationen

Im Rahmen der Tätigkeit am Institut für Regelungstechnik der Technischen Universität Braunschweig sind folgende Publikationen entstanden:

MÜLLER, Tobias Carsten ; KRIEGER, Olaf ; FORM, Thomas: Optimierung der Geführten Fehlersuche durch intelligente adaptive Modelle. In: *Diagnose in mechatronischen Fahrzeugsystemen*. Dresden : Expert Verlag, Mai 2011. – ISBN 978-3-8169-2821-8, S. 243–255

MÜLLER, Tobias C.: Komplexitätsreduzierte neuronale Netze zur Offboard-Diagnostik in Fahrzeugsystemen. In: HEINERT, Michael; RIEDEL, Björn (Hrsg.): *Theorie und Anwendung lernender Algorithmen für Ingenieure und Naturwissenschaftler*, Beyrich GmbH & Co. KG, 2010. – ISBN 987-3-926146-20-5, S. 71–81

KRIEGER, Olaf ; LANGE, Klaus ; FORM, Thomas ; MÜLLER, Tobias C.: Wahrscheinlichkeitsbasierte Fahrzeugdiagnose mit automatischer Generierung von Prüfanweisungen. In: *14. Internationale Konferenz Elektronik im Kraftfahrzeug, VDI-Berichte 2075*. Baden Baden : VDI Verlag GmbH, 2009. – ISBN 978-3-18-092075-7, S. 333–344

MÜLLER, Tobias C. ; LOCHAU, Malte ; DETERING, Stefan ; SAUST, Falko ; GARBERS, Henning ; MÄRTIN, Lukas ; FORM, Thomas ; GOLTZ, Ursula: Umsetzung eines modellbasierten durchgängigen Entwicklungsprozesses für AUTOSAR-Systeme mit integrierter Qualitätssicherung / Technische Universität Braunschweig. 2009-06. – Forschungsbericht

LOCHAU, Malte ; MÜLLER, Tobias C. ; STEINER, Jens ; GOLTZ, Ursula ; FORM, Thomas: Optimierung von AUTOSAR-Systemen durch automatisierte Architektur-Evaluation. In: *14. Internationale Konferenz Elektronik im Kraftfahrzeug, VDI-Berichte 2075*, VDI Verlag GmbH, 2009. – ISBN 978-3-18-092075-7, S. 827–838

LOCHAU, Malte ; MÜLLER, Tobias C. ; DETERING, Stefan ; GOLTZ, Ursula ; FORM, Thomas: Architektur-Evaluation von AUTOSAR-Systemen: Adaption und Integration. In: *Elektronik automotive congress*, Elektronik automotive, 2009, S. 77-98

MÜLLER, Tobias C. ; KRIEGER, Olaf ; BREUER, Andreas ; LANGE, Klaus ; FORM, Thomas: A Heuristic Approach for Offboard-Diagnostics in Advanced Automotive Systems. In: *SAE World Congress*, SAE International, Detroit, MI, USA, 2009. – ISSN 0148-7191. – SAE Dokument Number: 2009-01-1027

auch erschienen in: *SAE International Journal Passenger Cars – Electronic and Electrical Systems*

SAUST, Falko ; MÜLLER, Tobias C. ; WILLE, Jörn M. ; MAURER, Markus: Entwicklungsbegleitendes Simulations- und Testkonzept für autonome Fahrzeuge in städtischen Umgebungen. In: *AAET - Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel*, GZVB e.V., 2009. – ISBN 978–3–937655–19–2, S. 87-107

MÜLLER, Tobias C.: Entwicklungsunterstützung Diagnose 20xx / Technische Universität Braunschweig. Braunschweig, Dezember 2008. – Projektbericht

MÜLLER, Tobias C. ; KRIEGER, Olaf ; LANGE, Klaus ; BREUER, Andreas ; FORM, Thomas: Neuronale Netzmengen für die Fehlerdiagnose in komplexen Fahrzeugsystemen. In: *Diagnose in mechatronischen Fahrzeugsystemen* Bd. 1. München : Expert Verlag, Juli 2008. – ISBN 978–3–8169–2821–8, S. 168-179

KRIEGER, Olaf ; BREUER, Andreas ; MÜLLER, Tobias C.: Intelligente Fahrzeugdiagnose mit variabler Prüfstrategie. In: *Diagnose in mechatronischen Fahrzeugsystemen* Bd. 1. München : Expert Verlag, Juli 2008. – ISBN 978–3–8169–2821–8, S. 130-141

MÜLLER, Tobias C. ; FORM, Thomas ; KRIEGER, Olaf: Evaluierung von Offboard-Diagnosesystemen am Beispiel eines erfahrungsbasierten Diagnoseverfahrens. In: GÜHMANN, Clemens (Hrsg.): *Simulation und Test in der Funktions- und Softwareentwicklung für die Automobilelektronik* Bd. 2, Expert Verlag, 2008. – ISBN 978–3–8169–2818–8, S. 388-405

MÜLLER, Tobias C. ; LANGE, Klaus ; BREUER, Andreas ; KRIEGER, Olaf ; FORM, Thomas: Automatische erfahrungsbasierte Diagnose aus Felddaten mit neuronalen Netzen. In: *Elektronik im Kraftfahrzeug*. Düsseldorf : VDI Verlag GmbH, 2007. – ISBN 978–3–18–09200, 593-605. –

Ausgezeichnet mit dem „*Young Engineers Best Paper Award*“,
auch erschienen in: VDI Jahrbuch 2008 - Fahrzeug- und Verkehrstechnik

KRIEGER, Olaf ; BREUER, Andreas ; LANGE, Klaus ; MÜLLER, Tobias C. ; FORM, Thomas: Wahrscheinlichkeitsbasierte Fahrzeugdiagnose auf Basis individuell generierter Prüfabläufe. In: *Mechatronik 2007 - Innovative Produktentwicklung*. Wiesloch bei Heidelberg : VDI Verlag GmbH, Mai 2007. – ISBN 978–3–18–091971–3, S. 235-248

MÜLLER, Tobias C. ; YILMAZ, Hayattin ; BERROTH, Manfred: Entwicklung und Aufbau einer Elektrolarynx mit variabler Frequenz und integrierter Bluetooth - Funkschnittstelle für laryngektomierte und tracheotomierte Patienten. In: *13. ITG-/GMA Fachtagung Sensoren und Messsysteme* Institut für Elektrische und Optische Nachrichtentechnik, VDI Verlag GmbH, 2006. – ISBN 978–3–8007–2939–5, S. 395-398

Tabellenverzeichnis

3.1. Ausgewählte Arbeiten der Domäne Fahrzeugdiagnostik und deren Klassifikation.	44
5.1. Verschiedene Kennzahlen der Datenbasis	71
5.2. Ergebnisse der manuellen Untersuchung der Datenbasis	73
6.1. Vertrauenswerte des inferentiellen Diagnostik-Modells nach Definition 6.2 für Links: XOR-ähnliche Daten; Rechts: AND-ähnliche Daten als Vergleich.	89
7.1. Vertrauenswerte des inferentiellen Diagnostik-Modells nach Definition 6.2 für kontextabhängige Daten.	111
7.2. Das XOR-Muster	115
7.3. Realisierbare Verknüpfungsarten und die korrespondierenden Gewichtsregeln für ein Neuron mit 2 Eingängen	115
7.4. Varianten zur Realisierung der XOR-Funktion	116
7.5. (Un)Abhängigkeitsforderungen	118
7.6. Beispieldaten zur Untersuchung des Adaptionsverhaltens	131
9.1. Übersicht der Einzelbewertungen der Konzeptevaluierung	158
9.2. Bewertungssymbole und deren Bedeutung	159
9.3. Ausgabe des Diagnostik-Systems	160
9.4. Zusammenfassung der Ergebnisse der Evaluierung mit Falldaten	177
9.5. Ergebnis der manuellen Evaluierung	181
9.6. Ergebnis der Untersuchung exemplarischer Fälle	184
9.7. Fehlerfall 1 – Diagnostik-Hypothesen von Neucus	185
9.8. Fehlerfall 1 – Prüfanweisungen der GFS und deren Komponentenzuordnungen	185
9.9. Fehlerfall 2 – Diagnostik-Hypothesen von Neucus	186
9.10. Fehlerfall 2 – Prüfanweisungen der GFS und deren Komponentenzuordnungen	186
9.11. Fehlerfall 3 – Diagnostik-Hypothesen von Neucus	187
9.12. Fehlerfall 3 – Prüfanweisungen der GFS und deren Komponentenzuordnungen	187
9.13. Fehlerfall 4 – Diagnostik-Hypothesen von Neucus für ein defektes Spannungsversorgungsrelais des Motorsteuergeräts	188
9.14. Fehlerfall 4 – Prüfanweisungen der GFS für ein defektes Spannungsversorgungsrelais des Motorsteuergeräts	189
9.15. Fehlerfall 5 – Diagnostik-Hypothesen von Neucus	189

9.16. Fehlerfall 5 – Prüfanweisungen der GFS und deren Komponentenzuordnungen	190
9.17. Fehlerfall 6 – Diagnostik-Hypothesen von Neucus	190
9.18. Fehlerfall 6 – Prüfanweisungen der GFS und deren Komponentenzuordnungen	190
9.19. Fehlerfall 7 – Diagnostik-Hypothesen von Neucus	191
9.20. Fehlerfall 7 – Prüfanweisungen der GFS und deren Komponentenzuordnungen	191
9.21. Ergebnis der Praxiserprobung am Fahrzeug hinsichtlich Trefferquote und Gran.	192
10.1. Zusammenfassung der Konzeptbewertung	197
B.1. Verfügbare Informationen zu einem Ereignisspeichereintrag	224
B.2. Beispiel zur Symptom-Kodierung (DTC)	225
B.3. Beispiel zur Symptom-Kodierung (DISS)	226
B.4. Beispiel eines inkongruenten Datensatzes eines realen Reparaturfalls	228
B.5. Beispiel eines kongruenten Datensatzes eines realen Reparaturfalls .	230
B.6. Beispiel eines inkongruenten Datensatzes eines realen Reparaturfalls	231
C.1. Bewertung der Anwendungsfälle und Einsatzbereiche von Neucus ggü. der GFS	233
C.2. Bewertung der technischen Machbarkeit von Neucus ggü. der GFS .	234
C.3. Bewertung der Kosten von Neucus gegenüber der GFS	235
C.4. Bewertung der Leistungsfähigkeit gegenüber der GFS	237
C.5. Bewertung der Anwendbarkeit und Nutzerakzeptanz ggü. der GFS .	238
C.6. Basisdaten des exemplarischen Falls Tiguan 16-895	244
C.7. Basisdaten des exemplarischen Falls Tiguan 44-446	245
C.8. Basisdaten des exemplarischen Falls Tiguan 09-2403	246
C.9. Basisdaten des exemplarischen Falls Golf 01-16683	247
D.1. Vertrauenswerte des inferentiellen Diagnose-Modells für XOR-ähnliche Daten	248
D.2. Vertrauenswerte des inferentiellen Diagnostik-Modells für AND-ähnliche Daten	248
D.3. Vertrauenswerte des inferentiellen Diagnostik-Modells für OR-ähnliche Daten	249
D.4. Verschiedene Diagnostikservicefunktionen der Steuergeräte	249

Abbildungsverzeichnis

1.1. Gliederung	5
2.1. Zusammenstellung einiger Definitionen für den Begriff <i>Diagnostik</i> und <i>Diagnose</i>	9
2.2. Methoden des diagnostischen Schließens	16
2.3. Hypothesize-and-Test-Strategie	20
2.4. Der diagnostische Prozess	22
2.5. Überlagerte diagnostische Prozesse	23
2.6. Typische Vernetzung von Steuergeräten in einem modernen Fahr- zeug [Pet04]	29
2.7. Auswirkungen der Komplexität auf den Service	30
2.8. Anforderungen und Nutzer der Fahrzeugdiagnostik	31
2.9. Prinzipielle Darstellung der Fahrzeugdiagnostik	32
2.10. Taxonomie von Verfahren zur Fehlererkennung in mechatronischen Systemen	33
2.11. Übersicht von Klassifikationsverfahren und Wissensarten	35
2.13. Zusammenhang zwischen Ereignisspeichereinträgen (DTCs), Fehler- suchprogrammen (GFS) und Komponenten (K) in der Geführten Fehlersuche	37
2.14. Links: Beispiel für ein Fehlersuchprogramm der Geführten Fehlersu- che. Rechts: Einige Bildschirmausschnitte	37
3.1. Auswirkungen eines Sensorfehlers	40
3.2. Fehlerfortpflanzung für einen Steckfehler am Gateway	41
3.3. Ausschnitte aus [AD05b]	47
3.4. Ausschnitt aus einer erweiterten FMEA	48
3.5. Das Diagnostic Lifecycle Modell	53
3.6. Der von [BCF ⁺ 08] vorgeschlagene Hybrid-Ansatz zur Diagnostik und Reparatur in Servicewerkstätten.	55
3.7. Ausschnitte aus [KBM08]: Stromlaufplan und Bayes'sches Netz . . .	58
4.1. Datenflussdiagramm des Feedbackprozess	63
4.2. Anwendung der Diagnostik-Modelle	64
4.3. Schematische Darstellung der Gliederung	65
4.4. Generalisierter Reparaturprozess	66
4.5. Reparaturprozess als eine Abbildung von Symptombild S_R auf Kom- ponenten K_R	68
5.1. Anteil der auswertbaren, kongruenten und inkongruenten Reparatur- fälle in der manuellen Untersuchung der Datenbasis	72

5.2. Relevanz von Symptom-Attributen und Maßnahme-Attributen zur Kongruenz eines Datensatzes	72
5.3. Häufigkeitsverteilung von Symptomen und Maßnahmen	76
5.4. Anzahl verschiedener Kontextvarianten pro Steuergerät	77
7.1. Architektur eines mehrschichtigen neuronalen Feedforward-Netzes .	92
7.2. Beispiele für Netzstrukturen und Adjazenzmatrizen	93
7.3. Aufbau eines künstlichen Neurons	94
7.4. Die sigmoide Funktion für verschiedene T	95
7.5. Veranschaulichung des Gradientenabstiegsverfahrens für einen zwei-dimensionalen Suchraum	99
7.6. Einzelnes Neuron als Klassifikator.	107
7.7. Einschichtiges neuronales Diagnostik-Modell nach Definition 6.2 . .	108
7.8. Odds-Ratio und inverses Odds-Ratio im Verhältnis zum Verbindungsgewicht w_{ji}	110
7.9. Multi-Layer-Perceptron als neuronales Diagnostik-Modell	113
7.10. Neuronales Netz für das XOR-Muster	115
7.11. Erweitertes neuronales Diagnostik-Modell	122
7.12. Schematische Darstellung der Vorgehensweise zum Verlernen alter Reparaturfälle	130
7.13. Beispiel des Adaptionsverhaltens nach einer Fehlerabstellmaßnahme, mit und ohne Verlernen	132
7.14. Beispielhaft dargestellter Distanzgraph G und markierter Pfad zwischen s_8 und k_1	135
7.15. Verteilung und akkumulierte Häufigkeit der Phi-Korrelationswerte für Ersatzteilrumpfnummern (ETRN)	136
7.16. Anschauliche Darstellung der Merkmals-Auswahl anhand der Distanzmaße $d_{s_i k_j}$ und dem Parameter d_{th}	138
7.17. Veränderung der Stärke von Verbindungsgewichten durch Nachtraining am Beispiel der Komponente „Schalter für GRA“	141
7.18. Beispiel für Pruning: Teilnetz vor und nach dem Pruning	143
8.1. Screenshot der Trainings- und Validierungsumgebung Neucus, hier mit den Eigenschaften des Operators <i>GetNeucusFalldaten</i>	149
8.2. Screenshot der Analyse-Möglichkeiten, hier ein Histogramm der durch den Operator <i>Signifikanzmatrix</i> erzeugten Daten.	149
8.3. Das Konzept von JPPF	150
8.4. Screenshot der Anwendungsumgebung Neucus-Desktop	153
9.1. Ergebnis der Konzeptevaluierung	157
9.2. Prinzip der automatisierten Evaluierung mittels realer Reparaturfälle	162
9.3. Partitionierung der Falldaten	164
9.4. Evaluierungsquote q_n für verschiedene Maßnahmenklassen und Trainingsepochen	165
9.5. Abhängigkeit der Evaluierungsquote q_n von der steigenden Fallzahl $\#R_{FB}$	165

9.6. Evaluierungsquote q_n der MLP-Diagnostik-Modelle	168
9.7. Evaluierungsquote q_n der komplexitätserweiterten Diagnostik-Modelle	168
9.8. Modifiziertes Lernvorgehen mit Verlernen alter Reparaturfälle . . .	170
9.9. Einfluss des Verlernens alter Reparaturfälle auf die Evaluierungsquote q_n in Abhängigkeit der Maßnahmenklasse und disjunkter Partitionierung	171
9.10. Verteilung und akkumulierte Häufigkeit der Signifikanzwerte für Ersatzteilrumpfnummern (ETRN)	173
9.11. Verteilung und akkumulierte Häufigkeit der Distanzwerte für Ersatzteilrumpfnummern (ETRN)	173
9.12. Zusammenhang zwischen Evaluierungsquote q_n und Anzahl der Verbindungen $\#w_{ij} > 0$ für verschiedene Distanzgrenzen d_{th_i} und Tiefen δ der Merkmals-Reduktion am Beispiel von Ersatzteilrumpfnummern (ETRN)	174
9.13. Reduktion der Anzahl der Verbindungen durch Pruning für Arbeitspositionen (AP) und $Pr_{th} = 0,2$	175
9.14. Einfluss der Pruning-Grenze Pr_{th} (Wert über Markierungen) auf die Evaluierungsquote q_n und die Anzahl der Verbindungen	176
9.15. Screenshot des MS Access-Formulars zur manuellen Evaluierung . .	180
9.16. Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 16-895	182
9.17. Vergleich der Vertrauenswerte v , der Verbindungsgewichte w_{ij} mit den Werten der statistischen Häufigkeit $P(k_i)$ für den Beispielfall Golf, DTC 16-895	183
10.1. Funktionsprinzip nach [MM99]: Zyklische Integration, Verfeinerung und Extraktion von Regeln	203
10.2. Pragmatisches Migrationskonzept von Diagnostik-Modellen und Prüfprogrammen der GFS	207
A.1. Anforderungen und Nutzer der Fahrzeugdiagnostik	214
A.2. Datenerzeugung durch ein einzelnes System	220
A.3. Verschiedene Systemvarianten	220
C.1. Kriterien der Konzeptbewertung	232
C.2. Abhängigkeit der Evaluierungsquote q_n von verschiedenen Trainingsparametern	240
C.3. Streuung der Evaluierungsquote q_n für verschiedene Maßnahmenklassen	241
C.4. Abhängigkeit der Symptommenge $\#S_R$ und der Maßnahmenmenge $\#K_R$ für steigende Fallzahlen $\#R_{FB}$	241
C.5. Poisson-Verteilungen für ein Symptom s_i mit $\tilde{P}(s_i) = 0,018\%$. . .	243
C.6. Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 16-895	244
C.7. Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 44-446	245

C.8. Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 09-2403	246
C.9. Screenshot von Neucus-Desktop mit den Diagnose-Hypothesen für den Ereignisspeichereintrag DTC 01-16683	247

Index Symbole

C	Fahrzeugkontext, 68, 69	$Version_{\Sigma}$	Hardware- und Softwareversionen einer ECU (Menge aller), 77
C_R	Fahrzeugkontext eines Reparaturfalls, 68, 69	W	Gewichtsmatrix (neuronales Netz), 95
C_{Σ}	Fahrzeugkontext (Menge aller), 68	W_I	Gewichtsmatrix zwischen Eingangsschicht und verdeckter Schicht (neuronales Netz), 99
C_{ECU}	Verbauliste (verbaute ECUs), 77	W_O	Gewichtsmatrix zwischen verdeckter Schicht und Ausgangsschicht (neuronales Netz), 99
D_{Σ}	Diagnose-Hypothesen (Menge aller), 18	X	Eingangsvektor beobachtbarer Größen des Systems Σ , 227
E	Ereignis, 16	Y	Eingangsvektor nicht beobachtbarer Größen des Systems Σ , 227
E	Menge der Verbindungen (neuronales Netz), 94	Z	Ausgangsvektor des Systems Σ , 227
G	Gewichteter Graph (neuronales Netz), 94	Z_{Σ}	Zustände der Komponenten (Menge aller), 12
H_D	Diagnose-Hypothesen, 18	$Z_{A,B}$	Zuordnungsfunktion Stimulation $\rightarrow$ Beobachtungen, 14
$H_{DS_{\Sigma}}$	Symptom-Hypothesen (Menge aller), 19	$Z_{O,m}$	Zuordnungsfunktion Observablen $\rightarrow$ Merkmal, 15
H_{DS}	Symptom-Hypothesen, 19	Z_{ist}	Istzustand, 12
K_R	Komponenten eines Reparaturfalls, 69	Z_{soll}	Sollzustand, 12
K_{Σ}	Komponenten (Menge aller), 12, 14	Π	Modell (Diagnostik), 82
M	Merkmale, 15	$\Pi_{\mathfrak{D}}$	Diagnostikmodell, 18
M_{Σ}	Merkmale (Menge aller), 15	$\Pi_{\mathfrak{D}\mathfrak{E}}$	Diagnostikvorgang für nützliche Symptome, 19
$Modell_{\Sigma}$	Fahrzeugmodelle (Menge aller), 77	Σ	System, 12, 14
$N_{H_{DS}}$	Nutzenfunktion einer Symptomhypothese, 20	Θ	Schwellenwert eines Neurons (neuronales Netz), 96
R_{Ω}	Reparaturfälle (Menge aller möglichen), 69	Ξ_{Σ}	Beziehungen zwischen Komponenten (Menge aller), 12
R_{FB}	Reparaturfälle (Menge aller verfügbaren), 69	δ	Fehlerterm (neuronales Netz), 100
S_R	Symptome eines Reparaturfalls, 20	η	Lernrate (neuronales Netz), 100
S_{Σ}	Symptome (Menge aller), 15	$\mathcal{A}$	Aktion/Stimulation, 14
S_I	Initiale Symptomatik, 18	$\mathcal{A}_K$	Alphabet des Ensembles $K \in K_R$ (Komponente), 76
V	Menge der Neuronen, 94		

$\mathcal{A}_S$	Alphabet des Ensembles $S \in S_R$ (Symptom), 76	s	Symptom (Zufallsvariable, $s \in S \in S_R$), 76
$\mathcal{A}_\Sigma$	Stimulation (Menge aller), 14	v	Vertrauenswert, 18
$\mathcal{I}_R$	(Meta-)Informationen eines Reparaturfalls, 69	w	Gewicht einer Verbindung (neuronales Netz), 94
$\mathcal{I}_R$	Informationen eines Reparaturfalls, 69	y_j	Ausgabewert des Neuron j (neuronales Netz), 98
$\mathcal{O}$	Beobachtungen (Menge), 14	z	Zustand(svektor) einer Komponente, 12
$\mathcal{O}$	Beobachtungen (zeitlich veränderlich), 15		
$\mathcal{O}_\Sigma$	Observablen (Menge aller), 14		
$\mathcal{P}_S$	Wahrscheinlichkeiten des Alphabets $\mathcal{A}_S$, 76		
$\mathfrak{P}$	Prozess, datenerzeugend, 227		
a_j	Aktivierung des Zellkerns (Neuron j) (neuronales Netz), 96		
c_{Modell}	Fahrzeugmodell, 77		
$c_{Version}$	Hardware- und Softwareversion einer ECU, 77		
$f_{M,s}$	Zuordnungsfunktion Merkmale $\rightarrow$ Symptom, 15		
$f_{M,s}$	Zuordnungsfunktion Observablen $\rightarrow$ Merkmal, 15		
$f_{\mathcal{O},s}$	Zuordnungsfunktion Observablen $\rightarrow$ Symptom, 15		
f_{act}	Aktivierungsfunktion des Neuron j (neuronales Netz), 96		
f_{out}	Ausgabefunktion des Neuron j (neuronales Netz), 98		
f_{prop}	Propagierungsfunktion eines Neurons j (neuronales Netz), 96		
h	Diagnose-Hypothese (einzelne), 18		
h_{DS}	Symptom-Hypothese (einzelne), 19		
k	Komponente, 12		
m	Merkmal, 15		
m	Merkmale (einzelnes), 15		
o	Beobachtung (einzelne), 14		
r	Reparaturfall, 69		
r_j	Reizwert des Neurons j (neuronales Netz), 96		
r_j	Reizwert des Zellkerns (Neuron j) (neuronales Netz), 96		
s	Symptom, 15		

Literatur

- [ABMH08] ALTHOFF, Klaus-Dieter (Hrsg.); BERGMANN, Ralph (Hrsg.); MINOR, Mirjam (Hrsg.); HANFT, Alexandre (Hrsg.): *Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008, Trier, Germany, September 1-4, 2008. Proceedings*. Bd. 5239. Springer, 2008 (Lecture Notes in Computer Science). – ISBN 978-3-540-85501-9.
- [ABP06] ATZMUELLER, Martin; BAUMEISTER, Joachim; PUPPE, Frank: Semi-automatic learning of simple diagnostic scores utilizing complexity measures. In: *Artificial Intelligence in Medicine* 37 (2006), Nr. 1, S. 19–30. – ISSN 0933-3657.
- [Abu03] ABUOSBA, Dr. M.: Fahrzeug-Diagnose mit System. In: *Wissensmanagement* Sonderdruck 8/03 (2003), S. 44 – 45.
- [AD05a] ASSAF, Tariq; DUGAN, Joanne B.: Automatic Diagnosis via Sensors Modeled by Dynamic Fault Trees. In: *SAE Transactions Journal of Passenger Cars: Electronic and Electrical Systems* 2005-01-1442 (2005), S. 388–394.
- [AD05b] ASSAF, Tariq; DUGAN, Joanne B.: Build Better Diagnostic Decision Trees. In: *IEEE Instrumentation & Measurement Magazine* (2005), S. 48–53.
- [AHK97] ANDRESS, Hans-Jürgen; HAGENAARS, Jacques A.; KÜHNEL, Steffen: *Analyse von Tabellen und kategorialen Daten*. Springer, 1997. – 455 S. – ISBN 978-3-540-62515-5.
- [ARB⁺07] ALTHOFF, Klaus-Dieter; REICHLE, Meike; BACH, Kerstin; HANFT, Alexandre; NEWO, Regis: Agent Based Maintenance for Modularised Case Bases in Collaborative Multi-Expert Systems. In: *Proceedings of AI2007, 12th UK Workshop on Case-Based Reasoning*, 2007, S. 7–18.
- [Ash89] ASH, Timur: Dynamic node creation in backpropagation networks. In: *Connection Science* 4 (1989), Nr. 1, S. 365–375.
- [AUThra] VDI: *Steuerung und Regelung von Fahrzeugen und Motoren - AUTOREG*. Baden-Baden, jedes Jahr. – Fachtagung.
- [AUThrb] IEEE: *Systems Readiness Technology Conference - IEEE AUTO-TESTCON*. Baltimore, USA, jedes Jahr. – Konferenz.

- [BAB⁺03] BERGMANN, Ralph; ALTHOFF, Klaus-Dieter; BREEN, Sean; GÖCKER, Mehmet H.; MANAGO, Michel; TRAPHÜNER, Ralph; WESS, Stefan: *Lecture Notes in Computer Science*. Bd. 1612: *Developing Industrial Case-Based Reasoning Applications: The INRECA-Methodology*. Springer-Verlag Berlin Heidelberg NewYork, 2003. – 236 S. – ISBN 3-540-20737-6.
- [Bal98] BALZERT, Helmut: *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Heidelberg Berlin : Spektrum Akademischer Verlag, 1998. – ISBN 3-8274-0065-1.
- [BAP02] BAUMEISTER, Joachim; ATZMÜLLER, Martin; PUPPE, Frank: Inductive Learning for Case-Based Diagnosis with Multiple Faults. In: *Advances in Case-Based Reasoning* Bd. 2416/2002, Springer Berlin / Heidelberg, 2002, S. 173–216.
- [Bar94] BARTLETT, Eric B.: Dynamic node architecture learning: an information theoretic approach. In: *Journal of Neural Networks* 7 (1994), Nr. 1, S. 129–140. – ISSN 0893-6080.
- [BC04] BAEUMLE-COURTH, Peter: *Approximation, Reduktion und Regelextraktion: Semantikbeschreibung für Neuronale Netze*, Münster, Diss., 2004.
- [BCF⁺08] BANDINI, Stefania; COLOMBO, Ettore; FRISONI, Giuseppe; SARTORI, Fabio; SVENSSON, Joakim: Case-Based Troubleshooting in the Automotive Context: The SMMART Project. In: *ECCBR*, 2008, S. 600–614.
- [Ber68] BERTALANFFY, Ludwig von: *General system theory*. New York : George Braziller, 1968.
- [BF91] BELLIDO, Ignacio; FERNÁNDEZ, Gregorio: Backpropagation Growing Networks: Towards Local Minima Elimination. In: *IWANN '91: Proceedings of the International Workshop on Artificial Neural Networks*. London, UK : Springer-Verlag, 1991. – ISBN 3-540-54537-9, S. 130–135.
- [BH98] BALAKRISHNAN, Karthik; HONAVAR, Vasant: Intelligent Diagnosis Systems. In: *Journal of Intelligent Systems* 8 (1998), S. 239–290.
- [Bis06] BISHOP, Christopher M.; JORDAN, M. (Hrsg.); KLEINBERG, J. (Hrsg.); SHOLKOPF, B. (Hrsg.): *Pattern Recognition and Machine Learning*. Springer, 2006. – ISBN 978-0387-31073-2.
- [BKI08] BEIERLE, Christoph; KERN-ISBERNER, Gabriele: *Methoden wissensbasierter Systeme - Grundlagen, Algorithmen, Anwendungen*. 4., verbesserte Auflage : Vieweg+Teubner Verlag, 2008.

- [BL90] BAUM, Eric B.; LANG, Kevin J.: Constructing hidden units using examples and queries. In: *NIPS-3: Proceedings of the 1990 conference on Advances in neural information processing systems 3*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1990. – ISBN 1–55860–184–8, S. 904–910.
- [Bor04] BORTH, Michael: *Wissensgewinnung auf Bayes-Netz-Mengen*, Universität Ulm, Abteilung für Neuroinformatik, Fakultät für Informatik, Diss., 2004.
- [Bor08] BORGEEST, Kai: *Elektronik in der Fahrzeugtechnik*. Vieweg+Teubner, 2008 (ATZ-MTZ-Fachbuch). – ISBN 978–3–8348–0207–1.
- [BP99] BAMBERGER, Stefan; PUPPE, Frank: Kooperierende Diagnoseagenten. In: *Computer Science - Research and Development* Bd. 14, Springer Berlin / Heidelberg, Januar 1999, S. 135–144.
- [BR07] BECK, Fabian; REY, Günter D.: *Neuronale Netze: Eine Einführung*. Website. <http://www.neuronalesnetz.de>. Version: Oktober 2007.
- [Bra97] BRAUN, Heinrich: *Neuronale Netze: Optimierung durch Lernen und Evolution*. Springer Verlag, 1997.
- [Bre91] BRENT, Richard P.: Fast training algorithms for multilayer neural nets. In: *IEEE Transactions on Neural Networks* 2 (1991), Nr. 3, S. 346–354. – ISSN 1045–9227.
- [Bro06] BROCKHAUS; F.A. BROCKHAUS AG MANNHEIM F.A. BROCKHAUS GMBH, Bibliografisches I. (Hrsg.): *Brockhaus Enzyklopädie in 30 Bänden*. 21. F.A. Brockhaus GmbH, Leipzig, Bibliografisches Institut und F.A. Brockhaus AG, Mannheim, 2006.
- [BRRSA08] BACH, Kerstin; REICHLE, Meike; REICHLE-SCHMEHL, Alexander; ALTHOFF, Klaus-Dieter: Implementing a Coordination Agent for Modularised Case Bases. In: PETRIDIS, Miltos (Hrsg.); WIRATUNGA, Nirmalie (Hrsg.): *Proceedings of the 13th UK Workshop on Case-Based Reasoning (at the Artificial Intelligence Conference, AI 2008)*, 2008, S. 1–12.
- [BS84] BUCHANAN, Bruce G.; SHORTLIFFE, Edward H.: *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1984 ((The Addison-Wesley series in artificial intelligence)). – ISBN 0201101726.
- [BSP04] BAUMEISTER, Joachim; SEIPEL, Dietmar; PUPPE, Frank: Using Automated Tests and Restructuring Methods for an Agile Development of Diagnostic Knowledge Systems. In: *Proc. of the 17th International Florida Artificial Intelligence Research Society Confe-*

rence (*FLAIRS-2004*), 2004, S. 319–324.

- [BZ92] BAFFES, Paul; ZELLE, John: Growing Layers of Perceptrons: Introducing the Extentron Algorithm. In: *Proceedings of the 1992 International Joint Conference on Neural Networks*, 1992, S. 392–397.
- [Büh34] BÜHLER, Karl: *Sprachtheorie: Die Darstellungsfunktion der Sprache*. Stuttgart : Lucius & Lucius Verlagsgesellschaft mbH, 1934. – ISBN 3–8252–1159–2.
- [CB09] CHEN, Daqing; BURRELL, Phillip; MACINTYRE, John (Hrsg.): *Neural Computing and Applications*. Springer London, 2009. ISSN 0941–0643.
- [Chi97] CHIU, S.: Developing commercial applications of intelligent control. In: *Control Systems Magazine, IEEE* 17 (1997), Nr. 2, S. 94–100. – ISSN 0272–1708.
- [Chr97] CHRISTENSEN, R.: *Log-Linear Models and Logistic Regression*. 2nd. Springer Verlag, 1997. – ISBN 0–387–98247–7.
- [CIVhr] IEEE: *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems - CIVVS*. Nashville, TN, USA, jedes Jahr. – Workshop.
- [CM98] CHERKASSKY, V.; MULIER, F.: *Learning from Data*. John Wiley & Sons, 1998. – ISBN 0471154938.
- [CMM84] CARBONELL, J. G.; MICHALSKI, R. S.; MITCHELL, T. M.: An Overview of Machine Learning. In: MICHALSKI, R. S. (Hrsg.); CARBONELL, J. G. (Hrsg.); MITCHELL, T. M. (Hrsg.): *Machine Learning: An Artificial Intelligence Approach*. Berlin, Heidelberg : Springer, 1984, S. 3–23.
- [Cox46] COX, Richard T.: Probability, Frequency, and Reasonable Expectation. In: *American Journal of Physics* 14 (1946), S. 1–13.
- [CZ00] CLOETE, Ian (Hrsg.); ZURADA, Jacek M. (Hrsg.): *Knowledge-based Neurocomputing*. Cambridge, MA, USA : MIT Press, 2000. – 500 S. – ISBN 978–0262032742.
- [Def90] DEFFUANT, Guillaume: Neural units recruitment algorithm for generation of decision trees. In: *Proceedings of the Int. Joint Conf. on Neural Networks* Bd. 1, 1990, S. 637–642.
- [DHS00] DUDA, Richard O.; HART, Peter E.; STORK, David G.: *Pattern Classification*. 2nd. Wiley-Interscience, 2000. – ISBN 0471056693.
- [Diahra] Haus der Technik: *Diagnose in mechatronischen Fahrzeugsystemen*. München, jedes Jahr. – Fachtagung.

- [Diahrb] CTI: *Internationale CTI Fachkonferenz „Diagnosesysteme im Automobil“*. Stuttgart, jedes Jahr. – Fachkonferenz.
- [DIN08] Norm 2008. *Begriffe der Qualitätssicherung und Statistik*. – DIN 55350.
- [DLS⁺72] DOMBAL, F. T.; LEAPER, D. J.; STANILAND, J. R.; MCCANN, A. P.; HORROCKS, J. C.: Computer-aided diagnosis of acute abdominal pain. In: *Br Med J* 2 (1972), April, Nr. 5804, S. 9–13. – ISSN 0007–1447.
- [DP99] DRESSLER, Oskar; PUPPE, Frank: Knowledge-Based Diagnosis - Survey and Future Directions. In: *XPS-99: Knowledge-Based Systems* Bd. 1570/1999, Springer Berlin / Heidelberg, 1999. – ISBN 978–3–540–65658–6, S. 24–46.
- [Dre01] DREISEITL, Stephan: *Funktionsminimierung mittels Gradientenabstieg*. Website. <http://staff.fh-hagenberg.at/sdreisei/Teaching/WS2001-2002/>. Version: 2001.
- [Dre05] DREYFUS, Gérard: *Neural networks: methodology and applications*. 1. Springer, 2005. – 386 S. – ISBN 978–3540229803.
- [Duc] DUCH, Wlodzislaw: *Software & Databases for AI & Machine Learning*. Website. <http://www.is.umk.pl/~duch/software.html>. – Stand 22.04.2010.
- [Duc26] DUCASSE, C. J.: Explanation, mechanism, and teleology. In: *Readings in Philosophical Analysis* (1926), S. 540–544.
- [EHN⁺98] ENNIS, Marguerite; HINTON, Geoffrey; NAYLOR, David; REVOW, Mike; TIBSHIRANI, Robert: A comparison of statistical learning methods on the GUSTO database. In: *Statistics in Medicine* 17 (1998), Dezember, Nr. 21, S. 2501–2508.
- [EIKre] VDI: *Internationaler Kongress: Elektronik im Kraftfahrzeug*. Baden-Baden, alle zwei Jahre. – Kongress.
- [Fay96] FAYYAD, Usama M.: Data Mining and Knowledge Discovery: Making Sense Out of Data. In: *IEEE Intelligent Systems* 11 (1996), S. 20–25. – ISSN 0885–9000.
- [FB89] FELBER, Helmut; BUDIN, Gerhard: *Terminologie in Theorie und Praxis*. Gunter Narr Verlag Tübingen, 1989. – 315 S. – ISBN 978–3878087830.
- [FE94] FAHNER, Gerald; ECKMILLER, Rolf: Structural adaptation of parsimonious higher-order neural classifiers. In: *Journal of Neural Networks* 7 (1994), Nr. 2, S. 279–289. – ISSN 0893–6080.

- [FHZ92] FINNOFF, William; HERGERT, Ferdinand; ZIMMERMANN, Hans G.: A comparison of weight elimination methods for reducing complexity in neural networks. In: *Proceedings of the International Joint Conference on Neural Networks* Bd. 3. Baltimore, 1992, S. 980–987.
- [FHZ93] FINNOFF, William; HERGERT, Ferdinand; ZIMMERMANN, Hans-Georg: Improving model selection by nonconvergent methods. In: *Neural Networks* 6 (1993), Nr. 6, S. 771–783.
- [Fie93] FIESLER, E.: Minimal and high order neural network topologies. In: *Proceedings of the 1993 Int. Simulation Technology Conference (SimTec 93)*. San Diego, 1993, S. 173–178.
- [FJ05] FORAN, Thomas James DENSO Sales U.; JACKMAN, Brendan Waterford Institute of T.: An Intelligent Diagnostic System for Distributed, Multi-ECU Automotive Control Systems. In: *SAE Transactions Journal of Passenger Cars: Electronic and Electrical Systems* (2005), S. 395–402.
- [FK93] FORBUS, Kenneth D.; KLEER, Johan D.: *Building Problem Solvers*. Cambridge, MA, USA : MIT Press, 1993. – ISBN 0262560720.
- [FL89a] FAHLMAN, Scott E.; LEBIERE, Christian: The Cascade-Correlation Learning Architecture. In: *Advances in Neural Information Processing Systems 2, Conference, Denver, Colorado, USA*, 1989, S. 524–532.
- [FL89b] FERNANDO, Manoel; LEE, Tenorio Wei-Tsih: Self organizing neural networks for the identification problem. In: *Advances in neural information processing systems* Bd. 1. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1989. – ISBN 1–558–60015–9, S. 57 – 64.
- [FL91] FAHLMAN, Scott E.; LEBIERE, Christian: The Cascade-Correlation Learning Architecture / School of Computer Science, Carnegie Mellon University, Pittsburgh. 1991. – Forschungsbericht.
- [Fre90] FREAN, Marcus: The upstart algorithm: a method for constructing and training feedforward neural networks. In: *Neural Comput.* 2 (1990), Nr. 2, S. 198–209. – ISSN 0899–7667.
- [Fro05] FROMM, Sabine: *Binäre logistische Regressionsanalyse*. Bd. 11. Bamberg : Otto-Friedrich-Universität Bamberg, 2005.
- [FS06] FELDMAN, Ronen; SANGER, James: *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2006. – ISBN 0521836573.
- [Fu93] FU, L.M.: Knowledge-based connectionism for revising domain theories. In: *Systems, Man and Cybernetics, IEEE Transactions on*

- 23 (1993), jan/feb, Nr. 1, S. 173 –182.
- [Gal86] GALLANT, Stephen I.: Three constructive algorithms for network learning. In: *Proc. 8th Ann Conf of Cognitive Science Society*, 1986, S. 652–660.
- [Gal93] GALLANT, Stephen I.: *Neural Network Learning and Expert Systems*. The MIT Press, 1993. – ISBN 0262071452.
- [Gha98] GHALWASH, Atef Z.: A Recency Inference Engine for Connectionist Knowledge Bases. In: *Applied Intelligence* 9 (1998), Nr. 3, S. 201–215.
- [GHP05] GÖKER, Mehmet H.; HOWLETT, Robert J.; PRICE, Joseph E.: Case-based reasoning for diagnosis applications. In: *The Knowledge Engineering Review* 20 (2005), Nr. 03, S. 277–281.
- [Gmb07] GMBH, Robert B. (Hrsg.): *Autoelektrik / Autoelektronik*. 5. Vieweg Verlag, 2007. – 530 S. – ISBN 3528238720.
- [GRF07] GUSIKHIN, Oleg; RYCHTYCKYJ, Nestor; FILEV, Dimitar: Intelligent systems in the automotive industry: applications and trends. In: *Knowl. Inf. Syst.* 12 (2007), Nr. 2, S. 147–168. – ISSN 0219–1377.
- [GRS03] GÖRZ, Günther (Hrsg.); ROLLINGER, Claus-Rainer (Hrsg.); SCHNEEBERGER, Josef (Hrsg.): *Handbuch der Künstlichen Intelligenz*. 4. München : Oldenbourg, 2003. – ISBN 978–3–486–27212–3.
- [Hag90] HAGIWARA, Masafumi: Novel back propagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection. In: *Proceedings of the International Joint Conference on Neural Network* Bd. 1, 1990, S. 625–630.
- [HAJ⁺06] HUANG, Yingping; ANTORY, David; JONES, R. P.; GROOM, Craig; MCMURRAN, Ross; PETER, Earp; MCKINNEY, Francis: Bayesian Belief Network Based Fault Diagnosis in Automotive Electronic Systems / -International Automotive Research Centre (IARC), Warwick Manufacturing Group, University of Warwick. -School of Engineering and IARC, University of Warwick -Jaguar Cars Ltd. 2006. – Forschungsbericht. – AVEC ‘06.
- [Han89] HANSON, Stephen J.: Meiosis Networks. In: *Advances in Neural Information Processing Systems 2, Conference, Denver, Colorado, USA*, 1989, S. 533–541.
- [Har03] HARMANN, Hans-Peter: *Untersuchung und Modellierung der Ionenstrahlformung großflächiger Ionenquellen mit Hilfe einer beweglichen Faradaysondenzeile*, Universität Gießen, Diss., 2003.
- [HDN06] HACK, Christian; DRENKAHN, Georg; NICKLES, Jochen: Vernetzte Funktionen beherrschen - Diagnoseanforderung zur Bewältigung der Komplexität vernetzter Fahrzeugsysteme. In: *Automotive, Messen*

und Testen (2006).

- [Heb49] HEBB, Donald O.: *The Organization of Behavior: A Neuropsychological Theory*. New York : John Wiley & Sons Inc, 1949. – ISBN 978–750–8400.
- [Hei99] HEINZELMANN, Andreas: *Produktintegrierte Diagnose komplexer mobiler Systeme*, Universität Padaborn, Diss., 1999.
- [HH88] HOYNINGEN-HUENE, Paul: Kommunikation in der Wissenschaft: Fakten und Probleme. In: *Communications* 14 (1988), Februar, Nr. 2, S. 133–144. – ISSN 1613–4087.
- [HL00] HOSMER, David W.; LEMESHOW, Stanley: *Applied logistic regression*. Wiley-Interscience Publication, 2000 (Wiley Series in probability and statistics). – ISBN 0471356328.
- [Hof93] HOFFMANN, Norbert: *Kleines Handbuch, Neuronale Netze*. Braunschweig/Wiesbaden, Germany : Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 1993.
- [HS93] HASSIBI, Babak; STORK, David G.: Second Order Derivatives for Network Pruning: Optimal Brain Surgeon. In: GILES, C. L. (Hrsg.); HANSON, S. J. (Hrsg.); COWAN, J. D. (Hrsg.): *Advances in Neural Information Processing Systems 5*. San Mateo, California : Morgan Kaufmann Publishers Inc., 1993. – ISBN 1–55860–274–7, S. 164–171.
- [HS00] HOTZ, Lothar; STRUSS, Peter; GUCKENBIEHL, Thomas (Hrsg.): *INDIA - Intelligente Diagnose in der Anwendung*. Aachen : Shaker Verlag, 2000. – 283 S. – ISBN 3–8265–7002–2.
- [HSW89] HORNIK, K.; STINCHCOMBE, M.; WHITE, H.: Multilayer feed-forward networks are universal approximators. In: *Journal Neural Networks* 2 (1989), Nr. 5, S. 359–366. – ISSN 0893–6080.
- [Hud04] HUDI, Ricky: Erforderliche Prozesse und Kernkompetenzen zur Produktion komplexer Elektronik-Fahrzeuge. In: *Fachkongress Automobil Elektronik*. Ludwigsburg, 2004.
- [Hus01] HUSEMEYER, Uwe: *Heuristische Diagnose mit Assoziationsregeln*, Universität Paderborn, Diss., 2001.
- [HYH91] HIROSE, Yoshio; YAMASHITA, Koichi; HIJIYA, Shimpei: Back-propagation algorithm which varies the number of hidden units. In: *Journal of Neural Networks* 4 (1991), Nr. 1, S. 61–66. – ISSN 0893–6080.
- [Hüb02] HÜBSCH, Rico: *Analyse der Erweiterungsmöglichkeiten bestehender Kundendienst-Systemarchitekturen zur gebündelten Informationsbereitstellung in zukünftigen Fahrzeugprojekten am Beispiel der Volkswagen AG*, Hochschule für Technik und Wirtschaft Dresden (FH),

Diplomarbeit, 2002.

- [IHSK03] ISERMANN, Rolf; HARTMANSHENN, E.; SCHWARTE, A.; KIMMICH, F.: Fehlerdiagnosemethoden für Diesel- und Ottomotoren. In: *12. Aachener Kolloquium Fahrzeug- und Motorentechnik*, 2003, S. 407–448.
- [Ise96] ISERMANN, Rolf: Modellgestützte Überwachung und Fehlerdiagnose Technischer Systeme. In: *atp Automatisierungstechnische Praxis* 38 (1996), Nr. 5, 6.
- [Ise06] ISERMANN, Rolf: *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer Berlin Heidelberg, 2006. – ISBN 978-3-540-24112-6.
- [ISO06] Norm 2006. *ISO 15031: Road vehicles - Communication between vehicle and external equipment for emissions-related diagnostics*.
- [Jen02] JENSEN, Finn V.: *Bayesian Networks and Decision Graphs*. Corrected. Springer, 2002 (Information Science and Statistics). – ISBN 0387952594.
- [JJ90] JI, Jiahuang; JENG, Menkae: Bin-packing adjustable rectangles and applications to task scheduling on partitionable parallel computers. In: *Proceedings of the Second IEEE Symposium on Parallel and Distributed Processing*. Dallas, Texas, USA : IEEE Computer Society Press, Dezember 1990, S. 312–315.
- [JL04] JÄGER, Ludwig; LINZ, Erika: *Medialität und Mentalität: theoretische und empirische Studien zum Verhältnis von Sprache, Subjektivität und Kognition*. Wilhelm Fink Verlag, 2004. – 369 S. – ISBN 978-3770537457.
- [Jon08] JONES, Capers: *Applied Software Measurement: Global Analysis of Productivity and Quality*. 3. McGraw-Hill Osborne Media, 2008. – ISBN 978-0071502443.
- [JPP09] JPPF.ORG: JPPF 2.0 Manual / JPPF.org. 2009. – Manual.
- [JPP10] JPPF.ORG: *JPPF - Java Parallel Processing Framework*. Website. <http://www.jppf.org>. Version: 22.03.2010.
- [Kar92] KARAYIANNIS, Nicolaos B.: ALADIN: Algorithms for learning and architecture determination. In: *Procedings International Joint Conference on Neural Networks* Bd. 1. Baltimore, 1992, S. 601–606.
- [KBL⁺07] KRIEGER, Olaf; BREUER, Andreas; LANGE, Klaus; MÜLLER, Tobias C.; FORM, Thomas: Wahrscheinlichkeitsbasierte Fahrzeugdiagnose auf Basis individuell generierter Prüfabläufe. In: *Mechatronik 2007 - Innovative Produktentwicklung*. Wiesloch bei Heidelberg : VDI Verlag GmbH, Mai 2007. – ISBN 978-3-18-091971-3, S. 235–248.

- [KBM08] KRIEGER, Olaf; BREUER, Andreas; MÜLLER, Tobias C.: Intelligente Fahrzeugdiagnose mit variabler Prüfstrategie. In: *Diagnose in mechatronischen Fahrzeugsystemen* Bd. 1. München : Expert Verlag, Juli 2008. – ISBN 978-3-8169-2821-8, S. 130–141.
- [KF08] KINDEL, Olaf; FRIEDRICH, Mario: *Softwareentwicklung mit AUTOSAR: Grundlagen, Engineering, Management in der Praxis*. 1. dpunkt.verlag GmbH, 2008. – 292 S. – ISBN 3898645630.
- [KH05] KRISHNAPURAM, Balaji; HARTEMINK, Alexander J.: Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (2005), June, Nr. 6, S. 957–968. – ISSN 0162-8828.
- [KK02] KLEINBAUM, David G.; KLEIN, Mitchel: *Logistic Regression*. 2. Springer-Verlag New York, Inc., 2002. – ISBN 978-0-387-95397-7.
- [KKB93] KUMAR, P. S.; KUMAR, M. K.; BASU, A.: Parallel Algorithms for Sparse Triangular System Solution. In: *Parallel Computing* 19 (1993), Nr. 2, S. 187–196.
- [KKKC04] KORBICZ, Jozef (Hrsg.); KOSCIELNY, Jan M. (Hrsg.); KOWALCZUK, Zdzislaw (Hrsg.); CHOLEWA, Wojciech (Hrsg.): *Fault Diagnosis: Models, Artificial Intelligence, Applications*. 1. Springer, 2004. – 920 S. – ISBN 3540407677.
- [KKL⁺08] KRISHNA, R. P.; KODALI, Anuradha; LUO, Jianhui; CHOI, Kihoon; SINGH, Satnam; SANKAVARAM, Chaitanya; MANDAL, Suvasri; DONAT, William; NAMBURU, Setu M.; CHIGUSA, Shunsuke; QIAO, Liu: An Integrated Diagnostic Process for Automotive Systems. In: PROKHOROV, Danil V. (Hrsg.): *Computational Intelligence in Automotive Applications* Bd. 132. Springer, 2008. – ISBN 978-3-540-79256-7, S. 191–218.
- [KKP91] KONG, Xiangyun; KLAPPHOLZ, David; PSARRIS, Kleanthis: The I Test: An Improved Dependence Test for Automatic Parallelization and Vectorization. In: *IEEE Transactions Parallel Distributed Systems* 2 (1991), Nr. 3, S. 342–349.
- [KLFM09] KRIEGER, Olaf; LANGE, Klaus; FORM, Thomas; MÜLLER, Tobias C.: Wahrscheinlichkeitsbasierte Fahrzeugdiagnose mit automatischer Generierung von Prüfanweisungen. In: *14. Internationale Konferenz Elektronik im Kraftfahrzeug, VDI-Berichte 2075*. Baden Baden : VDI Verlag GmbH, 2009. – ISBN 978-3-18-092075-7, S. 333–344.
- [Kli01] KLIR, George J.: *Facets of Systems Science (IFSR International Series on Systems Science and Engineering)*. Springer, 2001. – ISBN 0306466236.

- [KMR90] KLEER, Johan de; MACKWORTH, Alan K.; REITER, Raymond: Characterizing Diagnoses. (1990), S. 324–330.
- [Koc00] KOCH, Karl-Rudolf: *Einführung in die Bayes-Statistik*. 1. Springer Verlag Berlin, 2000. – 225 S. – ISBN 978–3540666707.
- [Koh07] KOHUT, Roman: *Neuronale Netze als Modell Boolescher Funktionen*, TU Bergakademie Freiberg, Fakultät für Mathematik und Informatik, Dissertation, 2007.
- [Kol93] KOLODNER, Janet: *Case-Based Reasoning*. Morgan Kaufmann, 1993. – ISBN 1558602372.
- [Kor05] KORDON, A.K.: Application issues of industrial soft computing systems. In: *Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American.*, 2005. – ISBN 0–7803–9187–X, S. 110–115.
- [Kre01] KRELL, Andrei: *Neuronale Netze und deren Eignung zur Klassifikation im Data Mining*. Technische Universität Chemnitz, Chemnitz : Fakultät für Informatik, 2001.
- [Kri07] KRIESEL, David: *Ein Überblick über Neuronale Netze*. Website. <http://www.dkriesel.com>. Version: 2007.
- [Kriar] KRIEGER, Olaf: *Wahrscheinlichkeitsbasierte Fahrzeugdiagnose mit individueller Prüfstrategie*. Magdeburg, Technische Universität Braunschweig, Dissertation, to appear.
- [Krü06] KRÜGER, Thomas: *Entwurf eines Flugreglers auf Basis Künstlicher Intelligenz, Diplomarbeit*, Technische Universität Braunschweig, Diplomarbeit, 2006.
- [KS93] KLAGGERS, H; SOEGTROP, M: Limited Fan-In Random Wired Cascade-Correlation. In: *Neuroprose Archive, IBM Research Division, Physics Group*. München, 1993, S. 79–82.
- [KS97] KÖPPEN-SELIGER, Birgit: *Fehlerdiagnose mit künstlichen neuronalen Netzen*. Düsseldorf, GH-Duisburg, Dissertation, 1997.
- [Kub06] KUBACH, Timo A.: *Entwicklung einer auf Laser-induzierter Fluoreszenz basierenden Diagnostik zur Untersuchung von Edge Localized Modes im Divertor-Plasma von ASDEX Upgrade*, Universität Stuttgart, Diss., 2006.
- [Kur89] KURBEL, Karl: *Entwicklung und Einsatz von Expertensystemen*. Berlin : Springer, 1989. – ISBN 3540510133.
- [KW65] KIEPENHEUER; WITSCH: *Die Motivation menschlichen Handelns*. Köln : H. Thomae, 1965.

- [LAB⁺96] LENZ, Mario; AURIOL, Eric; BURKHARD, Hans-Dieter; MANAGO, Michel; PIRK, Petra: Fallbasierte Diagnostik und Entscheidungsunterstützung. In: *KI* 10 (1996), Nr. 1, S. 16–21.
- [LDS89] LECUN, Yann; DENKER, John S.; SOLLA, Sara A.: Optimal Brain Damage. In: *Advances in Neural Information Processing Systems 2*. Denver, Colorado, USA, November 1989, S. 598–605.
- [LDS90] LECUN, Y.; DENKER, J.S.; SOLLA, S. A.: Optimal Brain Damage. In: TOURETZKY, D.S. (Hrsg.): *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann, 1990, S. 598–605.
- [LLM93] LEVIN, Asriel U.; LEEN, Todd K.; MOODY, John E.: Fast Pruning Using Principal Components. In: *Advances in Neural Information Processing Systems 6*. Denver, Colorado, USA, 1993, S. 35–42.
- [LLS00] LIM, Tjen-Sien; LOH, Wei-Yin; SHIH, Yu-Shan: A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. In: *Machine Learning* 40 (2000), Nr. 3, S. 203–228. – ISSN 0885–6125.
- [LR92] LITTMANN, Enno; RITTER, Helge: Cascade Network Architectures. In: *International Joint Conference On Neural Networks*, 1992, S. 398–404.
- [LW07] LUNDKVIST, Joella; WAHNSTRÖM, Stina: *Dynamic Model Based Diagnosis for Combustion Engines in RODON*, Linköpings Universitet, Diplomarbeit, 2007.
- [Lük03] LÜKE, Stefan: *Dezentraler Diagnoseansatz für dynamische und mechatronische Systeme*, TU Dresden, Diss., 2003.
- [Mac03] MACKAY, David J.: *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [Mac07] MACKAY: *Bayesian methods for neural networks - FAQ*. Website, Oktober 2007.
- [Mahch] MAHMUD, Syed (Hrsg.): *SAE Transactions Journal of Passenger Cars: Electronic and Electrical Systems*. SAE International, halbjährlich.
- [MAS03] MEIRELES, M.R.G.; ALMEIDA, P.E.M.; SIMOES, M.G.: A comprehensive review for industrial applicability of artificial neural networks. In: *IEEE Transactions on Industrial Electronics* 50 (2003), Nr. 3, S. 585–601. – ISSN 0278–0046.
- [Mat02] MATTHIES, Prof. Dr. M.: *Einführung in die Systemwissenschaft (Vorlesungsskript)*. 2002.

- [Mec] VDI: *Mechatronik - Innovative Produktentwicklung*. Wiesloch bei Heidelberg, . – Fachtagung.
- [Men01] MENARD, Scott W.: *Applied Logistic Regression Analysis*. 2. Sage Pubn Inc., 2001. – 128 S. – ISBN 978-0761922087.
- [Met09] METTE, Andreas: *Entwicklung und Validierung verschiedener Verfahren mit neuronalen Netzen für die Fehlerdiagnose komplexer Fahrzeugsysteme*. Braunschweig, TU Braunschweig, Diplomarbeit, 2009.
- [MFK08] MÜLLER, Tobias C.; FORM, Thomas; KRIEGER, Olaf: Evaluierung von Offboard-Diagnosesystemen am Beispiel eines erfahrungsbasierten Diagnoseverfahrens. In: GÜHMANN, Clemens (Hrsg.): *Simulation und Test in der Funktions- und Softwareentwicklung für die Automobilelektronik* Bd. 2, Expert Verlag, 2008. – ISBN 978-3-8169-2818-8, S. 388–405.
- [Mic02] MICHLER, Thorsten: *Bionisches Diagnoseassistenzsystem für die Dauererprobung autonom fahrender Personenkraftwagen.pdf*. Braunschweig, Technische Universität Braunschweig, Dissertation, 2002.
- [Mit97] MITCHELL, T.: *Machine Learning*. McGraw-Hill Education (ISE Editions), 1997. – ISBN 0071154671.
- [MM99] MCGARRY, Kenneth J.; MACINTYRE, John: The Fusion of Neuro-Symbolic Knowledge within an Adaptive Hybrid System / School of Computing, Engineering and Technology, University of Sunderland, England. 1999. – Forschungsbericht.
- [MN89] MEZARD, M; NADAL, Jean-P: Learning in feedforward layered networks: the tiling algorithm. In: *Journal of Physics A: Mathematical and General* 22 (1989), Nr. 12, S. 2191.
- [Moo94] MOODY, John: Prediction Risk and Architecture Selection for Neural Networks. In: CHERKASSKY, V. (Hrsg.); FRIEDMAN, J.H. (Hrsg.); WECHSLER, H. (Hrsg.): *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*. Springer-Verlag, 1994, S. 147–165.
- [MP43] *Kapitel A logical calculus of ideas immanent in nervous activity*. In: MCCULLOCH, Warren S.; PITTS, Walter: *A logical calculus of ideas immanent in nervous activity*. 1943, S. 115–133.
- [MP69] MINSKY, Marvin; PAPERT, Seymour: *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.
- [MPI08] MAX-PLANCK-INSTITUT: *Bayessche Wahrscheinlichkeitstheorie-Theorie*. Website. <http://www.ipp.mpg.de/ippcms/de/for/bereiche/material/projekte/rpp/methoden/bayes.html>. Version: 18.12.2008. – Stand 18.12.2008.

- [MS89] MOZER, Michael C.; SMOLENSKY, Paul: Skeletonization: a technique for trimming the fat from a network via relevance assessment. In: TOURETZKY, David S. (Hrsg.): *Advances in neural information processing systems* Bd. 1. San Mateo, CA : Morgan Kaufmann Publishers Inc., 1989. – ISBN 1–558–60015–9, S. 107–115.
- [Mül08] MÜLLER, Tobias C.: Entwicklungsunterstützung Diagnose 20xx / Technische Universität Braunschweig. Braunschweig, Dezember 2008 (1). – Projektbericht.
- [Nad89] NADAL, Jean-Pierre: Study of a Growth Algorithm for a Feedforward Network. In: *International Journal of Neural Systems* 1 (1989), Nr. 1, S. 55–59.
- [Nau04] NAUMANN, David: *Benchmark Diagnosetester*, Fachhochschule Hannover, Diplomarbeit, 2004.
- [NDS07] NÄGELE, Andreas; DEJORI, Mathäus; STETTER, Martin: Bayesian Substructure Learning - Approximate Learning of Very Large Network Structures. In: *ECML '07: Proceedings of the 18th European conference on Machine Learning*. Berlin, Heidelberg : Springer-Verlag, 2007. – ISBN 978–3–540–74957–8, S. 238–249.
- [NZ94] NABHAN, Tarek M.; ZOMAYA, Albert Y.: Toward generating neural network structures for function approximation. In: *Journal of Neural Networks* 7 (1994), Nr. 1, S. 89–99. – ISSN 0893–6080.
- [NZ06] *Kapitel High Dimensional Classification with Bayesian Neural Networks and Dirichlet Diffusion Trees*. In: NEAL, Radford M.; ZHANG, Jianguo: *Studies in Fuzziness and Soft Computing*. Springer Berlin / Heidelberg, 2006. – ISBN 978–3–540–35487–1, S. 265–296.
- [Oel06] OELSCHLÄGER, Karl: *Einführung in die Stochastik*. 2006 Vorlesung WS 2006/07.
- [OW92] OMIDVAR, O. M.; WILSON., C. L.: Optimization of neural network topology and information content using Boltzmann methods. In: *Proceedings Int. Joint Conference on Neural Networks* Bd. 4, 1992, S. 594–599.
- [Pat08] PATAN, Krzysztof: *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*. Bd. 377. Springer Berlin, 2008. – ISBN 978–3–540–79871–2.
- [Pet04] PETERS, Dietmar: Erfolgsfaktor Diagnose –Stand und zukünftige Trends aus Sicht eines Automobilherstellers / Volkswagen AG. Stuttgart, 2004. – Forschungsbericht. – Euroforum-Konferenz Diagnose von E/E-Systemen im Automobil.

- [PGPB96] PUPPE, Frank; GAPPA, Ute; POECK, Karsten; BAMBERGER, Stefan: *Wissensbasierte Diagnose- und Informationssysteme*. 1. Springer-Verlag, Berlin, 1996. – 285 S. – ISBN 978-3540613695.
- [PMT04] PORTINALE, Luigi; MAGRO, Diego; TORASSO, Pietro: Multi-modal diagnosis combining case-based and model-based reasoning: a formal and experimental analysis. In: *Artificial Intelligence* 158 (2004), Nr. 2, S. 109 – 153. – ISSN 0004-3702.
- [Pol09] POLYCARPOU, Marios M. (Hrsg.): *IEEE Transactions on Neural Networks*. IEEE Computational Intelligence Society, 2009. ISSN 1045-9227.
- [PPS03] PERLICH, Claudia; PROVOST, Foster; SIMONOFF, Jeffrey S.: Tree induction vs. logistic regression: a learning-curve analysis. In: *Journal of Machine Learning Research* 4 (2003), S. 211–255. – ISSN 1532-4435.
- [Pre95] PRECHELT, Lutz: *Konstruktive neuronale Lernverfahren auf Parallelrechnern*, Universität Karlsruhe, Diss., 1995.
- [Pro08] PROKHOROV, Danil V.: Neural Networks in Automotive Applications. In: PROKHOROV, Danil V. (Hrsg.): *Computational Intelligence in Automotive Applications* Bd. 132. Springer, 2008. – ISBN 978-3-540-79256-7, S. 101–123.
- [Pup90] PUPPE, Frank: *Problemlösungsmethoden für Expertensysteme*. Springer-Verlag, Berlin, 1990.
- [Pup91] PUPPE, Frank: *Einführung in Expertensysteme*. 2. Springer-Verlag, Berlin, 1991.
- [PZMH00] PUPPE, Frank; ZIEGLER, Susanne; MARTIN, Ulrich; HUPP, Jürgen: *Wissensbasierte Diagnosesysteme im Service-support. Konzepte und Erfahrungen*. 1. Springer, Berlin, 2000. – 229 S. – ISBN 978-3540672883.
- [Pön05] PÖNICKE, Jörg: *Untersuchung von Prinzipien neuronaler und bayesscher Netzwerke für eine Anwendung bei Funktionsorientierter Diagnose mittels Prozessüberwachung*, Institut für Verbrennungsmotoren und Kraftfahrzeuge (IVK) TU-Dresden, Diplomarbeit, 2005.
- [Rap10] RAPIDMINER: *RapidMiner - Report the Future*. Website. <http://rapid-i.com>. Version: 22.03.2010.
- [Ree93] REED, Russe: Pruning algorithms: A survey. In: *IEEE Transactions on Neural Networks* (1993), Nr. 4, S. 740–747.
- [Rei92] REITER, Raymond: A theory of diagnosis from first principles. In: *Readings in model-based diagnosis*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1992. – ISBN 1-55860-249-6, S. 29–48.

- [Rei09] REIF, Konrad: *Automobilelektronik - Eine Einführung für Ingenieure*. 3. Vieweg+Teubner, 2009 (ATZ-MTZ-Fachbuch). – ISBN 978–3–8348–0446–4.
- [RH93] ROMANIUK, Steve G.; HALL, Lawrence O.: Divide and Conquer Neural Networks. In: *Journal of Neural Networks* 6 (1993), Nr. 8, S. 1105–1116. – ISSN 0893–6080.
- [Rip93] *Kapitel Statistical Aspects of Neural Networks*. In: RIPLEY, B.D.: *Statistical Aspects of Neural Networks*. Chapman and Hall, 1993. – ISBN 0–412–46530–2, S. 40–123.
- [RK06] ROUHANI-KALLEH, Omid: *Analysis, Theory and Design of Logistic Regression Classifiers Used for Very Large Scale Data Mining*. Chicago, Illinois, University of Illinois at Chicago, Diplomarbeit, 2006.
- [RM86] RUMELHART, D.E.; MCCLELLAND, J.L.: Learning Internal Representations by Error Propagation. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* 1 (1986), S. 318–362.
- [Ros62] ROSENBLATT, Frank: *Principles of neurodynamics AND perceptrons and the theory of brain mechanisms*. Spartan Books, 1962.
- [Row07] ROWLEY, Jennifer: The wisdom hierarchy: representations of the DIKW hierarchy. In: *Journal of Information Science* 33 (2007), Nr. 2, S. 163–180.
- [Rum06] RUMPE, Dr. Jutta S.: Herausforderungen an die Diagnose Integration der Diagnose in die Steuergeräteentwicklung. (2006).
- [Sac02] SACHS, Lothar: *Angewandte Statistik: Anwendung statistischer Methoden*. 10. Springer-Verlag, 2002. – ISBN 3–540–40555–0 11.
- [SAEhr] SAE: *SAE World Congress*. <http://www.sae.org/congress>. Version: jedes Jahr.
- [Sar94] SARLE, Warren S.: Neural Networks and Statistical Models. In: *in Proceedings of the Nineteenth Annual SAS Users Group International Conference*, 1994, S. 1538–1550.
- [Sch94] SCHREINER, Tobias: *Ausdünnungsverfahren für neuronale Netze*, Universität Stuttgart, Diplomarbeit, 1994.
- [Sch07] SCHUBERT, Matthias: *Datenbanken: Theorie, Entwurf und Programmierung relationaler Datenbanken*. 2. Wiesbaden : B. G. Teubner Verlag, 2007. – ISBN 978–3–8351–0163–0.
- [SD88] SIETSMA, Jocelyn; DOW, Robert J. F.: Neural network pruning – why and how. In: *Proceedings of the IEEE International Conference on Neural Networks*, IEEE, 1988, S. 325–333.

- [SD91] SIETSMA, Jocelyn; DOW, Robert J. F.: Creating artificial neural networks that generalize. In: *Journal of Neural Networks* 1 (1991), Nr. 4, S. 67–79.
- [Sha76] SHAFER, Glenn: *A Mathematical Theory of Evidence*. Princeton : Princeton University Press, 1976. – ISBN 0–608–02508–9.
- [Sho76] SHORTLIFFE, Edward H.: Computer Based Medical Consultations: MYCIN. In: *New York* (1976). ISBN 0444001794.
- [Sim93] SIMON, Natalio: *Constructive supervised learning algorithms for artificial neural networks*. Delft, Netherlands, Delft University of Technology, Department of Electrical Engineering, Master’s thesis, Juni 1993.
- [Sjø91] SJØGAARD, Steen: *A Conceptual Approach to Generalisation in Dynamic Neural Networks*. Aarhus, Danmark, Aarhus University, PhD thesis, 1991.
- [Sjø92] SJØGAARD, Steen: Generalization in cascade-correlation networks. In: *In Proceedings of the IEEE Signal Processing Workshop*, 1992, S. 59–68.
- [Smi02] SMITH, Kate; GUPTA, Jatinder (Hrsg.): *Neural Networks in Business: Techniques and Applications*. London : IRM Press, 2002. – 272 S. – ISBN 978–1930708310.
- [SMM09] SMMART: *SMMART Project*. Website. <http://www.smmart.eu/plone/smmart-project.html>. Version: 23.07.2009. – Besucht am 23.07.2009..
- [SN90] SIRAT, J. A.; NADAL, J.-P.: Neural trees: a new tool for classification. In: *Network: Computation in Neural Systems* 1 (1990), Oktober, Nr. 4, S. 423 – 438. – ISSN 1361–6536.
- [SP04] STRUSS, Peter; PRICE, Chris: Model-based systems in the automotive industry. In: *AI Magazine* 24 (2004), Nr. 4, S. 17–34. – ISSN 0738–4602.
- [Spe96] SPECKMANN, Heike: *Dem Denken abgeschaut: Neuronale Netze im praktischen Einsatz*. Braunschweig/Wiesbaden, Germany : Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 1996.
- [SRB⁺02] STRUSS, P.; REHFUS, B.; BRIGNOLO, R.; CASCIO, F.; CONSOLE, L.; DAGUE, P.; DUBOIS, P.; DRESSLER, O.; MILLET, D.: Model-based Tools for the Integration of Design and Diagnosis into a Common Process - A Project Report. In: *In Proceedings 13 th International Workshop on Principles of Diagnosis - DX02*, 2002, S. 25–32.

- [SS93] SPERDUTI, Alessandro; STARITA, Antonina: Speed up learning and network optimization with extended back propagation. In: *Journal of Neural Networks* 6 (1993), Nr. 3, S. 365–383. – ISSN 0893–6080.
- [SSB99] SCHULZ, Hans (Hrsg.); STRAUSS, Gerhard (Hrsg.); BASLER, Otto (Hrsg.): *Deutsches Fremdwörterbuch*. Bd. 4. 2. Walter de Gruyter & Co, 1999. – 971 S. – ISBN 3–11–016235–0.
- [Ste95] STEFIK, Mark: *Introduction to Knowledge Systems*. Morgan Kaufmann, 1995. – ISBN 155860166X.
- [Str02a] STREIBL, Tilo: *Entwicklung eines bildgebenden Partikeldiagnostikverfahrens zur Untersuchung Thermischer Spritzprozesse*, Universität der Bundeswehr München, Diss., 2002.
- [Str02b] STRUSS, P.: Automated Abstraction of Numerical Simulation Models - Theory and Practical Experience. In: *Proceedings of the International Workshop on Qualitative Reasoning*. Sitges, Spain, 2002.
- [Sun96] SUNA, Robert: Interaktives Lernen von Diagnose- und Inspectionsaufgaben / Forschungszentrum Informatik, Universität Karlsruhe. 1996. – Forschungsbericht.
- [Sup05] SUPKE, Jörg: Diagnose-Tools heute. In: *2. CTI Fachkonferenz - Diagnosesysteme im Automobil*, 2005.
- [SZ04] SCHÄUFFELE, Jörg; ZURAWKA, Thomas: *Automotive Software Engineering*. Vieweg, 2004.
- [Tag00] TAGSCHERER, Michael: *Dynamische Neuronale Netzarchitektur für Kontinuierliches Lernen*, Universität Chemnitz, Diss., 2000.
- [Tap04] TAPPE, Robert: *Verfahren zur Dynamisch Intelligenten Fehlersuche in komplexen Systemen*, Universität Siegen, Diss., 2004.
- [TBB⁺91] THRUN, S. B.; BALA, J.; BLOEDORN, E.; BRATKO, I.; CESTNIK, B.; CHENG, J.; JONG, K. D.; DZEROSKI, S.; FAHLMAN, S. E.; FISHER, D.; HAMANN, R.; KAUFMAN, K.; KELLER, S.; KONONENKO, I.; KREUZIGER, J.; MICHALSKI, R.S.; MITCHELL, T.; PACHOWICZ, P.; REICH, Y.; VAFAIE, H.; WELDE, W. Van D.; WENZEL, W.; WNEK, J.; ZHANG, J.: The MONK's problems: A Performance Comparison of Different Learning Algorithms / Carnegie Mellon University. 1991. – Forschungsbericht. – Technical Report CS-91-197.
- [Tew73] TEWARSON, Reginald P.: *Sparse matrices*. University of Michigan : Academic Press, 1973. – 160 S..
- [TS94] TOWELL, Geoffrey G.; SHAVLIK, Jude W.: Knowledge-based artificial neural networks. In: *Artif. Intell.* 70 (1994), Nr. 1-2, S. 119–165. – ISSN 0004–3702.

- [ULPR05] UNGER, A.; LANGE, K.; PETERS, D.; REUSS, H.-C.: Methode für eine gesamtheitliche Systembetrachtung zur funktionsorientierten Fehlererkennung und Diagnose in vernetzten Kfz-Systemen / TU Dresden, Volkswagen AG, Uni Stuttgart. 2005. – Forschungsbericht.
- [Wai89] WAIBEL, Alex: Modular construction of time-delay neural networks for speech recognition. In: *Neural Computation* 1 (1989), Nr. 1, S. 39–46. – ISSN 0899–7667.
- [WDMTM94] WANG, Zhenni; DI MASSIMO, Christine; THAM, Ming T.; MORRIS, A. J.: A procedure for determining the topology of multilayer feedforward neural networks. In: *Neural Netw.* 7 (1994), Nr. 2, S. 291–300. – ISSN 0893–6080.
- [Wer88] WERBOS, P.: Backpropagation: past and future. In: *Proceedings of the IEEE International Conference on Neural Networks*, IEEE Press, 1988, S. 343–353.
- [Wes72] WESTMEYER, Hans: *Logik der Diagnostik*. Stuttgart : Vrelag W. Kohlhammer GmbH, 1972. – 156 S. – ISBN 3–17–210151–7.
- [WH60] WIDROW, B.; HOFF, M. E.: Adaptive Switching Circuits. In: *IRE WESCON Convention Record* 4 (1960), S. 96–104.
- [Whi89] WHITE, Halbert: Learning in Artificial Neural Networks: A Statistical Perspective. In: *Journal of Neural Computation* 1 (1989), Nr. 4, S. 425–464.
- [Who63] WHORF, Benjamin L.; KRAUSSER, Peter (Hrsg.): *Sprache, Denken, Wirklichkeit*. Reinbek : Rowohlt, 1963. – 155 S. – ISBN 978–3499554032.
- [Wit08] WITTLING, Daniel: *Entwicklung und Optimierung von Anwendungs- und Validierungsmethoden für die Fehlerdiagnose mit neuronalen Netzen*. Braunschweig, TU Braunschweig, Diplomarbeit, 2008.
- [WJ93] WYNNE-JONES, Mike: Node Splitting: A Constructive Algorithm for Feed-Forward Neural Networks. In: *Journal of Neural Computing and Applications* 1 (1993), Nr. 1, S. 17–22.
- [WR06] WALLENTOWITZ, Henning (Hrsg.); REIF, Konrad (Hrsg.): *Handbuch Kraftfahrzeugelektronik: Grundlagen, Komponenten, Systeme, Anwendungen*. Wiesbaden : Vieweg, 2006. – ISBN 3–528–03971–X.
- [Wun00] WUNSCH, Gerhard: *Grundlagen der Prozesstheorie*. Bd. 1. Teubner Verlag, 2000. – 207 S. – ISBN 978–3519065241.
- [Yeu91] YEUNG, Dit-Yan: A Neural Network Approach to Constructive Induction. In: *ML*, 1991, S. 228–232.

- [Zel96] ZELL, Andreas: *Simulation neuronaler Netze*. Bonn : Addison-Wesley (Deutschland) GmbH, 1996. – ISBN 3-486-24350-0.
- [Zim94] *Kapitel Neuronale Netze als Entscheidungskalkül*. In: ZIMMERMANN, Hans G.: *Neuronale Netze in der Ökonomie: Grundlagen und finanzwirtschaftliche Anwendungen*. Vahlen Verlag, 1994. – ISBN 3800618710, S. 1–84.
- [Zsc08] ZSCHARNACK, Heiko: *Untersuchungen und Analyse zu Optimierungsmöglichkeiten bei der strukturierten Symptomabfrage in der Symptombasierten Diagnose (SBD)*, TU Dresden, Lehrstuhl Fahrzeugmechatronik, Diplomarbeit, 2008.
- [ZSWK92] ZOLLNER, R.; SCHMITZ, H. J.; WÜNSCH, F.; KREY, U.: Original Contribution: Fast generating algorithm for a general three-layer perceptron. In: *Neural Netw.* 5 (1992), Nr. 5, S. 771–777. – ISSN 0893-6080.